June 28, 2021

Abstract

Give a concise synopsis of the work, emphasizing the conclusions; you need not include the supporting arguments for the conclusions. It should be an accurate overall view of the work without needing to read it. State the subject of the paper immediately followed by a summary of the experimental or theoretical results and the methods used to obtain them.

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

1 Introduction

1.1 Background and Motivation

Below paragraph was taken from Advance Project-II and it will be used as a background information after summarizing.

A steel manufacture facility's production lines might be structured with different combinations of those steps based on the diversity of the demanded output product or the manufacture facility capacities. 'Production Constraints' which is a critical subject to optimization researches arise from those technology-driven production types related to facility capabilities [1]. Raw materials like coal, iron ores, and scrap metals are melted in blast/electric arc/basic oxygen furnaces to obtain liquid iron as primary product. Accordingly, liquid steel alloy is sent to a continuous casting line, and poured into a large customized volume trapezoid prism called tundish. The steel alloy comes off the mold close to the bottom outlet of the tundish and gets into an entry nozzle until it reaches the rollers. The shaped material is treated between rollers with a water cooling system, and converted into semi-finished casting products such as slabs, blooms, or billets. For the purpose of gaining the desired mechanical properties, uniform thickness of the material, and controlling width dimension, outputs are sent to the rolling process which is a pressing application on slabs by using rolls. The rolling process can be performed in two different modes: hot rolling and cold rolling. Hot rolling can be performed if the material temperature is above its re-crystallization temperature. Otherwise, reheat furnaces are used to obtain the proper temperature of the metal which is known as deformation temperature prior to the hot rolling process. Metals with a material temperature below re-crystallization temperature are treated in a cold rolling process. Surface finish and flatness can be improved and modification of material work hardening can be achieved with the cold rolling process. With the coiling step in the rolling process, slabs might be converted into compact coils featuring high lengths unless they will not be sent to further steps on the continuous production line. The following steps can be given as pickling process and hot-dip galvanizing process. As an effective coil coating technique, galvanizing is an application of protective zinc coating on the steel surface to improve corrosion resistance. If this method is applied by submerging the steel parts into a molten zinc bath, it is called hot-dip galvanizing process. Considering the roughly ex-

plained production steps above, each process has its own technical or physical constraints. Optimizing individual sequences in each process is a necessity for a successful local process. Local constraints on different processes are integral parts of a global optimization problem that is tackled by human expert planners for a solution. The sequences produced in production lines are available as data output, so-called 'imprints' of what has been on sequence designers' mind. Therefore, looking at the historical production data, and an investigation on the properties of those order sequences that have already been produced give indirect access to the patterns which are related to human experts' knowledge system. The technical and physical constraints mentioned in the introduction section are obviously unique to those specific production lines and they vary under differently customized production lines. Produced order properties such as thickness, width, temperature, and chemical composition are the characteristic features of order products that are possibly shaped under the effect of those constraints. An investigation on the features of orders in the same production sequences and comparison between them might give interesting clues about related constraints and correspondingly provide patterns about human experts' decision strategies.

I should discuss and refer to different constraints from the literature introduced for the manufacturing life cycle.

As a motivation, I need to introduce different categories of constraints as technical constraints, performance-indicator based constraints to be quantified in the context of the FBA in the further steps of this work.

1.2 Research Objective

Our hypothesis: different types of constraints create non-random features in the association networks for different binning schemes. Networks derived from various types of binning. Do they show non-random features when I have performance constraints or other types of constraints?

Explanation of my hypothesis is a theoretical/conceptual framework as a starting point for the investigation. It is a well-defined valid object and based on facts. Moreover, it is structured to discriminate the two types of constraints in the statistical properties of the production data.

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.

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.

I plan to quantify the characteristics of two hypothetical types of constraints with an Operations Research Model consisting 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.

Formulate the binning methods here because this will describe the hypothesis underlining my thesis.

My Operations Research Model (OR model) combines 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.

2 Methodology

Introduce proposed concepts in the OR model.

Here is a brief introduction to Association Networks, Modularity as a Complex Networks metric, Randomization with Null Models and Flux Balance Analysis (FBA).

Introducing the analysis of the real-life events with the related concepts and explain the relation of the thesis hypothesis with this analysis pipeline.

Here is an explanation for generating a data structure with OR-modeling in the combination of those. Usage of linear programming and creating sets of synthetic data allow comparing the statistical characteristics of their association network with those formed from the real-world data set from steel manufacturing.

Mention that more detailed information will be given in the following sections, guide the readers who know Association Network and FBA concepts to the Applications and Results Chapter.

2.1 Steel Manufacturing Data Analysis

Association Networks

Beyond a simple network graph representation of historical production data, the formation of association rules networks is an insightful graph-based framework combining the tools: association rules and complex networks, as Merten et al. (2020) performed in their article [2]. The relevant pipeline considers sequentially revealed events of a data set. It outputs a graph demonstrating the non-random occurrence of specific events together among the complete set that took place consecutively in the production period.

Assume we have an arbitrarily created manufacturing data set with chronological order, D, consists of k sequences and n events with Feature-A values and sequence id's included as given in Table 2.1.

By looking at such a data set, one can say the events with Feature-A values:

Event_ID	Feature-A	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: Arbitrary Manufacturing Data Set D.

890, 850, 650, 745, 795 or 540, 520, 630, 610 are positioned in common sequences and close to each other; thus, they are produced together and likely occur in the identical sequences. As a further argument, the conclusion mentioned above is probably a deliberate planning choice based on the related constraints acting on the manufacturing process performance. However, extracting such implicit knowledge is not a simple task for large and complicated real-life manufacturing data. For example, such a data set may consist of more than $300,000$ events and is likely to have various events aggregated randomly in its large sequence groups.

We extract the association rule from the set of production sequences to distinguish statistically unexpected occurrences from the non-random ones in production sequences and assess the complexity of production patterns. The association rule measure, known as " Lift ", was picked with a similar approach as Merten et al. (2020) applied in their article [2]. It was calculated for every possible pairwise subset of Feature-A values belonging to the events in identical production sequences. The Lift can be computed as the ratio of pair items joint probability divided by the multiplication of each item's marginal probability as

$$Lift(A \leftrightarrow B) = \frac{P(A,B)}{P(A) * P(B)}. \tag{2.1}$$

In the case of $Lift(A \leftrightarrow B) > 1$, B occurs likely if A occurs while $Lift(A \leftrightarrow B) <$

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

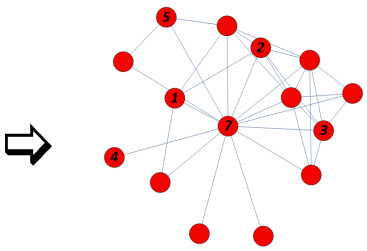


Figure 2.1: An Arbitrary Representation for Adjacency Matrix and Its Graph.

Binning Methods

The data set D events can be labelled with a typical value interval (the so-called binning size) for every Feature-A value with a slight difference to each other. Binning generation for the events allows us to investigate them in a sequential manufacturing system and can be performed in alternative ways.

Say that we do the Feature-A values labelling with a typical binning size, in our case, 99, so that all of the events in D must match the corresponding step interval, Fixed Step Size (FSS), as shown in Table 2.2.

Event ID	Feature-A	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

Table 2.2: Data Set D with FSS Bin Size Labels.

An alternative way of label generation is to create bins with equal event counts per bin among the complete data set, Fixed Bucket Size (FBS) given in Table 2.3. The alternative binning generation methods mentioned above let us

Event ID	Feature-A	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

Table 2.3: Data Set D with FBS Bin Size Labels.

derive two distinguished network approaches. The first one is the FSS Network; it has graph nodes as binning groups with equal bin sizes. Manipulation of binning size allows us to aggregate events in different network nodes. The second one is the FBS Network; its nodes are binning groups with an equal number of events per bin. Defining a typical bucket size for the network nodes results in arbitrary interval boundaries for each node, and it allows to control their population.

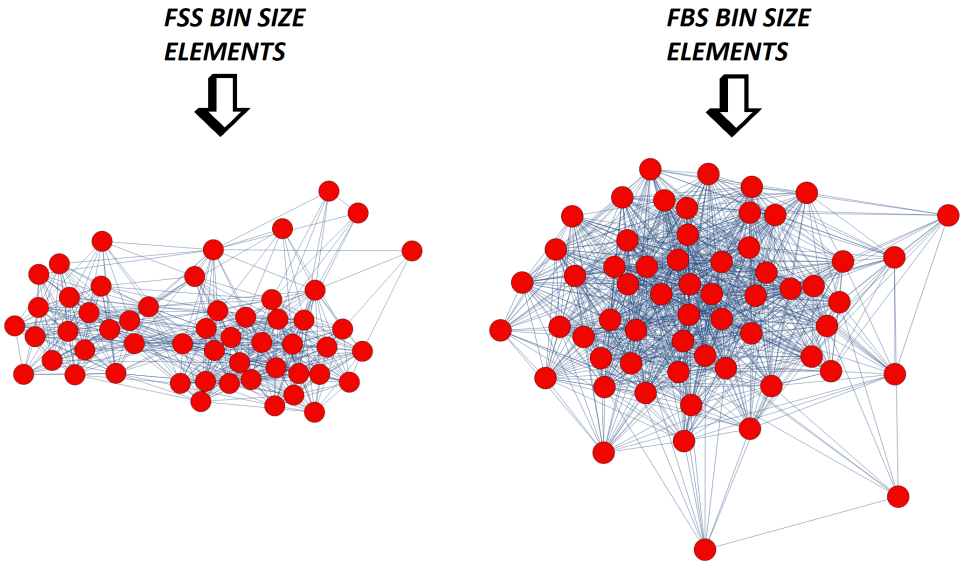


Figure 2.2: Graph Results For Two Different Network Approaches.

FSS and FBS networks generation for the production events underlie the developed hypothesis of this thesis work: Non-random features of the association networks derived from these two methods.

Network Metrics Analysis

Modularity

As explained in the previous subsection, one data set can be labelled differently, and networks are obtained with alternative approaches: FSS Network and FBS Network. Resultant graphs have various motifs, which we argue that they emerge from the statistical patterns in data.

The variety of dense textures arise from the nodes having different degree values within their neighbourhood. The degree is a network metric that quantifies one node's links to the other nodes [3]. It gives an idea about the connectivity patterns within the network and allows us to distinguish the group of nodes with a high degree from the nodes with a low degree.

Identification of tightly connected node groups is a way of quantifying community structure in networks [4]. Joined node groups with various degrees form significant modules in the graph; those patterns give insight into how the data elements are related in the whole network. Modularity is a network measure for community detection and quantifies the strength of community structure in that specific network. It is a way to express the network characteristics.

Newman (2006) formulated modularity in his article as

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \frac{s_i s_j + 1}{2} , \tag{2.2}$$

where the network has an m number of edges, and A_{ij} is the number of edges between vertices i and j. A_{ij} is the element of the adjacency matrix introduced in Fig. 2.1. It can be 0 or 1. k_i, and k_j are the vertex degrees, and $k_i k_j / 2m$ is the expected number of edges between i and j if edges are placed at random. s_i and s_j are the divided network groups. They are equal to 1 if i and j belong to the same group and 0 otherwise. Eq.(2.2) is used to divide the network into two communities only; however, many networks may contain more than two communities. Therefore, a repeated division into two is adapted: dividing the network into two graphs, then the two sub-graphs further divided into two only if that would maximize Q. After first partitioning, the edges falling between the further divided sub-graphs are neglected, leading to a wrong maximization quantity. For this reason, the author introduced the additional contribution ΔQ. [5]

Since the results obtained with the combination of Q and ΔQ do not significantly differ from the results obtained only using Q, modularity calculations in this work were performed with the latter to lower the computation timing.

Null Models

Generating the null models from a set of random scenarios by switch randomization and calculating standard scores for the modularity values is an efficient method to quantify randomness [2, 6].

Randomness is quantified by computing z, the standard score of overlaps for features one-to-one relation as given below.

$$z = \frac{x - \mu}{\sigma} \tag{2.3}$$

where x is the modularity value of the actual network graph, μ is the mean modularity value for the random graphs and σ is the standard deviation of the modularity values for the random graphs. The z-score values less than 1 indicate that the actual network graph is random, the z-score values between 1 and 2 or between -2 and -1 indicate that the graph is close to random characteristics whereas the z-score values higher than 2 or lower than -2 suggest a significant deviation from being a random modularity value.

[7]

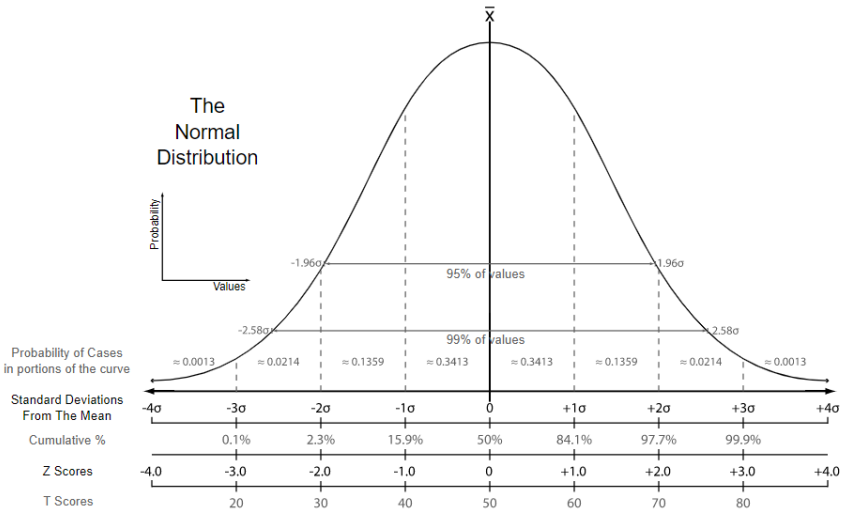


Figure 2.3: Chart Comparing the Various Grading Methods in A Normal Distribution.

If I have more than two modules, do I conserve the specific number of inter-module edges per module pair? Would a link between module 1 and 2 be shuffled with a link between modules 1 and 3? All the interlinks independent from the number of modules are shuffled in the same set. I did not consider different module interlinks in separate groups. It is a design decision.

In particular, if I do not have the pleasant situation that people usually think of in a modular graph where the connectivity between modules is typically about the same for any pair of modules. But if I have a slightly more realistic situation where module pairs differ very strongly in the way they are interconnected. In real-life data, I often have two challenging modules to separate because they are tightly interconnected and then maybe one or two other modules that are almost not connected to the other modules. That is the part that I do not conserve and might affect the result.

Modules within modules, leading to a hierarchical network [8].

Many real networks in nature and society share two generic properties: they are scale-free and they display a high degree of clustering. We show that these two features are the consequence of a hierarchical organization, implying that small groups of nodes organize in a hierarchical manner into increasingly large groups, while maintaining a scale-free topology. In hierarchical networks, the degree of clustering characterizing the different groups follows a strict scaling law, which can be used to identify the presence of a hierarchical organization in real networks. We find that several real networks, such as the Worldwideweb, actor network, the Internet at the domain level, and the semantic web obey this scaling law, indicating that hierarchy is a fundamental characteristic of many complex systems. [9]

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 null models: Degrees Fixed and Modularity and different choice of binning in terms of step-size & bucket-size.

Since plot results vary based on a specific randomization null model, discussing how shuffling was performed is essential. Random graphs were generated, and the below instruments were plotted. Modularity vs time windows Average modularity for single random graph vs time windows Z-scores for modularity randomized null models with modularity conserved vs time windows Z-scores for modularity randomized null models with fixed degree sequences vs time windows

The cartoon showing how I generated the null models in a pairwise shuffling fashion to prevent ambiguity.

The full z-score curves are only there because we want to ensure that we don't overlook something. If the full curves are drastically away from zero, then the type of modularity in the real graphs are somewhat different. They are either very asymmetric with respect to the modules or hierarchical or anything else

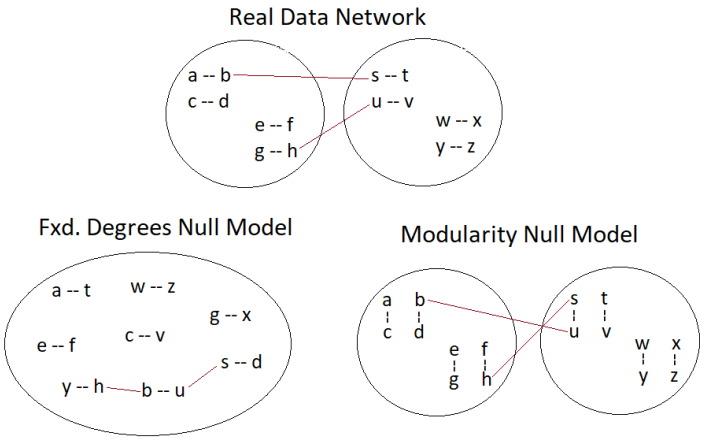


Figure 2.4: Formation of Different Null Models.

that is very complicated. In some sense, the full curves are just a reference check to whether everything works as expected and whether it is meaningful to discuss modularity. So what we are really looking at is the dashed curves. And I am looking at the z-score only; I am trying to figure out whether the step from FSS to FBS drastically changes the modularity. I am wondering whether we can condense this further to make this information is more accessible.

If the full z-score curves are always close to zero for both different network approaches, would that give us an idea about the modularity? Then it says that the modularity is as we expect the modularity to be.

If I have a very modular graph and then I randomize it such that I conserve the modularity. Then the comparison of the original modular graph with the randomized modular graphs will lead me to a zero z-score, no matter what the modularity is.

The dashed line is the more meaningful null model, but we need the other null model because modularity might have strange effects. For example, if the real network has a hierarchical structure, modules within modules within modules, I would still see modularity (positive z-score with respect to the null model behind the full curve) because I am destroying the other type of modularity (nested modularity) in my null model. So in some sense, this is just checking that we have that type of (nested) modularity in deep.

If you have a strongly modular network and each module is in itself modular, then in my null model that preserves modularity, I would destroy that internal modularity of modules. So I would have positive z-scores (full lines would be shifted upwards).

If the full curves are positive, it means we have modules in modules. If the full

curves are negative, it means (guessing a bit) that we have big differences in the number of inter-module links. So that some modules are tightly connected while other modules are sparsely connected. And in my null model, this is average. That means that the modularity in the randomized module-conserving networks is actually higher.

Is the dashed curve of FSS higher than the dashed curve of FBS? That's a type of information we should extract from those plots/bar charts.

Performing switch-randomization to a modular graph might fail even due to small details in randomization steps. That failure is probably the reason for the 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 the same module and switch them or links 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.

2.2 Flux Balance Analysis

The genome-scale integrated networks are necessary tools used by metabolic engineers on model design, theoretical and computational analysis for microbial organisms. In addition, integrated network theory tools expand the feasible space for analysis techniques in further work steps. As an initial step, one can construct a network showing interactions between metabolites, intermediate or end products and metabolic reactions for an organism.

The set of rules for the organism can be represented in a compact form by an m-by-r matrix formulation as

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1r} \\ s_{21} & s_{22} & \dots & s_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \dots & s_{mr} \end{bmatrix} = (s_{ij}) \in \mathbb{Z}^{mxr}. \tag{2.4}$$

The matrix S is called stoichiometric matrix, its column elements represent reactions that play a role in the chemical transformations, and its row elements represent metabolites. S also contains direction information for the related metabolite-reaction element in the matrix with positive or negative signs. [10]

Having transpose of S will reverse the columns and rows in the matrix as

$$S^T = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s_{r1} & s_{r2} & \dots & s_{rm} \end{bmatrix};$$

thus, by the product of S and S^T, we obtain two different matrices as

$$S.S^T = \begin{bmatrix} s^{'}_{11} & s^{'}_{12} & \dots & s^{'}_{1m} \\ s^{'}_{21} & s^{'}_{22} & \dots & s^{'}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ s^{'}_{m1} & s^{'}_{m2} & \dots & s^{'}_{mm} \end{bmatrix} \quad \text{and} \quad S^T.S = \begin{bmatrix} s^{''}_{11} & s^{''}_{12} & \dots & s^{''}_{1r} \\ s^{''}_{21} & s^{''}_{22} & \dots & s^{''}_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ s^{''}_{r1} & s^{''}_{r2} & \dots & s^{''}_{rr} \end{bmatrix},$$

where $S.S^T$ is a metabolite-centric matrix and $S^T.S$ is a reaction-centric matrix. Considering a normalizing step for those matrices as

$$f(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } x \neq 0 \end{cases},$$

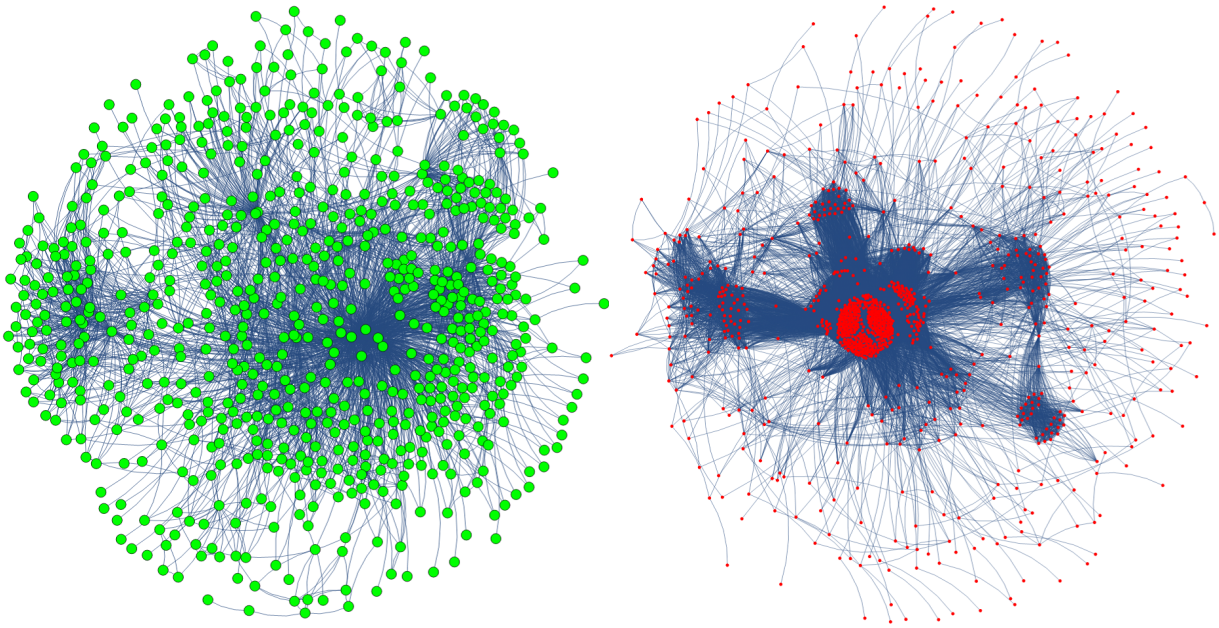
one can construct adjacency matrices, $A^m_{ij} = f(s^{'}_{ij})$ and $A^r_{ij} = f(s^{''}_{ij})$, to form networks like the one introduced in Fig. 2.1.

The graphs in Fig. 2.5 were generated from A^m and A^r using a stoichiometric matrix belonging to homo sapiens metabolism retrieved from BiGG Models Database [11]. In Fig. 2.5a, the graph nodes stand for the metabolites, and graph edges are the reactions. In contrast, in Fig. 2.5b, the roles are reversed so that the graph edges represent the metabolites, and the graph nodes represent the reactions.

Studying biological metabolic systems and designed models to achieve cellular objectives like cell growth or ATP (Adenosine Triphosphate Production) necessitates various tools to be integrated with reconstructed genome-scale networks [12, 13]. One of the commonly used tools is Flux Balance Analysis (FBA) as an optimization scheme. 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 [14, 15].

While one can express the metabolic fluxes in a one-dimensional array (the so-called flux vector V) as

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix} = (v_i) \in \mathbb{R}. \tag{2.5}$$



(a) Metabolic-centric Network

(b) Reaction-centric Network

Figure 2.5: Network Representations for Homo Sapiens Metabolic Model

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. Defining a mass-balance ($S.V = 0$) constraint in the FBA enables us to analyze the metabolic network operations in a steady-state solution space [14, 15].

$$S.V = \begin{bmatrix} s_{11}v_1 + s_{12}v_2 + \cdots + s_{1r}v_r \\ s_{21}v_1 + s_{22}v_2 + \cdots + s_{2r}v_r \\ \vdots \\ s_{m1}v_1 + s_{m2}v_2 + \cdots + s_{mr}v_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.6}$$

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 type of 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 [14]. Therefore, one can formulate an optimization approach to identify reaction network steady-states that maximize the biomass [14, 15] or control the production of specific metabolites [16]

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 [15]:

i. to discover allowable characteristic properties in the genome-scale network reconstruction,

ii. to mimic probable physiological functions like biomass or ATP production to be able to determine likely physiological states and

iii. to design a genetic variant or sub-type to obtain a desired particular product.

One can express objective function coefficients in a one-dimensional array as

$$O = \begin{bmatrix} o_1 & o_2 & \dots & o_r \end{bmatrix} = (o_i) \in \mathbb{R}. \tag{2.7}$$

As given in Eq.(2.8), the biomass formulation delivers the output with its non-zero coefficients, which are the decisive ones for the flux elements of V to be considered.

$$O.V = (o_1 v_1 + o_2 v_2 + \cdots + o_r v_r) \in \mathbb{R}_{\geq 0}. \tag{2.8}$$

Stoichiometric (or mass-balance) constraints were introduced so far in Eq.(2.4) and Eq.(2.6). In addition, upper and lower bounds are presented 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 influence the reactions for uptake and secretion, whereas no limitation is considered in the exchange reactions. Quantifying imported nutrients and exported outputs (resources and products) by constraining them with upper and lower bounds to fulfil a single objective function goal might significantly influence the optimization process.

As a summary of the above-explained series of constraints, mass-balance (Eq.(2.6)), upper & lower bounds for fluxes (Fig. 2.6), and the objective function (Eq.(2.7)) are the three fundamental constraints that set off a linear programming problem because it is possible to formulate them linearly [15]. The optimization result: flux vector V (Eq.(2.5)) maximizes the objective function in the form of a flux distribution [14, 15]. Since each term in Eq.(2.8) 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.

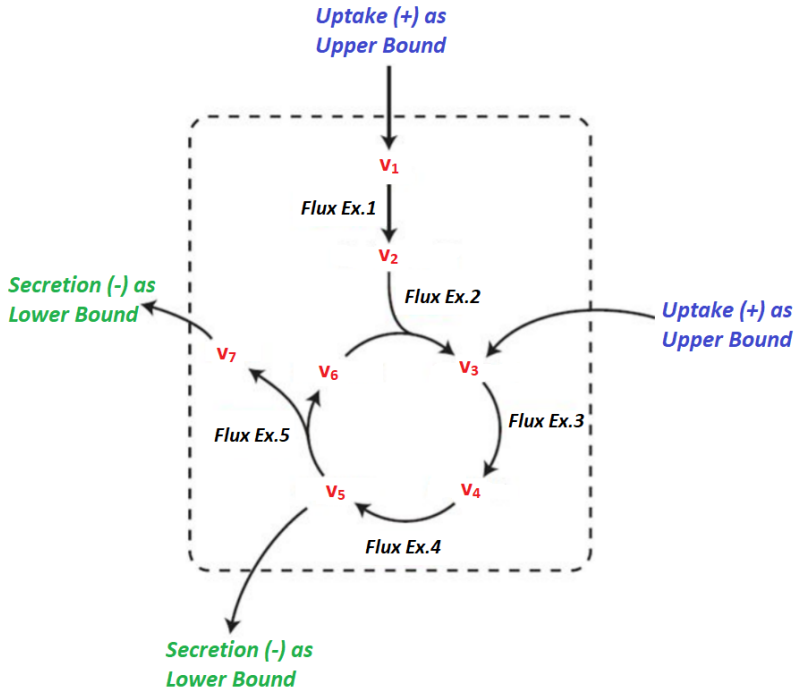


Figure 2.6: A Simplified Reaction-centric Network Sketch Shows The Reactions for Exchange, Uptake and Secretion.

Different solution vectors of V can be obtained from the linear optimization process by varying network conditions. As a compact set of rules, the stoichiometric constraints significantly influence the mass-balance equation; consequently, the solution vector V [17]. A stoichiometric matrix from scratch can be formulated, ensuring the mass-balance constraints are incorporated in the reaction cycles of the investigated system. However, the homo sapiens metabolic model was taken as the set of rules in this thesis work. Varying upper & lower flux bounds and the objective function are the two alternative approaches introduced in the following subsections to understand the model behaviour while the optimization is carried on.

Resource Utilization

Environmental conditions such as resource availability affect the pattern of outputs in a metabolic network. In case of fewer resources (nutrients) availability, the active production network gets more interconnected through more flux exchanges to produce the necessary input for the ongoing metabolic reactions. [15, 18, 19, 20]

$$V^b = v_1^b, v_2^b, \ldots, v_x^b = (-a \leq v_i^b \leq a) \in V \tag{2.9}$$

Let V^b is a list of fluxes with x elements randomly picked from V (Eq.(2.5)) to be limited with the bounds: $(-a, a)$. The same tolerance in both negative and positive direction for the bounds allows the network to treat the respective flux flow as uptake or secretion based on the system need. The fluxes that are not included in V^b are matched with extreme high boundary values so that they are not constrained while the linear optimization.

$$V^e = v_1^e, v_2^e, \ldots, v_y^e = (0 \leq v_i^e \leq 0) \in V \tag{2.10}$$

Assigning zero to the upper & lower bounds suppresses the respective flux exchange in the active production network. Deleted fluxes can not be used for the uptake, secretion, nor intermediate reactions. V^e (Eq.(2.10)) is the list of fluxes with y elements randomly selected from V (Eq.(2.5)) to be discarded from the network.

Limitations on resources serve as capacity constraints defining the active reactions and reversibility of flux exchanges [17]. Varying x, y, and a to fulfil a fixed objective function, we obtain various biomass values by the linear programming algorithm.

Product Portfolio Diversification

The objective function can be assumed as a production plan that rules the diversity of products that metabolism takes into account to maximize cellular growth [17]. As previously mentioned, this is because the pattern of output biomass (Eq.(2.8)) is governed by the objective function (Eq.(2.7)). Its non-zero coefficients force the network for an optimal solution with their value range and positive and negative signs.

Defining a variety of sets of objective functions, each has negative or positive signs, will allow us to create a diverse group of products that the network is capable of producing. In the same direction, deleting the objective function terms up to a certain level is the second step of that diversification approach.

2.3 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.

An event is a single run of my optimization algorithm.

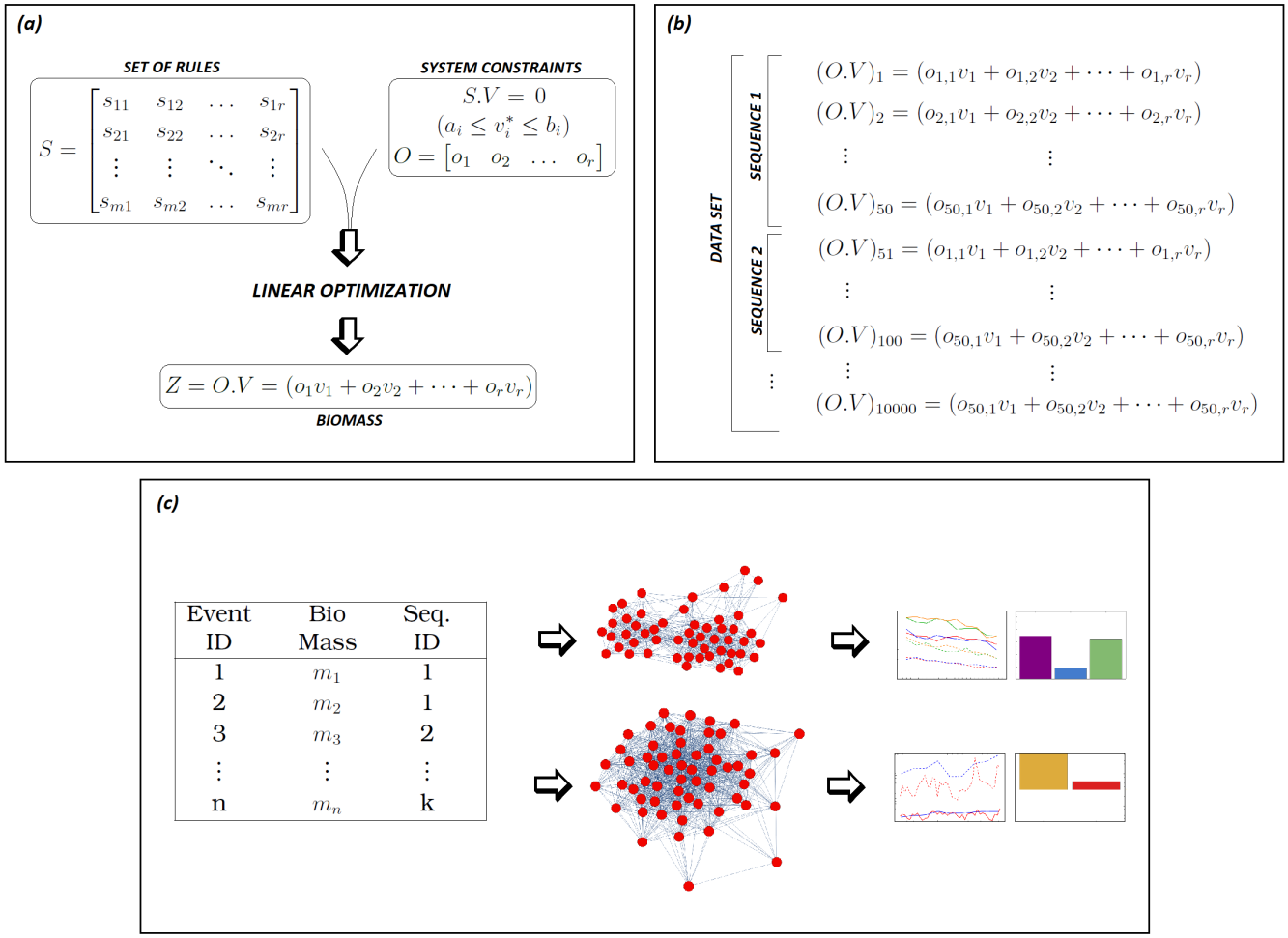


Figure 2.7: Complete Framework Sketch.

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, as the maximization attempt of biomass in the FBA model.

Explain how we introduce production sequence concept for the FBA.

Step size was increased from a few deleted reactions to many deleted reactions so that network step sizes were always kept between 40 and 50.

3 Applications and Results

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 average modularity metric plots were generated.

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.

After discussing the relevant features to be considered in this data set, 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 (Continuous Casting Machine) 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.

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.

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.

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

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.

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.

3.3 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 10,000 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.

Limitation on resources were performed in two different ways; first, restriction on upper & lower bounds and second, deletion of fluxes.

The fluxes used in the intermediate reactions were given the range of bounds as $(-500, 500)$ since it is not possible to define infinity values in the optimization algorithm. Randomly chosen 105 fluxes out of 1008 were matched with $(-5, 5)$ as the first step. And 105 was doubled (212) and then quadrupled (425). An important detail is all of the three set of choices were done randomly, they are not added on top of already selected 105. In every further step, the same three set of fluxes were used in the computations as restricted bounds.

Deletion goes in the line: $0, 50, 100, 150, 200, 250, 300, 350, 400, 450$. As explained previously, deletion was done by assigning $(0, 0)$ bounds to the fluxes. On the last step, almost half of the total fluxes (1008) were erased.

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

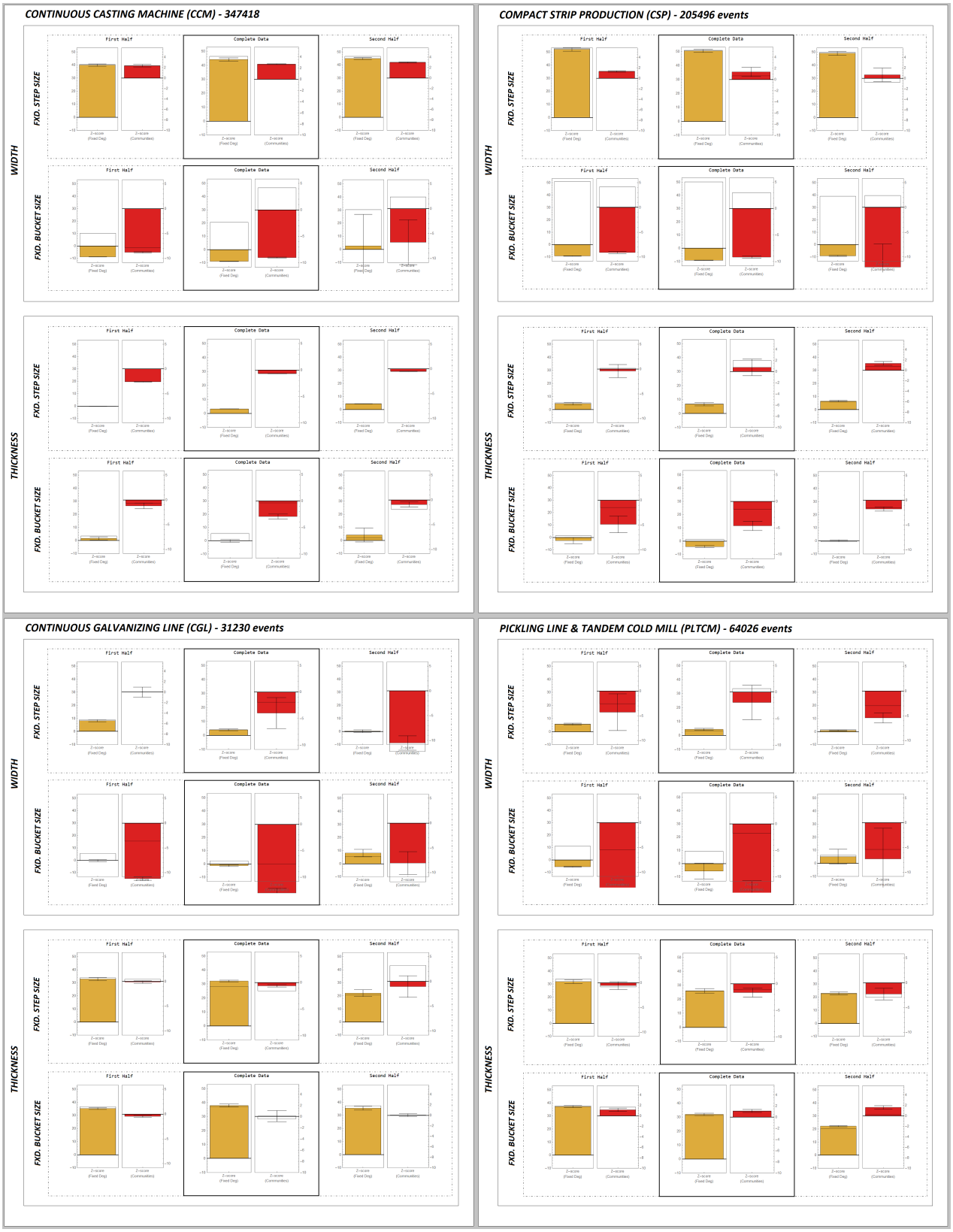


Figure 3.1: Real-life Events Analysis Results.

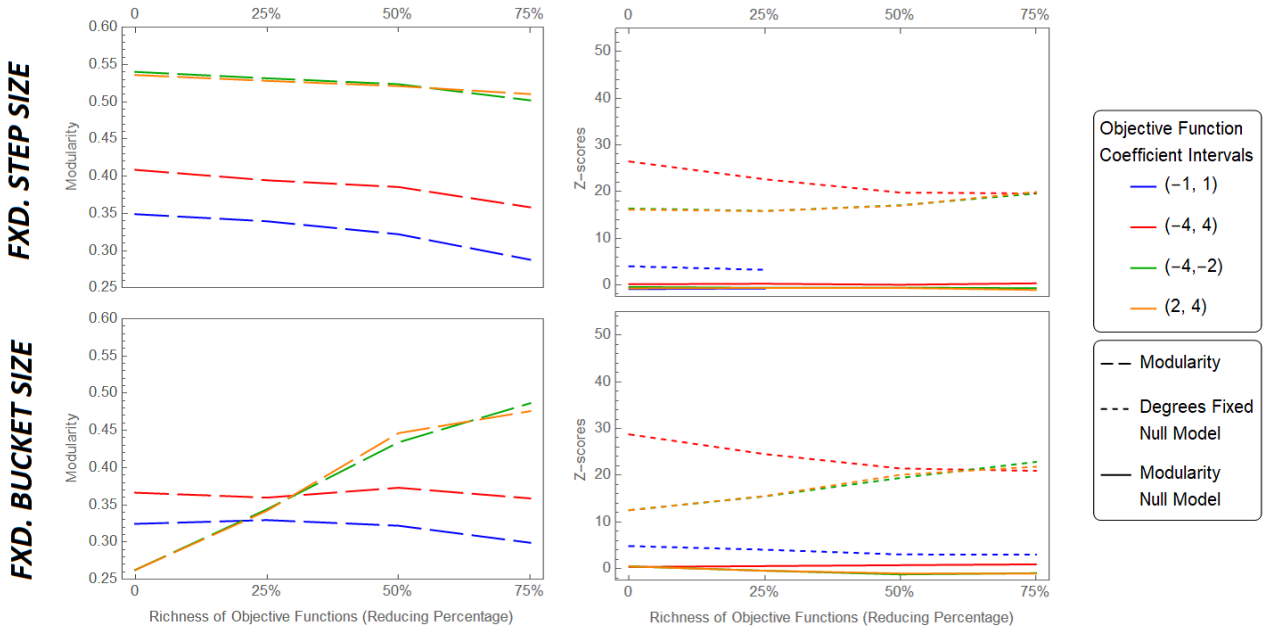


Figure 3.2: FBA Simulation Results.

4 Conclusion And Outlook

FBA is a good control model and can be used with a random graph considering some additional consistency constraints. Need to make sure that the cycles in the graph are suitable to create stuff out of nothing. There are some mass balance constraints that need to be incorporated.

Network perturbation might be another further study subject by upgrading the OR-model with advance tools.

5 Bibliography

[1] P. Cowling, "Design and implementation of an effective decision support system: A case study in steel hot rolling mill scheduling," *Human performance in planning and scheduling*, pp. 217–230, 2001.

[2] 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.

[3] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[6] M. Enders, M.-T. Hütt, and J. M. Jeschke, "Drawing a map of invasion biology based on a network of hypotheses," *Ecosphere*, vol. 9, no. 3, p. e02146, 2018.

[7] "Wikimedia Commons", "The re-drawn chart comparing the various grading methods in a normal distribution. includes: Standard deviations, cumulative percentages, percentile equivalents, z-scores and t-scores. inspired by figure 4.3 on page 74 of ward, a. w., murray-ward, m. (1999). assessment in the classroom. belmont, ca: Wadsworth. isbn 0534527043," 2007. [Online; accessed 27-June-2021].

[8] M. Müller-Linow, C. C. Hilgetag, and M.-T. Hütt, "Organization of excitable dynamics in hierarchical biological networks," *PLOS Computational Biology*, vol. 4, pp. 1–15, 09 2008.

[9] E. Ravasz and A.-L. Barabási, "Hierarchical organization in complex networks," *Phys. Rev. E*, vol. 67, p. 026112, Feb 2003.

[10] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons, 2005.

[11] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "BiGG Models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Research*, vol. 44, pp. D515–D522, 10 2015.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] J. S. Edwards, R. U. Ibarra, and B. O. Palsson, "In silico predictions of escherichia coli metabolic capabilities are consistent with experimental data," *Nature Biotechnology*, vol. 19, pp. 125–130, 2001.

[18] R. Mahadevan and C. Schilling, "The effects of alternate optimal solutions in constraint-based genome-scale metabolic models," *Metabolic Engineering*, vol. 5, no. 4, pp. 264–276, 2003.

[19] J. L. Reed and B. O. Palsson, "Genome-scale in silico models of e. coli have multiple equivalent phenotypic states: Assessment of correlated reaction subsets that comprise network states," *Genome Research*, vol. 14, no. 9, pp. 1797–1805, 2004.

[20] A. P. Burgard, S. Vaidyaraman, and C. D. Maranas, "Minimal reaction sets for escherichia coli metabolism under different growth requirements and uptake environments," *Biotechnology Progress*, vol. 17, no. 5, pp. 791–797, 2001.

6 Supplementary Material

- CCM Production Line for Discrete Time Windows, Fig.xx.
- CCM Production Line for Increasing Time Windows, Fig.xx.
- CCM Production Line for Sliding Time Windows, Fig.xx.
- CCM Production Line, Fig. S1.
- CSP Production Line, Fig. S2.
- CGL Production Line, Fig. S3.
- PLTCM Production Line, Fig. S4.
- FBA Simulation Results with Initial Terms of Objective Functions, Fig. S5.
- FBA Simulation Results with 25% Reduced Terms of Objective Functions, Fig. S6.
- FBA Simulation Results with 50% Reduced Terms of Objective Functions, Fig. S7.
- FBA Simulation Results with 75% Reduced Terms of Objective Functions, Fig. S8.

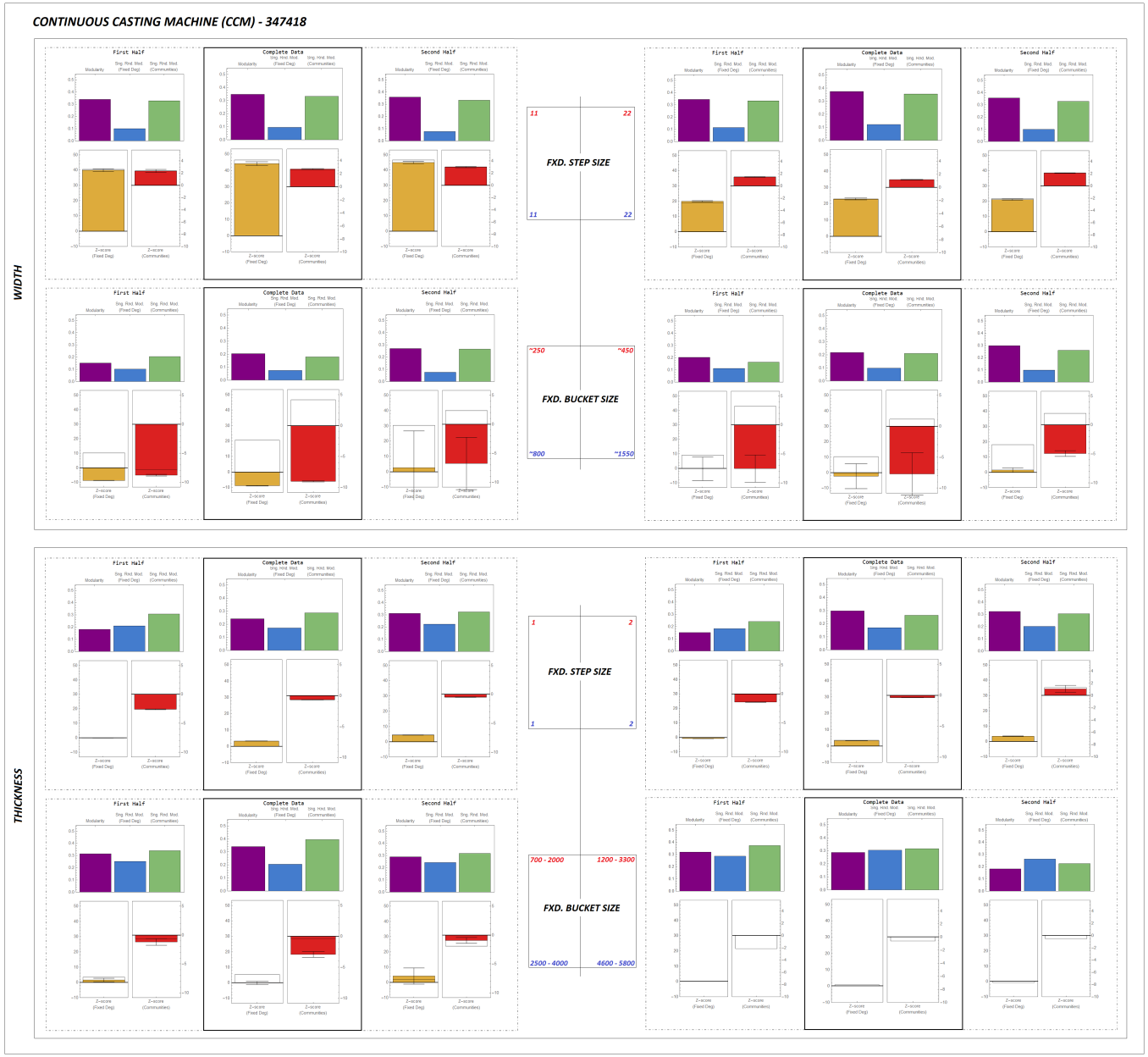


Figure S1: CCM Production Line.

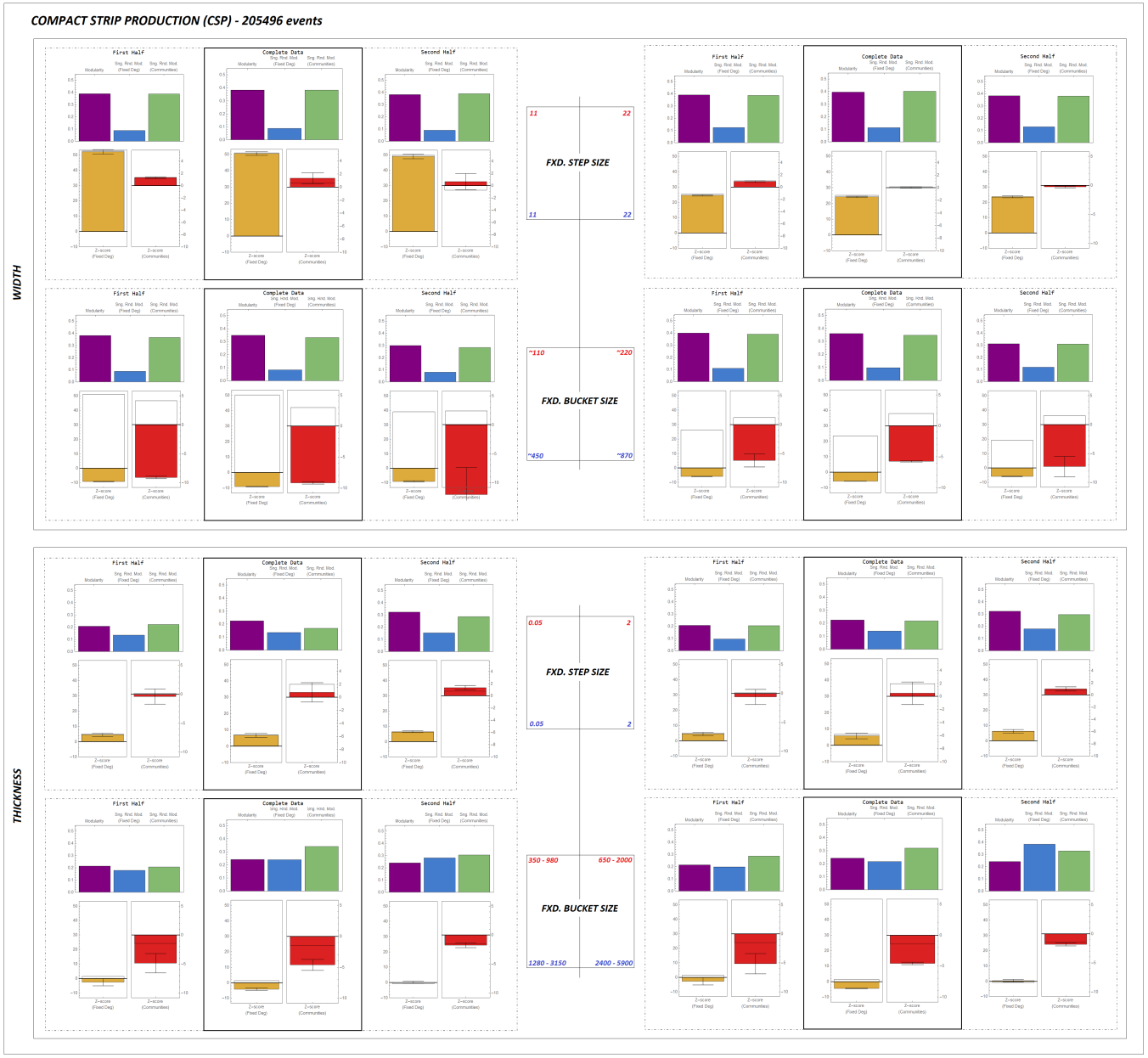


Figure S2: CSP Production Line.

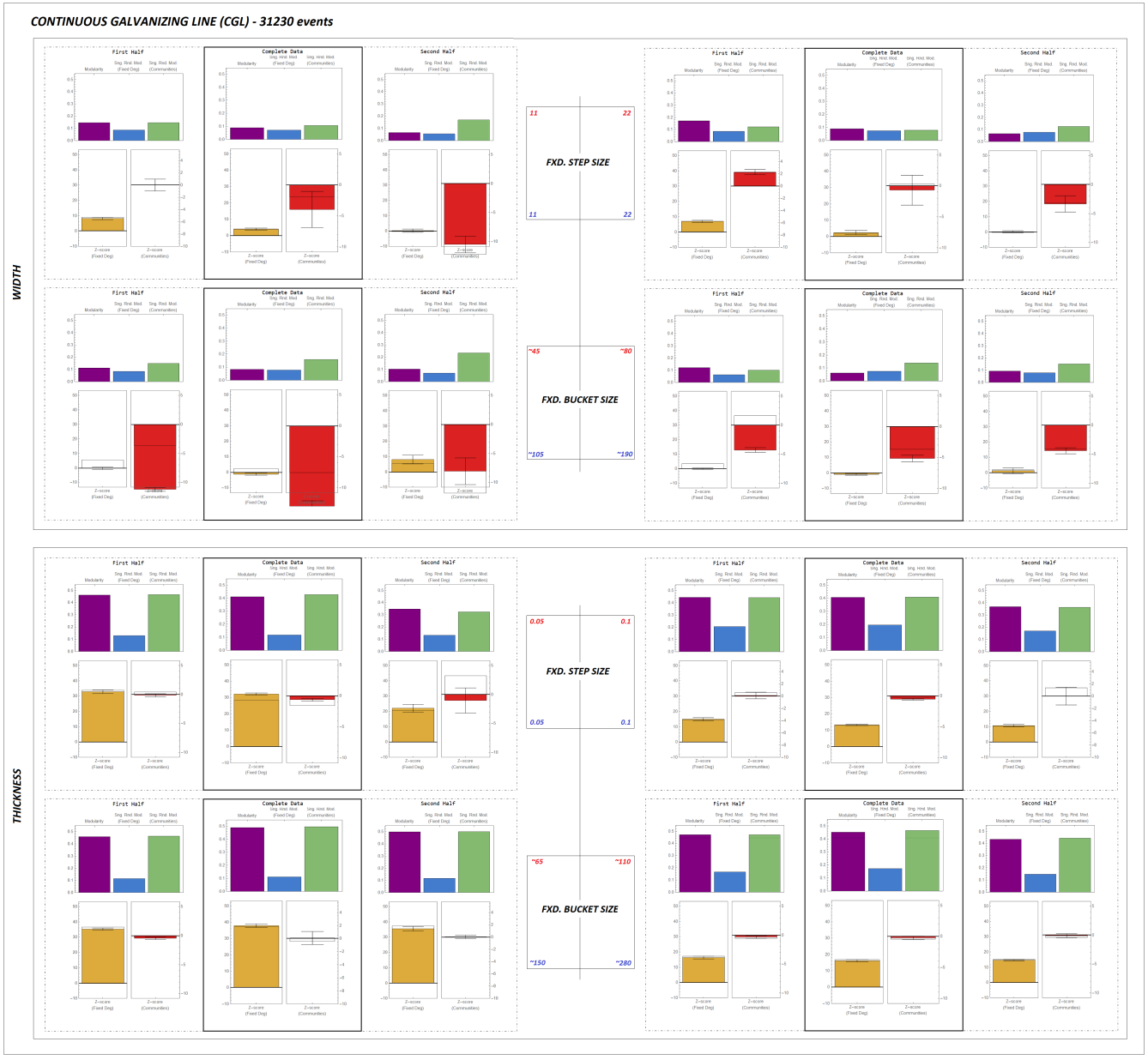


Figure S3: CGL Production Line.

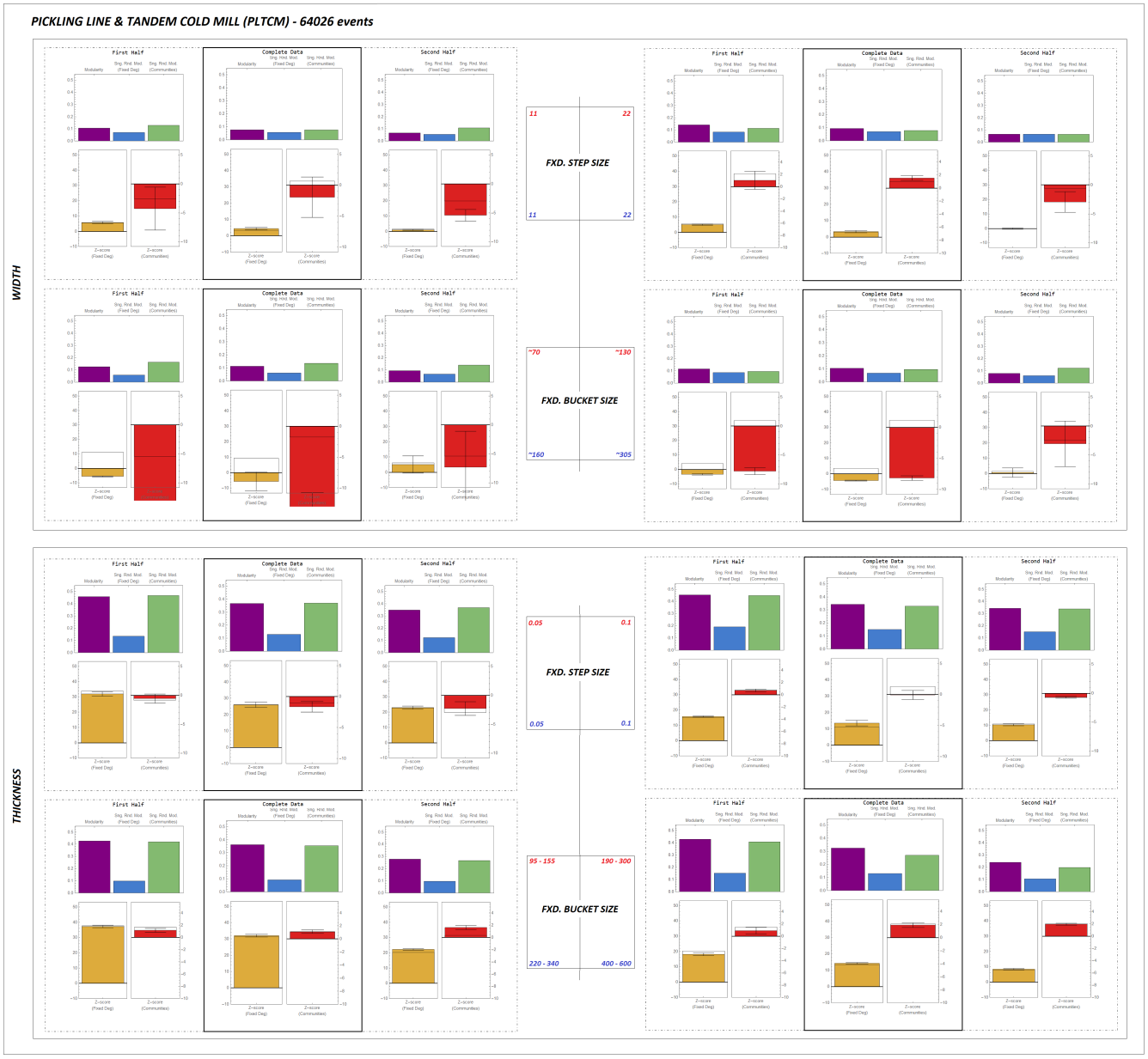


Figure S4: PLTCM Production Line.

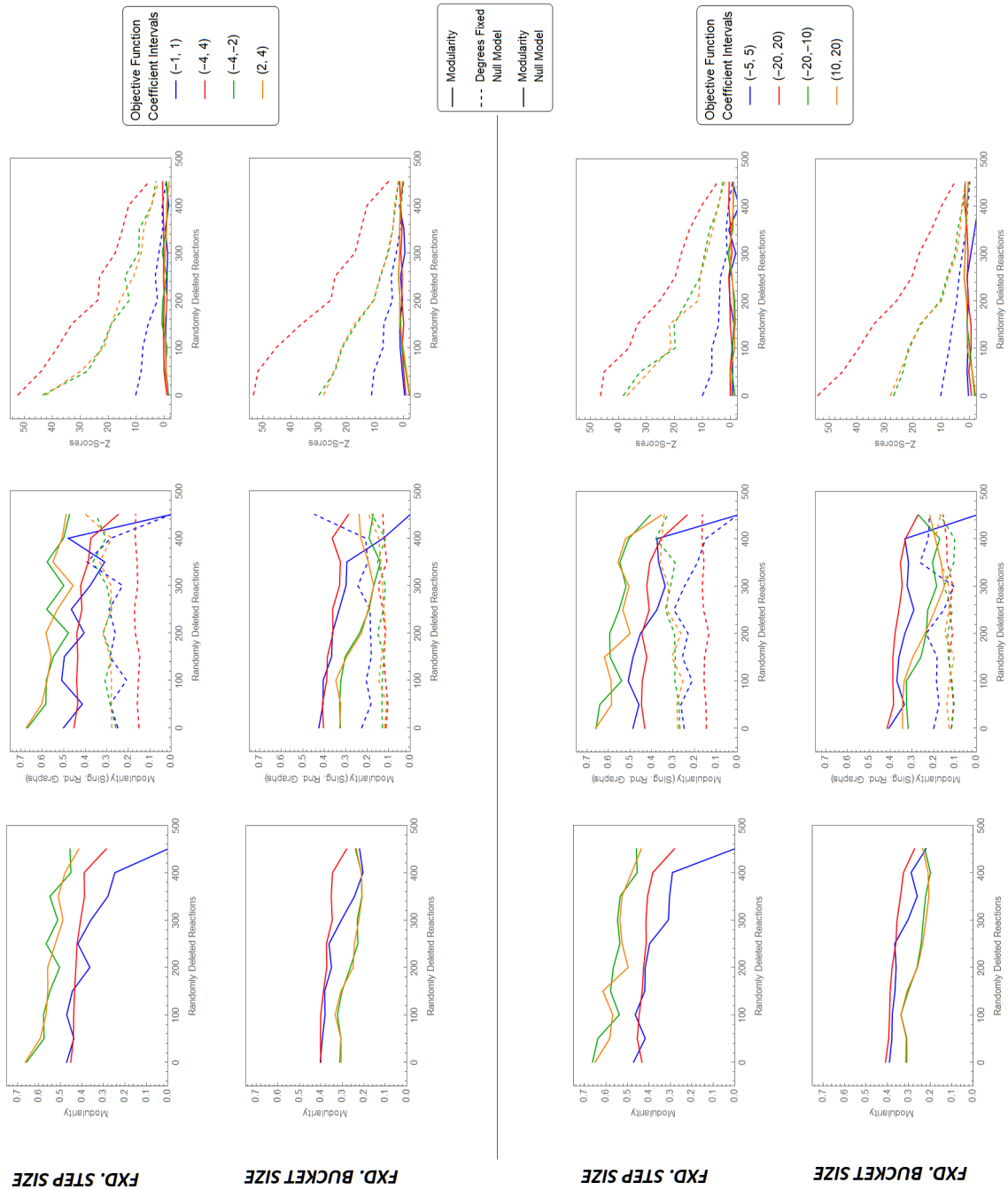


Figure S5: FBA Simulation Results with Initial Terms of Objective Functions.

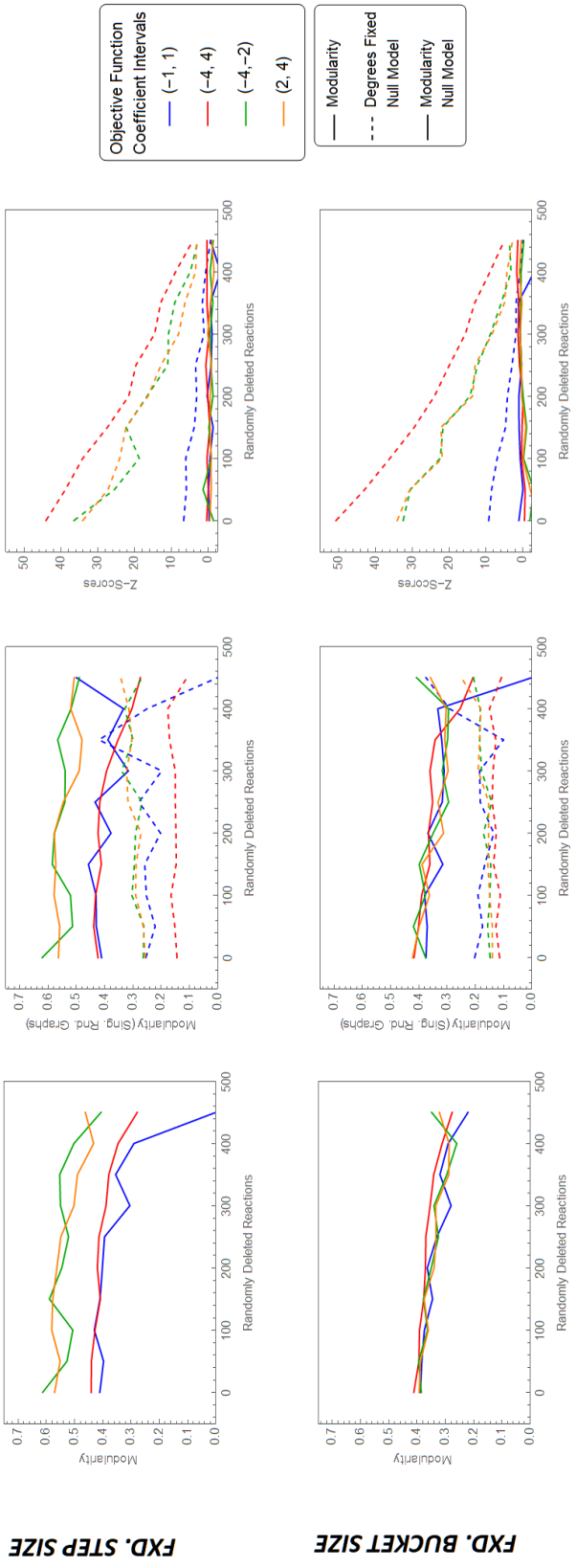


Figure S6: FBA Simulation Results with 25% Reduced Terms of Objective Functions.

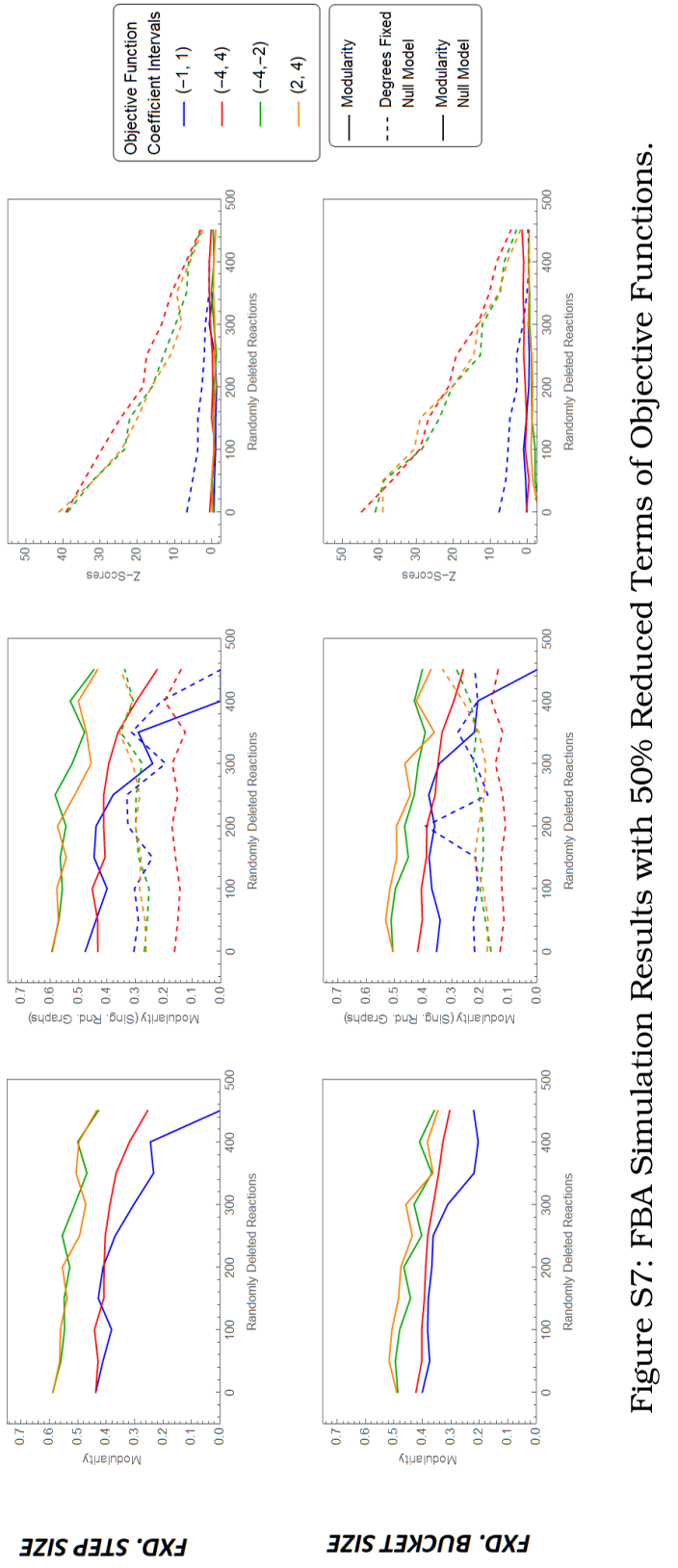


Figure S7: FBA Simulation Results with 50% Reduced Terms of Objective Functions.

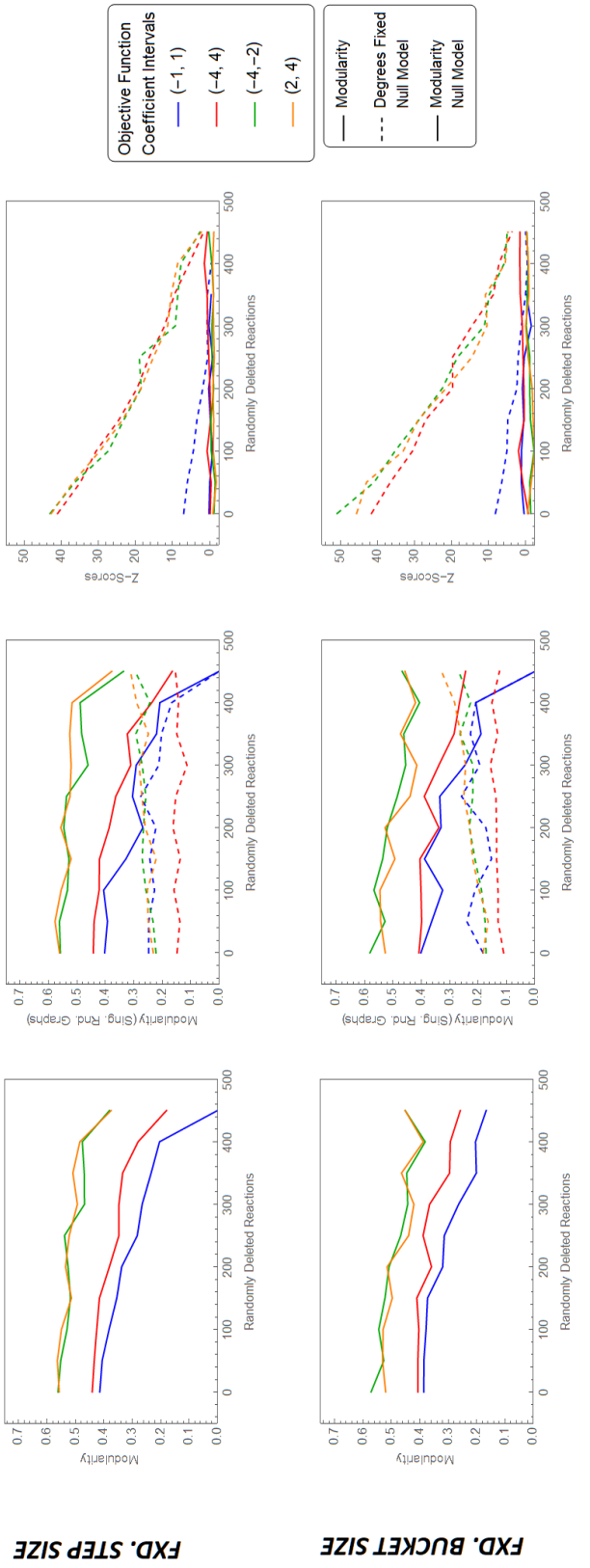


Figure S8: FBA Simulation Results with 75% Reduced Terms of Objective Functions.

List of Figures

List of Tables

List of Equations