Comparing between Simulated Annealing Neighborhood Generation with the Combination of Simulated Annealing and Genetic Algorithm in Facility Layout Problem

by Tanti Octavia

Comparing between Simulated Annealing Neighborhood Generation with the Combination of Simulated Annealing and Genetic Algorithm in Facility Layout Problem

Victor Khoesnadi Mandala Putra, Tanti Octavia, Herry Christian Palit

Petra Christian University,Industrial Engineering, Faculty of Industrial Technology, Surabaya, Indonesia

ARTICLE INFO

ABSTRACT

We present the comparison of the Simulated Annealing Neighborhood Generation (SANG) algorithm with the combination of Genetic Algorithm-Simulated Annealing Algorithm (GA-SA II). It is applied to a facility layout problem. We simulate two algorithms using Microsoft Visual Basic 6.0 and data is generated from a case study. The result shows SANG algorithm gives a better result in terms of moment. It occurs since SANG algorithm exchanges eight chosen facilities, simultaneously. Therefore, the best moment can be found faster.

INTRODUCTION

Facility layout can affect the total moments contained in a company facility layout. Moment is obtained by multiplying the distance with frequency if it resulting smaller moments, it will impact the reduction in the cost of handling. Many researchers have developed metaheuristics models to solve the facility layout problem. They are Simulated Annealing (Chwif *et al.*, 1998; McKendall *et al.*, 2006), Genetic Algorithm (Chutima, 2001; El-Baz, 2004), and Particle Swarm Optimization (Setiawan, 2010). Setiawan (2010) compared the combination of algorithms Simulated Annealing and Genetic Algorithm (GA-SA I and GA-SA II) with Particle Swarm Optimization algorithm. The result shows that GA-SA II has a better performance than the others. Fukushi† *et al.* (2009) developed the Simulated Annealing Neighborhood Generation (SANG) to solve facility layout problem. SANG could exchange 8 facilities (4 pairs) simultaneously, however Simulated Annealing only exchange 2 facilities (1 pair) simultaneously. This research compares the performance of Simulated Annealing Neighborhood Generation algorithm with GA-SA II.

Methods:
The layout of the facility can be defined as a procedure for setting the plant facilities to support the smooth running ofthe production process(Wignjosoebroto, 2009). The main objective is to minimize total cost of the construction and installation of facilities, handling costs, and production costs. Setting the layout facility can provide other benefits besides minimize total cost which is raising production output, reduce the waiting time at each facility production process, reducing handling process, improve efficiency area, reducing the current or frequency displacement, and facilitate supervision.

Method in Facilities Layout:
Designing the layout of the facility can be done by two algorithms, construct and relayout algorithms. Construct method is a method used to design the layout of the facility. Relayout algorithm is a method used to improve the design ofthe layout of the existing facility.

Simulated Annealing:
Simulated Annealing algorithm considers the re-layout and facilities layout problems in the form of *m* rows and *n* columns (Chibante, 2010). Facility in this algorithm will be considered to have the same extensive

Corresponding Author: Victor Khoesnadi Mandala Putra, Petra Christian University,Industrial Engineering, Faculty of Industrial Technology, Surabaya, Indonesia

facilities size. This algorithm will reach a minimum transportation costs by exchanging between two facilities randomly. The iterations will continue until the minimum moment is achieved. The figure 1 shows the graph results of simulated annealing in order to reach a global minimum.

Fig. 1: Graph Results of Simulated Annealing (Imperial Collage London: Teaching, 2012)

Genetic Algorithm:
This algorithm uses the concept of Darwin's theory of evolution to find the best moments with the system of natural selection. Each individual will be regarded as a chromosome that represents one of the solutions to be evolved in stages. Good chromosomes will have off spring that will be compared with other chromosomes until the global optimum is found.

Neighborhood Generation Algorithm:
Neighborhood Generation algorithm has a similar working principle with Simulated Annealing. The difference is Simulated Annealing algorithm exchanges four pairs of different facilities simultaneously, while the Simulated Annealing algorithm just exchanges 1 pair of facilities. Neighborhood generation will minimize moment by swapping adjacent facilities both vertically and horizontally, randomly. The steps of neighborhood generation algorithm applied according to Fukushi† *et al* (2008) are: 1st) the center point facility is selected randomly. For the example, facility 6 is selected as center point facility (the coordinate is (2,3)) It can be seen in Figure 2; 2nd) the neighborhood of center point facility is found. For example, neighborhood facilities are facility 4, 14, 16, and 11; 3rd) the neighborhood facility will swap randomly for four pairs facilities, simultaneously. In Figure 3, it is shown that facility 4, 14, 16 and 11 are exchanged with the facility 3, 7, 8 and 10; 4th) the step 3 will be repeated until the smallest moment is achieved.

2	12	*14*	5
1	*4*	*6*	*16*
13	3	*11*	8
7	10	15	9

Fig. 2: Early Position Facilities (Fukushi†, Yamamoto, Suzuki, & Tsujimura, 2008).

2	12	3	5
1	7	*6*	8
13	*14*	10	*16*
4	*11*	15	9

Fig. 3: Facility Layout after Exchange (Fukushi†, Yamamoto, Suzuki, & Tsujimura, 2008).

The Principle of Simulated Annealing Neighborhood Generation:
In this research, neighborhood generation will be combined with simulated annealing algorithm. There are three step of this hybrid neighboorhood generations. The first step is to determine which facilities will be selected as the center point. Facilities which can be chosen as the center point can be seen in Figure 4, which is a facility number 6, 7, 10, and 11 (horizontal stripes) facilities.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig. 4: An example of center point facility .

The facility Selection is done randomly with the condition $m \geq 2$ from the edge of the layout and $n-2 \geq 2$ of where m is the edge of the layout rows and n is the column. Terms of the limit order facility selected as the center point is not in the edge but in the middle because on the side of the horizontal and vertical center point it will be contain neighborhood. Determination of the center point be selected at random and is the facility number 6 (diagonal right stripes), as in Figure5.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig. 5: Facility that already been chosen as center point.

The second step is to determine which facility will serve as the neighborhood. The neighborhood facility is directly adjacent facilities horizontally and vertically with the center point. Figure 6 is an example of that are the center point of the facility number6 (diagonal right stripes), with the neighborhood facility number 2, 5, 7, and10 (vertical stripes).

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig. 6: Neighborhood Facility.

The third step is to determine which facilities will be exchanged with a neighborhood. Facilities that will be exchanged, with the neighborhood facilities should not facilities that already chosen as center point and are also not the neighborhood facilities. The selection of the facility must meet the requirements and they are selected randomly. Figure 7 shows some of the facilities that can be exchanged (vertical stripes) with neighborhood facilities (diagonal left stripes). The exchange facilities are calculated and the smallest moments will be selected. The exchange between facilities will occur as in Figure 8.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig. 7: Selection of the Exchange Facilities .

Fig. 8: The Exchange of Neighborhood Facilities with Selected Facilities.

Remarks:

 The selected Facilities

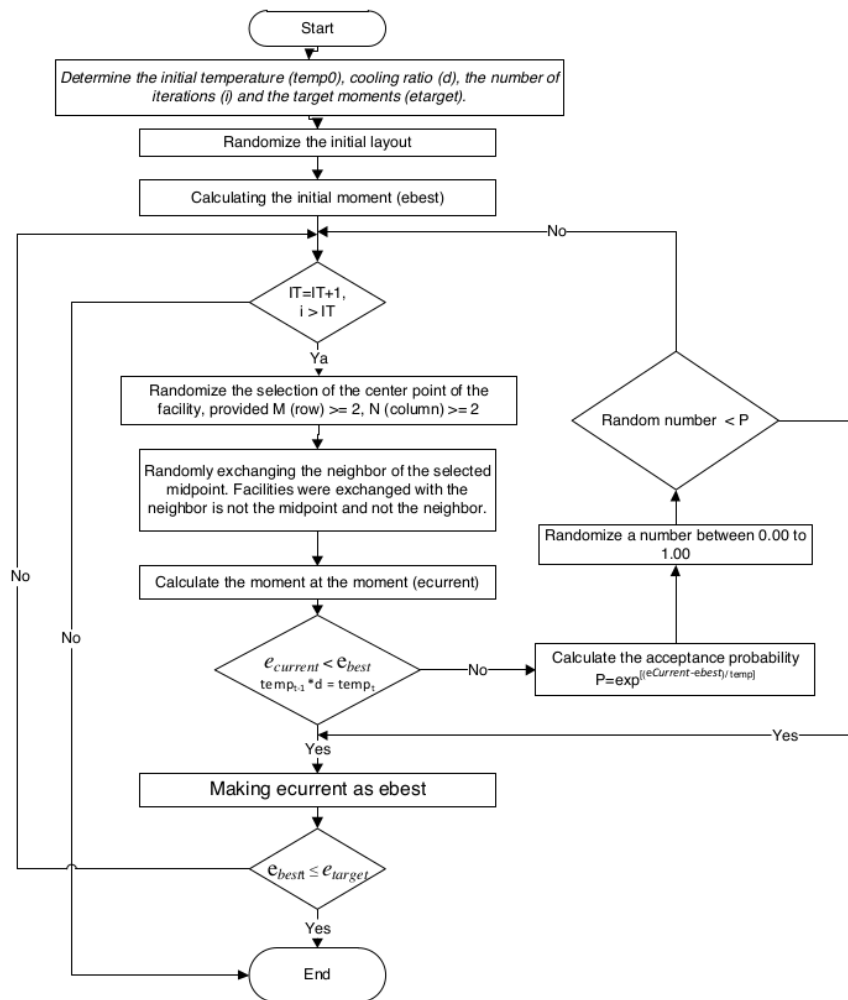 Neighborhood Facility

 Center point Facility

Fig. 9: Procedure of Simulated Annealing Neighborhood Generation (SANG) Algorithm.

Procedure of Simulated Annealing Neighborhood Generation (SANG) Algorithm:

Simulated Annealing Neighborhood Generation algorithm has a similar concept to dealing with Simulated Annealing. Figure 9 shows the procedure of Simulated Annealing Neighborhood Generation algorithm. Researchers must determine the initial temperature denoted by temp0, cooling ratio is denoted by *d*, limit the number of iterations is denoted by *i*, and the target moments are denoted by *etarget*. The initial layout is generated randomly and the moment of the layout is calculated. This initial moment will be considered as the best moment while the denoted with *ebest*. The three steps of simulated annealing neighborhood generation algorithm are applied and the moment of current layout is calculated. The current moment is denoted by *ecurrent*. The next stage is the temperature at the previous iteration will be multiplied by the cooling ratio and it will be the new temperature. The cooling ratio in this study was 0.95. This stage will compare also whether *ecurrent* is smaller than *ebest* moment or not. If the moment of *ecurrent* is smaller than the final result will be accepted and it will be *newebestvalue*. The rest, the probability of acceptance by the formula P=EXP[(ecurrent-ebest) /temp] is calculated and compared with random numbers generated between 0.00 to 1.00. If random number is smaller than the probability of acceptance, the moment of *ecurrent* is accepted and replaces the *ebest value*. *Ebest new value* will be compared with the value *etarget*, if the value is not smaller than *etarget* then the program will proceed to the next iteration.

RESULT AND DISCUSSION

The data in this study is obtained from the data from previous research (Setiawan 2010). The data taken are the number of trips between two facilities and the distance between departments. There are 18 departments and 18 facilities. Frequency of transfer material between facilities can be seen in Figure 10. In this problem considered, the matrix flow is not symmetric because of precedence constraints and the distance matrix is symmetric. For example, the frequency between facility 1 and 5 is 0 since there is no transfer of material between the facilities. The highest frequency is 382, the transfer material from facilities 11 to facilities 12 and from facilities 4 to facilities 5.

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0		0	71	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0		255	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	71	255		382	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	382		215	0	0	107	0	0	0	0	0	0	0	0	0
6	0	0	0	0	215		215	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	215		69	0	20	159	0	0	0	0	0	0	0
8	0	0	0	0	0	0	69		0	0	69	0	0	0	0	0	0	0
9	0	0	0	0	107	0	0	0		0	107	0	0	0	0	0	0	0
10	0	0	0	0	0	0	20	0	0		17	0	0	0	0	0	0	0
11	0	0	0	0	0	0	159	69	107	17		382	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	382		196	0	0	0	95	0
13	0	0	0	0	0	0	0	0	0	0	0	196		0	0	0	0	196
14	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
17	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0		0
18	0	0	0	0	0	0	0	0	0	0	0	0	196	0	0	0	0	

Fig. 10: Frequency of transfer material (Setiawan, 2010).

The distance between department is shown in Figure 11. The longest distance is 103, distance from department 15 to department 16. The SANG algorithm was simulated using Microsoft Visual Basic 6. Verification is applied in order to check whether the program has been built appropriately or not. The validation is also done by checking the similarity with the calculated results of the program manually.

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		22.5	27.5	30	33.6	47.5	52.5	35	25	55	40	59.3	74.2	30	80	22.5	68.4	85
2	22.5		20	17.5	11.1	40	45	27.5	32.5	42.5	32.5	36.8	51.7	27.5	72.5	30	60.9	62.5
3	27.5	20		17.5	25.6	20	25	12.5	22.5	42.5	12.5	46.8	61.7	17.5	52.5	50	40.9	72.5
4	30	17.5	17.5		23.1	22.5	27.5	30	40	25	25	29.3	44.2	35	55	47.5	43.4	55
5	33.6	11.1	25.6	23.1		45.6	50.6	33.6	43.6	48.1	38.1	26.5	45.6	38.6	78.1	24.4	66.5	53.1
6	47.5	40	20	22.5	45.6		20	12.5	22.5	42.5	7.5	46.8	61.7	17.5	52.5	70	38.9	72.5
7	52.5	45	25	27.5	50.6	20		32.5	42.5	22.5	27.5	26.8	41.7	37.5	32.5	75	19.9	52.5
8	35	27.5	12.5	30	33.6	12.5	32.5		10	55	5	59.3	74.2	5	65	57.5	52.4	65
9	25	32.5	22.5	40	43.6	22.5	42.5	10		65	15	69.3	84.2	5	75	47.5	62.4	95
10	55	42.5	42.5	25	48.1	42.5	22.5	55	65		50	21.6	19.2	60	30	72.5	18.4	30
11	40	32.5	12.5	25	38.1	7.5	27.5	5	15	50		54.3	69.2	10	60	62.5	47.4	80
12	59.3	36.8	46.8	29.3	26.5	46.8	26.8	59.3	69.3	21.6	54.3		19.1	64.3	51.6	50.9	40	26.6
13	74.2	51.7	61.7	44.2	45.6	61.7	41.7	74.2	84.2	19.2	69.2	19.1		79.2	32.5	70	21.8	10.8
14	30	27.5	17.5	35	38.6	17.5	37.5	5	5	60	10	64.3	79.2		70	52.5	57.4	90
15	80	72.5	52.5	55	78.1	52.5	32.5	65	75	30	60	51.6	32.5	70		103	12.6	25
16	22.5	30	50	47.5	24.4	70	75	57.5	47.5	72.5	62.5	50.9	70	52.5	103		98.9	77.5
17	68.4	60.9	40.9	43.4	66.5	38.9	19.9	52.4	62.4	18.4	47.4	40	21.8	57.4	12.6	98.9		32.6
18	85	62.5	72.5	55	53.1	72.5	52.5	65	95	30	80	26.6	10.8	90	25	77.5	32.6	

Fig. 11: The Distance Matrix between Departments (Setiawan, 2010).

The target moment each module has a different value and the target moment is obtained by calculating the lower bound of the moment.

Table 1: Input values

Initial temperature	100°C
Cooling ratio	0.95
Target moment (etarget)	22,175
Number of iterations	400 iterations

This 4x5 layout model can accommodate all the existing facilities at the beginning of the data since it can accommodate up to 20 facilities. The facilities derived from entry data there were only 18 other facilities so that two facilities will be dummy facilities. Dummy facilities will have the frequency of transfer material as 0. The ten best moments can be seen in Table 2. The average value of the initial moment and the average final moments are 89,855.04 and 43,044.91, respectively. The computational time of running this model is 0.2085 seconds.

Table 2: Ten Best Moment on 4x5 facilities.

Run	Initial Moment	Best Iteration	Final Moment	CT (second)
1	124,699	5	22,192	0.19
2	106,937	9	22,273	0.18
3	64,241	20	22,525	0.21
4	102,171	25	22,607	0.2
5	89,366	8	22,883	0.18
6	109,198	21	23,031	0.2
7	78,909	14	25,027	0.19
8	76,326	17	25,113	0.21
9	55,758	11	25,350	0.2
10	69,391	54	25,592	0.22

Comparison Moment Value:

The best moments of the value obtained from GA-SA method II and the average value of the moment are 46,653.0818 and 50,061.07, respectively (Setiawan, 2010). The comparison of moments can be seen in the Table 3. The best moment value and the average value of the moment obtained by Simulated Annealing algorithm Neighborhood Generation are 22,192 and 43,044.91, respectively. The result shows that Neighborhood Generation Simulated Annealing algorithm is smaller than the GA-SA algorithm II. Neighborhood Generation Simulated Annealing (SANG) algorithm aims various possible combinations at each iteration. Therefore, the opportunity in order to get the better result is large.

Table 3: Comparison of results Moment.

Solution	Best Moment	Average Moment
Initial Layout Company	73,533.79	-
GA-SA II	46,653.08	50,061.07
Simulated Annealing Neighborhood Generation	22,192	43,044.91

Conclusion:

In this paper we have compared Simulated Annealing Neighborhood Generation to the combination of Simulated Annealing and Genetic Algorithm in facility layout problem. We applied to a facility layout problem with eighteen locations, and simulate the algorithms for 400 iterations. The best moment value obtained with Simulated Annealing Neighborhood Generation algorithm as 22,192. The result shows that simulated Annealing Generation Neighborhood algorithm gives the better result than using GA-SA II algorithm in terms of moment. It occurs since simulated annealing neighborhood generation exchange eight chosen facilities directly. Therefore, it can lead to more extensive coverage of the exchange and the best moment can be found faster.

REFERENCES

Chibante, R., 2010. Simulated Annealing Theory with Applications. Rijeka: Sciyo.

Chutima, P., 2001. Genetic Algorithm for Facility Layout Design with Unequal Departmental Areas and Different Geometric Shape Constraints. Thammasat Int J.Sc.Tech, 6(2): 33-43.

Chwif, L., M.R.P., Barretto and L.A., Moscato., 1998. A Solution to the Facility Layout Problem Using Simulated Annealing. Computers in Industri, 36: 125-132. Elsevier.

El-Baz, M.A., 2004. A Genetic Algorithm for Facility Layout Problems of Different Manufacturing Environments. Computers and Industrial Engineering, 47: 233-246.

Fukushi†, T., H. Yamamoto., A. Suzuki and Y. Tsujimura, 2009. Efficient Algorithms for Solving Facility Layout Problem Using a New Neighborhood Generation Method Focusing on Adjacent Preference. IEMS, 8(1): 22-28

McKendall, A.R., J. Shang and S. Kuppusamy, 2006. Simulated Annealing Heuristics for the Dynamic Facility Layout Problem. Computer and Operation Research, 33: 2431-2444.

Setiawan, I.L. and H.C. Palit, 2010. PerbandinganKombinasi Genetic Algorithm-Simulated Annealing dengan Particle Swarm Optimization padaPermasalahan Tata Letak Fasilitas. Jurnal Teknik Industri UK Petra, 12(2): 119-124.

Wignjosoebroto, S., 2009. Tata Letak Pabrik dan Pemindahan Bahan. Surabaya: Guna Widya.

Comparing between Simulated Annealing Neighborhood Generation with the Combination of Simulated Annealing and Genetic Algorithm in Facility Layout Problem