2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Solving Dynamic Double-row Layout Problem via an Improved Simulated Annealing Algorithm

Shengli Wang, Xingquan Zuo, Xinchao Zhao

***Abstract*—Double-row layout problem (DRLP) is a new problem proposed in 2010. Different from single or multi-row layout problems, DRLP needs to determine not only the sequence of machines on both rows but also the exact location of each machine. Aiming at the dynamic environment of product processing in practice, in this paper we study DRLP under dynamic environment and propose a dynamic double-row layout problem (DDRLP) where the material flows may change over time. A mixed-integer programming model is established for the DDRLP. An improved simulated annealing (ISA) algorithm is proposed to for this problem. To represent a feasible solution, a mixed coding scheme is suggested to express the sequence of facilities and the exact location of each facility. Five operators are devised to make the ISA able to effectively solve this problem. Experiment results show that the proposed algorithm is able to find the optimal solutions for small size problem instances and outperform an exact approach (CPLEX) under limited run time for large size instances.**

I. INTRODUCTION

Facility layout problems (FLP) drew much attention over the years. It is to place a number of facilities in a production workshop to achieve such objectives as saving operational cost, enhancing production efficiency and balancing device capabilities.

For a manufacturing system, materials flows among facilities play an important role to achieve a superior layout. In a static facility layout problem (SFLP), the material flows between any two facilities are known and fixed. SRLP mainly includes two common types, namely single-row and multiple-row layout problems. Those two problems only need to determine the sequence of facilities and do not consider the exact location of each facility. In 2010, Chung *el al.* [1] first proposed double-row layout problem (DRLP) where facilities are placed on two parallel rows and the aisle width is assumed to be zero. They presented a mixed-integer programming model for the DRLP. Zuo *et al*. [2] proposed an extend DRLP, in which the aisle width is assumed to be non-zero and the two objectives of minimizing layout area and material flow cost are optimized.

In practical manufacturing environment, material flows among facilities may change over time because different product types are usually processed in different periods. This requirement leads to the problem of dynamic facility layout problem (DFLP), in which the processing time is divided into a number of periods and the material flows among facilities are fixed in one period but change over different periods. Each period of DFLP can be regarded as a SFLP. The cost objective of a dynamic layout is the sum of two types of costs, i.e., the material handling cost in all periods and the rearrangement cost of those facilities to be relocated.

In this paper, we extend the DRLP in [2] to a dynamic environment and propose a dynamic double-row layout problem, in which the material flow among facilities may change over time and we need to determine the exact location of each facility in each period to minimize the total cost. A mixed-integer programming model is established for the DDRLP. An improved simulated annealing algorithm is proposed to solve this problem. To represent a solution to the DDRLP, we devise a mixed coding scheme to express the sequence of facilities as well as the exact location of each facility. Five operators of ISA are suggested to make the ISA able to effectively handle this problem.

The structure of this paper is organized as follows. Section II gives a brief review of the related literatures. In Section III, a mathematical programming formulation of DDRLP is established. Section IV introduces the improved simulated annealing algorithm with its five operators. Experimental results are given in Section V. Finally Section VI concludes this paper.

II. RELATED LITERATURES

Rosenblatt [3] first proposed DFLP and a solution approach based on the dynamic programming model. This approach is computationally intractable and only suitable for small size problems. Due to the NP-hard property of DFLP, most of its solution approaches are heuristics. For example, Baykasoglu *et al*. [4] presented a simulated annealing algorithm for DFLP. McKendall *et al*. [5] proposed two improved SA for DFLP: the first one is the basic SA and the second one is a SA with a look-ahead/look-back strategy. McKendall *et al*. [6] presented a hybrid ant system for DFLP. Baykasoglu *et al*. [7] proposed an ant colony optimization algorithm to solve DFLP with budget constraints. Sahin *et al.* [8] proposed a simulated annealing algorithm for DFLP with budget constraints. Rezazadeh *et al*. [9] extended an improved version of the discrete particle swarm optimization algorithm for DFLP. . Pillai *et al.* [10] utilized a SA based meta-heuristic to produce a robust solution to DFLP. Chen [11] presented an approach to streamline the data structure of solution representation to improve the solution swapping and storing activities within a meta-heuristic framework.

All DFLPs mentioned above only need to determine the sequence of facilities and do not consider the exact location of each facility. Although there exists few literatures devoted to DFLP with the consideration of exact locations of facilities, DFLP in those literatures are much different from the DDRLP. For example, McKendall *et al*. [12] proposed a tabu search for DFLP using a continuous representation of layout with unequal-area departments and free orientations. Mazinani *et al*. [13] proposed a flexible bay structure and solved it by a genetic algorithm. To the best of our knowledge, there have been no studies on dynamic DRLP that involves both of combinatorial (sequence of facilities) and continuous (exact location of each facility) aspects.

III. Problem Formulation

We formulate a mathematical model for DDRLP. In this problem, facilities are located on two parallel rows and a facility can be located at any position on a specific row. Location of each facility may vary from one period to another. If the locations of a facility in two adjacent periods are different, rearrangement of this facility is needed. The optimization objective is to find a facility layout for each period in the planning horizon such that the sum of the material handling cost and rearrangement cost are minimized.

The problem parameters and decision variables are defined in Tables I and II, respectively.

Mixed integer programming model:

$$\text{Min} \quad \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} F_{tij}\left(\left|X_{tir} - X_{tjr}\right| + c\left(1 - Q_{tij}\right)\right) + \sum_{t=2}^{T}\sum_{i=1}^{N} A_i B_{ti} \quad (1)$$

TABLE I
PROBLEM PARAMETERS

Problem parameters	Description
N	Number of facilities
i, j	Facility index. $i, j \in \{1, 2,\ldots,N\}$
T	Number of periods
t	Period index. $t \in \{1, 2,\ldots,T\}$
r	Row index, $r \in R = \{1, 2\}$
W_i	Width of facility i
C_{ij}	Minimum clearance between facilities i and j
F_{tij}	Material flow of an unit distance from facility i to j in period t
c	Width of corridor
A_{ti}	Rearrangement cost of facility i in period t

TABLE II
DECISION VARIABLES

Decision variable	Description
X_{tir}	Continuous variable representing absolute position of facility i in period t if located on row r; otherwise, 0.
Y_{tir}	Binary variable. 1: if facility i is located on row r in period t; 0: otherwise.
Z_{tijr}	Binary variable. 1: if facilities i and j are located on row r in period t and facility i is to the left of facility j; 0: otherwise.
Q_{tij}	Binary variable. 1: if facilities i and j are located on the same row in period t ; 0: otherwise
B_{ti}	Binary variable. 1: if facility i is rearranged in period t where $t \geq 2$; 0 : otherwise.

Subject to

$$\frac{Y_{tir}W_i}{2} \leq X_{tir}, \quad t \in \{1,2..T\}, i \in \{1,2..N\}, r \in \{1,2\} \quad (2)$$

$$X_{tir} \leq MY_{tir}, \quad t \in \{1,2...T\}, i \in \{1,2...N\}, r \in \{1,2\} \quad (3)$$

$$\sum_{r \in R} Y_{tir} = 1, \quad t \in \{1,2...T\}, i \in \{1,2...N\}, r \in \{1,2\} \quad (4)$$

$$Z_{tijr} + Z_{tjir} \leq Y_{tir}$$
$$t \in \{1,2..T\} \quad i,j \in \{1,2..N\}, j \neq i \quad r \in \{1,2\} \quad (5)$$

$$Z_{tijr} + Z_{tjir} \leq Y_{tjr}$$
$$t \in \{1,2..T\} \quad i,j \in \{1,2..N\}, j \neq i \quad r \in \{1,2\} \quad (6)$$

$$Z_{tijr} + Z_{tjir} + 1 \geq Y_{tir} + Y_{tjr}$$
$$t \in \{1,2..T\} \quad i,j \in \{1,2..N\}, j \neq i \quad r \in \{1,2\} \quad (7)$$

$$Q_{tij} = \sum_{r \in R}\left(Z_{tijr} + Z_{tjir}\right)$$
$$t \in \{1,2...T\} \quad i,j \in \{1,2...N\} \quad j \neq i \quad (8)$$

$$\frac{Y_{tir}W_i}{2} + \frac{Y_{tjr}W_j}{2} + C_{ij}Z_{tijr} \leq X_{tjr} - X_{tir} + M\left(1 - Z_{tijr}\right)$$
$$t \in \{1,2..T\} \quad i,j \in \{1,2..N\}, j \neq i \quad r \in \{1,2\} \quad (9)$$

$$\frac{Y_{tir}W_i}{2} + \frac{Y_{tjr}W_j}{2} + C_{ij}Z_{tjir} \leq X_{tir} - X_{tjr} + M\left(1 - Z_{tjir}\right)$$
$$t \in \{1,2..T\} \quad i,j \in \{1,2..N\}, j \neq i \quad r \in \{1,2\} \quad (10)$$

$$\left|Y_{tir} - Y_{(t-1)ir}\right| \leq B_{ti} \quad t \in \{2...T\}, i \in \{1,2...N\}, r \in \{1,2\} \quad (11)$$

$$M\sum_{r \in R}\left|X_{tir} - X_{(t-1)ir}\right| \geq B_{ti} \quad t \in \{2...T\}, i \in \{1,2...N\} \quad (12)$$

$$\frac{\sum_{r \in R}\left|X_{tir} - X_{(t-1)ir}\right|}{M} \leq B_{ti} \quad t \in \{2...T\}, i \in \{1,2...N\} \quad (13)$$

$$X_{tir} \geq 0, \quad t \in \{1,2..T\}, i \in \{1,2..N\}, r \in \{1,2\} \quad (14)$$

$$Y_{tir} \in \{0,1\} \quad t \in \{1,2...T\}, i \in \{1,2...N\}, r \in \{1,2\} \quad (15)$$

$$Z_{tijr} \in \{0,1\} \quad t \in \{1,2..T\} \quad i,j \in \{1,2..N\}, j \neq i \quad r \in \{1,2\} \quad (16)$$

$$Q_{tij} \in \{0,1\} \quad t \in \{1,2...T\} \quad i,j \in \{1,2...N\}, j \neq i \quad (17)$$

The first term in objective function (1) is material handling cost and the second one is rearrangement cost. Constraint (2) is used to restrict the minimum distance from the most left position for each facility. Constraint (3) ensures that the absolute position of any facility on a row is zero if the facility is not located on this row. Constraint (4) ensures that one facility can only be located on one row. Constraints (5)-(7) are used to determine the relationship of decision variables Y_{tir} and Z_{tijr}. Constraint (8) is to determine whether any two facilities are on the same row. Constraints (9) and (10) ensure that any two facilities do not overlap and meet the minimum clearance constraint. Constraints (11)-(13) ensure a facility is rearranged if its exact position is different in two adjacent periods. M in constraint (3), (9), (10), (12) and (13) is a sufficiently large number.

IV. PROPOSED SIMULATED ANNEALING ALGORITHM

SA, first proposed by Kirkpatrick *et al*. [14], is a stochastic neighborhood search algorithm for optimization problems. Its basic idea comes from the annealing process of solids, in which a solid is heated and its temperature is slowly decreased until it reaches the lowest energy state.

SA has the capability of jumping out of local optima, which is achieved by accepting solutions worse than the current solution with certain probability. SA has been proved to be effective for facility layout problems [4-5]. Hence, in this paper we devise a SA to solve the DDRLP.

The steps of proposed ISA are as follow.

Step 1: Initialization. Initialize initial temperature T_i, termination temperature T_e, annealing factor β and inner loop constants K. Set current temperature T_c as T_i. Randomly produce an initial solution s as current solution and calculate its cost $f(s)$. Let *Best* be current optimal solution and it is initialized as s. Let inner loop counter k=0.

Step 2: Find a neighbor of current solution s. For current temperature T_c and each inner loop k, a neighbor s' of s is produced by one of the five operators to be introduced in Section IV B.

Step 3: Update current and best solutions. If s' is better than s, then s is replaced with s', otherwise, let $\triangle f = (f(s) - f(s')) / T_c$ and replace s by s' if a randomly produced real number $rand \in [0,1]$ satisfies $rand < e^{\triangle f}$. If s' is better than *Best*, then replace *Best* with s'.

Step 4: Inner loop. If $k < K$, then let $k=k + 1$ and return to Step 2; otherwise, go to Step 5.

Step 5: Outer loop. If $T_c < T_e$, then let $T_c = T_c \times \beta$ and return to Step 2; otherwise, the algorithm stops and output *Best* as the found solution.

A. Representation of solution

A coding scheme needs to be devised to represent a solution able to express the sequence of facilities as well as the exact position of each facility. To do this, we devise a mixed coding scheme that involves a sequence matrix and an addition matrix, each of which is a three-dimensional matrix denoted by $\{t, i, r\}$, where $t \in \{1,\ldots,T\}$, $i \in \{1,\ldots,N\}$ and $r \in \{1,2\}$ represents the period, facility and row, respectively. Sequence matrix represents the sequence of facilities on both rows and addition matrix donates the additional distance to the left of a facility (not including the minimum clearance between the facility and the one to the left of it). For example, if facilities i and j on row r are adjacent in period t and facility j is to the left of facility i, then their distance is $(W_i + W_j) / 2 + C_{ij}$ + add $[t][r][i]$, where add $[t][r][i]$ is an element of addition matrix and denotes the additional distance of i . If there exists no facility to the left of facility i, then add $[t][r][i]$ represents the distance between the most left position to the left side of facility i.

For instance, Figure 1 shows a layout (solution) with one period and five facilities. Facility 3 is located at the most left position of the upper row. The sequence matrix of this layout is [[[3,1,5,0,0],[4,2,0,0,0]]] and addition matrix is [[[0 , $b - C_{31}$, $c - C_{15}$, 0 , 0],[d , $e - C_{42}$, 0 , 0 , 0]]].

B. Searching neighboring solution

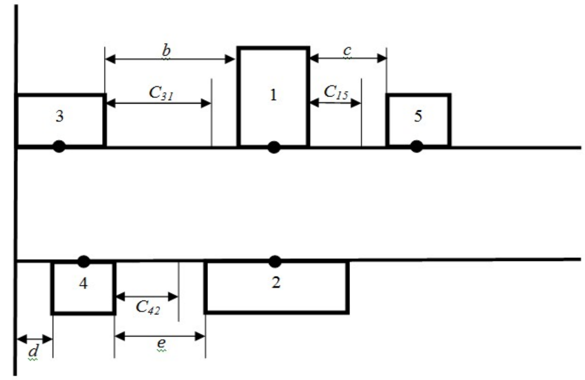


Fig. 1 A layout with five facilities and one period, where b is the gap between facilities 3 and 1 such that the additional distance of facility 1 is $b - C_{31}$.

In step 2 of ISA, a neighbor of the current solution is produced by one operator randomly chosen from the following five operators.

1) Swap operator. Randomly select a period t from T periods and two facilities in this period, and then swap the locations of the two facilities. For the solution shown in Figure 1, if facilities 1 and 2 are chosen to swap their locations, a new solution can be produced whose sequence and addition matrixes are [[[3,2,5,0,0],[4,1,0,0,0]]] and [[[0 , $e - C_{42}$, $c - C_{15}$, 0 , 0],[d , $b - C_{31}$, 0 , 0, 0]]], respectively.

2) Insert operator. Randomly select a period t from T periods and one facility in this period, and then the facility is inserted into any position on another row. For instance, in the solution in Figure 1, if facility 1 is selected and inserted into the position between facilities 4 and 2 on the lower row, then a new solution is produced whose two matrixes are [[[3,5,0,0,0],[4,1,2,0,0]]] and [[[0 , $c - C_{15}$, 0, 0 , 0], [d , $b - C_{31}$, $e - C_{42}$, 0 , 0]]], respectively.

3) Tuning operator. Randomly select a period t from T periods and one facility in this period, and then tune the exact position of this facility. For example, in the solution shown in Figure 1, assume facility 1 is selected. Generate a random real number *rand* in [-1, 1], and then a new solution is created by adding *rand* to the additional distance of facility 1. In this case, the addition matrix of the new solution is [[[0 , $b - C_{31} + rand$, $c - C_{15}$, 0 , 0],[d , $e - C_{42}$, 0 , 0 , 0]]] and its sequence matrix maintains unchanged. If $b - C_{31} + rand < 0$, reproduce a random real number *rand* to satisfy $b - C_{31} + rand > 0$.

4) Reverse operator. Randomly select a period t from T periods and reverse the sequence of all facilities while maintain the distance between any two facilities unchanged. For the solution in Figure 1, the sequence and addition matrixes of the new produced solution by the reverse operator are [[[5,1,3,0,0],[2,4,0,0,0]]] and [[[0 , c - C_{15}, b - C_{31}, 0 , 0],[$X_{151} + W_5 / 2 - (X_{122} + W_2 / 2)$, e - C_{42}, 0 , 0 , 0]]], respectively, where X_{tir} is the exact position of facility i on row r in period t in the solution in Figure 1.

Reverse operator is devised to handle the symmetry case in Figure 2. The relative positions of three facilities in period t and t +1 are identical, but all of them need to be rearranged in

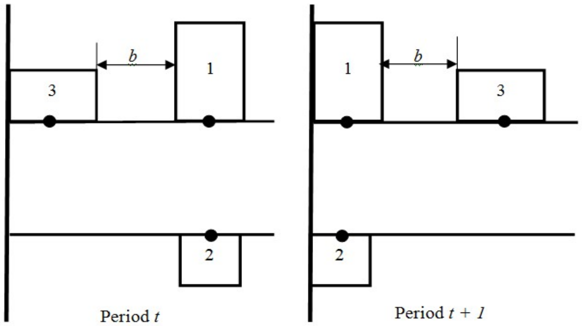
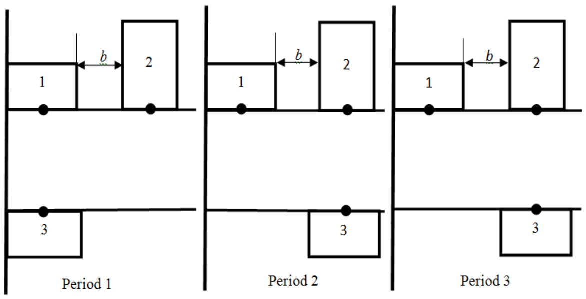
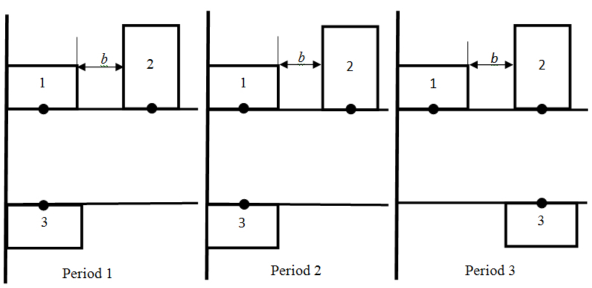


Fig. 2. An example that needs the reverse operator

period $t + 1$ because the absolute position of each facility is different in the two periods. When apply reverse operator to the facilities in period t or $t + 1$, those facilities do not need to



a)



b)

Fig. 3. a): A solution of the problem with three periods and three facilities. Each facility in the three periods has the same position except facility 3 in period 1. b).The optimal solution of the problem.

be rearranged, such that the rearrangement cost is saved.

*5) Consistent operator. R*andomly select a period t from 2 to T and a facility in this period, and then let the location of the facility in period (t+1) be the same as that in period t.

Figure 3 shows the case that needs this operator. In Figure 3 a), the exact positions of three facilities are the same in periods 2 and 3 but the position of facility 3 in period 1 is different from that in the following two periods. The solution shown in Figure 3 b) is the optimal one. Apparently, it is not easy to find the optimal solution from Figure 3 a) via previous four operators. For example, if we want to apply the tuning operator to facility 3 in period 2, the material flow cost may be decreased and at the same time facility 3 needs to be rearranged from period 2 to 3. If the rearrangement cost is greater than the reduction of materials flow cost, it is difficult to find the optimal solution by the tuning operator. When apply the consistent operator to facility 3 in period 2, we can easily find the optimal layout in Figure 3 b). We observe that the optimal solutions of some problem instances (e.g., P2 and P3 in Section V) cannot be found without the consistent operator.

V. EXPERIMENTAL RESULTS

The DDRLP includes parameters N, T, c, F_{tij}, W_i, C_{ij} and A_{ti}. 19 problem instances are created randomly, namely P1~P19. The number of periods and facilities in each instance are listed in Table III. The width of corridor is given by 1.5 or 2. In each period t, the number of product types is p～unif[8,10]; the percentage of facilities visited by each product type is r～unif[0.25,0.75] and n～unif[20,50] is the number of products for each type. Parameter F_{tij} is calculated as the sum of products whose routes include facility i immediately preceding facility j in period t. Other parameters are: w_i～unif[0.5,2.5], C_{ij}～unif[0.25,2.25] and A_{ti}～unif[30, 80].

The ISA is coded in Microsoft Visual Studio C++ 2010 and executed in Windows operation system on a desktop PC with Pentium Dual-core E2140 1.6GHz CPU and 1GB RAM.

Parameters of ISA are obtained by brief experiments and given in Table III. For each problem instance, five independent runs of ISA are done to obtain the average cost and standard deviation of the found solutions, as shown in Table IV. For P1 and P2, ISA only uses operators 1-4 in Section IV B because the fifth operator is used for problem instances involving more than one period.

For P1 and P2, five runs of ISA obtain the same results and each run takes about 2 minutes. For P3 and P4, each run of ISA is able to find the same or similar results within one minute. The results found by ISA for other large size problem instance are also given in Table IV. We can see that the ratio of standard deviation to average cost is less than one percent, which means the performance of ISA is stable for this problem.

Since there exists no solution approaches for this new defined problem in this paper, we compare the ISA with an exact approach -- a popular mathematical programming solver -- CPLEXL 12.4. CPLEX is used to solve the model in Section III and able to find the optimal solution for small size problems. Table V gives the comparison of the results obtained by ISA and CPLEX. For P1 and P2, CPLEX can't produce an optimal solution within ten hours, such that we restrict the runtime of CPLEX as one hour. For the two instances, ISA can find much better solutions than CPLEX. For P3 and P4, CPLEX is able to find their optimal solutions and ISA is able to find the optimal solution for P3 and the suboptimal solution (very close to the optimal one) for P4.

CPLEX can't find the optimal solutions for P5-P9 under a reasonable computational time, such that the runtime of CPLEX is restricted as one hour for those instances. We can see from Table V that the solutions of ISA are far better than those of CPLEX. For P10 and P11, CPLEX can't find a feasible solution for each of them within one hour, such that the run time of CPLEX is increased to 6 and 10 hours for P10 and P11, respectively. We find that the solutions obtained by CPLEX are much worse than those found by ISA for the two instances. Due to the NP-hard property of DDRLP, CPLEX cannot produce feasible solutions for P12-P19 even if its runtime is increased to ten hours.

TABLE III
PROBLEM AND ALGORITHM PARAMETERS

Problem instances	Problem parameters		Algorithm paramters			
	Number of periods (T)	Number of facilities (N)	Initial temperature (T_i)	Terminal temperature (T_e)	Annealing factors (β)	Iterations(K)
P1-P2	1	10	100	0.001	0.99	19000
P3-P4	3	3	100	0.01	0.99	10000
P5-P6	5	10	100	1	0.99	10000
P7	6	10	100	1	0.99	10000
P8	7	10	100	1	0.99	9500
P9	8	10	100	1	0.99	9500
P10	5	20	100	1	0.99	9000
P11	6	20	100	1	0.99	9000
P12	7	20	100	1	0.99	8900
P13	8	20	100	1	0.99	8600
P14	9	20	100	1	0.99	8000
P15	5	30	100	1	0.99	7200
P16-P17	6	30	100	1	0.99	7200
P18	7	30	100	1	0.99	6500
P19	8	30	100	1	0.99	6500

TABLE IV
EXPERIMENTAL RESULTS OF ISA

Problem instances	1st	2rd	3rd	4th	5th	Average values	Standard deviations
P1	4947.41	4947.41	4947.41	4947.41	4947.41	4947.41	0
P2	2991.79	2991.80	2991.79	2991.79	2991.79	2991.79	0
P3	1755.47	1755.36	1755.51	1755.56	1755.38	1755.45	0.1
P4	1683.69	1683.69	1683.69	1683.69	1683.69	1683.69	0
P5	25008.0	24978.8	25006.4	24985.2	24980.1	24991.7	14
P6	26084.3	25746.0	26075.3	25826.8	25707.2	25887.9	180
P7	28334.1	28246.1	28109.0	28069.0	28169.7	28185.6	106
P8	32571.2	31652.7	32370.5	33575.8	32196.7	32475.4	704
P9	44967.5	44729.1	45619.3	45043.4	45930.3	45257.9	498
P10	75180	76069	75050	75276	74205	75155	663
P11	89562	88781	87876	88401	86921	88309	989
P12	100603	99485	101060	101551	99916	100523	836
P13	139856	137373	139263	136610	142465	139113	2297
P14	135589	135662	136505	138768	138273	136959	1480
P15	177951	179245	174831	178807	180898	178347	2238
P16	229776	230203	229433	225286	227980	228535	1999
P17	222799	215148	220157	219013	217288	218881	2892
P18	250528	253656	251003	249934	251377	251300	1423
P19	293166	295804	296451	296250	290482	294431	2572

TABLE V
COMPARISION OF ISA AND CPLEX

Problem instances	ISA		CPLEX	
	Average cost	Time (min)	Cost	Time(min)

P1	**4947.41**	2.60	5326.16	60
P2	**2991.79**	2.70	3649.85	60
P3	1755.45	<1.00	**1755.32(optimal)**	8
P4	**1683.69**	<1.00	**1683.69(optimal)**	12.12
P5	**24991.7**	5.17	38867.9	60
P6	**25887.9**	5.50	42886.2	60
P7	**28185.6**	6.17	49801.8	60
P8	**32475.4**	6.83	71667.4	60
P9	**45257.9**	7.00	1074822	60
P10	**75155**	18.75	300861	360
P11	**88309**	22.83	299798	600
P12	**100523**	26.00	---	600
P13	**139113**	28.67	---	600
P14	**136959**	30.00	---	600
P15	**178347**	35.08	---	600
P16	**228535**	42.17	---	600
P17	**218881**	44.33	---	600
P18	**251300**	45.50	---	600
P19	**294431**	49.83	---	600

I. CONCLUSIONS

In this paper, a dynamic double-row layout problem (DDRLP) is proposed, which involves multiple periods and in each period facilities are located at any position on two parallel rows. A mixed-integer programming mode is established for this DDRLP and an improved simulated annealing (ISA) algorithm is proposed to resolve it. To express the sequence of facilities and their exact locations, a mixed encoding scheme is devised to represent a feasible solution. Five effective operators of ISA are suggested to produce a neighbor from the current solution. The ISA is compared with an exact approach (CPLEX). Experimental results show that for small-scale problem instances ISA is able to find the optimal solutions in most cases and for large-scale instances ISA outperforms CPLEX under a limited run time.

ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China (61374204; 61375066).

REFERENCES

[1] J. Chung and J. M. A. Tanchoco, “The double row layout problem,” *International journal of Production Research*, vol. 3, pp. 709-727, 2010.

[2] X. Zuo, C. C. Murray, and A. E. Smith. “Solving an extended double row layout problem using multi-objective tabu search and linear programming,” *IEEE Transactions on Automation Science and Engineering*, (DOI: 10.1109/TASE.2014.2304471).

[3] M. J. Rosenblatt, “The dynamics of plant layout,” *Management Science*, vol. 32, pp. 76-86, 1986.

[4] A. Baykasoglu and N. N.Z. Gindy, “A simulated annealing algorithm for dynamic layout problem,” *Computers & Operations Research*, vol. 28, pp. 1403-1426, 2001.

[5] A. R. Mckendall, J Shang and S. Kuppusamy, “Simulated annealing heuristics for the dynamic facility layout problem,” *Computers & Operations Research*, vol. 33, pp. 2431-2444, 2006.

[6] A. R. Mckendall and J Shang, “Hybrid ant system for the dynamic facility layout problem,” *Computers & Operations Research,* vol. 33, pp. 790-803, 2006.

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

[8] R. Sahin, K.Ertogral and O. Tuikbey, “A simulated annealing heuristic for the dynamic layout problem with budget constraint,” *Computers & Industrial Engineering*, vol. 59, pp. 308-313, 2010.

[9] H. Rezazadeh, M Ghazanfari, M. Saidimehrabad and S. J. Sadjadi, “An extended discrete particle swarm optimization algorithm for the dynamic facility layout problem,” *Journal of Zhejiang University Science A,* vol. 10, pp. 520-529, 2009.

[10] V. M. Pillai, I. B. Hunagund and K. K. Krishnan, “Design of robust layout for dynamic plant layout problems,” *Computers & Industrial Engineering,* vol. 61, pp. 813-823, 2011.

[11] G. Y. Chen, “A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems,” *International Journal of Production Economic*, vol. 142, pp. 362-371, 2013.

[12] A. R. Mckendall and A. Hakobyan, “Heuristics for the dynamic facility layout problem with unequal-area departments,” *European Journal of Operational Research*, vol. 201, pp. 171-182, 2010.

[13] M. Mazinani, M.Abedzadeh and N. Mohebali, “Dynamic facility layout problem based on flexible bay structure and solving by genetic algorithm,” *International Journal of Advanced Manufacturing Technology,* vol. 65, pp. 929-943, 2013.

[14] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi,”Optimization by Simulated Annealing,” *Science*, vol. 220, pp. 671-680, 1983.