Supplementary material: An efficient Variable Neighborhood Search for the Space-Free Multi-Row Facility Layout problem[☆]

Alberto Herrán, J. Manuel Colmenar, Abraham Duarte[*]

Dept. Computer Sciences, Universidad Rey Juan Carlos
C/. Tulipán, s/n, Móstoles, 28933 (Madrid), Spain

Abstract

The Space-Free Multi-Row Facility Layout problem (SF-MRFLP) seeks for a non-overlapping layout of departments (facilities) on a given number of rows satisfying the following constraints: no space is allowed between two adjacent facilities and the left-most department of the arrangement must have zero abscissa. The objective is to minimize the total communication cost among facilities. In this paper, a Variable Neighborhood Search (VNS) algorithm is proposed to solve this $\mathcal{NP}$-Hard problem. It has practical applications in the context of the arrangement of rooms in buildings, semiconductor wafer fabrication, or flexible manufacturing systems. A thorough set of preliminary experiments is conducted to evaluate the influence of the proposed strategies and to tune the corresponding search parameters. The best variant of our algorithm is tested over a large set of 528 instances previously used in the related literature. Experimental results show that the proposed algorithm improves the state-of-the-art methods, reaching all the optimal values or, alternatively, the best-known values (if the optimum is unknown) but in considerably shorter computing times. These results are further confirmed by conducting a Bayesian statistical analysis.

[☆]This work has been partially supported by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU/AEI/FEDER, UE) under grant ref. PGC2018-095322-B-C22; and Comunidad de Madrid y Fondos Estructurales de la Unión Europea with grant ref. P2018/TCS-4566.

[*]Corresponding author

Email address: `alberto.herran@urjc.es`, `josemanuel.colmenar@urjc.es`, `abraham.duarte@urjc.es` (Alberto Herrán, J. Manuel Colmenar, Abraham Duarte)

Appendix I

This appendix details how to efficiently compute both, *exchange* and *insert* moves boosting the exploration of neighborhoods. For this purpose, we define in Equation (1) the cost variation between two departments u and v which modify their positions from $\delta_\varphi(u)$ and $\delta_\varphi(v)$ to $\delta_{\varphi'(u)}$ and $\delta_{\varphi'(v)}$, respectively, after the move is performed. Notice that, in order to compute the effect of a move, there is no need to modify the current solution from φ to φ', but to provide the new position $\delta_{\varphi'(d)}$ after the move of each involved department d.

$$\Delta\left(\varphi, u, v, \delta_{\varphi'(u)}, \delta_{\varphi'(v)}\right) = w_{uv} \cdot \left(\left|\delta_{\varphi'(u)} - \delta_{\varphi'(v)}\right| - \left|\delta_{\varphi(u)} - \delta_{\varphi(v)}\right|\right) \tag{1}$$

Exchange move

Regarding the *exchange* move, we first consider the situation depicted in Figure 1, where both involved departments, u and v, are located in the same row i, with $\varphi(u) = (i, j)$ and $\varphi(v) = (i, l)$. In this figure, we identify four different zones in the layout: zones 1, 2 and 3 for the departments d such that $\varphi(d) = (i, t)$ with $t < j$, $j < t < l$ and $t > l$ respectively; and zone 4, for all the departments located in rows $r \neq i$. Notice that the relative position among departments with the same color does not change after the move.

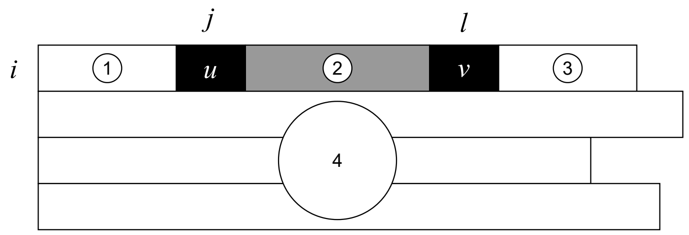

Figure 1: *Exchange* move between u and v from the same row. Four numbered zones are defined, according to the change on the cost devoted to their department.

As we can see, in this case, only u, v and the departments located in zone 2 change their position in the layout. Hence, the cost variation due to this move can be calculated by Equation (2). Notice that we assume that u is closer to the left-hand side of the layout than v. Thus, the first two terms of the equation calculate the cost change due to departments u and v, whose new position in the layout after the move is given by $\delta_{\varphi'(u)} = \delta_{\varphi(v)} - L_u/2 + L_v/2$ and $\delta_{\varphi'(v)} = \delta_{\varphi(u)} - L_u/2 + L_v/2$, with all the departments d in zones 1, 3 and 4, whose position does not change after the move. Similarly, the third and fourth terms calculate the cost change due to departments u and v with all the departments d in zone 2, whose new position in the layout after the move is given by $\delta_{\varphi'(d)} = \delta_{\varphi(d)} - L_u + L_v$. Finally, the last term summarizes the contribution to the cost due to all the departments in zone 2 with all the departments in zones 1, 3 and 4.

$$\begin{aligned}\Delta_{exc-\texttt{same}}(\varphi, u, v) = \; & \sum_{d \in D_{\{1,3,4\}}} \Delta\left(\varphi, u, d, \delta_{\varphi'(u)}, \delta_{\varphi(d)}\right) + \\ & \sum_{d \in D_{\{1,3,4\}}} \Delta\left(\varphi, v, d, \delta_{\varphi'(v)}, \delta_{\varphi(d)}\right) + \\ & \sum_{d \in D_2} \Delta\left(\varphi, u, d, \delta_{\varphi'(u)}, \delta_{\varphi'(d)}\right) + \\ & \sum_{d \in D_2} \Delta\left(\varphi, v, d, \delta_{\varphi'(v)}, \delta_{\varphi'(d)}\right) + \\ & \sum_{d \in D_2} \sum_{e \in D_{\{1,3,4\}}} \Delta\left(\varphi, d, e, \delta_{\varphi'(d)}, \delta_{\varphi(e)}\right)\end{aligned} \tag{2}$$

Next, we consider the case where departments u and v are located in different rows in the *exchange* move. Let us assume that u is located in row i, and v in row $k \neq i$. In this case, we identify 5 different zones in the layout: zones 1 and 2 for departments d such that $\varphi(d) = (i, t)$ with $t < j$ and $t > j$ respectively; zones 3 and 4 for departments d such that $\varphi(d) = (k, t)$ with $t < l$ and $t > l$ respectively; and zone 5, for all the departments located in rows $r \neq i$ and $r \neq k$. Figure 2 depicts these five different zones for this case using the same color legend for the departments than the one used in Section 3.2 of the paper.

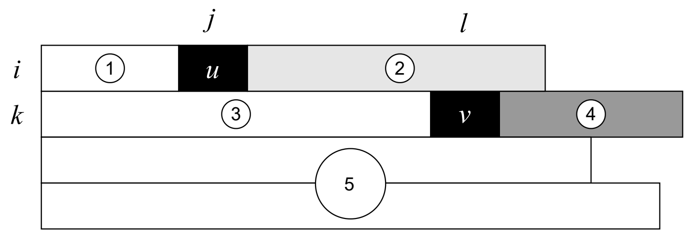

Figure 2: *Exchange* move between u and v from different rows. Five numbered zones are defined, according to the change on the cost devoted to their department.

As we can see, in this case, only u, v and the departments located in zones 2 and

4 change their position in the layout. Hence, the cost variation due to this move can be calculated by Equation (3). The first term obtains the cost variation due to the direct cost between u and v, whose new position in the layout after the move, in this case, is given by $\delta_{\varphi'(u)} = \delta_{\varphi(v)} - L_v/2 + L_u/2$ and $\delta_{\varphi'(v)} = \delta_{\varphi(u)} - L_u/2 + L_v/2$. The second and third terms calculate the cost change due to departments u and v with all the departments d in zones 1, 3 and 5, whose position does not change after the move. Similarly, the fourth and fifth terms calculate the cost change due to departments u and v with all the departments d in zones 2 and 4, whose new position in the layout after the move is given by $\delta_{\varphi'(d)} = \delta_{\varphi(d)} - L_u + L_v$ for $d \in D_2$ and $\delta_{\varphi'(d)} = \delta_{\varphi(d)} - L_v + L_u$ for $d \in D_4$. Finally, the sixth term calculates the contribution between departments d and e in zones 2 and 4, respectively, and the last term summarizes the contribution to the cost due to all the departments in zones 2 and 4 with all the departments in zones 1, 3 and 5.

$$\begin{aligned}\Delta_{exc-\mathtt{diff}}(\varphi,u,v) = \quad & \Delta\left(\varphi,u,v,\delta_{\varphi'(u)},\delta_{\varphi'(v)}\right) + \\ & \sum_{d\in D_{\{1,3,5\}}} \Delta\left(\varphi,u,d,\delta_{\varphi'(u)},\delta_{\varphi(d)}\right) + \\ & \sum_{d\in D_{\{1,3,5\}}} \Delta\left(\varphi,v,d,\delta_{\varphi'(v)},\delta_{\varphi(d)}\right) + \\ & \sum_{d\in D_{\{2,4\}}} \Delta\left(\varphi,u,d,\delta_{\varphi'(u)},\delta_{\varphi'(d)}\right) + \\ & \sum_{d\in D_{\{2,4\}}} \Delta\left(\varphi,v,d,\delta_{\varphi'(v)},\delta_{\varphi'(d)}\right) + \\ & \sum_{d\in D_2}\sum_{e\in D_4} \Delta\left(\varphi,d,e,\delta_{\varphi'(d)},\delta_{\varphi'(e)}\right) + \\ & \sum_{d\in D_{\{2,4\}}}\sum_{e\in D_{\{1,3,5\}}} \Delta\left(\varphi,d,e,\delta_{\varphi'(d)},\delta_{\varphi(e)}\right)\end{aligned} \tag{3}$$

Insert move

The proposal for the *insert* move is different. The idea is to improve the efficiency of the evaluation after this move by implementing it in a more systematic way Schiavinotto & Stützle (2005). Specifically, the complexity of evaluating moves can be dramatically improved if they are implemented interchanging the positions between consecutive departments in the same row. This fact has been extensively reported in the related literature (see Kothari & Ghosh (2014) Rubio-Sánchez et al. (2016) Palubeckis (2015)).

In order to illustrate how we can adapt this strategy to the context of the SF-MRFLP, we consider the situation depicted in Figure 3, where department `D` will be inserted in different positions of its current row. Again, the relative position among departments with the same color does not change after the move in all the following figures.

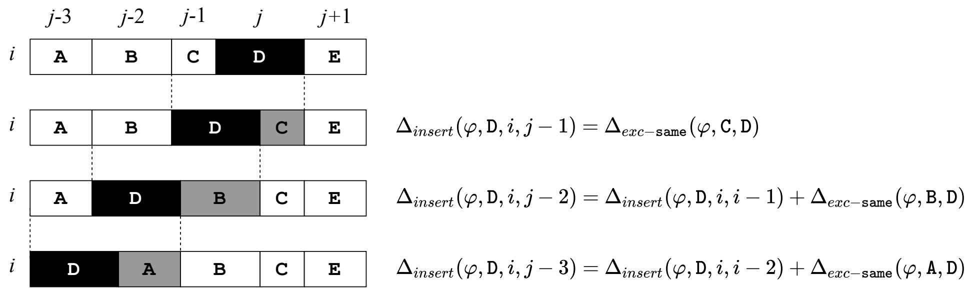
j-3
j-2
j-1
j
j+1
i
A
B
C
D
E
$\Delta_{insert}(\varphi, \mathtt{D}, i, j-1) = \Delta_{exc-\mathtt{same}}(\varphi, \mathtt{C}, \mathtt{D})$
$\Delta_{insert}(\varphi, \mathtt{D}, i, j-2) = \Delta_{insert}(\varphi, \mathtt{D}, i, i-1) + \Delta_{exc-\mathtt{same}}(\varphi, \mathtt{B}, \mathtt{D})$
$\Delta_{insert}(\varphi, \mathtt{D}, i, j-3) = \Delta_{insert}(\varphi, \mathtt{D}, i, i-2) + \Delta_{exc-\mathtt{same}}(\varphi, \mathtt{A}, \mathtt{D})$

Figure 3: Insertion of department D in different positions through sequential *swap* moves.

Assuming that department D is located at position $\varphi(\mathtt{D}) = (i, j)$, three *insert* moves are depicted in the figure, with destinations from $j-1$ to $j-3$ of the same row i. The cost change due to these three moves can be efficiently computed if they are performed in the order shown in Figure 3. This way of traversing the insertions of department D makes it possible to efficiently evaluate the associated cost change using the Equation (2) proposed for the *exchange* moves. Moreover, this equation can be further simplified when the *exchange* move is performed between departments with consecutive positions in the same row, since these moves are the ones that require less calculations (only departments u and v change their position in the layout). We call this exchange between two departments u and v located in adjacent positions, (i, j) and $(i, j+1)$, respectively, a *swap* move, and its corresponding cost change is denoted as $\Delta_{swap}(\varphi, u, v)$.

In order to show how to calculate $\Delta_{swap}(\varphi, u, v)$, we consider the situation depicted in Figure 4. As we can see, this situation is similar than the one depicted in Figure 1 but without any department between u and v. Hence, Equation (2) can be simplified to Equation (4), which only calculates the cost change due to departments u and v, whose new position in the layout after the move is given by $\delta_{\varphi'(u)} = \delta_{\varphi(u)} - L_v/2$ and $\delta_{\varphi'(v)} = \delta_{\varphi(v)} - L_u$, with all the departments d in zones 1, 3 and 4, whose position does not change after the move.

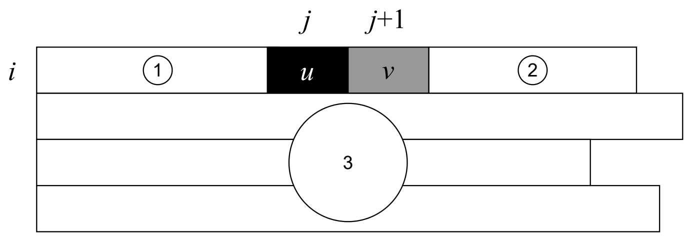

Figure 4: Swap of two departments u and v.

$$\Delta_{swap}(\varphi, u, v) = \sum_{d \in D_{\{1,2,3\}}} \Delta\left(\varphi, u, d, \delta_{\varphi'(u)}, \delta_{\varphi(d)}\right) + \sum_{d \in D_{\{1,2,3\}}} \Delta\left(\varphi, v, d, \delta_{\varphi'(v)}, \delta_{\varphi(d)}\right) \quad (4)$$

Figure 5 shows how a department u located at position j of row i can be inserted at any other position in the layout by iteratively exchanging it with a department located one position before or after u in the same row. Starting from the situation depicted in Figure 5(a), Equation (4) can be used to evaluate all the possible insertions of u, by performing consecutive *swap* moves, from positions with $l = j - 1$ (Figure 5(b)) to the first position of row i (Figure 5(c)). The same procedure can be applied to explore all the possible insertions of u in positions from $l = j + 1$ (Figure 5(d)) to the last position of the row i (Figure 5(e)). Once all the possible insertions of department u in row i have been evaluated, the insertions in the other rows $k \neq i$ can be performed. To this purpose, two costs must be calculated: the cost change of removing the department u when it is located at the end of row i (Figure 5(f)), and the cost change when adding it at the end of a different row k (Figure 5(g)). These values can be calculated by equations 5 and 6 respectively, where the new position in the layout after the add move is given by $\delta_{\varphi'(u)} = \sum_{v \in D_k} L_v + L_u/2$. Then, Equation (4) can be used again to evaluate all the possible insertions of u in row k from position $|\varphi_k|$ (Figure 5(h)) to the first position of row k (Figure 5(i)). Finally, the steps shown in Figures 5(g) to Figure 5(i) are repeated for all the rows $k' \neq i$ in the layout (Figure 5(j)).

$$\Delta_{drop}(\varphi, u) = -\sum_{d \in D} \Delta\left(\varphi, u, d, \delta_{\varphi(u)}, \delta_{\varphi(d)}\right) \quad (5)$$

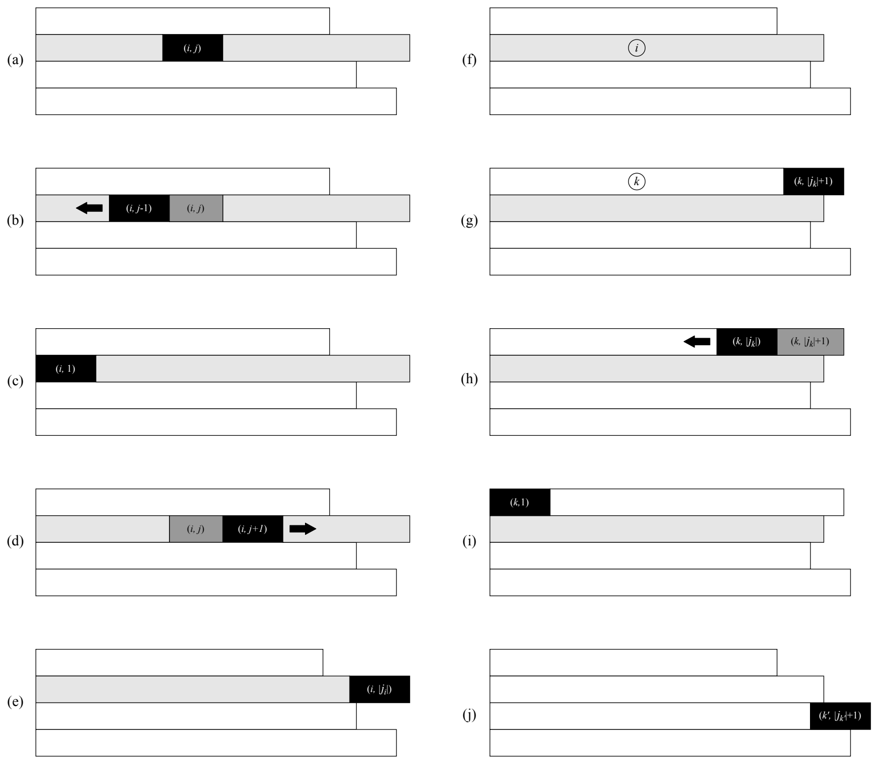

Figure 5: Sequence of *swap* moves to perform all the possible insertions of a department u with $\varphi(u) = (i, j)$ in the layout.

$$\Delta_{add}(\varphi, u) = \sum_{d \in D} \Delta\left(\varphi, u, d, \delta_{\varphi'(u)}, \delta_{\varphi(d)}\right) \tag{6}$$

Appendix II

Tables 1, 2 and 3 summarize the main features of the sets of instances used in our experimental experience. These sets are taken from (Ghosh & Kothari, 2012), (Ahonen et al., 2014) and (Fischer et al., 2019), respectively. Notice that several instances appear in different sets from different papers. We decided to maintain this distribution in the reported results in order to compare with our results.

Specifically, Table 1 shows the set of instances used in Ghosh & Kothari (2012). These are a set of small instances with $n \in [9, 15]$ taken from Simmons (1969), Amaral (2012) and Anjos & Vannelli (2008), and a larger set of instances with $n \in$

[25, 49] taken from Anjos & Vannelli (2008), Anjos & Yen (2009) and Hungerlnder & Rendl (2012).

Table 1: Features of instances from Ghosh & Kothari (2012).

Name	n	$L_{[min,max]}$	w_{max}	**Name**	n	$L_{[min,max]}$	w_{max}
S9	9	[2, 9]	8	P5	35	[3, 9]	10
S9H	9	[5, 8]	8	P6	35	[3, 9]	10
S10	10	[2, 9]	12	ste36_01	36	[1, 1]	316
S11	**11**	**[3, 10]**	**20**	ste36_02	36	[1, 37]	316
Am12a	12	[3,20]	15	ste36_03	36	[1, 19]	316
Am12b	12	[3, 9]	15	ste36_04	36	[1, 19]	316
Am13a	13	[3, 20]	10	ste36_05	36	[1, 19]	316
Am13b	13	[3, 21]	10	N40_01	40	[2, 10]	10
Am15	**15**	**[3, 20]**	**10**	N40_02	40	[1, 10]	10
N25_01	25	[1, 1]	10	N40_03	40	[1, 10]	10
N25_02	25	[1, 15]	10	N40_04	40	[1, 10]	10
N25_03	25	[1, 10]	10	N40_05	40	[1, 30]	10
N25_04	25	[1, 20]	10	sko42_01	42	[1, 1]	10
N25_05	25	[1, 6]	10	sko42_02	42	[1, 21]	10
N30_01	30	[1, 1]	10	sko42_03	42	[2, 16]	10
N30_02	30	[1, 5]	10	sko42_04	42	[1, 11]	10
N30_03	30	[1, 10]	10	**sko42_05**	**42**	**[1, 20]**	**10**
N30_04	30	[1, 15]	10	sko49_01	49	[1, 1]	10
N30_05	**30**	**[1, 30]**	**10**	sko49_02	49	[1, 21]	10
P1	33	[3, 9]	10	sko49_03	49	[1, 16]	10
P2	33	[3, 19]	10	sko49_04	49	[1, 11]	10
P3	33	[3, 19]	10	**sko49_05**	**49**	**[1, 30]**	**10**
P4	35	[3, 9]	10				

Table 2 summarizes the main features of the set of instances solved in (Ahonen et al., 2014). The full set of instances is divided into 4 subsets. The *Small* set is composed again by 9 instances with $n \in [9, 15]$ taken from Simmons (1969), Amaral (2012) and Anjos & Vannelli (2008). The *Medium* set has 5 instances instances with $n = 30$ from Anjos & Vannelli (2008). The *Large* set is a selection of the `sko` instances with $n \in [42, 64]$ from Anjos & Yen (2009) plus two other `AKV` instances from Anjos et al. (2005) with $n = 60$ and $n = 70$, respectively. In addition, the *Extra* set has 36 randomly generated instances having $n = 60$ from Ahonen et al. (2014).

Finally, Table 3 summarizes the main features of the set of instances solved in Fischer et al. (2019). This set also contains the 9 instances with $n \in [9, 15]$, plus other `Am` and `Small` instances taken from Amaral (2018), and a set of `HA` instances from Hungerlnder & Rendl (2012).

Table 2: Features of instances from (Ahonen et al., 2014).

Set	**Name**	n	$L_{[min,max]}$	w_{max}
Small	S9	9	[2, 9]	8
	S9H	9	[5, 8]	8
	S10	10	[2, 9]	12
	S11	**11**	**[3, 10]**	**20**
	Am12a	12	[3,20]	15
	Am12b	12	[3, 9]	15
	Am13a	13	[3, 20]	10
	Am13b	13	[3, 21]	10
	Am15	**15**	**[3, 20]**	**10**

Set	**Name**	n	$L_{[min,max]}$	w_{max}
Medium	N30_01	30	[1, 1]	10
	N30_02	30	[1, 5]	10
	N30_03	30	[1, 10]	10
	N30_04	30	[1, 15]	10
	N30_05	**30**	**[1, 30]**	**10**

Set	**Name**	n	$L_{[min,max]}$	w_{max}
Large	sko42_01	42	[1, 1]	10
	sko42_02	42	[1, 21]	10
	sko42_03	42	[2, 16]	10
	sko42_04	42	[1, 11]	10
	sko42_05	**42**	**[1, 20]**	**10**
	sko49_01	49	[1, 1]	10
	sko49_02	49	[1, 21]	10
	sko49_03	49	[1, 16]	10
	sko49_04	49	[1, 11]	10
	sko49_05	**49**	**[1, 30]**	**10**
	sko56_01	56	[1, 1]	10
	sko56_02	56	[1, 15]	10
	sko56_03	56	[1, 5]	10
	sko56_04	56	[1, 10]	10
	sko56_05	**56**	**[1, 20]**	**10**
	AKV_60_05	**60**	**[1, 59]**	**4**
	sko64_05	**64**	**[1, 10]**	**10**
	AKV_70_05	70	[2, 70]	16

Set	**Name**	n	$L_{[min,max]}$	w_{max}
Extra	CAP_n60_p30_s30_1	60	[1, 20]	10
	CAP_n60_p30_s30_2	60	[1, 20]	10
	CAP_n60_p30_s30_3	**60**	**[1, 20]**	**10**
	CAP_n60_p30_s40_1	60	[1, 20]	10
	CAP_n60_p30_s40_2	60	[1, 20]	10
	CAP_n60_p30_s40_3	**60**	**[1, 20]**	**10**
	CAP_n60_p30_s50_1	60	[1, 20]	10
	CAP_n60_p30_s50_2	60	[1, 20]	10
	CAP_n60_p30_s50_3	**60**	**[1, 20]**	**10**
	CAP_n60_p30_s60_1	60	[1, 10]	10
	CAP_n60_p30_s60_2	60	[1, 10]	10
	CAP_n60_p30_s60_3	**60**	**[1, 10]**	**10**
	CAP_n60_p60_s30_1	60	[1, 20]	10
	CAP_n60_p60_s30_2	60	[1, 20]	10
	CAP_n60_p60_s30_3	**60**	**[1, 20]**	**10**
	CAP_n60_p60_s40_1	60	[1, 20]	10
	CAP_n60_p60_s40_2	60	[1, 20]	10
	CAP_n60_p60_s40_3	60	[1, 20]	10
	CAP_n60_p60_s50_1	60	[1, 20]	10
	CAP_n60_p60_s50_2	60	[1, 20]	10
	CAP_n60_p60_s50_3	**60**	**[1, 20]**	**10**
	CAP_n60_p60_s60_1	60	[1, 10]	10
	CAP_n60_p60_s60_2	60	[1, 10]	10
	CAP_n60_p60_s60_3	60	[1, 10]	10
	CAP_n60_p90_s30_1	60	[1, 20]	10
	CAP_n60_p90_s30_2	60	[1, 20]	10
	CAP_n60_p90_s30_3	60	[1, 20]	10
	CAP_n60_p90_s40_1	60	[1, 20]	10
	CAP_n60_p90_s40_2	60	[1, 20]	10
	CAP_n60_p90_s40_3	60	[1, 20]	10
	CAP_n60_p90_s50_1	60	[1, 20]	10
	CAP_n60_p90_s50_2	60	[1, 20]	10
	CAP_n60_p90_s50_3	60	[1, 20]	10
	CAP_n60_p90_s60_1	60	[1, 10]	10
	CAP_n60_p90_s60_2	60	[1, 10]	10
	CAP_n60_p90_s60_3	**60**	**[1, 10]**	**10**

Table 3: Features of instances from (Fischer et al., 2019).

Name	n	$L_{[min,max]}$	w_{max}
Am11a	11	[5, 21]	20
Am11b	11	[4, 9]	20
Am11c	11	[4, 9]	21
Am11d	11	[3, 12]	6
Am11e	11	[3, 12]	6
Am11f	11	[3, 11]	6
Am12a	12	[3, 20]	10
Am12b	12	[3, 9]	15
Am12c	12	[3, 13]	15
Am12d	12	[6, 12]	6
Am12e	12	[6, 12]	6
Am12f	12	[3, 12]	6
Am13a	13	[3, 20]	10
Am13b	13	[3, 21]	10
Am13c	13	[3, 21]	10
Am13d	13	[3, 12]	19
Am13e	13	[3, 12]	19
Am13f	13	[3, 11]	19
HA5	5	[2, 10]	9
HA6	6	[2, 10]	9
HA7	7	[2, 10]	9
HA8	8	[2, 10]	9
HA9	9	[2, 10]	9
HA10	10	[2, 10]	10
HA11	11	[2, 10]	10
HA12	12	[2, 10]	10
HA13	13	[2, 10]	10
Small.09-1	9	[5, 8]	8
Small.09-2	9	[5, 8]	8
Small.09-3	9	[5, 7]	8
Small.09-4	9	[5, 7]	8
Small.09-5	9	[7, 10]	10
Small.09-6	9	[7, 10]	10
Small.09-7	9	[7, 10]	10
Small.09-8	9	[7, 10]	10
Small.09-9	9	[7, 10]	4
Small.09-10	9	[7, 10]	4
Small.10-1	10	[5, 10]	13
Small.10-2	10	[5, 10]	12
Small.10-3	10	[3, 9]	11
Small.10-4	10	[3, 9]	11
Small.10-5	10	[3, 8]	12
Small.10-6	10	[3, 8]	9
Small.10-7	10	[3, 8]	11
Small.10-8	10	[3, 8]	9
Small.10-9	10	[3, 9]	9
Small.10-10	10	[3, 9]	10
S9	9	[2, 9]	8
S9H	9	[5, 8]	8
S10	10	[2, 9]	12
S11	**11**	**[3, 10]**	**20**

References

Ahonen, H., de Alvarenga, A., & Amaral, A. (2014). Simulated annealing and tabu search approaches for the corridor allocation problem. *European Journal of Operational Research*, *232*, 221 – 233.

Amaral, A. (2018). A mixed-integer programming formulation for the double row layout of machines in manufacturing systems. *International Journal of Production Research*, *57*, 1–14.

Amaral, A. R. (2012). The corridor allocation problem. *Computers and Operations Research*, *39*, 3325 – 3330.

Anjos, M. F., Kennings, A., & Vannelli, A. (2005). A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization*, *2*, 113 – 122.

Anjos, M. F., & Vannelli, A. (2008). Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS Journal on Computing*, *20*, 611–617.

Anjos, M. F., & Yen, G. (2009). Provably near-optimal solutions for very large single-row facility layout problems. *Optimization Methods and Software*, *24*, 805–817.

Fischer, A., Fischer, F., & Hungerlnder, P. (2019). New exact approaches to row layout problems. *Math. Program. Comput.*, *11*, 703–754.

Ghosh, D., & Kothari, R. (2012). *Population Heuristics for the Corridor Allocation Problem*. IIMA Working Papers WP2012-09-02 Indian Institute of Management Ahmedabad, Research and Publication Department.

Hungerlnder, P., & Rendl, F. (2012). A computational study and survey of methods for the single-row facility layout problem. *Computational Optimization and Applications*, *55*.

Kothari, R., & Ghosh, D. (2014). An efficient genetic algorithm for single row facility layout. *Optimization Letters*, *8*, 679–690.

Palubeckis, G. (2015). Fast local search for single row facility layout. *European Journal of Operational Research*, *246*, 800 – 814.

Rubio-Sánchez, M., Gallego, M., Gortázar, F., & Duarte, A. (2016). GRASP with path relinking for the single row facility layout problem. *Knowledge-Based Systems*, *106*, 1 – 13.

Schiavinotto, T., & Stützle, T. (2005). The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, *3*, 367–402.

Simmons, D. M. (1969). One-dimensional space allocation: An ordering algorithm. *Operations Research*, *17*, 812–826.