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

A Tabu Search Heuristic for the Single Row Layout Problem with Shared Clearances

Meng Yu, Xingquan Zuo, Chase C. Murray

***Abstract*—The single row layout problem is a common and well-studied practical facility layout problem. The problem seeks the arrangement of a fixed number of facilities along one row that minimizes the objective of total material handling cost. In this paper, a single row layout problem with shared clearance between facilities is proposed. The shared additional clearance may be considered on one or both sides of each facility. To solve this problem tabu search is combined with a heuristic rule to solve problems of realistic size. Tabu search is used to find the sequence of facilities while the heuristic rule is determines the additional clearance for each facility. The proposed solution approach is applied to several problem instances involving 10, 20 and 30 facilities, and is compared against a popular mathematical programming solver (CPLEX). Computational results show that our approach is able to obtain high quality solutions and outperforms CPLEX under limited computational time for problems of realistic sizes.**

I. INTRODUCTION

Facility layout problems (FLP), whereby a pre-determined number of facilities (e.g., individual machines or cells formed by groups of machines) must be placed within a space in an arrangement that minimizes the total cost of operating the system, are the focus of myriad research works due to their wide-ranging applications. Typically, facility layout problems are classified by the arrangement of the facilities, such as single-row, double-row, or multiple-row layouts. This paper explores the single row layout problem (SRLP), which is also called the single row facility layout problem (SRFLP).

The classical SRLP, first proposed by Simmons [1], seeks to minimize the total cost associated with placing a collection of rectangular facilities along a straight line, where the cost is defined to be the sum of the flow between each pair of facilities times the distance between the center points of those facilities. Often this cost is described as denoting the "material handling cost." In most SRLP formulations, a parameter representing the minimum required clearance between each pair of machines is incorporated. Thus, the problem is to find the minimum-cost permutation of the facilities.

The SRLP has a wide array of practical applications, as noted by [14]. For example, arranging machines on the facility floor in modern manufacturing facilities [2], locating rooms along one corridor in hospitals or departments in office buildings or supermarkets [1], and allocating information items among the "cylinders" of a magnetic disk [3].

Although the SRLP is NP-hard [4-5], a number of exact approaches have been proposed for small-scale problems. For example, Simmons [1] proposed a branch-and-bound algorithm to obtain an exact solution of the SRLP. Amaral [6] presented a mixed-integer linear programming model of this problem. In [7], a semi-definite optimization approach was studied to provide a lower bound of the optimal value of a one-dimensional space-allocation problem. Cutting planes [8], dynamic programming [3] and branch-and-cut approaches [9] were also proposed to obtain the exact solution of small-scale SRLPs.

While these exact approaches are capable of determining optimal solutions for small-scale problems, the NP-hard nature of this problem motivates the need for heuristic approaches suitable for solving larger-scale problems in a reasonable amount of time. Numerous heuristics for the SRLP have been proposed, including genetic algorithms [10], simulated annealing [11], particle swarm optimization [12], ant colony optimization [13], tabu search [14], and greedy heuristics [15].

In practice, the assumptions of the classical SRLP are inappropriate for some settings. While the basic SRLP has received considerable attention, in practice there are additional complicating elements (extensions) that should be addressed (these are ignored by the traditional SRLP). Thus, recent research has considered extensions to the classical SRLP. For example, two characteristics of a linear single row flow path (i.e., path configuration and feasible flow path direction) were investigated in [16]. In [17], an enhanced SRLP is proposed with the consideration of the width of facilities, traffic loads between facilities, and installation cost.

In this paper, we consider a new practical extension to the SRLP. Specifically, this extension considers two classes of clearance between adjacent facilities. In the traditional SRLP, each pair of adjacent facilities is separated by the *minimum clearance* between them. This clearance is used to prevent facilities from touching each other and to allow ventilation. In practice, each facility may need another type of clearance, termed *additional clearance*, which is used to allow a technician access to the side of the facility or store work in process (WIP). This additional clearance can be shared for two adjacent facilities. Since the material handling cost is related to the distance between facilities, additional clearance sharing may decrease the total material handling cost.

To solve this new problem, which we term the *SRLP with shared clearances* (SRLP-SC), a tabu search approach is used to optimize the sequence of facilities while a new heuristic rule is devised to determine the additional shared clearance for each facility. Experimental results show that the proposed solution approach provides near-optimal solutions in a limited time.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

II. Problem definition

The SRLP-SC can be formally defined as follows. Let I= {1, 2… m} represent the set of m facilities, such that each facility $i \in I$ has a fixed width w_i. The minimum required clearance between facilities i and j is denoted as c_{ij}, defined as the minimum allowable distance between the sides of facilities i and j that are closest to each other. Let f_{ij} represent the unit cost times flow frequency from facility i to facility j. In the asymmetric case, f_{ij} does not necessarily equal f_{ij}. The additional required clearance is categorized such that a_i^l (a_i^r) represents the additional clearance needed on the left (right) side of facility i. A binary parameter b_i^{lr} is defined, such that both a_i^l and a_i^r are necessary if $b_i^{lr}=1$; otherwise, either a_i^l or a_i^r is needed, but not necessarily both. Suppose that facilities i and j are adjacent and facility i is to the left of facility j. If facility i (j) has an additional clearance to its right (left) side, their additional clearances can be shared.

The SRLP-SC seeks to determine the sequence of facilities and the additional clearance for each facility, with the objective of minimizing the sum of the weighted distance between each facility pair.

An example is given to clearly introduce this problem. Suppose that there are m=3 facilities, that the width of each facility is equal to 3 units ($w_i = 3, \forall i \in \{1,2,3\}$), and that the minimum required clearance between each pair of facilities is one unit ($c_{ij}=1, \forall i, j \in \{1,2,3\}, i \neq j$). Furthermore, suppose that an additional (sharable) clearance associated with each facility is defined such that $a_i^l = a_i^r = 1$, $\forall i \in \{1,2,3\}$, $b_1^{lr} = b_3^{lr} = 0$, and $b_2^{lr} = 1$. Thus, an additional clearance of one unit is required on either the left or right side of facilities 1 and 3 ($b_1^{lr} = b_3^{lr} = 0$), and the additional clearance of one unit is required on both sides of facility 2 ($b_2^{lr}=1$). A feasible solution to this problem is shown in Figure 1. Facilities 1 and 2 can share their additional clearance, such that the total distance between their center points is $(w_1 + w_2)/2 + c_{12} + \max\{a_1^r, a_2^l\}$. Similarly, the minimum clearance between facilities 2 and 3 is $(w_2 + w_3)/2 + c_{23} + \max\{a_2^r, a_3^l\}$.

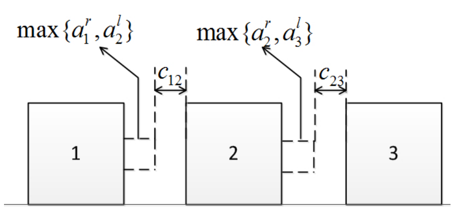

Fig. 1. A solution of SRLP-SC

A. Problem Formulation

Definitions of decision variables are given in Table I. A mixed-integer linear programming formulation is established for the SRLP-SC to minimize the total material handling cost.

TABLE I
DEFINITION OF DECISION VARIABLES

x_{ih}	Binary decision variable, such that $x_{ih}=1$ if facility $i \in I$ is placed in position $h \in I$ (each facility must occupy one position); otherwise $x_{ih}=0$.
q_{ij}	Binary decision variable, such that $q_{ij}=1$ if facility $i \in I$ is immediately to the left of facility $j \in I \setminus i$; otherwise $q_{ij}=0$.
$z^a_{h,h+1}$	Continuous decision variable representing the additional clearance between facilities in positions $h \in I$ and $(h+1) \in I$.
$z^m_{h,h+1}$	Required minimum clearance between facilities in positions $h \in I$ and $(h+1) \in I$.
l_h	Continuous decision variable denoting the location of the facility in position $h \in I$, measured by the distance from the middle point of the facility in the first position to that in the hth position.
u_i	Continuous decision variable representing the location of facility $i \in I$.
d_{ij}	Continuous decision variable representing the distance between facility $i \in I$ and $j \in I \setminus i$.
s_{ij}	Binary decision variable, such that $s_{ij}=1$ if facility $i \in I$ is placed immediately to the left of facility $j \in I \setminus i$ and they can share their additional clearance; otherwise $s_{ij}=0$.
$p_i^l (p_i^r)$	Binary decision variable, such that $p_i^l = 1$ ($p_i^r=1$) if a_i^l (a_i^r) is applied to the left (right) side of facility $i \in I$.

$$\text{Minimize} \quad \sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m} f_{ij} d_{ij} \tag{1}$$

Subject to

$$\sum_{i=1}^{m} x_{ih} = 1, \forall h \in I \tag{2}$$

$$\sum_{h=1}^{m} x_{ih} = 1, \forall i \in I \tag{3}$$

$$\sum_{\substack{i=1\\ i\neq j}}^{m} q_{ij} = \sum_{h=2}^{m} x_{jh}, \forall j \in I \tag{4}$$

$$\sum_{\substack{j=1\\ j\neq i}}^{m} q_{ij} = \sum_{h=1}^{m-1} x_{ih}, \forall i \in I \quad j \neq i, h \in \{1,2,...,m-1\} \tag{5}$$

$$q_{ij} \geq x_{ih} + x_{jh+1} - 1, \quad \forall i \in I, j \in I \tag{6}$$

$$p_i^r + p_i^l \geq b_i^{lr} + 1, \forall i \in I \tag{7}$$

$$s_{ij} \le q_{ij}, \forall i \in I, j \in I, i \ne j \quad (8)$$

$$\sum_{\substack{i=1 \\ i \ne j}}^{m} s_{ij} \le p_j^l, \forall j \in I \quad (9)$$

$$\sum_{\substack{j=1 \\ j \ne i}}^{m} s_{ij} \le p_i^r, \forall i \in I \quad (10)$$

$$s_{ij} \ge p_i^r + p_j^l + q_{ij} - 2, \ \forall i \in I, j \in I, i \ne j \quad (11)$$

$$l_1 = 0 \quad (12)$$

$$l_h = l_{h-1} + z_{h-1,h}^{\mathrm{m}} + z_{h-1,h}^{a} + \sum_{i=1}^{m} \frac{1}{2} w_i x_{ih-1} + \sum_{i=1}^{m} \frac{1}{2} w_i x_{ih}$$
$$\forall h \in \{2,...,m\} \quad (13)$$

$$z_{h,h+1}^{\mathrm{m}} \ge c_{ij} q_{ij} - M(2 - x_{ih} - x_{jh+1})$$
$$\forall i \in I, j \in I, i \ne j, h \in \{1,...,m-1\} \quad (14)$$

$$z_{h,h+1}^{\mathrm{m}} \le c_{ij} q_{ij} + M(2 - x_{ih} - x_{jh+1})$$
$$\forall i \in I, j \in I, i \ne j, h \in \{1,...,m-1\} \quad (15)$$

$$z_{h,h+1}^{a} \ge a_i^r p_i^r - a_i^r (1 - x_{ih}), \forall i \in I, h \in \{1,...,m-1\} \quad (16)$$

$$z_{h,h+1}^{a} \ge a_j^l p_j^l - a_j^l (1 - x_{jh+1}), \ \forall j \in I, h \in \{1,...,m-1\} \quad (17)$$

$$z_{h,h+1}^{a} \le a_i^r p_i^r + a_j^l p_j^l - \min(a_i^r, a_j^l) s_{ij} + M(2 - x_{ih} - x_{jh+1})$$
$$\forall i \in I, j \in I, i \ne j, h \in \{1,...,m-1\} \quad (18)$$

$$u_i \ge l_h - M(1 - x_{ih}), \forall i \in I, h \in I \quad (19)$$

$$u_i \le l_h + M(1 - x_{ih}), \forall i \in I, h \in I \quad (20)$$

$$d_{ij} \ge u_i - u_j, \forall i \in I, j \in I, i \ne j \quad (21)$$

$$d_{ij} \ge u_j - u_i, \forall i \in I, j \in I, i \ne j \quad (22)$$

$$z_{h,h+1}^{\mathrm{m}} \ge 0, \forall h \in \{1,...,m-1\} \quad (23)$$

$$z_{h,h+1}^{a} \ge 0, \forall h \in \{1,...,m-1\} \quad (24)$$

$$l_h \ge 0, \forall h \in I \quad (25)$$

$$u_i \ge 0, \forall i \in I \quad (26)$$

$$d_{ij} \ge 0, \forall i \in I, j \in I, i \ne j \quad (27)$$

$$x_{ih} \in \{0,1\}, \forall i \in I, h \in I \quad (28)$$

$$q_{ij} \in \{0,1\}, \forall i \in I, j \in I, i \ne j \quad (29)$$

$$p_i^l \in \{0,1\}, \forall i \in I \quad (30)$$

$$p_i^r \in \{0,1\}, \forall i \in I \quad (31)$$

$$s_{ij} \in \{0,1\}, \forall i \in I, j \in I, i \ne j \quad (32)$$

Objective function (1) serves to minimize the total material handling cost. Constraints (2) and (3) ensure that each facility is placed in one position. Constraints (4)–(6) are related to decision variables q_{ij}. Constraints (4) and (5) ensure that if a facility *i* is not at the first (last) position, there must exist another facility to the left (right) of facility *i*. Constraint (6) guarantees that q_{ij} equals 1 if facility *i* is to the left of *j*. Constraint (7) forces both p_i^l and p_i^r equal to 1 if $b_i^{lr} = 1$; otherwise, either p_i^l or p_i^r equals one (but not necessarily both). Constraint (8) ensures that facilities *i* and *j* cannot share their additional clearance unless they are immediately adjacent. Constraints (9) and (10) ensure that facilities *i* and *j* cannot share their additional clearance (facility *i* is to the left of *j*) unless there exists additional clearance to the right side of facility *i* and to the left side of facility *j*. Constraint (11) guarantees that if facility *i* is to the left of *j* and there are additional clearances to the right side of facility *i* and to the left side of *j*, then $s_{ij} = 1$. Constraint (12) sets the first position as zero. Constraint (13) is used to calculate the exact position of the facility located in the *h*th position. Constraints (14) and (15) determine the minimum clearance of any two adjacent facilities. Constraints (16)–(18) calculate the additional clearances between any two adjacent facilities. Facilities *i* and *j* must share their additional clearances if $s_{ij} = 1$. Constraints (19) and (20) determine the position of each facility. Constraints (21) and (22) give the lower bound of the distance between any two facilities. Constraints (23)–(32) define the decision variables. *M* is a large enough number and may be defined as

$$M = \sum_{i=1}^{m} (w_i + a_i^l + a_i^r + \sum_{\substack{j=1 \\ j \ne i}}^{m} c_{ij}) \quad (33)$$

III. Proposed Solution Approach

The popular tabu search (TS) meta-heuristic proposed by Glover [18] starts from an initial solution and moves to the best solution in the neighborhood of the current solution. Since the best solution may not be better than the current solution, a tabu mechanism (*tabu list*) is used to prevent cyclic search.

TS has proven to be an effective approach for combinatorial problems, including facility layout problems. In [14], the performance of TS, simulated annealing (SA), and genetic algorithms (GA) on various types of FLP was compared and the results show that TS outperforms other approaches in most cases. Hence, we choose TS to resolve the SRLP-SC.

The proposed TS combined with a heuristic rule (TS-HR) is illustrated in Figure 2. A solution in TS is expressed as a permutation of all facilities. First, an initial solution is produced randomly and *current_sol* is initialized as this initial solution. Then a candidate with the lowest cost in the neighborhood of *current_sol* is used to update *current_sol*. Each candidate solution represents a sequence of facilities and does not consider additional clearances. To enable a solution of TS to represent a feasible layout, the heuristic rule is devised to determine the additional clearance for each facility, such that a solution of TS can be evaluated by the layout it represents. If *current_sol* has a lower cost than the best solution found so far, denoted by *best_sol*, then *best_sol* is replaced with *current_sol*. TS stops when the number of iterations (*iter*) reaches a given number *iter_stop* (i.e., if *best_sol* is not updated for *iter_stop* iterations).

It is well known that the performance of TS is sensitive to its initial solution. To enable TS have stable performance, a restart technique is introduced into TS. Thus, TS will restart with a new random initial solution when *iter* reaches a given number of iterations, *iter_res* (that is, *best_sol* remains unchanged for *iter_res* iterations). In the event of a restart, the tabu list is emptied.

A. Solution Expression

For a SRLP-SC instance with m facilities, a solution in TS is expressed by a permutation of these m facilities. For instance, $S = \{4,3,1,2,5\}$ represents a solution for a SRLP-SC with 5 facilities. Let $S[i]$, $i \in I$, be the facility in position i. For example, $S[4]$ denotes the facility in position 4, i.e., the fourth facility (facility 2).

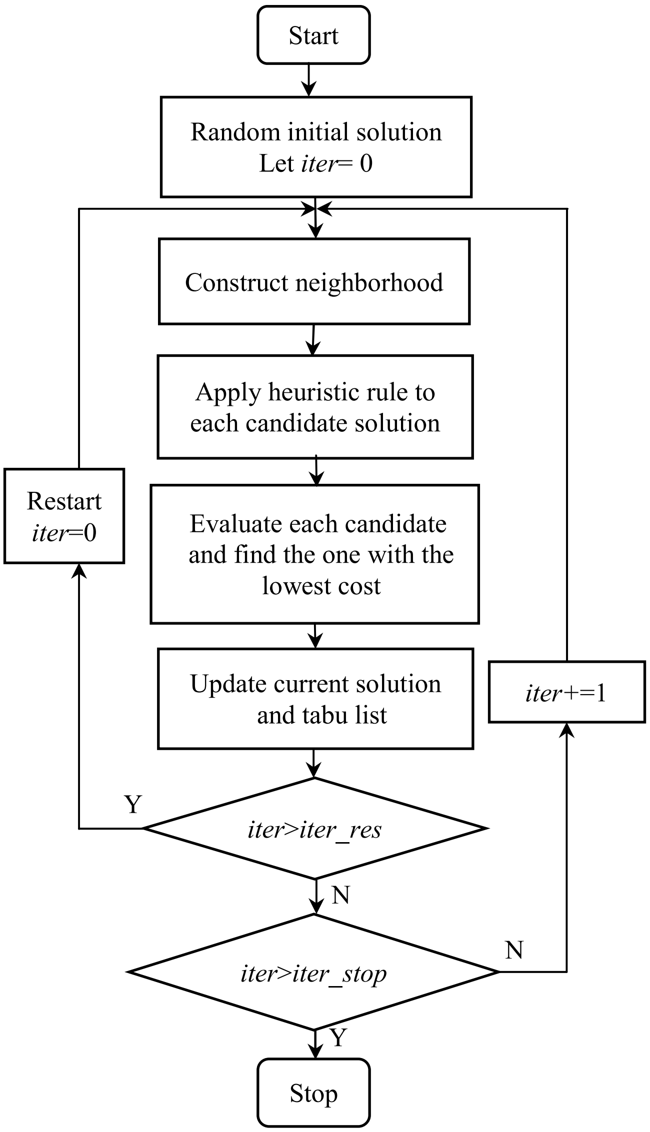

Fig. 2. Tabu search combined with heuristic rule

B. Neighborhood Structure

The neighborhood structure employed by TS is problem-specific. Thus, for the SRLP-SC, the neighborhood of a particular solution is constructed via facility swapping. A *move* (i, j) is defined as the position swapping of facilities i and j. A move is applied to the current solution to produce a new candidate solution (i.e., a *neighbor*), such that all neighbors of the current solution comprise the neighborhood. There are $m(m\text{-}1)/2$ neighbors in the neighborhood of a solution to the SRLP-SC, where m is the number of facilities.

In each iteration, each neighbor (candidate solution) is evaluated by its objective function value. The best one among those candidate solutions in the neighborhood is chosen to replace *current_sol*. If the best candidate solution is produced by applying move (i, j) to the current solution, then the move is added into the tabu list. In other words, move (i, j) is forbidden in the subsequent *tabu_size* iterations unless the move is able to produce a better solution than *best_sol*, where *tabu_size* is the length of the tabu list.

C. Heuristic rule

For SRLP-SC, not only must the minimum clearance between each facility pair be considered, but the side of the additional clearance for each facility i must also be determined (i.e., p_i^l and p_i^r).

In this paper, a heuristic rule is devised to determine the additional clearance for each facility. The main idea is that the procedure encourages facilities to share their additional clearance to minimize the distance between them, thus minimizing their associated material handling costs. For a given solution S of TS (i.e., a sequence of facilities), decision variables p_i^l and p_i^r associated with each facility i are determined by the following heuristic rule:

(1) Set $p_i^l = p_i^r = 1$ for each facility i with $b_i^{lr} = 1$;

(2) For the first (last) facility in the sequence, if $b_{S[1]}^{lr} = 0$ ($b_{S[m]}^{lr} = 0$), set $p_{S[1]}^l = 1$ and $p_{S[1]}^r = 0$ ($p_{S[m]}^l = 0$ and $p_{S[m]}^r = 1$). This serves to reduce the distance between this facility and other facilities, since the additional clearance to its left side (for the last one) or to its right side (for the first one) is zero.

(3) Except for the first and last facilities, and those facilities with $b_j^{lr} = 1$, p_j^l and p_j^r of each facility j is determined from left to right. Suppose that the facility to the left side of facility j is facility i and p_i^l and p_i^r have been determined. Now, the additional clearance of facility j (i.e., p_j^l and p_j^r) must be determined. As illustrated in Figures 3-6, there are four cases to consider. In each case, the heuristic rule is used to determine p_j^l and p_j^r. Note that facility k is to the right side of facility i, thus p_k^l and p_k^r have not yet been determined (as the procedure works from left to right).

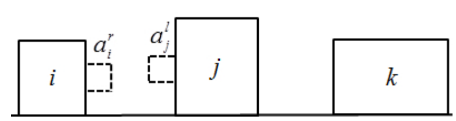

Fig. 3. If $p_i^r = 1$ and $b_k^{lr} = 0$, then let $p_j^l = 1$ and $p_j^r = 0$ to make facilities i and j share their additional clearances. In this case, the additional clearance between facilities i and j is max $\{a_i^r, a_j^l\}$.

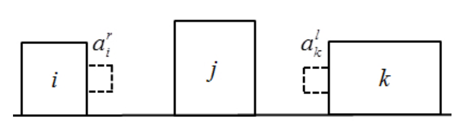

Fig. 4. If $p_i^r = 1$ and $b_k^{lr} = 1$ (i.e., $p_k^l = 1$ and $p_k^r = 1$), then let $p_j^l = 1$ and $p_j^r = 0$ ($p_j^l = 0$ and $p_j^r = 1$) if $\max\{a_i^r, a_j^l\} + a_k^l \le \max\{a_j^r, a_k^l\} + a_i^r$ ($\max\{a_i^r, a_j^l\} + a_k^l > \max\{a_j^r, a_k^l\} + a_i^r$).

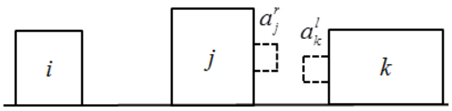

Fig. 5. If $p_i^r = 0$ and $b_k^{lr} = 1$ (i.e., $p_k^l = 1$ and $p_k^r = 1$), then let $p_j^r = 1$ and $p_j^l = 0$ to make facilities *j* and *k* to share their additional clearance.

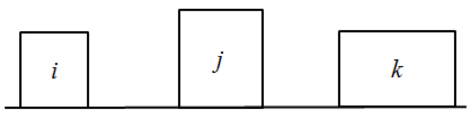

Fig. 6.If $p_i^r = 0$ and $b_k^{lr} = 0$, then let $p_j^l = 1$ and $p_j^r = 0$ ($p_j^l = 0$ and $p_j^r = 1$) if $a_j^l \le a_j^r$ ($a_j^l > a_j^r$).

D. *Update Current Solution*

In each iteration of TS, candidate solutions in the neighborhood need to be evaluated. For a candidate solution, the additional clearance of each facility is determined through the heuristic rule in Section III C, such that a layout can be constructed and is evaluated by the objective function (1).

The best solution is selected from candidate solutions in the neighborhood to update *current_sol*. If *current_sol* is better than *best_sol*, then *best_sol* is replaced with *current_sol* to ensure that *best_sol* stores the best solution found so far.

IV. Experimental Results

To verify its effectiveness, the proposed solution approach for the SRLP-SC is tested on multiple problem instances and is compared against an exact approach (CPLEX).

A. *Problem Instances*

SRLP-SC involves parameters m , w_i , a_i^l , a_i^r , c_{ij} , b_i^{lr} , and f_{ij} . The number of facilities, *m,* is chosen as 10, 20 and 30, respectively. There are a number of *product types* $p\sim$ unif[8,10]; the percentage of facilities visited by each product type is $r\sim$unif[0.25,0.75] and $n\sim$unif[20,50] is the number of products for each type. Parameter f_{ij} is calculated as the sum of products whose routes include facility *i* immediately preceding facility *j*. Parameter b_i^{lr} indicates whether both p_i^l and p_i^r need to be observed, such that more facilities with $b_i^{lr} = 0$ mean that more facilities require the determination of their additional clearance. The number of facilities with $b_i^{lr} = 0$ is set to *m*/2+1. Other parameters are created randomly by $w_i \sim$ unif[0.5,2.5], $c_{ij} \sim$ unif[0.25,1.5], $a_i^l \sim$ unif[0.2 w_i,0.25 w_i], and $a_i^r \sim$unif[0.2 w_i,0.25 w_i].

For each problem size (defined by the number of facilities), five problem instances are created.

B. *Algorithm Parameters*

Algorithm parameters of TS-HR include *tabu_size, iter_stop* and *iter_res*. Those parameters are set for different problem instances after short experiments, as shown in Table II. Generally, fewer iterations and a shorter length of the tabu list should be utilized for problems with a smaller number of facilities.

TABLE II
ALGORITHM PARAMETERS

Instances	*iter_stop*	*iter_res*	*tabu_size*
10_1-10_5	400	50	3
20_1-20_5	1000	100	5
30_1-30_5	2000	200	7

A. *Experimental results*

TS-HR was coded in Microsoft Visual Studio2010 and was executed in a Windows environment on a Dell desktop PC with 2.94GHZ Intel Core2 Duo CPU and 2.0G RAM.

Since no algorithms exist for this new problem (SRLP-SC), we compare the proposed TS-HR with an exact approach, a popular mathematical programming solver (CPLEX 12.4), which is used to solve the formulation of SRLP-SC in Section II B. While CPLEX can find the optimal solutions for small size problem instances with 10 facilities, it cannot obtain optimal solutions for large size instances with 20 or 30 facilities (due to the NP-hard property of the SRLP-SC). We restrict the computational time of CPLEX to be 3 and 5 hours for instances with 20 and 30 facilities, respectively.

Five independent runs of TS-HR were performed for each problem instance to obtain the average objective function values. For instances with 10 facilities, a comparison of TS-HR and CPLEX is given in Table III. From Table III, we can see that for instances 10_2, 10_3 and 10_4, each run of TS-HR is able to achieve the optimal solution that is found by CPLEX. For instances 10_1 and 10_5, the average cost obtained by TS-HR is very close to the optimal cost found by CPLEX. We can also observe that the run time of TS-HR is much less than that consumed by CPLEX. TS-HR only takes 18-23s to find high quality solutions while the computational time of CPLEX is about 3 hours.

For instances with 20 and 30 facilities, a comparison of TS-HR and CPLEX is presented in Table IV. In this table, the last column lists the cost values of solutions found by CPLEX within the given time. Table IV shows that for large size instances TS-HR can find solutions with less cost compared to CPLEX. For instances with 20 facilities, cost values obtained by TS-HR is about two thirds of those obtained by CPLEX while for instances with 30 facilities cost values found by TS-HR is only about half of those obtained by CPLEX. Standard deviations of the results in the five runs of TS-HR are also given in Table IV. We can observe that the standard deviations are quite small. For example, the standard deviation for instance 20_1 is 30.9, which only accounts for 0.23% of the average cost. A small standard deviation means that the performance of TS-HR is stable and each of its runs is able to obtain similar costs.

In addition, the run time of TS-HR is much shorter than that of CPLEX. TS-HR takes about 1-3 minutes for instances 20_1-20_5 and 18-27 minutes for instances 30_1-30_5 while CPLEX takes 3(5) hours for instances with 20 (30) facilities.

TABLE III
COMPARISON OF TS-HR WITH CPLEX FOR INSTANCES WITH 10 FACILITIES

Instances	TS-HR							CPLEX	
	1st run	2nd run	3th run	4th run	5th run	Average cost	Average time(s)	Optimal cost	Time(s)
10_1	3102.62	3102.62	3102.62	3102.62	3102.62	3102.62	**23.09**	3086.66	9405.66
10_2	3648.09	3648.09	3648.09	3648.09	3648.09	**3648.09**	**21.90**	3648.09	11015.87
10_3	4975.57	4975.57	4975.57	4975.57	4975.57	**4975.57**	**19.80**	4975.57	8266.65
10_4	5194.26	5194.26	5194.26	5194.26	5194.26	**5194.26**	**18.99**	5194.26	38327.98
10_5	7148.50	7148.50	7148.50	7148.50	7170.04	7152.81	**18.91**	7144.13	23090.60

TABLE IV
COMPARISON OF TS-HR WITH CPLEX FOR INSTANCES WITH 20 AND 30 FACILITIES

Instances	1st run	2nd run	3th run	4th run	5th run	Average cost	Average time(s)	Standard deviation	Cost found by CPLEX
20_1	13705.1	13705.1	13705.1	13774.6	13760.6	**13730.1**	**129.7**	30.9	-
20_2	17691.0	17691.0	17803.0	17874.5	17767.2	**17765.3**	**176.8**	68.2	21870.4
20_3	22727.5	22547.1	22635.9	22744.1	22727.5	**22676.4**	**106.6**	75.1	-
20_4	25025.7	24769.5	24780.9	24769.5	24769.5	**24823.0**	**182.1**	101.4	30613.3
20_5	33218.7	33330.8	33094.6	33094.6	33186.5	**33185.0**	**177.3**	88.0	39259.4
30_1	37283.1	37323.0	37283.1	37689.4	37323.0	**37380.3**	**1349.8**	155.6	90742.5
30_2	42898.4	42918.5	42854.8	42229.5	42622.9	**42704.8**	**1133.7**	260.1	80403.8
30_3	46048.3	45528.3	46062.6	46219.6	45481.0	**45868.0**	**1587.8**	328.2	102161.0
30_4	47464.3	47544.2	47420.0	47182.2	46935.5	**47309.2**	**1682.9**	222.4	90411.0
30_5	48871.0	49118.4	48993.0	48863.0	48433.6	**48855.7**	**1390.3**	230.8	104676.0

Note: "-" means that no feasible solution found in the limited run time.

V. CONCLUSION

In this paper, a new single row layout problem with shared clearance (SRLP-SC) is proposed, in which additional clearance is considered for each facility. A mixed-integer programming model is established for this problem. Since the SRLP-SC is NP-hard, the exact approach cannot find optimal solutions to large-size problems within a reasonable time. Therefore, an effective approach combining tabu search (TS) with a heuristic rule is proposed to resolve this problem. The TS is used to optimize the sequence of facilities while the heuristic rule is devised to determine decision variables related to additional clearances. Experimental results show that the proposed approach is able to find the optimal solutions in most cases for small size problem instances and outperforms an exact approach (CPLEX) under limited time for large size instances.

ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China under Grant 61374204.

REFERENCES

[1] D. M. Simmons, "One-dimensional space allocation: an ordering algorithm," *Operations Research*, vol. 17, no. 5, pp. 812-826, 1969.

[2] M. M. D. Hassan, "Machine layout problem in modern manufacturing facilities," *International Journal of Production research*, vol. 32, pp. 2559-2584, 2007.

[3] J. C. Picard and M. Queyranne, "On the one-dimensional space allocation problem," *Operations Research*, vol. 29, pp. 371-391, 1981.

[4] S. G. Ponnambalam and V. Ramkumar, "A genetic algorithm for the design of a single-row layout in automated manufacturing Systems," *International Journal of Advanced Manufacturing Technology*, vol. 18, pp. 512-519, 2001.

[5] M. Beghin-Picavet and P. Hansen, "Deux problèmes d'affectation non linéaires," *RAIRO - Operations Research - Recherche Opérationnelle*, vol. 16, pp. 263-276,1982.

[6] A. R. S. Amaral, "On the exact solution of a facility layout problem," *European Journal of Operational Research*, vol. 173, pp.508-518, 2006.

[7] M. F. Anjos, A. Kennings and A. Vannelli, "A semidefinite optimization approach for the single-row layout problem with unequal dimension," *Discrete Optimization*, vol. 2, pp. 113-122, 2005.
[8] A. R.S. Amaral, "A new lower bound for the single row facility layout problem," *Discrete Applied Mathematics,* vol. 157, pp. 183-190, Jan.2009.
[9] A. R. S. Amaral and A. N. Letchford, "A polyhedral approach to the single row facility layout problem," *Mathematical Programming*,vol. 141, pp. 453-477, 2013.
[10] D. Datta, A. R. S. Amaral and J. R. Figueira, "Single row facility layout problem using a permutation-based genetic algorithm," *European Journal of Operational Research*, vol. 213, pp. 388-394, 2011.
[11] M. Braglia, "Optimisation of a simulated-annealing-based heuristic for single row machine layout problem by genetic algorithm," *International Transactions in Operational Research*, vol. 3, pp. 37-49, 1996.
[12] H. Samarghandi, P. Taabayan and F. F. Jahantigh,"A particle swarm optimization for the single row facility layout problem," *Computers &industrial Engineering*, vol. 40,pp. 529-534, 2010.
[13] M. Solimanpur, P. Vrat and R. Shankar, "Ant colony optimization algorithm to the inter-cell layout problem in cellular manufacturing," *European Journal of Operational Research*, vol.157, pp. 592-606, 2004.
[14] H. Samarghandi and K. Eshghi, "An efficient tabu algorithm for the single row facility layout problem," *European Journal of Operational Research*, vol. 205, pp. 98-105, 2010.
[15] K. R. Kumar, G. C. Hadjinicola and T. L. Lin, "A heuristic procedure for the single-row facility layout problem," *European Journal of Operational Research*, vol. 87, pp. 65-73, 1995.
[16] Y. C. Ho and C. L. Moodie, "Machine layout with a linear single-row flow path in an automated manufacturing system," *Journal of Manufacturing System*s, vol. 17, pp. 1-22, 1998.
[17] C. O. Yang and A. Utamima, "Hybrid estimation of distribution algorithm for solving single-row facility layout problem," *Computers& Industrial Engineering*, vol. 66, pp.95-103, 2013.
[18] F. Glover, "Future paths for integer programming and links to artificial intelligence," Computers & Operations Research, vol. 13, pp. 533-549, 1986.