POMS PRODUCTION AND OPERATIONS MANAGEMENT

PRODUCTION AND OPERATIONS MANAGEMENT POMS

Vol. 0, No. 0, xxxx–xxxx 2021, pp. 1–19
ISSN 1059-1478 | EISSN 1937-5956 | 21 | 00 | 0001

DOI 10.1111/poms.13441

Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition

Nathan Sudermann-Merx
Data Science for Materials, BASF SE, Ludwigshafen, 67056, Germany, nathan.sudermann-merx@basf.com

Steffen Rebennack*
Institute for Operations Research, Karlsruhe Institute of Technology (KIT), Karlsruhe, 76185, Germany, steffen.rebennack@kit.edu

Christian Timpe
Advanced Business Analytics, BASF SE, Ludwigshafen, 67056, Germany, christian.timpe@basf.com

We present a new crossing number problem, which we refer to as the edge-constrained weighted two-layer crossing number problem (ECW2CN). The ECW2CN arises in layout planning of hose coupling stations at BASF, where the challenge is to find a crossing minimal assignment of tube-connected units to given positions on two opposing layers. This allows the use of robots in an effort to reduce the probability of operational disruptions and to increase human safety. Physical limitations imply maximal length and maximal curvature conditions on the tubes as well as spatial constraints imposed by the surrounding walls. This is the major difference of ECW2CN to all known variants of the crossing number problem. Such as many variants of the crossing number problem, ECW2CN is $\mathcal{NP}$-hard. Because the optimization model grows fast with respect to the input data, we face out-of-memory errors for the monolithic model. Therefore, we develop two solution methods. In the first method, we tailor Benders decomposition toward the problem. The Benders subproblems are solved analytically and the Benders master problem is strengthened by additional cuts. Furthermore, we combine this Benders decomposition with ideas borrowed from fix-and-relax heuristics to design the Dynamic Fix-and-Relax Pump (DFRP). Based on an initial solution, DFRP improves successively feasible points by solving dynamically sampled smaller problems with Benders decomposition. Because the optimization model is a surrogate model for its time-dependent formulation, we evaluate the obtained solutions for different choices of the objective function via a simulation model. All algorithms are implemented efficiently using advanced features of the GuRoBi-Python API, such as callback functions and lazy constraints. We present a case study for BASF using real data and make the real-world data openly available.

Key words: optimization; crossing number; mixed integer linear programming; Benders decomposition with direct cut calculation; Benders-based branch-and-cut; dynamic fix-and-relax pump; simulation
History: Received: July 2020; Accepted: April 2021 by Qi Feng after 1 revision.

1. Introduction

BASF is the world's leading chemical company (Tullo 2018). It has customers in over 170 countries and supplies about 8.000 products to almost all industries. BASF operates more than 350 sites worldwide, six of them being major, so called "Verbund" sites. The largest site is in Ludwigshafen, Germany, where BASF has its headquarters. The Verbund system is one of the core competencies of BASF; it creates efficient value chains that extend from basic chemicals to high-value-added products such as coatings and crop protection agents. In addition, the by-products of one plant can be used as the starting materials of another.

As part of its Verbund system, BASF operates more than 2.800 kilometers of pipelines in Ludwigshafen. While most of the pipes are dedicated to a certain product or utility, some of them are used for different products alternatingly. To maximize flexibility, BASF operates switching hubs, where dozens of pipes end in the same location, and any two of them might be connected on demand by flexible tubes. The work of connecting and disconnecting pipes was a manual and very unpopular task, due to the weight of the steel pipes and the need for full body safety equipment in the switching hub. However, necessary sub-tasks, such as leaking test, flushing, and drying made it difficult to obtain an automated solution. Currently, the engineering department of BASF is designing a dual-arm robotic system to accomplish that task. As the robot is

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

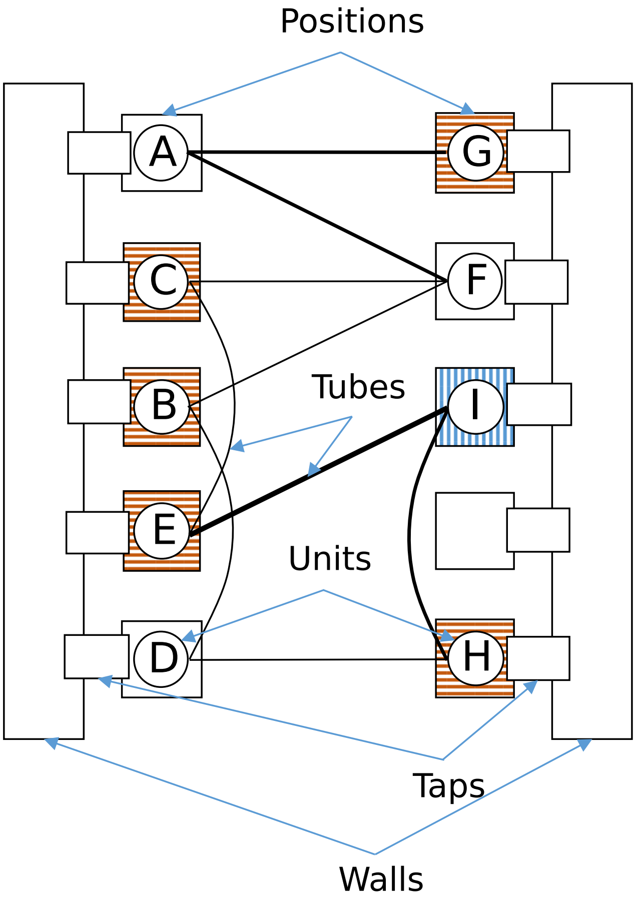
mounted on the ceiling, with its two hands, it cannot remove a pipe crossed by another one. For that, the robot must either remove and rebuild all blocking connections, or release the blocked pipe to the ground, where it has to be removed manually later. Both possibilities impede the efficiency of the system. Thus, the Advanced Business Analytics department within BASF was asked for a mathematical solution of this problem, which translates into a crossing minimal design where the number of blocking connections (so-called FIFO crossings) is to be reduced to a minimum.

To that end, we are given a two-layer hose coupling station where units are positioned on two opposing sides surrounded by walls. These units are to be connected by tubes. Figure 1 depicts such a hose coupling station. The connecting tubes (edges) have to obey three physical restrictions (constraints). First, tubes must lie inside the box drawn by the units because the units abut walls. A close distance to the surrounding walls is needed since the fluid flows through units and the tubes emerge from taps integrated in the walls. Due to security reasons, these taps have to be close to the units without any disturbing elements in between, *that is*, moving the units away from the wall to gain more freedom for tube placement is not an option. Second, the tubes' length has an upper bound which is imposed by the size of the tube storage. Third, the tubes can only be bent up to a maximal curvature; otherwise, they break. Moreover, the connections between the units change over time, *that is*, the tubes have to be reconfigured. Possible failure during the disconnection process may cause severe health risks due to leaking chemicals. Therefore, BASF is taking very high safety precautions, like full body protection. In the future, this is supposed to be executed by a dual-arm robot. As mentioned above, for the robot, crossing tubes pose a challenge and may lead to additional efforts, as all tubes stacked above need to be disconnected first. Therefore, a layout plan for the units, which is crossing minimal with respect to the connecting tubes, is highly valuable.

In the real-world problem at BASF, the locations of 75 units have to be determined. The connections between the units change over time, and we recorded historical connection data over a time span of more than 3 years. Next, we performed a thorough data cleansing in close cooperation with the operating engineering department at BASF and dropped all historical connections which did not generalize well for future considerations. Eventually, we agreed on 375 representative connections whose weights were determined by their historical connection frequency. This leads to a graph with 75 nodes and 375 weighted edges, having an edge-to-node ratio of 5.

The available data do not allow a meaningful probabilistic evaluation of possible future connections, because the number of observations is very low compared to the huge number of possible pairings. Consequently, the data are too sparse to extract distributional information, let alone the fact that some sets of pairings exclude each other, while other sets are likely to co-exist. Therefore, we do not set up a stochastic optimization model. Instead, we separate the problem into a deterministic design optimization problem, which serves as surrogate model, and a simulation problem, that is used for quality evaluation.

Figure 1 An Illustration of a Possible Assignment of 9 Tube-connected Units to 10 Positions. Due to the Maximal Curvature Condition, no Direct Neighbors may be Connected and a Maximal Tube Length has to be Respected. This is Illustrated for the Vertically Hatched Position (*I*) where all Reachable Positions are Horizontally Hatched (*B*, *C*, *E*, *G*, *H*). The Thickness of the Edges is Proportional to its Historical Connection Frequency [Color figure can be viewed at wileyonlinelibrary.com]



The deterministic design optimization problem leads to what we call the edge-constrained weighted two-layer crossing number problem (ECW2CN). We formulate this problem as a mixed integer linear program (MILP). The MILP finds an optimal assignment using aggregated data, *that is*, time independent data, considering both the number of connections between each pair of units and the aggregated connection frequencies. A natural objective function is the minimization of the number of reconfigurations over time which require the removal of FIFO connections. Because our proposed MILP is static, such an objective function cannot be modeled directly. To overcome this challenge, we set up multiple MILPs that differ in their treatments of the weighted edges by employing four different objective functions. Using these four different objective functions, we generate a series of feasible points for the MILP models. These points are then evaluated in a simulation model. This simulation model captures the complexity of the design problem by taking time dynamics and historical connections into account. In a second simulation model, only so-called FIFO crossings are counted, *that is*, reconfigurations which require the removal of blocking connections which cause operational disruptions. The best performing point in the second simulation model is then chosen as the optimal design. Computing the FIFO crossings for all feasible points is not possible since the second simulation model is very time consuming.

The ECW2CN is a non-standard edge-weighted crossing minimization problem (Schaefer 2020). Because this problem has not been discussed in the literature before, we develop an optimization model and tailor solution techniques to solve this real-world problem.

Our unique contributions are:

- We introduce the *edge-constrained weighted two-layer crossing number problem (ECW2CN)*. This is a new variant of the crossing number problem, motivated by a real-world problem at BASF.
- We analyze the computational complexity of ECW2CN and present MILP formulations, as well as two tailored decomposition algorithms: a Benders decomposition with direct cut calculation (BDC) and a Dynamic Fix-and-Relax Pump (DFRP). The master problem of BDC is strengthened by incorporating some information from relevant parts of the Benders subproblems. These additional cuts are provably stronger than the Benders optimality cuts. All cuts are computed analytically yielding a very efficient algorithm. The DFRP employs the BDC algorithm in a fix-and-relax fashion where the problem sizes are dynamically adjusted. This way, DFRP computes high quality solutions while being able to theoretically provide an optimality certificate.
- We provide a case study using real data of BASF. The real-world problem is solved by state-of-the-art implementations of all algorithms. Furthermore, final solutions of the real-world problem for different choices of the objective function are evaluated and compared in a time-dependent simulation model.

The remainder is organized as follows. We place the ECW2CN problem in the literature in section 2. In section 3, we present a mathematical programming model for the ECW2CN problem and show that ECW2CN is $\mathcal{NP}$-hard. This is followed by our two solution algorithms in section 4. The case study of the real-world problem is presented in section 5, and we conclude with section 6.

2. Crossing Number Literature

The crossing number problem for a given graph G consists of finding a drawing of G, such that this drawing possesses the lowest number of edge crossings (Schaefer et al. 2008, Székely 2004). Many variants of graph crossing number problems have been studied during the last decades. The review paper by Schaefer (2020) lists about 86 different variants. Therefore, we focus our review on important aspects of crossing number problems that are relevant in our context.

The crossing number problem appeared in the 1940s in the context of railway crossing minimization (Turán 1977) and in the 1960s in circuit design (Sinden 1966). However, it took until 2005 for the first exact integer linear programming (ILP) formulations to appear (Buchheim et al. 2005), although, for the related maximum planar subgraph problem, ILP formulations were published a decade before (Jünger and Mutzel 1996a,b. For a general graph $G(V, E)$, the ILP formulation by Buchheim et al. (2005, 2008) models edge crossing via binary decision variables $x_{e,f}$ for all edges $e \in E$ and $f \in E$. The difficulty lies then in ensuring that a graph exists for given values $x_{e,f}$. This is known as the realizability problem, which is itself an $\mathcal{NP}$-complete problem. Buchheim et al. (2005, 2008) ensure this realizability by a family of (potentially exponential many) constraints. In 2008, Chimani et al. (2008) presented an improved ILP formulation, compared to the one presented in Buchheim et al. (2005, 2008), requiring one order of magnitude fewer binary variables by finding a compacter way to model the graph realizability. The three mentioned ILP formulations are solved by tailored branch-and-cut algorithms, which combine advanced prepossessing (Buchheim et al. 2008) with combinatorial column generation (Chimani et al. 2008). To avoid the difficult realizability problem, we exploit the special structure of two-layer graphs present in ECW2CN (*cf.* section 3). This allows us to directly model the placement of the "nodes" (*that is*, units $u \in \mathcal{U}$ on possible places $p \in \mathcal{P}$) via binary decision variable $y_{u,p}$. The crossing is then modeled by considering all quadruples $p, p', \hat{p}, \hat{p}' \in \mathcal{P}$. This model allows us to consider various different ways of counting the edge crossings and

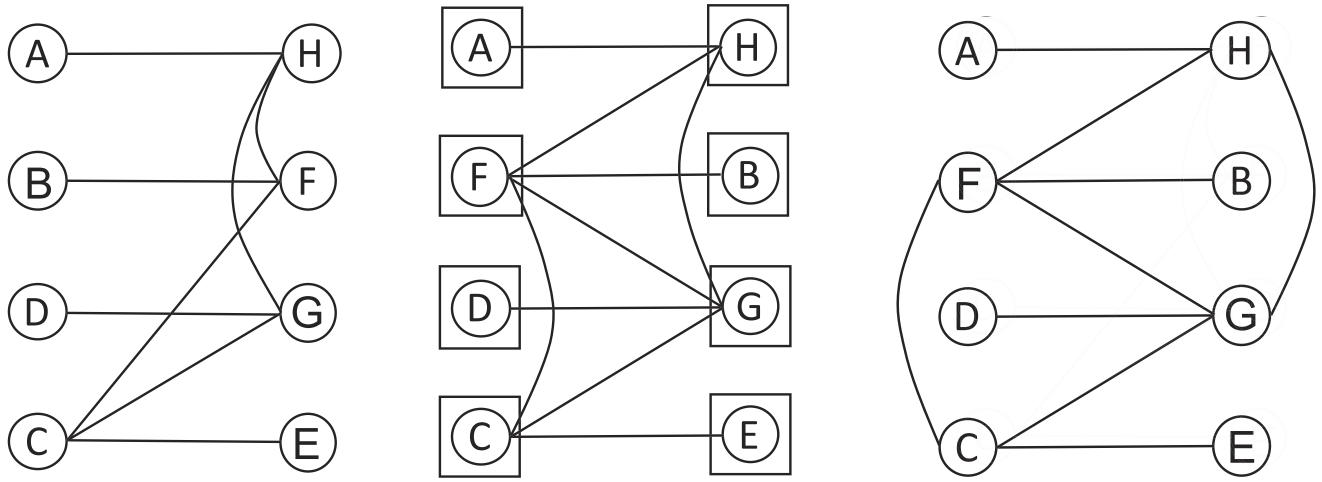
Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

to model the edge constraints regarding maximal length and maximal curvature explicitly.

In the literature, a variety of multi-layer and two-layer crossing problems are presented. Carpano (1980) discusses two types of hierarchical drawings (also called level drawings): horizontal drawings (nodes are placed on parallel lines) and radial drawings (nodes are placed on concentric circles) in a multi-layer context. All edges have to be drawn as straight lines, in contrast to general drawing problems where edges can contain bends. The level specification of each node has to be respected. A semidefinite programming approach for hierarchical drawing is presented by Chimani et al. (2012). Hierarchical drawings are extended to allow for intra-layer connections by Bachmaier et al. (2010); the intra-layer edges are allowed to bend. With that, Bachmaier et al. (2010) comes close to the ECW2CN problem, when restricting to two layers, which is illustrated in Figure 2. In the ECW2CN problem, we have also a two-layer set-up with inter-layer connections, but the units can be placed freely among the two sides in contrast to Bachmaier et al. (2010). Another similarity of the ECW2CN problem to Bachmaier et al. (2010) is that all edges need to stay inside the rectangle spanned by the location positions. Different to Bachmaier et al. (2010), the ECW2CN problem has restrictions on the inter- and intra-layer connections.

A special case of two-layer hierarchical drawings is obtained for bipartite graphs; in the context of multigraphs, Garey and Johnson (1983) call this problem the bipartite crossing number problem. Because only straight lines are allowed for the edges, Jünger and Mutzel (1996a, b) refer to this problem as the two-layer straight line crossing problem. This is also the usual setting for the *k*-layer case where only straight lines are allowed. In the two-layer straight line crossing problem, the bipartition of the node set has to be respected (hence, there are no intra-layer edges). An ordering of the nodes, for each of the two layers, determines the graph and its crossings. This allows for efficient permutation approaches (Kobayashi et al.

Figure 2 A Comparison of ECW2CN with Related Settings in the Literature, Yielding to Different Number of Minimal Crossings



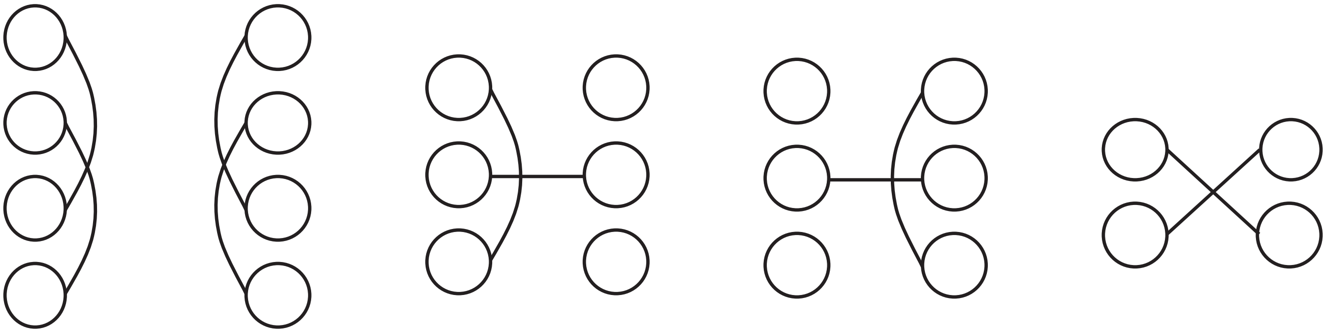
(a) 2-layer drawing allowing intra-layer connections (Bachmaier et al. 2010); 3 crossings are optimal for bipartition $(A, B, C, D) - (E, F, G, H)$

(b) ECW2CN with only 2 crossings; the bipartition $(A, B, C, D) - (E, F, G, H)$ is broken up

(c) Planar graph, allowing edges outside the rectangular box (Jünger and Mutzel 1996)

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

Figure 3 An Overview Over all Different Crossing Types in the ECW2CN



2014, Laguna and Marti 1999, Palubeckis et al. 2019). As such, the bipartite crossing number problem is highly related to the (quadratic) linear ordering problem (Buchheim et al. 2010, Shahrokhi et al. 2001). Edge-weighted variants of the bipartite crossing number problem are presented by Çakirog̅lu et al. (2009) and Kobayashi et al. (2014), where the weights are handled in a multiplicative manner. The ECW2CN problem also possess edge weights.

In sum, ECW2CN has the following key specifications:

- Maximal length and maximal curvature conditions for the tube (*that is*, edges) have to be met.
- Though we have a two-layer set-up, intra-layer connections are allowed; like the formulation in Bachmaier et al. (2010).
- All connections have to stay inside a rectangle spanned by the position locations; similar to the bipartite crossing number problem (Garey and Johnson 1983).
- Each node (*that is*, unit $u \in U$) can be placed freely among the two layers; thus, this problem differs from bipartite or two-layer crossing minimization problems and yields some additional overlapping types, see Figure 3.
- The edges are weighted due to the historical connection frequencies. However, in contrast to the set up of Kobayashi et al. (2014), weights can be handled in a very flexible framework. In this work, we examine four different manners.
- The corresponding graph does not have to be connected. Connectivity is a standard assumption as the crossing number problems decompose with their connected components. This is not true for ECW2CN because of the maximal length and maximal curvature conditions.
- The number of units and places do not have to be the same, *that is*, there might exist isolated nodes in the corresponding graph. This is a special case of a disconnected graph (see previous point). The "placement" of the isolated nodes is non-trivial due to the maximal length and maximal curvature conditions.
- We are facing an edge-to-node ratio of 5. Most instances found in the literature have a ratio between 1 and 2.

3. ECW2CN, its Complexity and MILP Formulations

Based on the definition of the ECW2CN in section 3.1, we prove its $\mathcal{NP}$-hardness (section 3.2) and present our MILP models (section 3.3).

3.1. The Edge-Constrained Weighted two-Layer Crossing Number Problem (ECW2CN)

We begin by defining the notation used in the definition of the ECW2CN as well as in the MILP.

Indices, sets and parameters

- $u \in \mathcal{U}$: Units
- $p \in \mathcal{P}$: Ordered set of positions; each position p has an (X_p, Y_p)-coordinate in the plane and all positions are aligned on two layers, *that is*, there are only two different values for X_p
- $\overline{D}$ >0: Maximal Euclidean distance between two units due to a maximal tube length
- $\overline{E}$ >0: Minimal distance between two units on the same layer due to maximal curvature
- $(p, p', \hat{p}, \hat{p}') \in \mathcal{C}$: Four tuples of positions, such that connections of units on positions $(p,p^{'})$ and $(\hat{p},\hat{p}^{'})$ respect the maximal tube length as well as the maximal curvature condition parameterized by $\overline{D}$ and $\overline{E}$ and induce one of the crossing types illustrated in Figure 3.
- $F_{u,u'} \geq 0$: Aggregated historical connection frequency of two units $u, u' \in \mathcal{U}$
- $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} \geq 0$: A penalty term that is used to penalize crossings weighted by their frequencies; therefore, $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} = 0$ if and only if $F_{u,u'} = 0$ and/or $F_{\hat{u},\hat{u}'} = 0$
- $(u, u', \hat{u}, \hat{u}') \in \mathcal{Q}$: Four tuples of *pairwise different* connected units, *that is*, $F_{u,u'} > 0$ and $F_{\hat{u},\hat{u}'} > 0$

- $(p,p') \in \mathcal{N}$: Pairs $(p, p^{'})$ that violate the maximal length or maximal curvature condition

We are now ready to formally define the ECW2CN.

Definition 1. (Edge-constrained weighted two-layer crossing number problem). Given is a set of units $\mathcal{U}$, positions $\mathcal{P}$ on two layers with coordinates (X_p, Y_p) for each position $p \in \mathcal{P}$, maximal Euclidean distance $\overline{D}$ between two units, minimal Euclidean distance $\overline{E} > 0$ between two units on the same layer, edge weights $F_{u,u'} \geq 0$ for pairs of units $u, u' \in \mathcal{U}$ and a function $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} \geq 0$ of edge weights for $u, u', \hat{u}, \hat{u}' \in \mathcal{U}$. Then, the edge-constrained weighted two-layer crossing number problem (ECW2CN) assigns each unit $u \in \mathcal{U}$ to at most one position $p \in \mathscr{P}$, such that the Euclidean distance of any pair of connected units is not greater than $\overline{D}$, at least $\overline{E}$ if both units are on the same layer and all edges stay within the rectangle spanned by the convex hull of the position candidates, while minimizing the sum over $P_{F_{u,u'},F_{\hat{u},\hat{u}'}}$ for all units $u, u', \hat{u}, \hat{u}' \in \mathcal{U}$ whose placement defines a crossing.

3.2. ECW2CN is $\mathcal{NP}$-Hard

To show that ECW2CN is $\mathcal{NP}$-hard, we first define a decision version of ECW2CN, denoted by ECW2CN-D.

Definition 2. (ECW2CN decision version).
Instance: A set of units $\mathscr{U}$, positions $\mathscr{P}$ on two layers with coordinates (X_p, Y_p) for each position $p \in \mathscr{P}$, maximal Euclidean distance $\overline{D}$ between two units, minimal Euclidean distance $\overline{E} > 0$ between two units on the same layer, edge weights $F_{u,u'} \geq 0$ for pairs of units $u, u' \in \mathscr{U}$, a function $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} \geq 0$ of edge weights for $u, u', \hat{u}, \hat{u}' \in \mathscr{U}$ and bound $C < \infty$.
Question: Is there an assignment of each unit $u \in \mathscr{U}$ to at most one position $p \in \mathscr{P}$, such that the Euclidean distance of any pair of connected units is not greater than $\overline{D}$, at least $\overline{E}$ if both units are on the same layer, all edges stay within the rectangle drawn by the position locations, and such that the sum over $P_{F_{u,u'},F_{\hat{u},\hat{u}'}}$ for all units $u, u', \hat{u}, \hat{u}' \in \mathscr{U}$ whose placement defines a crossing does not exceed C?Instance:

In our reduction, we use the decision version of the bipartite crossing number (BCN) problem, denoted by BCN-D.

Definition 3. (BCN decision version, (BCN decision version, Garey and Johnson (1983)).
Instance: A connected bipartite multigraph $G(V_1, V_2, E)$ and an integer $K < \infty$.
Question: Can G be embedded in a unit square so that all vertices of V_1 are on the northern boundary, all vertices in V_2 are on the southern boundary, all edges are within the square and there are at most K crossings?

We need the following complexity result.

Lemma 1. (Garey and Johnson (1983)). *The BCN-D is $\mathcal{NP}$-complete.*

For their reduction, Garey and Johnson (1983) utilize the Optimal Linear Arrangement problem which has itself been reduced from the MAX-Cut problem in (Garey et al. 1974). Therefore, BCN-D is $\mathcal{NP}$-complete in the strong sense.

This allows us to state and prove our main result.

Theorem 1. *The decision version ECW2CN is $\mathcal{NP}$-complete in the strong sense.*

Proof . Observe that ECW2CN-D is in $\mathcal{NP}$.

Given an instance of the BCN-D, *that is*, a connected bipartite multigraph $G(V_1, V_2, E)$, we construct an instance of the ECW2CN-D as follows:

- $\mathcal{U} = V_1 \cup V_2$
- $\mathcal{P} = \mathcal{U}$ with $(X_i, Y_i) = \left(0, \frac{i-1}{|V_1|-1}\right)$ for $i = 1, \ldots, |V_1|$ and $(X_i, Y_i) = \left(1, \frac{i-1}{|V_2|-1}\right)$ for $i = 1, \ldots, |V_2|$
- $\overline{D} = \overline{E} = 2$
- $F_{u,u'} = 0$ for all $u, u' \in \mathscr{U}$; $F_{u,u'} \leftarrow +1$ for all $(u, u') \in E$
- $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} = F_{u,u'} \cdot F_{\hat{u},\hat{u}'}$ for all $F_{u,u'}, F_{\hat{u},\hat{u}'} > 0$; o/w 0.

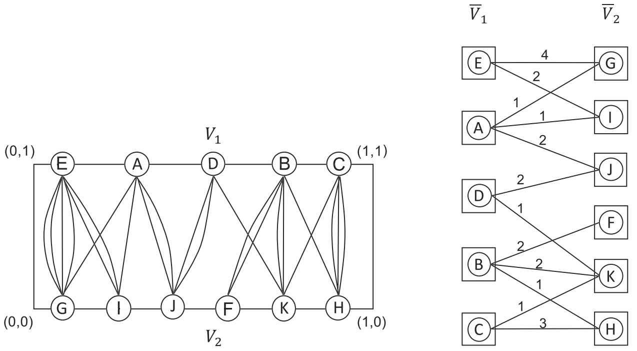
The idea is to use the restriction on the minimal distance between two units on the same layer to ensure that only nodes from V_1 (and V_2) are assigned to the same layer. This can be achieved by assigning any number $\overline{E} > 1$ (see below). The transformation is shown in Figure 4.

The maximal distance is no restriction as long as $\bar{D} \geq \sqrt{2}$. This construction is (strongly) polynomial in the size of G.

Next, we show that an instance for BCN-D with bound K is a “YES” instance, if and only if the transformed graph is a “YES” instance for ECW2CN-D with bound $C = K$. For that, we observe that both problems count the number of crossings. Thus, if the solution of one problem maps to a solution of the other, then the bounds are the same. It remains to show the existence of such a mapping.

“⇒” Any feasible point for ECW2CN-D yields a partition of $\mathscr{U}$ into two disjoint sets $\bar{V}_1$ and $\bar{V}_2$ where $\bar{V}_1 = V_1$ or $\bar{V}_1 = V_2$. To see this, assume that $v_1 \in V_1$ and $v_2 \in V_2$ are both assigned to $\bar{V}_1$. If $(v_1, v_2) \in E$, then v_1 and v_2 do not respect $\overline{E}$. If $(v_1, v_2) \notin E$, then there exits a path from v_1 to v_2 in G, because G is connected.

Figure 4 A Transformation of a "YES" Instance for $K \geq 5$ of the BCN-D to a "YES" Instance for $C \geq 5$ of the ECW2CN-D Using the Technique Described in the Proof of Theorem 1



(a) "YES" instance of the BCN-D

(b) "YES" instance of the ECW2CN-D

Choose any such path p. Then, there exists a node $v_3 \in V_1$ on p with $(v_2,v_3) \notin E$. If $v_3 \in \bar{V}_1$, then v_2 and v_3 do not respect $\overline{E}$; otherwise, $v_3 \in \bar{V}_2$ and $(v_1,v_3) \notin E$. Continuing this argument along path p from v_3 to v_1 shows the contradiction.

"$\Leftarrow$" Any feasible point for BCN-D respects $\overline{D}$ and $\overline{E}$. With that, it is a feasible point for ECW2CN-D.

3.3. The Mixed Integer Linear Programming Models

We use the indices, sets, and parameters as introduced in section 3.1. Furthermore, we require the following decision variables

- $y_{u,p} \in \{0,1\}$: Binary decision variable which is 1 if unit u is located at position p, 0 otherwise
- $o_{p,p',\hat{p},\hat{p}'} \geq 0$: Continuous decision variable that represents the maximal crossing frequency of arbitrary units located on the positions $(p,p',\hat{p},\hat{p}')$.

The ECW2CN can be modeled as the following MILP formulation, which we call the monolithic model.

$$(\mathcal{M}): z^* := \min_{y,o} \sum_{(p,p',\hat{p},\hat{p}') \in \mathcal{C}} o_{p,p',\hat{p},\hat{p}'} \tag{1}$$

$$\text{s.t.} \; (y_{u,p} + y_{u',p'} + y_{\hat{u},\hat{p}} + y_{\hat{u}',\hat{p}'} - 3) \cdot P_{F_{u,u'},F_{\hat{u},\hat{u}'}} \leq o_{p,p',\hat{p},\hat{p}'}$$

$$\forall (p,p',\hat{p},\hat{p}') \in \mathcal{C} \, \forall (u,u',\hat{u},\hat{u}) \in \mathcal{Q} \tag{2}$$

$$\sum_{p \in \mathscr{P}} y_{u,p} = 1 \quad \forall u \in \mathscr{U} \tag{3}$$

$$\sum_{u \in \mathscr{U}} y_{u,p} \leq 1 \quad \forall p \in \mathscr{P} \tag{4}$$

$$y_{u,p} + \sum_{u' : F_{u,u'} > 0} y_{u',p'} \leq 1 \quad \forall (p,p') \in \mathscr{N} \, \forall u \in \mathscr{U} \tag{5}$$

$$y_{u,p} \in \{0,1\} \quad \forall u \in \mathscr{U} \quad \forall p \in \mathscr{P} \tag{6}$$

$$o_{p,p',\hat{p},\hat{p}'} \geq 0 \quad \forall (p,p',\hat{p},\hat{p}') \in \mathcal{C} \tag{7}$$

The objective function (1) minimizes the weighted number of crossings. This weighted crossing is modeled through decision variables $o_{p,p',\hat{p},\hat{p}'}$. Constraints (2) together with the nonnegativity of $o_{p,p',\hat{p},\hat{p}'}$ ensures that whenever a crossing is induced by locating two pairs of connected units $u, u', \hat{u}, \hat{u}'$ on positions $p, p', \hat{p}, \hat{p}'$, we have $o_{p,p',\hat{p},\hat{p}'} = P_{F_{u,u'},F_{\hat{u},\hat{u}'}}$, *that is*, the crossing is weighted with $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} > 0$. Constraint blocks (3)–(4) are assignment restrictions. Each unit $u \in \mathscr{U}$ has to be put at exactly one place $p \in \mathscr{P}$ as ensured by constraints (3). That at most one unit $u \in \mathscr{U}$ is located at each place $p \in \mathscr{P}$ is given by the SOS-1 constraints (4). Constraints (5) ensure the maximal length and maximal curvature restrictions. We discuss these constraints now in detail.

Constraints (5) can be more naturally written as

$$y_{u,p} + y_{u',p'} \leq 1 \quad \forall (p,p') \in \mathscr{N} \, \forall u,u' \in \mathscr{U} : F_{u,u'} > 0. \tag{8}$$

These constraints (8) define a so-called conflict graph, where $y_{u,p}$ and $y_{u',p'}$ cannot both equal 1. Finding the maximal number of possible $y_{u,p} = 1$ values leads to the so-called stable set problem which is also known as the independent set problem. The stable set problem is related to the maximum clique problem through the inverse graphs. Both problems are classical and well-studied combinatorial optimization problems (Bomze 1997, Bomze et al. 1999, Rebennack et al. 2011). They are both $\mathscr{NP}$-hard (Garey and Johnson 1979). Constraints (8), together with nonnegativity of $y_{u,p}$, define the so-called stable set polytope. Therefore, we can make use of the knowledge about the stable set polytope in our context. Specifically, constraints (8) are facet defining for the stable set polytope, if and only if the edge in the conflict graph defines as maximal clique (Rebennack et al. 2012). This inspired constraints (5), which are a strengthened version of (8) as can be observed by adding the corresponding $F_u = |\{u' : F_{u,u'} > 0\}|$ constraints to obtain

$$F_u y_{u,p} + \sum_{u' : F_{u,u'} > 0} y_{u',p'} \leq F_u \quad \forall (p,p') \in \mathscr{N} \, \forall u \in \mathscr{U}.$$

Also note that there are $|\mathscr{N}| \sum_{u \in \mathscr{U}} F_u$ constraints with two non-zeros each, while there are only $|\mathscr{N}||\mathscr{U}|$ constraints (5) with F_u+1 non-zeros each. Therefore, (5) yield to fewer non-zeros, to less constraints and to stronger constraints compared to (8).

Nevertheless, constraints (5) are not facet defining for polytope (3)-(5) as they are dominated by

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

$$\sum_{u'\in\mathcal{U}} y_{u',p} + \sum_{u':F_{u,u'}>0} y_{u',p'} \leq 1 \quad \forall (p,p')\in\mathcal{N}\ \forall u\in\mathcal{U}. \qquad (9)$$

Because (9) are much denser than (5) and the maximal violation of constraints (9) is at most 1 for any point of the LP relaxation of (3)-(5), we consider (5) instead.

From the above discussions, we summarize

Proposition 1. *Formulation* $(\mathcal{M})$ *models ECW2CN correctly.*

We close this section with a few remarks.

Remark 1. The subsequent theoretical developments do not the depend on a particular choice of the crossing weights, *that is*, they hold true for all $P_{F_{u,u'},F_{\hat{u},\hat{u}'}}:\mathbb{R}^{>0}\times\mathbb{R}^{>0}\rightarrow\mathbb{R}^{>0}$. However, for the problem at hand we consider the following four candidates:

1. $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} = F_{u,u'} + F_{\hat{u},\hat{u}'}$ ("Sum")
2. $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} = \min(F_{u,u'}, F_{\hat{u},\hat{u}'})$ ("Minimum")
3. $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} = F_{u,u'} \cdot F_{\hat{u},\hat{u}'}$ ("Product")
4. $P_{F_{u,u'},F_{\hat{u},\hat{u}'}} = 1$ ("One")

For illustrative purposes, suppose we have a crossing induced by the positions of $(u, \overset{'}{u})$ and $(\hat{u}, \hat{u}')$. The durations of these connections are given by $F_{u,u'}$ and $F_{\hat{u},\hat{u}'}$. Then it is intuitive to see that the probability of having an actual crossing at a point in time increases for larger values of $F_{u,u'}$ and $F_{\hat{u},\hat{u}'}$. That motivates the "Sum" and the "Product" objective function for the time static MILP formulation. The second choice "Minimum" provides an upper bound on the expected number of frequencies since a crossing cannot last longer than $\min(F_{u,u'}, F_{\hat{u},\hat{u}'})$. The last candidate counts the unweighted number of crossings for benchmark purposes. These different variations are illustrated in Figure 5 in Example 1. In section 5, we show that there is indeed a high correlation between good feasible points with respect to the objective functions above and actual crossings which are computed by evaluating the time-dependent simulation model.

Example 1. The graph depicted in Figure 1 possesses nine nodes and nine edges whose weights are given by: (A, F): 10, (A, G): 8, (B, D): 6, (B, F): 2, (C, E): 3, (C, F): 5, (D, H): 1, (E, I): 15, (H, I): 12. Depending on the choice of the objective function, we obtain the following optimal assignments as illustrated in Figure 5.

Remark 2. In order to break some symmetry, we extend formulation $(\mathcal{M})$ by the constraint that the unit with the most connections must be located in the upper left half of the rectangle spanned by the positions.

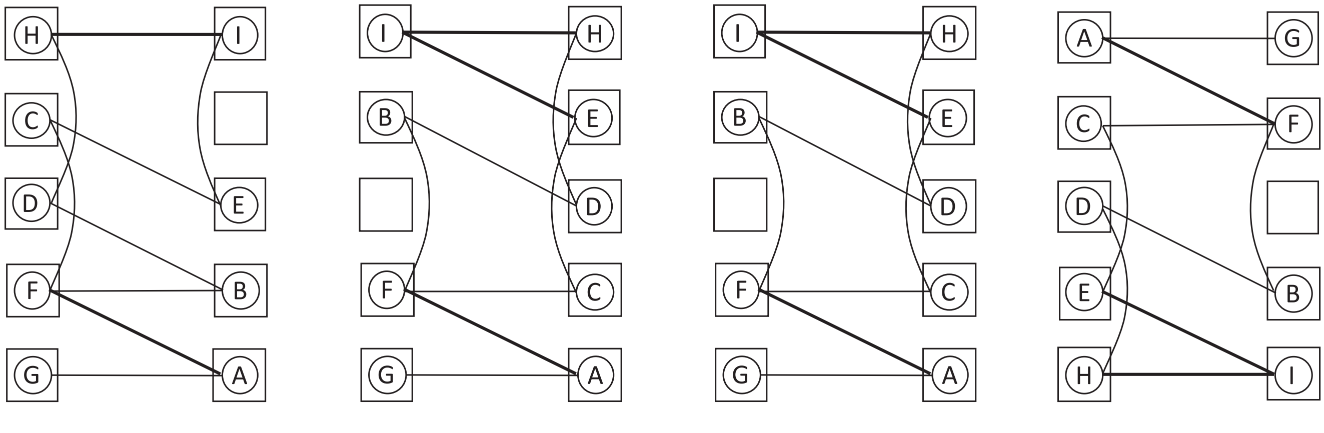
Remark 3. Observe that it is possible to reformulate the monolithic model $(\mathcal{M})$ as a minimax model of the form

$$\min_{y} \sum_{(p,p',\hat{p},\hat{p}')\in\mathcal{C}} \max_{(u,u',\hat{u},\hat{u})\in\mathcal{Q}} (y_{u,p}+y_{u',p'}+y_{\hat{u},\hat{p}}+y_{\hat{u}',\hat{p}'}-3)\cdot P_{F_{u,u'},F_{\hat{u},\hat{u}'}}$$
$$\text{s.t.}\,(3)-(6)$$

which reveals the intrinsic piecewise linear structure of its objective function.

Remark 4. For our real-world application, we have $|\mathcal{C}|=532{,}401$ and $|\mathcal{Q}|=129{,}922$, *that is*, Equation (2)

Figure 5 Optimal Configurations of the Graph in Figure 1 with Respect to Four Different Objective Functions



(a) Sum **(b)** Minimum **(c)** Product **(d)** One

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

translates into 69,170,602,722 constraints. Even to build the mathematical model $(\mathcal{M})$ on a personal computer with default settings is a challenge without running into memory issues. Due to its size, it is currently not possible to solve the monolithic model $(\mathcal{M})$ directly using state-of-the-art MILP solvers. Therefore, in section 4, we develop tailored solution approaches to tackle this problem.

4. Solution Methods

The monolithic formulation $(\mathcal{M})$ is a MILP problem of large scale, due to constraints (2) and variables $o_{p,p',\hat{p},\hat{p}'}$. We propose a tailored Benders decomposition to exploit its special structure (section 4.1). This Benders decomposition is then used within a fix-and-relax-inspired algorithm (section 4.2) in an effort to compute good feasible points for the real-world problem at hand.

4.1. A Tailored Benders Decomposition with Direct Cut Calculation (BDC)

4.1.1 Benders Decomposition. The idea of Benders (1962) was to separate a, what he called, mixed variables problem into a master problem and a subproblem. The mixed variables in our case are the $y_{u,p}$ and the $o_{p,p',\hat{p},\hat{p}'}$ variables. The master problem is a relaxation of the original problem and does not contain the so-called complicating constraints (in our case, constraints (2)), which connect both types of variables.

These constraints are passed onto the subproblem which computes, for a given solution of the master problem (in our case, trial values of variables $y_{u,p}$), the objective function value associated with an optimal "response" to $y_{u,p}$. Exploiting dual information, a feasibility or optimality cut is generated and passed on to the master problem. This way, the master problem is extended by information from the subproblem, in terms of dual extreme directions or extreme points. For a minimization problem, the master problem yields a lower bound and the trial solution together with the subproblem yields an upper bound. By iteratively solving master and subproblems until the upper and lower bounds are sufficiently close to each other, the original problem is solved. This so-called Benders algorithm always terminates after finitely many iterations because the dual subproblem contains only a finite number of extreme directions and extreme points. For further details, we refer to the literature (Rahmaniani et al. 2017, Rebennack 2016).

For a given trial solution $\hat{y}_{u,p}$ of the master problem, the subproblem $(\mathcal{S})$ is given by

$$z_S^*(\hat{y}) := \min \sum_{(p,p',\hat{p},\hat{p}')\in\mathcal{C}} o_{p,p',\hat{p},\hat{p}'} \tag{10}$$

$$\text{s.t.}\, o_{p,p',\hat{p},\hat{p}'} \geq (\hat{y}_{u,p} + \hat{y}_{u',p'} + \hat{y}_{\hat{u},\hat{p}} + \hat{y}_{\hat{u}',\hat{p}'} - 3)\cdot P_{F_{u,u'},F_{\hat{u},\hat{u}'}}$$

$$\forall (u,u',\hat{u},\hat{u}')\in\mathcal{Q}, (p,p',\hat{p},\hat{p}')\in\mathcal{C} \tag{11}$$

$$o_{p,p',\hat{p},\hat{p}'} \geq 0 \forall (p,p',\hat{p},\hat{p}')\in\mathcal{C} \tag{12}$$

Notice that $(\mathcal{S})$ is a linear programming (LP) problem which is feasible for any trial values $\hat{y}_{u,p}\in[0,1]$. Consequently, we do not need feasibility cuts.

Remark 5. Observe that for given $\hat{y}$ the unique solution of $(\mathcal{S})$ is given by

$$o^*_{p,p',\hat{p},\hat{p}'} = \max_{(u,u',\hat{u},\hat{u}')\in\mathcal{Q}} \left\{0, (\hat{y}_{u,p} + \hat{y}_{u',p'} + \hat{y}_{\hat{u},\hat{p}} + \hat{y}_{\hat{u}',\hat{p}'} - 3)\cdot P_{F_{u,u'},F_{\hat{u},\hat{u}'}}\right\} \quad \forall (p,p',\hat{p},\hat{p}')\in\mathcal{C}$$

which, in particular, implies its non-degeneracy. This rules out the possibilities to add further cuts of different strengths from degenerate solutions as elaborated by Magnanti and Wong (1981), Papadakos (2008) and Sherali and Lunday (2013).

Let $\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'}$ be an optimal dual (basic) solution associated with constraints (11). Because of strong duality, we obtain

$$z_S^*(\hat{y}) = z_D^*(\hat{y}) := \sum_{(u,u',\hat{u},\hat{u}')\in\mathcal{Q}} \sum_{(p,p',\hat{p},\hat{p}')\in\mathcal{C}} \Big((\hat{y}_{u,p} + \hat{y}_{u',p'} + \hat{y}_{\hat{u},\hat{p}} + \hat{y}_{\hat{u}',\hat{p}'} - 3)\cdot P_{F_{u,u'},F_{\hat{u},\hat{u}'}}\Big)\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'}.$$

For some (cut) index c, we define the (variable) cut coefficient

$$\pi^{\text{var}}_{u,p,c} := \sum_{u',\hat{u},\hat{u}':(u,u',\hat{u},\hat{u}')\in\mathcal{Q}} \; \sum_{p',\hat{p},\hat{p}':(p,p',\hat{p},\hat{p}')\in\mathcal{C}} \pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'} \quad \forall u\in\mathscr{U}, p\in\mathscr{P} \tag{13}$$

and cut constant

$$\pi^{\text{con}}_c := -3 \sum_{(u,u',\hat{u},\hat{u}')\in\mathcal{Q}} \sum_{(p,p',\hat{p},\hat{p}')\in\mathcal{C}} P_{F_{u,u'},F_{\hat{u},\hat{u}'}}\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'}.$$

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

Because the dual subproblem is a maximization problem, we obtain

$$z^*_S(y) \geq \sum_{u\in\mathscr{U},p\in\mathscr{P}} \pi^{\text{var}}_{u,p,c} y_{u,p} + \pi^{\text{con}}_c \quad \forall y \in \{0,1\}^{|\mathscr{U}|\times|\mathscr{P}|}.$$

This yields the optimality cut

$$\eta \geq \sum_{u\in\mathscr{U},p\in\mathscr{P}} \pi^{\text{var}}_{u,p,c} y_{u,p} + \pi^{\text{con}}_c$$

for the master problem with free variable η.

The master problem $(\mathscr{MP})$ can then be summarized as

$$z^*_M := \min \eta \tag{14}$$

$$\text{s.t.}(3)-(6) \tag{15}$$

$$\eta \geq \sum_{u\in\mathscr{U},p\in\mathscr{P}} \pi^{\text{var}}_{u,p,c} y_{u,p} + \pi^{\text{con}}_c \quad \forall c \in \mathscr{O} \tag{16}$$

$$\eta \geq 0 \tag{17}$$

where set $\mathscr{O}$ is the cut set of Benders optimality cuts. The master problem is a MILP.

For a given cut set $\mathscr{O}$, the master problem yields a lower bound on the optimal solution value, *that is*, $z^*_M \leq z^*$. In our case, the subproblem yields an upper bound for any trial $\hat{y}$, *that is*, $z^*_S(\hat{y}) \geq z^*$. If

$$z^*_S(\hat{y}) - z^*_M \leq \varepsilon,$$

then an ε-optimal solution to $(\mathcal{M})$ has been computed; for some $\varepsilon > 0$. Otherwise, an optimality cut is generated for trial $\hat{y}$ and the master problem is resolved.

4.1.2 Direct Cut Calculation.

The idea of Benders decomposition is to exploit the special structure of the original problem $(\mathcal{M})$. Sometimes, the subproblem also exhibits a special structure which may allow a more efficient solution than solving it as an LP. For example, the stochastic minimum $s-t$ cut problem features a maximum flow subproblem (Rebennack et al. 2020). Similarly, the nonlinear (but convex) subproblem of the power system expansion problem in Lohmann and Rebennack (2017) is a market clearing mechanism which can be solved efficiently by a tailored sorting algorithm. Another example arises in a stochastic production-inventory planning problem where the dual subproblems are solved using some ordering property (Golari et al. 2017). An entire class of Benders algorithms exploiting the special structure of the subproblems is logic-based Benders decomposition (Naderi et al. 2021). In our case, the subproblem $(\mathcal{S})$, and its dual, can also be solved very efficiently.

For subproblem $(\mathcal{S})$, it suffices to obtain an optimal dual (basic) solution $\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'}$. With such a dual point, we can calculate the cut coefficient $\pi^{\text{var}}_{u,p,c}$, the cut constant π^{con}_c and the optimal objective function value $z^*_S(\hat{y})$. Specifically, we do not require the solution of the primal subproblem $(\mathcal{S})$.

The dual subproblem is a continuous knapsack problem of special form. For trial $\hat{y}$, observe that an optimal dual (basic) solution is given by

$$\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'} = \begin{cases} 1, & \text{if}\,(u,u',\hat{u},\hat{u}') = \text{argmax}_{(u,u\prime,\hat{u},\hat{u}\prime)\in\mathcal{Q}}\left\{(\hat{y}_{u,p}+\hat{y}_{u\prime,p'}+\hat{y}_{\hat{u},\hat{p}}+\hat{y}_{\hat{u}\prime,\hat{p}'}-3)\cdot P_{F_{u,u\prime},F_{\hat{u},\hat{u}\prime}}\right\}>0 \\ 0, & \text{o/w} \end{cases} \tag{18}$$

for all $(u,u',\hat{u},\hat{u}')\in\mathcal{Q}$ and $(p,p',\hat{p},\hat{p}')\in\mathcal{C}$. The maximum is positive, if and only if there is a tuple $(u,u',\hat{u},\hat{u}')\in\mathcal{Q}$ of units which are placed on positions $(p,p',\hat{p},\hat{p}')\in\mathcal{C}$.

Assume that $\hat{y}$ is binary. Because at most one unit u can be placed at any position p, this tuple $(u,u',\hat{u},\hat{u}')$ must be unique, if it exists. Consequently, for binary values of $\hat{y}$, argmax is unique (implying that the solution is basic) and can be computed by inspecting all tuples $(u,u',\hat{u},\hat{u}')$. Thus, $\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'}=1$ if and only if units $(u,u',\hat{u},\hat{u}')\in\mathcal{Q}$ cause an overlap for positions $(p,p',\hat{p},\hat{p}')\in\mathcal{C}$. These overlaps can be computed efficiently by inspecting whether the tuples $(u,u',\hat{u},\hat{u}')$ cause an overlap. This yields a runtime of $O(|\mathcal{Q}|)$ to identify the 1 entries for $\pi^*_{p,p',\hat{p},\hat{p}',\cdot,\cdot,\cdot,\cdot}$. The cut coefficients $\pi^{\text{var}}_{u,p,c}$ can also be calculated in $O(|\mathcal{Q}|)$ through (13). This is extremely efficient when observing that subproblem $(\mathcal{S})$ contains $|\mathcal{Q}|\cdot|\mathcal{C}|$ functional constraints.

In case $\hat{y}$ is fractional, the argmax in (18) may contain more than a single value >0. Again, it suffices to run through all tuples $(u,u',\hat{u},\hat{u}')\in\mathcal{Q}$, retrieve the corresponding position tuples $(p,p',\hat{p},\hat{p}')$, check if they induce a crossing (*that is*, check if $(p,p',\hat{p},\hat{p}')\in\mathcal{C}$) and save them if necessary for later computation of

$$(y_{u,p}+y_{u',p'}+y_{\hat{u},\hat{p}}+y_{\hat{u}',\hat{p}'}-3)\cdot P_{F_{u,u'},F_{\hat{u},\hat{u}'}}$$

and to choose the maximum value among them.

4.1.3 Benders Based Branch-and-Cut.

Next to the efficient calculation of the cut coefficients, the particular implementation of the Benders decomposition can have a significant effect on the algorithm's computational performance. We chose a state-of-the-art implementation based on lazy constraints using callback functions. For that, observe that a valid Benders optimality cut is associated with each trial $\hat{y}$, not necessarily being binary. This allows us to compute and add Benders optimality cut(s) each time a trial has been computed in the branch-and-bound tree. The resulting algorithm is also known in the literature as "Benders based branch-and-cut" (Naoum-Sawaya and Elhedhli 2013).

We follow a strategy similar to Adulyasak et al. (2015) and add Benders cuts in two cases: (1) at the root node of the branch-and-bound tree, these are global cuts, and (2) for feasible (incumbent) solutions, these are local cuts. This strategy avoids computing an excessively large number of Benders cuts. Specifically, we add Benders optimality cuts in the root node until

$$z^*_S(\hat{y}) - z^*_{M,\text{LP}} \le \varepsilon$$

for $\hat{y}$ of the master problem, where $z^*_{M,\text{LP}}$ is the optimal objective function value of the LP relaxation of the master problem $(\mathscr{MP})$. This ensures that

$$z^*_{M,\text{LP}} - z^*_{\text{LP}} \le \varepsilon, \tag{19}$$

with optimal objective function value z^*_{LP} of the LP relaxation of the monolith $(\mathcal{M})$. Condition (19) certifies that we have solved the LP relaxation of the monolith with the Benders decomposition approach before the branching starts. The solution of the LP relaxation of $(\mathcal{M})$ via Benders decomposition is efficient because the master problem works only with the $y_{u,p}$ variables, avoiding the excessively large number of variables and constraints in (2). Additional efficiency is gained by using the direct cut calculation via (18) instead of solving LPs in order to compute the optimality cuts.

4.1.4 Strengthening the Master Problem. Inspired by Rahmaniani et al. (2017), we strengthen the master problem $(\mathscr{MP})$ guided by the LP relaxation as follows. We first solve the LP relaxation of the monolith with the Benders decomposition approach as described in section 4.1.3 using only the Benders optimality cuts (16) and the continuous $y_{u,p}$ and η variables. Once the LP relaxation has been solved to optimality, we inspect all computed Benders cuts. Specifically, we collect all occurring tuples $(p, p', \hat{p}, \hat{p}', u, u', \hat{u}, \hat{u}')$ in a set $\mathcal{T}^{pu} \subset \mathcal{C} \times \mathcal{Q}$, *that is*, we collect those tuples with a positive dual variable $\pi^*_{p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}'} > 0$. Next, we project $\mathcal{T}^{pu}$ onto the "position-space," *that is*, we define

$$\mathcal{T}^p := \{(p,p',\hat{p},\hat{p}') \in \mathcal{C} \mid \exists (u,u',\hat{u},\hat{u}') \in \mathcal{Q} :$$
$$(p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}') \in \mathcal{T}^{pu}\} \subset \mathcal{C}.$$

Note that for most instances, $\mathcal{T}^p$ contains much fewer elements than $\mathcal{C}$.

This allows us to define the strengthened master problem $(\mathscr{MP}^\star)$, where besides some initial Benders optimality cuts (16), we have also integrated the additional cuts (22).

$$z^*_{M^*} := \min \eta \tag{20}$$

$$\text{s.t.}(15) - (17) \tag{21}$$

$$\eta \ge \sum_{(p,p',\hat{p},\hat{p}') \in \mathcal{T}^p} \tau_{p,p',\hat{p},\hat{p}'} \tag{22}$$

$$\tau_{p,p',\hat{p},\hat{p}'} \ge \pi^{\text{var}}_{u,p,c} y_{u,p} + \pi^{\text{con}}_c \quad \forall (p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}') \in \mathcal{T}^{pu}, c \in \mathcal{O} \tag{23}$$

$$\tau_{p,p',\hat{p},\hat{p}'} \ge 0 \quad \forall (p,p',\hat{p},\hat{p}') \in \mathcal{T}^p. \tag{24}$$

Remark 6. Note that the cuts in (22) are valid cuts and, at the same time, stronger than the related Benders cuts. To see that, observe that a Benders cut is constituted by the sum over all tuples of units and positions that cause crossings. Cuts in (22) separate these sums onto multiple cuts chosen by the tuple of overlapping positions, implying that these cuts are valid and stronger. This comes at the cost that we have to introduce an additional variable for each cut of type (22) whereas Benders cuts are not enlarging the space of decision variables.

While $(\mathscr{MP}^\star)$ might look similar to the monolithic model $(\mathcal{M})$, the key aspect in the formulation of $(\mathscr{MP}^\star)$ is that considerably fewer variables and constraints of type (2) are needed. Thus, $(\mathscr{MP}^\star)$ is a hybrid version that combines the strength of the monolithic formulation with the nimbleness of its Benders decomposition. This way, the Benders algorithm resembles the delayed row generation approach in the restricted set $\mathcal{T}^p$.

Starting from a feasible solution $\bar{y}$ of $(\mathscr{MP}^\star)$ we suggest the following procedure:

1. Compute the tuples $(p, p', \hat{p}, \hat{p}', u, u', \hat{u}, \hat{u}')$ induced by all crossings of $\bar{y}$ and check whether we have $(p, p', \hat{p}, \hat{p}') \in \mathcal{T}^p$.
2. If $(p, p', \hat{p}, \hat{p}') \notin \mathcal{T}^p$, add a Benders cut of type (16) over all $(p,p',\hat{p},\hat{p}',u,u',\hat{u},\hat{u}')$ with $(p,p',\hat{p},\hat{p}') \notin \mathcal{T}^p$.
3. If $(p, p', \hat{p}, \hat{p}') \in \mathcal{T}^p$, add cuts of type (22) for all $(p, p', \hat{p}, \hat{p}', u, u', \hat{u}, \hat{u}')$ with $(p, p', \hat{p}, \hat{p}') \in \mathcal{T}^p$.

4.1.5 The Tailored Benders Based Branch-and-Cut Algorithm. The resulting algorithm is summarized in pseudo-code in Algorithm 1. The algorithm is initialized in step 1. Because the master problem $(\mathscr{MP})$ is unbounded for empty cut set $\mathcal{O}$, we set an artificial lower bound on η. This lower bound can be any value ≤ 0 because 0 is a trivial lower bound of the LP relaxation of $(\mathcal{M})$.

Algorithm 1 Benders based branch-and-cut with direct cut calculation and strengthened master problem (BDC)

```
Input: Problem instance, tolerance $\varepsilon > 0$
Output: $\varepsilon$-optimal decision $y^*$ and its objective function value
1: Set lower bound $\eta \geq 0$, $\mathcal{O} = \emptyset$, $c = 1$, $z_S^*(\cdot) = +\infty$, $z_{\text{best}} = +\infty$.
2: // Solve LP relaxation of $(\mathcal{M})$ through Benders decomposition:
   repeat
3:     Solve LP relaxation of master problem $(\mathcal{M})$. Obtain $z_{M,\text{LP}}^*$ and trial $\hat{y}$.
4:     Calculate "true" objective function value $z_S^*(\hat{y})$ for trial $\hat{y}$ via (18).
5:     If $z_S^*(\hat{y}) - z_{M,\text{LP}}^* \leq \varepsilon$, then GoTo 7.
6:     // Add Benders optimality cut:
       calculate cut coefficient $\pi_{u,p,c}^{\text{var}}$, cut constant $\pi_c^{\text{con}}$ and optimal objective function value $z_S^*(\hat{y})$
       via (18); $\mathcal{O} \leftarrow \mathcal{O} \cup \{c\}$; $c \leftarrow c + 1$.
7: // Check optimality of LP relaxation:
   If trial $\hat{y}$ is integral, then $y_{\text{best}} = \hat{y}$, $z_{\text{best}} = z_S^*(\hat{y})$ and GoTo 21.
8: // Prepare the set for the strengthened master problem cuts:
   Compute sets $\mathcal{T}^{pu}$ and $\mathcal{T}^p$.
9: // Branching:
   repeat
10:    Continue to solve master problem $(\mathcal{M})$ via branch-and-cut until feasible point has been
       computed or until MIP-gap $\leq \varepsilon$.
11:    If MIP-gap $\leq \varepsilon$: GoTo 21. O/w let $\hat{y}$ be the incumbent with objective function value $z(\hat{y})$.
12:    Calculate "true" objective function value $z_S^*(\hat{y})$ for trial $\hat{y}$ via (18).
13:    If $z_S^*(\hat{y}) - z(\hat{y}) \leq \varepsilon$, then
14:        Update solution: If $z(\hat{y}) < z_{\text{best}}$, then $y_{\text{best}} = \hat{y}$ and $z_{\text{best}} = z_S^*(\hat{y})$
15:        Accept incumbent and fathom node in branch-and-bound tree; GoTo 10.
16:    // Prepare Benders optimality cut (for local branching node only):
       calculate cut coefficient $\pi_{u,p,c}^{\text{var}}$ and cut constant $\pi_c^{\text{con}}$ via (18).
17:    Create a partition of the set of all tuples $(p, p', \hat{p}, \hat{p}', u, u', \hat{u}, \hat{u}')$ induced by all crossings of
       $\bar{y}$ based on whether we have $(p, p', \hat{p}, \hat{p}') \in \mathcal{T}^p$ or $(p, p', \hat{p}, \hat{p}') \notin \mathcal{T}^p$.
18:    Add a Benders cut of type (16) over all $(p, p', \hat{p}, \hat{p}', u, u', \hat{u}, \hat{u}')$ with $(p, p', \hat{p}, \hat{p}') \notin \mathcal{T}^p$.
19:    Add cuts of type (22) for all $(p, p', \hat{p}, \hat{p}', u, u', \hat{u}, \hat{u}')$ with $(p, p', \hat{p}, \hat{p}') \in \mathcal{T}^p$.
20:    $\mathcal{O} \leftarrow \mathcal{O} \cup \{c\}$; $c \leftarrow c + 1$.
21: return $y^* = y_{\text{best}}$ and $z_{\text{best}}$.
```

In the first "repeat" loop (steps 2–6), the LP relaxation of the monolith $(\mathcal{M})$ is solved through Benders decomposition (McDaniel and Devine 1977). Every trial solution from the master problem (step 3) is evaluated to obtain its "true" objective function value $z_S^*(\hat{y})$ (step 4). If this objective function value matches the objective function value of the master problem, then the LP relaxation has been solved and the "repeat" loop is left (step 5). Otherwise, a Benders cut for the fractional-valued trial $\hat{y}$ is added (step 6). Step 7 checks integrality of the computed incumbent $\hat{y}$; integrality implies that $\hat{y}$ is an optimal solution of the monolith $(\mathcal{M})$. Step 8 prepares the sets required for the strengthened Benders master problem cuts (22).

In the second "repeat" loop (steps 9–20), the master problem is solved to optimality by adding Benders cuts until all integral trial solutions are evaluated with the correct objective function value. Therefore, the master problem is solved (step 10) either until optimality (then the algorithm terminates in step 11) or until an integral trial solution $\hat{y}$ has been computed. If the objective function value of this trial solution is evaluated correctly in the master problem, then the solution is checked for possible update (step 14) and the corresponding node in the branch-and-bound tree is fathomed (step 15). Otherwise, Benders optimality cuts (16) or (22) are added (step 18 and 19, respectively).

4.2. The Dynamic Fix-and-Relax Pump (DFRP)

To improve the best feasible point found by BDC, we introduce the so-called *Dynamic Fix-and-Relax Pump (DFRP)*. DFRP is inspired by other numerical methods in large-scale optimization such as block coordinate descent (Hildreth 1957, Wright 2015) and stochastic gradient descent (Bottou 2010, Robbins and Monro 1951) where, due to the model size, not all the information available is used in each iteration. We also borrow ideas from the fix-and-relax heuristic, which successively improves a feasible solution by fixing parts of the variables and optimizing the resulting problem only in the free remaining variables (Toledo et al. 2015). Because ECW2CN does not exhibit a sequential decision-making process which can be exploited algorithmically, the proposed method differs from rolling horizon approaches in that any decision variables can be freed or fixed throughout the algorithm (Thevenin et al. 2021).

Based on a feasible solution $\bar{y}$, we select randomly a batch of units $B \subseteq \mathcal{U}$. For these units, an optimal positioning is computed using BDC. The batch size of units to be optimized is controlled in a dynamic way: If we see the same optimal value for some iterations, the batch size is increased. Once we reach a batch size, such that the corresponding problem cannot be solved to optimality within the time limit, we reset the sample size to an initial value. DFRP is stated explicitly in Algorithm 2.

Algorithm 2 Dynamic Fix-and-Relax Pump (DFRP)

```
Input: Problem instance, a feasible point $\bar{y}$, parameters $b_{init}, b_{+}, s_{\max} \in \mathbb{N}$
Output: A feasible point of $(\mathcal{M})$ and its objective value
1: $b \leftarrow b_{init}$, $y^{\text{best}} \leftarrow \bar{y}$, $z_{best} \leftarrow z(\bar{y})$, $iter \leftarrow 0$, $s \leftarrow 0$
   repeat until a stopping criterion is met:
2: Pick a random batch of units $B \subseteq \mathcal{U}$ with $|B| = b$.
3: Fix all variables whose units are not in $B$, i.e., set $y_{u,p} = y^{\text{best}}_{u,p}$ for $p \in \mathcal{P}$ if $u \notin B$.
4: Solve the partially fixed problem instance for all variables $y_{u,p}$ with $u \in B$ using Algorithm 1
   and retrieve its objective value $z'$ and optimal point $y'$.
5: If the optimization problem in 4. could not be solved to optimality, reset the batch size, i.e.,
   set $b \leftarrow b_{init}$
6: if $z' < z_{\text{best}}$:
     $z_{\text{best}} \leftarrow z'$, $y^{\text{best}} \leftarrow y'$
   elif $z' == z_{\text{best}}$:
     $s \leftarrow s + 1$
7: If $s == s_{\max}$:
     $b \leftarrow b + b_{+}$, $s \leftarrow 0$
8: return $y^{\text{best}}$ and $z_{\text{best}}$.
```

4.2.1 Stopping Criteria and Deterministic Variations of DFRP. We want to discuss some stopping criteria and alternative versions of DFRP:

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

- A maximal number of iterations, a time limit or a maximal number of iterations without changes in the objective value are classical stopping criteria which, of course, may also be used for DFRP.
- Note that DFRP computes a global optimal point of $(\mathcal{M})$, if we have $z_{\text{best}} = 0$ or if the problem in step 4 of Algorithm 2 is solved to optimality for $|B| = |\mathscr{U}|$.
- One deterministic variation of the DFRP is to drop step 5 if Algorithm 2 and to increase the batch sizes until $|B| = |\mathscr{U}|$. This variation can also be seen as warm-started BDC, preceded by a "trajectory" of feasible points generated by DFRP.

5. Solution of the Real-World Problem at BASF

5.1. Problem Specifications

In this use case, we have 75 units with 375 connections that have to be located on 76 positions. The maximal tube length permits connecting two units on the same side with a distance of 20 positions and two units on opposing sides whose distance does not exceed 18 positions. The maximal curvature condition translates into the constraint that connected units may not be located on neighbored positions.

5.2. Implementation Details

All computations are run on an Intel Xeon Processor with 4 cores at 3.7 GHz and 128 GB RAM. The code was implemented using Python 3.7 and GuRoBi 9.1 using the following GuRoBi methods/attributes:

- `GRB.Callback.MIPSOL`: This attribute indicates the occurrence of a new incumbent. If a new feasible point was computed, we add a new optimality cut with `Model.cbLazy()`.
- `Model.cbLazy()` which adds the cuts in a lazy manner.
- The attribute `GRB.Callback.MIPNODE` can be used to get an entry points for actions that are triggered by the computation of new feasible points of the continuous relaxation, for example, for adding fractional cuts in the strengthening step of the master problem.
- `GRB.Callback.MIPNODE_NODCNT` and `master.terminate()` are used to disrupt the solve routine after all fractional cuts are collected in the root node.

For Algorithm 2, we set $b_{init} = 6$, $b_{+} = 1$, $\varepsilon = 10^{-8}$ and $s_{max} = 5$ for all computations. The time limit for each problem within DFRP is set to 2 hours and we set the tolerable relative MIP gap to 1%.

5.3. DFRP Statistics

To strengthen the master problem as described in section 4.1.4, we collect 47,265 fractional cuts while solving the root node, *that is*, we have $|\mathcal{T}^p| = 47{,}265$. Based on the same initial solution, we run DFRP for each of the four objective functions for 48 hours. In each iteration of DFRP, an ECW2CN instance is solved for partially fixed units. The solution statistics are summarized for the different objective functions in Tables 1–4. The column "Sample Size" refers to the number of non-fixed units whose position is to determine optimally. "#Entries" is the number of instances for each row, *for example*, we have 25 iterations of DFRP for the objective function "min" with sample size 6. Column "Av. # Opt. Cuts" describes the average number of optimality cuts introduced for each problem instance in that row, whereas its average relative optimality is given in column "Av. Gap."

Table 1 Numerical Results for Objective Function "minimum"

Sample Size	# Entries	Av. # Opt. Cuts	Av. Gap	Av. CPU Time [s]	Av. # BB Nodes
6	25	428,843	0.0041	100.3	11.08
7	27	426,632	0.0034	61.6	13.37
8	29	515,654	0.0045	128.0	17.76
9	64	837,583	0.0061	138.6	44.59
10	41	1,974,287	0.0073	570.7	161.68
11	57	4,505,610	0.0079	873.3	439.19
12	30	6,621,153	0.0094	1820.4	815.70
13	8	11,584,127	0.0287	3955.1	2027.88

Table 2 Numerical Results for Objective Function "product"

Sample Size	# Entries	Av. # Opt. Cuts	Av. Gap	Av. CPU Time [s]	Av. # BB Nodes
6	24	306,481	0.0033	148.4	4.92
7	27	521,725	0.0038	307.1	24.67
8	29	600,600	0.0045	543.2	25.76
9	43	683,431	0.0035	525.7	34.35
10	16	2,086,097	0.0149	1572.5	184.62
11	10	2,577,913	0.0078	1724.9	226.30
12	38	3,032,248	0.0079	1825.7	313.95
13	4	3,582,362	0.0290	2542.0	781.50

Table 3 Numerical Results for Objective Function "one"

Sample Size	# Entries	Av. # Opt. Cuts	Av. Gap	Av. CPU Time [s]	Av. # BB Nodes
6	36	422,135	0.0034	361.1	15.00
7	24	835,595	0.0046	758.0	47.46
8	27	756,887	0.0047	877.9	37.63
9	43	1,290,207	0.0061	1396.5	77.67
10	22	1,837,311	0.0062	1603.9	131.55
11	9	3,165,046	0.0081	2572.3	252.33

Table 4 Numerical Results for Objective Function "sum"

Sample Size	# Entries	Av. # Opt. Cuts	Av. Gap	Av. CPU Time [s]	Av. # BB Nodes
6	45	373,946	0.0035	93.6	11.04
7	38	378,917	0.0022	91.1	10.97
8	48	452,494	0.0039	135.1	14.04
9	44	830,338	0.0047	221.3	44.32
10	75	1,442,428	0.0062	245.2	101.83
11	39	3,294,341	0.0081	990.6	324.33
12	35	2,753,871	0.0071	732.3	270.54
13	19	9,086,700	0.0183	3413.9	1453.05

Finally, "Av. CPU Time [s]" and "Av. # BB Nodes" describe the average CPU time in seconds and the number of searched branch-and-bound nodes, respectively.

In Tables 1–4 we observe that the sample sizes vary between 6 and 13 for most objective functions and we see that in general the number of optimality cuts increases with the sample size. The optimality gaps reported by GuRoBi are close to zero and exceed the given threshold of 1% only if the model cannot be solved within the time limit of 2 hours. As expected, the average computing time as well as the number of searched branch-and-bound nodes increases for larger sample sizes.

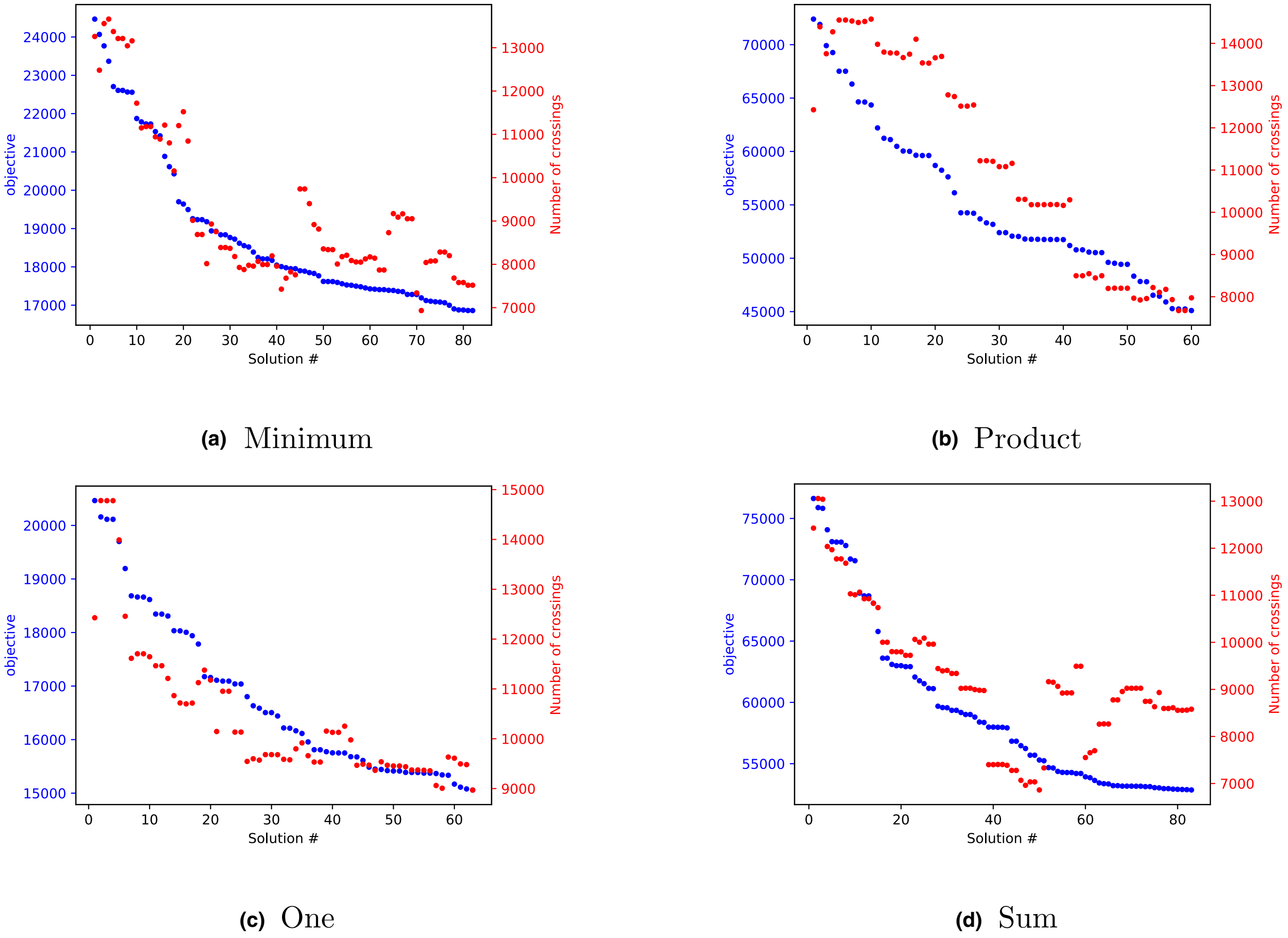
The "one" objective function seems to generate rather hard problem instances since we always faced a problem instance of sample size 11 that could not be solved within the time limit of 2 hours before increasing the sample size (*cf*. Table 3).

5.4. Results

Each objective functions leads between 60 and 83 feasible points. These points are then evaluated by a simulation model that counts the total number of crossings in a time dependent framework, based on the historic connections. The result is depicted in Figure 6.

Each of the four graphs in Figure 6 is generated as follows. For a specific objective function, all computed

Figure 6 For Each Feasible Point, the Objective Function Value on the Left Axis and the Number of Crossings as Output of the Simulation Model are Shown. Each Graph Corresponds to a Different Objective Function and a Different Run of Algorithm Dynamic Fix-and-Relax Pump [Color figure can be viewed at wileyonlinelibrary.com]



(a) Minimum

(b) Product

(c) One

(d) Sum

Table 5 Comparison of Relationship between Objective Function Value and Simulation Result

Objective function	Correlation	Rank Correlation
Minimum	0.91	0.67
Product	0.90	0.96
One	0.91	0.90
Sum	0.84	0.73

Table 6 Computation of Critical FIFO Crossings for the Best Seven Solutions Found by Dynamic Fix-and-Relax Pump (DFRP) Evaluated on a Time Period of Three Years

Instance	Number of FIFO crossings
Minimum - best_obj	17
Minimum - best_sim	8
Product - best_obj	10
Product - best_sim	**0**
One - best_obj and best_sim	26
Sum - best_obj	26
Sum - best_sim	18

feasible points by DRFP are sorted according to decreasing objective function value. This is shown by the blue dots. For each feasible point, the resulting crossings, as evaluated by the simulation model, are shown by the red dots. Therefore, in a perfect (*that is*, dynamic) model, the red dots would lie on the blue line. We observe that the two y-axis values live on different scales.

Next, we want to evaluate how good the obtained solutions ("blue dots") mimic the simulation results ("red dots"); therefore, consider Table 5. This table lists both the Pearson correlation and the Spearman's rank correlation of the objective function value to the simulation result. Such as the correlation, the rank correlation lies between −1 and +1 and a value close to +1 indicates strong dependence. While the correlation measures linear dependence, the rank correlation captures monotonic relationships, *that is*, only the ranks of the data points are compared and not their values. Therefore, the rank correlation is a more relevant measure for our data than the correlation because we are just interested which solutions perform best.

For the results in Figure 6, the "product" objective function value performs very strong for both the correlation and the rank correlation. The "sum" and "minimum" objectives perform well with respect to the correlation of objective values and simulation results but show weaknesses with respect to rank correlation. That can also be seen in Figure 6 since for both objectives there exist points with very good objective values but poor simulation performance. It is striking to see that also the "one" objective performs very well for both performance measures. However, while the results for "sum," "minimum," and "product" have been very similar in previous versions of the numerical results, "one" did behave much worse in these runs. So, the good performance of "one" might be a "lucky shot" caused by the stochastic nature of DFRP.

Finally, for each objective function, the best feasible points with respect to their objective value and the simulation result are evaluated with the second, much more detailed simulation model. Because for the "one" objective these points coincide, we computed the number of FIFO crossings for seven feasible points and ended up with the results in Table 6.

The best solution found without any FIFO crossings is illustrated in Figure 7.

The final solution was presented to the engineering department. Although we computed a fully feasible layout in the sense that no FIFO crossings occurred on historical connection data, we were aware of the fact that that might not hold true for futures connections. However, the best layout generated by our algorithm reduces the expected number of such events to a few per year, which was deemed to be acceptable by the engineers.

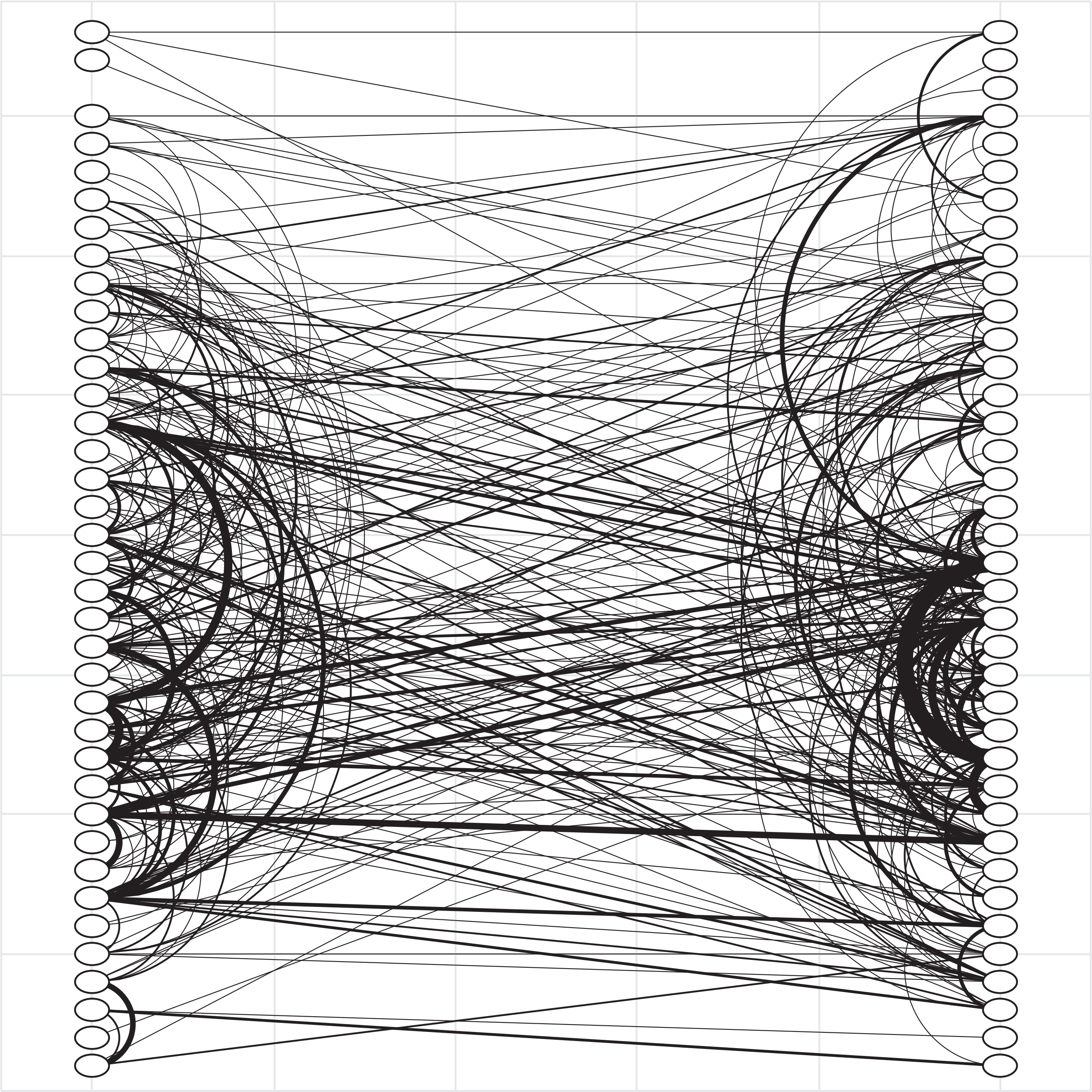
As a side effect, our models provided valuable insights for the engineers. In particular, the question of how many pipes of the different lengths will be needed was an important parameters for the dimensioning of the tube handling system. Also, more general questions, such as the effect of the width of the switching hub (the distance between the two lines of connectors) on other parameters, could be answered easily.

Our results removed the last concerns toward the applicability of the robotic invention, and a decision was made to build it. The robotic system will automatically prevent connections between wrong units, decrease the risk of spillage and increase the safety and satisfaction of the workforce.

6. Conclusions

In an effort to increase human safety, BASF is developing a dual-arm robotic system to reconnect pipes in a switching hub. The robot can undertake almost all reconnections, except such reconnections which require the removal of multiple pipes at ones, which is left to the humans. This raised the question on how to design the layout of a new switching hub in order to minimize the number of reconnections involving humans. We have modeled this real-world problem as a special minimal crossing number problem in conjunction with two simulation models. The resulting edge-constrained weighted two-layer crossing number problem (ECW2CN) leads to instances of gigantic size (approximately 70 billion constraints), when tackled via MILP approaches. Therefore, we have developed a tailored Benders decomposition which is

Figure 7 Best Solution with Regard to the Number of Critical FIFO Crossings Computed by Dynamic Fix-and-Relax Pump



employed in a dynamic fix-and-relax pump. For different objective functions, we have identified a total of 454 feasible layouts. These layouts are then evaluated by two different simulation models. The obtained solution not only increases human safety but also provides valuable feedback to engineers regarding the design of the switching hub.

We see three possible directions for future academic research. The first direction is to increase the computational efficiency of the tailored solution algorithms or to propose better performing models and methods. For Benders decomposition, there might be possibilities to further strengthen the master problem through tailored cuts. Another possibility might arise when combining column generation (on the τ-variables) with Benders decomposition. A second direction is to enhance the model by dynamic and stochastic aspects, capturing the entire complexity of the real-world problem. This comes at the cost that: (1) no real-world data are available, and (2) the model becomes even more challenging to solve. A third direction is to obtain better theoretic lower bounds

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441

through specific crossing number lemmas, taking the special structure of ECW2CN into account.

Acknowledgment

We thank Alfred Krause and his team for the interesting and fruitful collaboration which gave rise to the subject of this article. Furthermore, we thank Debora Morgenstern for handling the project during its first steps. Also, many thanks to Marcel S. Schmidt for assisting with early versions of the numerical computations and creating the visualizations. Finally, we thank Silke Horn, Sonja Mars and Kostja Siefen from GuRoBi for answering our questions about implementation details and Carsten Linz for supporting the project in an outstanding way.

References

Adulyasak, Y., J.-F. Cordeau, R. Jans. 2015. Benders decomposition for production routing under demand uncertainty. *Oper. Res.* **63**(4): 851–867.

Bachmaier, C., H. Buchner, M. Forster, S. H. Hong. 2010. Crossing minimization in extended level drawings of graphs. *Discrete Appl. Math.* **158**(3): 159–179.

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **4**(1): 238–252.

Bomze, I. M. 1997. Evolution towards the maximum clique. *J. Global Optim.* **10**(2): 143–164.

Bomze, I. M., M. Budinich, P. M. Pardalos, M. Pelillo. 1999. The maximum clique problem. In *Handbook of Combinatorial Optimization*, (pp. 1–74). Springer, Boston, MA.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010*. 177–186.

Buchheim, C., D. Ebner, M. Jünger, G. W. Klau, P. Mutzel, R. Weiskircher. 2005. Exact crossing minimization. In *International Symposium on Graph Drawing*, (pp. 37–48). Springer, Berlin, Heidelberg.

Buchheim, C., M. Chimani, D. Ebner, C. Gutwenger, M. Jünger, G. W. Klau, P. Mutzel, R. Weiskircher. 2008. A branch-and-cut approach to the crossing number problem. *Discrete Optim.* **5**(2): 373–388.

Buchheim, C., A. Wiegele, L. Zheng. 2010. Exact algorithms for the quadratic linear ordering problem. *INFORMS J. Comput.* **22**(1): 168–177.

Çakiroḡlu, O. A., C. Erten, Ö. Karataş, M. Sözdinler. 2009. Crossing minimization in weighted bipartite graphs. *J. Discrete Algorithms* **7**(4), 439–452.

Carpano, M. J. 1980. Automatic display of hierarchized graphs for computer-aided decision analysis. *IEEE Trans. Syst. Man Cy-s* **10**(11): 705–715.

Chimani, M., P. Mutzel, I. Bomze. 2008. A new approach to exact crossing minimization. *Algorithms - ESA 2008. Lect. Notes Comput. Sci.* **5193**: 284–296.

Chimani, M., P. Hungerländer, M. Jünger, P. Mutzel. 2012. An SDP approach to multi-level crossing minimization. *J. Exp. Algorithmics* **17**: 1–26.

Garey, M. R., D. S. Johnson. 1979. Computers and intractability, a guide to the theory of NP-completeness. V. Klee, eds. *A Series of Books in the Mathematical Sciences*. W. H. Freeman and Company, New York.

Garey, M. R., D. S. Johnson. 1983. Crossing number is NP-complete. *SIAM J. Algebraic Discrete Meth.* **4**(3): 312–316.

Garey, M. R., D. S. Johnson, L. Stockmeyer. 1974. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing* 47–63.

Golari, M., N. Fan, T. Jin. 2017. Multistage stochastic optimization for production-inventory planning with intermittent renewable energy. *Prod. Oper. Manag.* **26**(3): 409–425.

Hildreth, C. 1957. A quadratic programming procedure. *Nav. Res. Logist. Q.* **4**: 79–85.

Jünger, M., P. Mutzel. 1996a. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica* **16**(1): 33–59.

Jünger, M., P. Mutzel. 1996b. Exact and heuristic algorithms for 2-layer straightline crossing minimization. *International Symposium on Graph Drawing GD 1995: Lecture Notes in Computer Science* **1027**: 337–348.

Kobayashi, Y., H. Maruta, Y. Nakae, H. Tamak. 2014. A linear edge kernel for two-layer crossing minimization. *Theoretical Comput. Sci.* **554**: 74–81.

Laguna, M., R. Marti. 1999. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS J. Comput.* **11** (1): 44–52.

Lohmann, T., S. Rebennack. 2017. Tailored Benders decomposition for a long-term power expansion model with short-term demand response. *Management Sci.* **63**(6): 2027–2048.

Magnanti, T. L., R. T. Wong. 1981. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.* **29**(3): 464–484.

McDaniel, D., M. Devine. 1977. A modified Benders' partitioning algorithm for mixed integer programming. *Management Sci.* **24** (3): 312–319.

Naderi, B., V. Roshanaei, M. A. Begen, D. M. Aleman, D. R. Urbach. 2021. Increased surgical capacity without additional resources: Generalized operating room planning and scheduling. *Prod. Oper. Manag.* forthcoming.

Naoum-Sawaya, J., S. Elhedhli. 2013. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Ann. Oper. Res.* **210**(1): 33–55.

Palubeckis, G., A. Tomkevičius, A. Ostreika. 2019. Hybridizing simulated annealing with variable neighborhood search for bipartite graph crossing minimization. *Appl. Math. Comput.* **348**: 84–101.

Papadakos, N. 2008. Practical enhancements to the Magnanti-Wong method. *Oper. Res. Lett.* **36**(4): 444–449.

Rahmaniani, R., T. G. Crainic, M. Gendreau, W. Rei. 2017. The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.* **259**(3): 801–817.

Rebennack, S. 2016. Combining sampling-based and scenario-based nested Benders decomposition methods: Application to stochastic dual dynamic programming. *Math. Program.* **156**(1): 343–389.

Rebennack, S., M. Oswald, D. O. Theis, H. Seitz, G. Reinelt, P. M. Pardalos. 2011. A branch and cut solver for the maximum stable set problem. *J. Comb. Optim.* **21**(4): 434–457.

Rebennack, S., G. Reinelt, P. M. Pardalos. 2012. A tutorial on branch and cut algorithms for the maximum stable set problem. *Int. Trans. Oper. Res.* **19**(1-2): 161–199.

Rebennack, S., O. A. Prokopyev, B. Singh. 2020. Two-stage stochastic minimum s-t cut problems: Formulations, complexity and decomposition algorithms. *Networks* **75**: 235–258.

Robbins, H., S. Monro. 1951. A stochastic approximation method. *Ann. Math. Statist.*, **22**(3): 400–407.

Schaefer, M. 2020. The graph crossing number and its variants: A survey, 5th version. *Electron. J. Comb* **#DS21**, 1–126.

Schaefer, M., E. Sedgwick, D. Stefankovic. 2008. Computing Dehn Twists and Geometric Intersection Numbers in Polynomial

Time. In *Proceedings of the 20th Canadian Conference on Computational Geometry* 20, 111–114.

Shahrokhi, F., O. Sýkora, L. A. Székely, I. Vrto. 2001. On bipartite drawings and the linear arrangement problem. *SIAM J. Comput.* **30**(6), 1773–1789.

Sherali, H. D., B. J. Lunday. 2013. On generating maximal nondominated Benders cuts. *Ann. Oper. Res.* **210**(1): 57–72.

Sinden, F. W. 1966. Topology of thin film RC circuits. *Bell Syst. Tech. J.* **45**(9): 1639–1662.

Székely, L. A. 2004. A successful concept for measuring nonplanarity of graphs: The crossing number. *Discrete Math.*, **276** (1–3): 331–352.

Thevenin, S., Y. Adulyasak, J. Cordeau. 2021. Material requirements planning under demand uncertainty using stochastic optimization. *Prod. Oper. Manag.*, **30**(2): 475–493.

Toledo, C. F. M., M. da Silva Arantes, M. Y. B. Hossomi. 2015. A relax-and-fix with fix-and-optimize heuristic applied to multilevel lot-sizing problems. *J. Heurist.* **21**(5), 687–717.

Tullo, A. H. 2018. C&EN's global top 50 chemical companies. *Chem. Eng. News* **96**(31): 36–41.

Turán, P. 1977. A note of welcome. *J. Graph Theory* **1**(1): 7–9.

Wright, S.-J. 2015. Coordinate descent algorithms. *Math. Program.* **151**(1): 3–34.

Please Cite this article in press as: Sudermann-Merx, N., et al. Crossing Minimal Edge-Constrained Layout Planning using Benders Decomposition. *Production and Operations Management* (2021), https://doi.org/10.1111/poms.13441