Abstract

Constraints lead to statistical patterns in data. The initial step of this master thesis work is to quantify the characteristics of two hypothetical types of constraints in industrial production: technology-driven constraints and load-driven constraints. This will be achieved by analyzing the statistical properties of association networks over time in a large data set from steel manufacturing. Based on these results, an abstract theoretical framework will be developed to better understand the connection between each type of constraint and the statistical patterns created by it.

Acknowledgements

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and Motivation

introducing the different types of constraints in manufacturing life cycle.

1.2 Research Objective

A valid theoretical framework for discrimination of the two types of constraints in statistical properties of the production data

First, formulate the binning methods here because this will describe the hypothesis underlining my thesis.

Explanation of my hypothesis is a theoretical/conceptual framework as a starting point for the investigation. It is a well-defined object and based on facts.

Our hypothesis: different types of constraints create non-random features in the association networks for different binning schemes. Network derived from other type (FBS?) of binning. they show non-random features mostly when I have performance constraints.

Need to introduce different categories of constraints; technical constraints, performance-indicator based constraints in the context of FBA.

1.3 Research Plan and Thesis Organization

Methods are introduced here as indicative of two fundamentally different constraints acting on the manufacturing process: technological constraints on the one hand and constraints related to material flow and production capacity on the other.

2 Methodology

The initial step of this master thesis work was to quantify the characteristics of two hypothetical types of constraints in industrial production: technology-driven constraints and load-driven constraints.

I am planning to achieve this with an Operations Research Model consists of two steps. First, analyzing the statistical properties of association networks over Time in an extensive data set from steel manufacturing; second, developing an abstract theoretical framework to understand better the connection between each type of constraint and the statistical patterns created by them.

Introduce proposed concepts in the Operations Research Model (OR model).

The OR model is a combination of Steel Manufacturing Events Analysis and Flux Balance Analysis. The art form of this model is to structure a standard data format and a shared analysis logic that allows comparing the results from manufacturing data and simulation data.

A brief introduction for Association Networks and FBA. An explanation for generating a data structure with OR-modeling in the combination of those. More detailed information to be given in the Backgroud Information Section, guiding the readers who have knowledge of FBA and Association Network concepts to the Concept Implementation Section.

Usage of linear programming and generating sets of synthetic data allow comparing the statistical characteristics of their association network with those created from the real-world data set from steel manufacturing.

2.1 Steel Manufacturing Data Analysis

Investigation of constraints impact in time windows was performed by analyze in two different type of association networks; the networks with fixed step size nodes and the networks with fixed bucket size nodes.

those two different type of networks were applied in all 10 time windows and some relevant network metric plots were generated. Average degree vs time

windows Average betweenness centrality vs time windows Average modularity vs time windows

Association Networks

Beyond a simple network graph representation of historical production data, the formation of association networks is an insightful graph-based framework combining the tool: association rules and complex networks as Merten et al. (2020) performed in their article [1]. The relevant pipeline considers sequentially revealed events of a data set. It outputs a graph that unfolds the non-random occurrence of specific events together among the complete set that took place consecutively in the production period.

Assume we have a manufacturing data set with historical order, D, consists of k sequences and n events with mass values and sequence id's included as given in Table 2.1.

Event_ID	Mass	Sequence_ID
1	280	1
2	250	1
3	890	2
4	850	2
5	650	2
6	745	2
7	795	2
8	150	3
⋮	⋮	⋮
n-4	940	k-1
n-3	540	k
n-2	520	k
n-1	630	k
n	610	k

Table 2.1: Arbitrarily Created Data Set D.

Examining the data set, one can say that the events with mass values: 890, 850, 650, 745 or 540, 520, 630, 610 are close to each other; thus, they are produced together and likely occur in the identical sequences among the complete data. In a further step, one can label the events mentioned above with a value interval (the so-called binning size) typical for every mass value with a tiny difference to each other. Binning generation for the events allows us to investigate them in

a mass-production manner. Alternative binning methods will be addressed in the following subsection.

One can hypothetically argue that the information mentioned above patterns are probably deliberate planning choices based on the related constraints acting on the manufacturing process performance. However, forming prevailing arguments is not a simple task for large and complicated data sets. Such a real-life data set may consist of more than $300,000$ events likely to have various events aggregated randomly in its large sequence groups.

To distinguish random co-occurrences from meaningful ones in production sequences and assess the complexity of production patterns before creating the network graphs, we extract the association rule from the set of sequences. With a similar approach as Merten et al. (2020) applied [1], an association rule measure known as "Lift" was picked and calculated for every possible pairwise subset of events that occurred in the same sequences. By having a natural threshold of Lift measure 1. The lift can be computed by the ratio of pair items joint probability divided by the multiplication of each item's marginal probability as

$$Lift(A \leftrightarrow B) = \frac{P(A,B)}{P(A) * P(B)}, \tag{6}$$

thus, in the case of $Lift(A \leftrightarrow B) > 1$, B occurs likely if A occurs whereas $Lift(A \leftrightarrow B) < 1$, B unlikely occurs if A occurs. Indication of random and non-random co-occurrences as 0 and 1 in an adjacency matrix will provide the data structure to form a network as shown in Fig. 2.1.

Events	1	2	3	...	n-1	n
1	0	1	0	...	0	0
2	1	0	0	...	0	0
3	0	0	0	...	0	0
⋮	⋮	⋮	⋮	⋱	⋮	⋮
n-1	0	0	0	...	0	1
n	0	0	0	...	1	0

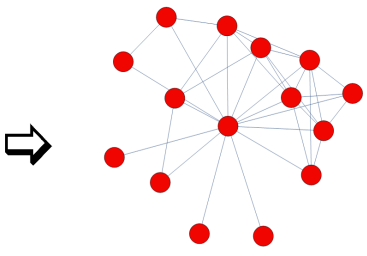
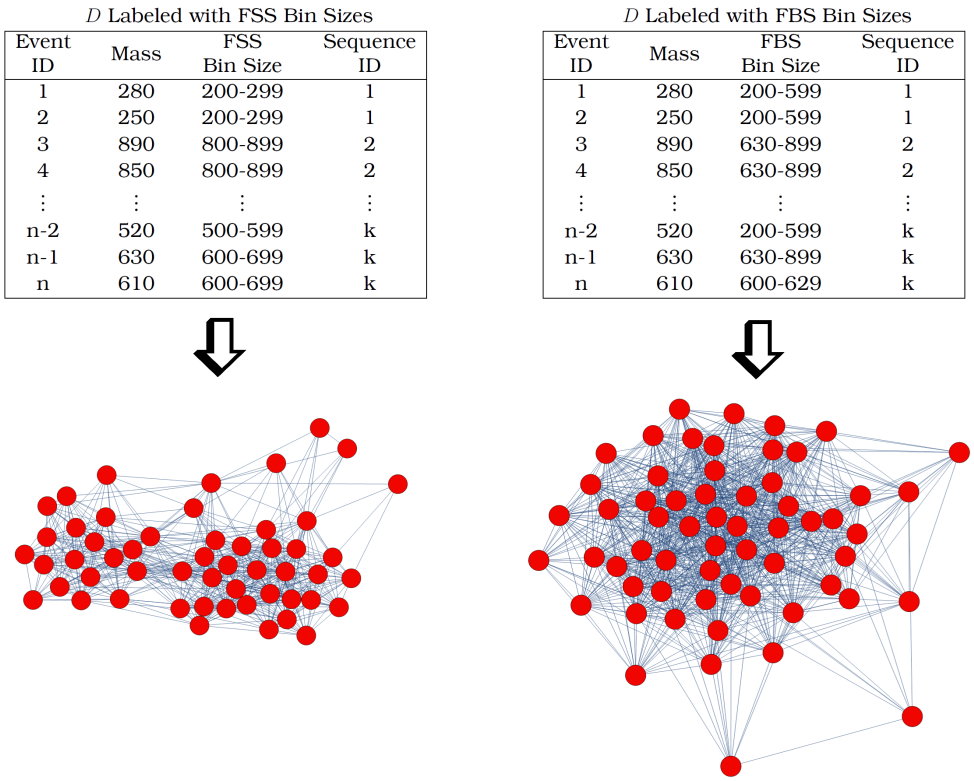


Figure 2.1: An Arbitrary Representation for Adjacency Matrix and Its Graph.

Binning Methods

Alternative binning methods for the production events underlie the developed hypothesis of this thesis work: Non-random features of the association networks derived from these two methods explained in the introduction part.

Two distinguished network generation approach can be derived considering the alternative binning tools mentioned above: Fixed Step Size network (FSSn); it has graph nodes as binning members with equal bin sizes, and Fixed Bucket Size network (FBSn); its nodes are binning members with an equal amount of events per bin. Forcing events to occur in the nodes with constant interval boundaries allows us to see how the aggregations take place within orders, whereas defining a typical bucket size for the network nodes results in arbitrary interval boundaries for each node; still, it allows to control their population.



D Labeled with FSS Bin Sizes

Event ID	Mass	FSS Bin Size	Sequence ID
1	280	200-299	1
2	250	200-299	1
3	890	800-899	2
4	850	800-899	2
⋮	⋮	⋮	⋮
n-2	520	500-599	k
n-1	630	600-699	k
n	610	600-699	k

D Labeled with FBS Bin Sizes

Event ID	Mass	FBS Bin Size	Sequence ID
1	280	200-599	1
2	250	200-599	1
3	890	630-899	2
4	850	630-899	2
⋮	⋮	⋮	⋮
n-2	520	200-599	k
n-1	630	630-899	k
n	610	600-629	k

Figure 2.2: Graph Results For Two Different Network Approaches.

Network Metrics Analysis

Modularity

modularity equations
$Q = \frac{1}{4m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\ s_i s_j$
$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$
$Q = \frac{1}{4m} s^T B s = \frac{1}{4m}\sum_{i=1}^{n}(u_i^T . s)^2 \beta_i$
$\Delta Q = \frac{1}{4m} s^T B^{(g)} s$
$B_{ij}^{(g)} = B_{ij} - \delta_{ij}\sum_{k\in g} B_{ik}$

It is important to discuss how randomization has been done since the plot results can vary based on the generated null model via that specific randomization method. Random graphs were generated and below instruments were plotted.

Average degree vs time windows Average betweenness centrality vs time windows Modularity (Wolfram method) vs time windows Modularity (GN method, algorithm by me) vs time windows Average modularity for single random graph vs time windows Z-scores for GN-modularity with Erdös-Renyi randomized null models vs time windows Z-scores for GN-modularity randomized null models with fixed degree sequences vs time windows

Performing switch-randomization to a modular graph might fail even due to small details in randomization steps. That failure is probably the reason of high values of Z-scores in two different plots of Z-scores. If I try to switch randomize a modular graph, I could imagine a procedure where I take links only from same module and switch them or links that are across modules and switch them. And I mixed the sets of intra-module edges and inter-module edges separately. This null model might give a different result in Z-scores.

before treating time windows, having the modularity as function of bucket size and as function of step size. At this stage, choosing suitable step and bucket sizes and accordingly repeat all progress mentioned above. The aim is to obtain big amount of nodes as possible as we can and keeping that amount of nodes same in both graph structures (fixed step size and fixed bucket size). The difference between modularity values at highest graph nodes amount in fixed bucket and fixed step sized graphs shows which network structure is more effective on generating clear communities. In other words, modularity values is more meaningful when the node number is high.

Different Types of Null Model

Fixed Degree Sequence random graphs generation, a third one: Conserving Inter-edges and Intra-edges Among Modules random graphs were generated,

accordingly, Z-scores were computed among those null models.

I have recomputed my constraint impact analysis pipeline for four production lines with re-defined null models: Degrees Fixed and Modularity and different choice of binning in terms of step-size & bucket-size. The results are given in the attached PNG file.

Clarification of some details in the plot results as, • Please find below the cartoon showing how I generated the null models in a pairwise shuffling fashion to prevent ambiguity.

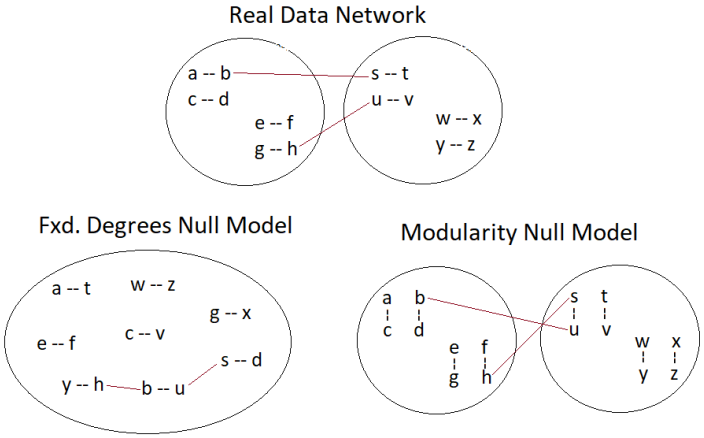


Figure 2.3: Formation of Different Null Models.

2.2 Flux Balance Analysis

The genome-scale integrated networks are necessary tools used by metabolic engineers on model generation, theoretical and computational analysis for microbial organisms. In addition, the network theory tools expand the feasible space for the following analysis techniques in the field.

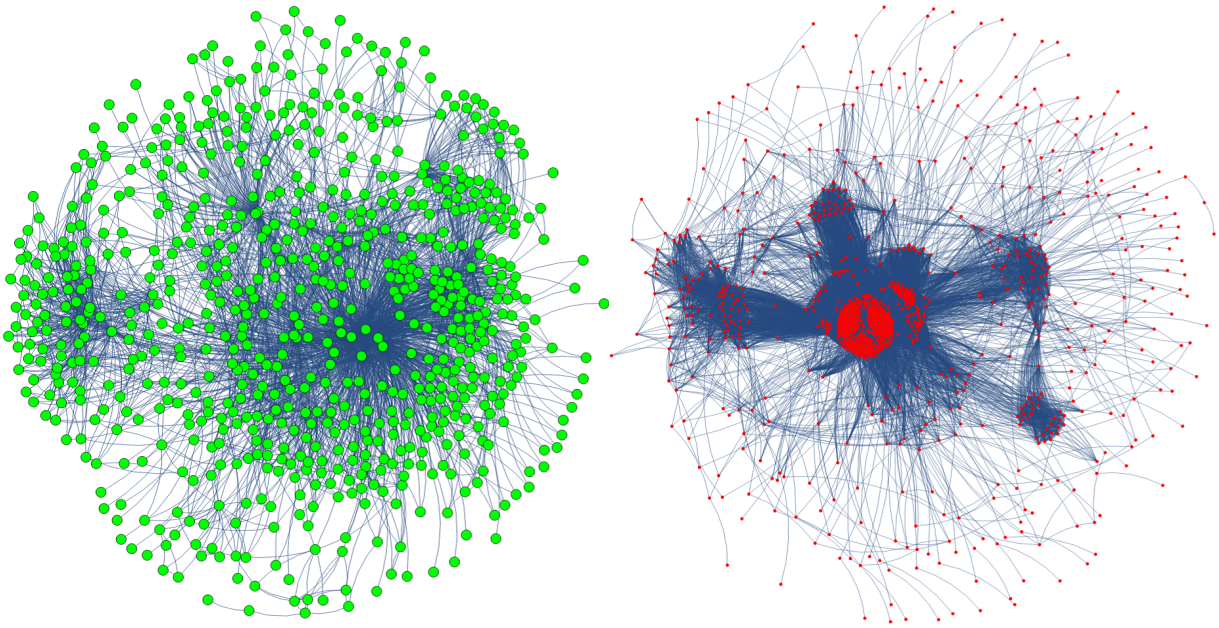
Introducing stoic. matrix. Explain how the two graphs are formally obtained by manipulated the stoichiometric matrix, explain the metabolite-centric network is $S * S^T$ and binarized. In contrast, the reaction-centric network is $S^T * S$ and binarized. Introduce the general idea of FBA as an optimization scheme in a steady-state solution space. Although the networks shown in Fig. 2.4 do not contain any information about directionality or effectiveness of the reactions to the system, the set of rules take place in networks can be represented in more detail and stoichiometrically by an m-by-r matrix formulation (the so-called Stoichiometric Matrix S), whereas its column elements represent reactions that play a role in the chemical transformation, and its row elements represent metabolites as

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1r} \\ s_{21} & s_{22} & \dots & s_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \dots & s_{mr} \end{bmatrix} = (s_{ij}) \in \mathbb{Z}^{mxr}, \tag{1}$$

Fig. 2.4 shows two differently constructed networks showing interactions between metabolites, intermediate or end products and metabolic reactions for a particular metabolism: Homo Sapien. In Fig. 2.4a the graph nodes stand for the metabolites, graph edges are the reactions. In contrast, in Fig. 2.4b the roles are reversed so that the graph edges represent the metabolites, and the graph nodes represent the reactions.

Studying biological metabolic systems, generated models to achieve cellular objectives like cell growth or ATP production brings the necessity of various tools to analyze reconstructed genome-scale networks. [2, 3]. One of the commonly used tools is Flux Balance Analysis (FBA). It is a constraint-based modelling approach to simulate microbial metabolisms and can be applied to biochemical-reaction networks containing the chemical transformations and flux exchanges in that particular network [4, 5].

while one can express the fluxes in a one-dimensional array (the so-called Flux



(a) Metabolic-centric Network

(b) Reaction-centric Network

Figure 2.4: Network Representations for Homo Sapiens Metabolic Model

Vector V) as

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix} = (v_i) \in \mathbb{R}. \tag{2}$$

V contains flux exchange values for the corresponding reactions in the system and gives information about the flux distribution; hence, those can be both positive and negative real numbers. Definition of a mass balance ($S.V = 0$) constraint in the FBA enables us to analyze the metabolic network operations in a steady-state [4, 5].

$$S.V = \begin{bmatrix} s_{11}v_1 + s_{12}v_2 + \cdots + s_{1r}v_r \\ s_{21}v_1 + s_{22}v_2 + \cdots + s_{2r}v_r \\ \vdots \\ s_{m1}v_1 + s_{m2}v_2 + \cdots + s_{mr}v_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3}$$

The higher amount of metabolite consideration in the set of rules, S, in other words, the larger matrix size by its rows amount means the more complex organization structure taken into account while preserving the steady-state in the whole system.

More than one steady-state solution might be present since it is impossible to identify all constraints in a cellular system [4]. Therefore, one can formulate an optimization approach to identify reaction network steady-states that maximize the biomass [4, 5] or control the production of specific metabolites [6] within a defined objective function under the consideration of the system constraints. According to Price et al. (2004), there are three primary purposes to generate objective functions: to discover allowable characteristic properties in the genome-scale network reconstruction; to mimic probable physiological functions like biomass or ATP production to be able to determine likely physiological states; and lastly, to design a genetic variant or sub-type to obtain a desired particular product [5].

The objective function can be thought as a production plan that gives an idea about the diversity of products that the relevant system can produce, and one can express its coefficients in a one-dimensional array as

$$O = \begin{bmatrix} o_1 & o_2 & \dots & o_r \end{bmatrix} = (o_i) \in \mathbb{R}. \tag{4}$$

As given in Eq.(5), the Objective Function, Z, rules the maximized output based on its non-zero coefficients, which are the decisive ones for the flux elements of V to be considered.

$$Z = O.V = (o_1 v_1 + o_2 v_2 + \cdots + o_r v_r) \in \mathbb{R}_{\geq 0}. \tag{5}$$

Stoichiometry and mass-balance are the constraints introduced so far in Eq.(1) and Eq.(3). In addition, upper and lower bounds are introduced for particular fluxes in V during the optimization process. The bounds are used in the reactions for uptake and secretion of any organic metabolite. In the uptake reactions, nutrients are transported to the inside of the metabolic network. In the secretion reactions, products are exported to the outside of the network. The rest of the fluxes in V are used in the exchange reactions, namely the intermediate reactions in the network. The constraints are decisive on the reactions for uptake and secretion, whereas no limitation is considered in the exchange reactions. Quantification of imported nutrients and exported outputs (the so-called Resources and Wastes) by constraining them with upper and lower bounds to fulfil a single objective function goal might play a significant role in the optimization process.

The above-explained optimization process is a linear programming problem since the mass balance (Eq.(3)), the Objective Function (Eq.(5)), and linear equations formulate the upper & lower bounds for fluxes. The linear optimization result maximizes the structured Objective Function in the form of a flux distribution [4, 5]. Since each term in Eq.(5) is a produced biomass expression for the fluxes, the summation of those terms will give the overall growth of the system for a single network state.

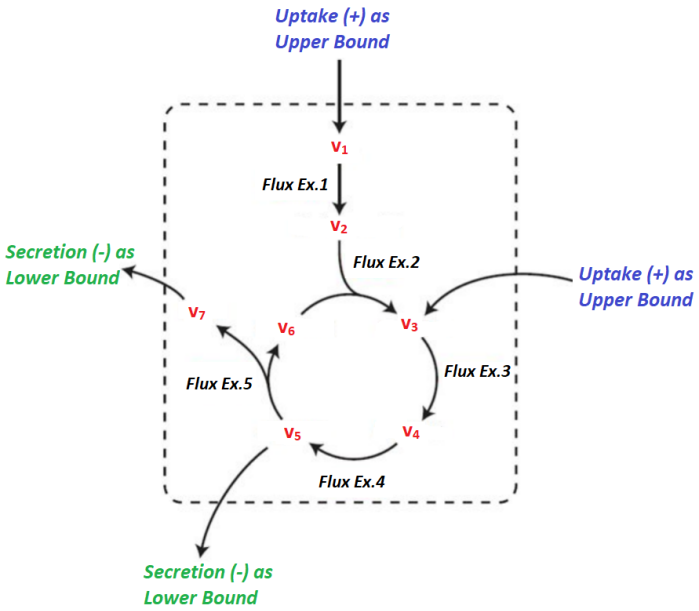


Figure 2.5: A Simplified Reaction-centric Network Sketch Shows The Reactions for Exchange, Uptake and Secretion.

Fluxes for Uptake and Secrete Reactions

Let

$$V^* = (v_1^*, v_2^*, \ldots, v_x^*) = (a_i \leq v_i^* \leq b_i) \in V \tag{7}$$

is a set of fluxes picked from V to be limited with the bounds: a_i and b_i which are used in the reactions for uptake and secretion as previously introduced.

Upper and Lower Bounds

Objective Functions

3 Applications and Results

3.1 Data Cleaning

Upon to two distinctive constraint definitions in my advance project 2 report, checking those hypothetical terms in real life data is decided. To be able to observe interesting patterns, a big data set with 2-3 years production orders is agreed to be investigated through time windows.

I and Daniel started to discuss about the relevant features to be considered in this data set. At final stage below given SQL query was generated to pull the data set from the SMS database. The resultant data set consists of 459203 rows and 15 columns.

First two columns and 4^{th} column features: ROS.R_OS_ID, ROS.PRODUCTION_LINE_NAME and ROS.REFERENCE_DATE come from "Reporting data: Operation step" table. 3^{rd} column feature SEQUENCE_ID is actual casting sequence ID from the table "Reporting data: additional data of CCM (explain this)". 5^{th}., 6^{th}., 7^{th}. and 14^{th}. SLAB.PIECE_ID, SLAB.MATERIAL_ID, SLAB.MOLD_WIDTH, and SLAB.EXIT_TEMP come from "Reporting data: additional data of CCM which are slab related". Rest of the columns: MAT.WIDTH, MAT.THICKNESS, MAT.WEIGHT, MAT.LENGTH, MAT.HEAR_ID, MAT.STEEL_GRADE_ID_INT, and MAT.SLAB_TRANSITION come from "Material ; For slabs, coils, plates and heats" table.

```
SELECT  ros.r_os_id, ros.production_line_name, ccm.sequence_id,
ros.reference_date, NVL( TO_CHAR(slab.piece_id),'NA')
piece_id, NVL( TO_CHAR(slab.material_id),'NA') material_id,
NVL(TO_CHAR(slab.mold_width),'NA') mold_width,
NVL( TO_CHAR(mat.width),'NA') width,
NVL( TO_CHAR(mat.thickness),'NA') thickness,
NVL( TO_CHAR(mat.weight),'NA') weight,
NVL( TO_CHAR(mat.length),'NA')
length, NVL( TO_CHAR(mat.heat_id),'NA') heat_id,
NVL( TO_CHAR(mat.steel_grade_id_int),'NA') steel_grade_id_int,
NVL( TO_CHAR(slab.exit_temp),'NA') exit_temp,
NVL( TO_CHAR(mat.slab_transition),'NA') slab_transition

FROM    L3MAIN.r_os ros
```

```
LEFT JOIN L3MAIN.r_ccm ccm ON ros.r_os_id = ccm.r_os_id
LEFT JOIN L3MAIN.r_ccm_slab slab ON ros.r_os_id = slab.r_os_id
LEFT JOIN L3MAIN.r_mat mat ON ros.r_os_id = mat.r_os_id

WHERE   sequence_id IS NOT NULL;
```

Parsed data belongs to CCM (Compact Cold Mill) production line.

$7.85g/cm^3$ = $7850kg/m^3$ = $0.284lb/in^3$ = $490lb/ft^3$

Converting strings to numbers and correction for punctuation marks between digits were performed, null values (NA) were converted into 0 values in the beginnning of data cleaning process. After completing minor stages, some pre-conditions were generated as below to be able to manipulate data columns. Steel density is considered between $7.00x10^{-6}$ kg/mm^3 and $8.50x10^{-6}$ kg/mm^3. Width varies between 800 - 2000 mm. Thickness varies between 40 - 90 mm. Weight varies between 2669 - 26690 kg. Length unit is mm.

Starting to modify width, thickness, and weight values corresponding to thickness values with 2 digits.

The data set has below given shape just before starting to analysis. Weight Zero Rows: 10484 Thickness + Width + Weight Zero Rows: 61320 The rows with densities that do not match within above mentioned interval: 1787 Usable Rows: 396096

Time Windows Generation by Data Partitioning:

the dataset with length 396096 was partitioned in 10 time windows starting from the beginning of the data. In each step, it's increased by 39610 rows more or less (increasing windows). The exact increase step dimension was specified by the last order of corresponding sequence. For my dataset, exact time window lengths are 39871, 79567, 118358, 158421, 198041, 237352, 277147, 316411, 356385, 396096. Almost always same statistics for every window. Strange increase modularity increase towards the end due to increasing window size. If there is a shift in the way the data behave, I will almost not see it because it is mast by the other data that still be present in my analysis. The modularity curves seem drift upwards little bit. There is a trend of going up now matter how it behaves in the middle. My reason was to do this to check the load effect.

Partitioning was repeated with discrete time windows (sliding windows). Shifting window within equal windows size. To see the results of same analysis in each discrete time window. Whether the rules I discovered from the first dataset (1st time window) and the second dataset (2nd time window) are really fundamentally different or rather the same.

21.04.21 Below steps were performed for the data sets belong to different production lines. • Sequences with less than 50 events were removed from the data sets considering those short sequences might be generated for some test processes. • The events with the densities out of the interval (6.5x10-6, 8.5x10-6) were removed from the data sets.

At the final stage, obtained data set lengths are given below.

- PLTCM data set: 64,026 events
- CGL data set: 31,230 events
- CSP data set: 205,496 events
- CCM data set: 347,418 events

3.2 Real-life Events Analysis

The steel manufacturing data sets were analyzed in six different dimensions: Production Line, Production Constraints, Production Feature, Time, Network Resolution and Null Model.

For the first dimension, distinguished data sets were considered among four different production line: Continuous Casting Machine (CCM), Compact Strip Production (CSP), Continuous Galvanizing Line (CGL), and Pickling Line & Tandem Cold Mill (PLTCM). In principle, CCM and CSP production lines have similar functionalities; however, they were kept separated in the analysis pipeline since their labels are different in the database. Two fundamentally different constraints acting on the manufacturing process: technology-driven constraints and load-driven constraints, were shaped hypothetically and defined as two distinguished network approaches: fixed step-sized and fixed bucket-sized networks. Those attempts are in the second dimension of the analysis. As the third dimension, the width and thickness features of slabs were picked to investigate different production constraints that play a role in the machines for those features. Time is the fourth dimension and considered to check constraints impact on the historically ordered production events. The data set was treated in both discrete-time windows and increasing-time windows to study the behaviour of changing fixed step size and fixed bucket size. Generated networks were diversified in two different resolutions by changing the node amount in the fifth dimension. As a concept of characterizing, modularity was calculated for the networks. The aim is to keep the networks with a similar number of nodes in both network approaches (fixed step size and fixed bucket size) so that the modularity quantification would be meaningful to compare. As the last dimension, two types of null model: shuffling the links in degree

conservation and shuffling them based on the communities, were considered to check the randomness of the networks. Obtained Z-scores can vary based on the generated null model via that specific randomization method.

The data sets were partitioned into two halves, and analysis steps were applied for the first half, second half, and the complete data set. At the top of bar chart sets, modularity values were presented for the original network and a single randomized network. Z-scores belong to different null models for 1000 randomized networks were given in the bottom part of the bar chart set, indicated with a colorless line finish. For each of the z-scores, error bars were included by removing and putting back 10% of the data several times. The colored bar border indicates the mean value, and the T-shaped symbol represents the standard deviation of the error bars.

24.03.21 * Regarding behavior on networks with changing fss and fbs amounts, first column plots show a calibration curve which have graph node numbers corresponding to changing fss and fbs. For Weight feature, fbs paradigm leads to higher modularity than the fss paradigm no matter which bucket/step size we pick. This result becomes opposite when it comes to length and width. For thickness, there is no clear result to say as the others have. They are actually sometimes on the same level. The fact that modularity tends to be higher in one paradigm and lower in the other which is an interesting thing.

Investigation of constraints impact in the data with time-resolved fashion confirms my previous investigation on the data with increasing time windows. Our hypothesis at the moment is that the physical constraints are rather about step sizes than about bucket sizes. Because step size graphs are less random.

In my previous project, the fixed step size graphs had a high modularity. This means that the actual quantity I discretize creates the constraints while in the case of fixed bucket size it would be the volume of orders that creates my constraints. This summarizes our hypothesis.

What I would get = On the width level, we have a fairly constant behavior over time and it confirms what I see on the total dataset, strong difference in modularity between the two network approaches. For the fbs approach, there might be a transition, the fixed bucket size becomes very modular in the end.

For thickness, its less clear because the modularity is in same level for both fss and fbs but the increase in modularity for fbs is fairly drammatic. It goes from 0.3 to 0.6 while the other remains at 0.3 and fluctuates. In the last two time windows in fbs, the process is dominated by something else. It is an evidence that something changed of the constraints involved really takes place.

For length, both modularity changes in different network structures are between 0.1 and 0.2 is beyond the resolution of what I can do. These two curves

are almost identical in the statistical point of view. Also they are close to what you would get for totally random graphs in fact. There is no modularity here we can talk about.

For weight, it is same as length. The z-scores are fairly close to zero compare the ones for other feature z-scores. In the case of the fbs, you can see it might be slightly higher than the modularity of the random graphs but it is not a deservedly modularity.

current situation The Modularity (Single Random Graph) and Z-score plots, dashed curves are provided for the Fixed Degrees Null Model and unbroken curves are provided for Modularity Null Model. Network node numbers were kept equal for the same time windows in different network approaches but not in the consecutive time windows in the same network approaches. Other than the below-given networks, including fewer nodes than 15 in some cases, all networks have varying node numbers between 25-90.

- CSP Thickness Network with narrow node binning
- CSP Thickness Network with large node binning
- CCM Thickness Networks with narrow and large node binning

Results are not stable which is a bit expected with these complicated data structures. This is why we do the sensitivity analysis on top of that: varying the resolution, doing things with slightly different methods (different null models) over and over again. In some sense, we try to do the data as simpler as they are, by trying to pretend that they are more homogeneous than they are. We pretend that the data is statistically reliable and not distorted and most of all that this temple of the modular network really fits. All of these assumptions are slightly wrong, what we see here is the consequence of that. We need to try to make the statistical assumptions about the data that are not necessarily fulfilled and these assumptions can lead to things sometimes looking statistically significant even though they are not. And those statistical assumptions are most of the time that the data are not very distorted but somehow reliably distributed. And top of that, modularity as a fairly simple concept of characterizing networks, is a meaningful concept here. We should say that these are all assumptions.

3.3 Integration of Concepts

What happens here is to bring the simulated data into a format that is compatible with my analysis of the real-life events data.

In an ideal scneario, we would find that association networks derived from the generated data, in the one case; produce high modularity for FBS and in

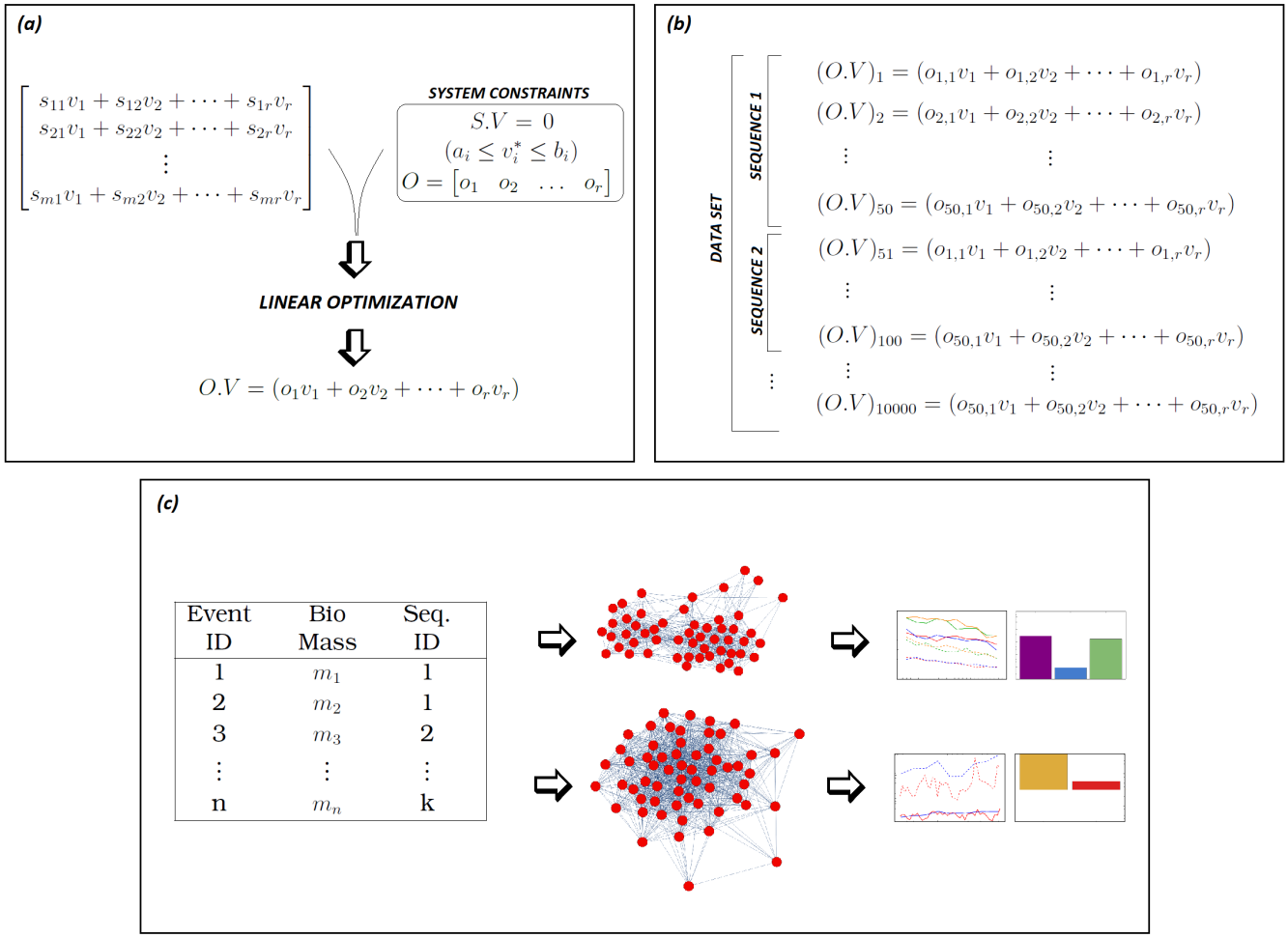


Figure 3.1: Complete Framework Sketch.

the other case produce high modularity for FSS. Because then we have linked these two data processing schemes to different forms/to different categories of constraints.

3.4 Simulation Results

Briefly explain *in silico analyses* attempts /numerical experiments from the generated data.

Plots in Part-1 of the file belong to association networks of four different synthetically created sequence data sets, and each represented in various colors: green, blue, orange, and red. Part-1 data sets were derived with fixed reaction bounds but with varying coefficients of objective functions. Part-2 also

presents plots for four different synthetically created sequence data sets with fixed objective function coefficients but with variable reaction bounds.

Each data set has a length of 60,000 events shared equally in 200 sequences. Randomly picked subsets of fluxes were kept the same within the sequences but having varying coefficients of objective functions.

The association network data sets were structured by the dot products of objective function vectors and optimized solution vectors. This product results maximize the output, if I am not wrong, as the maximization attempt of biomass in the FBA model.

4 Conclusion And Outlook

5 Bibliography

[1] D. Merten, M.-T. Hütt, and Y. Uygun, "A network analysis of decision strategies of human experts in steel production," *submitted to IISE Transactions*, 2020.

[2] B. Kim, W. J. Kim, D. I. Kim, and S. Y. Lee, "Applications of genome-scale metabolic network model in metabolic engineering," *Journal of Industrial Microbiology and Biotechnology*, vol. 42, pp. 339–348, 03 2015.

[3] T. Hao, D. Wu, L. Zhao, Q. Wang, E. Wang, and J. Sun, "The genome-scale integrated networks in microorganisms," *Frontiers in Microbiology*, vol. 9, p. 296, 2018.

[4] K. J. Kauffman, P. Prakash, and J. S. Edwards, "Advances in flux balance analysis," *Current Opinion in Biotechnology*, vol. 14, no. 5, pp. 491–496, 2003.

[5] N. D. Price, J. L. Reed, and B. . Palsson, "Genome-scale models of microbial cells: evaluating the consequences of constraints," *Nature Reviews Microbiology*, vol. 2, no. 11, pp. 886–897, 2004.

[6] A. Varma, B. W. Boesch, and B. O. Palsson, "Biochemical production capabilities of escherichia coli," *Biotechnology and Bioengineering*, vol. 42, no. 1, pp. 59–73, 1993.

6 Supplements