An Optimization Model for 3D Pipe Routing with Flexibility Constraints

Gleb Belov\textsuperscript{1(E)}, Tobias Czauderna\textsuperscript{1}, Amel Dzaferovic\textsuperscript{2}, Maria Garcia de la Banda\textsuperscript{1}, Michael Wybrow\textsuperscript{1}, and Mark Wallace\textsuperscript{1}

\textsuperscript{1} Faculty of Information Technology, Monash University, Melbourne, Australia \{gleb.belov,tobias.czauderna,maria.garcia.delabanda,michael.wybrow, mark.wallace\}@monash.edu

\textsuperscript{2} Woodside Energy Ltd., Perth, Australia amel.dzaferovic@woodside.com.au

Abstract. Optimizing the layout of the equipment and connecting pipes that form a chemical plant is an important problem, where the aim is to minimize the total cost of the plant while ensuring its safety and correct operation. The complexity of this problem is such that it is still solved manually, taking multiple engineers several years to complete. Most research in this area focuses on the simpler subproblem of placing the equipment, while the approaches that take pipe routing into account are either based on heuristics or do not consider sufficiently realistic scenarios. Our work presents a new model of the pipe routing subproblem that integrates realistic requirements, such as flexibility constraints, and aims for optimality while solving the largest problem instance considered in the literature. The model is being developed in collaboration with Woodside Energy Ltd. for their Liquefied Natural Gas plants, and is implemented in the high-level modeling language MiniZinc. The use of MiniZinc has both reduced the amount of time required to develop the model, and allowed us to easily experiment with different solvers.

1 Introduction

A chemical process plant produces chemicals through the transformation or separation of materials. This is achieved as the materials pass through the different equipment in the plant via the required connecting pipes. Perhaps surprisingly, the cost of the pipes and associated support structures takes the largest share of the material cost for constructing such plants. This paper focuses on determining the 3D layout of the pipes [5] required to connect the equipment in a chemical plant. The aim is to obtain a layout that minimizes the cost of the pipes and their support structures, while satisfying the constraints needed to ensure the safety and proper functioning of the plant. The pipe routing problem, as we will refer to this application, occurs in many different industries, from water desalination to natural gas production. It is part of the more general process plant layout problem [9], which is in turn a special case of the spatial packaging problem [15].

While there has been some research in optimization methods to solve these problems (e.g., [9,12,15,19–21]), the approaches that consider more realistic pipe
Routing scenarios are based on local/heuristic search methods. Realistic scenarios include not only a reasonably large amount of pipes, but also take into account features like pipe flexibility constraints (which deal with the ability for each pipe to cope with the stress inflicted by thermal expansion), and pipe supporting structures (such as pipe racks and other equipment). These features are crucial for the correct functioning and costing of the plant. Currently, producing layouts for such realistic scenarios is a manual task and may take multiple engineers several years to complete.

This paper presents a new model for the 3D pipe routing problem that aims for optimality and explicitly incorporates flexibility constraints and supporting structures, while solving the largest problem instance considered in the literature. The model, which has been developed in collaboration with engineers from Woodside Energy Ltd. for their Liquefied Natural Gas (LNG) plants, is implemented in MiniZinc \cite{13}, a high-level, solver-independent Constraint Programming (CP) modeling language for combinatorial optimization and satisfiability problems. This has allowed us to produce a high-level but quite realistic model relatively quickly, and then explore different solving approaches to this model: from constraint propagation solvers with a variety of search strategies, to mixed-integer linear programming (MIP) solvers with different translations for non-linear constraints.

Our aim is to build a tool that will allow plant designers and piping engineers at Woodside to explore and evaluate alternative layouts for realistic pipe routing scenarios and, thus, support them in designing a better plant in a shorter amount of time. Note that we are not attempting to algorithmically compute the final plant layout without human intervention, as we do not believe this is yet feasible. This is both due to the huge complexity of modern plants and to the amount of “undocumented” requirements that seem to exist mainly in the head of the plant and pipe layout engineers. Therefore, while it is important for our system to find a high quality layout solution to a realistic pipe routing scenario, it is also important to display the solution via a visual interface that allows engineers to interrogate the proposed solution, as well as guide the optimization process by requesting changes to various parameters and constraints. We have already performed the first steps towards such an interactive optimization system, in the form of a 3D visualization tool that is connected to our modeling system. This visualization tool enables engineers to explore the produced layout, and to evaluate and validate the proposed solution in a familiar way.

The rest of the paper is structured as follows. Section 2 discusses the related literature and past work on the general plant design problem. Section 3 describes the problem in greater detail and provides the decision variables, constraints and search strategies used in our MiniZinc model. Section 4 briefly describes how the non-linear constraints in the model are appropriately translated by MiniZinc for MIP and CP solvers. Section 5 presents a series of experiments aimed at evaluating the scalability and accuracy of our model. Section 6 describes the interactive 3D visualization tool we have developed for communicating and exploring the
results provided by the model. Finally, Sect. 7 presents our conclusions and future work.

# Literature Review

Solving the overall plant design problem requires finding 3D location coordinates for all the equipment and connecting pipes within a plant’s volume (referred to as the *container cuboid*), in such a way as to minimize the total cost of the plant while, at the same time, ensuring its safety and correct functionality. For small problem instances, Sakti et al. [15] successfully apply an integrated approach for a satisfaction version of the simultaneous equipment and piping layout design problem, where the aim is to find any feasible solution that places the equipment and connects the pipes within the given container cuboid. In particular, they considered 10 equipment pieces and up to 15 pipes with 4 segments on average.

For larger, more realistic problem instances, and those where the goal is to find an optimal (or high quality) solution, their integrated approach does not scale. In these cases the problem is naturally divided into two phases [9]. The first phase aims at positioning the equipment, that is, obtaining the 3D location coordinates for each piece of equipment, while minimizing an approximate total cost of the plant. In this phase the focus is on the equipment, ensuring it is supported, safely positioned and can correctly function, while the cost of the pipes is approximated using rough measures, such as Manhattan distances. The second phase aims at determining an optimal layout of the pipes connecting the already positioned equipment. The focus this time is on the pipe routing, taking into consideration issues such as pipe stress and flexibility, the need to support the pipes and the cost associated to these supports. Theoretically, this separation into phases can lead to infeasibilities in the later phases, which can be made less probable using ample safety distances between equipment.

There has already been some research devoted to this problem. However, most of it (e.g., [19,20]) focuses only on the efficient modeling and solving of the first phase (equipment location), which is considerably simpler than the second. While there has been research that includes the second phase (e.g., [9,15]) or even focuses on it (e.g., [12,21]), the existing approaches do not satisfy Woodside’s requirements.

On the one hand, the more realistic approaches, which take into account the simultaneous optimized routing of several pipes (including branching pipes and support placement), are based on heuristic algorithms (rather than complete search methods), such as the ant-colony evolutionary algorithms used by [7,12]. This is not our focus, as Woodside is more interested in pursuing complete approaches.

On the other hand, it is difficult to extend the approaches that rely on complete search methods to take into account some of the required constraints, particularly flexibility constraints. In its basic version, single-pipe routing can be modeled as a 3D rectilinear shortest path problem solvable by Dijkstra’s algorithm. One of the most realistic of the complete approaches is that of Guirardello
and Swaney [9], which provides a detailed MIP model for solving phase one and a general overview of a network-flow MIP model for solving phase two. This second MIP model relies on the construction of a reduced connection graph that limits the possible routes of the pipes. This is used to route pipes one by one, since they suggest that simultaneous routing of the pipes is too costly for a MIP model. While they do not give enough details regarding how the connection graph is constructed, an approach to construct such a connection graph is given, e.g., by de Berg et al. [6], who present a higher-dimensional rectilinear shortest path model that considers bend costs. A more hierarchical method using cuboid free space decomposition is given by Zhu and Latombe [21] and applied to pipe routing. Unfortunately, even if these methods are used, it is not clear how [9] performs sequential pipe routing when pipes interfere with each other ([9] talks about “some tuning by hand” which might be required for these cases). Further, none of these methods can be easily extended to take pipe flexibility constraints into account. In fact, we are not aware of any published results on a general pipe routing method that incorporates flexibility constraints. While [9] mentions the use of Guided Cantilever flexibility constraints in an iterative barrier method to eliminate over-stressed pipe solutions, extending their method to achieve this is not straightforward, and their work provides no information on how to do so.

3 Problem Description and Associated Model

As mentioned before, this paper focuses on the second phase of the process plant layout problem, where the equipment has already been positioned safely and correctly within the container cuboid, and the aim now is to determine the best routing for the pipes that connect the equipment (see the rightmost picture in Fig. 2 for a final solution to our full benchmark). While we have also implemented a MiniZinc model for solving phase one that provides the equipment locations, due to space considerations we will not provide details regarding this model.

As done in the literature, we limit our pipe routing approach to rectilinear axis-parallel routing; in particular, we constrain all bends to be 90°. This is acceptable as non-90° bends are extremely rare in real plants, and can be added later by a post-process step that looks for possible pipe simplifications based on the addition of such bends and non-axis-parallel segments. Further, we approximate pipe segments by cuboids with a square cross-section and ignore bend radius at this stage. This simplification is also acceptable since the resulting loss of space in a large process plant (versus, say, the pipe routing in a jet engine as performed in [15]) is negligible. In this form, the resulting model extends the one in [15] to account for supports and flexibility constraints, and is related to 3D orthogonal packing [14,17].

3.1 Input and Derived Data

The input data to the second phase includes the following:
- Dimensions of the container cuboid (integer values in millimeters) and an associated discretization parameter $K \geq 1$ for placement positions, where $K$ is the size of the model’s length unit in mm. The smaller the value of $K$, the higher the number of possible position points for placing pipes and, thus, the more difficult the problem to solve.

- Set $\mathcal{M}$ of equipment and their locations as provided by phase one, which models each equipment piece $m \in \mathcal{M}$ as a cuboid with given length, width and height $W^m \in \mathbb{Z}^3$ (depending on the chosen rotation), and returns its location as the 3D coordinates $X^m \in \mathbb{Z}^3$ of the corner with the smallest $x$, $y$ and $z$ values. The elements of $\mathcal{M}$ can be of several types, including vessels, heat exchangers, pumps, the pipe rack (a multi-platform structure that traverses the entire plant), and the source/sink points that connect the main pipe to parts of the plant not considered by the problem instance. Note that, in this second phase, the elements of $\mathcal{M}$ represent either support structures or obstacles to which some pipes are attached via “nozzles”. Only some of these elements (mostly vessels and racks) can provide support to nearby pipes.

- Two pieces of information for each nozzle: its location modeled as the 3D coordinates of the center of the attachment area (i.e., the intersection of the equipment surface and the pipe axis), and its direction, modeled as a value from the set $\{0, \ldots, 5\}$, where values $\{0, 1, 2\}$ indicate the nozzle has a positive direction along the $x$, $y$, $z$-axis, respectively, and values $\{3, 4, 5\}$ indicate a negative direction along the same axes.

- Set $\mathcal{P}$ of pipes and, for each pipe $p \in \mathcal{P}$, the following information: external diameter $D_p$ (corresponding to the length of any side of the square cross-section), cost per length unit $C_p^L$, cost per bend $C_p^B$, maximal number of pipe segments allowed $N^S_p$, safety distance to equipment and other pipes $s^M_p$, thermal expansion unit $e_p$, elasticity modulus $E_p$ (which is associated to $p$’s material), and maximum stress allowed (stress capacity) $S^A_p$. We also input lower/upper pipe segment length bounds, where segment length denotes the distance between the segment’s defining nodes, and a node is either a bend (assuming bend radius zero) or the pipe nozzle attachment point.

- Set $\mathcal{SZ}$ of cuboid support zones (i.e., areas that can support pipes), and for each $j \in \mathcal{SZ}$ the following information: its location as the smallest corner $X^j \in \mathbb{Z}^3$ and a length, width, and height vector $W^j \in \mathbb{Z}^3$ as above, together with the cost penalty $C^{SZ}_{jp}$ for any pipe $p$ that uses $j$. Note that some of these support zones are associated to equipment (mostly vessels), while some are associated to each of the platforms in the pipe rack. We currently have only two cost classes for support zones: $C^{SZ}_{jp} \in \{0, C^SZ_p\}$ with the zero value being the one given to the rack’s platform support levels.

From the above input data, our model derives the following parameters for each pipe $p \in \mathcal{P}$:

- Set $\mathcal{S}_p$ of pipe segment indices $\mathcal{S}_p = \{1, \ldots, N^S_p\}$. Note that if the number $n$ of segments of $p$ is less than the maximum allowed ($n < N^S_p$), the length of all segments $j : n < j \leq N^S_p$ will be 0.
3.2 Decision Variables

The solution to the problem is expressed in terms of the values of the decision variables representing for each pipe \( p \in \mathcal{P} \), the pipe node positions \( X^p_i \in \mathbb{Z}^3 \) of each \( i \in \mathcal{N}_p \), subject to the container cuboid. The objective function (and associated decision variable \( \text{obj} \)) is the sum of the cost penalties associated to the length, bends and supports of each pipe.

In addition to the decision variables associated to the solution representation and objective function, the following intermediate decision variables are used in our model to better express the required constraints and/or search strategy:

- Pipe segment existence flags \( f^p_i \in \{0,1\} \), \( p \in \mathcal{P}, i \in \mathcal{S}_p \), with \( f^p_i = 1 \) iff segment \( i \) of pipe \( p \) has length greater than 0.
- Pipe segment directions \( d^p_i \in \{0,\ldots,5\} \), \( p \in \mathcal{P}, i \in \mathcal{S}_p \) using the same convention as for nozzles, that is, values \( \{0,1,2\} \) indicate segment \( i \) of pipe \( p \) has a non-negative direction along the \( x, y \) and \( z \) axes, respectively, while values \( \{3,4,5\} \) indicate a non-positive direction along the same axes. Note that the values for the first and last nodes (the nozzles) are given as input.
- Absolute pipe segment lengths \( L^p_{ix}, L^p_{iy}, L^p_{iz} \) for each pipe \( p \in \mathcal{P} \) and segment \( i \in \mathcal{S}_p \) along the axes \( x, y, z \), respectively.
- Pipe segment cuboid hulls as the smallest-corner coordinates \( \tilde{X}^p_i \in \mathbb{Z}^3 \) and sizes \( \tilde{W}^p_i \in \mathbb{Z}^3 \) (ignoring bend radius), for each pipe \( p \in \mathcal{P} \), similar to the notation for equipment and support zone cuboids.

3.3 Constraints

**Symmetry Breaking Constraints.** We want to avoid searching for symmetric solutions, that is, solutions where the flags \( f^p_i \) are either 0 at the end, or 0 at the beginning of a pipe. The following symmetry breaking constraints ensure only those with 0s at the end are considered solutions, by imposing a non-increasing order among the segment existence flags of every pipe:

\[
f^p_i \geq f^p_{i+1}, \quad p \in \mathcal{P}, i \in \mathcal{S}_p \setminus \{\max\{\mathcal{S}_p\}\}
\]  

**Orthogonal Direction Change Constraints.** They ensure consecutive segments of a pipe do not form a line by, imposing a change in pipe segment direction for any two consecutive segments of every pipe:

\[
0 = f^p_{i+1} \lor d^p_i \mod 3 \neq d^p_{i+1} \mod 3, \quad p \in \mathcal{P}, i \in \mathcal{S}_p \setminus \{\max\{\mathcal{S}_p\}\}
\]
**Object Non-overlapping Constraints.** They ensure the pipes do not overlap with other pipes and any other equipment (note that the non-overlapping among equipment has already been achieved in phase one) and take into account the required safety distances. They are implemented by disjunctions over the coordinate points at each of the three axes, similarly to [14,17]. For example, for ensuring non-overlapping between pipes and other equipment, the constraints are as follows:

\[ \bigvee_{c=1}^{3} \left( \tilde{X}_{ic}^p + \tilde{W}_{ic}^p + s_p^M \leq X_c^m \lor X_c^m + W_c^m + s_p^M \leq \tilde{X}_{ic}^p \right), \quad p \in \mathcal{P}, i \in \mathcal{S}_p, m \in \mathcal{M} \quad (3) \]

For subsets of objects with equal safety distances, we could have instead used the `diffn.k` global constraint [1]. However, among the solvers we tested only OR-Tools [8] has it, and it is not enough to express the support constraints.

**Support Constraints.** Each bend’s base point (the intersection of the neighboring segments’ base lines) is required to be in a provided support zone, whose cost penalty is the one added to the objective function. Supporting only bends can be an under approximation, as long segments might also need to be supported. However, this seems to happen very rarely (once in our biggest benchmark).

The placement of bends within a support zone is modeled as a reified form of condition (3), namely we denote this placement by a boolean variable \( b_{SZ}^{pij} \) for each pipe \( p \), bend \( i \) and support zone \( j \):

\[ b_{SZ}^{pij} \leftrightarrow \bigwedge_{c=1}^{3} \left( \tilde{X}_{ic}^p \geq X_j^i \land \tilde{X}_{ic}^p \leq X_j^i + W_j^c \right), \quad p \in \mathcal{P}, i \in \mathcal{S}_p \setminus \{1\}, j \in \mathcal{S}_Z \quad (4) \]

To ensure each bend is placed within a valid support zone, we demand:

\[ \exists_j b_{SZ}^{pij}, \quad p \in \mathcal{P}, i \in \mathcal{S}_p \setminus \{1\} \quad (5) \]

According to our cost assumption, namely \( C_{SZ}^{jp} \in \{0, C_{SZ}^p\} \) \( \forall j \in \mathcal{S}_Z, p \in \mathcal{P} \), with the zero cost belonging to (disjoint) rack support levels, we add to the objective function the following variables for each bend’s support cost:

\[ C_{SZ}^p = C_{SZ}^p \sum_{j: C_{SZ}^{jp} = \{0\}} b_{SZ}^{pij}, \quad p \in \mathcal{P}, i \in \mathcal{S}_p \setminus \{1\} \quad (6) \]

**Flexibility Constraints.** Several approximate methods are described in [5], including the Guided Cantilever Method (GCM), which is the one implemented by our model, as it is reasonably accurate and not too complex. This method assumes that pipes are only fixed at the nozzle attachment points and their bends are rectilinear. The thermal expansion of pipe \( p \in \mathcal{P} \) along axis \( x \) is defined as:

\[ \Delta_p^x = L_p^x e_p, \quad (7) \]

where \( L_p^x \) is the \( x \)-distance between \( p \)’s nozzles and \( e_p \) is the unit thermal expansion of \( p \). The \( \Delta_p^y \) and \( \Delta_p^z \) are defined similarly.
According to the GCM, a segment \( i \in S_p \) with a non-zero length \( L \) in the \( y \) or \( z \) direction (with length \( L = L_{iy}^p \) or \( L = L_{iz}^p \), respectively), absorbs the following portion of the thermal expansion in the \( x \)-direction:

\[
\delta_x = \frac{L^3}{\sum_i(L_{iy}^p)^3 + \sum_i(L_{iz}^p)^3} \Delta_x^p,
\]

(8)

where \( \delta_x \) is the segment’s lateral deflection in the \( x \)-direction, \( \Delta_x^p \) is the overall thermal expansion of \( p \) in the \( x \)-direction given by (7), and \( \sum_i(L_{iy}^p)^3 + \sum_i(L_{iz}^p)^3 \) is the sum of the cubed lengths of all pipe segments of \( p \) that are perpendicular to \( x \). Similar equations can be written for the lateral deflections of a segment in the \( y \)- and \( z \)-directions.

Also, the deflection capacity \( \bar{\delta} \) of a segment under the method’s assumptions can be given as:

\[
\bar{\delta} = \frac{48L^2S_p^A}{E_pD_p},
\]

(9)

where \( S_p^A \) is the allowable stress range for \( p \), \( L \) is the segment length, \( E_p \) is the modulus of elasticity associated to \( p \)'s material, and \( D_p \) is the external diameter of pipe \( p \), all in appropriate units. Finally, the expansion stress on \( p \)'s segment is permissible when:

\[
\max\{\delta_x, \delta_y, \delta_z\} \leq \bar{\delta}.
\]

(10)

For \( \delta_x \) this can be re-written as:

\[
\kappa L \leq \sum_i(L_{iy}^p)^3 + \sum_i(L_{iz}^p)^3
\]

(11)

with \( \kappa = \Delta_x^pE_pD_p/S_p^A/48 \). We add inequalities (11) and their \( y \)-, \( z \)-counterparts to the model. Value \( \max\{\delta_x, \delta_y, \delta_z\}/\bar{\delta} \) is called the stress ratio of the segment.

3.4 Search Strategy

MiniZinc allows us to specify a custom search strategy for the current optimization model being solved. One of the advantages of constraint programming solvers is that they can benefit from such a custom search strategy, either following it strictly or interleaved with their own search strategy. Our MiniZinc code declared the following strategy for routing each pipe \( p \): the search first explores the values of search pairs \((f^p, d^p)\) in the order of the pipe’s segments \( i \in S_p \). For each variable, the value selection strategy uses binary search (called \texttt{indomain_split} in MiniZinc), which first splits the domain of the variable around the integer mean of its lower and upper bounds, and then searches within the lower half, followed by the upper half. Search strategies are simply ignored by MIP solvers.

4 MiniZinc and Its Solver-Specific Redefinitions

We have implemented the above model in the MiniZinc language [13] and solved it using several CP and MIP solvers. This is despite constraints (6) and (11) being
non-linear and, thus, not directly supported by MIP solvers. The MIP interface of MiniZinc [2] handles this by using an automatic solver-specific redefinition mechanism for constraints defined as predicates or functions: when the model is compiled for a target solver, the front-end looks for a solver-specific redefinition of each predicate or function used in the model. If none is provided, MiniZinc uses the default decomposition appearing in its standard library or forwards the constraint to the solver backend. For example, the standard library definition for the pow function of an integer variable, is as follows:

```plaintext
/** @group builtins.arithmetic Return \( a^x \) */
function var int : pow ( var int : x , var int : y ) =
let {
    int : yy = if is_fixed (y) then fix (y) else -1 endif ;
} in
    if yy = 0 then 1
    elseif yy = 1 then x else
    let { var int : r ;
            constraint int_pow(x,y,r);
} in r
endif ;
```

which calls predicate int_pow(x,y,r). As no solver we tested handles this predicate, we defined it to represent \( x^3 \) as \( x \cdot x \cdot x \) for CP solvers, and as \( \sum_{v=lb(x)}^{ub(x)} v^3(x = v) \) for MIP ones. For example, the latter was achieved by adding the following MiniZinc code to the linearization library:

```plaintext
predicate int_pow ( var int : x , var int : y , var int : r ) =
let {
    array [int , int ] of int : x2y
    = array2d (lb (x)..ub (x), lb (y)..ub (y),
                [ pow (X ,Y ) | X in lb (x)..ub (x) , Y in lb (y)..ub (y) ] )
} in
    r == x2y [x , y ];
```

5 Evaluation

We have performed several experiments aimed at evaluating the scalability and accuracy of our method. All these experiments were executed as a 1-thread process on an Intel(R) Core(TM) i7-4771 CPU @ 3.50 GHz. Figure 2 provides a view of solutions obtained for the largest benchmark.

5.1 Default Benchmark

All our benchmarks modify a default benchmark by considering either subsets of its \( M \) and \( P \) sets, or a different discretization parameter. This default benchmark models the acid gas removal unit of an existing LNG plant. Its container cuboid is sized 76 × 40 × 43 m length by width by height, and its discretization parameter is \( K = 200 \) mm, which gives 381 × 201 × 216 position points along axes \( x, y, z \), respectively. It also has a set \( P \) with 27 pipes, with diameters \( D_p \) between 50 and 750 mm, and a set \( M \), with the following 17 equipment pieces already positioned by phase one:
- 4 column vessels, with heights between 17 and 40 m
- 1 horizontal drainage vessel, grounded, with a footprint of $7 \times 2.5$ m
- 3 heat exchanger groups and 2 individual heat exchangers. Two of the groups are fin-fan blocks of sizes $17 \times 15 \times 2.5$ m and $10 \times 15 \times 2.5$ m. The two individual exchangers are $21 \times 3 \times 3$ m each and the third group is $8 \times 2 \times 3$ m
- 4 pump groups, grounded, of sizes from $3 \times 1 \times 1.5$ to $15 \times 8 \times 1.5$ m
- a source point and a sink point connecting the current unit to other parts of the plant
- a pipe rack of size $13 \times 13$ m cross-section running through the container cuboid length-wise (see below for details on support areas).

The set $SZ$ of support zones, where all pipe bends must be located, is as follows:
- 3 m zones around the 5 vessels, the 2 individual heat exchanges, one of the heat exchanger groups, and the multi-level pipe rack
- 50 cm thick “preferred levels” at heights 3, 6, 9, and 12 m in the pipe rack, corresponding to the established platforms in the rack
- 0–3 m layer above ground.

The above pipes, bends and support zones have the following costs:
- length cost $C^L_p$: $25$–$400$ per meter, depending on diameter and routing requirements
- bend cost $C^B_p$: twice the per-meter cost
- support cost $C^{SZ}_{jp}$: $10 \times$ the per-meter length cost of $p$ for all bends except those located in the 0–3 m “ground level zone” or in the “preferred rack levels” at heights 3, 6, 9, and 12 m, which have no cost.

All benchmarks require a safety distance $S^M_p$ of 75 mm between pipes and between pipes and equipment. The upper bound for the length of a nozzle segment is 6 m. The lower bound for the length of any segment is $2 \times$ the diameter. Finally, all benchmarks impose the following flexibility/stress capacity requirements on all pipes in the plant, which are taken from the example in Sect. 4.5 in [5]:
- $e = 0.078$ in/ft, corresponding to the temperature range from 70 to 480 °F
- $E = 29 \cdot 10^6$ psi
- $S^A = 21625$ psi.

While the costs and stress values given above are not the real ones, we use them to report on the approach (the actual values cannot be disclosed).

### 5.2 Overall Approach

The overall approach follows a single-pipe configuration strategy, where each pipe is routed in sequence and the resulting route becomes an obstacle for the next pipe to be routed. In other words, the first pipe is routed in the context of the equipment, plus other pipes’ nozzle segments, as obstacles. The second
pipe is routed in the same context plus the obstacles resulting from the segments of the first pipe routed, and so on. We experimented with several pipe orders, including widest pipe first (according to diameter), and largest-surface pipe first (according to a combined measure of approximated length × width). With a minimal advantage for the latter, we choose it for the experiments. As shown below, the loss in accuracy due to our single-pipe routing is small.

Each pipe starts with 10 as the maximal number \( N_p^S \) of segments, and increases it if infeasible. Also, each pipe is first routed without the GCM flexibility constraints, and subsequently re-routed with them if stress violations occur (i.e., if the stress is greater than that allowed by \( S_p^A \)). For example, Fig. 1 shows the result of routing pipes without GCM constraints (left), and the subsequent loops added to reduce the stress down to allowable levels (right).

![Fig. 1.](image)

**Fig. 1.** Left: shows pipes routed without flexibility considerations, where dashed pipes indicate pipe segments with greater than the allowable stress \( S_p^A \): the shorter the dashes the higher the stress. Right: shows re-routed pipes with extension loops to relieve stress.

### 5.3 Results for Solvers Gurobi, IBM ILOG CPLEX, Chuffed, Gecode, and OR-Tools

The wide choice of solver backends for MiniZinc allowed us to try several solvers, including the two state-of-the-art MIP solvers Gurobi 7.0.2 [10] and IBM ILOG CPLEX 12.7.1 [11]. In terms of CP solvers, we tried the following three: Chuffed [4], one of the best solvers to combine constraint propagation with (SAT-style) clause learning, compiled from the develop branch on [3]; OR-Tools FlatZinc 5.1.4045 [8], Google’s fast and portable suite for combinatorial optimization; and Gecode 5.1.0 [16], one of the most popular CP solvers based on traditional constraint propagation. All these solvers have shown prominent performance in the annual MiniZinc Challenges [18].

For the MIP solvers it proved best to use the simple ‘big-\( M \)’ translation of logical constraints, cf. [2]. For CP solvers, OR-Tools did not produce any feasible solutions in the 1800-seconds time limit allowed per benchmark, while the best results for Chuffed were produced with option \(-f\) (free search).
Table 1. Results for routing default benchmark without flexibility constraints. The total objective value is different among solvers despite every pipe being routed optimally. This is due to the sequential routing approach.

<table>
<thead>
<tr>
<th>Solver</th>
<th>$N_{\text{calls}}$</th>
<th>$N_{\text{init_obst}}$</th>
<th>$N_{\text{last_obst}}$</th>
<th>$N_{\text{opt}}$</th>
<th>$t_{\text{min}}$</th>
<th>$t_{\text{max_opt}}$</th>
<th>$t_{\text{Total}}$</th>
<th>$L$</th>
<th>$S$</th>
<th>Obj</th>
<th>NB</th>
<th>NOS</th>
<th>$r_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>27</td>
<td>68</td>
<td>152</td>
<td>27</td>
<td>3.1</td>
<td>14</td>
<td>150</td>
<td>741</td>
<td>982</td>
<td>1765</td>
<td>111</td>
<td>55</td>
<td>2809%</td>
</tr>
<tr>
<td>Gurobi</td>
<td>27</td>
<td>68</td>
<td>153</td>
<td>27</td>
<td>3.6</td>
<td>71</td>
<td>318</td>
<td>741</td>
<td>982</td>
<td>1769</td>
<td>114</td>
<td>60</td>
<td>2755%</td>
</tr>
<tr>
<td>Chuffed</td>
<td>27</td>
<td>68</td>
<td>153</td>
<td>27</td>
<td>0.5</td>
<td>565</td>
<td>2038</td>
<td>753</td>
<td>985</td>
<td>1757</td>
<td>114</td>
<td>61</td>
<td>2809%</td>
</tr>
<tr>
<td>Gecode</td>
<td>27</td>
<td>68</td>
<td>155</td>
<td>27</td>
<td>0.3</td>
<td>261</td>
<td>685</td>
<td>764</td>
<td>987</td>
<td>1758</td>
<td>116</td>
<td>59</td>
<td>2809%</td>
</tr>
<tr>
<td>indep.</td>
<td>27</td>
<td>16</td>
<td>16</td>
<td>27</td>
<td>1.0</td>
<td>3</td>
<td>49</td>
<td>758</td>
<td>1004</td>
<td>1660</td>
<td>106</td>
<td>48</td>
<td>2809%</td>
</tr>
</tbody>
</table>

The top part of Table 1 shows the results of sequentially routing each of the 27 pipes in the default benchmark without flexibility constraints using CPLEX, Gurobi, Chuffed and Gecode. The meaning of each column is as follows: number of routing instances $N_{\text{calls}}$ attempted (would be larger than 27 if there is at least one pipe that needs more than 10 segments, or such a solution is hard to find); number of obstacles (i.e., already placed equipment or pipe segments) $N_{\text{init\_obst}}$ and $N_{\text{last\_obst}}$ for the first and last pipe, respectively; number of optimally solved pipes $N_{\text{opt}}$ (would be less than 27 if any pipe timed-out); minimum solving time $t_{\text{min}}$; maximum solving time for an optimally solved instance $t_{\text{max\_opt}}$; total time spent $t_{\text{Total}}$; total pipe length $L$, total pipe surface $S$, total objective value Obj; total number of bends NB; total number of overstressed segments NOS; and maximal stress ratio $r_{\text{max}}$. All times are given in seconds, lengths (surface area) in (square) meters, and objective values in multiples of $1000. Note that the initial number of obstacles is larger than the cardinality of set $\mathcal{M}$, because the start and end nozzle segments of each pipe are also considered as obstacles for all other pipes. Also note that no pipe needed more than 10 segments (as $N_{\text{calls}} = 27$).

All four solvers produce similar solutions, solving each pipe optimally. Thus, differences in the total objective value are explained only by the sequentiality of the approach (pipe after pipe) and the possibly different optimal solutions found by each solver. Since CPLEX is the fastest overall, we select CPLEX to perform the initial routing of each pipe without GCM constraints.

The last line of Table 1 presents a computation (with CPLEX) where each pipe was routed independently, i.e., ignoring other pipes. The total objective value is 6.2% smaller than the worst one among sequential approaches, giving a lower bound on what can be achieved with simultaneous routing. Note that this lower bound is independent of the solver, as long as every pipe is optimal.

As shown in Table 1, the maximal stress ratio $r_{\text{max}}$ is well above the allowed 100%, which is why flexibility constraints are needed. Table 2 shows the results for the full method, where we first try each pipe without flexibility constraints and only re-solve with GCM flexibility constraints if the stress is over the allowable limit. The first four rows of Table 2 show the results for the default benchmark, which is the same as that in Table 1 (27 pipes in $\mathcal{P}$, 17 elements in $\mathcal{M}$, and discretization parameter $K = 20$ cm), and different solvers for GCM routing:
Table 2. Comparison of various configurations of routing with GCM constraints

| Config | $|P|$ | $N_{pipes}^{GCM}$ | $N_{init}^{obst}$ | $N_{lst}^{obst}$ | $N_{opt}$ | $t_{min}^{opt}$ | $t_{max}^{opt}$ | $t_{Total}$ | $L$ | $S$ | $Obj$ | $NB$ | $L_{max}$ |
|--------|-----|-------------------|-------------------|-----------------|-----------|----------------|----------------|-------------|----|----|-------|-----|---------|
| Default | CPLEX | 27 | 22 | 72 | 181 | 22 | 8.0 | 925 | 2361 | 1002 | 1379 | 2089 | 144 | 98% |
|        | Gurobi | 27 | 21 | 72 | 180 | 18 | 13.9 | 1501 | 1004 | 1010 | 1381 | 2048 | 143 | 99% |
|        | Chuffed | 27 | 23 | 72 | 181 | 23 | 1.1 | 441 | 1476 | 997 | 1362 | 2105 | 144 | 99% |
|        | Gecode | 27 | 23 | 72 | 181 | 23 | 0.9 | 1661 | 3711 | 1008 | 1378 | 2053 | 144 | 98% |
| indep. |         | 27 | 22 | 16 | 16 | 22 | 2.8 | 691 | 1228 | 1008 | 1380 | 1999 | 133 | 99% |
| Decrease | $|M| = 13$ | 17 | 13 | 44 | 108 | 13 | 1.5 | 420 | 925 | 656 | 889 | 1270 | 88 | 96% |
|        | $|M| = 9$ | 10 | 6 | 30 | 59 | 6 | 1.8 | 121 | 234 | 419 | 631 | 573 | 51 | 98% |
|        | $|M| = 5$ | 3 | 3 | 8 | 19 | 3 | 1.9 | 267 | 281 | 252 | 594 | 389 | 18 | 97% |
| Discr  | CPLEX | 27 | 22 | 68 | 175 | 21 | 12.2 | 328 | 2414 | 1009 | 1377 | 1948 | 144 | 98% |
|        | Chuffed | 27 | 23 | 68 | 182 | 20 | 4.1 | 823 | 8531 | 1016 | 1387 | 2218 | 151 | 99% |

CPLEX, Gurobi, Chuffed and Gecode. The data shown is similar to that of Table 1, with the addition of $N_{pipes}^{GCM}$, which shows the number of pipes that needed to be re-routed with GCM constraints. Again, no pipes needed more than 10 segments (data not shown).

The next row shows, again, a computation where each pipe was routed (with CPLEX) independently of other pipes. The objective value improvement from the worst sequential approach is 5%. Interestingly, independent routing with the chosen objective function reduces the number of bends, but not the total length.

The next three rows show the results for decreasing numbers of equipment (and thus, of associated pipes) as follows: we first removed from the original $M$ all but 1 heat exchanger, leaving 13 elements; then removed all pump groups, leaving 9; and finally removed all but the 2 main components of the plant unit plus the source, sink, and pipe rack. GCM routing was done with Chuffed. The results show that even with fewer obstacles, routing remains a hard problem, probably due to the GCM constraints.

The final two rows show the results for the default benchmark with a smaller discretization parameter: $K = 10$ cm. CPLEX had a timeout on one pipe but the total objective value is 6.7% smaller than its own with $K = 20$ cm and smaller than independent routing with $K = 20$ cm. This shows that finer discretization can lead to qualitatively better solutions for the current model.

6 Visualization of Layout Solutions

The solutions to our model are text-based geometric descriptions of the optimized plant layout. These descriptions list the positions and dimensions of the equipment and pipe racks (from phase one), and the routes for the pipes as a series of intermediate bend locations (from phase two). This format is not understood by humans as easily as an image of the 3D layout. Moreover, plant layout and piping engineers are nowadays used to working with technical drawings and interactive 3D CAD models that allow them to zoom in and out and rotate the model in three dimensions in order to fully understand it.
We have developed an interactive 3D visualization that enables engineers to explore the produced layout, and to evaluate and validate the proposed solution in a familiar way (see Fig. 2 for two examples). This visualization displays pipes in different colors, drawing them as cylinders (rather than cuboids) of the appropriate diameter with visual bends.

![Fig. 2. Solutions for our largest problem instance with (right) and without (left) GCM constraints. The brown base represents the ground level. The cuboids represent equipment. The pipe rack is comprised of the four stacked plane-like cuboids that span most of the length of the volume. Pipes are depicted in different colors to differentiate them. (Color figure online)](image)

Our visualization is displayed in a web-based 3D viewer, allowing the 3D model to be rotated, panned and zoomed. We attach additional metadata to objects in the 3D model, such as equipment and pipe IDs and pipe segment stress. Engineers can see this information by selecting these elements and viewing their properties. We also attach high-level information (such as parameters of the optimization model) to the base of the container cuboid, allowing this to be easily viewed. We have also experimented with using colors or dashes to visualize stress on pipe segments when it is over the allowable levels, although this is mainly for our own benefit while working on the optimization model, since our final solutions must always keep stress within allowable levels.

We have found that using 3D visualizations during our discussions with Woodside engineers not only leads to a more fluid discussion, but allows them to quickly identify layouts that look very different from what they would expect. Often this means there is some requirement we were not aware of. In this case a nice benefit of the problem being specified as a high-level constraint model is that these additional requirements can generally be encoded as constraints and easily added to the model. This has happened multiple times resulting in an iterative evolution of our MiniZinc model and the overall approach over time. On other occasions, the unexpected results did not violate any requirement. Instead, they challenged Woodside engineers to reconsider long-held process plant con-
An Optimization Model for 3D Pipe Routing with Flexibility Constraints 335

construction and layout conventions, where diverging from the status quo has the potential to result in significant cost savings.

While discussing various solutions and visualizations, Woodside engineers have identified the need to easily evaluate the quality of various solutions. This requires us to annotate the 3D models with information regarding pipe and support structure material costs, on a pipe-by-pipe basis. These annotations will allow engineers to compare multiple potential solutions and compare the costs of particular subsets of the plant.

We are currently extending our plant layout visualization in two ways. The first extension allows engineers at different physical locations to explore the same visualization in a collaborative way, as is required in, say, a video conference setting. This extension allows the interactions (e.g., rotations and pipe selections) made in one location to be seen at all other locations. The second extension will allow engineers to modify parameters and constraints, such as safety distances or location of some equipment and/or pipes, thus either triggering a re-optimization step (for the pipes connected to the moved equipment) or a re-evaluation step to obtain the new stress and length values for the moved pipes. While re-optimization steps might take hours (depending on how many pipes need to be re-computed), this can be a significant improvement from the current manual process. A comparison between the two layouts will be performed in both cases. The aim is for both extensions to eventually form part of an interactive optimization tool for plant layout that can be used directly by plant layout and piping engineers and be closely integrated into the design process.

7 Conclusions and Outlook

We have presented a MiniZinc model to solve the 3D pipe routing problem under thermal expansion requirements (modeled by the Guided Cantilever Method) and support constraints, thus addressing the major constraints of the practical problem to a certain degree of detail. Different MIP and CP solvers were used to solve this model for the largest benchmark presented in the literature, and the results used to fine-tune a solving method that was shown to achieve near-optimal results with reasonable efficiency. The flexibility constraints represent a major challenge in the current version, significantly increasing the optimal subproblem solving times (which are nevertheless a step forward compared to the manual design process). We plan to extend this model to include maintenance constraints and other units within the plant.

We have also developed a 3D visualization of solutions as a first step towards an interactive optimization system. While it already allows users to evaluate and validate a solution, the aim is to transform the current version into an interactive visual interface that allows users to make changes of the optimization model via direct manipulation of a 3D model. These actions would trigger the optimization software to compute a new solution, which is then visualized and compared to the previous one.
Acknowledgments. This research was funded by Woodside Energy Ltd. We thank all our Woodside collaborators, particularly Solomon Faka, for the many useful discussions, as well as for the enlightening visit to their LNG plant.

References

