A Mixed-Integer Linear Programming Formulation for the Modular Layout of Three-Dimensional Connected Systems

Sam O'Neill*, Paul Wrigley, Ovidiu Bagdasar

Department of Electronics, Computing and Mathematics, University of Derby, Kedleston Road, DE22 1GB, United Kingdom

---

Abstract

Given the considerable complexity of process plants, there has been a great deal of research focused on aiding the design of plant layout through mathematical optimisation, i.e. optimising the positioning of the equipment in the plant for space and cost efficiency. Recently, the use of modular approaches within the construction industry, whereby work is performed off-site before being assembled on-site, has become a popular and powerful way of reducing build schedules and costs. Modular approaches have many other real applications where items must be packed to minimise the connections between them (e.g. piping, wiring, modular office and factory layouts) and consider the modular layout of the system.

In this paper, we provide a formulation of the problem that, in addition to the standard layout problem, considers a modular block layout to allow modular construction and transportation of the plant. The problem is represented as a directed network, with the aim to pack the items into predefined containers and minimise the rectilinear distance between the connected items. We propose mixed-integer linear programming (MILP) models for the 2-dimensional and 3-dimensional problems and solve them using the state-of-the-art mathematical programming solver, Gurobi. Because of the combinatorial nature of the problem, solutions involving a large number of items may not converge and a suboptimal solution must be considered. However, our results suggest that even in the case of optimising a large number of items, the suboptimal solutions found after a reasonable number of iterations where deemed, by a domain expert, to be a good enough starting point to continue the design process, especially in the early concept phase.

Keywords: Floor packing, Bin packing, Facility layout, mixed-integer linear programming, MILP, Process plant design.

---

1. Introduction

In recent decades, owing to a decline in productivity in the construction industry (Bock [7]), the use of modular construction, whereby work is performed off-site in more productive factories, has been shown to take work off the critical path, reduce schedules and save costs (Hosseini et al. [15], Jin et al. [18]). In industrial process plants this can mean more certainty in construction, reducing risks and thereby investment costs. Large modularisation, moving work to an onsite assembly area such as in shipbuilding (Jaquith et al. [16], Barry [5]), has been applied to remote, weather adverse process plant locations in the oil and gas (Smith and Bowtell [36]) and nuclear industries (Lapp and Golay [21], Sutharshan et al. [37]). Recent research has been focused

*Corresponding author

Email addresses: S.Oneill@derby.ac.uk (Sam O'Neill), p.wrigley@derby.ac.uk (Paul Wrigley), o.bagdasar@derby.ac.uk (Ovidiu Bagdasar)

on smaller, factory-built plants (Seifert et al. [35]) and has concentrated on earlier parts of the plant design process (database creation, equipment selection, modular Process and Instrumentation Diagrams).

Plant layouts are normally complex and have traditionally been accomplished using the heuristic knowledge of plant design engineers (Moran [27]). Heuristic methods were first developed in the 1970s, to help the plant design engineers achieve the optimum layout, using a learning procedure to lay out a chemical process unit such that the operational costs of the plant were minimised (Amorese et al. [2], Rosenblatt [34]). Kern [20], developed techniques to implement and order different equipment in a plant.

Layout optimisation can assess a multitude of possible layouts in a fraction of the time, especially as designs get more complex, providing the plant designers with an aid to designing the plant. The expert heuristic knowledge of the plant designers must be captured and codified in some way. Throughout the design process, layout optimisation can be utilised to reduce design time and achieve more cost-effective results.

Early mathematical layout optimisation research explored the layout of a chemical plant using a hill climb and Nonlinear Programming (NLP) approach, considering piping and building costs (Gunn and Al-Asadi [12]). Jayakumar and Reklaitis [17] introduced graph partitioning to minimise connections between components. Barbosa-Póvoa and Macchietto [4] developed a mixed-integer nonlinear programming (MINLP) formulation for the optimal selection of both the equipment units and the network of connections to satisfy production requirements in a multipurpose batch plant. The objective function was to minimise the capital costs and to maximize the plant profit. Penteado and Ciric [33] introduced a mixed-integer nonlinear programming approach for safe process plant layouts, minimising the cost of piping, land cost, financial risk, and device protection cost.

Georgiadis and Macchietto [9] applied a mixed-integer linear programming (MILP) approach to the chemical process plant layout problem by considering the allocation of equipment to equal area locations and allocation of equipment to non-equal areas [10]. The objective function considers pipe, land and pumping costs. This was developed further with continuous domain single floor (Papageorgiou and Rotstein [28]) and multi floor (Patsiatzis and Papageorgiou [32]) and a decomposition and an iterative approach (Patsiatzis and Papageorgiou [30]). They also suggested an mixed-integer nonlinear programming model by adopting rectangular shapes and rectilinear distances (Patsiatzis and Papageorgiou [31]). Guirardello and Swaney [11] consider pipe routings and Martinez-Gomez et al. [25] future expansions.

Metaheuristic approaches have been proposed to add functionality to mathematical programming or to produce a near optimal solution to a problem which would be computationally too time-intensive to search fully. Suzuki et al. [38] proposed an evolutionary algorithm approach for plant layouts as it is difficult to solve costs and preferences simultaneously using mixed-integer linear programming. Piping and site costs are considered in the objective function as weighted preferences. More recent methods include genetic algorithm approaches (Balakrishnan et al. [3], Xu et al. [44], Lee [24], Wu et al. [42], Wang et al. [39]) and genetic algorithm approaches assisted by a quadrilateral packing algorithm (Wang et al. [40]). Other methods were developed such as coevolutionary, hyper-heuristic approaches (Furuholmen et al. [8]), particle swarm optimization, (Lee [23], Park and Lee [29], Yang and Lee [45]), Monte Carlo simulation integrated with simulated annealing (Alves et al. [1], He et al. [14]), bat-inspired metaheuristic algorithm (Latifi et al. [22]), Kruskal's algorithm (Wu and Wang [43]), simulated annealing, two stage optimisation process (Jung et al. [19]).

The authors are not aware of the use of a modular layout being previously considered in either plant design or layout optimisation. This paper considers the problem of finding an optimal modular layout of a directed network, with the aim of packing the vertices (items) into predefined containers (modules), whilst minimising the rectilinear distance of the edges (connections between items). Viewed as a process plant layout optimisation problem the aim is to pack the set of connected equipment into a set of containers (modules) such that the cost is minimised, thus allowing off-site construction and simple assembly on-site.

The structure of the paper is as follows. In Section 2, a description of the problem and the assumptions made in this paper are given. Section 3 provides four mixed-integer linear programming models (full models are given in Appendix A). Section 4 illustrates the results for a small process plant example, analyses the performance of the models on a randomly generated dataset, using the state-of-the-art mathematical programming solver Gurobi [13] [26] and gives preliminary results for a modular 3D layout of a process plant system. Finally, in Section 5, we present our concluding remarks and areas of future work.

2. Problem Description

Let $G = (V, E)$ be a directed network where V is the set of items (vertices) and E is a set of connections (edges). The cost of a connection (i, j) is denoted by c_{ij}, representing the unit cost of the edge distance. The dimensions and positions of an item are given by (w_v, h_v, d_v) and (x_v, y_v, z_v), respectively. For each connection (i, j), item i is connected via a relative shift from the centre point of item i by $(xs_{ij}, ys_{ij}, zs_{ij})$ and similarly for item j by $(xs_{ji}, ys_{ji}, zs_{ji})$. Finally, we have a set K of $n \geq 1$ containers, and for each $k \in K$ the size, position and cost mappings are given by (W_k, H_k, D_k), (mx_k, my_k, mz_k) and g_k, respectively.

Our primary objective is to obtain an optimal (or good) partitioning of the items of the directed network G into the set of containers K that minimises the combined cost of the connections and the containers used.

2.1. Basic assumptions

To model the problem as a mixed-integer linear programming problem, the following basic assumptions are made:

AS1 Items are rectangular in shape.

AS2 Containers are rectangular in shape.

AS3 Items must belong to one and only one container.

AS4 Items within the same container cannot overlap each other.

AS5 Containers do not overlap.

AS6 The position of containers is fixed.

AS7 Connections between items follow a rectilinear (taxicab) geometry.

AS8 Distance between connected items is calculated from the centre points of the items.

AS9 Coordinates of items and containers are calculated from the origin (distances for items in different containers take into account the relative offset due to container positioning).

AS10 Items can be rotated $90°, 180°, 270°$ in the xy plane.

AS11 Items are fixed to the floor, i.e. a z-axis coordinate of 0.

AS12 Two items are connected (in the same direction) by only one connection. Note that we can introduce multiple connections in the same direction by introducing dummy items (vertices) to the graph.

AS13 Items are separated by a fixed margin (to allow connections, or, for example, crawl space).

AS14 Items must fit into at least one container.

AS15 Every item has depth less than or equal to the smallest container depth.

2.1.1. Provided to the model

The following data is provided to the model:

- item size (width, height, depth)
- container size (width, height, depth)
- container positions (fixed)
- directed connections between items
- connection points on items
- costs of connections and containers
- global margin between items.

2.1.2. Determined by the model

The following are to be determined by the model:

- positions of items
- rotations of items
- number of containers used.

2.2. Extensions

In addition to the basic assumptions, we can replace the assumption that the fixed position of the items z-axis (AS11) and/or the assumption that connection distances are calculated from the centre point of an item (AS8).

EX1 Items are not fixed to the floor, but are restricted to rotation in the xy plane;

EX2 Distance between connected items is calculated from the relative surface points of the items.

3. Models

Nomenclature

Binary Variables

m_k Container k is used (has at least one item)

m_{vk}	Item v in container k
n_{uvk}	Item u and item v share container k
N_{uv}	Item u and item v share the same container
r_{vi}	Item v is rotated by $[0°, 90°, 180°, 270°]$
x_{uv}, y_{uv}, z_{uv}	Set of binary variables used to prevent overlapping of items u and v

Constants

c_{ij}	Unit cost of connection $(i, j) \in E$
$Dx_{ij}, Dy_{ij}, Dz_{ij}$	Rectilinear distance in the x, y, z axes respectively, between connected items $(i, j) \in E$
g_k	Cost of a container $k \in K$
L	Fixed margin between items
M	A sufficiently large number
mx_k, my_k, mz_k	Fixed position of container k for the x, y, z axes respectively
W_k, H_k, D_k	Dimensions of container k for the x, y, z axes, i.e. width, height, depth
w_v, h_v, d_v	Dimensions of item v for the x, y, z axes, i.e. width, height, depth
$xs_{ij}, ys_{ij}, zs_{ij}$	Relative shifts of the connection point on item i for connection $(i, j) \in E$, measured from the centre point of item i for the x, y, z axes, respectively

Continuous Variables

R_{ij}, B_{ij}, F_{ij}	Absolute value of the rectilinear distance between connected items $(i, j) \in E$ for the x, y, z axes respectively
x_v, y_v, z_v	Position of item v for the x, y, z axes

Sets

E	Set of connections between items (directed edges)
K	Set of containers
V	Set of items (vertices)
V'	Set of ordered vertices $V' = \{(u, v) \in V^2 : u < v\}$

3.1. Model 1A

Given basic assumptions (AS11) and (AS15) it is not necessary to consider the z-axis and a model can be formulated as a 2-dimensional floor packing problem.

Consideration must be given to the following to determine a solution to the problem:

- Items must occupy only one rotation.
- Items must be in one and only one container.
- Items within container bounds.
- Items must not overlap.
- Number of containers used.
- Rectilinear distance between connected items.

All other assumptions are assumed to be feasible for the data supplied to the mixed-integer linear programming model.

3.1.1. Items must occupy only one rotation

Let $r_{vi} \in \{0,1\}, i \in \{1,2,3,4\}$ represent the anti-clockwise rotation of item v by $90(i-1)^\circ$, i.e. $r_{v2} = 1$ represents a rotation of 90°. As items only take one rotational position, we have the constraint

$$\sum_{i=1}^{4} r_{vi} = 1, \quad \forall v \in V. \tag{1}$$

3.1.2. Items must be in one and only one container

Let m_{vk} denote whether item v is within container k, then the following constraint is sufficient to ensure that items can occupy only one container

$$\sum_{k \in K} m_{vk} = 1, \quad \forall v \in V. \tag{2}$$

3.1.3. Items within container bounds

As all containers are positioned at the origin (AS9), to ensure that the items remain within the bounds of a container we introduce the inequalities for all $v \in V$ and $k \in K$

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v \leq W_k + M(1 - m_{vk}) \tag{3a}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v \leq H_k + M(1 - m_{vk}). \tag{3b}$$

Coupled with equation (1), inequalities (3a) force an item to be within the bounds of a container based on either its width or height dependent on its current rotation.

3.1.4. Items must not overlap

To model the non-overlapping of items, we first consider the simpler case of one container. To ensure that items u and v do not overlap, it is sufficient that at least one of these conditions hold:

1. Item u is to the left of item v.

2. Item u is to the right of item v.

3. Item u is below item v.

4. Item u is above item v.

It is also only necessary to consider the placement of u relative to v and not vice versa. Define the set of ordered pairs $V' = \{(u, v) \in V^2 : u < v\}$, were the cardinality of the set $|V'| = \frac{|V|\cdot(|V|-1)}{2}$. Let x_{uv}, $y_{uv} \in \{0, 1\}$, then for a sufficiently large M, and for all $(u, v) \in V'$ we have:

$$x_u + (r_{u1} + r_{u3})w_u + (r_{u2} + r_{u4})h_u + L \leq x_v + M(x_{uv} + y_{uv}) \tag{4a}$$

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v + L \leq x_u + M(1 - x_{uv} + y_{uv}) \tag{4b}$$

$$y_u + (r_{u1} + r_{u3})h_u + (r_{u2} + r_{u4})w_u + L \leq y_v + M(1 + x_{uv} - y_{uv}) \tag{4c}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v + L \leq y_u + M(2 - x_{uv} - y_{uv}). \tag{4d}$$

The permutations of variables x_{uv} and y_{uv} ensure that only one of the set of inequalities defined by (4) is active, i.e, not satisfied by sufficiently large M.

To consider multiple containers we introduce a binary variable N_{uv} to represent whether items u and v share a container, i.e. $N_{uv} = 1$ if and only if u and v share the same container. It is also necessary to introduce an auxiliary binary variable n_{uvk} to consider whether items u and v share container k. Then the following inequalities model whether items u and v share a container for all $(u, v) \in V'$, and $k \in K$:

$$n_{uvk} \geq m_{uk} + m_{vk} - 1 \tag{5}$$

$$n_{uvk} \leq m_{uk} \tag{6}$$

$$n_{uvk} \leq m_{vk} \tag{7}$$

$$N_{uv} = \sum_{k \in K} n_{uvk}. \tag{8}$$

If (u, v) are in the same container, inequalities (5 - 7) state that $(n_{uvk} \geq 1) \wedge (n_{uvk} \leq 1) \implies n_{uvk} = 1$. If (u, v) are not in the same container, either inequality (6), inequality (7) or both will result in the inequality, $n_{uvk} \leq 0,\ \forall k \in K \implies n_{uvk} = 0$. Note that the value of N_{uv} is determined by equation (8) which is constrained by equation (2) as item u can only be in a single container and thus items u and v can only share a single container, i.e. $\sum_{k \in K} n_{uvk} \leq 1$. Thus N_{uv} is bounded by $0 \leq N_{uv} \leq 1$.

Inequalities (4) can then be amended, for all $(u, v) \in V'$, to account for overlaps within a given container by logically considering whether items share that container or not:

$$x_u + (r_{u1} + r_{u3})w_u + (r_{u2} + r_{u4})h_u + L \leq x_v + M(x_{uv} + y_{uv}) + M(1 - N_{uv}) \tag{9a}$$

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v + L \leq x_u + M(1 - x_{uv} + y_{uv}) + M(1 - N_{uv}) \tag{9b}$$

$$y_u + (r_{u1} + r_{u3})h_u + (r_{u2} + r_{u4})w_u + L \leq y_v + M(1 + x_{uv} - y_{uv}) + M(1 - N_{uv}) \tag{9c}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v + L \leq y_u + M(2 - x_{uv} - y_{uv}) + M(1 - N_{uv}). \tag{9d}$$

In the case of (u, v) not sharing a container, inequalities (9) are satisfied for sufficiently large M.

3.1.5. Number of containers used

Let $m_k \in \{0, 1\}$ denote whether a container k has at least one item. If the container is empty, $m_k = 0$, else $m_k = 1$. Using a logical OR [41] we have:

$$m_k \geq m_{vk} \qquad \forall v \in V, \forall k \in K \tag{10}$$

$$m_k \leq \sum_{v \in V} m_{vk} \qquad \forall k \in K. \tag{11}$$

Thus by inequality (11), $(m_{vk} = 0,\ \forall v) \implies (m_k \leq 0,\ \forall k) \implies (m_k = 0)$. Also by inequality (10), if any $m_{vk} = 1$, say p, then $(m_{pk} = 1) \implies (m_k \geq 1)$ and by inequality (11), $(m_{pk} = 1) \implies (\sum_{v \in V} m_{vk} \geq 1) \implies m_k \leq 1 \leq \sum_{v \in V} m_{vk}$, therefore $m_k = 1$.

The total number of containers used is therefore

$$\sum_{k \in K} m_k. \tag{12}$$

3.1.6. Rectilinear distance between connected items

The rectilinear distance in the x-axis between a pair of connected items i and j, $(i, j) \in E$, can be calculated by considering the distance between item i and item j and then applying the relative offset of the container position. Define Dx_{ij} by

$$\begin{aligned} Dx_{ij} = &\left[x_i + (r_{i1} + r_{i3})\frac{w_i}{2} + (r_{i2} + r_{i4})\frac{h_i}{2} + \sum_{k \in K} m_{ik} mx_k \right] \\ &- \left[x_j + (r_{j1} + r_{j3})\frac{w_j}{2} + (r_{j2} + r_{j4})\frac{h_j}{2} + \sum_{k \in K} m_{jk} mx_k \right], \quad \forall (i, j) \in E \end{aligned} \tag{13}$$

and note that depending on the positioning of items i and j this can be positive or negative. Therefore we consider the absolute distance $|Dx_{ij}|$.

The absolute distance of Dx_{ij} is given by $|Dx_{ij}| = \max\{Dx_{ij}, -Dx_{ij}\}$ and can be modelled by introducing a minimax variable R_{ij} and the following two inequalities for all $(i, j) \in E$ we have:

$$Dx_{ij} \leq R_{ij} \tag{14a}$$

$$-Dx_{ij} \leq R_{ij}. \tag{14b}$$

Similarly, the rectilinear distance in the y-axis is modelled by

$$\begin{aligned} Dy_{ij} = &\left[y_i + (r_{i1} + r_{i3})\frac{h_i}{2} + (r_{i2} + r_{i4})\frac{w_i}{2} + \sum_{k} m_{ik} my_k \right] \\ &- \left[y_j + (r_{j1} + r_{j3})\frac{h_j}{2} + (r_{j2} + r_{j4})\frac{w_j}{2} + \sum_{k} m_{jk} my_k \right], \quad \forall (i, j) \in E, \end{aligned} \tag{15}$$

and for all $(i, j) \in E$:

$$Dy_{ij} \leq B_{ij} \tag{16a}$$

$$-Dy_{ij} \leq B_{ij}. \tag{16b}$$

3.1.7. Objective function

Given that the unit cost of a connection between i and j is c_{ij} and the cost of a container is given by g_k, minimise the following objective function

$$T = \sum_{(i,j)\in E} c_{ij}\left(R_{ij} + B_{ij}\right) + \sum_{k\in K} g_k m_k. \tag{17}$$

3.2. Model 2A

To model the extension EX1, it is necessary to introduce the z-axis.

3.2.1. Items within container bounds

In addition to inequalities (3a) add the inequality

$$z_v + d_v \le D_k + M(1 - m_{vk}), \quad \forall v \in V, \quad \forall k \in K, \tag{18}$$

which limits the position of an item v in the z-axis to be within the bounds of the container k.

3.2.2. Items must not overlap

Inequalities (9) are amended, for all $(u, v) \in V'$, as below

$$x_u + (r_{u1} + r_{u3})w_u + (r_{u2} + r_{u4})h_u + L \le x_v + M(x_{uv} + y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{19a}$$

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v + L \le x_u + M(1 - x_{uv} + y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{19b}$$

$$y_u + (r_{u1} + r_{u3})h_u + (r_{u2} + r_{u4})w_u + L \le y_v + M(1 + x_{uv} - y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{19c}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v + L \le y_u + M(2 - x_{uv} - y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{19d}$$

$$z_u + d_u + L \le z_v + M(2 - x_{uv} + y_{uv} - z_{uv}) + M(1 - N_{uv}) \tag{19e}$$

$$z_v + d_v + L \le z_u + M(2 + x_{uv} - y_{uv} - z_{uv}) + M(1 - N_{uv}). \tag{19f}$$

The introduction of the additional binary variable z_{uv} results in $2^3 = 8$ possible binary sums, which can be restricted to a choice of 6 through the addition of the inequalities

$$x_{uv} + y_{uv} + z_{uv} \le 2 \tag{20}$$

$$x_{uv} + y_{uv} + M(1 - z_{uv}) \ge 1. \tag{21}$$

Inequalities (20) and (21) ensure that $(x_{uv}, y_{uv}, z_{uv}) \notin \{(1, 1, 1), (0, 0, 1)\}$.

3.2.3. Rectilinear distance between connected items

As rotation is restricted to the xy plane, the rectilinear distance in the z-axis, F_{ij} is given by

$$Dz_{ij} = \left[z_i + \sum_k m_{ik} m z_k\right] - \left[z_j + \sum_k m_{jk} m z_k\right], \quad \forall (i, j) \in E \tag{22}$$

and for all $(i, j) \in E$

$$Dz_{ij} \le F_{ij} \tag{23a}$$

$$-Dz_{ij} \le F_{ij}. \tag{23b}$$

Objective Function

Therefore the updated objective function is

$$T = \sum_{(i,j)\in E} c_{ij}\left(R_{ij} + B_{ij} + F_{ij}\right) + \sum_{k\in K} g_k m_k. \tag{24}$$

3.3. Model 1B and 2B

Model 1B and 2B are equivalent except for the addition of the z-axis, therefore we present only model 2B, all models can be found in full in the appendices.

3.3.1. Relative shift of connection points

For modelling the extension EX2, for any connection $(i,j) \in E$, the relative shifts of the connection point measured from the centre point of item i are given by $(xs_{ij}, ys_{ij}, zs_{ij})$. The relative shift is achieved by amending the equations (13), (15) and (22) written for all $(i,j) \in E$ as

$$Dx'_{ij} = Dx_{ij} + (r_{i1}xs_{ij} - r_{i2}ys_{ij} - r_{i3}xs_{ij} + r_{i4}ys_{ij}) - (r_{j1}xs_{ji} - r_{j2}ys_{ji} - r_{j3}xs_{ji} + r_{j4}ys_{ji}), \tag{25}$$

$$Dy'_{ij} = Dy_{ij} + (r_{i1}ys_{ij} + r_{i2}xs_{ij} - r_{i3}ys_{ij} - r_{i4}xs_{ij}) - (r_{j1}ys_{ji} + r_{j2}xs_{ji} - r_{j3}ys_{ji} - r_{j4}xs_{ji}), \tag{26}$$

$$Dz'_{ij} = Dz_{ij} + zs_{ij} - zs_{ji}, \tag{27}$$

and thus for all $(i,j) \in E$ the inequality pairs (14), (16) and (23) become

$$Dx'_{ij} \le R_{ij}, \qquad \forall (i,j) \in E \tag{28a}$$

$$-Dx'_{ij} \le R_{ij}, \qquad \forall (i,j) \in E \tag{28b}$$

$$Dy'_{ij} \le B_{ij}, \qquad \forall (i,j) \in E \tag{29a}$$

$$-Dy'_{ij} \le B_{ij}, \qquad \forall (i,j) \in E \tag{29b}$$

$$Dz'_{ij} \le F_{ij}, \qquad \forall (i,j) \in E \tag{30a}$$

$$-Dz'_{ij} \le F_{ij}, \qquad \forall (i,j) \in E. \tag{30b}$$

3.4. Size of Models

The size of each of the models is given by table 1. As the number of items $|V|$ in the model increases the number of constraints and variables will be of order $O(|V'|)$, where $|V'| = \frac{|V|\cdot(|V|-1)}{2}$.

Model	No. Variables		No Constraints
	Continuous	Binary	
1A & 1B	$2(\|V\| + \|E\|)$	$4\|V\| + 3\|V'\| + \|K\| + \|V\|\|K\| + \|V'\|\|K\|$	$2\|V\| + 4\|E\| + \|K\| + 5\|V'\| + 3\|V\|\|K\| + 3\|V'\|\|K\|$
2A & 2B	$3(\|V\| + \|E\|)$	$4\|V\| + 4\|V'\| + \|K\| + \|V\|\|K\| + \|V'\|\|K\|$	$2\|V\| + 6\|E\| + \|K\| + 9\|V'\| + 4\|V\|\|K\| + 3\|V'\|\|K\|$

Table 1: Number of constraints and variables for the models.

4. Results and Discussion

Models 1B and 2B were implemented and solved using the commercial solver Gurobi [13] on a Intel(R) Core(TM) i5-7500, 8.00GB RAM PC.[1] Models 1A and 2A are recoverable from models 1B and 2B respectively by setting all relative shifts of the connections points to zero for all items.

First, we present a small example based on a chemical process plant system to showcase the visual solutions and allow a visual inspection for both feasibility and optimality within the domain.

Second, given the combinatorial nature of the problem, we analyse the impact of increasing the number of items, measuring the effect this has on the relative gap (i.e. the relative difference between the best upper and lower bounds). It should be noted that a suboptimal solution in a specific domain requires the visual inspection of a domain expert to determine whether it is "good".

Finally, we present preliminary results for a modular 3D layout of a process plant system which is imported into the plant design software Bentley Plantwise [6]. A detailed analysis would be required to understand the impacts of this design and the additional requirements specific to the task.

4.1. Chemical process plant system example

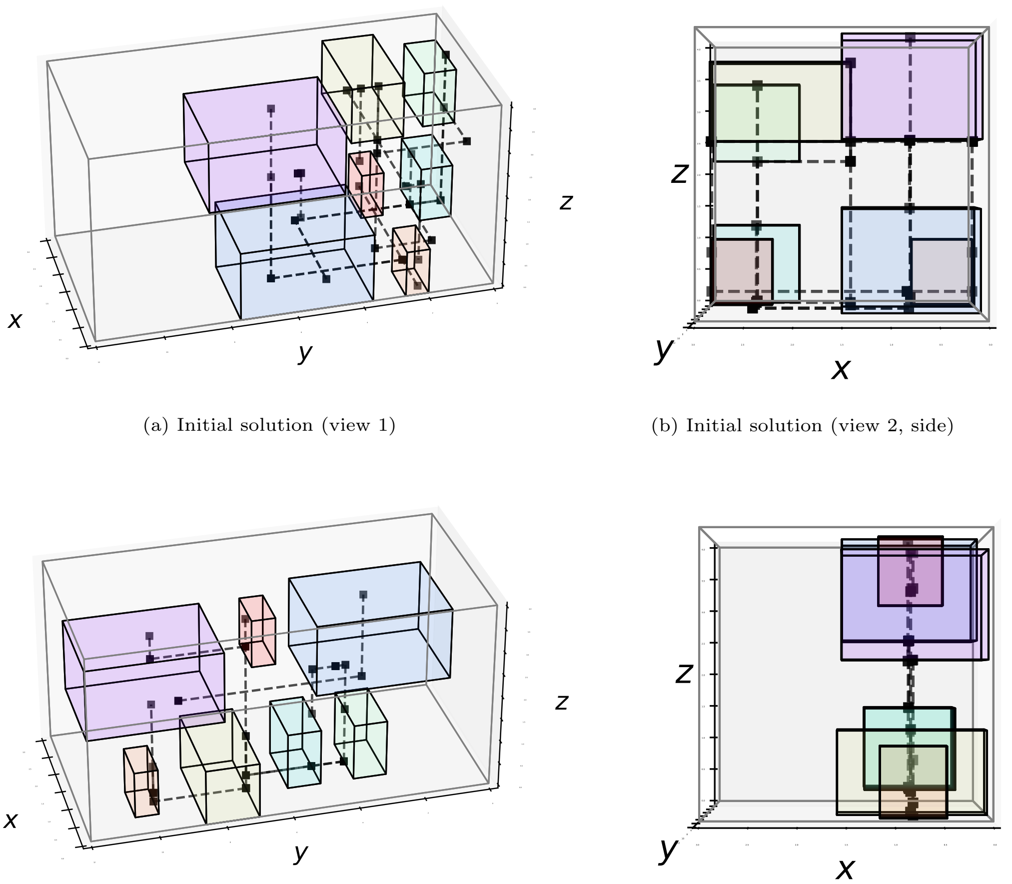
The following example is typical for a chemical process plant system that forms part of a bigger process plant. It comprises of heat exchangers, filters and ion exchange columns. Equipment items are connected by nozzles in which fluid flows. This example consists of a single container of dimensions $(W_k, H_k, D_k) = (3, 12, 4)$ and an objective to minimise the rectilinear distance of the connections between items.

The directed network $G = (V, E)$ is comprised of items, $V = \{1, 2, 3, 4, 5, 6, 7\}$ and connections, $E = \{(1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (4, 5), (5, 6), (5, 7), (6, 1), (7, 1)\}$, which have a uniform cost represented by $c_{ij} = 1$. This particular example is sufficiently small that the necessary information required by the model can be easily displayed in Table 2. This was solved using model 2B which consisted of 355 constraints, 51 continuous variables and 141 integer (binary) variables. The initial feasible solution had an objective value of 158.1 and an optimal solution was obtained in 5.05 seconds with a objective value of 131.4. Figs. 1a, 1b and 1c, 1d provide a scaled 3-dimensional visualisation of the initial feasible solution and the optimal feasible solution, respectively. It is noticeable from Figs. 1c and 1d that the optimal solution is extremely structured with the connections inhabiting the xz plane, reflecting the single goal of minimising the rectilinear distance of the connections.

4.2. Analysis for an increased number of items

It is important to analyse how the problem performs as the number of items increase, to ascertain this a total of 32 networks were randomly generated (4 per value of $n = 4, 6, 8, \ldots, 18$) via the Watts-Strogatz model for random network generation with a probability of rewiring $p = 0.7$, the vertices (items) and edges (connections) were then randomly allocated data within a given range. Each network was then passed to Model 1B and Model 2B and given a time limit of 600 seconds to obtain a solution using Gurobi; results were then averaged over each value of n. Figs. 2a, 2b, 3a and 3b show the trend over time of the incumbent objective value (best known objective value) and relative gap for each value of n for models 1B and 2B, minor discrepancies in the plot where the incumbent objective value or relative gap appears to worsen and are a result of the difficulty averaging the

[1] Source code, data for test cases and results can be found at https://github.com/samtoneill/connected-bin-packing



(a) Initial solution (view 1)

(b) Initial solution (view 2, side)

(c) Optimal solution (view 1)

(d) Optimal solution (view 2, side)

Figure 1: Initial and optimal solutions of model 2B for the small chemical process plant system example.

Item (v)	w_v	h_v	d_v
1	4	1.5	1.5
2	4	1.5	1.5
3	1	1	1.2
4	1	1	1.2
5	1.6	1.6	1.2
6	0.7	0.7	1
7	0.7	0.7	1

(a) Item data

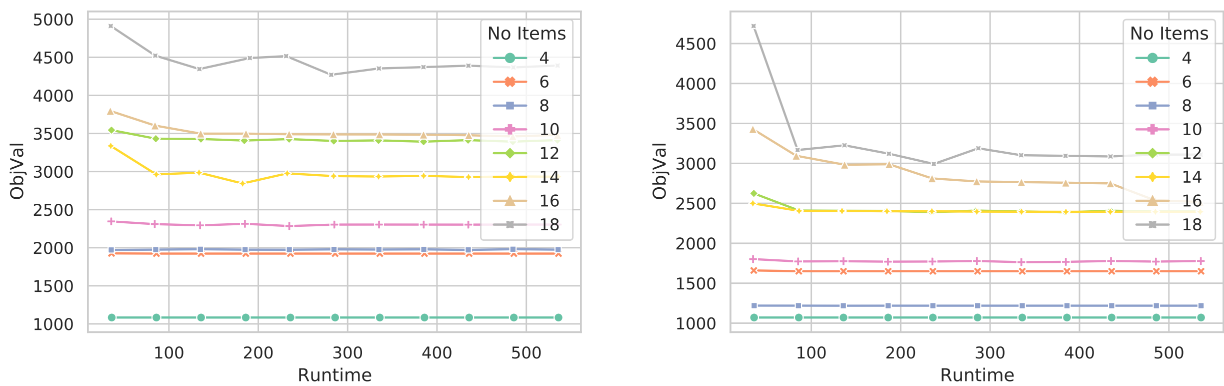
Connection (i,j)	xs_{ij}	ys_{ij}	zs_{ij}	xs_{ji}	ys_{ji}	zs_{ji}
$(1,2)$	1	0	-0.75	0.2	0	0.75
$(2,3)$	1	0	-0.75	0	0.5	0.6
$(2,4)$	1	0	-0.75	0	0.5	0.6
$(3,4)$	0	0.5	-0.6	0	0.5	0.6
$(3,5)$	0	0.5	-0.6	0	0.8	0.6
$(4,5)$	0	0.5	-0.6	0	0.8	0.6
$(5,6)$	0	0.8	-0.6	0	-0.35	0.3
$(5,7)$	0	0.8	-0.6	0	-0.35	0.3
$(6,1)$	0	-0.35	-0.3	0.2	0	-0.75
$(7,1)$	0	-0.35	-0.3	0.2	0	0.75

(b) Connection data

Table 2: Data required for a part of a typical chemical process plant system.

real-time data that was logged. The relative gap is a convergence measure and is the relative distance between the upper and lower bound, given by the formula

$$\text{Relative Gap} = \frac{\text{Best upper bound - Best lower bound}}{\text{Best upper bound}}.$$



(a) Model 1B

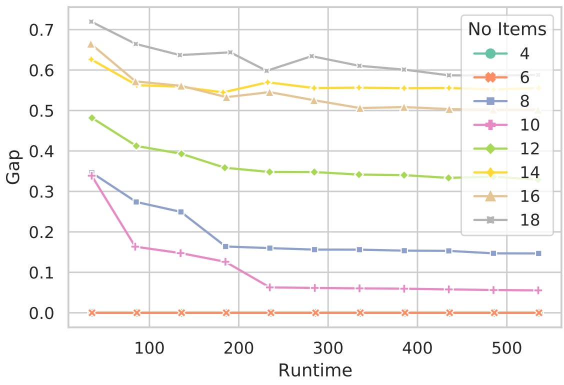
(b) Model 2B

Figure 2: Incumbent objective value of models averaged over values of n.

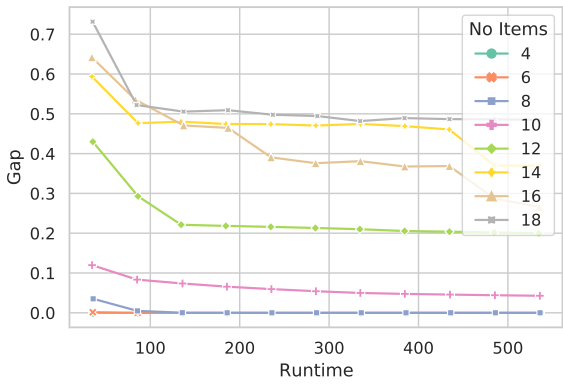
4.3. Preliminary results and CAD output for a modular 3D layout of a process plant system

As aforementioned in the Introduction, productivity in the construction industry has declined in recent decades and the use of modular construction has resulted in reduced schedules and costs.

Fig. 4 displays the results for a process plant system consisting of 29 items to be packed into 8 modules, stacked in a 2x4 configuration and solved to a relative gap tolerance of 60% using model 2B. The solution was then imported into the computer-aided-design (CAD) package, Bentley plant design software, for display purposes.



(a) Model 1B



(b) Model 2B

Figure 3: Relative gap of models averaged over values of n.

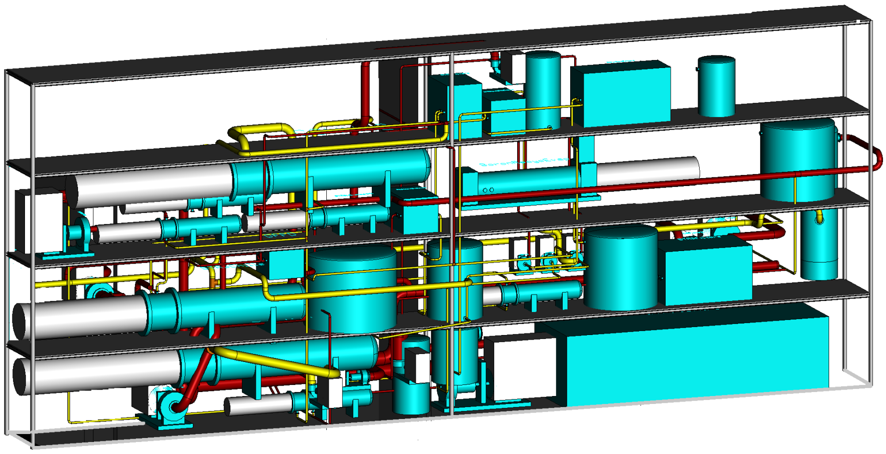


Figure 4: CAD output given by Bentley Plantwise.

4.4. Discussion

Although the relative gap remains high for a large number of items and the prospect of finding an optimal solution becomes increasingly unlikely, it is evident that both the relative gap and objective value settle down within a reasonable amount of time. On visual inspection, the solutions given by Figs. 1 and 4 appear to be sufficiently good for a starting point for a domain expert; however, a number of additional constraints and proper costing would be required for them to resemble reality. This raises two pertinent questions: how long do the models require before the solution is "good enough"?; and what is a suitable measurement to determine if a solution is "good enough"?

The solutions produced by the models exhibit a number of deficiencies due to either the underlying assumptions or shortcomings of the model. We list the following which is by no means exhaustive:

- Items in models 2A and 2B can float.
- Heavier items can be placed on lighter items in the models 2A and 2B.
- The layout of the connections is not given, only the position of the items.
- No thickness of connections considered; currently a fixed margin.

- There are no fixed items or connection points. e.g. fixed gas supply connection. Note that both of these can be modelled using dummy items whose position is restricted by additional bounds on their value.
- There is no rotation in the z-axis.

Each of the above affects the quality of the solution, which could become less accurate for a given domain.

5. Concluding Remarks and Future Work

In this paper we presented four mixed-integer linear programming models that solved the novel problem of modularisation of a set of connected items, demonstrating the solution quality using Gurobi and providing graphical illustrations through the Matplotlib Python library and Bentley Plantwise software.

In Section 4.2 we analysed the behaviour of the models and method by considering the relative gap and objective value of a set of randomised problems. In most cases, the relative gap remained high, yet the objective value appeared to settle and visual inspection confirmed the solution to be of reasonable quality. This raises questions about the usefulness of the relative gap as a measure in determining the quality of a suboptimal solution, i.e. is it "good enough"? For this particular set of models, one can postulate that the observed behaviour of the relative gap and objective value is due to the objective making most gains early on, especially if the container cost is high and of ample dimensions; subsequent gains become increasingly marginal as items become close to aligned. By inspection of the graphical output obtained by the Bentley Plantwise software, we were able to determine that a suboptimal solution may well be "good enough as a good starting point for a domain expert to continue the design process, especially in the early concept phase.

A strength of our approach is that it is a mixed-integer linear programming model and therefore is readily solved by free and commercial software, which continue year-on-year to increase in performance [26]. For smaller problems mathematical programming solvers will return an optimal solution and therefore these results can be used for tuning and benchmarking other methods. For the models to be of practical use, a degree of domain-specific tailoring would be required, although they provide a strong foundation for such work to be undertaken.

Future work will involve the creation of a holistic process which will allow a domain expert to input relevant features of their problem and to tailor the optimisation model to provide a solution which can then be imported into the domain specific software as a starting point for more detailed design. This requires automation of the model building process using data specified by a domain expert and providing integration between the optimisation software and the computer-aided design software.

To further assess the proposed approach, it is important to consider the computational efficiency of the LP solvers and whether the quality and speed of the solution can be improved through parallel and distributed optimisation and to determine at what point the proposed methods fail to provide a reasonable solution. Additional methods for solving the models should be implemented, such as evolutionary algorithms, to compare with the results in this paper.

References

[1] D.T.S. Alves, J.L. de Medeiros, O.d.Q.F. Araújo, Optimal determination of chemical plant layout via minimization of risk to general public using Monte Carlo and Simulated Annealing techniques, Journal of Loss Prevention in the Process Industries 41 (2016) 202–214.

[2] L. Amorese, V. Cena, C. Mustacchi, A heuristic for the compact location of process components, Chemical Engineering Science 32 (1977) 119–124.

[3] J. Balakrishnan, C.H. Cheng, D.G. Conway, C.M. Lau, A hybrid genetic algorithm for the dynamic plant layout problem, International Journal of Production Economics 86 (2003) 107–120.

[4] A.P. Barbosa-Póvoa, S. Macchietto, Detailed design of multipurpose batch plants, Computers and Chemical Engineering 18 (1994) 1013–1042.

[5] K. Barry, Modularization of Equipment for New Nuclear Applications – Benchmarking (1019213), Technical Report, 2009.

[6] I. Bentley Systems, Bentley Plantwise, 2018.

[7] T. Bock, The future of construction automation: Technological disruption and the upcoming ubiquity of robotics, Automation in Construction 59 (2015) 113–121.

[8] M. Furuholmen, K. Glette, M. Hovin, J. Torresen, A coevolutionary, hyper heuristic approach to the optimization of three-dimensional process plant layouts - A comparative study, in: 2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010.

[9] M.C. Georgiadis, S. Macchietto, Layout of process plants: A novel approach, Computers and Chemical Engineering 21 (1997) S337–S342.

[10] M.C. Georgiadis, G. Schilling, G.E. Rotstein, S. MacChietto, A general mathematical programming approach for process plant layout, Computers and Chemical Engineering 23 (1999) 823–840.

[11] R. Guirardello, R.E. Swaney, Optimization of process plant layout with pipe routing, Computers and Chemical Engineering 30 (2005) 99–114.

[12] D.J. Gunn, H.D. Al-Asadi, Computer-aided layout of chemical plant: A computational method and case study, Computer-Aided Design 19 (1987) 131–140.

[13] Gurobi Optimization, Gurobi Optimizer Reference Manual, 2020.

[14] Q. He, X. Ji, L. Zhou, Y. Dang, A hybridizing meta-heuristic approach towards optimal multi-floor process plant layout, in: Computer Aided Chemical Engineering, volume 47, Elsevier B.V., 2019, pp. 181–186.

[15] M.R. Hosseini, I. Martek, E.K. Zavadskas, A.A. Aibinu, M. Arashpour, N. Chileshe, Critical evaluation of off-site construction research: A Scientometric analysis, 2018.

[16] P.E. Jaquith, R.M. Burns, S.E. Dunbarr, B.J. Fontaine, H.A. Nelson, J.L. Silveira, T.D. Thompson, Modular engine room design and construction for the strategic sealift ships, Journal of Ship Production 12 (1996) 230–243.

[17] S. Jayakumar, G.V. Reklaitis, Chemical plant layout via graph partitioning-1. Single level, Computers and Chemical Engineering 18 (1994) 441–458.

[18] R. Jin, S. Gao, A. Cheshmehzangi, E. Aboagye-Nimo, A holistic review of off-site construction literature published between 2008 and 2018, Journal of Cleaner Production 202 (2018) 1202–1219.

[19] S.K. Jung, M.I. Roh, K.S. Kim, Arrangement method of a naval surface ship considering stability, operability, and survivability, Ocean Engineering 152 (2018) 316–333.

[20] R. Kern, How to manage plant design to obtain minimum cost, Chem. Eng 84 (1977) 130–136.

[21] C.W. Lapp, M.W. Golay, Modular design and construction techniques for nuclear power plants, Nuclear Engineering and Design 172 (1997) 327–349.

[22] S.E. Latifi, E. Mohammadi, N. Khakzad, Process plant layout optimization with uncertainty and considering risk, Computers and Chemical Engineering 106 (2017) 224–242.

[23] C.J. Lee, Optimal multi-floor plant layout based on the mathematical programming and particle swarm optimization, Industrial Health 53 (2015) 491–497.

[24] C.K. Lee, A review of applications of genetic algorithms in operations management, Engineering Applications of Artificial Intelligence 76 (2018) 1–12.

[25] J. Martinez-Gomez, F. Nápoles-Rivera, J.M. Ponce-Ortega, M. Serna-González, M.M. El-Halwagi, Optimization of facility location and reallocation in an industrial plant through a multi-annual framework accounting for economic and safety issues, Journal of Loss Prevention in the Process Industries 33 (2015) 129–139.

[26] H. Mittelmann, Decision Tree for Optimization Software, 2021.

[27] S. Moran, Process Plant Layout, Butterworth-Heinemann, 2nd edition, 2016.

[28] L.G. Papageorgiou, G.E. Rotstein, Continuous-domain mathematical models for optimal process plant layout, Industrial and Engineering Chemistry Research 37 (1998) 3631–3639.

[29] P.J. Park, C.J. Lee, The research of optimal plant layout optimization based on particle swarm optimization for ethylene oxide plant, Journal of the Korean Society of Safety 30 (2015) 32–37.

[30] D.I. Patsiatzis, L.G. Papageorgiou, Optimal multi-floor process plant layout, Computers and Chemical Engineering 26 (2002) 575–583.

[31] D.I. Patsiatzis, L.G. Papageorgiou, Safe process plant layout using mathematical programming, in: Computer Aided Chemical Engineering, volume 10, Elsevier, 2002, pp. 295–300.

[32] D.I. Patsiatzis, L.G. Papageorgiou, Efficient solution approaches for the multifloor process plant layout problem, Industrial and Engineering Chemistry Research 42 (2003) 811–824.

[33] F.D. Penteado, A.R. Ciric, An MINLP Approach for Safe Process Plant Layout, Industrial and Engineering Chemistry Research 35 (1996) 1354–1361.

[34] M.J. Rosenblatt, The facilities layout problem: A multi-goal approach, International Journal of Production Research 17 (1979) 323–332.

[35] T. Seifert, S. Sievers, C. Bramsiepe, G. Schembecker, Small scale, modular and continuous: A new approach in plant design, Chemical Engineering and Processing: Process Intensification 52 (2012) 140–150.

[36] C. Smith, G. Bowtell, The First Modularisation of an LNG Plant: North West Shelf Venture Karratha Gas Plant Phase V Expansion Project, 2007.

[37] B. Sutharshan, M. Mutyala, R.P. Vijuk, A. Mishra, The AP1000™ reactor: Passive safety and modular design, in: Energy Procedia, volume 7, Elsevier Ltd, 2011, pp. 293–302.

[38] A. Suzuki, T. Fuchino, M. Muraki, T. Hayakawa, An evolutionary method of arranging the plot plan for process plant layout., Journal of Chemical Engineering of Japan 24 (1991) 226–231.

[39] R. Wang, Y. Wu, Y. Wang, X. Feng, An industrial area layout design methodology considering piping and safety using genetic algorithm, Journal of Cleaner Production 167 (2017) 23–31.

[40] R. Wang, H. Zhao, Y. Wu, Y. Wang, X. Feng, M. Liu, An industrial facility layout design method considering energy saving based on surplus rectangle fill algorithm, Energy 158 (2018) 1038–1051.

[41] H.P. Williams, Modelling in logic for integer programming, in: International Series in Operations Research and Management Science, volume 130, Springer New York LLC, 2009, pp. 71–103.

[42] Y. Wu, R. Wang, Y. Wang, X. Feng, An area-wide layout design method considering piecewise steam piping, Chemical Engineering Transactions 61 (2017) 457–462.

[43] Y. Wu, Y. Wang, A chemical industry area-wide layout design methodology for piping implementation, Chemical Engineering Research and Design 118 (2017) 81–93.

[44] Y. Xu, Z. Wang, Q. Zhu, An improved hybrid genetic algorithm for chemical plant layout optimization with novel non-overlapping and toxic gas dispersion constraints, Chinese Journal of Chemical Engineering 21 (2013) 412–419.

[45] J.S. Yang, C.J. Lee, The research of layout optimization for LNG liquefaction plant to save the capital expenditures, Korean Chemical Engineering Research 57 (2019) 51–57.

Appendix A. Models

The appendix presents the full versions of models 1A, 1B, 2A and 2B.

*Appendix A.1. **Model 1A***

Appendix A.1.1. Rotational Constraints

$$\sum_{i=1}^{4} r_{vi} = 1, \quad \forall v \in V. \tag{A.1}$$

Appendix A.1.2. Container Constraints

$$\sum_{k \in K} m_{vk} = 1, \quad \forall v \in V. \tag{A.2}$$

For all $(u, v) \in V'$ and $k \in K$:

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v \leq W_k + M(1 - m_{vk}) \tag{A.3}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v \leq H_k + M(1 - m_{vk}). \tag{A.4}$$

Appendix A.1.3. Non-overlapping Constraints

For all $(u, v) \in V'$ and $k \in K$:

$$n_{uvk} \geq m_{uk} + m_{vk} - 1 \tag{A.5}$$

$$n_{uvk} \leq m_{uk} \tag{A.6}$$

$$n_{uvk} \leq m_{vk} \tag{A.7}$$

$$N_{uv} = \sum_{k \in K} n_{uvk}, \quad \forall (u, v) \in V'. \tag{A.8}$$

For all $(u, v) \in V'$:

$$x_u + (r_{u1} + r_{u3})w_u + (r_{u2} + r_{u4})h_u + L \leq x_v + M(x_{uv} + y_{uv}) + M(1 - N_{uv}) \tag{A.9}$$

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v + L \leq x_u + M(1 - x_{uv} + y_{uv}) + M(1 - N_{uv}) \tag{A.10}$$

$$y_u + (r_{u1} + r_{u3})h_u + (r_{u2} + r_{u4})w_u + L \leq y_v + M(1 + x_{uv} - y_{uv}) + M(1 - N_{uv}) \tag{A.11}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v + L \leq y_u + M(2 - x_{uv} - y_{uv}) + M(1 - N_{uv}). \tag{A.12}$$

Appendix A.1.4. No of Containers Used Constraints

$$m_k \geq m_{vk}, \qquad \forall v \in V, \forall k \in K \tag{A.13}$$

$$m_k \leq \sum_{v \in V} m_{vk}, \qquad \forall k \in K. \tag{A.14}$$

Appendix A.1.5. Rectilinear Distance Constraints

For all $(i,j) \in E$:

$$Dx_{ij} = \left[x_i + (r_{i1} + r_{i3})\frac{w_i}{2} + (r_{i2} + r_{i4})\frac{h_i}{2} + \sum_{k \in K} m_{ik} m x_k\right] - \left[x_j + (r_{j1} + r_{j3})\frac{w_j}{2} + (r_{j2} + r_{j4})\frac{h_j}{2} + \sum_{k \in K} m_{jk} m x_k\right]$$

$$Dx_{ij} \leq R_{ij} \tag{A.15}$$

$$-Dx_{ij} \leq R_{ij} \tag{A.16}$$

$$Dy_{ij} = \left[y_i + (r_{i1} + r_{i3})\frac{h_i}{2} + (r_{i2} + r_{i4})\frac{w_i}{2} + \sum_{k} m_{ik} m y_k\right] - \left[y_j + (r_{j1} + r_{j3})\frac{h_j}{2} + (r_{j2} + r_{j4})\frac{w_j}{2} + \sum_{k} m_{jk} m y_k\right]$$

$$Dy_{ij} \leq B_{ij} \tag{A.17}$$

$$-Dy_{ij} \leq B_{ij}. \tag{A.18}$$

Appendix A.1.6. Objective Function

$$T = \sum_{(i,j) \in E} c_{ij}\left(R_{ij} + B_{ij}\right) + \sum_{k \in K} g_k m_k. \tag{A.19}$$

Appendix A.2. ***Model 1B***

Appendix A.2.1. Relative Shift of Connection Points

For all $(i,j) \in E$, replace inequalities (A.15 - A.18) in Model 1A with the following:

$$Dx_{ij} + (r_{i1}xs_{ij} - r_{i2}ys_{ij} - r_{i3}xs_{ij} + r_{i4}ys_{ij}) - (r_{j1}xs_{ji} - r_{j2}ys_{ji} - r_{j3}xs_{ji} + r_{j4}ys_{ji}) \leq R_{ij} \tag{A.20}$$

$$-\left[Dx_{ij} + (r_{i1}xs_{ij} - r_{i2}ys_{ij} - r_{i3}xs_{ij} + r_{i4}ys_{ij}) - (r_{j1}xs_{ji} - r_{j2}ys_{ji} - r_{j3}xs_{ji} + r_{j4}ys_{ji})\right] \leq R_{ij} \tag{A.21}$$

$$Dy_{ij} + (r_{i1}ys_{ij} + r_{i2}xs_{ij} - r_{i3}ys_{ij} - r_{i4}xs_{ij}) - (r_{j1}ys_{ji} + r_{j2}xs_{ji} - r_{j3}ys_{ji} - r_{j4}xs_{ji}) \leq B_{ij} \tag{A.22}$$

$$-\left[Dy_{ij} + (r_{i1}ys_{ij} + r_{i2}xs_{ij} - r_{i3}ys_{ij} - r_{i4}xs_{ij}) - (r_{j1}ys_{ji} + r_{j2}xs_{ji} - r_{j3}ys_{ji} - r_{j4}xs_{ji})\right] \leq B_{ij}. \tag{A.23}$$

Appendix A.3. ***Model 2A***

Appendix A.3.1. Container Constraints

For all $v \in V$ and $k \in K$, replace inequalities (A.3 - A.4) in Model 1A with the following:

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v \leq W_k + M(1 - m_{vk}) \tag{A.24}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v \leq H_k + M(1 - m_{vk}) \tag{A.25}$$

$$z_v + d_v \leq D_k + M(1 - m_{vk}). \tag{A.26}$$

Appendix A.3.2. Non-overlapping Constraints

For all $(u, v) \in V'$, replace inequalities (A.9 - A.12) in Model 1A with the following:

$$x_u + (r_{u1} + r_{u3})w_u + (r_{u2} + r_{u4})h_u + L \leq x_v + M(x_{uv} + y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{A.27}$$

$$x_v + (r_{v1} + r_{v3})w_v + (r_{v2} + r_{v4})h_v + L \leq x_u + M(1 - x_{uv} + y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{A.28}$$

$$y_u + (r_{u1} + r_{u3})h_u + (r_{u2} + r_{u4})w_u + L \leq y_v + M(1 + x_{uv} - y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{A.29}$$

$$y_v + (r_{v1} + r_{v3})h_v + (r_{v2} + r_{v4})w_v + L \leq y_u + M(2 - x_{uv} - y_{uv} + z_{uv}) + M(1 - N_{uv}) \tag{A.30}$$

$$z_u + d_u + L \leq z_v + M(2 - x_{uv} + y_{uv} - z_{uv}) + M(1 - N_{uv}) \tag{A.31}$$

$$z_v + d_v + L \leq z_u + M(2 + x_{uv} - y_{uv} - z_{uv}) + M(1 - N_{uv}) \tag{A.32}$$

$$x_{uv} + y_{uv} + z_{uv} \leq 2 \tag{A.33}$$

$$x_{uv} + y_{uv} + M(1 - z_{uv}) \geq 1. \tag{A.34}$$

Appendix A.3.3. Rectilinear Distance Constraints

For all $(i, j) \in E$, replace inequalities (A.15 - A.18) in Model 1A with the following:

$$Dx_{ij} = \left[x_i + (r_{i1} + r_{i3})\frac{w_i}{2} + (r_{i2} + r_{i4})\frac{h_i}{2} + \sum_{k \in K} m_{ik} m x_k\right] - \left[x_j + (r_{j1} + r_{j3})\frac{w_j}{2} + (r_{j2} + r_{j4})\frac{h_j}{2} + \sum_{k \in K} m_{jk} m x_k\right]$$

$$Dx_{ij} \leq R_{ij} \tag{A.35}$$

$$-Dx_{ij} \leq R_{ij} \tag{A.36}$$

$$Dy_{ij} = \left[y_i + (r_{i1} + r_{i3})\frac{h_i}{2} + (r_{i2} + r_{i4})\frac{w_i}{2} + \sum_{k} m_{ik} m y_k\right] - \left[y_j + (r_{j1} + r_{j3})\frac{h_j}{2} + (r_{j2} + r_{j4})\frac{w_j}{2} + \sum_{k} m_{jk} m y_k\right]$$

$$Dy_{ij} \leq B_{ij} \tag{A.37}$$

$$-Dy_{ij} \leq B_{ij} \tag{A.38}$$

$$Dz_{ij} = \left[z_i + \sum_{k} m_{ik} m z_k\right] - \left[z_j + \sum_{k} m_{jk} m z_k\right]$$

$$Dz_{ij} \leq F_{ij} \tag{A.39}$$

$$-Dz_{ij} \leq F_{ij}. \tag{A.40}$$

Appendix A.3.4. Objective Function

$$T = \sum_{(i,j) \in E} c_{ij}\left(R_{ij} + B_{ij} + F_{ij}\right) + \sum_{k \in K} g_k m_k. \tag{A.41}$$

Appendix A.4. ***Model 2B***

Appendix A.4.1. Relative Shift of Connection Points

For all $(i, j) \in E$, replace inequalities (A.35 - A.40) in Model 1A with the following:

$$Dx_{ij} + (r_{i1}xs_{ij} - r_{i2}ys_{ij} - r_{i3}xs_{ij} + r_{i4}ys_{ij}) - (r_{j1}xs_{ji} - r_{j2}ys_{ji} - r_{j3}xs_{ji} + r_{j4}ys_{ji}) \leq R_{ij} \quad (A.42)$$

$$-[Dx_{ij} + (r_{i1}xs_{ij} - r_{i2}ys_{ij} - r_{i3}xs_{ij} + r_{i4}ys_{ij}) - (r_{j1}xs_{ji} - r_{j2}ys_{ji} - r_{j3}xs_{ji} + r_{j4}ys_{ji})] \leq R_{ij} \quad (A.43)$$

$$Dy_{ij} + (r_{i1}ys_{ij} + r_{i2}xs_{ij} - r_{i3}ys_{ij} - r_{i4}xs_{ij}) - (r_{j1}ys_{ji} + r_{j2}xs_{ji} - r_{j3}ys_{ji} - r_{j4}xs_{ji}) \leq B_{ij} \quad (A.44)$$

$$-[Dy_{ij} + (r_{i1}ys_{ij} + r_{i2}xs_{ij} - r_{i3}ys_{ij} - r_{i4}xs_{ij}) - (r_{j1}ys_{ji} + r_{j2}xs_{ji} - r_{j3}ys_{ji} - r_{j4}xs_{ji})] \leq B_{ij} \quad (A.45)$$

$$Dz_{ij} + zs_{ij} - zs_{ji} \leq F_{ij} \quad (A.46)$$

$$-[Dz_{ij} + zs_{ij} - zs_{ji}] \leq F_{ij}. \quad (A.47)$$