Designing Machine Layout Using Tabu Search and Simulated Annealing

Pong-amorn Wangta and Pupong Pongcharoen*

Management Engineering Program, Industrial Engineering Department, Faculty of Engineering, Naresuan University, Phitsanulok, 65000

*Corresponding Author. E-mail address: pupongp@nu.ac.th, pupongp@yahoo.com

Received 23 December 2010; accepted 10 March 2011

Abstract

The machine layout design (MLD) problem usually arises when a manufacturing company aimed to expand their production capacity and/or decrease the handling distances of materials or parts flow through a predefined sequence of machines for manufacturing a product. The problem is known to be Non-deterministic Polynomial (NP) hard, which is usually solved by metaheuristics such as Simulated Annealing (SA) and Tabu Search (TS). This paper presents the application of SA and TS for minimising the material handling distance associated with the layout required for manufacturing process of multiple products. A computer based machine layout designed tool was developed and tested using five datasets adopted from literature. The analysis on the computational results obtained from numerical experiments indicated that the average best so far solutions obtained from SA are marginally better than TS but the average execution times taken by TS were at least 50% faster than SA. The convergences of best so far solutions during TS iteration were quicker than those obtained from SA especially for small- and medium-size problems.

Keywords: Simulated Annealing, Tabu Search, Metaheuristics, Facility layout, Machine Layout Design

INTRODUCTION

Manufacturing companies generally are being aware of an uncertainty issue on the changes of customers' demand. In order to remain competitive, production system should therefore be flexible in order to cope with the demand changes and technological innovations. Considering on the trade-off between product quantity and diversity, an efficient machine layout design with low material handling distance required for manufacturing various products in the same shop floor is one of the crucial aspects. This issue has a significant impact upon manufacturing costs, work in process, lead times, productivity, material handling and transportation. Up to 50% of the total operating expenses can be reduced by adopting a good facility layout (Tompkins et al., 1996). The machine layout design (MLD) problem can be viewed as a generalisation of the quadratic assignment problem and therefore belong to the class of Non-deterministic Polynomial (NP) hard problem (Drira et al., 2007), which means that the computational time required by the conventional optimisation algorithms to solve a very large problem is expensively resource consuming and reasonably impractical. Therefore, alternative nature-inspired optimisation techniques called metaheuristics are rapidly growing and being applied to solve very large NP hard problems.

There have been a number of research works focusing on designing machine or facility layout using various metaheuristics such as Ant Algorithm (Solimanpur et al, 2005), Artificial Bee Colony (Soimart & Pongcharoen, 2011), Differential Evolution Algorithm (Nearchou, 2006), Genetic Algorithms (Ariyawong, 2007; Ficko et al., 2004; Cheng & Gen, 1998), Particle Swarm Optimisation (Rezazadeh et al., 2009; Kamkhad, 2008), Rank-based Ant System (Leechai et al., 2009), Shuffled Frog Leaping Algorithm (Iamtan & Pongcharoen, 2009), Simulated Annealing (Dong et al., 2009; McKendall et al., 2006) and Tabu Search (Scholz et al., 2009).

This paper presents the development of optimisation program, in which two classical metaheuristics called Tabu Search and Simulated Annealing with four types of cooling schemes were embedded for minimising material (and part) handling distance associated with machine layout required for manufacturing various products. The machines considered in this work were non-identical, non-rotatable and rectangular shapes with predefined sizes. The optimisation program was computationally experimented using tested five benchmarking datasets adopted from literature. Finally, SA and TS performances were comparatively studied based on the quality and the convergence of the best so far solutions obtained and the computational time required.

The remaining sections in this paper are organised as follows. Section 2 presents the machine layout design

(MLD) problem and its assumptions. Simulated Annealing and Tabu Search including its pseudo codes are described in section 3 and 4, respectively. Section 5 presents the computational experiments. Finally, the conclusions are drawn in section 6.

MACHINE LAYOUT DESIGN (MLD) PROBLEM

In this work, the design task on machine layout problem involves with the process of arranging non-identical rectangular machines on the specified area of shop floor in such the way that the material handling distance totally required for manufacturing various products, each of which performs on different routes of a predefined machine sequence, is minimised. Figure 1 illustrates an example of a multiple row layout problem, in which there are 14 heterogeneous-size machines to be arranged into a limited manufacturing shop floor area. Machines are sequentially placed one by one in the first row (R1) by considering a specified aisle gap (G) between machines and the wall. Once the remaining space at the end of the row is not enough, the next machine (M3) is therefore placed on the second row (R2).

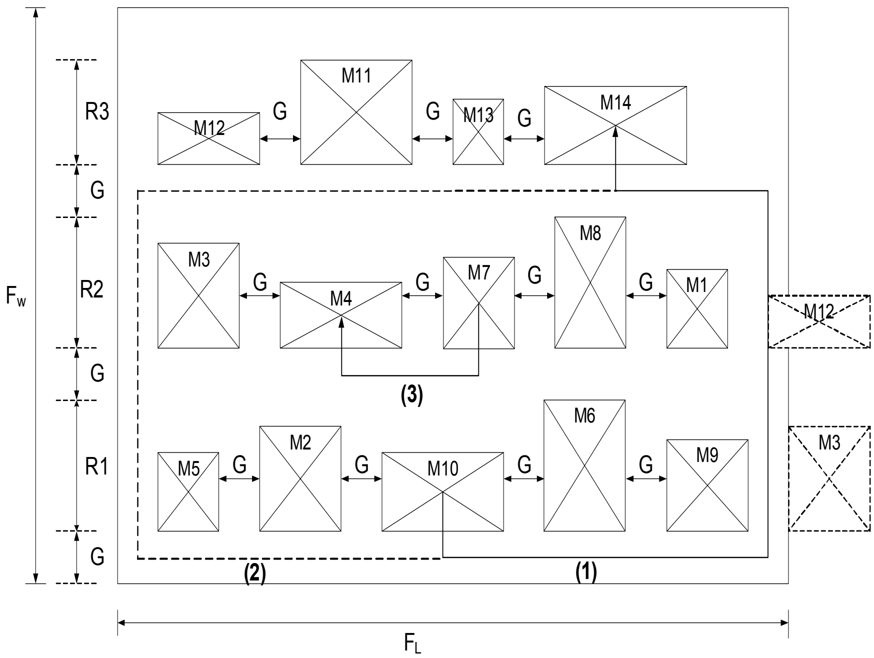

Fig. 1 Example of machine layout design and material handling path.

The measure of performance was to minimise the total material handling distance required for manufacturing products, each of which has a different route on machine sequence as shown in the equation (1).

$$\text{Minimise total distance } Z = \sum_{j=1}^{M}\sum_{i=1}^{M} f_{ij} D_{ij} \quad ; i \neq j \qquad (1)$$

Where the parameter M is the amount of machines to be arranged, the index i and j denote the machine index (i and j = 1, 2, 3, ..., M); the f_{ij} is the frequency of materials or parts flow from machine i to machine j; and the parameter D_{ij} is the travelling distance between machine i and j. It should be noted that the D_{ij} is quite straight forward when both machine i and j are in the same row. If both machines are in different rows, the parts may be moved from machine i to machine j parallel with the left or right wall. For example in figure 1, if a part is moved from machine M10 to M14, the moving path can be either left or right direction as denoted with line (1) or (2) respectively. In this work, the shorter route (for instant, the right direction or line no. 1) is taken in account for measure of performance.

There have been a variety of MLD problems. The classification of the MLD problems depends on the criterion used such as manufacturing systems, layout configurations, devices, layout evaluations, data types, objectives/performance measures and resolution approaches (Tompkins et al., 1996; Vitayasak, 2010). Considering the manufacturing systems as a factor for classification, the MLD problems can be categorised as fixed layout, process layout, product layout and cellular layout. Layout configurations can be classified as single row (e.g. linear layout, semi circular layout and U-shape layout) or multiple rows, loop layout, open-field layout and multiple floor layout. Material handling devices can be conveyor, automated guided vehicle, robot and elevator. The facility (e.g. machines and shop floor area) shapes can also be classified as regular and irregular shapes.

In this work, the following assumptions were made in order to simplify and formulate the problem: i) the size of the shop floor area must be greater than the space required for arranging the predefined machines; ii) the distance between the heterogeneous rectangular machines is determined from the machines' centroid; iii) machines are arranged in rows aligned with the x-coordinate and each row is aligned along a constant y-coordinate value; iv) the material handling devices must move in a straight line; v) the processing time and the moving time are not taken into consideration; and vi) machine cannot be oriented and the height of each machine is not taken into consideration.

SIMULATED ANNEALING (SA) ALGORITHM

Simulated annealing (SA) is a random search optimisation technique inspired by the annealing of metals proposed by Kirkpatrick et al. (Kirkpatrick et al., 1983), who was inspired by the physical annealing of solid metal. In the basic concept of annealing, a metal is first heated to a high temperature and then cooled down with a slow cooling rate into the room temperature or ground state. If the initial temperature is not high enough or if the temperature is decreased rapidly, the solid at the ground state will have many defects or imperfections. The most important components of the SA are the probability of acceptance and the annealing schedule. The probability of acceptance is defined as the probability of accepting a non-improving solution as the current solution. This is determined based on the following probability (Dong et al., 2009):

$$P = \exp(-\Delta E / T_c) \qquad (2)$$

where P is the probability of accepting the neighbour as a new solution, T_c is the current temperature and ΔE represents the difference of total distances associated with the neighbour solution and the current solution.

3.1 Cooling Schemes

The ability of the SA algorithm to provide a good solution of an optimisation problem strongly depends on the applied cooling scheme. It is important to use a relatively slow cooling scheme, in order to avoid premature convergence in local minima. The initial temperatures (T_{max} or T_1) are determined using equation (3). Three cooling schemes including Geometric (Kirkpatrick et al., 1983), Lundy & Mees (Lundy & Mees, 1986) and Linear (Menon & Gupta, 2004) considered in this work were expressed in the equation (4), (5) and (6) respectively. Moreover, we have proposed a modified Geometric cooling scheme (see more details in Wangta & Pongcharoen, 2010) as shown in the equation (7). The values of the temperature decrement factors (α, β, γ and δ) were determined with equations (8), (9), (10) and (11) where k is the iteration index (k = 1, 2, 3, ..., n), n is the number of iterations, both T_{max} or T_1 are the initial temperatures, T_{min} or T_n are the final temperatures.

$$T_{\max} = \overline{\Delta E} / \ln(0.85^{-1}) \qquad (3)$$

$$T_{k+1} = \alpha T_k \qquad (4)$$

$$T_{k+1} = T_k / (1 + \beta T_k) \qquad (5)$$

$$T_{k+1} = T_k - \gamma \qquad (6)$$

$$T_{k+1} = \delta^k T_k \qquad (7)$$

$$\alpha = (T_{\min} / T_{\max})^{1/(n-1)} \qquad (8)$$

$$\beta = (T_{\max} - T_{\min}) / [(n-1) T_{\max} T_{\min}] \qquad (9)$$

$$\gamma = (T_{\max} - T_{\min}) / (n-1) \qquad (10)$$

$$\delta = (T_{\min} / T_{\max})^{1/(\sum_{n=1}^{n-1} n)} \qquad (11)$$

3.2 Pseudo code of SA algorithm for machine layout design

Step 1: Upload problem data

Step 2: Setting parameters; n, T_{max}, T_{min}, L_{max} (number of inner loops), neighbour search operator, and cooling schemes.

Step 3: Randomly sequence the machine to create an initial solution (S_i) and calculate the distance (E_i) associated with S_i.

Step 4: Adopt a neighbour search operator (two machines random swap) S_j of S_i and calculate its distance E_j.

Step 5: If $E_j \leq E_i$, then go to step 6, otherwise go to

step 7
Step 6: Assign $S_j = S_i$ and $E_j = E_i$, then go to step 9
Step 7: Compare a uniform random variable (U) with probability (P), IF ($P > U$) then go to step 6 else go to step 8
Step 8: Reject S_j and E_j, Assign $S_i = S_i$ and $E_i = E_i$, then go to step 9
Step 9: Assign $L = L + 1$ and check inner loop, IF ($L < L_{max}$) then go to step 4, otherwise go to step 10
Step 10: Decrease temperature by cooling scheme is assigned and then go to step 11
Step 11: Check current temperature. IF ($T_c \leq T_{min}$) then go to step 12, else go to step 4
Step 12: Choose the best so far solution in overall solutions and stop criterion.

TABU SEARCH (TS) ALGORITHM

Tabu Search (TS) introduced by Glover (1989) is a well-known metaheuristic with a hill climbing concept. Tabu search process can avoid the local optimum and simultaneously continue its exploration using major components including tabu list and aspiration criteria. Tabu list is aimed to prevent a duplicated search cycle of visited solutions. Tabu list therefore keeps record of the solutions that has been previously considered. The size of the tabu list therefore plays an important role during search process. In this work, tabu list size (LS_T) was assumed to be associated with the problem size and was therefore equal to the number of machines (MC) to be arranged. Thus, when generating the neighbourhood candidates with size (LS_c), the solutions appeared in the tabu list will not be considered. If tabu list is full, the First In First Out (FIFO) rule is adopted. This means that the oldest solution recorded in the tabu list is removed whilst the new solution is inserted in the list. The role of aspiration criteria is to provide a flexibility to choose good moves by allowing the tabu move (candidates solution appeared in the tabu list) to be overridden if the aspiration level is satisfied (Glover, 1989).

4.1 Pseudo code of TS algorithm for machine layout design

Step 1: Upload problem data.
Step 2: Setting parameters; LS_C, LS_T, I_{max} (number of iterations); neighbour search operator.
Step 3: Generate a random initial layout (x) and calculate its distance $c(x)$. Update best solution and aspiration (A) values.
Step 4: Generating the neighbourhood candidates $S(x)$ of x using two machine random swap and calculate its distance $c(x)$.
Step 5: If some candidates in $S(x) \leq$ A, then go to step 6, else go to step 8.
Step 6: Choose the best solution in $S(x)$ then set the best solution is x where $x \in S(x)$.
Step 7: If $x \in$ tabu list, then go to step 12, else go to step 11.
Step 8: Remove the candidates in $S(x)$ that appear in tabu list.
Step 9: If $S(x)$ are removed until empty, then go to step 18, else go to step 10.
Step 10: Choose the best solution in $S(x)$ then set the best solution is x where $x \in S(x)$. Then go to step 11.
Step 11: Update x in tabu list.
Step 12: Update $A = c(x)$.
Step 13: If $c(x) \leq c(x^*)$, then go to step 14, else go to step 15.
Step 14: Let $x^* = x$, then go to step 16.
Step 15: Let $x = x$.
Step 16: If $I < I_{max}$, then go to step 17, else go to step 18
Step 17: Set $I = I + 1$ then return to step 4.
Step 18: Choose x^* was last improved and stop criterion.

COMPUTATIONAL EXPERIMENTS

The computation study in this work was based on five benchmarking datasets of the machine layout design problem adopted from (Nearchou, 2006; Cheng & Gen, 1998), in which the sequence of machines required for each part was predefined. The first problem, which is relatively small, was based on 3 products to be performed on 10 machines (P3M10). The second problem involved 5 products to be performed on 20 machines (P5M20). The third and forth problems considered 9 and 10 products to be performed on 15 and 30 machines, respectively (P9M15 and P10M30). Finally, the last problem involved 27 products to be processed on 30 machines (P27M30). Due to the limitation of detailed datasets available in previous research, the dimension of machines has not been provided. The width and length of each machine was therefore estimated.

The SA parameters used in this experiment were based on previous study (Wangta & Pongcharoen, 2010). For each problem size, the number of iterations (n), the final temperature (T_{min}) and the number of inner loops (L_{max}) were set at 100, 0.001 and 25, respectively. Two-machine random swap was used for neighbour search. Four cooling schemes including Geometric (Geo), Lundy & Mees (L&M) and Linear (Linear) and modified Geometric (Mo Geo) were adopted. The initial temperatures (T_{max}) for each problem size including P3M10, P5M20, P9M15, P10M30 and P27M30 were set at 126, 352, 313, 731 and 1,311 respectively. These values were determined using equation (3).

The TS parameters used in this experiment were determined as follows. The candidate list size (LS_C) and the number of iterations (I_{max}) were set at 25 and 100, respectively. The combination of both values (2,500 solutions) determines the total exploration search of the TS algorithm, which was equaled to SA. The tabu list size (LS_T) was varied and depended on the problem sizes. In this work, the LS_T is equal to the number of machine (MC) to be arranged. Again, two-machine random swap was used for neighbour search operator. A one metre gap between machines and the wall was assumed in this work. These parameters can be adjusted via graphic user interface provided in the developed program.

The developed program for designing the machine layout using the TS and SA was written in modular style using the Tool Command Language and Tool Kit (Tcl/Tk) programming language (Ousterhout, 1994). A computational experiment was conducted on personal computer with Intel Core 2 Duo 2.8 GHz CPU and 1 GB DDR2 RAM. The computational runs using TS and SA with four types of cooling scheme for each problem size were repeated 30 times using different random seed numbers. The experimental results obtained using the proposed methods for each problem sizes were summarised in terms of the best solution found (BSF), mean, standard deviation (SD) and the execution time (see Table 1).

Table 1 Results on the quality of average solution obtained and average execution time

Prob.	Algorithm	Best so far results (30 runs) Min (m.)	Max (m.)	Mean (m.)	S.D. (m.)	Time (s.)
P3M10	SA (Geo)	186.975	204.725	188.465	3.783	29
	SA (L&M)	186.975	226.225	196.793	10.519	29
	SA (Linear)	186.975	203.425	191.845	4.802	30
	SA (Mo Geo)	186.975	203.425	190.568	5.503	29
	TS	186.975	209.325	192.293	7.040	13
P5M20	SA (Geo)	1211.450	1346.550	1274.582	40.866	97
	SA (L&M)	1202.250	1382.200	1280.812	49.798	97
	SA (Linear)	1300.100	1497.150	1379.568	53.230	103
	SA (Mo Geo)	1205.200	1343.850	1261.262	34.909	100
	TS	1198.150	1357.600	1275.370	36.417	47
P9M15	SA (Geo)	1336.250	1401.350	1365.177	20.145	100
	SA (L&M)	1336.250	1441.350	1379.650	27.987	100
	SA (Linear)	1351.150	1463.150	1406.537	29.647	103
	SA (Mo Geo)	1336.250	1434.950	1372.297	30.140	101
	TS	1336.250	1406.950	1368.230	18.502	45
P10M30	SA (Geo)	4257.925	4616.275	4421.452	93.523	240
	SA (L&M)	4205.125	4618.175	4422.965	98.069	240
	SA (Linear)	4509.125	5134.025	4805.340	141.809	240
	SA (Mo Geo)	4279.125	4588.725	4408.042	81.221	240
	TS	4302.830	4676.430	4466.420	80.419	116
P27M30	SA (Geo)	8254.050	8923.650	8593.412	182.091	432
	SA (L&M)	8295.300	9220.600	8782.632	235.194	435
	SA (Linear)	8854.100	9912.100	9419.668	271.801	450
	SA (Mo Geo)	8233.200	9044.700	8618.540	164.163	435
	TS	8324.750	9155.350	8693.190	189.593	181

From Table 1, it can be seen that the performance of TS and SA with four cooling schemes were marginally different in terms of the minimum, maximum and average of the best so far solutions found for all problem sizes. However, the average execution times taken by TS were significantly quicker than SA up to 200%. In order to investigate the convergences of the best so far solutions, the computational runs that yielded the best so far (minimum) results obtained from the proposed algorithms were representatively taken from each problem dataset and plotted against the loop of generations as shown in Figure 2-6. The stopping criterion for each run was depended on the convergence of the best so far solution.

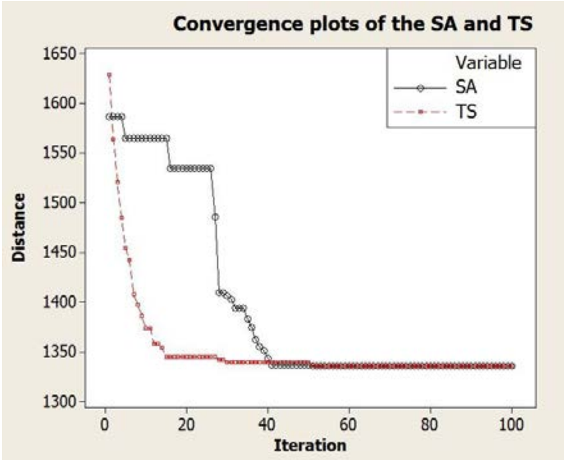

Fig. 4 Convergence plots of the SA and TS on P9M15.

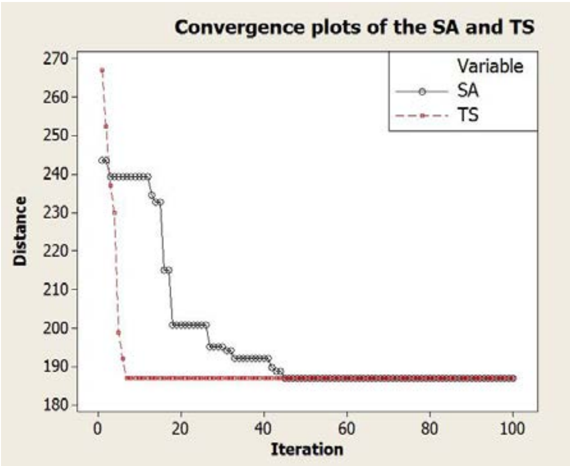

Fig. 2 Convergence plots of the SA and TS on P3M10.

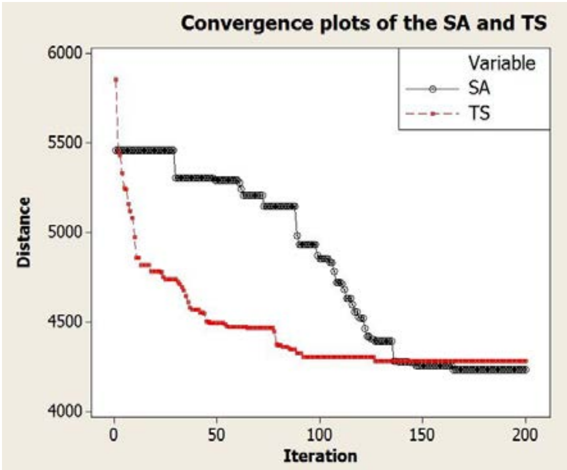

Fig. 5 Convergence plots of the SA and TS on P10M30.

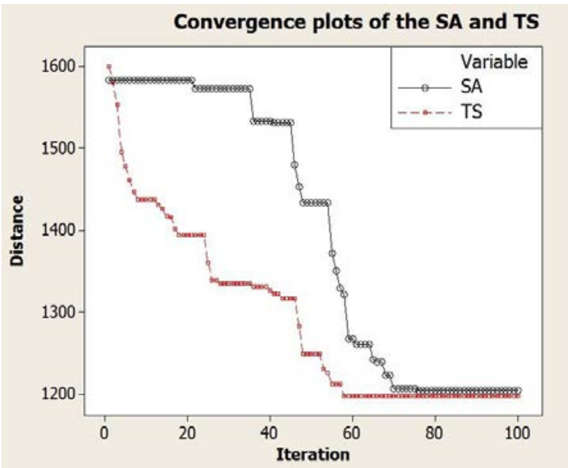

Fig. 3 Convergence plots of the SA and TS on P5M20.

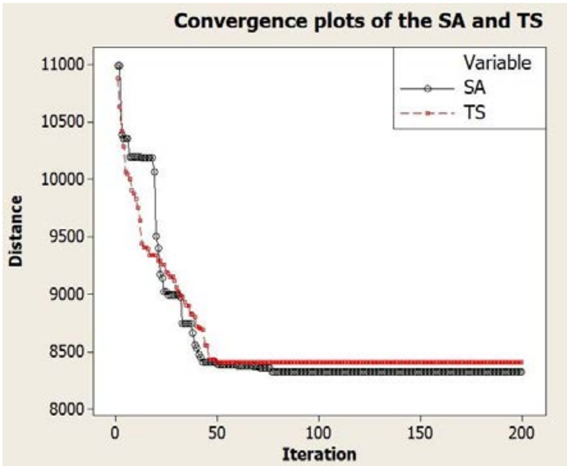

Fig. 6 Convergence plots of the SA and TS on P27M30.

From Figure 3, the best so far solution was found by TS at the 7th iteration while SA was found at 52th iteration. Figure 4, the best so far solution was found by TS at the 58th iteration while SA was found at 76th iteration. Figure 5, the best so far solution was found by TS at the 52th iteration while SA was found at 51th iteration. Figure 6, the best so far solution was found by TS at the 92th iteration while SA was found at 166th iteration. Figure 7, the best so far solution was found by TS at the 50th iteration while SA was found at 77th iteration. It can be seen that the convergences of the results obtained from the TS were dramatically quicker than the SA especially for small- and medium-size problems. However, SA yielded better results than TS for large-size problem.

CONCLUSIONS

This paper presents the application of the Tabu Search (TS) and Simulated Annealing (SA) algorithms for solving non-rotatable non-identical rectangular machine layout design (MLD) problem in multiple row environment. Four cooling schemes were considered and embedded in the SA. The proposed algorithms were aimed to minimise the material handling distance associated with the single floor layout required for manufacturing process of multiple products. The automatic layout designed program was written in modular style and computationally tested using five benchmarking datasets of the MLD problem adopted from literature. The analysis on the experimental results obtained from the proposed methods indicated the performance of TS and SA with four cooling schemes were marginally different in terms of the minimum, maximum and average of the best so far solutions found for all problem sizes. However, the convergence of the best so far solutions during iterations using TS was quicker than SA for small- and medium-size problems but SA yielded better solutions than TS for large-size problem. However, the average execution time taken by TS was less than SA by half.

REFERENCES

Ariyawong, P. (2007). A Genetic Algorithm for multiple rows layout problem in flexible manufacturing systems. Independence Study M.S., Naresuan University, Phitsanulok.

Cheng, R. & Gen, M. (1998). Loop layout design problem in flexible manufacturing systems using Genetic Algorithms. Computers & Industrial Engineering, 34(1), 53-61.

Dong, M., Wu, C. & Hou, F. (2009). Shortest path based Simulated Annealing algorithm for dynamic facility layout problem under dynamic business environment. Expert Systems with Applications, 36(8), 11221-11232.

Drira, A., Prireval, H. & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. Annual Reviews in Control, 31(2), 255-267.

Ficko, M., Brezocnik, M. & Balic, J. (2004). Designing the layout of single- and multiple-rows flexible manufacturing system by Genetic Algorithms. Journal of Materials Processing Technology, 150-158.

Glover, F. (1989). Tabu Search-Part I. ORSA Journal on Computing, 1(3), 190-206.

Kamkhad, N. (2008). Application of Particle Swam Optimisation for machine layout design in flexible manufacturing systems. Master Thesis, Naresuan University, Phitsanulok.

Iamtan, T. & Pongcharoen, P. (2009). A comparison of swap and adjustment techniques in Shuffled Frog Leaping algorithm for solving machine layout design problem. Proceedings of the Industrial Engineering Network Conference, KhonKhan, Thailand.

Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220, 671-680.

Leechai, N., Iamtan, T. & Pongcharoen, P. (2009). Designing machine layout using Rank-based Ant System and Shuffled Frog Leaping. Srinakharinwirot University Engineering Journal, 4(2), 102-115.

Lundy, M. & Mees, A. (1986). Convergence of an annealing algorithm. Mathematical Programming, 34, 111-124.

McKendall Jr., A. R., Shanga, J. & Kuppusamyb, S. (2006). Simulated Annealing heuristics for the dynamic facility layout problem. Computers & Operations Research, 33(8), 2431-2444.

Menon, S. & Gupta, R. (2004). Assigning cells to switches in cellular networks by incorporating a pricing mechanism into Simulated Annealing. IEEE Transactions on systems, man, and cybernetics-part B: Cybernetics, 34(1), 558-565.

Nearchou, A.C. (2006). Meta-heuristics from nature for the loop layout design problem. International Journal of Production Economics, 101(2), 312-328.

Ousterhout, J.K. (1994). Tcl and the Tk toolkit. Massachusetts, USA: Addison-Wesley.

Rezazadeh, H., Ghazanfari, M., Saidi-Mehrabad, M. & Sadjadi, S. J. (2009). An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem. Journal of Zhejiang University-Science A, 10(4), 520-529.

Scholz, D., Petrick, A. & Domschke, W. (2009). A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. European Journal of Operational Research, 197(1), 168-172.

Soimart, P. & Pongcharoen, P. (2011). Multi-row machine layout design using Artificial Bee Colony, Accepted for presentation in the International Conference on Economics and Business Information (ICEBI 2011), Bangkok, Thailand.

Solimanpur, M., Vrat, P. & Shankar, R. (2005). An Ant Algorithm for the single row layout problem in flexible manufacturing systems. Computers & Operations Research, 32(3), 583-598.

Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. & Trevino, J. (1996). Facilities planning. New York: Wiley.

Vitayasak, S. (2010). Facility layout problem: a 10-year review and research perspectives. Naresuan University Engineering Journal, 5(2), 46-62.

Wangta, P. & Pongcharoen, P. (2010). Investigation of cooling schemes embedded in simulated annealing for designing machine layout. Proceedings of the National Operations Research Conference, Bangkok, Thailand, 117-121.