

A new Heuristic Algorithm for the Dynamic Facility Layout Problem with Budget Constraint

Parham Azimi, Hamid Reza Charmchi

Abstract—In this research, we have developed a new efficient heuristic algorithm for the dynamic facility layout problem with budget constraint (DFLPB). This heuristic algorithm combines two mathematical programming methods such as discrete event simulation and linear integer programming (IP) to obtain a near optimum solution. In the proposed algorithm, the non-linear model of the DFLP has been changed to a pure integer programming (PIP) model. Then, the optimal solution of the PIP model has been used in a simulation model that has been designed in a similar manner as the DFLP for determining the probability of assigning a facility to a location. After a sufficient number of runs, the simulation model obtains near optimum solutions. Finally, to verify the performance of the algorithm, several test problems have been solved. The results show that the proposed algorithm is more efficient in terms of speed and accuracy than other heuristic algorithms presented in previous works found in the literature.

Keywords—Budget constraint, Dynamic facility layout problem, Integer programming, Simulation

I. INTRODUCTION

ACCORDING to [1], the problem associated with the placement of facilities in a plant area, often referred to as the “facility layout problem” is known to have a significant impact on manufacturing costs, work in processes, lead times, and company productivity. According to [2]-[3]-[4], a well-designed placement of facilities contributes to the overall efficiency of operations and can reduce up to 50% of the total operating expenses. The facility may be a manufacturing plant, warehouse, port, administrative office building or service facility. Nowadays, manufacturing plants must be able to respond quickly to changes in demand, production volume and product mix. According to [5], on average, 40% of a company’s sales come from new products. However, changes in product mixes would require modifications to the production flow, thus affecting the facility layout. According to [2] \$ 250 billion is being invested annually in the USA for reorganizing facility layouts. Many researchers have tried to take such an important issue into account when designing facility layouts. Most layout problems addressed in many research articles are implicitly considered as static; in other words, they assume that the key data of the workshop and the intended product will remain constant for a sufficiently long period. Recently, the concept of dynamic layout problems has been introduced by several researchers. According to [6]-[7], dynamic layout problems take into account possible changes in the material handling flow over multiple periods. In the dynamic facility layout problem (DFLP), the planning horizon is generally divided into certain periods which may be defined in weeks, months, or even years.

Parham Azimi, Islamic Azad University (Qazvin Branch), Faculty of Mechanical and Industrial Engineering, Qazvin, Iran, (e-mail: pazimi@yahoo.com).

Hamid Reza Charmchi, Iran Khodro Industrial Group, Sales Department, Tehran, Iran, (e-mail: charmchi@hotmail.com).

The flow data for each period are forecasted, and it is assumed that these data remain constant throughout the defined period. Therefore, the facility layout problem for each period can be considered as a static facility layout problem (SFLP) and can be solved independently. The total cost of finding a solution to the DFLP can be divided into two parts: material handling costs in each period, and rearrangement costs for facilities that need to be relocated from one period to the next. It is sub-optimal to solve the DFLP as a series of static layout problems, with one problem used separately for each period, because this approach does not consider the costs of relocating facilities from one period to the next.

Rosenblatt [8] showed the first research to develop an optimization approach based on a dynamic programming model for the DFLP. However, this approach is computationally intractable for real-life problems. The author showed that the number of layouts to be evaluated to guarantee optimality for a DFLP with N departments and T periods is $(N!)^T$. Because of the computational difficulties inherent in such a problem, several heuristics have been developed. Rosenblatt [8] proposed two heuristics that were based on dynamic programming, each of which simply considers a set of limited good layouts for a single period. Urban [9] developed a steepest-descent heuristic based on a pair-wise-exchange idea, which is similar to CRAFT. Lacksonen and Ensore [10] introduced and compared five heuristics to solve the DFLP, which were based on dynamic programming, a branch and bound algorithm, a cutting plane algorithm, cut trees, and CRAFT.

It should be mentioned that in addition to exact algorithms, many meta-heuristic algorithms have been reported in the literature such as a genetic algorithm by [11] and a tabu search (TS) heuristic by [12]. This TS heuristic is a two-stage search process that incorporates diversification and intensification strategies. Baykasoglu and Gindy [13] developed a simulated annealing (SA) heuristic for the DFLP, in which they used the upper and lower bound of the solution of a given problem instance to determine the SA parameters. Balakrishnan et al. [14] presented a hybrid genetic algorithm. Erel et al. [15] introduced a new heuristic algorithm to solve the DFLP. They used weighted flow data from various time periods to develop viable layouts, and suggested the shortest path for solving the DFLP. McKendall and Shang [16] developed three hybrid ant systems (HAS). McKendall et al. [17] introduced two (SA) heuristics. The first one, (SA I), is a direct adaptation of SA for the DFLP while the second one, (SA II), is the same as SA I, except that it incorporates an added look-ahead/look-back strategy. A hybrid meta-heuristic algorithm based on a genetic algorithm and tabu search was introduced by Rodriguez et al. [18]. Krishnan et al. [19] used a new tool, the “Dynamic From-Between Chart”, for an analysis of redesigned layouts. This tool models changes in the production rates using a continuous function. Balakrishnan and Cheng [6] investigated the performance of algorithms under fixed and rolling horizons, differing shifting costs, flow variability, and forecast

uncertainty [6]. For an extensive review on the DFLP, one can refer to the studies presented by [20] and [21]. The studies described above share a common assumption that all departments are of equal size. However, some studies do not make this assumption. Two recent examples of such studies are [22] and [23].

It should be noticed that most previous researches did not consider the company budget for rearranging the departments. Because these rearrangements are costly activities, therefore it is normal for a company to have a limited budget in this regard. According to the literature, there are just three studies on DFLP with the budget constraints; [24]-[25]-[26]. The last one which is the newest related research used a budget constraint for each period separately. They developed a simulated annealing algorithm for the problem and showed that their algorithm is more efficient than the two previous researches.

In this paper, we first introduce the problem formulation for the DFLPB in Section 1. Then, in Section 2, this model is replaced with a similar linear model that is easier to solve. In Section 3, we introduce the idea which was proposed in [27]. They used the optimal solution of their linear model as a probability distribution in a simulation model. While they used this approach toward solving the traveling salesman problem, this empirical distribution can also be used here to determine the probability of assigning facilities to certain locations. This technique has an important role in the proposed algorithm. The number of necessary runs also computed in this section. In Section 4, the proposed algorithm is introduced. In Section 5, computational results are summarized, and finally, some concluding remarks are presented in Section 6.

II. DELPB FORMULATION

The DFLP can be modeled as a modified quadratic assignment problem, similar to the static facility layout problem (SFLP). The notations used in the model are given below:

$$x_{ijt} = \begin{cases} 1 & \text{if department } i \text{ is assigned to location } j \text{ at period } t \\ 0 & \text{Otherwise} \end{cases}$$

N: Both the number of departments and the number of locations,

T: The number of periods in the planning horizon,

C_{ijkl} : Cost of material handling between department i in location j and department k in location l during period t ,

A_{ijl} : Cost of rearranging department i from location j to location l at the beginning of period t ,

LB_t : Left-over budget from period t to period $t + 1$,

B_t : Available budget for period t .

AB_t : Allocated budget for period t .

Problem 1:

$$\text{Min } z = \left(\sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} * x_{t-1,ij} * x_{itl} + \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N C_{ijkl} * x_{ij} * x_{ikl} \right) \quad (1)$$

s. t.

$$\sum_{j=1}^N x_{ij} = 1, \forall i = 1, 2, \dots, N, \forall t = 1, 2, \dots, T \quad (2)$$

$$\sum_{i=1}^N x_{ij} = 1, \forall j = 1, 2, \dots, N, \forall t = 1, 2, \dots, T \quad (3)$$

$$LB_t = B_t - \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} * x_{t-1,ij} * x_{itl}, \forall t = 1, 2, \dots, T \quad (4)$$

$$B_t = AB_t + LB_{t-1}, \forall t = 1, 2, \dots, T \quad (5)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} * x_{t-1,ij} * x_{itl} \leq B_t, \forall t = 1, 2, \dots, T \quad (6)$$

$$x_{ij} \in \{0, 1\}, \forall i, j = 1, 2, \dots, N, \forall t = 1, 2, \dots, T \quad (7)$$

$$LB_t, B_t, AB_t \geq 0, \forall t = 1, 2, \dots, T \quad (8)$$

In Problem 1, the objective function (1) is used to minimize the sum of the rearrangement and material handling costs. Constraint set (2) restricts each location to be assigned to only one department during each period and constraint set (3) ensures that exactly one department is assigned to each location within each period. Constraint set (4) is for equating the total available budget in a period to sum of the leftover budget from previous period and the allocated budget in the current period. Finally, the constraint set (5) represents the budget constraints for each period. This zero-one programming problem has been shown to be an NP-hard model [20]. To solve the problem, a linear interpolation was used to change the objective function into a linear function. This technique makes the problem easier to solve using certain degrees of accuracy. However, we demonstrate that the computational results have sufficient accuracy in comparison to the results of the previous works. Therefore, the two non-linear expressions, $x_{t-1,ij} * x_{itl}$ and $x_{ij} * x_{ikl}$, should be transformed through the linear interpolation. Assume a non-linear function as follows:

$$f(x_i, x_j) = x_i x_j \quad \text{where } 0 \leq x_i, x_j \leq 1 \quad (9)$$

The linear interpolation of the above function at point (0.5, 0.5) will be:

$$f(x_i, x_j) \cong 0.5x_i + 0.5x_j - 0.25 \quad (10)$$

Therefore, the same process will be used for the two non-linear expressions mentioned in Problem 1, while neglecting the constant (-0.25), which has no effect on the optimum solution, by introducing two new variables:

$$x_{t-1,ij} + x_{itl} = 2z_{t-1,ijl} \quad (11)$$

$$x_{ji} + x_{ikl} = 2w_{ijkl} \quad (12)$$

When the two new variables are being replaced in Problem 1 and relaxing the 0-1 variables, the resulting problem will form the following linear continuous model, which is the simplest model in mathematical programming theory:

Problem 2:

$$\text{Min } z = \left(\sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} * z_{t-1,ijl} + \right. \quad (13)$$

$$\left. \sum_{t=2}^T \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N C_{ijkl} * w_{ijkl} \right)$$

s.t.

$$\sum_{j=1}^N x_{ij} = 1, \forall i = 1, 2, \dots, N, \forall t = 1, 2, \dots, T \quad (14)$$

$$\sum_{i=1}^N x_{ij} = 1, \forall j = 1, 2, \dots, N, \forall t = 1, 2, \dots, T \quad (15)$$

$$x_{t-1,ij} + x_{t,il} = 2z_{t-1,ijl}, \forall i, j, l = 1, 2, \dots, N, \forall t = 2, 3, \dots, T \quad (16)$$

$$x_{t,ij} + x_{t,kl} = 2w_{ijkl}, \forall i, j, k, l = 1, 2, \dots, N, \quad (17)$$

$$\forall t = 1, 2, \dots, T, i \neq j, k \neq l$$

$$LB_t = B_t - \sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} * z_{t-1,ijl}, \forall t = 1, 2, \dots, T \quad (18)$$

$$B_t = AB_t + LB_{t-1}, \forall t = 1, 2, \dots, T \quad (19)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{l=1}^N A_{ijl} * z_{t-1,ijl} \leq B_t, \forall t = 1, 2, \dots, T \quad (20)$$

$$0 \leq x_{ij} \leq 1, \forall i, j = 1, 2, \dots, N, \forall t = 1, 2, \dots, T \quad (21)$$

$$0 \leq z_{t-1,ijl} \leq 1, \forall i, j, l = 1, 2, \dots, N, \forall t = 2, 3, \dots, T \quad (22)$$

$$0 \leq w_{ijkl} \leq 1, \forall i, j, k, l = 1, 2, \dots, N, \forall t = 1, 2, \dots, T, \quad (23)$$

$$, i \neq j, k \neq l$$

$$LB_t, B_t, AB_t \geq 0, \forall t = 1, 2, \dots, T \quad (24)$$

Problem 2 has more variables and constraints compared to Problem 1, but because this is a linear model, the computational time will be much lower than Problem 1, according to the computational results.

III. INPUT VALUES FOR SIMULATION MODEL

Before describing the simulation model for the DFLPB, an idea that was developed by [27] is presented for the traveling salesman problem. In this idea, because all variables in Problem 2 are between zero and one, and according to constraints (14) and/or (15), their summations are equal to 1,

the definition of x_{ij} can be interpreted as the probability of assigning department i to location j during period t . As mentioned before, this concept plays a key role in the simulation model, where it must use a probability distribution for randomly assigning each facility to each location during a certain period. In fact, the optimal values which come from Problem 2 are empirical distributions that will be used by the simulation model. Because Problem 2 is an estimation of the real problem, these empirical distributions help us to reduce the simulation runs in order to find the best solution faster. Because the algorithm uses a linear interpolation, the optimal solution of Problem 2 will be an estimator for the real values. Now, suppose that the optimal solution of Problem 1 is known and x^*_{ij} is the optimal solution of Problem 2.

Define A_t as the probability of finding the optimum solution of Problem 1 at period t as follows:

$$A_t = \prod_i \prod_j x^*_{ij} \quad (25)$$

To find the optimal assignment values for a maximum n_t runs in the simulation model, with $\%(1 - \alpha)$ as the significance level:

$$A_t + (1 - A_t)A_t + (1 - A_t)^2 A_t + \dots + (1 - A_t)^{n_t-1} A_t \geq 1 - \alpha \quad (26)$$

Then:

$$n_t \log(1 - A_t) \leq \log(\alpha) \quad (27)$$

Therefore:

$$n_t \geq \frac{\log(\alpha)}{\log(1 - A_t)} \quad (28)$$

Because n_t is the number of minimum needed runs to obtain optimal solution at period t , NR can be defined as the number of minimum needed runs, then we have:

$$NR = \text{Max}_t \{n_t\} \quad (29)$$

For calculating A_t and NR, a heuristic algorithm has been developed as follows:

Step 0) Define $K \subset N$ as a subset of facilities that have been assigned to the locations yet and set $K = \{ \}$. Denote P as the set of correspondent probabilities of the elements in K .

Step 1) Solve Problem 2 and find the optimal solutions as x^*_{ij} .

Step 2) Select a facility that has the maximum value for the assignment and, if there is a tie, then select a facility randomly. Assume that facility m has the maximum assignment value among all x^*_{ij} , which is denoted by x^*_{mj} .

Now, add facility m to set K and x^*_{mj} to P .

Step 3) If $K = N$, then go to Step 4; otherwise, go back to Problem 2. For each facility in set K , set the correspondent assignment variables to 1, and then add these new constraints to Problem 2. Go to Step 1.

Step 4) Now, A_t can be estimated using the product of all elements in set P according to (25).

And the value for NR can be computed based on (29).

As it will be shown in the computational results:

- The value of NR is relatively low in comparison to other developed meta-heuristics which results in quick run times. This is because, the optimal values obtained from Problem 2 are near to the optimal solutions, and their values are greater than 0.76 in most cases. This fact will cause NR to be sufficiently low in the computations based on (28).

- Because the simulation model does not depend on simulation clock, all runs will be completed very quickly (less than 0.01 min. in most cases). Therefore, this effective criterion will help the simulation find the near optimum solutions as quickly as possible.

The simulation model has been designed according to Problem 1. All constraints have coded using Enterprise Dynamics 8.1 software with 4DScript language. All input parameters such as the rearrangements costs, material handling costs, and the optimal solutions of Problem 2 are stored in tables within the software, and when the software assigns a facility to a location at each period, the related costs will be stored in another table which calculates the objective function of Problem 1 at the end of each run. This simulation technique is a good tool for such a difficult problem because it produces feasible solutions based on Problem 1, i.e. all solutions produced by the simulation model are integers and satisfy all mentioned constraints. All necessary runs are conducted within the experimentation wizard in the simulation software. When a run is completed, the resulting feasible solutions will be stored in a table. Therefore when an experiment ends after NR runs, one can access the best solutions based on the data set stored in the table. Because, the simulation model does not depend on a time process, it does not need to calculate a warm-up period.

IV. PROPOSED ALGORITHM

Based on the previous explanations, the heuristic algorithm can be defined as follows:

Step 1) Initialize:

Assume that we have a DFLPB with N facilities, N locations at T periods, in which all other parameters such as C_{ijkl} ,

A_{ijl} , are given. Formulate Problem 2 as previously described and code the simulation model according to Problem 1. During this step, assume that the significance level is $\%(1 - \alpha)$.

Step 2) Calculate the assignment probabilities:

Solve Problem 2. The optimal solutions of this problem will be used as an empirical distribution for assigning a facility to a location during each period.

Step 3) Calculate the minimum number of needed runs:

According to the results of Step 2, calculate the minimum number of needed runs (NR) based on (29).

Step 4) Run the simulation model:

Run the simulation model with NR replications. At each replication r, store the objective function, Z_r , and the corresponding assignments for each period.

Step 5) Find the best solution:

The best solution is determined as:

$$Z^* = \text{Min}_{.r} \{Z_r\} \quad (30)$$

V. COMPUTATIONAL RESULTS

As mentioned before, Enterprise Dynamics 8.1 software has been used for the simulation, Lingo 8.0 for finding the optimal solution of Problem 2, and Microsoft Visual Basic

2007 as the coordinator between the simulation software and mathematical programming software. All computations were run on a PC with a 4.8 GHz CPU and 4GB of RAM. All parameters for the DFLPBs were taken from a data set provided by [11]. For comparison, the results have been compared with those reported in [26]. Their report has been selected, because they had compared their results with two previous papers and showed that their proposed heuristic algorithm was the best one among all of others. The computational results are summarized in Table 1 over a wide range of test problems (48 problems). These problems contain cases using 6, 15, and 30 facilities (N = 6, 15, 30) each with 5 and 10 periods. The second column in this table is the problem number denoted by [26]. For each problem without a budget constraint, [25] shows three problems with different budget constraints. First, a total budget constraint is obtained by solving an unconstrained problem with respect to the budget, and by setting the total rearrangement cost of this solution as the total budget constraint. Then the allocation of this total budget is carried out in three ways: (1) Divide the total budget by the number of periods- 1 (number of transactions), and allocate equally to the periods. (2) The level of the budget for each period is found by taking the half of the rearrangement cost for the same period in solution of the unconstrained problem. (3) The level of the budget for each period is found by adding 10% more to the rearrangement costs for the same period in the solution of the unconstrained problem. Through this process three sets of problems are obtained which were denoted by 1, 2, and 3 in the third column. For the exact parameter values of the obtained problems please see [25]. The forth column, labeled "Average Probabilities", lists the average optimal solutions obtained from Problem 2. The number of needed runs is listed in the fifth column. The optimal solutions under the proposed algorithm are listed in the sixth column. The seventh column lists the optimal solutions reported by [26]. The percentage of deviation, denoted by "%Dev." of the best solution obtained from the proposed algorithm, which is lower than the best solution obtained from [26], is given in the eighth column, for each test problem. In the last column, the average run times are given in minutes.

Another important factor regarding the proposed algorithm is the "Average CPU Time" which is sufficiently fast for use in these applications. As previously explained, the simulation time depends on many factors such as NR, N, and T. However, [26] did not report any running time in their experiments. As listed in the table, the average percentage of deviation is -1.89% for the solved problems and the average run time is 0.33 min. The average probabilities in the forth column is 0.8348 which shows the effectiveness of mathematical programming for estimating the optimal solutions. In Tables II and III, the same structures and calculations are given for N = 15 and N = 30 respectively.

As listed in Table II, in certain cases both algorithms have the same best solutions, but the average is -1.70%, i.e. the proposed algorithm is better than that of [26] when N = 15. In addition, the running time of the algorithm is reasonable in both cases.

In Table III, again in some cases both algorithms have the same best solutions, but the average is -1.25%, i.e. the

algorithm has better solutions than that of [26] when N=30. Meanwhile, the speed of the algorithm is around 1 h. This fact is very important, because according to [26], for such a large size problems, their algorithm took several hours.

To sum up, the proposed algorithm provides good solution quality in comparison to the algorithms developed in previous

researches. It was able to improve the optimal solution for a known data-set by 1.72% on average. Regarding the run time, the algorithm has reasonable run time in comparison to previous researches.

TABLE I
 THE COMPUTATIONAL RESULTS WITH SIGNIFICANT LEVEL 99%
 WITH N=6

T	Problem Number	Budget	Average probabilities	NR	Best Solution	Best Solution by Sahin et al.	% Dev.	Average Time
5	P01	1	0.7608	21	106419	106419	0	0.53
		2	0.7984	15	106419	106419	0	0.38
		3	0.8432	10	106419	106419	0	0.26
	P02	1	0.7745	19	105731	105731	0	0.47
		2	0.8432	10	105731	105731	0	0.26
		3	0.8004	15	103429	104834	-1.34	0.38
	P03	1	0.8346	11	103541	106011	-2.33	0.28
		2	0.8343	11	106049	107609	-1.45	0.28
		3	0.7954	16	102092	105762	-3.47	0.39
	P04	1	0.7922	16	106547	106583	-0.03	0.41
		2	0.7732	19	107984	107984	0	0.48
		3	0.851	10	106906	106906	0	0.24
P05	1	0.8848	7	104786	106328	-1.45	0.18	
	2	0.8679	8	107870	107870	0	0.21	
	3	0.821	13	106285	106.328	-0.04	0.31	
P06	1	0.8847	7	104315	104315	0	0.18	
	2	0.8771	8	107698	107698	0	0.19	
	3	0.8213	13	104001	104262	-0.25	0.31	
P07	1	0.9235	5	103582	107406	-3.56	0.12	
	2	0.9133	5	104752	108114	-3.11	0.13	
	3	0.9011	6	106173	106439	-0.25	0.15	
P08	1	0.8261	12	107248	107248	0	0.3	
	2	0.8022	15	107248	107248	0	0.37	
	3	0.8045	15	107248	107248	0	0.36	
P09	1	0.7892	17	220301	220367	-0.03	0.52	
	2	0.7954	16	220776	220776	0	0.49	
	3	0.8124	14	217251	217251	0	0.42	
P10	1	0.8103	14	216607	217106	-0.23	0.43	
	2	0.8092	14	216767	217201	-0.2	0.43	
	3	0.8674	8	211837	212134	-0.14	0.26	
P11	1	0.8464	10	211951	214960	-1.4	0.31	
	2	0.8522	10	206178	215622	-4.38	0.3	
	3	0.8775	8	215393	215393	0	0.23	
P12	1	0.8923	7	216828	216828	0	0.2	

P13	2	0.8955	6	216828	216828	0	0.2
	3	0.798	15	216828	216828	0	0.48
	1	0.8563	9	205695	211620	-2.8	0.28
P14	2	0.8543	9	210958	213304	-1.1	0.29
	3	0.8439	10	205060	211620	-3.1	0.32
	1	0.7845	17	211916	212341	-0.2	0.54
P15	2	0.798	15	207966	213430	-2.56	0.48
	3	0.8238	12	205335	213424	-3.79	0.38
	1	0.8842	7	217221	217460	-0.11	0.22
P16	2	0.8906	7	218291	218794	-0.23	0.21
	3	0.8578	9	214136	214823	-0.32	0.28
	1	0.77	20	171712	220144	-22	0.61
Average	2	0.7903	16	189324	220144	-14	0.51
	3	0.8431	10	181917	219177	-17	0.32
	Average	0.8348	12	157616	161343	1.89	0.33

TABLE II
 THE COMPUTATIONAL RESULTS WITH SIGNIFICANT LEVEL 99%
 WITH N=15

T	Problem Number	Budget	Average probabilities	NR	Best Solution	Best Solution by Sahin et al.	% Dev.	Average Time
5	P17	1	0.9306	11	481675	481675	0	0.48
		2	0.88	29	480208	481682	-0.31	1.25
		3	0.7845	173	494401	480453	2.9	7.45
	P18	1	0.8758	31	468932	484799	-3.27	1.35
		2	0.7849	172	483921	490290	-1.3	7.39
		3	0.782	182	478213	486726	-1.75	7.82
	P19	1	0.8221	85	474661	489583	-3.05	3.64
		2	0.8762	31	492274	493018	-0.15	1.34
		3	0.9911	2	489450	489450	0	0.1
	P20	1	0.9317	11	477414	484876	-1.54	0.47
		2	0.9618	6	484856	489912	-1.03	0.24
		3	0.8198	88	470294	484954	-3.02	3.8
P21	1	0.8261	79	475885	488262	-2.54	3.38	
	2	0.811	104	476112	487935	-2.42	4.49	
	3	0.8148	97	469153	487822	-3.83	4.18	
P22	1	0.9153	15	473,148	486493	-2.74	0.64	
	2	0.8581	43	473392	488199	-3.03	1.87	
	3	0.9523	7	485532	487360	-0.37	0.3	
P23	1	0.9302	11	458388	478000	-4.1	0.48	

	2	0.8334	69	466110	487007	-4.29	2.95
		0.8044	118	467295	486801	-4.01	5.08
		0.8628	40	480468	491080	-2.16	1.71
P24	1	0.8628	40	480468	491080	-2.16	1.71
	2	0.9481	8	489292	494369	-1.03	0.33
	3	0.9867	3	476618	491237	-2.98	0.12
P25	1	0.9476	8	939786	981531	-4.25	0.38
	2	0.8936	23	985031	985031	0	1.1
	3	0.9821	3	979638	979638	0	0.16
P26	1	0.906	18	979655	979655	0	0.87
	2	0.7792	192	955783	981478	-2.62	9.41
	3	0.954	7	952918	977462	-2.51	0.33
P27	1	0.9215	13	955190	984103	-2.94	0.65
	2	0.9272	12	972096	993049	-2.11	0.58
	3	0.8726	33	960196	983112	-2.33	1.63
P28	1	0.9512	7	950604	971759	-2.18	0.35
	2	0.8484	52	974385	974385	0	2.54
	3	0.9101	17	973223	974792	-0.16	0.81
P29	1	0.7854	170	936480	978456	-4.29	8.34
	2	0.9871	3	980346	980346	0	0.13
	3	0.7638	260	947673	978748	-3.18	12.73
P30	1	0.782	182	949566	970024	-2.11	8.91
	2	0.793	147	972765	972765	0	7.2
	3	0.7929	147	969998	970435	-0.04	7.22
P31	1	0.7887	160	962403	978549	-1.65	7.83
	2	0.8457	55	990976	990976	0	2.67
	3	0.7747	210	979339	979339	0	10.27
P32	1	0.8746	32	971053	985001	-1.42	1.57
	2	0.8432	57	958486	986493	-2.84	2.8
	3	0.9894	2	977270	985817	-0.87	0.12
Average		0.8729	67	721720	733644	-1.7	3.19

TABLE III

THE COMPUTATIONAL RESULTS WITH SIGNIFICANT LEVEL 99% WITH N=30

T	Problem Number	Budget	Average probabilities	NR	Best Solution	Best Solution by Sahin et al.	% Dev.	Average Time
5	P33	1	0.887	166	576451	577086	-0.11	10.27
		2	0.9611	13	579704	579704	0	0.79
		3	0.9256	44	577493	577493	0	2.76
	P34	1	0.9588	14	551951	571846	-3.48	0.86
		2	0.995	2	559139	572396	-2.32	0.15
		3	0.9009	103	556359	570.537	-2.49	6.39
	P35	1	0.8387	899	566291	579113	-2.21	55.76
		2	0.8735	264	556438	579406	-3.96	16.37

	3	0.887	166	566301	574225	-1.38	10.28
		0.9824	5	557872	572964	-2.63	0.32
		0.9731	8	554936	578631	-4.09	0.49
P36	1	0.9824	5	557872	572964	-2.63	0.32
	2	0.9731	8	554936	578631	-4.09	0.49
	3	0.9262	44	545506	569880	-4.28	2.7
P37	1	0.9224	50	552347	559934	-1.35	3.08
	2	0.9568	15	551905	559078	-1.28	0.92
	3	0.9379	29	555069	559506	-0.79	1.81
P38	1	0.9888	4	544879	569457	-4.32	0.23
	2	0.941	26	559640	567166	-1.33	1.62
	3	0.8689	310	546839	567749	-3.68	19.2
P39	1	0.8817	199	569470	569470	0	12.33
	2	0.7843	6740	570521	570521	0	417.85
	3	0.9374	30	563648	569382	-1.01	1.84
P40	1	0.992	3	556582	579411	-3.94	0.19
	2	0.927	42	565906	586310	-3.48	2.63
	3	0.9413	26	560792	577719	-2.93	1.61
P41	1	0.9291	39	1133743	1171634	-3.23	2.88
	2	0.986	4	1155647	1172520	-1.44	0.32
	3	0.9198	54	1129068	1171500	-3.62	3.96
P42	1	0.9475	21	1166613	1174896	-0.71	1.52
	2	0.966	11	1137578	1175998	-3.27	0.77
	3	0.9457	22	1162838	1177009	-1.2	1.62
P43	1	0.9223	50	1169208	1169208	0	3.63
	2	0.8739	260	1179660	1179660	0	19.01
	3	0.8867	167	1134677	1164129	-2.53	12.23
P44	1	0.9085	80	1140598	1151468	-0.94	5.81
	2	0.7854	6462	1152874	1152874	0	471.72
	3	0.9123	70	1122006	1147234	-2.2	5.11
P45	1	0.9703	9	1114861	1127044	-1.08	0.65
	2	0.858	453	1141881	1141881	0	33.09
	3	0.8781	225	1128472	1129703	-0.11	16.44
P46	1	0.8762	240	1132099	1146000	-1.21	17.55
	2	0.7867	6149	1154691	1154691	0	448.88
	3	0.8583	449	1145044	1145858	-0.07	32.75
P47	1	0.778	8585	1210573	1210573	0	626.71
	2	0.9444	23	1210573	1210573	0	1.7
	3	0.8937	132	1210573	1210573	0	9.62
P48	1	0.7786	8389	1199048	1189154	0.83	612.37
	2	0.872	278	1152896	1201885	-4.08	20.3
	3	0.9003	105	1181360	1181360	0	7.68
Average		0.9076	864	858596	870759	-1.58	63.02

VI. CONCLUSIONS

In this research, a new heuristic algorithm for solving the dynamic facility layout problem with budget constraint was developed by combining mathematical programming and simulation methods. The first contribution of the current study is that it defines the optimal solution of the linear programming model in terms of empirical distributions for a simulation model. This idea can decrease the number of replications required in the simulation model, leading to better speed. The performance of the proposed algorithm was tested over a wide range of test problems taken from the literature. The proposed algorithm improved the objective function of the problem by 1.72% on average, whereas the time required for the largest problem with $N = 30$ and $T = 10$ was around 1 hour. This is the second contribution of the current research. The proposed algorithm not only avoids uncommon issues in meta-heuristic algorithms such as premature events, parameter tuning and trapping in local optimums but also uses a simulation technique that produces feasible solutions without the use of any specific nonrealistic assumptions. Regarding the constraints, inherent in this kind of research, we think that if we use the new version of Lingo software and run the algorithm on a faster computer (in particular, one with a faster CPU) the results will be further improved. Finally, for the future works, we strongly suggest concentrating on a cost sensitivity-process (including the rearrangement and material handling costs), which will occur in future periods and have a great influence on the optimal solution. As a suggestion, fuzzy costs may be useful under uncertainty conditions, or at least the time value of the monetary investment must be considered.

REFERENCES

- [1] Drira, A, Pierreval, H., Hajri-Gabouj, H., 2007. Facility layout problems: A survey. *Annual Reviews in Control*. 31, 255–267.
- [2] Tompkins, J., White, J., Bozer, Y., Tanchoco, J., 2003. *Facilities planning*. third ed. John Wiley & Sons, New Jersey.
- [3] Dong, M., Wua, C., Hou, F., 2009. Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment. *Expert Systems with Applications*. 36, 11221–11232.
- [4] Gary, Y., Chen, K.J., Rogers, A., 2009. Multi-objective evaluation of dynamic facility layout using ant colony optimization. *Proceedings of the 2009 Industrial Engineering Research Conference*.
- [5] Page, A.L., 1991. New product development survey: Performance and best practices. PDMA conference.
- [6] Balakrishnan, J., Cheng, C.H., 2009. The dynamic plant layout problem: Incorporating rolling horizons and forecast uncertainty. *Omega*, 37, 165–177.
- [7] Kouvelis, P., Kurawarwala, A. A., Gutierrez, G. J., 1992. Algorithms for robust single and multiple periods layout planning for manufacturing systems. *European Journal of Operations Research*. 632, 287–303
- [8] Rosenblatt M.J., 1986. The dynamics of plant layout. *Management Science*. 321, 76–86.
- [9] Urban, T. L., 1998. Solution procedures for the dynamic facility layout problem. *Annals of operations research*. 761, 323–342.
- [10] Lacksonen, T.A., Enscore, E.E., 1993. Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research*. 313, 503–17.
- [11] Balakrishnan, J., Cheng, C. H., 2000. Genetic search and the dynamic layout problem: An improved algorithm. *Computers and Operations Research* 276, 587–593.
- [12] Kaku, B., Mazzola, J. B., 1997. A tabu search heuristic for the plant layout problem. *INFORMS Journal on Computing*. 94, 374–384.
- [13] Baykasoglu, A., Gindy, N. N. Z., 2001. A simulated annealing algorithm for dynamic facility layout problem. *Computers and Operations Research*. 2814, 1403–1426.
- [14] Balakrishnan, J., Cheng, C.H., Daniel, G. Conway, A., Lau, C.M., 2003. A hybrid genetic algorithm for the dynamic plant layout problem. *International Journal of Production Economics*. 86, 107–120.
- [15] Erel, J.B., Ghosh, J., Simon, J.T., 2005. New heuristic for the dynamic layout problem. *Journal of the Operational Research Society*. 568, 1001.
- [16] McKendall, A.R., Shang, J., 2006a. Hybrid ant systems for the dynamic facility layout problem. *Computers & Operations Research*. 333, 790–803.
- [17] McKendall, A.R., Shang, J., Kuppusamy, S., 2006b. Simulated annealing heuristics for the dynamic facility layout problem, *Computers & Operations Research*. 33, 2431–2444.
- [18] Rodriguez, J.M., MacPhee, F.C., Bonham, D.J., Bhavsar, V.C., 2006. Solving the dynamic plant layout problem using a new hybrid meta-heuristic algorithm. *International Journal of High Performance Computing and Networking*. 45/6, 286–294.
- [19] Krishnan, K.K., Cheraghi, S.H., Nayak, C. N., 2006. Dynamic from-between charts: A new tool for solving dynamic facility layout problems. *International Journal of Industrial and Systems Engineering*. 11/2, 182–200.
- [20] Balakrishnan, J., Cheng, C. H., 1998. Dynamic layout Algorithms: A state-of-the art survey. *Omega*. 264, 507–521.
- [21] Kulturel-Konak, S., 2007. Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st century. *Journal of Intelligent Manufacturing*. 182, 273–284.
- [22] Dunker, T., Radons, G., Westkamper, E., 2005. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. *European Journal of Operational Research*. 1651, 55–69.
- [23] McKendall A.R., Hakobyan, A., 2010. Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research*. 201, 171–182.
- [24] Balakrishnan, J., Jacobs, F. R., Venkataramanan, M. A., 1992. Solutions for the constrained dynamic facility layout problem. *European Journal of Operational Research*, 57(2), 280–286.
- [25] Baykasoglu, A., Dereli, T., Sabuncu, I., 2006. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega*, 34(4), 385–396.
- [26] Sahin, R., Ertogral, K., Turkbey, O., 2010. A simulated annealing heuristic for the dynamic facility layout problem with budget constraint. *Computers & Industrial Engineering*, 59, 308–313.
- [27] Azimi P., Daneshvar Ghorbai P., 2010. An efficient heuristic algorithm for the traveling salesman problem. 8th Heinz Nixdorf Symposium in Advanced Manufacturing and Sustainable logistics, Paderborn, Germany.