BACHELOR'S DEGREE THESIS

Degree in Mathematics

Title: On the use of Integer Programming to pursue optimal Microaggregation

Author: Enrique Spagnolo Arrizabalaga

Advisor: Claudio Gentile

Institute: Consiglio Nazionale delle Ricerche – Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti"

Presenter: Jordi Castro Perez

Department: Statistics and Operations Research

Academic year: 2015/16

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Bachelor's Degree Thesis

On the use of Integer Programming to pursue optimal Microaggregation

Enrique Spagnolo Arrizabalaga

Advisor: Dr. Claudio Gentile

Presenter: Dr. Jordi Castro

Consiglio Nazionale delle Ricerche - Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti"

Preface

This is the memory document of a Final Degree project for the Bachelor Degree in Mathematics at *Universitat Politécnica de Catalunya-BarcelonaTech.* The project itself has been developed during a stay abroad in *Consiglio Nazionale delle Ricerche - Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti" (CNR-IASI)* in Rome, Italy. Furthermore, the research collaboration in IASI will still continue until mid March as agreed before the beginning of the project. Thus, this document reports the work done until the 7th of January 2016. Despite this, by now, I have already achieved several results, both theoretical and practical included in this document. In general terms we could say that, currently, we have an interesting line of research based on a new strategy proposed and described in this document.

The topic of the project is a clustering problem known as Microaggregation, and the main mathematical branch involved is Operations Research, particularly in the field of Integer Programming.

The advisor of the project has been Dr. Claudio Gentile, a researcher in IASI with a deep experience in Operations Research and especially in Integer Programming. The presenter of the project is professor Dr. Jordi Castro in *Universitat Politécnica de Catalunya-BarcelonaTech.* Jordi Castro and Claudio Gentile are usual collaborators and this way professor Castro could introduce me to Claudio in order to bring me the opportunity to develop a project abroad on the field of Operations Research.

The topic of the project was agreed by both researchers since both shared the idea of collaborating in this subject. This way I am taking part as a collaborator in a real research project for an open clustering problem. My contribution might be expected to give a boost to the project that both professors share while at the same time I develop my Final Degree thesis. Besides, it allows me to work on real research with all the advantages and difficulties it might imply.

Having such an opportunity has been very beneficial for me and certainly it has been an instructive experience absolutely worth doing. That is the reason why I would like to include a special section for acknowledgements in this Preface.

Acknowledgements

Of course my first acknowledgements go to Dr. Claudio Gentile. His experience and capabilities are obvious when collaborating with him. Besides, his guide has been essential to advance in the project. In summary Claudio has provided me with all the necessary tools to learn all the knowledge on Integer Programming and Coding I have obtained during the project. In addition, Claudio has strongly helped me achieving the skills to apply all this knowledge on the project. My experience here, for sure, would have not been the same without him.

Secondly, I want to thank the original responsible for this opportunity. Professor Dr. Jordi Castro was very committed since the beginning to bring me an opportunity like this and finally he provided me with this great chance. This is the reason why I must also give him a preference place in the acknowledgements.

Finally I would like to thank CNR-IASI also for this opportunity. Working in an environment like CNR-IASI everyday, sharing the space with more experienced students (the most of them are working on their Ph.D.), has been a great opportunity for me. I have been able to help, discuss and learn with all them when commenting our respective projects.

Abstract

Public use of Microdata files require preprocessing to protect privacy. Microaggregation consist of perturbing the data in order to reach k-anonymity. This is done by aggregating the data of individuals into clusters such that the spread is minimized. In case the problem deals with multivariate data, there is no procedure achieving optimal microaggregation. Besides, this problem is known to be NP-Hard.

This project brings an insight on the main known techniques to achieve good microaggregations, including the state of art on heuristic clustering algorithms as well as on Integer Programming. Besides, it also includes a new strategy based on Integer Programming and Column Generation that at its current stage provides, in any case, with a lower bound on the spread for optimal microaggregation. This strategy breaks the main microaggregation problem into two separate problems: the Master Problem (LP) and the Pricing Problem (BIP). The theory contributions are focused in developing a deep Polyhedral study of the Pricing Problem, including new general families of facet-defining inequalities.

Finally, a CPLEX code has been developed to test the strategy including part of the facet-defining inequalities obtained. The code has been tested on real data typically used in the state of the art and the results encourage us to continue working in this line.

Key words: Microaggregation. Heuristic Algorithms. Integer Programming. Column Generation. CPLEX.

Notation

$\mathbb{N}$	Natural numbers.
$\mathbb{Z}$	Integer numbers.
$\mathbb{Q}$	Rational numbers.
$\mathbb{R}$	Real numbers.
$\mathbb{C}$	Complex numbers.
$\square$	Finish a proof.
$\blacksquare$	Finish an example, finish a Claim proof.

Vector Notation:

$v \in \mathbb{R}^m$	$v^2 = v^T v$.
$u, b \in \mathbb{R}^m$	$u \leq b \Leftrightarrow u_i \leq b_i \ i = 1, \ldots, m$.
$u \in \mathbb{R}^m$	$u \geq 0 \Leftrightarrow u_i \leq 0 \ i = 1, \ldots, m$.
$c, b \in \mathbb{R}^m$	$cb = c^T b = b^T c$.

Graph Notation: Given $G = (V, E)$ a graph. V the set of nodes, E the set of edges.

$v \in V$, $w \in V$, $v \neq w$	(v, w), if exists, notes the edge in E between v and w.
$W \subseteq V$	$\delta(W) = \{(v, w) \in E \mid w \in W, v \in V \setminus W\}$.
$v \in V$	$\delta(v) = \delta(\{v\})$.
$W \subseteq V$	$E(W) = \{(v, w) \in E \mid w \in W, v \in W, \ v \neq w\}$.
$W_i \subseteq V$, $i = 1, \ldots, m$	$E(W_1, \ldots, W_m) = \bigcup_{i=1}^{m} E(W_i)$.
$S \subseteq V$, $T \subseteq V$, $S \cap T = \emptyset$	$[S : T] = \{(v, w) \in E \mid v \in S, w \in T\}$.
$K_n = (V_n, E_n)$	Complete undirected graph of n nodes, $n \geq 1$ integer.

Contents

Introduction

1. Microdata & Privacy

Microdata consists of a file with individuals (people, companies, etc) and attributes for those individuals. Formally a microdata V with s individuals and t attributes can be defined as:

Definition (Microdata): $V : S \subseteq P \rightarrow D(V_1) \times D(V_2) \times \ldots \times D(V_t)$; where P is a population, S is a subset of the population and $D(V_i)$ is the domain of the attribute $i \in \{1, \ldots, t\}$.
In case $|S| < |P|$ we call S a sample of the population. With this definition both categoric and numeric attributes for the individuals are considered.

Microdata files contain statistical information and therefore are usually released for research studies or public use. On account of that, in many cases a great deal of external users have access to this type of files. For this reason, there exists a certain concern about protecting the privacy of the individuals that form the microdata. In other words, microdata should be treated before publication in order to prevent external users from identifying the actual individuals in the microdata file.
We can classify the different attributes based on the risk of external identification:

- **Direct identifiers:** Attributes that directly allow identification of the individual (e.g., ID numbers, phone numbers, etc). They should all be deleted at a first stage.
- **Almost identifiers:** Groups of variables that can be used to identify an individual (e.g., sex, profession, population, etc). For example, if it is known that in a certain village there is only one doctor, then by checking combinations of "profession" and "village" attributes in the microdata, this doctor could be identified.
- **Confidential attributes:** Attributes that contain sensitive information of the individual (e.g., salary, health, etc). If an external user is capable of identifying an individual in the microdata file, then he might have access to this sensitive information. Preventing external identification of the individual becomes therefore a major task.
- **Non-confidential attributes:** Attributes with no sensitive information (e.g., civil state).

External users intending to identify individuals in the microdata are called *attackers*. There exist different risk scenarios all them defined in relation to the information an *attacker* has. The point is that all risk scenarios and measures of identification risks are based on analysing almost identifiers (combinations of different attributes) values in the microdata and determining the probability that an *attacker* manages to identify individuals using information on these almost identifiers. Several measures of this risk exist in the literature based on contingency tables for almost identifiers values and statistics in general, see [**5**], [**1**] for more information.

Methods to prevent identification

There exist several methods to treat microdata files prior to publication in order to avoid that possible attackers could identify the individuals, i.e., to reduce identification risk. All of them are encompassed in the so called *Statistical Disclosure Control* (SDC) topic. All SDC methods can be basically classified between *perturbing* and *non-perturbing* methods:

- **Perturbing** methods modify the microdata values for the attributes of the individuals in order to reduce the risk of identification. The idea is to modify the data in a way that protects the privacy of the individuals and at the same time preserves the information value of the microdata as much as possible (the resulting microdata file should allow researchers or general "good" users to obtain similar conclusions than those that would extract from the original file). Eventually microdata files are published with those modifications. Between the different methods, there stand out the following:
  - **Microaggregation:** The topic of the project. Described afterwards.
  - Rank Swapping: Similar values for different attributes are swapped between individuals.
  - Data shuffling: A more sophisticated Rank Swapping method. See [**14**].
- **Non-perturbing** methods do not modify the data. On the other hand, not all the data is published. For example, *Global Recoding* technique intends to group possible values for categoric attributes into general categories (e.g., group "taxonomists" and "ecologists" as "specialised biologists").

All techniques except for microaggregation are out of the topic of this project.

2. Microaggregation

Microaggregation is a *perturbing* technique mainly considered for numeric variables. Suppose we consider an almost identifier consisting of one or more attributes (e.g. the group formed by the numeric variables "employees", "profits" and "sales" in case of a microdata for companies). Microaggregation technique arises from the concept of *k-anonymity* [**3**].

Definition (*k-anonymity*): Given $k \in \mathbb{N}$, $k \geq 1$, let V be a microdata with s individuals $s \geq k$ and t attributes $V_1, \ldots, V_t$. Let $g = (V_{j_1}, \ldots, V_{j_m})$ be an almost identifier for V, $j_q \in \{1, \ldots, t\}$, $q = 1, \ldots, m$. Then we say V is *k-anonymous* if for every possible value in $D(V_{j_1}) \times \ldots \times D(V_{j_m})$, there exist either 0 or at least k individuals in V with this value for the attributes in g.

Microaggregation intends to modify the values for the attributes involved in a given almost identifier so that eventually the microdata satisfies *k-anonymity* for this considered almost identifier. Therefore, it first joins different individuals of the microdata file in sets of at least k individuals. Then, for every set of individuals, it substitutes the values of the attributes of the given almost identifier by common values for all the individuals in the set. This way *k-anonymity* for the given almost identifier is satisfied by construction in the modified microdata. The resulting sets of individuals will be called **clusters** from now on. A partition of the whole set of individuals into clusters is called a **clustering**. Generally the common values taken for a cluster (after data perturbation) are the values of the centroid of the cluster. That is to reduce as much as possible the information loss, or spread, after the aggregation. With this idea it is clear that the objective of microaggregation technique is to minimize the total sum of distances of the data of the individuals to the centroids of their respective clusters. In practical cases in general the value of k is relatively small (classical microaggregation uses k around 3, see [**6**], [**3**])

Example 1. Let $g = (Employees, Surface)$ be a numeric almost identifier for a microdata of industrial factories. Suppose we want to achieve *k-anonymity* with $k = 2$ and that our microaggregation procedure suggests us to join the 3 factories in Table 1. to form a cluster (size is greater than 2 so 2-*anonymity* would be satisfied). The centroid of this group (average for the values of the attributes of the almost identifier)

Factory	Employees	Surface (m^2)
f_1	55	1410
f_2	48	1205
f_3	41	1120

TABLE 1. Values for the attributes in g for factories f_1, f_2 and f_3 in the original microdata.

is $\frac{55+48+41}{3} = 48$ employees and $\frac{1410+1205+1120}{3} = 1245$ m^2 of surface. Therefore in the eventual published microdata the factories f_1, f_2 and f_3 will all have 48 employees and 1245 m^2 of surface ■.

3. Structure & Guide of the document

The document is structured in four main parts with separate chapters. The first two parts report a general insight on the problem of microaggregation and the most remarkable state of the art about the use of Integer Programming in it. Follows then the report of the contributions this project is doing in this topic.

In more concrete terms. The first part *First Insight in Micro aggregation, Heuristic Clustering Algorithms* introduces the description of microaggregation as a clustering problem and defines important characteristics as the information loss or SSE. Includes also a report on the most common heuristic algorithms to obtain reasonably good solutions and the optimal solution in case of univariate data.

The second part *Integer Programming on Microaggregation (I), State of the Art* introduces the reader to the use of Integer Programming in microaggregation. Reports general theory and includes a model of microaggregation with binary variables as well as an approximation of its objective function to linearize the problem. Finally, it includes a report on the state of art of the use of Integer Programming in microaggregation problem, based on the approximate model.

The third part *Integer Programming on Microaggregation (II), New contributions of the project* reports our contributions to the topic. It includes a description of a new strategy to be applied based on Column Generation. The strategy intends to add iteratively useful clusters of data. To add them it introduces a new Binary Integer Problem known as Pricing Problem. This part also includes a Polyhedral study of this subproblem, bringing new valid and strong inequalities for it.

Finally, the fourth part titled *Implementation of Column Generation & Future Research* describes in general terms the code that has been implemented during the project to test the new strategy proposed. Eventually, a chapter of Future Research is included. There the document describes the lines the project will follow after the submitting of this document. It includes specific ideas to be executed immediately, specially in the code, as well as general ideas to perform.

This document has been written such that anybody with the knowledge of Operations Research from the Bachelor Degree in Mathematics in *Universitat Politécnica de Catalunya-BarcelonaTech* could read and easily understand it. In fact, someone with knowledge on linear optimization in basic terms (not necessarily Integer Programming given that its basic theory is included in the document) and a basic sense on graphs could follow it with no relevant difficulties.

Part 1

First Insight in Microaggregation, Heuristic Clustering Algorithms

Chapter 1
Problem definition

This is a brief introduction chapter to the problem statement of microaggregation. Recall that given $k \geq 1$ an anonymity, microaggregation intends to obtain a clustering of the data with clusters of size at least k, such respecting $k - anonymity$ in the eventual data aggregated. The objective is that, besides, such a clustering minimizes the information loss, or spread, in the data perturbation. In this chapter we introduce a measure of this information loss or spread and therefore define the problem in terms of the data to aggregate.

For simplicity, from now on we consider only the attributes related to an almost identifier. We will also use an abuse of notation when referring to the square of a vector v as $v^2 = v^T v$, for $v \in \mathbb{R}^m$, see *Notation*. On account of all, that we can consider our problem in the following way.
Suppose we have n individuals data vectors a_i, $i \in \{1, \ldots, n\}$, with the attributes of the almost identifier. Suppose $k \geq 1$ the integer that fixes the anonymity. The target is to separate the individuals a_i in clusters with size at least k such that:

$$\sum_{s=1}^{q} \sum_{j=1}^{n_s} d(a_{sj}, \overline{a_s})$$

is minimized, where $d(a, b)$ is a distance between a and b data vectors, q is the number of clusters, a_{sj} is the element j in cluster s and $\overline{a_s}$ is the centroid of cluster s, i.e.:

$$\overline{a_s} = \frac{1}{n_s} \sum_{j=1}^{n_s} a_{sj}$$

Where n_s is the size of cluster s. Note $n_s \geq k \ \forall s \in \{1, \ldots, q\}$ to satisfy $k - anonymity$.

This function introduced above is a general measure of the information loss, or spread, in microaggregation. In general the distance $d(\cdot, \cdot)$ considered is the euclidean distance. For practical reasons, in general we minimize the sum of its square values. This sum is called SSE in [**6**]. We will use this expression as a standard measure of the information loss, or spread, in microaggregation:

$$SSE = \sum_{s=1}^{q} \sum_{j=1}^{n_s} (a_{sj} - \overline{a_s})^T (a_{sj} - \overline{a_s}) = \sum_{s=1}^{q} \sum_{j=1}^{n_s} (a_{sj} - \overline{a_s})^2$$

With the same notation of a_i's described above our clustering problem looks for a SSE as minimal as possible (i.e., a minimal information loss) satisfying that the clusters have size at least k. From now on, we will denote as **feasible clustering** a partition into clusters of size at least k for each of them.

Remark 1. Domingo-Ferrer and Mateo-Sanz in [**6**] proved that the eventual clusters of optimal microaggregation must have size less or equal than $2k - 1$. The proof is very simple and it simply relies on the following fact. Given a feasible clustering π containing a cluster with size greater or equal than $2k$, we can split this cluster into two of size greater or equal than k obtaining a partition π'. It is clear that we can split it reducing the sum of square distances inside this cluster and therefore $SSE_{\pi'} \leq SSE_{\pi}$.

Chapter 2
Clustering Algorithms

From now on, we distinguish the cases of *univariate data* and *multivariate data*. *Univariate data* occurs when the vectors a_i have one single attribute (i.e., the almost identifier has only one variable) and *multivariate data* when there is more than one variable considered.

This chapter explores the most remarkable algorithms to obtain feasible clusterings in microaggregation with low SSE.

1. Univariate data

For the particular case of *univariate data*, optimal microaggregation can be achieved with a polynomial algorithm based on minimal paths in a weighted graph. The method was provided by Hansen and Mukherjee in [**12**]. The method relies on two properties proved in [**6**]. The first one has already been stated: optimal microaggregation has clusters of sizes between k and $2k-1$. The second one claims that in *univariate data* clusters must be **consecutive**. This means that after ordering the data points by value a_i (decreasing or non-decreasing order), the eventual clusters cannot be interleaved. In other words: if C_1 and C_2 are two separate clusters in an optimal univariate microaggregation, then for each x, y with $x < y$ and both in C_1, there exists no $z \in C_2$ with $x \leq z \leq y$. It turns out to be very intuitive as consecutive clusters are less spread and therefore SSE is lower for clustering in consecutive groups.
The optimal algorithm looks for consecutive clusters with sizes between k and $2k-1$ and is explained below.

Algorithm for optimal *Univariate* Microaggregation [12]

Let $G = (V, E)$ be a weigthed graph with the nodes $V = \{0, 1, \ldots, n\}$ being the data in increasing order plus an extra node 0 as an origin. The edges E are formed as follows:
Given two nodes i, j in V with $i < j$, we consider the edge $e = (i, j)$ if and only if $i + k \leq j \leq i + 2k \leq n$ (i.e. if $i + 1$ and j can be in a feasible cluster of size between k and $2k - 1$ in an optimal solution). The weight for this edge e then would be the contribution to the SSE of a cluster formed by the nodes $\{i + 1, \ldots, j\}$. We would take the values in this set and compute the sum of square distances to the centroid of this set. This would directly be the weight of $e = (i, j)$.
The resulting weighted graph G would lead to a bijection between consecutive feasible clusterings with cluster sizes between k and $2k - 1$ and paths between node 0 and node n. This bijection is described as follows:
If we consider a path $0 \to i_1 \to i_2 \to \ldots \to i_m \to n$ then the corresponding consecutive feasible clustering is $\{\{1, \ldots, i_1\}, \{i_1 + 1, \ldots, i_2\}, \ldots, \{i_m + 1, \ldots, n\}\}$. It is clear that it is a feasible clustering of $\{1, \ldots, n\}$ and that it is consecutive, for construction. Clearly given a feasible consecutive clustering we could reverse this correspondence and obtain its associated path.
Furthermore note that the sum of the weights of a path taken directly provides the SSE corresponding to the clustering associated with this path. Consequently a minimal path from node 0 to n in this computed graph would provide us with an optimal solution that minimizes total SSE by taking its associated consecutive clustering as described above.
In order to determine the complexity of this algorithm we note that the graph construction requires sorting the

nodes ($\mathcal{O}(n \log n)$) and computing kn edges at most, k edges for each node. Every edge weight computation requires $\mathcal{O}(k)$ operations. Therefore the cost for the graph construction is $\max\{\mathcal{O}(n \log n), \mathcal{O}(k^2 n)\}$. We have to add then the cost of a minimal path algorithm. Dijkstra with a priority queue for example would be $\mathcal{O}(kn \log n)$ for this graph. The result is clearly that the algorithm has a polynomial cost and moreover it is very efficient even for large data sets.

2. Multivariate data

In the case of multivariate data, there is no total order anymore and that means that the property of consecutive clusters has sense no more. The only property from [**6**] we can still exploit is that cluster size in an optimal solution must be between k and $2k-1$. Nevertheless at this stage we must recall the result from [**7**] that claims that *multivariate* optimal microaggregation is an *NP-Hard* problem. That means optimal multivariate microaggregation should not be expected to be solved in polynomial time. On account of that, there exists a lot of literature about heuristic algorithms that obtain feasible clusterings with reasonable SSE. Below we introduce the most remarkable ones, provided by [**3**] and [**13**]:

(1) **Maximum Distance (MD)**: Let $S = \{a_1, \ldots, a_n\}$ be the set of all the data, $a_i \in \mathbb{R}^m$, $m \geq 2$.
   (i) Find a_i and a_j the most distant values in S.
   (ii) Find the $k-1$ closest points to a_i in S and form a cluster of size k with them and a_i. Same for a_j. Remove the two clusters from S.
   (iii) If $|S| > 2k$ return to (i).
   (iv) If $k \leq |S| \leq 2k-1$ form a cluster with those points in S and STOP.
   (v) If $|S| < k$ add every remaining value in S to the cluster with the closest centroid and STOP.

   This provides a feasible clustering with a computational cost of $\mathcal{O}(n^3)$ approximately ($\lfloor n/2k \rfloor$ steps and $\mathcal{O}(n^2)$ to find the most distant values at every step, in major terms).

(2) **Maximum Distance to Average Value (MDAV)**: This is a computational improvement of MD algorithm. With the same notation as in MD, we begin with the set $S = \{a_1, \ldots, a_n\}$.
   (i) Find r the centroid of S and a_i the most distant value from r in S. Find a_j the most distant value from a_i in S.
   (ii) Find the $k-1$ closest values to a_i in S and form a cluster of size k with them and a_i. Same for a_j. Remove the two clusters from S.
   (iii) If $|S| > 2k$ return to (i).
   (iv) If $k \leq |S| \leq 2k-1$ form a cluster with those points in S and STOP.
   (v) If $|S| < k$ add every remaining value in S to the cluster with the closest centroid and STOP.

   This provides a feasible clustering too. The difference with MD is that at every instance it does not find the furthest values in S (quadratic cost). Instead, it computes a centroid with cost $\mathcal{O}(n)$ at most, finds its furthest value ($\mathcal{O}(n)$) and again the furthest value to this last one with cost $\mathcal{O}(n)$ at most too. Eventually at every step it has a cost of $\mathcal{O}(n)$ at most and there are $\lfloor n/2k \rfloor$ steps so the total cost is $\mathcal{O}(n^2)$ at most.

(3) **Variable-Maximum Distance to Average Value (V-MDAV)** [**13**]: This is a variation from MDAV algorithm. In the previous introduced algorithms most of the clusters had fixed size k and only in the final step the possibility of creating clusters with different size was considered. With V-MDAV the philosophy changes. It is exactly the same procedure as in MDAV but in the stage where we look for the $k-1$ closest values to a certain point and we form a cluster we can consider the possibility of amplifying this cluster. For this extension of a group of size k, we suppose we have such a cluster N of size k with a certain a_i and its closest values. Suppose we want to extend N.
   (i) We find a_{in} the closest point in $S \setminus N$ to any of the points in N and d_{in} its distance to its closest point in N.
   (ii) We find a_{out} the closest point in $S \setminus (N \cup \{a_{in}\})$ to a_{in} and d_{out} its distance to a_{in}.
   (iii) If $d_{in} < \gamma d_{out}$ then we add a_{in} to N and we come back to (i) until N has size $2k-1$ or cannot be extended.

   The value of γ is a gain factor discussed in the literature about this method [**13**].

Experimental results discussed in [**3**] show that V-MDAV outperforms the two previous ones in synthetic grouped data and also that MDAV and MD have similar performance in general although MDAV has a lower computational cost.

Part 2

Integer Programming on Microaggregation (I), State of the Art.

Chapter 3
Integer Programming Theory

In this chapter we are introducing the most remarkable theory from [**16**] on Integer Programming. This theory will be later applied to microaggregation. However, by now we are putting our problem aside to introduce this basic theory.

Before moving to some general definitions we will give a first view of what is the idea in Integer Programming. Briefly, it is solving optimization linear problems in discrete sets.
In general terms, a linear Integer Programming Problem is an optimization problem of the form:

$$\begin{aligned} \min_{x} \quad & cx \\ & Ax \leq b \\ & x \in X \end{aligned}$$

Where $X \subseteq \{0,1\}^n \times \mathbb{Z}^q \times \mathbb{R}^p$. Let $s = n + q + p$. Then c, x and are vectors of size s, A is a matrix of size $m \times s$ and b a vector of size m. The product cx denotes implicitly the euclidean scalar product, see *Notation*. Vectors c, b and matrix A are given, whereas vector x corresponds to the variables. The relation $\leq$ for a vector here means that it is satisfied for every component, see *Notation*. In this particular form it is a minimization problem. This form of presenting an optimization problem is called model.

We will start with some general definitions for Mathematical Programming and particularly oriented to Integer Programming.

Definition (Polyhedron): A subset $P \subseteq \mathbb{R}^n$ described by a finite set of linear inequalities, $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, is a *polyhedron*.

From now on, to simplify the writing, we will make an abuse of notation by referring to a linear inequality simply as an inequality.

Definition (Formulation): A polyhedron $P \subseteq \mathbb{R}^{n+q+p}$ is a *formulation* for a set $X \subseteq \{0,1\}^n \times \mathbb{Z}^q \times \mathbb{R}^p$ if and only if $X = P \cap (\{0,1\}^n \times \mathbb{Z}^q \times \mathbb{R}^p)$.

With this definition, we consider first the combinations when $n = 0$. In this case, if $p > 0$, $q > 0$ we obtain a Mixed Integer Problem (MIP, integer and real variables). Else if $p = 0$, $q > 0$ we are talking about an Integer Problem (IP, only integer variables). Otherwise, if $q = 0$, $p > 0$ it is a Linear Problem (LP, only real variables).
Whenever $n > 0$, our problem includes binary variables. In particular, if $n > 0$ and $q = p = 0$ we are considering a Binary Integer Problem (BIP). This will certainly be our case for microaggregation.

A formulation is therefore only the polyhedron described by the different inequalities which, combined with the "type condition" constraint for the variables (integer, binary, mixed, etc), give us the constraints of the problem.
Note that solving the linear problem using its formulation as the polyhedron to explore is solving the so called **linear relaxation** of the problem.

The main idea of Integer Programming is to obtain *good* formulations for our problems. At this point we should emphasize on how a formulation is *finer* or *better* than another.

Definition: Given P_1 and P_2 two formulations for a same set X, we say P_1 is *finer* than P_2 if and only if $P_1 \subset P_2$.

For example, for a BIP of dimension two, in Figure 1., three formulations P_3, P_2, P_1 satisfying the chain $P_3 \subset P_2 \subset P_1$ are represented. The set of feasible points is $X = \{(0,0),(0,1),(1,0),(1,1)\}$.

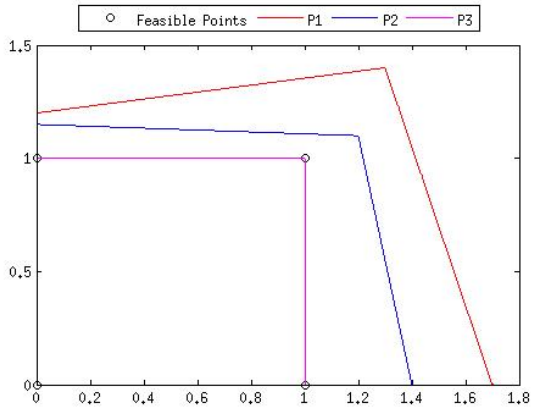

FIG. 1. X marked in black cercles, P_1 is the *polyhedron* interior to the red line. P_2 is interior to the blue line and P_3 is interior to the pink line.

Furthermore, in case of Figure 1., P_3 is not only the finest formulation of the three introduced but is also clearly finer than any other possible formulation for X.
This introduces us the concept of Convex Hull. The Convex Hull of a set X is the minimal polyhedron that contains X, in this case P_3. Note that it can be obtained by all the possible convex combinations of points in X.

Definition: Given $X \subseteq \mathbb{R}^n$, the Convex Hull of X, noted as $conv(X)$, is defined as:

$$\text{conv}(X) = \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^{t} \lambda_i x^i,\ \sum_{i=1}^{t} \lambda_i = 1,\ \lambda_i \geq 0 \text{ for } i = 1,\dots,t;\ \forall\ x^1,\dots,x^t \in X,\ t \geq 1\}$$

Remark 2. The Convex Hull of a set $X \subseteq \mathbb{R}^n$ is a polyhedron in $\mathbb{R}^n$. It has the property that all its extreme points lie in X.
This means that given a Convex Hull formulation for an Integer Programming problem we can obtain its optimal solution simply by applying the Simplex Algorithm. That is the reason why we call $\text{conv}(X)$ the *ideal* formulation for X. Nevertheless, in many times it cannot be expected to obtain the Convex Hull formulation. Simplex is polynomial time on average so if a problem is *NP-Hard* it might not be expected to find the Convex Hull formulation of the problem, or at least not to obtain it in polynomial time. However it can be useful to pursue *finer* formulations in order to make the problem approachable by a Branch & Bound method for example. We include a brief insight on this method in the paragraph below.

Branch & Bound

Branch & Bound method is a *divide and conquer* procedure. It divides iteratively the polyhedron of the problem until and at every stage it solves the corresponding linear relaxation. If the solution is feasible (belongs to X) then it stops the exploration of this subset of the original polyhedron. After all the successive divisions, it ends up constructing an exploration tree of subproblems. This procedure is optimized by using incumbents that can cut exploration of subproblems.
For example, given a minimization BIP with variables x_i, we start by taking its linear relaxation (adding

the bound constraints $0 \leq x_i \leq 1$). Then suppose we find x^* a fractional optimal solution. Then, there exist at least one variable $0 < x_j^* < 1$ fractional. The problem is then divided into two subproblems, one after adding the constraint $x_j = 0$ and the other after adding $x_j = 1$. This procedure is done recursively for every subproblem, what leads to a tree of subproblems as mentioned before. If a given linear relaxation provides with an integer solution, then the exploration of this subproblem finishes. Besides, its objective value is an upper bound, denoted incumbent, for the problem solution (it comes from a feasible solution). Thus, every subproblem exploration in the tree of subproblems with a greater linear relaxation solution can be directly cut.
In case the problem is an IP, the procedure is based on the same idea of recursivity. Given a fractional linear relaxation solution, there must be a variable x_j^* fractional. Then, the problem can be split into two by adding either $x_j \geq \lceil x_j^* \rceil$ or $x_j \leq \lfloor x_j^* \rfloor$.
Branch & Bound is usually combined with the search of inequalities that can bound the problem. These combinations are in general known as Branch & Cut methods. See chapters 7, 8 and 9 in [**16**] for deeper information.

1. Valid Inequalities and Cutting Plane Methods

Chapters 8 and 9 in [**16**] are the source for this section. As stated before a target in Integer Programming is obtaining finer formulations to make the problem approachable by standard methods. Those formulations are obtained by adding inequalities to the formulation. However when adding those inequalities the resulting polyhedron must remain a formulation. In other words, the inequalities must be *valid*.

Definition: Given $X \subseteq \mathbb{R}^n$, $\pi \in \mathbb{R}^n$, $\pi_o \in \mathbb{R}$. Then, $\pi x \leq \pi_o$ is a *valid inequality* for X if and only if $\pi x \leq \pi_o$ for all $x \in X$.

The idea of bounding an Integer Programming problem is then to obtain families of valid inequalities and add them in order to obtain finer formulations. However, sometimes families of valid inequalities are exponential in number and thus sometimes even unfeasible to generate. In this cases we might add iteratively some of the inequalities in the family. That is what is known as a **cutting plane method**.
A cutting plane method follows this procedure. Suppose that $\mathcal{F}$ is a family of valid inequalities for a given problem to be solved in $X \subseteq \mathbb{R}^n$:

Cutting Plane Algorithm:

(i) Set P an initial formulation.
(ii) Solve the linear relaxation of the problem in P and let x^* be the optimal solution.
(iii) If $x^* \in X$ stop with optimality.
(iv) If $x^* \notin X$, find a valid inequality $\pi x \leq \pi_o$ in $\mathcal{F}$ such that $\pi x^* > \pi_o$. Add this inequality to the formulation, i.e., set $P = P \cap \{x \mid \pi x \leq \pi_o\}$. Return to (ii).

The proces of finding a valid inequality $\pi x \leq \pi_o$ in $\mathcal{F}$ violated by x^* in (iv) is called *separation routine*. Note that the scheme of a cutting plane algorithm also includes the possibility to solve the problem with optimality. However the main point in this scheme is that if the algorithm terminates without finding an optimal solution the resulting polyhedron P is a finer formulation than the original.

Example 2. A widely known cutting plane scheme is the Gomory Cut method for integer problems. It is applied in Integer Problems (IP). In this case the family of valid inequalities considered are the *Chvátal-Gomory inequalities*. See chapter 8 in [**16**].

Given $X \subseteq \mathbb{Z}^n$ and $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ a formulation, with A matrix of size $m \times n$, $b \in \mathbb{R}^m$. Let $a_j \in \mathbb{R}^m$ the columns of A, $j = 1, \ldots, n$. Let $u \in \mathbb{R}^m_+$. Then the inequality:

$$\sum_{j=1}^{n} u a_j x_j \leq ub$$

is valid since $u \geq 0$. Furthermore, as the solution $x \in X$ must be integer the inequality

$$\sum_{j=1}^{n} \lfloor ua_j \rfloor x_j \leq \lfloor ub \rfloor$$

must be valid too. This last is known as the Chávtal-Gomory inequality for a given $u \geq 0$.

The Gomory Cut method considers B^{-1} the inverse of the submatrix B in A (extended with slack variables if necessary for the standard form) corresponding to the basic variables x_B of an optimal solution for a linear relaxation. Note that B has size $m \times m$. If the solution is not feasible there exists a component ν in x_B such that ${x^*}_{B_\nu}$ is not integer. Let β be the row ν in B^{-1} and $q = \beta - \lfloor \beta \rfloor \geq 0$. Then the cut added by the method -written in terms of the original variables (before adding slack variables to obtain the standard form)- is:

$$\sum_{j=1}^{n} \lfloor qa_j \rfloor x_j \leq \lfloor qb \rfloor$$

which clearly corresponds to a Chávtal-Gomory inequality for $q = \beta - \lfloor \beta \rfloor \geq 0$, considering the formulation without the slack variables ■.

2. Strong valid inequalities

In this section, we will consider valid inequalities for a given polyhedron, rather than a set. Note that if we consider the case this polyhedron is a formulation for a given discrete set in an Integer Programming problem, then the valid inequalities for this set are still valid for its formulation. Then, the definitions and results below are also extendable to this case in Integer Programming. In fact, they are mainly conceived for this case.

An important issue regarding new valid inequalities to be added is how *strong* these inequalities are. This leads us to the concept of **dominance** between inequalities.

Definition: Given two valid inequalities $\pi x \leq \pi_o$ and $\mu x \leq \mu_o$ for $P \subseteq \mathbb{R}^n_+$, $\pi x \leq \pi_o$ *dominates* $\mu x \leq \mu_o$ if there exists $u \in \mathbb{R}$, $u > 0$, such that $\pi \geq u\mu$, $\pi_o \leq u\mu_o$ and $(\pi, \pi_o) \neq (u\mu, u\mu_o)$.

Note that if $\pi x \leq \pi_o$ dominates $\mu x \leq \mu_o$, then $\{x \in \mathbb{R}^n_+ \mid \pi x \leq \pi_o\} \subseteq \{x \in \mathbb{R}^n_+ \mid \mu x \leq \mu_o\}$.
With this definition we can state that a given valid inequality $\mu x \leq \mu_o$ is *redundant* for the description of a polyhedron $P = \{x \in \mathbb{R}^n_+ \mid a_j x \leq b_j, j = 1, \ldots, m\}$ if linearly combining inequalities in the description of P we can obtain an inequality that dominates $\mu x \leq \mu_o$.
With the idea of redundancy of a given valid inequality introduced we can start discussing about the concepts of *affine independence* of points as well as *full dimensionality*, *faces* and *facets* of polyhedrons.

Definition: Given the points $x_1, \ldots, x_k \in \mathbb{R}^n$, we say they are *affinely independent* if the $k-1$ vectors $x_2 - x_1, \ldots, x_k - x_1 \in \mathbb{R}^n$ are linearly independent.

Definition: Given $P \subseteq \mathbb{R}^n$ a polyhedron. The dimension of P, noted as $\dim(P)$, is one less than the maximum number of affinely independent points in P.

Remark 3. Given $P \subseteq \mathbb{R}^n$ a polyhedron, we can define the *directions* in P as the set of vectors $x^i - x^j \in \mathbb{R}^n$ for pairs x^i, $x^j \in P$. Then, the dimension of P corresponds to the maximum number of linearly independent directions in P.
Note that P is contained in $\mathbb{P} \subseteq \mathbb{R}^n$ an affine subspace, such that $\dim(\mathbb{P}) = \dim(P)$. Let us use all linearly independent directions in P as a base to construct a vector space $\mathbb{V}$. Clearly $\dim(\mathbb{V}) = \dim(P)$ by definition. If we take $x^i \in P$ arbitrary, then the affine subspace $\mathbb{P} = x^i + \mathbb{V} = \{x \in \mathbb{R}^n \mid x = x^i + v,\ v \in \mathbb{V}\}$ contains P and $\dim(\mathbb{P}) = \dim(\mathbb{V}) = \dim(P)$. The difference between P and $\mathbb{P}$ is that P can be bounded.
Later in this writing, we will refer to the orthogonal of P as the orthogonal of $\mathbb{V}$ the vector space defined by its directions. Note that this orthogonal has dimension $n - \dim(P)$ if $P \subseteq \mathbb{R}^n$.

Definition: A polyhedron $P \subseteq \mathbb{R}^n$ is *full dimensional* if and only if $\dim(P) = n$.

THEOREM 3.1. *If P is a full dimensional polyhedron, it has a unique minimal description:*

$$P = \{x \in \mathbb{R}^n \mid a_i x \leq b_i \text{ for } i = 1, \ldots, m\}$$

where each inequality is unique to within a positive multiple.

The sense of minimal means that every inequality is *necessary* and that by removing just one of them in the description, the polyhedron is no longer P. It also implies that any other existing valid inequality which is not a positive multiple of one of them, is *redundant* for this description of P.

Definition (face): Given $P \subseteq \mathbb{R}^n$ a polyhedron, then $F \subset P$ defines a face of P if and only if $F = \{x \in P \mid \pi x = \pi_o\}$ for a valid inequality $\pi x \leq \pi_o$ of P. In this case the inequality $\pi x \leq \pi_o$ is said to define F (*face-defining inequality*).

Note that a face is a polyhedron.

Definition (facet): Given $P \subseteq \mathbb{R}^n$ a polyhedron, then $F \subset P$ is a *facet* of P if and only if F is a face of P and $\dim(F) = \dim(P) - 1$. In this case if $\pi x \leq \pi_o$ defines F then this inequality is said to be facet-defining.

PROPOSITION 3.2. *Given P a polyhedron, let $\{x \in \mathbb{R}^n \mid a_i x \leq b_i \text{ for } i = 1, \ldots, m\}$ be a description of P with no redundancy. Then there is a one to one correspondence between inequalities $a_i x \leq b_i$ and facets of P.*

Remark 4. In case P is full dimensional, then its unique minimal description corresponds to the set of all facet-defining inequalities. That is to say that a valid inequality $\pi x \leq \pi_o$ for P is necessary in the description of P if and only if $\pi x \leq \pi_o$ is facet-defining.

Bearing all this in mind, facet-defining inequalities are for sure the strongest valid inequalities considerable for a given polyhedron. Furthermore, if X is the set of all feasible points for an integer problem, an ideal Integer Programming approach is to consider facet-inequalities for $\text{conv}(X)$ and apply a cutting plane method adding them iteratively. In case all facet-defining inequalities for $\text{conv}(X)$ are known, considering them directly would provide with a no redundant description of $\text{conv}(X)$. In case of full dimensionality, this not only would provide the ideal formulation for the problem but also the unique minimal description of this formulation. In any case the most common approach for an Integer Programming problem is to look for valid inequalities, especially if they are facet-defining, and considering a cutting plane method to iteratively add these inequalities. If the cutting plane method does not provide with an optimal solution in any case it provides with a finer formulation that can possibly be the initialization for a Branching method.

Chapter 4
Models for Microaggregation in Integer Programming

In this chapter we are going to model microaggregation with binary variables. In section *1. General Model*, we will formulate the problem in general, without considering any approximation. It results unfortunately in a non-linear problem so it will follow an approximation linearising it in section *2. Approximate Model*.

1. General Model

The original microaggregation problem can be modelled with binary variables if the contribution of a cluster to the total SSE of microaggregation is expressed in a different way. First of all, recall the problem statement notation in *Chapter 1*.

PROPOSITION 4.1. *Given a cluster S containing n_s elements denoted by $\{a_{sj} \mid j \in \{1,\dots,n_s\}\}$ following the notation of Part 1. Let $\overline{a_s}$ be its centroid. Then,*

$$n_s \sum_{j=1}^{n_s} (a_{sj} - \overline{a_s})^2 = \frac{1}{2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} (a_{si} - a_{sj})^2$$

PROOF. Developing the square product and substituting $\overline{a_s}$ the left hand side equals to:

$$n_s \sum_{j=1}^{n_s} a_{sj}^2 + n_s \sum_{j=1}^{n_s} \frac{1}{n_s^2} (\sum_{i=1}^{n_s} a_{si})^2 - 2n_s \sum_{j=1}^{n_s} (a_{sj} \frac{1}{n_s} \sum_{i=1}^{n_s} a_{si})$$

In particular, the term

$$n_s \sum_{j=1}^{n_s} \frac{1}{n_s^2} (\sum_{i=1}^{n_s} a_{si})^2 = \frac{1}{n_s} \sum_{j=1}^{n_s} (\sum_{i=1}^{n_s} a_{si})^2 = (\sum_{i=1}^{n_s} a_{si})^2$$

as $(\sum_{i=1}^{n_s} a_{si})^2$ does not depend on j as is added n_s times. Finally substituting $(\sum_{i=1}^{n_s} a_{si})^2 = \sum_{j=1}^{n_s} \sum_{i=1}^{n_s} a_{sj} a_{si}$ we obtain that the left hand side is equal to:

$$n_s \sum_{j=1}^{n_s} a_{sj}^2 - \sum_{j=1}^{n_s} \sum_{i=1}^{n_s} a_{sj} a_{si}$$

For the right hand side simply substituting the square product and using the same argument

$$\frac{1}{2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} a_{si}^2 = \frac{n_s}{2} \sum_{i=1}^{n_s} a_{si}^2 \quad \text{and} \quad \frac{1}{2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} a_{sj}^2 = \frac{n_s}{2} \sum_{j=1}^{n_s} a_{sj}^2,$$

the global result is exactly the one computed for the left hand side. □

This result immediately means that the contribution of cluster S to the total SSE is,

$$\sum_{j=1}^{n_s}(a_{sj}-\overline{a_s})^2=\frac{1}{2n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}(a_{si}-a_{sj})^2 \tag{1}$$

Now let us denote $N=\{1,\ldots,n\}$ the set of indexes of the Microdata individuals. Let us define the following binary variables x_{ij} for all $i,\ j\in N$ such that:

$$x_{ij}=1\Leftrightarrow a_i,\ a_j \text{ belong to the same cluster after microaggregation}$$
$$x_{ij}=0 \qquad \text{otherwise}$$

With this definition it is assumed that $x_{ii}=1,\ \forall\ i\in N$ and that x_{ij} is symmetric, i.e. $x_{ij}=x_{ji}$.
To simplify the writing we will denote as node i the corresponding individual a_i.
Now let r be a node corresponding to individual a_r, $r\in N$. Let n_r be the number of elements of the cluster where r is. Let SQ_r be the contribution to the total SSE of the cluster where r is. Following the idea of equation (1) we can obtain SQ_r by summing all square distances between the nodes in the cluster where r is (i.e., the nodes i, j with $x_{ir}=1$ and $x_{jr}=1$) and then divide it by twice n_r the number of nodes in the cluster.

$$SQ_r=\frac{1}{2n_r}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_j-a_i)^2x_{ir}x_{jr} \tag{2}$$

Now we can face the SSE computation. SSE is the total sum of the contributions of the different final clusters, one contribution for each cluster. If we consider every $r\in N$ and we add its corresponding SQ_r eventually we will be adding n_r times this SQ_r, one for each element in the cluster where r is. Therefore, if we divide by n_r this SQ_r contribution for every r, we will be obtaining the total SSE:

$$SSE=\sum_{r=1}^{n}\frac{SQ_r}{n_r}=\sum_{r=1}^{n}\frac{\frac{1}{2n_r}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_j-a_i)^2x_{ir}x_{jr}}{n_r}=\sum_{r=1}^{n}\frac{1}{2n_r^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_j-a_i)^2x_{ir}x_{jr}$$

The previous expression only requires applying equation (2) for SQ_r. Now, we reorder the sums as they are finite

$$SSE=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_j-a_i)^2\sum_{r=1}^{n}\frac{x_{ir}x_{jr}}{n_r^2}$$

Focusing on the term on the right $x_{ir}x_{jr}$, we notice it can only be different from 0 when i and j are in the same cluster as r. Moreover, in this case i and j must be in the same cluster so x_{ij} must be 1. Besides, in this case it is also true that $n_r=n_i=n_j$ because we are talking of the same cluster. If we sum it for all r we are repeating n_r (or equivalently n_i, n_j) times the value $x_{ir}x_{jr}=1$. Therefore it is easy to see that

$$\sum_{r=1}^{n}\frac{x_{ir}x_{jr}}{n_r^2}=\frac{x_{ij}}{n_i}$$

Substituting this in the expression of SSE above, we eventually obtain

$$SSE=\frac{1}{2}\sum_{i=1}^{n}\frac{\sum_{j=1}^{n}(a_i-a_j)^2x_{ij}}{n_i}$$

In order to simplify the problem we notice that for the case x_{ii} the contribution to the SSE is 0 because $(a_i-a_i)^2=0$. Therefore, we can omit the existence of variables x_{ii} and define initially the variables such that $i\neq j$ (this deletes n variables for the problem). Now we consider n_i the number of elements of the cluster where i is in terms of the variables x_{ij}:

$$n_r=\sum_{\substack{j=1\\ j\neq r}}^{n}x_{jr}+1$$

And finally substituting the term in the expression of SSE omitting variables x_{ii} we obtain an expression of the total SSE in terms only of the variables x_{ij}, $i \neq j$:

$$SSE = \frac{1}{2}\sum_{i=1}^{n} \frac{\sum_{j=1,\ j\neq i}^{n} (a_i - a_j)^2 x_{ij}}{\sum_{j=1, j\neq i}^{n} x_{ij} + 1} \tag{3}$$

Our initial objective was to construct the clusters such that SSE was minimized. Consequently we should obtain the values of the variables x_{ij} such that they define a feasible clustering and they minimize the expression for SSE in equation (3). Forcing them to define a feasible clustering will give us the constraints. Equation (3) will bring us our objective function to minimize.

To force the variables to describe clusters, they must be coherent with their definition. First, they must be symmetric ($x_{ij} = x_{ji}$). Second, they must be **complete** in the sense that every node in the cluster is related to every other node in the same cluster. For example if i and r are in the same cluster (i.e., $x_{ir} = 1$) and j and r are on the same cluster too (i.e., $x_{jr} = 1$) then it must be that i and j are in the same cluster ($x_{ij} = 1$). It can be described by the following set of inequalities, named **triangle inequalities** from now on:

$$x_{ir} + x_{jr} - x_{ij} \leq 1 \quad \text{for all } i, j, r \in N, \quad i \neq j,\ r \neq j,\ i \neq r$$

From now on we will denote as **cliques** the description of clusters in terms of variables x_{ij}. Sometimes we may refer to them as complete clusters, what at some sense may sound like an abuse of notation. Mainly they are simply combinations of variables x_{ij} satisfying the triangle inequalities. We will refer to the size of a clique as the size of the cluster it corresponds to.

To force the variables to originate feasible clusters, we must force the clusters (or cliques) to have size at least k. This means $n_i \geq k\ \forall\ i \in N$. We will call it the **size inequality** and it can be easily expressed as:

$$\sum_{j=1,\ j\neq i}^{n} x_{ij} \geq k - 1$$

Putting all pieces together, our clustering problem can be described as an optimization problem with the binary symmetric variables x_{ij}, $i, j \in N$, $i \neq j$:

$$\begin{aligned}
&\underset{x_{ij},\ i,\ j\in N,\ i\neq j}{\text{minimize}} \ \frac{1}{2}\sum_{i=1}^{n} \frac{\sum_{j=1,\ j\neq i}^{n} (a_i - a_j)^2 x_{ij}}{\sum_{j=1,\ j\neq i}^{n} x_{ij} + 1} \\
&\text{subject to:} \\
&x_{ij} = x_{ji}, \quad \text{for all } i, j \in N, \quad i \neq j \\
&\sum_{j=1, j\neq i}^{n} x_{ij} \geq k - 1, \quad \text{for all } i \in N \\
&x_{ir} + x_{rj} - x_{ij} \leq 1, \quad \text{for all } i, j, r \in N, \quad i \neq j,\ i \neq r,\ r \neq j \\
&x_{ij} \in \{0, 1\}, \quad \text{for all } i, j \in N, \quad i \neq j
\end{aligned} \tag{4}$$

By looking at the objective function, we first notice that the factor $1/2$ can be omitted if we suppress the symmetries. This means that for every pair $i \neq j$ we only consider one variable x_{ij} instead of the two symmetric variables x_{ij} and x_{ji}. Without loss of generality, from now on we consider only variables x_{ij} with $i < j$. This means there only survives the half of the total SSE sum and therefore we can neglect the $1/2$ factor and the symmetry constraint. Therefore, at this point we redefine the variables x_{ij} neglecting

symmetric pairs and pairs with the same node.

$$\begin{aligned} & x_{ij} = 1 \Leftrightarrow a_i, a_j \text{ belong to the same cluster} \\ & x_{ij} = 0 \qquad \text{otherwise} \\ & i,\, j \in N, \quad i < j \end{aligned}$$

This eventually redefines the minimization problem as follows after neglecting unnecessary variables:

$$\begin{aligned} & \underset{x_{ij},\, i,\, j \in N,\, i<j}{\text{minimize}} \sum_{i=1}^{n} \frac{\sum_{j=i+1}^{n} (a_i - a_j)^2 x_{ij}}{\sum_{j \in N,\, j<i} x_{ji} + \sum_{j \in N,\, j>i} x_{ij} + 1} \\ & \text{subject to:} \\ & \sum_{j \in N,\, j<i} x_{ji} + \sum_{j \in N,\, j>i} x_{ij} \geq k - 1, \quad \text{for all } i \in N \\ & x_{ir} + x_{rj} - x_{ij} \leq 1, \quad \text{for all } i,\, j,\, r \in N, \quad i < r,\, r < j \\ & x_{ij} \in \{0, 1\}, \quad \text{for all } i,\, j \in N, \quad i < j \end{aligned} \tag{5}$$

In any case, the most remarkable issues are that:

(1) The function is non-linear because of the dividing term in the fraction (size of a cluster).
(2) The function is non-convex. For example for the case $n = 2$ there is only one variable x_{12}. We note it by simply x and we note also $c = (a_1 - a_2)^2 > 0$. The objective function is:

$$f(x) = \frac{cx}{1+x} \Rightarrow f''(x) = \frac{-2c}{(1+x)^3}$$

The second derivative is clearly non positive for $x > -1$ and therefore also for $x \geq 0$ (note that x_{12} is non-negative).

This means that, using model (5), the problem has serious difficulties to be solved by schemes of Binary Integer Problems. The idea currently used in the state of the art is simplifying it with an approximation, obtaining a new model (6).

2. Approximate Model

We approximate our objective function by assuming the cliques have similar size, exactly as in the heuristics in *Chapter 2*, where the size of the clusters was fixed for most of them. With this approximation we can minimize neglecting the dividing term for the objective function:

$$\begin{aligned} & \underset{x_{ij},\, i,\, j \in N,\, i<j}{\text{minimize}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (a_i - a_j)^2 x_{ij} \\ & \text{subject to:} \\ & \sum_{j \in N,\, j<i} x_{ji} + \sum_{j \in N,\, j>i} x_{ij} \geq k - 1, \quad \text{for all } i \in N \\ & x_{ir} + x_{rj} - x_{ij} \leq 1, \quad \text{for all } i,\, j,\, r \in N, \quad i < r,\, r < j \\ & x_{ij} \in \{0, 1\}, \quad \text{for all } i,\, j \in N, \quad i < j \end{aligned} \tag{6}$$

With model (6) the problem is linear and, particularly, it is a Binary Integer Problem (BIP).

Chapter 5
State of the Art in the Approximate Model of Microaggregation

In this chapter we are going to explore the nature of the polyhedron formulation in our approximate model of microaggregation problem. This study is mainly reporting the literature in [**8**], [**9**] and [**10**]. First, we recall our approximate model (6) for microaggregation:

$$
\begin{aligned}
&\underset{x_{ij},\ i,\ j\in N,\ i<j}{\text{minimize}} \sum_{i=1}^{n}\sum_{j=i+1}^{n}(a_i-a_j)^2x_{ij}\\
&\text{subject to:}\\
&\sum_{j\in N,\ j<i} x_{ji} + \sum_{j\in N,\ j>i} x_{ij} \geq k-1, \quad \text{for all } i\in N\\
&x_{ir}+x_{rj}-x_{ij}\leq 1, \quad \text{for all } i,\ j,\ r\in N, \quad i<r,\ r<j\\
&x_{ij}\in\{0,1\}, \quad \text{for all } i,\ j\in N, \quad i<j
\end{aligned}
\tag{6}
$$

At this stage, it can be useful to switch to a graph notation to simplify the writing.
Let $K_n=(V_n,E_n)$ be a complete undirected weighted graph with n nodes. Then, we can clearly correspond every individual $i\in N$ to a single node in V_n. Furthermore we can correspond every variable x_{ij}, $i<j$, to a single undirected edge $e\in E_n$ such that $e=(i,j)\in E_n$. We can also define the parameters c_e, $e\in E_n$ such that $c_e=(a_i-a_j)^2$ the weight for every edge $e=(i,j)$ in E_n. With this new notation the model of our approximate problem becomes (7):

$$
\begin{aligned}
&\underset{x_e,\ e\in E_n}{\text{minimize}} \sum_{e\in E_n} c_e x_e\\
&\text{subject to:}\\
&\sum_{e\in\delta(v)} x_e \geq k-1, \quad \text{for all } v\in V_n\\
&x_a+x_b-x_c\leq 1, \quad \text{for all } a,b,c \text{ triangle in } E_n\\
&x_e\in\{0,1\}, \quad \text{for all } e\in E_n
\end{aligned}
\tag{7}
$$

Our formulation of the approximate problem is the intersection of the triangle inequalities and size inequalities. Every one of this two sets of inequalities by separate define two polyhedrons and its intersection is the final formulation. In Integer Programming when the formulation includes the intersection of two polyhedrons sometimes it is useful to study them separately. In our case we will start by considering the polyhedrons provided by triangle inequalities and then we will consider what happens when adding size constraints. All results and statements below in this chapter are extracted from papers [**8**], [**9**] and [**10**].

1. Clique Partitioning Problem

Let us consider the approximate microaggregation problem with only triangle inequalities. Recall that the triangle inequalities forced the eventual clusters in the solutions to be cliques, i.e., to be complete in edges. That means that our approximate problem considering only triangle inequalities is solved by obtaining a clique partitioning of K_n such that the total weight of the eventual cliques is minimized. This problem statement is commonly known as the Clique Partitioning Problem (CPP) and can be modelled as follows:

$$\begin{aligned} &\underset{x_e,\, e\in E_n}{\text{minimize}} \sum_{e\in E_n} c_e x_e \\ &\text{subject to:} \\ &x_a + x_b - x_c \leq 1, \quad \text{for all } a,b,c \text{ triangle in } E_n \\ &x_e \in \{0,1\}, \quad \text{for all } e \in E_n \end{aligned} \tag{8}$$

In our particular approximate case it is obvious that given the fact that the coefficients $c_e > 0$ for our problem, the solution to CPP would be taking $x_e = 0$ for all $e \in E_n$ and therefore take every node in V_n as a single clique. Nevertheless, studying the polyhedron of this problem also provides inequalities which are valid for our approximate problem.
In the general case, CPP considers negative edge weights too. On account of that, a clique partitioning with cliques different than the single nodes can have a total negative weight and therefore lower than the null solution.

CPP in the State of the Art

CPP is efficiently solved for low data cases by Grötschel & Wakabayashi in [**9**], [**8**] by defining strong valid inequalities and using a cutting plane method that searches for violated inequalities of this type at every step. In [**8**], the facial structure of CPP is studied and families of facet-defining valid inequalities are provided. The first observation is that the Convex Hull of the clique partitionings is full dimensional, i.e. has dimension $|E_n|$.

Let X be the set of all clique partitionings of K_n. Let $\mathcal{P}_n$ be its Convex Hull. With this description we can define $\mathcal{P}_n$ as:

$$\mathcal{P}_n = \text{conv}(\{x \in \{0,1\}^{|E_n|} \mid x_a + x_b - x_c \leq 1 \ \ \forall a,b,c \text{ triangle in } K_n\})$$

Clearly the points $x_e = 0\ \forall e \in E_n$; $x_a = 1$, $a \in E_n$ and $x_e = 0\ \forall e \in E_n \setminus \{a\}$; are $|E_n| + 1$ different points and they are affinely independent too. Note that taking the differences to the null point $x_e = 0\ \forall e \in E_n$, the result is the set of vectors in the canonical base of $\mathbb{R}^{|E_n|}$. On account of that, $\mathcal{P}_n$ is full dimensional.
With a similar argument we can state that inequalities of the type $x_b \geq 0$ for a given $b \in E_n$, are facet-defining. If we take the face $\{x \in \mathcal{P}_n \mid x_b = 0,\ b \in E_n\}$ corresponding to one of this inequalities we observe that in this face there are $|E_n|$ affinely independent points (the null one and those with $x_a = 1$ for a given $a \in E_n \setminus \{b\}$ and $x_e = 0\ \forall e \in E_n \setminus \{a\}$) so the face has dimension $|E_n| - 1$ and therefore is a facet.
Another preliminary result shows that the inequalities $x_b \leq 1$ for a given $b \in E_n$, are not facet-defining. If we sum the valid triangle inequalities $x_a + x_b - x_c \leq 1$ and $x_b + x_c - x_a \leq 1$ for a, $c \in E_n$ two edges forming a triangle with b, we obtain $x_b \leq 1$ and therefore this inequality is redundant.
With this preliminaries stated we can move on to the main results extracted from [**8**]. We will only proof the triangle inequalities are facet-defining as well as the validity for some of the inequalities proposed, the rest of the proofs are all in [**8**]. All the following results are stated for X and $\mathcal{P}_n$ in CPP, with $n \geq 3$:

(1) $x_e \geq 0$ defines a facet for $\mathcal{P}_n$.
(2) $x_a + x_b - x_c \leq 1$ is facet-defining for $\mathcal{P}_n$ for every triangle a, b, c in E_n.

PROOF. Let $a = (u,v)$, $b = (v,w)$, $c = (u,w)$ the triangle for 3 nodes u, v, w in V_n. Let us denote, in this case, a given point in X as only the set of edges which are activated, i.e., equal to 1. The following $|E_n|$ solutions are affinely independent and all satisfy $x_a + x_b - x_c = 1$:
$\{a\}, \{b\}, \{a,b,c\}$
$\{a,e\}\ \ \forall e \in E_n \setminus \{\delta(u) \cup \delta(v)\}$

$\{b, e\}$ $\forall e \in \delta(u) \setminus \{a, c\}$
$E_n(\{u, v, w, z\})$ $\forall z \in V_n \setminus \{u, v, w\}$ □

(3) For every two nonempty disjoint sets S, T of V_n

$$\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(S)} x_e - \sum_{e\in E_n(T)} x_e \leq min\{|S|, |T|\}$$

is valid for X. It is facet-defining for $\mathcal{P}_n$ iif $|S| \neq |T|$. It is called 2-partition inequality.

Proof. We only proof validity. See [**8**] for the facet-defining proof.
We assume without loss of generality that $|S| \leq |T|$. Let $s = |S|$, $t = |T|$. We apply induction over $s + t$.
First we suppose $s = 1$. If $t = 1$ is trivial. If $t = 2$ it is a triangle inequality. Then for $t \geq 3$ the induction hypothesis imposes that given $v \in T$,

$$\sum_{e\in[S:T\setminus\{v\}]} x_e - \sum_{e\in E_n(T\setminus\{v\})} x_e \leq 1$$

is valid for $\mathcal{P}_n$. We add them for all $v \in T$.

$$(t-1)\sum_{e\in[S:T]} x_e - (t-2)\sum_{e\in E_n(T)} x_e \leq t$$

We add the valid inequality $-\sum_{e\in E_{n(T)}} x_e \leq 0$ and therefore,

$$\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(T)} x_e \leq \frac{t}{t-1}$$

and taking integer parts given that the solutions are integer,

$$\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(T)} x_e \leq \lfloor \frac{t}{t-1} \rfloor = 1 = s = min\{s, t\}$$

Now let $s \geq 2$, $t \geq 2$. The case $s = 2$, $t = 2$ is the sum of two triangle inequalities and therefore is valid. For induction we know that for every $v \in S$,

$$\sum_{e\in[(S\setminus\{v\}):T]} x_e - \sum_{e\in E_n(S\setminus\{v\})} x_e - \sum_{e\in E_n(T)} x_e \leq s-1$$

Similarly, for every $v \in T$

$$\sum_{e\in[S:(T\setminus\{v\})]} x_e - \sum_{e\in E_n(S)} x_e - \sum_{e\in E_n(T\setminus\{v\})} x_e \leq min\{s, t-1\}$$

We add them for every $v \in T$, $v \in S$ and then,

$$(s+t-2)[\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(S)} x_e - \sum_{e\in E_n(T)} x_e] \leq s(s-1) + t \cdot min\{s, t-1\} \quad (*)$$

If $s < t$ then,

$$\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(S)} x_e - \sum_{e\in E_n(T)} x_e \leq \lfloor \frac{s(s+t-1)}{s+t-2} \rfloor = s$$

If $s = t$ we rewrite (*) as

$$(2s-2)[\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(S)} x_e - \sum_{e\in E_n(T)} x_e] \leq 2s(s-1) = s(2s-2)$$

and hence,

$$\sum_{e\in[S:T]} x_e - \sum_{e\in E_n(S)} x_e - \sum_{e\in E_n(T)} x_e \leq s = min\{s, t\}$$

On account of everything, it is clear that in any case the 2-partition inequality is valid for $\mathcal{P}_n$. It also shows that if $s = t$ the inequality is obtained by only adding other valid inequalities and therefore it defines no facet. There only lacks to prove it is facet defining if $s \neq t$, see [**8**]. □

(4) Let C be a cycle of length $q \geq 5$. $C = \{(v_i, v_{i+1}) \mid i = 1, \ldots, q-1\} \cup \{(v_1, v_q)\}$.
We define $\bar{C} = \{(v_i, v_{i+2}) \mid i = 1, \ldots, q-2\} \cup \{(v_1, v_{q-1}), (v_2, v_q)\}$ its 2-chords. Figure 1. includes a graphic representation.

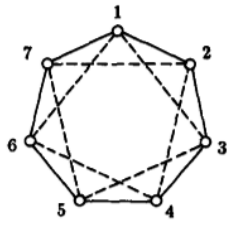

FIG. 1. Cycle C between nodes 1 and 7 in continuous line and $\bar{C}$ in dashed line.

With this definition,

$$\sum_{e \in C} x_e - \sum_{e \in \bar{C}} x_e \leq \lfloor \frac{1}{2} |C| \rfloor$$

is valid for X. It is facet-defining for $\mathcal{P}_n$ if and only if $|C|$ is odd. We call it 2-chorded cycle inequality induced by cycle C.

PROOF. We only prove validity. See [**8**] for further proofs.
For each edge $c \in \bar{C}$ we take c_1 and c_2 in C that together with c form a triangle $\{c, c_1, c_2\}$ in E_n, see Figure 1. Then the triangle inequality imposes $x_{c_1} + x_{c_2} - x_c \leq 1$. We add all those triangle inequalities for each $c \in \bar{C}$ and eventually we obtain,

$$2 \sum_{e \in C} x_e - \sum_{e \in \bar{C}} x_e \leq |\bar{C}| = |C|$$

We add then the valid inequality $-\sum_{e \in \bar{C}} x_e \leq 0$ and take integer parts. It results that,

$$\sum_{e \in C} x_e - \sum_{e \in \bar{C}} x_e \leq \lfloor \frac{1}{2} |C| \rfloor$$

is valid for X. If $|C|$ is even it can be obtained adding valid triangle inequalities and therefore the 2-chorded cycle inequality is not facet-defining for even cycles. Lacks only to prove it is facet-defining for $|C|$ odd. See [**8**]. □

(5) Let P be a path $P = \{(v_i, v_{i+1}) \mid i = 1, \ldots, k-2\}$. We define $\bar{P} = \{(v_i, v_{i+2}) \mid i = 1, \ldots, k-3\}$ its 2-chord. Given $z \in V_n$ not in P let $R = \{(v_i, z) \mid i = 1, \ldots, k-1;\ i \text{ even}\}$ and $\bar{R} = \{(v_i, z) \mid i = 1, \ldots, k-1;\ i \text{ odd}\}$. Then,

$$\sum_{e \in P \cup R} x_e - \sum_{e \in \bar{P} \cup \bar{R}} x_e \leq \lfloor \frac{1}{2} (|P| + 1) \rfloor$$

is valid for X. It is facet-defining for $\mathcal{P}_n$ if and only if $|P|$ is even. We call it 2-chorded path inequality induced by P.
Figure 2. represents the sets of edges $P \cup R$ and $\bar{P} \cup \bar{R}$ with an example.

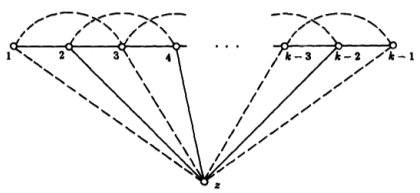

FIG. 2. $P \cup R$ in continuous line and $\bar{P} \cup \bar{R}$ in dashed line.

(6) Let C be an even cycle (even number of nodes and edges), $\bar{C}$ its 2-chord and $\{V, \bar{V}\}$ a bipartition of the nodes in C. Let z be an outside node of C. Let $R = \{(v, z) \mid v \in V\}$ and $\bar{R} = \{(v, z) \mid v \in \bar{V}\}$. Then,

$$\sum_{e \in C \cup R} x_e - \sum_{e \in \bar{C} \cup \bar{R}} x_e \leq \frac{1}{2}|C|$$

is valid and facet-defining for $\mathcal{P}_n$. We call it 2-chorded even wheel inequality.
Figure 3. represents the sets of edges $C \cup R$ and $\bar{C} \cup \bar{R}$ with an example of 8 nodes.

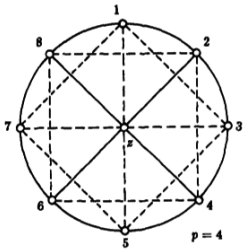

FIG. 3. $C \cup R$ in continuous line and $\bar{C} \cup \bar{R}$ in dashed line. p nodes of the bipartition sets.

On the other hand, Grötschel & Wakabayashi in [**9**] obtained good experimental results for CPP only considering inequalities 1, 2 and 3 from the list in the following cutting plane method:

Cutting plane method for CPP:

(i) Let $\mathcal{F}$ be a set of inequalities. Set $\mathcal{F} = \{0 \leq x_e \leq 1, e \in E_n\}$. Set MAXCUT an integer parameter such that MAXCUT $\in \{400, 500\}$ (empirically performs well with these values).
(ii) Solve the linear relaxation of CPP with the formulation from $\mathcal{F}$ and obtain an optimal solution x^*.
(iii) If x^* satisfies all triangle inequalities and is integer, stop.
Else, check all $3\binom{n}{3}$ triangle inequalities and order the violated ones by gap of violation. The gap of violation is the difference $x^*_a + x^*_b - x^*_c - 1$, i.e., the degree of violation in x^* for a given violated triangle inequality. Choose the MAXCUT more violated ones and add them to $\mathcal{F}$. If there are less than MAXCUT violated inequalities add them all to $\mathcal{F}$. Apply row elimination (see the paragraph below the algorithm) to $\mathcal{F}$ and return to (ii).
If x^* satisfies all triangle inequalities but is not integer run this heuristic to find 2-partition inequalities:
(a) Consider $|S| = 1$. For every node $v \in V_n$ we set $W = \{w \in V_n \setminus \{v\} \mid 0 < x^*_e < 1; \quad e = (v, w)\}$. Then we take and order in W (increasing order on the nodes in W for example) and pick the first $w \in W$ and set $T = \{w\}$. For every $i \in W \setminus \{w\}$ we set $T = T \cup \{i\}$ if $x^*_{ij} = 0$ for all $j \in T$.
We check whether the resulting set T satisfies $\sum_{e \in [\{v\}:T]} x^*_e > 1$ and in this case we add

$$\sum_{e \in [\{v\}:T]} x_e \leq 1$$

to $\mathcal{F}$. Then, we repeat this process conversing the order in W. In case we find violated inequalities, apply row elimination to $\mathcal{F}$ and return to (ii).
In case we have not found new inequalities, we repeat the process considering $T = T \cup \{i\}$ if $x^*_{iv} - \sum_{j \in T} x^*_{ij} > 0$.
(b) If we could not find valid inequalities with $|S| = 1$, there is another routine not described in [**9**] but it is not necessary in the practical cases reported.

Note that row elimination consists on: everytime we have an optimal x^* we erase all inequalities $\alpha x \leq \beta$ in $\mathcal{F}$ which are strictly satisfied, i.e., we eliminate all $\alpha x \leq \beta$ that satisfy $\alpha x^* < \beta$.

In conclusion, we can state that if we split our approximate problem formulation and consider the clique condition, the problem remaining (CPP) is deeply developed in the state of the art. Plenty of families of

inequalities are known. Besides, many of them are proved to be facet-defining. The polyhedron is full-dimensional too. On top of that, well performing (for low data cases) cutting plane implementations exist using part of these inequalities. Now we are interested on how is the scenario modified when adding size constraints. In particular we require the cliques to have a minimum size. With this requirement we clearly are targeting our microaggregation approximate problem model directly.

2. Clique Partitioning Problem with Minimum Cluster Size

Let us recall the approximate model of microaggregation. With the graph notation in K_n we can rewrite it:

$$
\begin{aligned}
&\underset{x_e,\, e\in E_n}{\text{minimize}} \sum_{e\in E_n} c_e x_e \\
&\text{subject to:} \\
&\sum_{e\in\delta(v)} x_e \geq k-1, \quad \text{for all } v \in V_n \\
&x_a + x_b - x_c \leq 1, \quad \text{for all } a,b,c \text{ triangle in } K_n \\
&x_e \in \{0,1\} \quad \text{for all } e \in E_n
\end{aligned} \tag{9}
$$

The objective is then to obtain a clique partitioning of K_n such that the total weight of the cliques is minimized and requiring the cliques to have size at least $k \geq 1$. This problem is known as Clique Partitioning Problem with Minimum Size Requirement or simply CPPMIN. Therefore CPPMIN is exactly the optimization problem of our approximate model for microaggregation. In [**10**] valid inequalities and other results are found for this problem as well as a Branch & Cut method that works well for low data cases.

Let $\mathcal{P}_{n,k}$ be the Convex Hull of the clique partitionings with minimum size k for the cliques, i.e.,

$$
\begin{aligned}
\mathcal{P}_{n,k} = \text{conv}(\{x \in \{0,1\}^{|E_n|} \mid & \sum_{e\in\delta(v)} x_e \geq k-1, \quad \text{for all } v \in V_n, \\
& x_a + x_b - x_c \leq 1 \quad \text{for all } a,\, b,\, c \text{ triangle in } E_n\})
\end{aligned}
$$

Note that all inequalities in CPP are still valid on CPPMIN as it only consists on adding the constraint of the minimum size for the cliques. In [**10**] the dimensionality of $\mathcal{P}_{n,k}$ is explored based on results for the equipartition problem. The equipartition problem consists on dividing the nodes into two clusters of equal size or with a difference of 1 node at most when the number of nodes is odd. The polyhedron for the equipartition problem has dimension $|E_n| - n$, see [**10**]. The results for dimensionality in [**10**] imply that:

(a) If $n > 2k$, $\mathcal{P}_{n,k}$ is full dimensional, see [**10**].
(b) If $n = 2k$, the only feasible solutions are the equipartitions of the graph and the one with only one total cluster including all nodes. Clearly all other solutions would imply clusters with less than k elements. Therefore, $\dim(\mathcal{P}_{n,k}) = |E_n| - n$ for the equipartition problem.
(c) If $n < 2k$ then $k > n/2$ and the only feasible solution is to include all nodes in V_n in one cluster. For this reason clearly the dimension of $\mathcal{P}_{n,k} = 0$ in this case.

Now, assume we are in the case $n > 2k$. Otherwise we would either consider the total cluster solution or the equipartition problem. In [**10**] new valid inequalities for CPPMIN are found, some of them facet-defining for $\mathcal{P}_{n,k}$:

(1) U, $W \subseteq V_n$ with $|U| > |W|$ and $n > k|U| + 2k + 1$.
Then, the 2-partition constraint,

$$
\sum_{e\in[U:W]} x_e - \sum_{e\in E_n(U)} x_e - \sum_{e\in E_n(W)} x_e \leq min\{|U|, |W|\}
$$

is facet-defining for $\mathcal{P}_{n,k}$. Recall we already know from CPP that this inequality is valid.

(2) Let $m = \lfloor n/k \rfloor$, $r \equiv n \bmod k$, i.e., $n = mk+r$, $m > 1$ given that $k < n/2$. Let $P \subseteq V_n$ of size $p = tm+q$ with $t \geq 1$ and $1 \leq q < m$.
Then, the pigeon constraint,

$$\sum_{e \in E_n(P)} x_e \geq \frac{1}{2}t(t-1)(m-q) + \frac{1}{2}t(t+1)q$$

   (a) If $r = 0$ is valid but not facet-defining.
   (b) If $r = 1$ is facet-defining only if $P = V_n$.
   (c) If $r > 1$ is facet-defining always.

(3) Let $W \subseteq V_n$ of size $w = k + q$, $q < k$. Then the flower inequality

$$\sum_{e \in E_n(W)} x_e + \sum_{e \in \delta(W)} x_e \geq \binom{k}{2} + \binom{q}{2} + q(k-q)$$

is valid for CPPMIN.

(4) Under the same circumstances as in 3.

$$(k-q) \sum_{e \in E_n(W)} x_e + k \sum_{e \in \delta(W)} x_e \geq (k-q)\binom{k+q}{2}$$

is valid for CPPMIN.

In [**10**] there is also a description of a Branch & Cut scheme that looks first for violated triangle inequalities, 2-partition inequalities, pigeon inequalities of size $n-1$, pigeon inequalities of size $m+1$ and flower constraints in this order of preference. It uses MD algorithm (see *Chapter 2., Section 2.*) to obtain a first feasible solution as incumbent. It also uses a heuristic to obtain a small weight clique of size $m+1$ when looking for a clique P that raises a violated pigeon constraint of size $m+1$ (separation routine). Other pigeon constraints are explored enumerating subsets. The computational results in [**10**] showed that flower and specially pigeon constraints had to be added in order to achieve tighter solutions. The tightness of a solution is defined here as the difference in relative value between the optimal linear relaxation and the integer solution obtained. In general, this is known as the GAP of an integer solution. It directly relates to how *fine* is the formulation with the cutting planes added.

At this stage, we can claim that the state of the art for our approximate microaggregation problem is highly developed. The subsequent problems CPP and CPPMIN, which is directly our problem approximated, provides us with several valid inequalities. The facial structure of the subsequent polyhedrons has been deeply explored. Furthermore, cutting plane methods and other strategies are also available in the literature.
However, the problem is still very tough as we can conclude from the computational results from those methods. In particular, Mitchell and Ji in [**10**] test their Branch & Cut scheme with data sets with up to 103 individuals with only two attributes. This information gives us a reference on the size of the data sets we might expect to work with. In the case of [**10**] they are working on approximated microaggregation, so in case non-approximate microaggregation is approached, one should be aware of those computational limitations. Just to give an idea, in [**10**] it is also noted that CPLEX can spend a day to solve CPPMIN with instances of only 25 nodes in case there are no added new cutting planes. We absolutely cannot expect to work on big instances of data, by now.

As part of the state of the art in the approximate model of microaggregation, there still lacks the report of an important paper by Mitchell and Ji in CPPMIN with a Branch & Price & Cut scheme [**11**]. This paper actually inspires our contributions and therefore we leave it for the next part.

Our objective is now to face the non-linearity (and non-convexity) of our objective function in the original microaggregation formulation. The idea is to open a research line on target.
One possible idea is to try to reformulate the problem using another distance. Distance L_1 has been considered for the weights of the edges. However, the problem modelling difficulty increases given that, in this case, we cannot skip the difficulty of the centroid of the clusters in our objective function. Every expression of our objective function includes the term of the centroid which is non linear as the number of elements in a cluster (which eventually works as a variable) is a dividing term. All final objective functions

using this idea failed at pursuing convexity and linearity, even increasing the number of variables to include cluster variables and more. For this reason, this alternative was rapidly discarded.

One alternative approach is given by Sande in [**15**], where the weight of a given clique is computed as the weight of the minimal spanning tree inside this clique, considering still the distances L_2 between nodes as edge weights. The idea for this new approach is that in univariate data the range of a cluster (distance between the lowest and greatest element) is a good measure for the spread in a cluster. On account of that, it uses the minimal spanning tree weight as an extension of the range of a cluster for higher dimensions, where the sense of order between elements is lost. It also defends this particular clustering spread measure on account of the fact that microaggregation data tends to be highly skewed, as usually consists on economic data. In [**15**] Sande claims that the range of the data in one dimension is a better measure of the spread than the variance, which corresponds to the total sum of square distances to the centroid we have been considering. Besides, [**15**] proposes a heuristic greedy algorithm for multivariate data that consists on starting from the minimal spanning tree of the whole data and erasing edges of high distance with the constraint that the remaining clusters still have size at least k, i.e., are feasible yet.

At this point the document finishes the part dedicated to report the general knowledge and the state of the art in microaggregation. Follows now the most remarkable block of this project, the contributions of it at a current stage. Recall this project has continuity and that this document only reports its contributions until the 7th of January 2016.

As a brief introduction, we simply state that a new approach to microaggregation is considered for the original model without approximating for similar clique sizes. It is based on another technique for Operations Research in general known as Column Generation. Its application to microaggregation problem is inspired by the same authors John E. Mitchell and Xiaoyun Ji in [**10**], who developed a Branch&Price method to solve CPPMIN using Column Generation, see [**11**].

Part 3

Integer Programming on Microaggregation (II), Contributions of the Project

Chapter 6
Column Generation Theory

In major terms, by now, two main techniques for Integer Programming problems have been mentioned. On the one hand, Branching intends to iteratively divide the problem in subproblems and solve the subsequent linear relaxations for them until it reaches feasible solutions. On the other hand, cutting plane methods intend to tighten the formulation adding strong valid inequalities. Both techniques can be combined in Branch & Cut schemes where cutting plane methods provide with finer formulations which can be afterwards treatable by Branching. See chapters 7, 8 and 9 for deeper information in [**16**]. However there is an alternative and sophisticate technique to deal with linear problems in general and also practical for certain Integer Programming problems. This technique is explored in chapter 11 in [**16**], and applied in [**11**] for CPPMIN. It is called Column Generation and it is especially useful for problems with a large number of variables.

The idea in this case is to deal with linear problems that include plenty of variables, taking into account the fact that many of them might be zero eventually. Let us consider by now the general case of a standard linear problem. Recall that, if there exists an optimal solution, then there must be a basic feasible solution which is optimal. In this case, at least all non basic variables must be zero for the optimal solution. Column Generation intends to exploit this condition in order not to consider all variables thus reducing the size of the problem.

Firstly, we state that the following explanation is in the sense of a minimization problem like ours. Note that maximization problems are just a little variation with no substantial change in the idea however.
The objective in Column Generation is only taking into account variables whose consideration might improve the value for the objective function. In minimization terms these are simply variables with negative reduced cost. To do so, the problem is divided into two separate problems: the Master Problem and the Pricing Problem.

The Master Problem is itself the minimization problem but considering a subset of the variables, whereas the Pricing Problem is a separate problem whose solution provides with a new variable to be considered in the Master Problem. This new variable must be worth adding, i.e., must improve the value of the objective function when considered. This is equivalent to say that this new variable must have a negative reduced cost for the Master Problem. In every stage the Master Problem is solved and used to formulate the Pricing Problem. Then Pricing Problem is solved to provide a new variable for the Master Problem. This iteration scheme is followed until the Pricing Problem establishes there are no new variables with negative reduced cost.

Note that the two problems are absolutely separated. In the Master Problem the variables are just a subset of the original variables. In the Pricing Problem the variables modelling must be developed such that the final result indicates a new variable to consider for the Master Problem. The mindset is therefore totally different in the two problems and the variables in both problems have nothing to do with each other.
The name of Column Generation for this technique comes from the fact that adding a new variable to the Master Problem implies adding a new column to the matrix of constraints in the formulation of the Master Problem. The issue of the new column coefficients to add is essential to see how does the Pricing Problem works.

The idea in the Pricing Problem is to consider the reduced cost of the new variable as the objective function to minimize. It is clear that in order to obtain new variables for the Master Problem it is not necessary to solve the Pricing Problem to optimality. If a given feasible solution is found with a negative objective function value, it can be directly added. Nevertheless, if no new variables with negative reduced cost are found in a first stage, it is necessary to solve the Pricing Problem to optimality to confirm there exist no new variables with negative reduced cost to be added. In other words, a non negative optimal solution for the Pricing Problem means that no new variables have to be added to the Master Problem.

There is no standard procedure to model the Pricing Problem. Nevertheless, in any case the Pricing Problem must be constructed such that its objective function is the reduced cost of a variable to be added. This involves using dual variables from the Master Problem.
That point arises a remarkable issue with respect to the Master Problem. Obtaining its dual variables forces the Master Problem to be an LP (continuous variables). This means that in case that Column Generation is considered for Integer Programming, at some point the integrality of the solution must be an issue to deal with. Branching could be a strategy, or also adding cutting planes to the Master Problem, but in any case this must be taken into account.

Follows the general idea for the use of dual variables in the reduced cost expression in the Pricing Problem. Given a linear problem

$$\min_{x \in \mathbb{R}^n_+} \{cx : Ax \leq b\}$$

in its extended/original form.

Recall the typical expression of the reduced cost of a non basic variable. Considering all variables x_i, $i = 1, \ldots, n$. Let $\mathcal{B}$ be a subset of indexes corresponding to a basic feasible solution, i.e., a set of basic variables. Let $\mathcal{NB}$ be the set of indexes for the remaining non basic variables and c_B the coefficients of $\mathcal{B}$ in the objective function. Let $j \in \mathcal{NB}$ and A_j its corresponding column in A the matrix of constraints. Let B be the submatrix in A corresponding to the basic variables. Then, the reduced cost r_j for variable x_j is:

$$r_j = c_j - c_B B^{-1} A_j$$

From dual linear problems theory it is known the following:

PROPOSITION 6.1. *Given the linear problem*

$$\begin{aligned} &\min_{x \in \mathbb{R}^s} qx \\ &s.t. \\ &Dx \leq b \end{aligned}$$

with D matrix of size $m \times s$, b of size m and q of size s. Suppose it has an optimal solution that takes variables corresponding to a submatrix B in D. Consider its dual problem with variables $\lambda \in \mathbb{R}^m$. Then, $\lambda = c_B B^{-1}$ is a dual problem optimal solution.

We can exploit the idea in *Proposition 6.1* in the expression of r_j in the objective function of the Pricing Problem. Suppose that, at a given stage, our Master Problem is considering only a set S of variables with corresponding submatrix A_s and objective function coefficients c_s. Take $D = A_s$ and $q = c_s$ in the proposition. Then the Master Problem solution might provide with a subset of basic variables $\mathcal{B}$ in A_s that is itself a subset of basic variables in A the original problem. Let B be again its corresponding submatrix. We can substitute the term $c_B B^{-1}$ in the reduced cost expression of a new variable by the dual variables provided by the Master Problem at this stage. This way we can forget about the basic variables term in the objective function of the Pricing Problem.
Note also that, after adding a variable j to the Master Problem, in no case the Pricing Problem will add it again. This is justified by the fact that every variable already in the Master Problem has a non-negative reduced cost. Suppose j is a variable already in the Master Problem. Then in the Pricing Problem its value for the objective function would be its reduced cost, i.e.,

$$r_j = c_j - c_B B^{-1} A_j = c_j - \lambda A_j$$

Nevertheless in the dual of the Master Problem the variable definition of $x_j \geq 0$ means a constraint $\lambda A_j \leq c_j$ and so $r_j = c_j - \lambda A_j \geq 0$. The dual coefficients used in the Pricing Problem automatically force variable j to have a positive reduced cost and therefore discard its addition.

Below, it is exposed an overview of the case of Column Generation applied to CPPMIN in [**11**]. It includes an example of the use of dual variables to compute the reduced cost of a new variable. Furthermore it includes the statement of the Pricing Problem although its modelling is avoided as it might be of non use in our problem.

1. Column Generation for CPPMIN

In [**11**], CPPMIN is formulated with a new set of variables noted as x_p. For every cluster P in V_n of feasible size ($|P| \geq k$), we define x_p such that:

$$x_p = 1 \Leftrightarrow \text{Cluster } P \text{ is used in the final CPPMIN solution}$$
$$x_p = 0 \quad \text{otherwise}$$

With this new variables then we consider w_p the total weight of a given cluster P, i.e.,

$$w_p = \frac{1}{2} \sum_{i \in P} \sum_{j \in P} (a_i - a_j)^2 \tag{10}$$

Therefore, CPPMIN can be modelled as follows (11), considering all feasible P, i.e., all $P \subseteq V_n$, $|P| \geq k$:

$$\begin{aligned} &\underset{x_p,\ P \subseteq V_n}{\text{minimize}} \sum_P w_p x_p \\ &\text{subject to:} \\ &\sum_{P: v \in P} x_p = 1, \quad \text{for all } v \in V_n \\ &x_p \in \{0,1\}, \quad \text{for all } P \subseteq V_n,\ |P| \geq k \end{aligned} \tag{11}$$

The equality $\sum_{P: v \in P} x_p = 1 \quad \forall v \in V_n$ forces the solution to be a partition of V_n. Every node $v \in V_n$ must be exactly in one cluster. We will call them **partition constraints**. Note that there is one single constraint corresponding to each node.

The Master Problem is exactly the same but neglecting some variables and, more importantly, relaxing the problem. Suppose $\mathcal{P}$ are the variables that the Master Problem is considering at a given stage. Then, the Master Problem at this stage is:

$$\begin{aligned} &\underset{x_p, P \in \mathcal{P}}{\text{minimize}} \sum_{P \in \mathcal{P}} w_p x_p \\ &\text{subject to:} \\ &\sum_{P \in \mathcal{P}: v \in P} x_p = 1, \quad \text{for all } v \in V_n \\ &x_p \geq 0, \quad \text{for all } P \in \mathcal{P} \end{aligned} \tag{12}$$

Note that is not necessary to consider the constraint $x_p \leq 1$ since $\sum_{P \in \mathcal{P}: v \in P} x_p = 1$, $\forall v \in V_n$ and $x_p \geq 0$. Given the model for the Master Problem, now it only lacks an statement of the Pricing Problem.

Before moving on it, we include a brief example of the use of the dual variables in the Master Problem to express the reduced cost of a new variable, or feasible cluster.

Example 3. Suppose it is given the case $n = 3$, nodes $\{1, 2, 3\}$ and $k = 2$. Assume that at a given stage in our Master Problem we are considering the clusters $P_1 = \{1, 2\}$, $P_2 = \{2, 3\}$ and $P_3 = \{1, 3\}$. We want to know if it is worth adding the cluster $P_4 = \{1, 2, 3\}$. In this case the column A_4 corresponding to the variable x_4 of this last cluster is a vector of ones because every node belongs to P_4 (see Master Problem formulation above). We solve the dual of the Master Problem and we obtain the variables $\lambda = (\lambda_1, \lambda_2, \lambda_3)$

corresponding to each constraint (corresponding to each node in V_n). Then we simply consider the reduced cost of P_4 the new variable as $r_4 = w_4 - \lambda A_4 = w_4 - \lambda_1 - \lambda_2 - \lambda_3$ ■.

With this example it is shown the use of dual variables from the Master Problem. Every dual variable from the Master Problem corresponds to a partition constraint in the Master Problem. Nevertheless, Master Problem formulation above shows that every partition constraint is associated to a single node. This implies that a pricing coefficient can be assigned to each node by taking its corresponding dual variable in the dual solution of the Master Problem.
Then, considering the reduced cost rexpression with dual variables it is easy to see that every dual variable is multiplying the coefficient of its corresponding node in the column to be added to the Master Problem, which is always 1. In other words if we want to add a new cluster P, then its reduced cost will be:

$$r_p = w_p - \sum_{i \in P} \lambda_i$$

In the example above the new cluster P included every node and the previous expression is clearly coherent.

On account of all that, in [**11**], the Pricing Problem is constructed as follows:

Construct a weighted undirected graph $K_n = (V_n, E_n)$ with edge weights the square distances between the nodes as in the previous chapter. Nevertheles, now there are node weights included in the graph. These node weights correspond to the value of the dual variables in the solution of the Master Problem. We can also call them pricing coefficients.
Then a new variable x_p is generated by finding a cluster P with size at least k in this graph such that the sum of its edge weights minus the sum of its node weights is negative. Finding this cluster P in the -edge and node- weighted graph defined is directly the Pricing Problem statement in [**11**].

In [**11**], there is also a model included for this problem as a BIP. We do not report it because we reconsider the variables to use for it. Instead, the statement as a clique problem is essential for our contribution and, on account of that, we have included it above.

Chapter 7
Column Generation on Non-Approximate Microaggregation

In this chapter, we are going to introduce new contributions of this project on non approximate microaggregation. The idea of applying a Column Generation approach is inspired in [**11**]. However, in our case we modify the weight of a cluster coefficient and more importantly the whole Pricing Problem scheme for new cluster generation.

1. Master Problem & Column Generation Scheme

Following the notation of variables in [**11**], suppose P is a feasible cluster in V_n. Taking into account the result in [**6**] for the maximum size in optimal microaggregation we can state that $k \leq |P| \leq 2k-1$. Let x_p be the binary variable that indicates whether cluster P is or not in the solution for microaggregation. Let w_p be the weight of P. In [**11**], w_p was taken as the sum of all inner square distances. In our case we are considering the same sum but divided by the number of elements such that the result is the real contribution to the SSE of cluster P in non-approximate microaggregation. From *Chapter 4. Section 1.* we already know it can be computed as:

$$w_p = \frac{1}{2}\sum_{i\in P}\frac{\sum\limits_{j\in P}(a_i - a_j)^2}{|P|} \tag{13}$$

On account of this we can describe non-approximate microaggregation as follows (14) (same as in [**11**] for CPPMIN but modifying the weights of the clusters). Considering all feasible clusters P in optimal microaggregation, i.e., clusters with size $k \leq |P| \leq 2k-1$,

$$\begin{aligned}
&\underset{x_p,\ P\subseteq V_n}{\text{minimize}} \sum_{P} w_p x_p \\
&\text{subject to:} \\
&\sum_{P:v\in P} x_p = 1, \quad \text{for all } v \in V_n \\
&x_p \in \{0,1\}, \quad \text{for all } P \subseteq V_n,\ k \leq |P| \leq 2k-1
\end{aligned} \tag{14}$$

Remark 5. With this new model we can easily skip the difficulty of the centroid by computing w_p with equation (5). This is the reason why we explore these new variables from [**11**].

Exactly as in [**11**], our Master Problem takes as reference the relaxed formulation of non-approximate microaggregation. Suppose that $\mathcal{P}$ are the variables that the Master Problem is considering at a given stage.

Then the Master Problem at this stage is again (12):

$$
\begin{aligned}
&\underset{x_p, P\in\mathcal{P}}{\text{minimize}} \sum_{P\in\mathcal{P}} w_p x_p \\
&\text{subject to:} \\
&\sum_{P\in\mathcal{P}:v\in P} x_p = 1, \quad \text{for all } v \in V_n \\
&x_p \geq 0, \quad \text{for all } P \in \mathcal{P}
\end{aligned} \tag{12}
$$

This means that the Master Problem is the same as in [**11**] with only the modification in the weight of a cluster provided by equation (13). When it comes to the Pricing Problem there raise important differences.

Recall Example 3. in *Chapter 6. Section 1.* extracted from [**11**]. The same use of dual variables from the Master Problem to express the reduced cost of a new feasible cluster can be applied in our procedure. This means that we also consider the problem of finding a feasible cluster in a complete undirected weighted graph with weights in the edges and in the nodes. However, in [**11**], the Pricing Problem looks for a cluster of size at least k that minimizes the sum of edge weights minus the sum of node weights in this cluster. In our case, our Pricing Problem is solved by finding a cluster of fixed size η that minimizes the difference between edge weights (divided by η) and node weights. Then, we solve this Pricing Problem k times, one for each $\eta \in \{k, \ldots, 2k-1\}$. For every η we can add a new feasible cluster. This means we preserve the "statement" of the Pricing Problem from [**11**] although our objective cluster has a fixed size. Moreover, the column generation scheme changes on account of the fact that we solve k different Pricing Problems with fixed size at every stage.

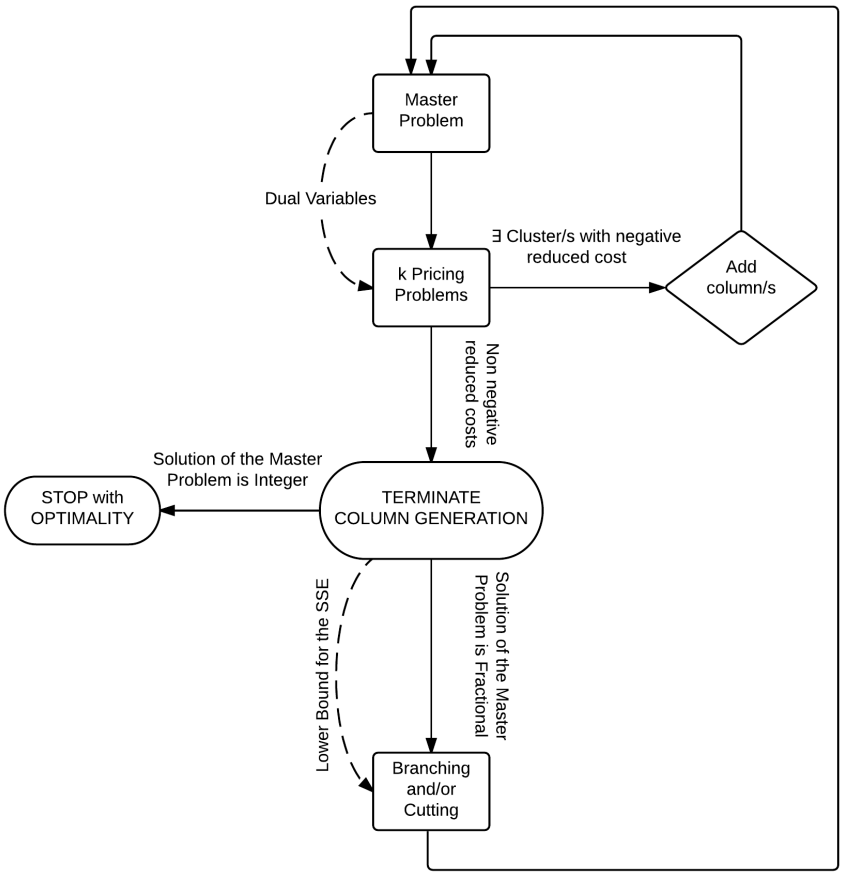

FIG. 1. Procedure to solve non-approximate microaggregation to optimality.

It is clear that, when none of these k Pricing Problems adds a cluster with negative reduced cost (i.e., when the objective function value for every Pricing Problem is non-negative), then, there are no more feasible clusters to consider in the Master Problem. At this point, the solution of the Master Problem must checked. If the solution is integer (0 or 1 given our constraints) then, the scheme has provided with an optimal microaggregation. Else if the solution is fractional, then some branching and/or cutting must be performed. This Branching eventual procedure implies reinitializing the column generation scheme with a new constraint. Nevertheless, at this point we must be aware that, in any case, the first column generation scheme provides with a valid lower bound for the total SSE of optimal microaggregation, which corresponds to the objective function of the Master Problem with this fractional solution obtained.
The Block Diagram in Figure 1. illustrates the scheme described above to obtain optimal non-approximate microaggregation. By now we will focus on the upper block of Column Generation, starting from the Pricing Problem which we will formulate as a BIP.

2. Pricing Problem with fixed cluster size

At this point, we have a statement for our Pricing Problem given η the size of a new cluster to generate. Given $K_n = (V_n, E_n)$ our complete undirected graph with edge weights c_e, $e \in E_n$, and node weights, λ_v, $v \in V_n$, we want to obtain a feasible cluster P, $|P| = \eta$, such that the difference between its edge weights (divided by η) and its node weights is negative.

The objective now is to propose a BIP model for our Pricing Problem with fixed cluster size. Given that we are looking for a negative reduced cost (computed as the difference mentioned above), we can simply minimize it in the problem model.
In [**11**], there is given a BIP model for the Pricing Problem. However the model used in [**11**] includes variables for every node and edge in K_n the weighted graph to be explored. In our case we will simply include the edge variables because we have noticed that, with our formulation, there are fractional results excluded which satisfy the formulation in [**11**]. Moreover, thus we can work with less variables. It means, from now on we will note clusters as cliques given the edge variables description.

First of all, we suppose $k > 1$ because otherwise, clearly, optimal microaggregation would mean to take each node as a single clique and final SSE would be 0. This means that in every case $\eta \geq 2$. Suppose P is the new cluster we want to generate with the Pricing Problem. As stated above, we define z_e, $e \in E_n$, $e = (i, j)$, $i, j \in V_n$, $i \neq j$ such that:

$$z_e = 1 \Leftrightarrow i,\ j \text{ are in cluster } P$$
$$z_e = 0 \quad \text{otherwise}$$

Note that with these variables the solution of the Pricing Problem will be a clique of size η if,

$$\sum_{e \in E_n} z_e = \frac{\eta(\eta - 1)}{2}$$

The equation above will be our fixed size constraint in our formulation. Another group of constraints could be the so called triangle inequalities, exactly as in *Chapter 4*. Using them we can ensure the problem solution provides with a clique in edges.

$$z_a + z_b - z_c \leq 1 \ \text{ for all } a,\ b,\ c \text{ triangle in } E_n$$

Differently to CPP however, it lacks to force the solution to represent a single clique. In CPP we looked for a clique partitioning and adding the triangle inequalities was useful to enforce every component (set in the partitioning) is a clique. In our case, we must also force that there is only one component, i.e., the solution is connected.
For example if we consider the case $n = 6$, $V_n = \{1, 2, 3, 4, 5, 6\}$, $\eta = 3$, the solution provided by activating edges $(1, 2)$, $(3, 4)$ and $(5, 6)$ satisfies the fixed size constraint and the triangle inequalities. It is actually a clique partitioning in three separated subcliques. We must discard this type of solution by forcing connectivity. To do that we introduce the *Node-to-Node Inequalities*.

Definition: Let i, $j \in V_n$, $i \neq j$. We define the ***Node-to-Node Inequality*** for i, j as,

$$\sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta - 2) z_{ij} \geq 0$$

This inequality is clearly valid. Let $z \in Z$ be a feasible clique. The inequality contains negative terms in the left hand side only when $z_{ij} = 1$. This is therefore the only case to explore. Nevertheless in this case clearly nodes i and j are included in the clique. Then, the amount of activated edges connecting i except z_{ij} is clearly $\eta - 2$ given that the clique must be feasible (size η). Consequently, the inequality is satisfied (at equality).

Moreover, this inequality makes the solution connected. Any type of solution which is not connected consists on different separated cliques. All those clusters must have size strictly less than η given the fixed size equation and that we are supposing there is more than one clique component. Then, taking i, j two nodes in one of those cliques (they must have size at least two to exist), trivially the *Node-to-Node Inequality* for i, j is violated.

Note that given a pair of nodes i, $j \in V_n$, $i \neq j$, there exist two *Node-to-Node Inequalities* for i, j:

$$\sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta - 2) z_{ij} \geq 0$$

$$\sum_{e \in \delta(j) \setminus (i,j)} z_e - (\eta - 2) z_{ij} \geq 0$$

Another important observation is that considering all existing *Node-to-Node Inequalities* and the fixed size constraint, forces the solution to be complete, i.e., to describe a clique.

PROPOSITION 7.1. *Given $\eta \geq 2$, the Node-to-Node Inequalities for i, $j \in V_n$, $i \neq j$, combined with the fixed size equation, force the solution to be clique.*

PROOF. The case $\eta = 2$ is trivial. The fixed size equation itself forces the solution to be a clique.
For $\eta \geq 3$, let us suppose the edge (v, w) is activated between nodes v, $w \in V_n$. Imposing the *Node-to-Node Inequality*,

$$\sum_{e \in \delta(v) \setminus (v,w)} z_e - (\eta - 2) z_{vw} \geq 0$$

means there are at least $\eta - 2$ other nodes connected to v apart from w. Let us call $\mathcal{K} \subset V_n$ this set of nodes, $|\mathcal{K}| \geq \eta - 2$. Then we impose the reverse *Node-to-Node Inequality* for v, w

$$\sum_{e \in \delta(w) \setminus (v,w)} z_e - (\eta - 2) z_{vw} \geq 0$$

which forces w to be connected to at least $\eta - 2$ nodes apart from v. To start we can suppose first those nodes are $\mathcal{K}$ too. Then we take k_1 a node in $\mathcal{K}$ and we impose the *Node-to-Node Inequality* for k_1, v for example (we can consider the one for k_1, w too). Given we are supposing k_1 is connected to both v, w, it means k_1 is connected to at least $\eta - 3$ other nodes. We can suppose those nodes are yet the other nodes in $\mathcal{K}$. We could do the same for the next k_t nodes in $\mathcal{K}$. At every stage we would be adding exactly $|\mathcal{K}| - t$ new edges.
If $|\mathcal{K}| = \eta - 2$, then the resulting amount of edges would be

$$\begin{aligned} &1 + \eta - 2 + \eta - 2 + \sum_{t=1}^{\eta-2} \eta - 2 - t = \\ &= 2\eta - 3 + (\eta - 2)^2 - \sum_{t=1}^{\eta-2} t = \\ &= 2\eta - 3 + (\eta - 2)^2 - \frac{(\eta - 2)(\eta - 3)}{2} = \\ &= \frac{4\eta - 6 + \eta^2 - 5\eta + 6}{2} = \frac{\eta(\eta - 1)}{2} \end{aligned}$$

and the fixed size constraint would be satisfied. Besides, the solution would be a clique.
Note that if $|\mathcal{K}| > \eta - 2$, then, applying succesively the *Node-to-Node Inequality* as above, would lead us to a greater amount of edges and this would violate the fixed size constraint. More than that, if at some point our construction was not strictly satisfied, in the sense that at some stage t the nodes connected to k_t where not exactly the nodes in $\mathcal{K}$ plus v and w, then, there would be an external node $z \notin \mathcal{K} \cup \{v, w\}$ connected to k_t. If we took the *Node-to-Node Inequality* for z, k_t this node z would be connected to at least $\eta - 2$ nodes different than k_t. This nodes could be in $\mathcal{K}$ or not but in any case this term $\eta - 2$ added in the sum of edges described above would eventually end up violating the equality between the total amount of edges and the right hand side of the fixed size equation. The same reasoning works for the instance previous to k_1 when we suppose that the other nodes connected to w apart from v are the ones in $\mathcal{K}$.
In other words, the only possible edges construction that ends up satisfying the fixed size constraint is the one described above. Otherwise the *Node-to-Node Inequalities* would force to introduce an excess of edges. □

COROLLARY 7.1.1. *The triangle inequalities are not necessary in our formulation of the Pricing Problem with fixed size.*

Carrying on with the model construction of the Pricing Problem with variables z_e (recall those variables describe a cluster P solution of the Pricing Problem).

Given $e \in E_n$, $e = (i, j)$, if we consider $c_e = (a_i - a_j)^2$, it is clear that with these variables we can express w_p the weight of P in non-approximate microaggregation (see equation (13) in *Chapter 7.*) as:

$$w_p = \sum_{e \in E_n} \frac{c_e z_e}{\eta}$$

This expression for w_p is the edge weights positive contribution to the objective function in the Pricing Problem with fixed cluster size. Now we look for the node weight negative contribution in terms of the variables z_e.

As in [**11**], let λ_v be the node weight of $v \in V_n$. Recall that this corresponds to the dual variable solution of the dual of the Master Problem. Then note that if the node v is in the solution P of the Pricing Problem, then v will be adjacent to $\eta - 1$ nodes. Otherwise, v will be adjacent to zero nodes. This means that clearly the expression

$$\frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1}$$

is 1 when $v \in P$ and 0 otherwise.
On account of that, we can simply compute the node weight contribution to the objective function as:

$$\sum_{v \in V_n} \lambda_v \frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1}$$

In summary, the objective function in our Pricing Problem with fixed size is:

$$\sum_{e \in E_n} \frac{c_e z_e}{\eta} - \sum_{v \in V_n} \lambda_v \frac{\sum\limits_{e \in \delta(v)} z_e}{\eta - 1}$$

What leads us to a BIP model (15) of the Pricing Problem with fixed cluster size $\eta \geq 2$:

$$\underset{z_e,\, e\in E_n}{\text{minimize}} \sum_{e\in E_n} \frac{c_e z_e}{\eta} - \sum_{v\in V_n} \lambda_v \frac{\sum\limits_{e\in\delta(v)} z_e}{\eta - 1}$$

subject to:

$$\sum_{e\in E_n} z_e = \frac{\eta(\eta-1)}{2} \tag{15}$$

$$\sum_{e\in\delta(i)\setminus(i,j)} z_e - (\eta-2) z_{ij} \geq 0, \quad \text{for all } i,\, j \in V_n,\, i \neq j$$

$$z_e \in \{0,1\}, \quad \text{for all } e \in E_n$$

Chapter 8
Polyhedral study of the Pricing Problem with fixed cluster size

To study our Pricing Problem first we impose two bound limits for η. The first one is already taken into consideration in the problem definition when we impose $\eta \geq 2$. There is also an upper bound limit that imposes $\eta \leq n-2$. Otherwise when $\eta = n$ the only solution is the complete set of nodes and when $\eta = n-1$ we simply take the complementary of every node in V_n as a solution. In total those are only $n+1$ solutions easy to enumerate and therefore we discard those cases in our analysis. In summary, we consider $2 \leq \eta \leq n-2$.

Given an Integer Programming problem to be solved in Z an integer, binary or mixed set, a polyhedral study of the problem consists generally on studying the following issues:

(i) Dimension of conv(Z).
(ii) Valid inequalities for Z.
(iii) Which inequalities in *(ii)* are also facet-defining inequalities for conv(Z).

This corresponds to the scheme previously seen in CPP and CPPMIN. We will follow this scheme in our analysis. First of all, consider all the possible clusters of size η in V_n and describe them as cliques in terms of the variables of our Pricing Problem. From now on, let $Z \subseteq \{0,1\}^{|E_n|}$ be this set of feasible clusters described as cliques in edges.

$$Z = \{z \in \{0,1\}^{|E_n|} \mid \sum\nolimits_{e \in E_n} z_e = \frac{\eta(\eta-1)}{2} \\ \sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta-2) z_{ij} \geq 0, \quad \text{for all } i,\, j \in V_n,\, i \neq j\}$$

Every $z \in Z$ corresponds to a clique in K_n of size η. Let $\mathcal{P}_{n,\eta}$ correspond to the Convex Hull of solutions of the Pricing Problem with fixed size, in other words, $\mathcal{P}_{n,\eta} = \text{conv}(Z)$.

Before moving on to results of the polyhedral analysis we might first introduce a lemma that might be very useful for working on dimensionality of polyhedrons and its faces.

LEMMA 8.1. *Given $P \subseteq \mathbb{R}^m$ a polyhedron, the dimension of P is m minus the number of linearly independent linear equations satisfied in P.*

PROOF. Recall Remark 3. in *Chapter 3. Section 2.* In this remark we have noted that given $P \subseteq \mathbb{R}^m$ a polyhedron, its orthogonal has dimension $m-\dim(P)$. Recall that by orthogonal of P we mean the orthogonal of the vector space defined by the directions of P. To prove the lemma, we can find a correspondence between linear equations satisfied in P and vectors in the orthogonal of P.
This correspondence is trivial since it is the same one used in affine geometry to describe affine subspaces as intersection of hyperplanes. Recalling Remark 3., let $\mathbb{V}$ be the vector space given by the directions in P. It is constructed from differences $z^i - z^j \in \mathbb{R}^m$ for pairs z^i, $z^j \in P$ and its orthogonal vector space is the orthogonal of P.

Now, let $\pi \in \mathbb{R}^m$ be a vector in the orthogonal of P. Clearly $\pi(z_i - z_j) = 0$ for every pair z_i, $z_j \in P$. This means the product πz takes one single value, denoted by π_o, for every $z \in P$ and therefore equation $\pi z = \pi_o$ is satisfied for all $z \in P$.
On the other hand, let $\pi z = \pi_o$ be a linear equation satisfied for all $z \in P$, then clearly $\pi(z_i - z_j) = 0$ for every $z_i - z_j$ direction in P. This immediately means, for construction of $\mathbb{V}$, that π belongs to the orthogonal of P. This proves that every linear equation in P corresponds to a direction in the orthogonal of P and viceversa. Therefore, it is immediate that the number of linearly independent linear equations in P is exactly the number of linearly independent directions in the orthogonal of P. □

From now on, we will make an abuse of notation when referring to linear equations simply as equations. This way the writing will be simplified. Bearing *Lemma 8.1* in mind, we can proceed with a relevant result on the dimension of $\mathcal{P}_{n,\eta}$.

THEOREM 8.2. *dim($\mathcal{P}_{n,\eta}$)* $= |E_n| - 1$

PROOF. First we note the following:

Claim 1: Proving the theorem is equivalent to proving that the only equation (except for multiplicity) satisfied by all points in $\mathcal{P}_{n,\eta}$ is the fixed size equation,

$$\sum_{e \in E_n} z_e = \frac{\eta(\eta - 1)}{2}$$

Proof of Claim 1: Immediate using *Lemma 8.1* and the definition of $\mathcal{P}_{n,\eta} = \text{conv}(Z)$ ■.

We denote $\pi \in \mathbb{R}^{|E_n|}$ as the vector with all ones. We also denote $\pi_o = \eta(\eta - 1)/2$. In other words, let $\pi z = \pi_o$ be the fixed size equation. Let $az = a_o$ be an equation satisfied by all $z \in \mathcal{P}_{n,\eta}$. We must prove it is a multiple of $\pi z = \pi_o$ to prove the theorem. We must prove, then, that all coefficients in $a \in \mathbb{R}^{|E_n|}$ are equal. It is clear that in this case, if $a_c \equiv a_e$: ct for all $e \in E_n$, then a_o necessarily would be $a_o = a_c \pi_o$, otherwise the equation $az = a_o$, would not be coherent with the fixed size equation.

Claim 2: Let W, $H \subset V_n$ with $|W| = |H| = \eta$. Then,

$$\sum_{e \in E_n(W)} a_e = \sum_{e \in E_n(H)} a_e$$

Proof of Claim 2: Immediate noting that W, H correspond to two different cliques in $Z \subseteq \mathcal{P}_{n,\eta}$ ■.

Claim 3: Given i, $j \in V_n$, $i \neq j$. Let $T = V_n \setminus \{i, j\}$ and let $\mathcal{K} \subset T$, $|\mathcal{K}| = \eta - 1$. Note that $\mathcal{K}$ exists given that $|T| = n - 2 \geq \eta$. Then,

$$\sum_{k_s \in \mathcal{K}} a_{ik_s} = \sum_{k_s \in \mathcal{K}} a_{jk_s}$$

Proof of Claim 3: Let $W = \mathcal{K} \cup \{i\}$, $H = \mathcal{K} \cup \{j\}$. Then $|W| = |H| = \eta$ and we apply Claim 2 to W and H. Then,

$$\sum_{e \in E_n(W)} a_e = \sum_{e \in E_n(H)} a_e$$

If we remove the common terms a_e with $e \in E_n(\mathcal{K})$ from both sides, there only remain the terms a_{ik_s}, a_{jk_s} in the respective right and left hand sides, with $k_s \in \mathcal{K}$ ■.

Claim 4: Given i, $j \in V_n$, $i \neq j$. Let $T = V_n \setminus \{i, j\}$ and let $m \in T$. Then

$$a_{im} = a_{jm}$$

Proof of Claim 4: Let g, $l \in T$, $g \neq l$. Note they do exist given that $|T| = n - 2 \geq \eta \geq 2$ what means $n \geq 4$ and $|T| \geq 2$.
Let $\mathcal{K}_1$, $\mathcal{K}_2 \subset T$ of size $\eta - 1$ such that $\{g, l\} = (\mathcal{K}_1 \cup \mathcal{K}_2) \setminus (\mathcal{K}_1 \cap \mathcal{K}_2)$. This means $\mathcal{K}_1$, $\mathcal{K}_2$ have $\eta - 2$ common elements and then one of them contains g and the other contains l. Note those sets exist given that to construct them we only lack to have at least $\eta - 2$ elements in $T \setminus \{g, l\}$. Nevertheless, $|T \setminus \{g, l\}| = n - 4 \geq \eta - 2$.

We suppose without loss of generality that $g \in \mathcal{K}_1$ and $l \in \mathcal{K}_2$.
We apply Claim 3 to both sets $\mathcal{K}_1$ and $\mathcal{K}_2$. This implies

$$\sum_{k_s \in \mathcal{K}_1} a_{ik_s} = \sum_{k_s \in \mathcal{K}_1} a_{jk_s}$$

$$\sum_{k_s \in \mathcal{K}_2} a_{ik_s} = \sum_{k_s \in \mathcal{K}_2} a_{jk_s}$$

Given that $\{g, l\} = (\mathcal{K}_1 \cup \mathcal{K}_2) \setminus (\mathcal{K}_1 \cap \mathcal{K}_2)$ if we take the difference between the two equations above it leads to

$$a_{ig} - a_{il} = a_{jg} - a_{jl} \qquad (i)$$

Equation (i) above is valid for every pair g, $l \in T$ given $g \neq l$. Now consider $m \in T$ from the statement of Claim 4. Let $L \subset T \setminus \{m\}$ be and arbitrary set of size $\eta - 2$. Note that L exists given that $|T \setminus \{m\}| = n - 3 > \eta - 2$. Let us denote the terms in L as $L = \{l_1, \ldots, l_{\eta-2}\}$. For each pair $\{m, l_r\}$, $r = 1, \ldots, \eta - 2$ consider its corresponding equation (i):

$$a_{im} - a_{il_1} = a_{jm} - a_{jl_1}$$

$$\vdots$$

$$a_{im} - a_{il_{\eta-2}} = a_{jm} - a_{jl_{\eta-2}}$$

Summing up all the equations above leads to

$$(\eta - 2)a_{im} - a_{il_1} - \ldots - a_{il_{\eta-2}} = (\eta - 2)a_{jm} - a_{jl_1} - \ldots - a_{jl_{\eta-2}} \qquad (ii)$$

Finally we consider $\mathcal{K} = \{m, l_1, \ldots l_{\eta-2}\} = L \cup \{m\} \subset T$. Note that $\mathcal{K}$ has size $\eta - 1$ and we can apply Claim 3 to $\mathcal{K}$

$$a_{im} + a_{il_1} + \ldots + a_{il_{\eta-2}} = a_{jm} + a_{jl_1} + \ldots + a_{jl_{\eta-2}}$$

Adding this equation to (ii) leads to the equality

$$(\eta - 1)a_{im} = (\eta - 1)a_{jm}$$

and given that $\eta \geq 2$ it is equivalent to the statement of Claim 4 ■.

It only lacks to prove $a_{it} = a_{jp}$ for all t, p, i, j, $i < t$, $j < p$. Note that, if there is an index coincidence between $\{i, t\}$, $\{j, p\}$, then the equality directly corresponds to Claim 4 with m the repeated index. Therefore we only lack to consider the case when all indexes differ, i.e., $\{i, t\} \cap \{j, p\} = \emptyset$.
Applying Claim 4 for the pair $\{i, j\}$ and $m = t$, p it is valid that

$$a_{it} = a_{jt}$$

$$a_{ip} = a_{jp}$$

Now if we consider the pair $\{t, p\}$ and apply Claim 4 with $m = i$ then

$$a_{it} = a_{ip}$$

In conclusion, combining the equations above it is immediate that

$$a_{it} = a_{jp}$$

□

1. Introduction to Inequalities on the fixed size Pricing Problem

Recall this chapter studies the already existent valid inequalities, provides with new families of valid inequalities and also studies under which circumstances they are facet-defining. This section in particular introduces useful theory applied and also the methodology followed to obtain these results.

Theorem 8.2 is stating that the only equation satisfied by all the points in $\mathcal{P}_{n,\eta}$ is the size constraint. This is essential to prove if an inequality is facet-defining.
Recall Z is the set of all feasible clusters of size η described as cliques with the variables of our Pricing Problem model. Recall that $\mathcal{P}_{n,\eta} = \text{conv}(Z)$, $\mathcal{P}_{n,\eta} \subseteq \mathbb{R}^{|E_n|}$, with dimension $|E_n| - 1$. Given a valid inequality $\alpha z \leq \beta$ for Z, there are two approaches to prove if it is facet-defining or not for $\mathcal{P}_{n,\eta} = \text{conv}(Z)$.

(1) Take the constraint as an equation $\alpha z = \beta$ and find $|E_n| - 1$ affinely independent points in Z satisfying it. This is simply applying the definition since it would mean $\dim(\{z \in \mathcal{P}_{n,\eta} \mid \alpha z = \beta\}) = |E_n| - 2$. This strategy in general is not very practical and in particular in our problem will not be used. Nevertheless, if we cannot find such a set of points in Z, it is clear that it is not facet-defining. This is justified by the fact that the number of independent directions generated by differences of points in Z is the same as the number of independent directions in $\text{conv}(Z)$. The proof of this is trivial simply by writing the shape of an element in $\text{conv}(Z)$.
(2) The second approach is based on the following lemma.

LEMMA 8.3. *Let $\alpha z \leq \beta$ be a valid inequality for Z. Then,*

$$conv(\{z \in Z \mid \alpha z = \beta\}) = conv(Z) \cap \{z \in \mathbb{R}^{|E_n|} \mid \alpha z = \beta\}$$

PROOF. Recall Z is finite (its elements are the cliques of size η in V_n, i.e., subsets with size η in V_n). Let us write $Z = \{z^1, \ldots, z^m\}$. Let r be the number of elements $z^i \in Z$ satisfying $\alpha z^i = \beta$. If $r = 0$, clearly the left hand side is the empty set. Nevertheless, it is easy to see that the right hand side set is empty too. Suppose there exists $z \in \text{conv}(Z) \cap \{z \in \mathbb{R}^{|E_n|} \mid \alpha z = \beta\}$. Imposing $\alpha z = \beta$ and writing z as a convex combination of $z^1, \ldots, z^m$ directly implies,

$$\alpha z = \lambda_1 \alpha z^1 + \ldots + \lambda_m \alpha z^m = \beta, \quad \lambda_i \geq 0, \quad \sum_{i=1}^{m} \lambda_i = 1 \qquad (*)$$

However, since $\alpha z \leq \beta$ is valid and we have supposed $r = 0$, clearly $\alpha z^i < \beta$. It leads then to contradiction in (*) and then, $\text{conv}(Z) \cap \{z \in \mathbb{R}^{|E_n|} \mid \alpha z = \beta\}$ must be also empty and the lemma is satisfied.
Now let $r \geq 1$. We prove by inclusion. The inclusion in the $\subseteq$ sense is trivial.
To prove the inclusion in the $\supseteq$ sense we start taking a given $z \in \text{conv}(Z) \cap \{z \in \mathbb{R}^{|E_n|} \mid \alpha z = \beta\}$. Again we write z as a convex combination of $z^1, \ldots, z^m$,

$$z = \lambda_1 z^1 + \ldots + \lambda_m z^m, \quad \lambda_i \geq 0, \quad \sum_{i=1}^{m} \lambda_i = 1$$

We can suppose without loss of generality that the first r points in Z are the ones satisfying $\alpha z^i = \beta$. Imposing this and also imposing $\alpha z = \beta$, it directly implies,

$$\lambda_{r+1} \alpha z^{r+1} + \ldots \lambda_m \alpha z^m = \lambda_{r+1} \beta + \ldots \lambda_m \beta$$

Similarly as in the case $r = 0$, we can note that the points z^{r+j} all satisfy $\alpha z^{r+j} < \beta$ given the inequality is valid and that we have excluded all z^i with $\alpha z^{r+j} = \beta$. Therefore the only possibility is that $\lambda_{r+j} = 0$ $j = 1, \ldots, m - r$. This immediately implies $\supseteq$. □

Now suppose there exists an equation $\mu z = \mu_o$ satisfied by all $\{z \in Z \mid \alpha z = \beta\}$. Clearly the Convex Hull of this set will still satisfy $\mu z = \mu_o$. This together with *Lemma 8.3* above results in $\{z \in \text{conv}(Z) \mid \alpha z = \beta\} \subseteq \{z \in \text{conv}(Z) \mid \mu z = \mu_o\}$. Then proving that $\mu z = \mu_o$ is linearly dependent with $\alpha z = \beta$ and $\sum_{e \in E_n} z_e = \eta(\eta - 1)/2$ is equivalent to prove that $\{z \in \text{conv}(Z) \mid \alpha z = \beta\}$, the face defined by $\alpha z \leq \beta$, has dimension $|E_n| - 2$ and therefore $\alpha z \leq \beta$ is facet-defining. To see that simply, recall that the dimension of a facet is $|E_n| - 2$ for *Theorem 8.2* and note also that *Lemma 8.1* establishes a correspondence between linearly independent equations and dimensions lost in a given

polyhedron. From a dimensions point of view, the face $\{z \in \text{conv}(Z) \mid \alpha z = \beta\} \subseteq \mathbb{R}^{|E_n|}$ has dimension $|E_n| - 2$ (note that a face is a polyhedron) if and only if there are at most two linearly independent equations satisfied in $\{z \in \text{conv}(Z) \mid \alpha z = \beta\}$, see *Lemma 8.1.* Clearly, those two equations must be the fixed size equation and the equation $\alpha z = \beta$ defining the face.
In summary, $\alpha z \leq \beta$ is facet-defining if and only if we can write any equation $\mu z = \mu_o$ satisfied by all $\{z \in Z \mid \alpha z = \beta\}$ as a linear combination of $\alpha z = \beta$ and $\sum_{e \in E_n} z_e = \eta(\eta - 1)/2$.

In our study, for general cases we will use only approach 2 to prove when a valid inequality is facet-defining. Moreover, to prove a valid inequality is not facet-defining under a given circumstance we will usually look for a linearly independent equation containing the face defined by the given valid inequality. However, it does not mean that approach 1 has not been used to prove the facet-defining condition with computer assistance in certain cases, see the paragraph below.

In fact, to obtain the following results computer assistance has been used since the beginning. There is an open software tool used to obtain a non redundant inequalities description of the Convex Hull of a given set of integer points. This software is called Porta, see [**4**] a reference webpage for this software.

The procedure was the following. With a simple program in C++ the 0-1 solutions for the Pricing Problem for lower node cases ($n = 4$, $n = 5$, $n = 6$ and $n = 7$) and sizes $\eta = 2, \ldots, n - 2$, were generated. That was simply generating subsets of size η in V_n and writing them in terms of edges in the model variables description, i.e., the corresponding set Z entirely. Those solutions where plugged into Porta and it provided with the facet-defining inequalities of $\text{conv}(Z)$ for those lower cases. Those inequalities, however, are written in terms of the variables. The objective is then to interpret these inequalities, see to what idea of constraint they respond and therefore generalise them for greater cases. Note that, nevertheless, with this procedure a priori two inequalities of the same family might seem different because any inequality can be written as a multiple of itself plus a multiple of the fixed clique size equality. It requires then a laborious work to distinguish inequalities of the same type in Porta output. With this procedure anyway it is possible to detect valid inequalities and lacks only to explore in which cases they are facet-defining.
To do so, the main strategy applied was to extract the solutions Z plugged into Porta that also satisfied those inequalities at equality, i.e., the set $\{z \in Z \mid \alpha z = \beta\}$. Then Matlab was used to check if this set contained at least $|E_n| - 1$ affinely independent points. In case it was, the facet-defining condition would be empirically proved for this case by approach 1.
With this strategy it was easy to empirically see under which circumstances those inequalities were facet-defining. Afterwards it only lacked to extend the proof in the general case. This theoretical extended proof in any case applied approach 2 explained above.
In the case we empirically detected a given valid inequality was not facet-defining in a given case, Matlab was also used to prove it in general. Suppose $\alpha z \leq \beta$ is a valid inequality which we detected that was no facet-defining for a certain case (the set $Z \cap \{\alpha z = \beta\}$ did not contain at least $|E_n| - 1$ affinely independent points). Simply we take again the set $\{z \in Z \mid \alpha z = \beta\}$ and use commands in Matlab to obtain the orthogonal of the affine space those points generate. This orthogonal directly corresponds to equations satisfied at equality by all these points. The idea is then to study these equations and generalise them. This way it is relatively easy to obtain linearly independent equations of the type $\pi z = \pi_o$ containing $\{z \in Z \mid \alpha z = \beta\}$.

After this introduction intending to explain the methodology applied, follows the main theoretical results obtained on the facial structure of our Pricing Problem. Those are divided into two main sections: *Original Inequalities for the Pricing Problem with fixed size* and *New Inequalities for the Pricing Problem with fixed size.*

2. Original Inequalities for the Pricing Problem with fixed size

In this section we study the original inequalities for the Pricing Problem with fixed size. With the term "original" we mean the inequalities appearing in our initial formulation for the Pricing Problem with fixed size. This includes the variable bounds $0 \leq z_{ij} \leq 1$ and the so called *Node-to-Node Inequalities.*

First of all, recall that our analysis establishes that $\eta \geq 2$ and $\eta \leq n-2$. This immediately means we consider n at least 4.

Another preliminary observation is noting that the valid inequalities $z_e \leq 1$ are never facet-defining. The same argument as in CPP taking two triangle inequalities and adding them works here, see CPP in *Chapter 5*. Moreover, given that we are not using the triangle inequalities in our formulation (they are not necessary to make the solution a clique, see *Proposition 7.1*), we can also note that the face defined by $z_e = 1$ is precisely contained in the face defined by a *Node-to-Node inequality*.
Given $e \in E_n$ and $z_e \leq 1$ its upper bound inequality, $e = (i, j)$ for some i, $j \in V_n$. Then, given $z \in Z$ such that $z_e = 1$, it means the clique corresponding to z contains nodes i and j. It is clear, then, that z satisfies $\sum_{e \in E_n} z_e - (\eta - 2) z_{ij} \geq 0$ given that z corresponds to a feasible clique containing i and j. Then using *Lemma 8.3* it is immediate that the face defined by $z_e = 1$ in $\mathcal{P}_{n,\eta} = \text{conv}(Z)$ is contained in the one defined by $\sum_{e \in E_n} z_e - (\eta - 2) z_{ij} = 0$.
This is a simple preliminary observation. The section is concluded with *Propositions 8.4, 8.5* below.

PROPOSITION 8.4. *The inequalities $z_{ij} \geq 0$, $(i, j) \in E_n$, are valid for Z. For $\eta = 2$ they are always facet-defining for $\mathcal{P}_{n,\eta}$. For $\eta > 2$ they are facet-defining if and only if $n \geq \eta + 3$.*

PROOF. Validity is clear, they are simply lower bounds on the variables.

Now let us consider the case $\eta = 2$ and the inequality $z_{ij} \geq 0$ for i, j different in V_n. Given that $2 \leq \eta \leq n-2$ it is clear that $n \geq 4$. Let us suppose there exists an equation $\mu z = \mu_o$ satisfied by all $\{z \in Z \mid z_{ij} = 0\}$. Clearly any $z \in Z$ satisfying $z_{ij} = 0$ corresponds to a clique containing i, j or none of them, but in no case can contain both. Let us consider r, q, two different nodes in V_n appart from i and j (they exist given that $n \geq 4$). Now consider the cliques $\{i, r\}$, $\{i, q\}$, $\{j, q\}$, $\{j, r\}$, $\{q, r\}$, all with size $\eta = 2$ and therefore correspond to elements z in Z. All them also satisfy $z_{ij} = 0$ and therefore satisfy $\mu z = \mu_o$ by hypothesis on μ, μ_o. Imposing this on those five cliques it is immediate that $\mu_{ir} = \mu_{iq} = \mu_{jq} = \mu_{jr} = \mu_{qr} = \mu_o$. Given that q, r are arbitrary, it is clear that μ must satisfy that $\mu_e = \mu_o$ constant $\forall\ e \in E_n$, $e \neq (i, j)$. In other words $\mu z = \mu_o \sum_{e \neq (i,j)} z_e + \mu_{ij} z_{ij}$. This in fact means $\mu z = \mu_o$ is directly a linear combination of $z_{ij} = 0$ and $\sum_{e \in E_n} z_e = \eta(\eta - 1)/2 = 1$, for $\eta = 2$, with multiplying coefficients $\mu_{ij} - \mu_o$ and μ_o respectively. This proves the proposition for $\eta = 2$.

Now let $\eta \geq 3$. To proof they are facet defining iif $n \geq \eta + 3$ we start by discarding the only possible remaining case $n = \eta + 2$ (recall that $2 \leq \eta \leq n - 2$).
We can simply notice that the number of cliques $z \in Z$ satisfying $z_{ij} = 0$ in this case are strictly less than $|E_n| - 1$ and so there can not be enough affinely independent points to consider a facet. Recall that the cliques satisfying $z_{ij} = 0$ contain i, j or none of them. The total number of cliques containing nor i nor j is clearly th subsets of size η in the remaining $n - 2$ nodes. Similarly, the cliques containing i but not j is the binomial between $n - 2$ and $\eta - 1$ (choose $\eta - 1$ more nodes appart from i) and same for the cliques with j and not i. In total we have,

$$\binom{n-2}{\eta} + 2\binom{n-2}{\eta-1} \underset{n=\eta+2}{=} 1 + 2\eta$$

Which clearly is always strictly less than $|E_n| - 1 = \binom{n}{2} - 1 \underset{n=\eta+2}{=} \dfrac{\eta^2 + 3\eta}{2}$ $\forall \eta \geq 3$.

Now it only lacks to prove that for $n \geq \eta + 3$ and $\eta \geq 3$, $z_{ij} \geq 0$ is always facet-defining. Let us denote $L = V_n \setminus \{i, j\}$. When we use a term l_s we implicitly mean it belongs to L. Not just in this proof but in everyone from now on, when we refer to nodes l_r, l_q, l_s, l_t,... with different subindex, we implicitly mean they are different. Note that $|L| \geq \eta + 1$ given that $n \geq \eta + 3$. The feasible cliques satisfying the inequality at equality contain i and $\eta - 1$ nodes in L, j and $\eta - 1$ nodes in L or simply η nodes in L.

Let us suppose again there exists an equation $\mu z = \mu_o$ satisfied by all those cliques. Given two cliques $\mathcal{A}$, $\mathcal{B}$ satisfying $z_{ij} = 0$ both, we will note $\mathcal{A} : \mathcal{B}$ as imposing this equation on them and thus imposing

$\mu z^{\mathcal{A}} = \mu_o = \mu z^{\mathcal{B}}$. Remark that this notation will be used in further proofs.

$$\begin{aligned}\{i, l_1, \ldots, l_{\eta-1}\} :&\{j, l_1, \ldots, l_{\eta-1}\}\\ &\Rightarrow \mu_{il_1} + \ldots + \mu_{il_{\eta-1}} = \mu_{jl_1} + \ldots + \mu_{jl_{\eta-1}}\\ \{i, l_1, \ldots, l_{\eta-2}, l_\eta\} :&\{j, l_1, \ldots, l_{\eta-2}, l_\eta\}\\ &\Rightarrow \mu_{il_1} + \ldots + \mu_{il_{\eta-2}} + \mu_{il_\eta} = \mu_{jl_1} + \ldots + \mu_{jl_{\eta-2}} + \mu_{jl_\eta}\end{aligned}$$

Substracting those those equations and considering l_s arbitrary nodes in L we can state:

$$\begin{aligned}&\mu_{il_{\eta-1}} - \mu_{il_\eta} = \mu_{jl_{\eta-1}} - \mu_{jl_\eta}\\ \underset{\substack{l_{\eta-1}\ ,\ l_\eta\\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{il_r} - \mu_{il_q} = \mu_{jl_r} - \mu_{jl_q}\ \forall\ l_r, l_q \in L \qquad (i)\end{aligned}$$

Now we consider *(i)* fixing $l_r = l_{\eta-1}$ and moving $l_q = l_1, \ldots, l_{\eta-2}$.

$$\begin{aligned}\mu_{il_{\eta-1}} - \mu_{il_1} &= \mu_{jl_{\eta-1}} - \mu_{jl_1}\\ &\vdots\\ \mu_{il_{\eta-1}} - \mu_{il_{\eta-2}} &= \mu_{jl_{\eta-1}} - \mu_{jl_{\eta-2}}\end{aligned}$$

Summing up all these equations we obtain,

$$(\eta-2)\mu_{il_{\eta-1}} - \mu_{il_1} - \ldots - \mu_{il_{\eta-2}} = (\eta-2)\mu_{jl_{\eta-1}} - \mu_{jl_1} - \ldots - \mu_{jl_{\eta-2}}$$

Adding the equality provided by,

$$\{i, l_1, \ldots, l_{\eta-1}\} : \{j, l_1, \ldots, l_{\eta-1}\} \rightarrow \mu_{il_1} + \ldots + \mu_{il_{\eta-1}} = \mu_{jl_1} + \ldots + \mu_{jl_{\eta-1}}$$

It eventually results that,

$$\begin{aligned}(\eta-1)\mu_{il_{\eta-1}} = (\eta-1)\mu_{jl_{\eta-1}} &\underset{\eta\geq 3}{\Rightarrow} \mu_{il_{\eta-1}} = \mu_{jl_{\eta-1}}\\ \underset{\substack{l_{\eta-1}\\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{il_r} = \mu_{jl_r}\ \forall\ l_r \in L\end{aligned}$$

We can simply denote now $\mu_{l_r} \equiv \mu_{il_r} = \mu_{jl_r}$. Now consider,

$$\begin{aligned}\{i, l_1, \ldots, l_{\eta-1}\} :&\{i, l_1, \ldots, l_{\eta-2}, l_\eta\}\\ &\Rightarrow \mu_{l_{\eta-1}} + \mu_{l_1 l_{\eta-1}} + \ldots + \mu_{l_{\eta-2} l_{\eta-1}} = \mu_{l_\eta} + \mu_{l_1 l_\eta} + \ldots + \mu_{l_{\eta-2} l_\eta}\\ \{l_1, \ldots, l_{\eta-1}, l_{\eta+1}\} :&\{l_1, \ldots, l_{\eta-2}, l_\eta, l_{\eta+1}\}\\ &\rightarrow \mu_{l_1 l_{\eta-1}} + \ldots + \mu_{l_{\eta-2} l_{\eta-1}} + \mu_{l_{\eta-1} l_{\eta+1}} = \mu_{l_1 l_\eta} + \ldots + \mu_{l_{\eta-2} l_\eta} + \mu_{l_\eta l_{\eta+1}}\end{aligned}$$

Subtracting both equations and moving the terms in the equation we obtain,

$$\begin{aligned}&\mu_{l_{\eta-1}} - \mu_{l_\eta} = \mu_{l_{\eta-1} l_{\eta+1}} - \mu_{l_\eta l_{\eta+1}}\\ \underset{\substack{l_{\eta-1}\ ,\ l_\eta\ ,\ l_{\eta+1}\\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{l_r} - \mu_{l_q} = \mu_{l_r l_t} - \mu_{l_q l_t}\ \forall \qquad l_r, l_q, l_t \in L \qquad (ii)\end{aligned}$$

Similarly as in *(i)* now we pick *(ii)* fixing $l_r = l_{\eta-1}$ and $l_t = l_{\eta+1}$, while we move $l_q = l_1, \ldots, l_{\eta-2}$.

$$\begin{aligned}\mu_{l_{\eta-1}} - \mu_{l_1} &= \mu_{l_{\eta-1} l_{\eta+1}} - \mu_{l_1 l_{\eta+1}}\\ &\vdots\\ \mu_{l_{\eta-1}} - \mu_{l_{\eta-2}} &= \mu_{l_{\eta-1} l_{\eta+1}} - \mu_{l_{\eta-2} l_{\eta+1}}\end{aligned}$$

Summing up all these equations we obtain,

$$(\eta-2)\mu_{l_{\eta-1}} - \mu_{l_1} - \ldots - \mu_{l_{\eta-2}} = (\eta-2)\mu_{l_{\eta-1} l_{\eta+1}} - \mu_{l_1 l_{\eta+1}} - \ldots - \mu_{l_{\eta-2} l_{\eta+1}}$$

Adding then the equality provided by,

$$\{i, l_1, \ldots, l_{\eta-1}\} : \{l_1, \ldots, l_{\eta-1}, l_{\eta+1}\} \rightarrow \mu_{l_1} + \ldots + \mu_{l_{\eta-1}} = \mu_{l_1 l_{\eta+1}} + \ldots + \mu_{l_{\eta-1} l_{\eta+1}}$$

It results that,

$$(\eta-1)\mu_{l_{\eta-1}} = (\eta-1)\mu_{l_{\eta-1} l_{\eta+1}}$$

Given again the arbitrarity on $l_{\eta-1}$, $l_{\eta+1}$ taken and given that $\eta \geq 3$, we can state:

$$\mu_{l_r} = \mu_{l_r l_q} \qquad \forall\ l_r, l_q \in L \qquad (iii)$$

It is immediate to see then,

$$\{i, l_1, \ldots, l_{\eta-1}\} : \{i, l_2, \ldots, l_\eta\} \rightarrow \mu_{l_1} + \mu_{l_1 l_2} + \ldots + \mu_{l_1 l_{\eta-1}} = \mu_{l_\eta} + \mu_{l_2 l_\eta} + \ldots + \mu_{l_{\eta-1} l_\eta}$$
$$\underset{(iii)}{\Rightarrow} (\eta-1)\mu_{l_1} = (\eta-1)\mu_{l_\eta}$$
$$\underset{\substack{l_1\ ,\ l_\eta \\ \text{arbitrary}}}{\Rightarrow} \mu_{il_r} = \mu_{jl_r} \equiv \mu_{l_r} = \mu_{l_q} \equiv \mu_{il_q} = \mu_{jl_q} \qquad \forall\ l_r, l_q \in L \qquad (iv)$$

Combining *(iii)* and *(iv)* it is immediate that $\mu_e : ct$, $\forall e \in E_n \setminus (i,j)$. The rest of the proof is trivial since we can denote as $\mu_c \equiv \mu_e$, for all $e \neq (i,j)$ this constant. Clearly $\mu_o = \mu_c \eta(\eta-1)/2$ considering $\mu z = \mu_o$ for any clique with $z_{ij} = 0$. Then,

$$\mu z = \mu_c \sum_{e \neq (i,j)} z_e + \mu_{ij} z_{ij} = \mu_o = \mu_c \eta(\eta-1)/2$$

is a linear combination of $z_{ij} = 0$ and $\sum_{e \in E_n} z_e = \eta(\eta-1)/2$. □

All the other demonstrations involving facets use a similar approach. Given two cliques $\mathcal{A}$, $\mathcal{B}$ satisfying a given valid inequality at equality, we will always note $\mathcal{A} : \mathcal{B}$ as imposing $\mu z^{\mathcal{A}} = \mu_o = \mu z^{\mathcal{B}}$ just as done in the proof above.

PROPOSITION 8.5. *Given i, $j \in V_n$, $i \neq j$, the Node-to-Node Inequality for i, j is valid for Z. Furthermore, it is facet-defining if and only if $\eta \geq 3$ and $n \geq \eta + 3$.*

PROOF. Validity was proven in the section *Pricing Problem* in *Chapter 7*. Moreover, those are inequalities from the original formulation. Recalling the inequality,

$$\sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta-2) z_{ij} \geq 0$$

To check when it is facet-defining we begin by distinguishing the cliques $z \in Z$ satisfying the inequality at equality. There are clearly two cases: cliques not containing i *(a)* and cliques containing i and j *(b)*.
We start discarding the case $\eta = 2$ and $n = \eta + 2$ which are the only cases the theorem statement excludes from defining facets.

Let $\eta = 2$. In this case the *Node-to-Node Inequality* for i, j, is written simply like

$$\sum_{e \in \delta(i) \setminus (i,j)} z_e \geq 0$$

which trivially can be generated by summing the valid inequalities $z_e \geq 0\ \forall\ e \in \delta(i) \setminus (i,j)$ so it is trivially not facet-defining.

Let $n = \eta + 2 \Rightarrow \eta = n - 2$, we prove it is not facet defining by enumerating solutions with equality. The idea, already used previously, is to notice that between those there are less than $|E_n| - 1$. This immediately means that the dimension of $\{z \in \mathcal{P}_{n,\eta} \mid \sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta-2) z_{ij} = 0\}$ is less than $|E_n| - 2$ and therefore the inequality is not facet-defining.
On the one hand, we enumerate the cliques of type *(a)*. Feasible cliques not containg i are simply subsets of size $\eta = n - 2$ in $V_n \setminus \{i\}$ which has size $n - 1$. There are clearly $n - 1$ cliques of this type. On the other hand, cliques of type *(b)* are cliques containing i, j. They can be constructed by taking i, j and a subset of size $\eta - 2$ in the remaining nodes in $V_n \setminus \{i,j\}$. There are exactly $\binom{n-2}{\eta-2} = \binom{n-2}{n-4} = \binom{n-2}{2}$ cliques of this type. In total there are $n - 1 + \binom{n-2}{2}$ feasible cliques in $\{z \in Z \mid \sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta-2) z_{ij} = 0\}$. However,

$$n - 1 + \binom{n-2}{2} = \frac{n^2 - 3n + 4}{2} < \frac{n^2 - n - 2}{2} = \binom{n}{2} - 1 = |E_n| - 1 \qquad \forall\ n \geq 4$$

Lacks only to prove that for $\eta \geq 3$, $n \geq \eta + 3$ the *Node-to-Node Inequality* for i, j is facet-defining.
Let $\mu z = \mu_o$ be an equation containing $\{z \in Z \mid \sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta-2) z_{ij} = 0\}$, exactly as in previous

demonstrations. Now we can explore the characteristics on μ and μ_o. We can use cliques of type *(a)* and *(b)* and impose this equation on them as done in the previous demonstration.

Let us denote again $L = V_n \setminus \{i,j\}$, $|L| = n-2 \geq \eta+1$. Besides, let us denote l_s as an arbitrary node in L, $s = 1,\dots,\eta+1,\dots$. Again given l_r, l_q written with different subindexes, we implicitly mean they are different. We proceed with the notation $\mathcal{A} : \mathcal{B}$ meaning we are imposing $\mu z^{\mathcal{A}} = \mu z^{\mathcal{B}} = \mu_o$ for $\mathcal{A}$, $\mathcal{B}$ cliques of type *(a)*, *(b)*.

$$\begin{aligned}\{j, l_1 \dots, l_{\eta-1}\} : \{j, l_2, \dots, l_\eta\}& \\ \Rightarrow \mu_{jl_1} + \mu_{l_1 l_2} + \dots + \mu_{l_1 l_{\eta-1}} &= \mu_{jl_\eta} + \mu_{l_2 l_\eta} + \dots + \mu_{l_{\eta-1} l_\eta} \\ \{l_1 \dots, l_{\eta-1}, l_{\eta+1}\} : \{l_2, \dots, l_{\eta+1}\}& \\ \Rightarrow \mu_{l_1 l_2} + \dots + \mu_{l_1 l_{\eta-1}} + \mu_{l_1 l_{\eta+1}} &= \mu_{l_2 l_\eta} + \dots + \mu_{l_{\eta-1} l_\eta} + \mu_{l_\eta l_{\eta+1}}\end{aligned}$$

By subtracting and reordering, the result is

$$\begin{aligned}&\mu_{jl_1} - \mu_{jl_\eta} = \mu_{l_1 l_{\eta+1}} - \mu_{l_\eta l_{\eta+1}} \\ \underset{\substack{l_1,\, l_\eta,\, l_{\eta+1} \\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{jl_r} - \mu_{jl_q} = \mu_{l_r l_t} - \mu_{l_q l_t} \qquad \forall\, l_r, l_q, l_t \in L \qquad (i)\end{aligned}$$

Now we consider *(i)* fixing $l_r = l_1$ and $l_t = l_\eta$ and moving $l_q = l_2, \dots, l_{\eta-1}$.

$$\mu_{jl_1} - \mu_{jl_2} = \mu_{l_1 l_\eta} - \mu_{l_2 l_\eta}$$

$$\vdots$$

$$\mu_{jl_1} - \mu_{jl_{\eta-1}} = \mu_{l_1 l_\eta} - \mu_{l_{\eta-1} l_\eta}$$

Summing those $\eta-2$ equations, adding equation $\{j, l_1, \dots, l_{\eta-1}\} : \{l_1, \dots, l_\eta\}$ and finally dividing by $\eta-1 \geq 2$ results in,

$$\begin{aligned}&\mu_{jl_1} = \mu_{l_1 l_\eta} \\ \underset{\substack{l_1,\, l_\eta \\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{jl_r} = \mu_{l_r l_q} \qquad \forall\, l_r, l_q \in L \qquad (ii)\end{aligned}$$

Applying directly *(ii)* $\mu_{jl_r} = \mu_{l_r l_q} = \mu_{jl_q} = \mu_{l_q l_t}$ it is trivial to notice that $\mu_e :$ ct $\forall\, e \in E_n(L) \sqcup [\{j\} : L]$, i.e., $\forall\, e \notin \delta(i)$. Note that $E_n = \delta(i) \sqcup E_n(L) \sqcup [\{j\} : L]$. Let us call this constant $\mu_2 \equiv \mu_e$, $e \notin \delta(i)$.

Now let us use two cliques of type *(b)*,

$$\begin{aligned}\{i, j, l_1 \dots, l_{\eta-2}\} : \{i, j, l_2, \dots, l_{\eta-1}\}& \\ \Rightarrow \mu_{il_1} + \mu_{jl_1} + \mu_{l_1 l_2} + \dots + \mu_{l_1 l_{\eta-2}} &= \mu_{il_{\eta-1}} + \mu_{jl_{\eta-1}} + \mu_{l_2 l_{\eta-1}} + \dots + \mu_{l_{\eta-2} l_{\eta-1}} \\ \underset{\substack{\mu_e = \mu_2 \\ e \notin \delta(i)}}{\Rightarrow}\ &\mu_{il_1} + (\eta-2)\mu_2 = \mu_{il_{\eta-1}} + (\eta-2)\mu_2 \\ \underset{\substack{l_1,\, l_{\eta-1} \\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{il_r} = \mu_{il_q} \qquad \forall\, l_r, l_q \in L\end{aligned}$$

What directly means $\mu_e :$ ct $\forall\, e \in [\{i\} : L] = \delta(i) \setminus (i,j)$. Let us call this constant μ_1. Now follows finding a relation between μ_{ij} and μ_1, μ_2.

$$\begin{aligned}\{i, j, l_1 \dots, l_{\eta-2}\} : \{j, l_1, \dots, l_{\eta-1}\}& \\ \Rightarrow \mu_{ij} + \mu_{il_1} + \dots + \mu_{il_{\eta-2}} &= \mu_{jl_{\eta-1}} + \mu_{l_1 l_{\eta-1}} + \dots + \mu_{l_{\eta-2} l_{\eta-1}} \\ \Rightarrow \mu_{ij} &= (\eta-1)\mu_2 - (\eta-2)\mu_1\end{aligned}$$

We can also compute μ_o in terms of μ_1, μ_2. Computing μz for the clique $\{l_1, \dots, l_\eta\}$ of type *(a)* for example results in $\mu_o = \mu_2 \cdot \eta(\eta-1)/2$. In summary the equation $\mu z = \mu_o$ can be written as,

$$\mu_1 \sum_{e \in \delta(i) \setminus (i,j)} z_e + \mu_2 \sum_{e \notin \delta(i)} z_e + \mu_{ij} z_{ij} = \mu_1 \sum_{e \in \delta(i) \setminus (i,j)} z_e + \mu_2 \sum_{e \notin \delta(i)} z_e + [(\eta-1)\mu_2 - (\eta-2)\mu_1] z_{ij} = \mu_2 \frac{\eta(\eta-1)}{2}$$

Finally note that we can generate this equation above with a linear combination of the *Node-to-Node Inequality* for i, j at equality and the fixed size equation.

$$(\mu_1 - \mu_2)[\sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta - 2) z_{ij}] = 0$$

$$\oplus$$

$$\mu_2 \sum_{e \in E_n} z_e = \mu_2 \frac{\eta(\eta - 1)}{2}$$

□

3. New Inequalities for the Pricing Problem with fixed size

Definition: Let $n \geq 5$. Given $(i, j) \in E_n$, we define the ***Upper bound pair inequality*** for i, $j \in V_n$ as,

$$\sum_{e \in \delta(i,j)} z_e - (\eta - 3) z_{ij} \leq \eta - 1$$

Note that in case $n = 4$, the only possible η is 2. Let us note r, q the two other nodes in V_n appart from i and j. For $\eta = 2$ and $n = 4$ the fixed size equality expression is $z_{ir} + z_{iq} + z_{jr} + z_{jq} + z_{ij} + z_{rq} = 1$. On the other hand the *Upper bound pair inequality*, if considered, would look like $z_{ir} + z_{iq} + z_{jr} + z_{jq} + z_{ij} \leq 1$. Consider now the valid inequality $z_{rq} \geq 0$. Then simply note that,

$$\begin{array}{rcl} z_{ir} + z_{iq} + z_{jr} + z_{jq} + z_{ij} + z_{rq} & = & 1 \\ -z_{rq} & \leq & 0 \\ \hline z_{ir} + z_{iq} + z_{jr} + z_{jq} + z_{ij} & \leq & 1 \end{array}$$

What directly means that for $n = 4$ and the only possible $\eta = 2$, the *Upper bounded pair inequality* for i, j is in fact the inequality $z_{rq} \geq 0$. For this reason we assume $n \geq 5$ when defining the *Upper bound pair inequality*.

THEOREM 8.6. *Let $n \geq 5$. The Upper bound pair inequalities are valid for Z. They are also facet-defining for $\mathcal{P}_{n,\eta}$ if and only if $\eta \geq 3$.*

PROOF. First we prove the validity by cases. Given $z \in Z$ we must prove z satsifies the inequality. There are three different cases for the fixed size clique corresponding to z:

(a) Contains nor i nor j. Then clearly the left hand side of the inequality is $0 \leq \eta - 1$.
(b) Contains either i or j, but not both. Then the sum on the left hand side takes values $\eta - 1$ (i or j the one in the clique must be connected to $\eta - 1$ other nodes) and z_{ij} is 0. The inequality is satisfied at equality.
(c) Contains both i and j. Then the sum in the left hand side is $2(\eta - 2)$ and $z_{ij} = 1$. The inequality is therefore satisfied at equality.

The inequality is therefore valid $\forall\ z \in Z$. Note that cases *(b)*, and *(c)* are the only ones satisfying the inequality at equality.

Now let $n \geq 5$. We want to prove that the *Upper bound pair inequality* is facet-defining iif $\eta \geq 3$. Again we discard first of all the case $\eta = 2$. Similarly as in the proof of *Proposition 8.5* we discard it by proving there are strictly less than $|E_n| - 1$ different $z \in Z$ satisfying the inequality at equality.
For $\eta = 2$ the inequality is written like,

$$\sum_{e \in \delta(i,j)} z_e + z_{ij} \leq 1$$

The only solutions with equality are of type *(b)* and *(c)* as stated previously. Solutions of type *(b)* are cliques of size two with i or j. We can construct them by choosing one node appart from i and j. There are $2(n-2)$

solutions considering both cases with i or j. Solutions of type *(c)* are cliques with both i and j. There is only one single option obviously. In total we have $2(n-2)+1$ solutions. However,

$$2(n-2)+1 < |E_n| - 1 = \binom{n}{2} - 1 = \frac{n^2-n-2}{2} \quad \forall\ n \geq 5$$

Now it only lacks to prove that the *Upper bound pair inequality* is facet-defining for the general case when $n \geq 5$, $\eta \geq 3$. Let us denote $L = V_n \setminus \{i,j\}$. When we use a term l_s we implicitly mean it belongs to L again and when referring to terms l_r, l_q,... we implicitly mean they are different by writing a different subindex. Note that $|L| \geq \eta$ given that $n \geq \eta + 2$. Let us suppose there exists an equation $\mu z = \mu_o$ satisfied by all $z \in Z$ with $\sum_{e \in \delta(i,j)} z_e - (\eta-3) z_{ij} = \eta - 1$. We can consider cliques of type *(b)* and *(c)* again to explore conditions on μ and μ_o.

$$\begin{aligned} \{i, l_1, \ldots, l_{\eta-1}\} : &\{j, l_1, \ldots, l_{\eta-1}\} \\ &\Rightarrow \mu_{il_1} + \ldots + \mu_{il_{\eta-1}} = \mu_{jl_1} + \ldots + \mu_{jl_{\eta-1}} \\ \{i, l_1, \ldots, l_{\eta-2}, l_\eta\} : &\{j, l_1, \ldots, l_{\eta-2}, l_\eta\} \\ &\Rightarrow \mu_{il_1} + \ldots + \mu_{il_{\eta-2}} + \mu_{il_\eta} = \mu_{jl_1} + \ldots + \mu_{jl_{\eta-2}} + \mu_{jl_\eta} \end{aligned}$$

Subtracting those two equations and noting l_η, $l_{\eta-1}$ are arbitrary nodes in L we can state:

$$\mu_{il_r} - \mu_{il_q} = \mu_{jl_r} - \mu_{jl_q} \qquad \forall\ l_r, l_q \in L \qquad (i)$$

Now we apply *(i)* fixing $l_r = l_{\eta-1}$ and moving $l_q = l_1, \ldots, l_{\eta-2}$.

$$\begin{gathered} \mu_{il_{\eta-1}} - \mu_{il_1} = \mu_{jl_{\eta-1}} - \mu_{jl_1} \\ \vdots \\ \mu_{il_{\eta-1}} - \mu_{il_{\eta-2}} = \mu_{jl_{\eta-1}} - \mu_{jl_{\eta-2}} \end{gathered}$$

Summing up all these equations we obtain,

$$(\eta-2)\mu_{il_{\eta-1}} - \mu_{il_1} - \ldots - \mu_{il_{\eta-2}} = (\eta-2)\mu_{jl_{\eta-1}} - \mu_{jl_1} - \ldots - \mu_{jl_{\eta-2}}$$

Adding the equality given by $\{i, l_1, \ldots, l_{\eta-1}\} : \{j, l_1, \ldots, l_{\eta-1}\}$ it directly results that,

$$\begin{gathered} (\eta-1)\mu_{il_{\eta-1}} = (\eta-1)\mu_{jl_{\eta-1}} \\ \underset{\eta \geq 3}{\Rightarrow} \mu_{il_r} = \mu_{jl_r} \qquad \forall\ l_r \in L \end{gathered}$$

We can denote therefore $\mu_{l_r} \equiv \mu_{il_r} = \mu_{jl_r}$.
Now let us consider,

$$\begin{aligned} \{i, l_1, \ldots, l_{\eta-1}\} : &\{i, l_2, \ldots, l_\eta\} \\ &\Rightarrow \mu_{l_1} + \mu_{l_1 l_2} + \ldots + \mu_{l_1 l_{\eta-1}} = \mu_{l_\eta} + \mu_{l_2 l_\eta} + \ldots + \mu_{l_{\eta-1} l_\eta} \\ \{i, j, l_1, l_3, \ldots, l_{\eta-1}\} : &\{i, j, l_3, \ldots, l_\eta\} \\ &\Rightarrow 2\mu_{l_1} + \mu_{l_1 l_3} + \ldots + \mu_{l_1 l_{\eta-1}} = 2\mu_{l_\eta} + \mu_{l_3 l_\eta} + \ldots + \mu_{l_{\eta-1} l_\eta} \end{aligned}$$

Subtracting, we obtain that,

$$\begin{gathered} \mu_{l_1} - \mu_{l_1 l_2} = \mu_{l_\eta} - \mu_{l_2 l_\eta} \\ \underset{\substack{l_1\ ,\ l_2\ ,\ l_\eta \\ \text{arbitrary}}}{\Rightarrow} \mu_{l_r} - \mu_{l_r l_q} = \mu_{l_t} - \mu_{l_q l_t} \qquad \forall\ l_r, l_q, l_t \in L \qquad (ii) \end{gathered}$$

Again we consider then the following $\eta - 2$ equations *(ii)* as done before:

$$\begin{gathered} \mu_{l_1} - \mu_{l_1 l_2} = \mu_{l_\eta} - \mu_{l_2 l_\eta} \\ \vdots \\ \mu_{l_1} - \mu_{l_1 l_{\eta-1}} = \mu_{l_\eta} - \mu_{l_{\eta-1} l_\eta} \end{gathered}$$

Adding all them results in

$$(\eta-2)\mu_{l_1} - \mu_{l_1 l_2} - \ldots - \mu_{l_1 l_{\eta-1}} = (\eta-2)\mu_{l_\eta} - \mu_{l_2 l_\eta} - \ldots - \mu_{l_{\eta-1} l_\eta}$$

Adding then the equation provided by $\{i, l_1, \dots, l_{\eta-1}\} : \{i, l_2, \dots, l_\eta\}$ described above, it results

$$\begin{aligned} &(\eta-1)\mu_{l_1} = (\eta-1)\mu_{l_\eta} \\ \underset{\substack{l_1\ ,\ l_\eta \\ \text{arbitrary}}}{\Rightarrow} \quad &\mu_{l_r} = \mu_{l_q} \qquad \forall\ l_r, l_q \in L \\ \Rightarrow\ &\mu_e :\ \text{ct} \qquad \forall\ e \in \delta(i,j) \end{aligned}$$

We can denote this constant by $\mu_1 \equiv \mu_e$, $e \in \delta(i,j)$. By imposing this in *(ii)* we immediately see that $\mu_{l_r l_q} = \mu_{l_q l_t}\ \forall\ l_r, l_q, l_t \in L$. In case L has size 3 it immediately means $\mu_e : \text{ct}\ \forall\ e \in E_n(L)$. If L has size at least 4 we can consider also l_p and $\mu_{l_r l_q} = \mu_{l_q l_t} = \mu_{l_p l_t}$ and therefore $\mu_e : \text{ct}\ \forall\ e \in E_n(L)$ too. We denote as μ_2 this constant.

In addition, we can write μ_{ij} in terms of constants μ_1, μ_2. Simply imposing $\{i, j, l_1, \dots, l_{\eta-2}\}$:$\{j, l_1, \dots, l_{\eta-1}\}$ and substituting the corresponding terms by the constants μ_1, μ_2, it results that:

$$\mu_{ij} + (\eta-2)\mu_1 = \mu_1 + (\eta-2)\mu_2$$

This allows us to construct $\mu z = \mu_o$ as a linear combination of the *Upper bound pair inequality* at equality and the fixed size equation.
Note first we can express μz in terms of μ_1 and μ_2 like,

$$\mu z = \mu_1 \sum_{e\in\delta(i,j)} z_e + \mu_2 \sum_{e\in E_n(L)} z_e + \mu_{ij} z_{ij} = \mu_1 \sum_{e\in\delta(i,j)} z_e + \mu_2 \sum_{e\in E_n(L)} z_e + [(\eta-2)\mu_2 - (\eta-3)\mu_1] z_{ij} \qquad (*)$$

Note that $E_n = \delta(i,j) \sqcup E_n(L) \sqcup (i,j)$.
Note that μ_o can be computed in terms of μ_1, μ_2 too. Simply consider μz for the clique $\{i, l_1, \dots, l_{\eta-1}\}$ for example. In this clique there are $\eta - 1$ nodes in L and node i so $\mu z = (\eta-1)\mu_1 + \mu_2(\eta-1)(\eta-2)/2 = \mu_o$.
Then consider the following multiples of the *Upper bound pair inequality* at equality and the fixed clique size equation in terms of μ_1, μ_2:

$$(\mu_1 - \mu_2)[\sum_{e\in\delta(i,j)} z_e - (\eta-3) z_{ij}] = (\mu_1 - \mu_2)(\eta-1)$$

$$\mu_2 \sum_{e\in E_n} z_e = \mu_2 \frac{\eta(\eta-1)}{2}$$

It is trivial to see that adding both results in $\mu z = \mu_o$ written in terms of μ_1, μ_2 (*). □

Definition: Given $W \subset V_n$ with $|W| \le \lfloor n/2 \rfloor$. Let w be the size of W. Discarding the case $w = 2$ and $\eta = 3$, we define the ***δ-Set inequality*** for W as,

$$(i) \qquad \sum_{e\in\delta(W)} z_e \le w(\eta - w) \qquad \text{if } w \le \lfloor \eta/2 \rfloor$$

$$(ii) \qquad \sum_{e\in\delta(W)} z_e \le \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil \qquad \text{if } w > \lfloor \eta/2 \rfloor$$

Note that the definition includes fixing $|W| \le \lfloor n/2 \rfloor$. In case we considered a set W with $|W| > \lfloor n/2 \rfloor$, note that its *δ-Set inequality* would be the same provided by the set $W^c = V_n \setminus W$. The set W^c clearly satisfies $|W^c| \le \lfloor n/2 \rfloor$. On account of that, we only have to consider the cases $|W| \le \lfloor n/2 \rfloor$.
Note, moreover, that the case $w = 2$ and $\eta = 3$ corresponds to the *Upper bound pair inequality* with i, j the two nodes in W. We already know it is valid and facet-defining but we can discard it as an example of another valid inequality.

PROPOSITION 8.7. *Given $W \subset V_n$, $|W| \le \lfloor n/2 \rfloor$, the δ - Set inequality for W is valid for Z.*

PROOF. Given $z \in Z$ corresponding to a feasible clique in V_n with size η. Let γ be the number of nodes in this clique belonging to W, $\gamma \le w$, $\gamma \le \eta$. On account of that, clearly,

$$\sum_{e\in\delta(W)} z_e = \gamma(\eta - \gamma)$$

We would like to find an upper bound on $\gamma(\eta-\gamma)$. Let us consider $f(x)=x(\eta-x)$ its extended continuous function. It is a quadratic polynomial with vertex in its maximum at $x=\eta/2$. Considering only $x\in[0,\eta]$ the function is simply a non negative bell symmetric with respect to $x=\eta/2$, where it takes its maximum value. In fact, for $x\in[0,\eta/2]$ it is strictly increasing and for $x\in[\eta/2,\eta]$ its strictly decreasing given the symmetry.
Coming back to the original expression $\gamma(\eta-\gamma)$ it corresponds to taking the integer values in this bell for $x\in[0,\eta]$.

When $w\leq\lfloor\eta/2\rfloor$ all possible γ belong to $[0,\lfloor\eta/2\rfloor]\subseteq[0,\eta/2]$ and so $\gamma(\eta-\gamma)$ is strictly increasing in γ. Its maximum is then clearly taking $\gamma=w$. This proves the case *(i)* in the definition of the *δ-Set inequality*.

When $w>\lfloor\eta/2\rfloor$ we can solve by distinguishing the case η even an odd. When η is even the right hand side of is clearly the maximum on $f(x)$ and therefore on $\gamma(\eta-\gamma)$. It is therefore immediately valid for η even. When η is odd we only have to notice that the maximum value for $f(\gamma)$ in the integer values $\gamma\in[0,\eta]$ are the ones for $\gamma=\lfloor\eta/2\rfloor$ and its symmetric $\gamma=\lceil\eta/2\rceil$. In any of those two cases the value of $\gamma(\eta-\gamma)$ is the right hand side of the inequality. Consequently, *(ii)* is valid for η odd two. □

THEOREM 8.8. *Given $W\subset V_n$, $w=|W|\leq\lfloor n/2\rfloor$, the δ - Set inequality for W, if defined, is facet-defining if and only if $w\geq 3$, $\eta\in\{3,\dots,2w-3\}$, η odd.*

PROOF. We will proof the theorem by proving 5 claims in the following order:

(1) It is never facet-defining for $w=1$.
(2) For $w=2$, it is facet-defining only for $\eta=3$, and this corresponds in fact to the *Upper bound pair inequality* so the *δ-Set inequality* is not defined.
(3) For $w\geq 3$, it is never facet-defining for $\eta=2$ nor $\eta\geq 2(w-1)$.
(4) For $w\geq 3$, it is never facet-defining for η even.
(5) For $w\geq 3$, it is facet-defining for $\eta\in\{3,\dots,2w-3\}$, η odd.

When looking for facet-defining conditions, Claims (1)-(2) force considering only cases with $w\geq 3$. Claims (3)-(4) force considering only $\eta\in\{3,\dots,2w-3\}$ with η even. Claim (5) finishes the proof.

Proof of Claim 1: Let $w=1$, $W=\{v\}$ a node in V_n. Given that $2\leq\eta\leq n-2$ then $w\leq\lfloor\eta/2\rfloor$ so we can always consider the *δ-Set inequality* as

$$\sum_{e\in\delta(v)} z_e\leq\eta-1$$

If we consider the feasible cliques satisfying the inequality at equality, $\{z\in Z\mid\sum_{e\in\delta(v)}z_e=\eta-1\}$, we clearly note they are simply feasible cliques containing v. Given that $n\geq 4$ we can consider another arbitrary node $q\neq v$. Let us consider the equation

$$-(\eta-2)z_{vq}+\sum_{e\in\delta(q)\setminus(v,q)} z_e=0$$

This equality clearly contains all cliques with size η and containing v, i.e., contains $\{z\in Z\mid\sum_{e\in\delta(v)}z_e=\eta-1\}$.
Moreover the new equation above is linearly independent from the *δ-Set inequality* at equality and the fixed size equation. To see that simply we can find a feasible $z\in Z$ tight for $-(\eta-2)z_{vq}+\sum_{e\in\delta(q)\setminus(v,q)}z_e=0$ but not for $\sum_{e\in\delta(v)}z_e=\eta-1$. Simply let $L=V_n\setminus\{v,q\}$. Given that $n\geq\eta+2$ then clearly $|L|\geq\eta$ and we can pick η nodes in L forming a clique with them. Clearly this clique is not tight for $\sum_{e\in\delta(v)}z_e=\eta-1$ as it does not contain v, although it is tight for $-(\eta-2)z_{vq}+\sum_{e\in\delta(q)\setminus(v,q)}z_e=0$ as contains nor v nor q. ■

Proof of Claim 2: Let $w=2$, $W=\{i,j\}$. For $\eta=3$ we already know this is the *Upper bound pair inequality* and we already know it is facet-defining. Lacks only to discard it is facet-defining for $2\leq\eta\leq n-2$, $\eta\neq 3$. We proceed by cases.
For $\eta=2$ the *δ-Set inequality* for W is written like $\sum_{e\in\delta(i,j)}z_e\leq 1$. Trivially it is not facet-defining given that we can generate it considering the fixed size equation $\sum_{e\in E_n}z_e=1$ for $\eta=2$ and add it the valid

inequalities $-z_e \leq 0 \; \forall \; e \notin \delta(i,j)$.
For $\eta \geq 4$ clearly $w = 2 \leq \lfloor \eta/2 \rfloor$ and therefore the *δ-Set inequality* is always written like

$$\sum_{e \in \delta(i,j)} z_e \leq 2(\eta - 2)$$

To satisfy it at equality the clique solutions must contain both i and j. This means all $\{z \in Z \mid \sum_{e \in \delta(i,j)} z_e = 2(\eta-2)\}$ also satisfies $z_{ij} = 1$. It directly constructs then an equation containing $\{z \in Z \mid \sum_{e \in \delta(i,j)} z_e = 2(\eta-2)\}$. To see this equation is linearly independent to the *δ-Set inequality* at equality and the fixed size equation we simply construct a general linear combination of these two last equations with coefficients λ, γ respectively.

$$(\lambda + \gamma) \sum_{e \in \delta(i,j)} z_e + \gamma \sum_{e \in E_n(W) \sqcup E_n(V_n \setminus W)} z_e = \lambda 2(\eta - 2) + \gamma \frac{\eta(\eta-1)}{2}$$

Imposing it to be equal to $z_{ij} = 1$ leaves no solution for λ, γ. Note that $E_n = \delta(W) \sqcup E_n(W) \sqcup E_n(V_n \setminus W)$. ■

Proof of Claim 3: We prove by cases demonstrating that it is not facet-defining.
For $\eta = 2$, proceed as in *Claim 2* given that the inequality is yet written as $\sum_{e \in \delta(W)} z_e \leq 1$.
For $\eta \geq 2w \Rightarrow w \leq \lfloor \eta/2 \rfloor \Rightarrow \sum_{e \in \delta(W)} z_e \leq w(\eta - w)$.
The equality takes place only for cliques $z \in Z$ containing all W so clearly they do also satisfy

$$\sum_{e \in E_n(W)} z_e = \frac{w(w-1)}{2}$$

This is a new equation containing all $\{z \in Z \mid \sum_{e \in \delta(W)} z_e = w(\eta - w)\}$. To see the independence recall the procedure in *Claim 2* to write any linear combination of the *δ-Set inequality* at equality and the fixed size equation,

$$(\lambda + \gamma) \sum_{e \in \delta(W)} z_e + \gamma \sum_{e \in E_n(W) \sqcup E_n(V_n \setminus W)} z_e = \lambda w(\eta - w) + \gamma \frac{\eta(\eta-1)}{2}$$

Clearly there is no solution λ, γ when imposing it to be equal to $\sum_{e \in E_n(W)} z_e = w(w-1)/2$.
For $\eta = 2w - 1 \Rightarrow \eta$ odd, $w = \lceil \eta/2 \rceil \Rightarrow \sum_{e \in \delta(W)} z_e \leq \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil = (\eta - w)w$.
In this case the equality is achieved by cliques containg all w nodes from W *(a)* or containing $w-1$ nodes from W *(b)*. Let $s = w$, $s \geq 3$, and let $W = \{w_1, \ldots, w_s\}$. Consider then the equation

$$-(s-2)z_{w_1 w_2} - (s-2)z_{w_1 w_3} + \sum_{j=4}^{s} z_{w_1 w_j} + \sum_{r \notin W} z_{w_1 r} + (s-1)z_{w_2 w_3} = s - 1 \qquad (*)$$

Note that the term $\sum_{j=4}^{s} z_{w_1 w_j}$ only exists when $s \geq 4$.
See that *(*)* is independent exactly as in *Claims 2, 3*. Lacks only to demonstrate that *(*)* is tight for cliques *(a)* and *(b)* defined above. We proceed by cases:

(a) Consider taking all W in the clique solution. Then the left hand side in *(*)* is $-(s-2)-(s-2)+(s-3)+(\eta-s)+(s-1) = s-1$ and then equation *(*)* is tight. Recall that $\eta = 2s-1$.
(b) Consider taking $w-1$ nodes from W in the solution.
  (I) Suppose w_1 is not between those nodes. Then the left hand side in *(*)* is $-0-0+0+0+s-1 = s-1$ and equation and *(*)* is tight.
  (II) Suppose w_1 is between those nodes. Given that there are $w-1$ nodes in the clique from W, at least w_2 or w_3 one of them must be included.
    (i) We include w_2 or w_3 but not both. The left hand side in *(*)* is $-(s-2)+(s-3)+(\eta-(s-1)) = s-1$ given that $\eta = 2s-1$. Equation *(*)* is tight.
    (ii) Both w_2 and w_3 are included. It means necessarily that $s \geq 4$ because there are $w-1$ nodes from W in the clique solution and in this case we know we include at least w_1, w_2 and w_3. The left hand side in equation *(*)* again is $-(s-2)-(s-2)+(s-4)+(\eta-(s-1))+(s-1) = s-1$ and therefore equation *(*)* is tight too.

In summary, all $\{z \in Z \mid \sum_{e\in\delta(W)} z_e = \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil = (\eta - w)w\}$ for $\eta = 2w - 1$ are tight for equation *(*)*.
For $\eta = 2w - 2 \Rightarrow \eta$ pair, $w = 1 + \eta/2 \Rightarrow \sum_{e\in\delta(W)} z_e \leq \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil = (w-1)^2$.
In this case the equality takes place only for feasible cliques containing exactly $w - 1$ nodes from W. This means all those cliques satisfy necessarily

$$\sum_{e\in E_n(W)} z_e = \frac{(w-1)(w-2)}{2}$$

and the rest is analogous to the case $\eta \geq 2w$. ■

Proof of Claim 4: Let $\eta \in \{3, \ldots, 2w-3\}$, $\eta = 2p$, $p \geq 2$, $w \geq 3$. Given that $\eta \leq 2w - 3$ it is clear that $w > \lfloor \eta/2 \rfloor$ and therefore the inequality is written as

$$\sum_{e\in\delta(W)} z_e \leq \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil = p^2$$

Considering the inequality at equality it is clear that the feasible cliques satisfying this equation contain exactly p elements from W. This means they do satisfy $\sum_{e\in E_n(W)} z_e = p(p-1)/2$ and the rest is analogous to *Claim 3*, the case $\eta \geq 2w$. ■

Proof of Claim 5: Let $\eta \in \{3, \ldots, 2w-3\}$, $\eta = 2p+1$ odd, $p \geq 1$. In this case $w > \lfloor \eta/2 \rfloor$ and so the *δ-Set inequality* for W is always written like,

$$\sum_{e\in\delta(W)} z_e \leq \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil = p(p+1)$$

Note also that $\eta \leq 2w-3 \Rightarrow w \geq (\eta+3)/2 = p+2$. Let us denote $L = V_n \backslash W$. Note that $|L| = n - w$ and recall that $n \geq 2w$. Immediately $|L| \geq w \geq p+2$. Let us denote $W = \{w_1, \ldots, w_{p+2}, \ldots\}$ and $L = \{l_1, \ldots, l_{p+2}, \ldots\}$. Note that the feasible cliques $z \in Z$ satsifying the inequality at equality $\sum_{e\in\delta(W)} z_e = p(p+1)$ are cliques containing p elements from W and $p+1$ elements from L or $p+1$ elements from W and p from L. As usual, we suppose there exists an equation $\mu z = \mu_o$ satisfied by all those cliques and look for properties on μ, μ_o. We follow the notation $\mathcal{A} : \mathcal{B}$ to impose $\mu z^{\mathcal{A}} = \mu_o = \mu z^{\mathcal{B}}$ for two cliques $\mathcal{A}$, $\mathcal{B}$ of size η satisfying $\sum_{e\in\delta(W)} z_e = p(p+1)$, i.e., containing p or $p+1$ elements of type w_s and the rest of type l_s. With all this taken into consideration we proceed.

$$\begin{aligned}
&\{w_1, \ldots, w_p,\, l_1, \ldots, l_{p+1}\} : \{w_2, \ldots, w_p, w_{p+2}, l_1, \ldots, l_{p+1}\} \\
&\qquad \Rightarrow \mu_{w_1w_2} + \ldots + \mu_{w_1w_p} + \mu_{w_1l_1} + \ldots + \mu_{w_1l_{p+1}} = \mu_{w_2w_{p+2}} + \ldots + \mu_{w_pw_{p+2}} + \mu_{w_{p+2}l_1} + \ldots + \mu_{w_{p+2}l_{p+1}} \\
&\{w_1, \ldots, w_{p+1},\, l_1, \ldots, l_p\} : \{w_2, \ldots, w_{p+2}, l_1, \ldots, l_p\} \\
&\qquad \Rightarrow \mu_{w_1w_2} + \ldots + \mu_{w_1w_{p+1}} + \mu_{w_1l_1} + \ldots + \mu_{w_1l_p} = \mu_{w_2w_{p+2}} + \ldots + \mu_{w_{p+1}w_{p+2}} + \mu_{w_{p+2}l_1} + \ldots + \mu_{w_{p+2}l_p}
\end{aligned}$$

Subtracting both equations we obtain,

$$\begin{aligned}
&\mu_{w_1l_{p+1}} - \mu_{w_1w_{p+1}} = \mu_{w_{p+2}l_{p+1}} - \mu_{w_{p+1}w_{p+2}} \\
\underset{\substack{w_1,\ w_{p+1},\ w_{p+2},\ l_{p+1} \\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{w_rl_s} - \mu_{w_rw_q} = \mu_{w_tl_s} - \mu_{w_qw_t} \qquad \forall\ w_r, w_q, w_t \in W,\, l_s \in L \qquad (i)
\end{aligned}$$

On the other hand and in parallel,

$$\begin{aligned}
&\{w_1, \ldots, w_p,\, l_1, \ldots, l_{p+1}\} : \{w_1, \ldots, w_p, l_2, \ldots, l_{p+2}\} \\
&\qquad \Rightarrow \mu_{w_1l_1} + \ldots + \mu_{w_pl_1} + \mu_{l_1l_2} + \ldots + \mu_{l_1l_{p+1}} = \mu_{w_1l_{p+2}} + \ldots + \mu_{w_pl_{p+2}} + \mu_{l_2l_{p+2}} + \ldots + \mu_{l_{p+1}l_{p+2}} \\
&\{w_1, \ldots, w_{p+1},\, l_1, \ldots, l_p\} : \{w_1, \ldots, w_{p+1}, l_2, \ldots, l_p, l_{p+2}\} \\
&\qquad \Rightarrow \mu_{w_1l_1} + \ldots + \mu_{w_{p+1}l_1} + \mu_{l_1l_2} + \ldots + \mu_{l_1l_p} = \mu_{w_1l_{p+2}} + \ldots + \mu_{w_{p+1}l_{p+2}} + \mu_{l_2l_{p+2}} + \ldots + \mu_{l_pl_{p+2}}
\end{aligned}$$

Subtracting again both equations,

$$\begin{aligned}
&\mu_{l_1l_{p+1}} - \mu_{w_{p+1}l_1} = \mu_{l_{p+1}l_{p+2}} - \mu_{w_{p+1}l_{p+2}} \\
\underset{\substack{l_1,\ l_{p+1},\ l_{p+2},\ w_{p+1} \\ \text{arbitrary}}}{\Rightarrow}\ &\mu_{l_rl_q} - \mu_{w_sl_r} = \mu_{l_ql_t} - \mu_{w_sl_t} \qquad \forall\ l_r, l_q, l_t \in L,\, w_s \in W \qquad (ii)
\end{aligned}$$

Taking *(ii)* we fix $l_r = l_1$, $l_t = l_{p+2}$ and move $l_q = l_2, \ldots, l_{p+1}$ and $w_s = w_1, \ldots, w_p$ simultaneously,

$$\mu_{l_1 l_2} - \mu_{w_1 l_1} = \mu_{l_2 l_{p+2}} - \mu_{w_1 l_{p+2}}$$
$$\vdots$$
$$\mu_{l_1 l_{p+1}} - \mu_{w_p l_1} = \mu_{l_{p+1} l_{p+2}} - \mu_{w_p l_{p+2}}$$

We sum all these equations and finally add the equation provided by $\{w_1, \ldots, w_p, l_1, \ldots, l_{p+1}\} : \{w_1, \ldots, w_p, l_2, \ldots, l_{p+2}\}$ described above. The result after dividing all by 2 is

$$\mu_{l_1 l_2} + \ldots + \mu_{l_1 l_{p+1}} = \mu_{l_2 l_{p+2}} + \ldots + \mu_{l_{p+1} l_{p+2}} \qquad (*)$$

Besides now consider *(i)* fixing $w_r = w_1$ and $w_t = w_{p+2}$ and moving $w_q = w_2, \ldots, w_{p+1}$ and $l_s = l_1, \ldots, l_p$ simultaneously. Adding those equations and also adding the equality provided by $\{w_1, \ldots, w_{p+1}, l_1, \ldots, l_p\} : \{w_2, \ldots, w_{p+2}, l_1, \ldots, l_p\}$ above and dividing by 2 (analogously as done to get (*)) the result is,

$$\mu_{w_1 l_1} + \ldots + \mu_{w_1 l_p} = \mu_{w_{p+2} l_1} + \ldots + \mu_{w_{p+2} l_p} \qquad (**)$$

Note that we consider (**) with different sets of nodes $l_1, \ldots, l_p \in L$ and two main nodes $w_1, w_{p+2} \in W$. Thus we can consider those two equations of type (**) fixing w_1, w_{p+2} but changing the nodes in L.

$$\mu_{w_1 l_1} + \ldots + \mu_{w_1 l_p} = \mu_{w_{p+2} l_1} + \ldots + \mu_{w_{p+2} l_p}$$
$$\mu_{w_1 l_2} + \ldots + \mu_{w_1 l_{p+1}} = \mu_{w_{p+2} l_2} + \ldots + \mu_{w_{p+2} l_{p+1}}$$

Taking the difference results in,

$$\mu_{w_1 l_1} - \mu_{w_1 l_{p+1}} = \mu_{w_{p+2} l_1} - \mu_{w_{p+2} l_{p+1}}$$
$$\underset{\substack{w_1,\ w_{p+2},\ l_1,\ l_{p+1} \\ \text{arbitrary}}}{\Rightarrow} \mu_{w_r l_q} - \mu_{w_r l_s} = \mu_{w_t l_q} - \mu_{w_t l_s} \qquad \forall\ w_r, w_t \in W,\ l_q, l_s \in L \qquad (iii)$$

Now considering *(iii)* above fixing $l_q = l_1$, $w_r = w_1$, $w_t = w_{p+2}$ and moving $l_s = l_2, \ldots, l_p$ and adding all those equations results in,

$$(p-1)\mu_{w_1 l_1} - \mu_{w_1 l_2} - \ldots - \mu_{w_1 l_p} = (p-1)\mu_{w_{p+2} l_1} - \mu_{w_{p+2} l_2} - \ldots - \mu_{w_{p+2} l_p}$$

If add directly (**), divide by $p \geq 1$ and consider the arbitrarity on w_1, $w_{p+2} \in W$ and $l_1 \in L$ we note that,

$$\mu_{w_r l_s} = \mu_{w_q l_s} \qquad \forall\ w_r, w_q \in W,\ l_s \in L \qquad (iv)$$

Now consider the equations,

$$\mu_{w_1 l_{p+1}} - \mu_{w_1 w_{p+1}} = \mu_{w_{p+2} l_{p+1}} - \mu_{w_{p+1} w_{p+2}} \qquad (i)$$
$$\mu_{w_1 l_{p+1}} = \mu_{w_{p+2} l_{p+1}} \qquad (iv)$$

By taking its difference and the arbitrarity on the nodes the result is the statement

$$\mu_{w_r w_q} = \mu_{w_q w_s} \qquad \forall\ w_r, w_q, w_s \in W \qquad (v)$$

Using *(v)* it is trivial to see that μ_e : ct $\forall\ e \in E_n(W)$. Let us denote it as μ_w.
Now take the equation provided by $\{w_1, \ldots, w_p, l_1, \ldots, l_{p+1}\} : \{w_1, \ldots, w_p, l_2, \ldots, l_{p+2}\}$ defined above and subtract (*) to it directly. The result is,

$$\mu_{w_1 l_1} + \ldots + \mu_{w_p l_1} = \mu_{w_1 l_{p+2}} + \ldots + \mu_{w_p l_{p+2}}$$
$$\underset{(iv)}{\Rightarrow} p \cdot \mu_{w_1 l_1} = p \cdot \mu_{w_p l_{p+2}}$$

Dividing by $p \geq 1$ and noting again that w_1, $w_p \in W$ and l_1, $l_{p+2} \in L$ are arbitrary as well as considering *(iv)*, it is trivial to see that μ_e : ct $\forall\ e \in \delta(W)$. Let us denote it as $\mu_{\delta(w)}$.

Now let us consider the equation provided by $\{w_1, \ldots, w_p, l_1, \ldots, l_{p+1}\}$:$\{w_1, \ldots, w_{p+1}, l_1, \ldots, l_p\}$,

$$\mu_{w_1 l_{p+1}} + \ldots + \mu_{w_p l_{p+1}} + \mu_{l_1 l_{p+1}} + \ldots + \mu_{l_p l_{p+1}} = \mu_{w_1 w_{p+1}} + \ldots + \mu_{w_p w_{p+1}} + \mu_{w_{p+1} l_1} + \ldots + \mu_{w_{p+1} l_p}$$

Introducing the terms μ_w, $\mu_{\delta(w)}$ it results in,

$$\mu_{l_1 l_{p+1}} + \ldots + \mu_{l_p l_{p+1}} = p \cdot \mu_w$$

Analogously, we can change the role of $l_1, \ldots, l_p$ by $l_2, \ldots, l_p, l_{p+2}$ and obtain,

$$\mu_{l_2 l_{p+1}} + \ldots + \mu_{l_p l_{p+1}} + \mu_{l_{p+1} l_{p+2}} = p \cdot \mu_w$$

By subtracting both and using the arbitrariness again, we note that $\mu_{l_r l_q} = \mu_{l_q l_s}\ \forall\ l_r, l_q, l_s \in L$. It is trivial to see with it that $\mu_e : \text{ct}\ \forall\ e \in E_n(L)$. Let us denote this constant as $\mu_{E_n(L)}$. Moreover the equation above can be rewritten as $p \cdot \mu_{E_n(L)} = p \cdot \mu_w$ so that immediately $\mu_{E_n(L)} = \mu_w$.

In summary we have eventually noticed that $\mu_e = \mu_{\delta(w)}$ constant $\forall\ e \in \delta(W)$ and that $\mu_e = \mu_w$ constant for all $e \in E_n(W) \sqcup E_n(L)$, i.e., $e \notin \delta(W)$. Note that $E_n = E_n(W) \sqcup E_n(L) \sqcup \delta(W)$.

With this, we can compute μ_o in terms of μ_w, $\mu_{\delta(w)}$ simply considering the solution $\{w_1, \ldots, w_p, l_1, \ldots, l_{p+1}\}$. The result after simple computations is $\mu_o = p(p+1)\mu_{\delta(w)} + p^2\mu_w$. Consequently and taking everything into account we can write the equation $\mu z = \mu_o$ like,

$$\mu_{\delta(w)} \sum_{e \in \delta(W)} z_e + \mu_w \sum_{e \in E_n(W) \cup E_n(L)} z_e = p(p+1)\mu_{\delta(w)} + p^2\mu_w$$

Now we consider the linear combination of the *δ-Set inequality* for W at equality and the fixed size equation for Z, with coefficients λ, γ respectively,

$$(\lambda + \gamma) \sum_{e \in \delta(W)} z_e + \gamma \sum_{e \in E_n(W) \cup E_n(L)} z_e = \lambda p(p+1) + \gamma \frac{\eta(\eta-1)}{2} = \lambda p(p+1) + \gamma p(2p+1)$$

If we set $\lambda = \mu_{\delta(w)} - \mu_w$ and $\gamma = \mu_w$ we obtain the expression of $\mu z = \mu_o$ in terms of μ_w and $\mu_{\delta(w)}$. Consequently the equation $\mu z = \mu_o$ is a linear combination of the *δ-Set inequality* for W at equality and the fixed size equation. This means the *δ-Set inequality* for W is facet-defining. ■ □

Finally, we have also noticed an extension of the *Node-to-Node Inequalities* considering sets of nodes A with size at least 2. We naturally call them *Node-to-Set Inequalities* to emphasize they are an extension of the *Node-to-Node Inequalities*.

Definition: Given $i \in V_n$ and $A \subseteq V_n \setminus \{i\}$, $|A| \geq 2$, we define the ***Node-to-Set Inequality*** for i, A as,

$$\sum_{j \in V \setminus A} z_{ij} + (\eta - 1) \sum_{e \in E_n(A)} z_e - (\eta - 2) \sum_{j \in A} z_{ij} \geq 0$$

Note that if we considered a subset $A \subset V_n \setminus \{i\}$ of size 1 it would be the *Node-to-Node Inequality* for i and j the unique node in A.
We can rewrite the *Node-to-Set Inequality* for i and A in another way that might be more useful afterwards. Let $L = V_n \setminus (A \cup \{i\})$. Then the *Node-to-Set Inequality* for i and A becomes

$$\sum_{e \in [\{i\}:L]} z_e + (\eta - 1) \sum_{e \in E_n(A)} z_e - (\eta - 2) \sum_{e \in [\{i\}:A]} z_e \geq 0$$

PROPOSITION 8.9. *Given $i \in V_n$ and $A \subseteq V_n \setminus \{i\}$, $|A| \geq 2$, the Node-to-Set Inequality for i, A is valid for Z.*

PROOF. Let $z \in Z$ be a feasible clique. This clique has size η and might or might not contain node i. In case i is not in the clique clearly the left hand side of the inequality only contains terms of positive coefficient so the inequality is immediately satisfied.
Suppose now that i is contained in this clique. Let q be an integer corresponding to the number of nodes in this clique belonging to A. We can compute the left hand side of the inequality for z as a polynomial in q like,

$$(\eta - 1 - q) + (\eta - 1)\frac{q(q-1)}{2} - (\eta - 2)q$$

Lacks to see that this polynomial takes non-negative values for $q \geq 0$, $q \leq |A|$, $q \leq \eta - 1$. In fact, we will see after few computations that it takes non-negative values for all q integer.

$$\begin{aligned}
&(\eta - 1 - q) + (\eta - 1)\frac{q(q-1)}{2} - (\eta - 2)q = \\
= \ &\frac{2\eta - 2 - 2q + \eta q^2 - \eta q - q^2 + q - 2\eta q + 4q}{2} = \\
= \ &\frac{q^2(\eta - 1) + q(3 - 3\eta) + 2\eta - 2}{2} = \\
= \ &\frac{(q^2 - 3q + 2)(\eta - 1)}{2}
\end{aligned}$$

Given that $\eta - 1 \geq 1$ then $(\eta - 1)/2 > 0$ and remains only the term $q^2 - 3q + 2$ which is clearly non-negative. Simply plotting it it is seen that it takes only negative values in the range $q \in (1, 2)$ where non integers are considered. Consequently, the left hand side of the inequality is greater or equal than 0 and the inequality is satisfied by z for any clique containing node i too. □

PROPOSITION 8.10. *Given $i \in V_n$ and $A \subseteq V_n \setminus \{i\}$, the Node-to-Set Inequality for i, A is never facet-defining for $n \leq \eta + |A|$.*

PROOF. Recall $2 \leq \eta \leq n - 2$. Let $L = V_n \setminus (A \cup \{i\})$. We begin by proving that the *Node-to-Set Inequality* for i, A is never facet-defining if $\eta = 2$. In this case the *Node-to-Set Inequality* is written like

$$\sum_{e \in [\{i\}:L]} z_e + \sum_{e \in E_n(A)} z_e \geq 0$$

which can be trivially generated adding the valid inequalities $z_e \geq 0$ for all $e \in [\{i\} : L] \cup E_n(A)$.

Now let $\eta \geq 3$, $n \leq \eta + |A|$. Then $|L| = n - |A| - 1 \leq \eta - 1$. The *Node-to-Set Inequality* for i, A at equality can be written as

$$\sum_{e \in [\{i\}:L]} z_e + (\eta - 1) \sum_{e \in E_n(A)} z_e - (\eta - 2) \sum_{e \in [\{i\}:A]} z_e = 0$$

Now we distinguish which feasible cliques satisfy this equality.
Suppose one of those cliques does not contain i. Then $(\eta - 1)\sum_{e \in E_n(A)} z_e$ must be 0 so it must contain at most one node from A.

Now suppose node i is contained in the clique. Let q be the integer number of nodes in the clique belonging to A. Recalling the proof of *Proposition 8.9* we know that, to satisfy the inequality at equality, it must be that,

$$\frac{(q^2 - 3q + 2)(\eta - 1)}{2} = 0$$

Given that in this case $\eta - 1 \geq 2$ the only possible q are $q = 1$, $q = 2$. This allows us to characterise all the feasible cliques satisfying the *Node-to-Set Inequality* at equality.

(a) Not containing node i and
  (i) Containing 0 nodes in A and η in L. (1)
  (ii) Containing 1 node in A and $\eta - 1$ in L. (2)
(b) Containing node i and
  (i) Containing 1 node in A and $\eta - 2$ in L. (3)
  (ii) Containing 2 nodes in A and $\eta - 3$ in L. (4)

Note, nevertheless, that cliques of type (1) are not possible in this case given that $|L| \leq \eta - 1$. We must only consider cliques of type (2), (3) and (4).
First of all we discard the case $L = \emptyset$. In this case given that $\eta \geq 3$ there only remain solutions of type (4) and only in the case $\eta = 3$ exactly. In this case clearly $|A| = n - 1$ and it is easy to see there are only $\binom{n-1}{2}$ solutions of type (4). It is also easy to see that this number of solutions is strictly less than $|E_n| - 1 = \binom{n}{2} - 1$ for all $n \geq 4$ and therefore there is no facet if $L = \emptyset$.
Now suppose $L \neq \emptyset$. Let us construct an equation (*) containing solutions (2), (3) and (4) as done in

previous proofs. The independence of equation (*) with respect to the *Node-to-Set Inequality* at equality and the fixed size equation will imply there is no facet. Let l be an arbitrary node in L. It exists because $L \neq \emptyset$. Consider the equation

$$\sum_{e \in \delta(i) \setminus (i,l)} z_e + \sum_{e \in \delta(l) \setminus (i,l)} z_e - (\eta - 3) z_{il} = \eta - 1 \quad (*)$$

Now we check if equation (*) is tight for solutions of type (2), (3) and (4).

- Consider a solution of type (2). It does not contain i but contains $\eta - 1$ nodes in L. Given that $|L| \leq \eta - 1$ it must have its maximum size and be fully contained in the solution $\Rightarrow l$ must be in the solution. Consequently we have
$$\sum_{e \in \delta(l) \setminus (i,l)} z_e = \eta - 1$$
and equation (*) is tight.
- Consider a solution of type (3). It contains i, one element in A and $\eta - 2$ in L.
Suppose l is not in the solution. Then in the left hand side of equation (*) only remains $\sum_{e \in \delta(i) \setminus (i,l)} z_e$ which takes value $\eta - 1$ and equation (*) is tight.
Suppose l is in the solution. Then the left hand side of equation (*) is
$$(\eta - 2) + (\eta - 2) - (\eta - 3) = \eta - 1$$
so certainly equation (*) is tight too.
- Consider a solution of type (4). Analogous to the case on (3) distinguishing the case l is contained or not in the solution.

To see the independence of equation (*) we can simply find a feasible clique tight for (*) but not for the *Node-to-Set inequality* at equality. Let any clique of size $\eta \geq 3$ containing l, not containing i and containing at least two nodes in A. It exists because, $\eta \geq 3$ and $|A| \geq 2$. Clearly this clique is not tight for the *Node-to-Set inequality* at equality given that it does not contain i and is not of type (1) or (2). However this clique is tight for (*) given that in the left hand side of (*) there only remains $\sum_{e \in \delta(l) \setminus (i,l)} z_e$ which takes value $\eta - 1$ exactly. □

Before exploring in general the *Node-to-Set inequality* we will focus first in the particular case $\eta = 3$ given that it has singularities with respect to the facial structure. Until now, we have discarded having facets for $n \leq \eta + |A|$ so for $\eta = 3$ we know there exist no facets for $n \leq 3 + |A|$. Furthermore, in this case we can also discard the case $n = 4 + |A|$.

PROPOSITION 8.11. *Let $\eta = 3$. Given $i \in V_n$, $A \subset V_n \setminus \{i\}$, $|A| \geq 2$, the Node-to-Set inequality for i, A is not facet-defining if $n = 4 + |A|$.*

PROOF. Let $L = V_n \setminus A \cup \{i\} \Rightarrow |L| = n - |A| - 1 = 3$. Let $L = \{l_1, l_2, l_3\}$. Let us recall the *Node-to-Set inequality* for i, A written in terms of L at equality (for $\eta = 3$):

$$\sum_{e \in [\{i\}:L]} z_e + 2 \sum_{e \in E_n(A)} z_e - \sum_{e \in [\{i\}:A]} z_e = 0$$

Let us recall from the proof of *Proposition 8.10* the different types of feasible cliques satisfying the inequality at equality (for $\eta = 3$):

(a) Cliques not containing node i and
  (i) Containing 0 nodes in A and 3 in L. (1)
  (ii) Containing 1 node in A and 2 in L. (2)
(b) Cliques containing node i and
  (i) Containing 1 node in A and 1 in L. (3)
  (ii) Containing 2 nodes in A and 0 in L. (4)

Similarly to the proof of *Proposition 8.10*, we introduce an equation satisfied by all solutions of type (1), (2), (3) and (4) and linearly independent to the *Node-to-Set inequality* at equality and the fixed size equation. The new equation is denoted again as (*). Let $l = l_1$, for example (works also for l_2, l_3).

$$\sum_{e \in \delta(i) \setminus (i,l)} z_e + 2 \sum_{e \in [\{l\}:A]} z_e + 2z_{l_2 l_3} - z_{il} = 2 \qquad (*)$$

Now we check that equation (*) is tight for solutions of type (1), (2), (3) and (1).

- Consider a clique of type (1). It simply contains all L and in the left hand side there only remains $2z_{l_2 l_3}$ which is exactly 2 and equation (*) is therefore tight.
- Consider a clique of type (2). It might contain l or not. If it does not then it contains l_2, l_3 and its analogous to case (1). If it contains l, let $a \in A$ be the node in the clique belonging to A. In the left hand side of equation (*) there only remains $2z_{al}$ and equation (*) is tight too.
- Consider a clique of type (3). Let $a \in A$ be the node in the clique belonging to A. Suppose l is in the solution. Then the left hand side is $z_{ia} + 2z_{al} - z_{il} = 2$ and equation (*) is tight.
  Suppose l is not in the solution. Let l' be the other node in the clique belonging to L. Regardless on whether l' is l_2 or l_3 in the left hand side there only remains $z_{ia} + z_{il'}$ so equation (*) is tight.
- Consider a clique of type (4). Let a_1, $a_2 \in A$ be the two nodes in the clique belonging to A. Then we have $z_{ia_1} + z_{ia_2}$ and equation (*) is tight once more.

To check the independence if we write a general linear combination of the *Node-to-Set inequality* at equality and the fixed size equation with coefficients λ, γ respectively (for $\eta = 3$),

$$(\lambda + \gamma) \sum_{e \in [\{i\}:L]} z_e + [2\lambda + \gamma] \sum_{e \in E_n(A)} z_e - [\lambda - \gamma] \sum_{e \in [\{i\}:A]} z_e + \gamma \sum_{e \in [A:L] \cup E_n(L)} z_e = 3\gamma$$

Note that $E_n = E_n(A) \sqcup E_n(L) \sqcup [A:L] \sqcup [\{i\}:A] \sqcup [\{i\}:L]$. Imposing this linear combination to be equal to (*) leaves no solution for λ, γ. To see that simply note that in any linear combination z_{il_2} and z_{il} have the same multiplying coefficient $(\lambda + \gamma)$ whereas in (*) they have coefficients 1 and -1 respectively. □

PROPOSITION 8.12. *Let $\eta = 3$. Given $i \in V_n$, $A \subset V_n \setminus \{i\}$, $|A| \geq 2$, the Node-to-Set inequality for i, A is facet-defining if and only if $n \geq 5 + |A|$.*

PROOF. We already know that for $\eta = 3$ it is not facet-defining for $n \leq 4 + |A|$ by *Propositions 8.10, 8.11*. Lacks only to prove that it is facet-defining for $n \geq 5 + |A|$. We will use again the notation for the feasible cliques satisfying the inequality at equality used in the proof of *Proposition 8.11* (cliques of size of type (1), (2), (3) and (4)). Let us denote as l_s an arbitrary node in $L = V_n \setminus (A \cup \{i\})$, $|L| = n - |A| - 1 \geq 4$. Let us also note as a_s an arbitrary node in A, $|A| \geq 2$. Again different subindexes r, q, s, t, ... mean implicitly the elements are different, as usual.

Suppose there exists an equation $\mu z = \mu_o$ satisfied by all cliques of type (1), (2), (3) and (4). Let us explore the characteristics of μ, μ_o using cliques of these types.

$$\begin{aligned}
&\{l_1, l_2, l_3\} : \{l_2, l_3, l_4\} \\
&\qquad \Rightarrow \mu_{l_1 l_2} + \mu_{l_1 l_3} = \mu_{l_2 l_4} + \mu_{l_3 l_4} \\
&\{l_1, l_2, l_4\} : \{l_2, l_3, l_4\} \\
&\qquad \Rightarrow \mu_{l_1 l_2} + \mu_{l_1 l_4} = \mu_{l_2 l_3} + \mu_{l_3 l_4} \\
&\{l_1, l_3, l_4\} : \{l_2, l_3, l_4\} \\
&\qquad \Rightarrow \mu_{l_1 l_3} + \mu_{l_1 l_4} = \mu_{l_2 l_3} + \mu_{l_2 l_4}
\end{aligned}$$

By summing the first two and subtracting the third one, the result is

$$\begin{aligned}
&2\mu_{l_1 l_2} = 2\mu_{l_3 l_4} \\
\underset{\substack{l_1,\, l_2,\, l_3,\, l_4 \\ \text{arbitrary}}}{\Rightarrow} \quad &\mu_{l_r l_q} = \mu_{l_s l_t} \qquad \forall\, l_r, l_q, l_s, l_t \in L \qquad (i)
\end{aligned}$$

In parallel,

$$\begin{aligned}\{a_1,l_1,l_2\}:\{a_1,l_3,l_4\}&\\ \Rightarrow \mu_{a_1l_1}+\mu_{a_1l_2}+\mu_{l_1l_2}&=\mu_{a_1l_3}+\mu_{a_1l_4}+\mu_{l_3l_4}\\ \underset{(i)}{\Rightarrow} \mu_{a_1l_1}+\mu_{a_1l_2}&=\mu_{a_1l_3}+\mu_{a_1l_4}\end{aligned}$$

Analogously exchanging the roles of l_2 and l_4.

$$\Rightarrow \mu_{a_1l_1}+\mu_{a_1l_4}=\mu_{a_1l_2}+\mu_{a_1l_3}$$

Given $a_1 \in A$ and l_1, l_2, l_3, $l_4 \in L$ are arbitrary, this immediately implies that given $a \in A$, μ_{al_r} : ct for all $l_r \in L$. We can denote as μ_a this constant for every $a \in A$. Furthermore,

$$\begin{aligned}\{a_1,l_1,l_2\}:\{a_1,l_1,l_3\}&\\ \Rightarrow\ \mu_{a_1}+\mu_{l_1l_2}&=\mu_{a_1}+\mu_{l_1l_3}\\ \Rightarrow\ \mu_{l_1l_2}&=\mu_{l_1l_3}\\ \underset{\substack{l_1,\,l_2,\,l_3,\,l_4\\ \text{arbitrary}\\ (i)\mu_{l_rl_q}=\mu_{l_sl_t}}}{\Rightarrow}\ \mu_e\ :\ ct&\quad \forall\ e\in E_n(L)\end{aligned}$$

We denote this constant as $\mu_{E_n(L)}$.
Now we note that,

$$\begin{aligned}\{a_1,l_1,l_2\}:\{a_2,l_1,l_2\}&\\ \Rightarrow\ 2\mu_{a_1}&=2\mu_{a_2}\\ \underset{\substack{a_1,\,a_2\\ \text{arbitrary}}}{\Rightarrow}\ \mu_e\ :\ ct&\quad \forall\ e\in[A:L]\end{aligned}$$

By considering the equation provided by $\{a_1,l_1,l_2\}$:$\{l_1,l_2,l_3\}$ we can easily see that in fact this constant is the same as $\mu_{E_n(L)}$. In summary μ_e : ct $\forall\ e \in [A:L] \cup E_n(L)$. We denote this constant as $\mu_2 = \mu_{E_n(L)}$ to be more inclusive. Continuing,

$$\begin{aligned}\{i,a_1,l_1\}:\{i,a_1,l_2\}&\\ \Rightarrow\ \mu_{il_1}+\mu_2&=\mu_{il_2}+\mu_2\\ \underset{\substack{l_1,\,l_2\\ \text{arbitrary}}}{\Rightarrow}\ \mu_e\ :\ ct&\quad \forall\ e\in[\{i\}:L]\end{aligned}$$

We denote this constant as μ_1. More than that,

$$\begin{aligned}\{i,a_1,l_1\}:\{i,a_2,l_1\}&\\ \Rightarrow\ \mu_{ia_1}+\mu_2&=\mu_{ia_2}+\mu_2\\ \underset{\substack{a_1,\,a_2\\ \text{arbitrary}}}{\Rightarrow}\ \mu_e\ :\ ct&\quad \forall\ e\in[\{i\}:A]\end{aligned}$$

We denote this constant as μ_3. We can obtain a relation between μ_3 and the constants μ_1, μ_2.

$$\begin{aligned}\{i,a_1,l_1\}:\{a_1,l_1,l_2\}&\\ \Rightarrow\ \mu_{ia_1}+\mu_{il_1}&=\mu_{a_1l_2}+\mu_{l_1l_2}\\ \Rightarrow\ \mu_3&=2\mu_2-\mu_1\end{aligned}$$

Finally,

$$\begin{aligned}\{i,a_1,a_2\}:\{i,a_1,l_1\}&\\ \Rightarrow\ \mu_{a_1a_2}+\mu_3&=\mu_1+\mu_2\\ \Rightarrow\ \mu_{a_1a_2}&=\mu_1+\mu_2-\mu_3=2\mu_1-\mu_2\\ \underset{\substack{a_1,\,a_2\\ \text{arbitrary}}}{\Rightarrow}\ \mu_e\ :\ ct&\quad \forall\ e\in E_n(A)\end{aligned}$$

In summary,

$$\begin{aligned}
&\mu_e = \mu_1 && \forall\ e \in [\{i\} : L] \\
&\mu_e = \mu_2 && \forall\ e \in [A : L] \cup E_n(L) \\
&\mu_e = 2\mu_2 - \mu_1 && \forall\ e \in [\{i\} : A] \\
&\mu_e = 2\mu_1 - \mu_2 && \forall\ e \in E_n(A)
\end{aligned}$$

We can compute now μ_o in terms of μ_1 and μ_2. Considering the cliques of type (1) $\{l_1, l_2, l_3\}$ it is clear that $\mu z = \mu_o = \mu_2\eta(\eta-1)/2 = 3\mu_2$ given that all edges are in $E_n(L)$. As usual, we write the equation $\mu z = \mu_o$ in terms of μ_1, μ_2.

$$\mu_1 \sum_{e\in[\{i\}:L]} z_e + (2\mu_1 - \mu_2) \sum_{e\in E_n(A)} z_e + (2\mu_2 - \mu_1) \sum_{e\in[\{i\}:A]} z_e + \mu_2 \sum_{e\in[A:L]\cup E_n(L)} z_e = 3\mu_2$$

Let us recall the expression for a general linear combination of the *Node-to-Set Inequality* for i, A, at equality, and the fixed size equation, with $\eta = 3$.

$$(\lambda+\gamma) \sum_{e\in[\{i\}:L]} z_e + [2\lambda+\gamma] \sum_{e\in E_n(A)} z_e - [\lambda-\gamma] \sum_{e\in[\{i\}:A]} z_e + \gamma \sum_{e\in[A:L]\cup E_n(L)} z_e = 3\gamma$$

It is easy to see that taking $\lambda = \mu_1 - \mu_2$ and $\gamma = \mu_2$ generates the equation $\mu z = \mu_o$. Consequently the *Node-to-Set Inequality* for i, A is facet-defining if $\eta = 3$, $n \geq 5 + |A|$. □

In conclusion for the case $\eta = 3$ we have demonstrated that the *Node-to-Set Inequalities* are facet-defining if and only $n \geq 5 + |A|$. Now we can study the general case for $\eta \geq 4$. Recall *Proposition 8.10* stating there is no facet for $n \leq \eta + |A|$. In the proof of *Proposition 8.10* we also noticed there is no facet if $\eta = 2$. Follows a theorem proving that in the remaining cases, the *Node-to-Set Inequalities* define facets.

THEOREM 8.13. *Let $\eta \geq 4$. Given $i \in V_n$, $A \subset V_n \setminus \{i\}$, $|A| \geq 2$, the Node-to-Set Inequality for i, A is facet-defining if and only if $n \geq \eta + 1 + |A|$.*

PROOF. We already know, for *Proposition 8.10*, that it is not facet-defining for $n \leq \eta + |A|$ so lacks only to prove it is facet-defining for $n \geq \eta + 1 + |A|$. Let us denote as usual $L = V_n \setminus (A \cup \{i\})$. Note that $|L| = n - |A| - 1 \geq \eta$. Let us denote as l_s an arbitrary node in L, a_s an arbitrary node in A, $|A| \geq 2$. Again different subindexes r, q, s, t,... mean implicitly that the elements are different. Recall the *Node-to-Set Inequality* for i, A at equality written in terms of L.

$$\sum_{e\in[\{i\}:L]} z_e + (\eta-1) \sum_{e\in E_n(A)} z_e - (\eta-2) \sum_{e\in[\{i\}:A]} z_e = 0$$

Let us recall the feasible cliques satisfying this equation:

(a) Not containing node i and
  (i) Containing 0 nodes in A and η in L. (1)
  (ii) Containing 1 node in A and $\eta - 1$ in L. (2)
(b) Containing node i and
  (i) Containing 1 node in A and $\eta - 2$ in L. (3)
  (ii) Containing 2 nodes in A and $\eta - 3$ in L. (4)

As always we are going to suppose there exists an equation $\mu z = \mu_o$ satisfied for all those cliques and demonstrate it can be generated with a linear combination of the *Node-to-Set Inequality* at equality and the fixed size equation. We will use the typical notation $\mathcal{A} : \mathcal{B}$ to impose $\mu z^{\mathcal{A}} = \mu z^{\mathcal{B}}$ for $\mathcal{A}$, $\mathcal{B}$ two different cliques from the families (1), (2), (3) and (4) defined above.

$$\begin{aligned}
\{a_1, l_1 \ldots, l_{\eta-1}\} : \{a_2, l_1, \ldots, l_{\eta-1}\} \\
\Rightarrow \mu_{a_1 l_1} + \ldots + \mu_{a_1 l_{\eta-1}} = \mu_{a_2 l_1} + \ldots + \mu_{a_2 l_{\eta-1}} \\
\{a_1, l_1, \ldots, l_{\eta-2}, l_\eta\} : \{a_2, l_1, \ldots, l_{\eta-2}, l_\eta\} \\
\Rightarrow \mu_{a_1 l_1} + \ldots + \mu_{a_1 l_{\eta-2}} + \mu_{a_1 l_\eta} = \mu_{a_2 l_1} + \ldots + \mu_{a_2 l_{\eta-2}} + \mu_{a_2 l_\eta}
\end{aligned}$$

By subtracting, the result is

$$\begin{array}{c} \mu_{a_1 l_{\eta-1}} - \mu_{a_1 l_\eta} = \mu_{a_2 l_{\eta-1}} - \mu_{a_2 l_\eta} \\ \underset{\substack{a_1,\, a_2,\, l_{\eta-1},\, l_\eta \\ \text{arbitrary}}}{\Rightarrow} \mu_{a_r l_s} - \mu_{a_r l_t} = \mu_{a_q l_s} - \mu_{a_q l_t} \qquad \forall\, a_r, a_q \in A\ l_s, l_t \in L \end{array} \qquad (i)$$

If we consider *(i)* fixing $a_r = a_1$, $a_q = a_2$, $l_s = l_{\eta-1}$ and moving $l_t = l_1, \ldots, l_{\eta-2}$ and sum all those $\eta - 2$ equations to $\{a_1, l_1 \ldots, l_{\eta-1}\} : \{a_2, l_1, \ldots, l_{\eta-1}\}$ we obtain,

$$\mu_{a_1 l_{\eta-1}} = \mu_{a_2 l_{\eta-1}}$$

and given the arbitrariness it means $\mu_{a_r l_s} = \mu_{a_q l_s}\ \forall\ a_r,\ a_q \in A,\ l_s \in L$. We can denote as μ_{l_s} this constant for each node $l_s \in L$. Moreover,

$$\begin{array}{l} \{i, a_1, l_1, \ldots, l_{\eta-2}\} : \{i, a_2, l_1, \ldots, l_{\eta-2}\} \\ \qquad \Rightarrow\ \mu_{ia_1} + \mu_{a_1 l_1} + \ldots + \mu_{a_1 l_{\eta-2}} = \mu_{ia_2} + \mu_{a_2 l_1} + \ldots + \mu_{a_2 l_{\eta-2}} \\ \qquad \Rightarrow\ \mu_{ia_1} + \mu_{l_1} + \ldots + \mu_{l_{\eta-2}} = \mu_{ia_2} + \mu_{l_1} + \ldots + \mu_{l_{\eta-2}} \\ \qquad \Rightarrow\ \mu_{ia_1} = \mu_{ia_2} \\ \underset{\substack{a_1,\, a_2 \\ \text{arbitrary}}}{\Rightarrow}\ \mu_{ia_r} = \mu_{1a_q} \qquad \forall\, a_r, a_q \in A \\ \qquad \Rightarrow\ \mu_e \ :\ ct \qquad \forall\, e \in [\{i\} : A] \end{array}$$

On the other hand, subtracting $\{i, a_1, l_1 \ldots, l_{\eta-3}, l_\eta\} : \{i, a_1, l_2, \ldots, l_{\eta-3}, l_{\eta-1}, l_\eta\}$ and $\{i, a_1, a_2, l_1 \ldots, l_{\eta-3}\} : \{i, a_1, a_2, l_2, \ldots, l_{\eta-3}, l_{\eta-1}\}$ the result is

$$\mu_{l_1 l_\eta} - \mu_{a_2 l_1} = \mu_{l_{\eta-1} l_\eta} - \mu_{a_2 l_{\eta-1}}$$

which added to $\{a_2, l_1 \ldots, l_{\eta-2}, l_\eta\} : \{a_2, l_2, \ldots, l_\eta\}$ results in

$$2\mu_{l_1 l_\eta} + \mu_{l_1 l_2} + \ldots + \mu_{l_1 l_{\eta-2}} = 2\mu_{l_{\eta-1} l_\eta} + \mu_{l_2 l_{\eta-1}} + \ldots + \mu_{l_{\eta-2} l_{\eta-1}} \qquad (*)$$

In parallel,

$$\begin{array}{l} \{i, a_1, l_1 \ldots, l_{\eta-3}, l_\eta\} : \{i, a_1, l_2, \ldots, l_{\eta-3}, l_{\eta-1}, l_\eta\} \\ \qquad \Rightarrow \mu_{il_1} + \mu_{a_1 l_1} + \mu_{l_1 l_2} + \ldots + \mu_{l_1 l_{\eta-3}} + \mu_{l_1 l_\eta} = \mu_{il_{\eta-1}} + \mu_{a_1 l_{\eta-1}} + \mu_{l_2 l_{\eta-1}} + \ldots + \mu_{l_{\eta-3} l_{\eta-1}} + \mu_{l_{\eta-1} l_\eta} \\ \{i, a_1, l_1 \ldots, l_{\eta-2}\} : \{i, a_1, l_2, \ldots, l_{\eta-1}\} \\ \qquad \Rightarrow \mu_{il_1} + \mu_{a_1 l_1} + \mu_{l_1 l_2} + \ldots + \mu_{l_1 l_{\eta-2}} = \mu_{il_{\eta-1}} + \mu_{a_1 l_{\eta-1}} + \mu_{l_2 l_{\eta-1}} + \ldots + \mu_{l_{\eta-2} l_{\eta-1}} \end{array}$$

By subtracting, the result is

$$\begin{array}{c} \mu_{l_1 l_\eta} - \mu_{l_1 l_{\eta-2}} = \mu_{l_{\eta-1} l_\eta} - \mu_{l_{\eta-2} l_{\eta-1}} \\ \underset{\substack{l_1,\, l_{\eta-2},\, l_{\eta-1},\, l_\eta \\ \text{arbitrary}}}{\Rightarrow} \mu_{l_r l_t} - \mu_{l_r l_q} = \mu_{l_s l_t} - \mu_{l_q l_s} \qquad \forall\, l_r, l_q, l_s, l_t \in L \end{array} \qquad (ii)$$

Now we add $\eta - 3$ equations *(ii)* fixing $l_r = l_1$, $l_s = l_{\eta-1}$, $l_t = l_\eta$ and moving $l_q = l_2, \ldots, l_{\eta-2}$. The result is,

$$(\eta-3)\mu_{l_1 l_\eta} - \mu_{l_1 l_2} - \ldots - \mu_{l_1 l_{\eta-2}} = (\eta-3)\mu_{l_{\eta-1} l_\eta} - \mu_{l_2 l_{\eta-1}} - \ldots - \mu_{l_{\eta-2} l_{\eta-1}}$$

which added directly to (*) brings

$$\begin{array}{c} (\eta-1)\mu_{l_1 l_\eta} = (\eta-1)\mu_{l_{\eta-1} l_\eta} \\ \underset{\substack{l_1,\, l_{\eta-1},\, l_\eta \\ \text{arbitrary}}}{\Rightarrow} \mu_{l_r l_q} = \mu_{l_s l_q} \qquad \forall\, l_r, l_q, l_s \in L \end{array}$$

what eventually means $\mu_e : \text{ct}\ \forall\ e \in E_n(L)$. The rest is a generalisation of the proof of *Proposition 3.11* so we will avoid the details on computation.
Considering the equation $\{a_1, l_1, \ldots, l_{\eta-1}\} : \{a_1, l_2, \ldots, l_\eta\}$ and taking into account $\mu_e : \text{ct}\ \forall\ e \in E_n(L)$ it is

immediate to notice that

$$\begin{aligned} & \mu_{a_1 l_1} = \mu_{a_1 l_\eta} \\ \underset{\substack{a_1,\, l_1,\, l_\eta \\ \text{arbitrary}}}{\Rightarrow} & \mu_{a_r l_q} = \mu_{a_r l_s} \qquad \forall\, a_r \in A,\, l_q, l_s \in L \\ & \Rightarrow \mu_e : ct \qquad \forall\, e \in [A : L] \end{aligned}$$

Considering equation $\{l_1, \ldots, l_\eta\} : \{a_1, l_1, \ldots, l_{\eta-1}\}$ it is immediate to see that the constants $\mu_e\ \forall e \in [A : L]$ and $\mu_e\ \forall e \in E_n(L)$ are equal. We can denote as μ_2 this constant just as in the proof of *Proposition 8.12*. In conclusion, $\mu_2 \equiv \mu_e\ \forall e \in E_n(L) \cup [A : L]$, i.e., $\forall e \notin \delta(i) \cup E_n(A)$. Note again that $E_n = E_n(A) \sqcup E_n(L) \sqcup [A : L] \sqcup [\{i\} : A] \sqcup [\{i\} : L] = E_n(A) \sqcup E_n(L) \sqcup \delta(i) \sqcup [A : L]$.

Imposing this in $\{i, a_1, l_1, \ldots, l_{\eta-2}\} : \{i, a_1, l_2, \ldots, l_{\eta-1}\}$ it is immediate that

$$\begin{aligned} & \mu_{i l_1} = \mu_{i l_{\eta-1}} \\ \underset{\substack{l_1,\, l_{\eta-1} \\ \text{arbitrary}}}{\Rightarrow} & \mu_{i l_r} = \mu_{i l_q} \qquad \forall\, l_r, l_q \in L \\ & \Rightarrow \mu_e : ct \qquad \forall\, e \in [\{i\} : L] \end{aligned}$$

At this point, we can denote the constants we already know,

$$\begin{aligned} \mu_e &= \mu_1 && \forall\, e \in [\{i\} : L] \\ \mu_e &= \mu_2 && \forall\, e \in [A : L] \cup E_n(L) \\ \mu_e &= \mu_3 && \forall\, e \in [\{i\} : A] \end{aligned}$$

Now we use them considering,

$$\begin{aligned} \{i, a_1, a_2, l_1 \ldots, l_{\eta-3}\} : \{i, a_1, l_1, \ldots, l_{\eta-2}\} & \\ \Rightarrow \mu_{i a_2} + \mu_{a_1 a_2} + \mu_{a_2 l_1} + \ldots + \mu_{a_2 l_{\eta-3}} &= \mu_{i l_{\eta-2}} + \mu_{a_1 l_{\eta-2}} + \mu_{l_1 l_{\eta-2}} + \ldots + \mu_{l_{\eta-3} l_{\eta-2}} \\ \Rightarrow \mu_3 + \mu_{a_1 a_2} + (\eta - 3)\mu_2 &= \mu_1 + (\eta - 2)\mu_2 \\ \Rightarrow \mu_e : \text{ct} \qquad & \forall\, e \in E_n(A) \\ \Rightarrow \mu_e = \mu_1 + \mu_2 - \mu_3 \qquad & \forall\, e \in E_n(A) \end{aligned}$$

Similarly to the proof of *Proposition 8.12* we can express μ_3 in terms of μ_1, μ_2.

$$\begin{aligned} \{i, a_1, l_1 \ldots, l_{\eta-2}\} : \{a_1, l_1, \ldots, l_{\eta-1}\} & \\ \Rightarrow \mu_{i a_1} + \mu_{i l_1} + \ldots + \mu_{i l_{\eta-2}} &= \mu_{a_1 l_{\eta-1}} + \mu_{l_1 l_{\eta-1}} + \ldots + \mu_{l_{\eta-2} l_{\eta-1}} \\ \Rightarrow \mu_3 + (\eta - 2)\mu_1 &= (\eta - 1)\mu_2 \\ \Rightarrow \mu_3 = (\eta - 1)\mu_2 - (\eta - 2)\mu_1 & \end{aligned}$$

Eventually, we have noticed that every edge coefficient in μ depends on μ_1, μ_2 only two independent coefficients. Putting all together similarly as in the proof of *Proposition 8.12*

$$\begin{aligned} \mu_e &= \mu_1 && \forall\, e \in [\{i\} : L] \\ \mu_e &= \mu_2 && \forall\, e \in [A : L] \cup E_n(L) \\ \mu_e &= (\eta - 1)\mu_2 - (\eta - 2)\mu_1 && \forall\, e \in [\{i\} : A] \\ \mu_e &= (\eta - 1)\mu_1 - (\eta - 2)\mu_2 && \forall\, e \in E_n(A) \end{aligned}$$

We can compute μ_o considering the clique of type (1) $\{l_1, \ldots, l_\eta\}$ such that $\mu_o = \mu_2 \cdot \eta(\eta - 1)/2$. Analogously to the proof *Proposition 8.12*, taking the equality for the *Node-to-Set Inequality* multiplied by $\mu_1 - \mu_2$ and the fixed size equation multiplied by μ_2 and adding both, results in the equation $\mu z = \mu_o$. Therefore, the equation $\mu z = \mu_o$ is linearly dependant with the *Node-to-Set Inequality* at equality and the fixed size equation. Consequently the *Node-to-Set Inequality* at equality defines a facet. □

4. Conclusions on the Pricing Problem with fixed size

In summary we can state briefly the different theoretical results of our polyhedral analysis on the Pricing Problem with fixed size.
Let $G = (V_n, E_n)$ be the complete graph with n nodes, η an integer with the fixed size the clique solution must have. Let Z be the set of all feasible (size η) cliques described with the variables for the edges. Recall that $\mathcal{P}_{n,\eta} = \text{conv}(Z)$ is the Convex Hull polyhedron for our Pricing Problem.

We state below the different results obtained in relation to Z the set of feasible cliques as well as the polyhedral structure for $\mathcal{P}_{n,\eta}$.

(1) $\dim(\mathcal{P}_{n,\eta}) = |E_n| - 1$
The polyhedron is not full dimensional but there is one single equation satisfied by all points, the fixed size equation actually.
(2) The upper bound original inequalities $z_e \leq 1$ are never facet-defining. Moreover, the faces they define are contained in faces of *Node-to-Node inequalities*.
(3) The lower bound original inequalities $z_e \geq 0$, $e \in E_n$, are always facet-defining for $\mathcal{P}_{n,\eta}$ if $\eta = 2$. For $\eta > 2$ they are facet-defining if and only if $n \geq \eta + 3$.
(4) Let $n \geq 5$. Given $e \in E_n$, $e = (i, j)$ for i, $j \in V_n$. The *Upper bound pair inequality* for i and j,

$$\sum_{e \in \delta(i,j)} z_e - (\eta - 3) z_{ij} \leq \eta - 1$$

is valid for Z. It is also facet-defining for $\mathcal{P}_{n,\eta}$ if and only if $\eta \geq 3$.
(5) Given $W \subset V_n$ with $|W| \leq \lfloor n/2 \rfloor$. Let w be the size of W. Let us discard the case $w = 2$ and $\eta = 3$, which corresponds to an *Upper bound pair inequality*. Then, the *δ-Set inequality* for W,

$$(i) \quad \sum_{e \in \delta(W)} z_e \leq w(\eta - w) \qquad \text{if } w \leq \lfloor \eta/2 \rfloor$$

$$(ii) \quad \sum_{e \in \delta(W)} z_e \leq \lfloor \eta/2 \rfloor \lceil \eta/2 \rceil \qquad \text{if } w > \lfloor \eta/2 \rfloor$$

is valid for Z and facet-defining if and only if $w \geq 3$, $\eta \in \{3, \ldots, 2w - 3\}$, η odd.
(6) Let i, $j \in V_n$, $i \neq j$. The *Node-to-Node Inequality* for i, j,

$$\sum_{e \in \delta(i) \setminus (i,j)} z_e - (\eta - 2) z_{ij} \geq 0$$

is valid for Z and facet-defining if and only if $\eta \geq 3$ and $n \geq \eta + 3$.
(7) Given $i \in V_n$ and $A \subseteq V_n \setminus \{i\}$, $|A| \geq 2$, the *Node-to-Set Inequality* for i, A,

$$\sum_{j \in V \setminus A} z_{ij} + (\eta - 1) \sum_{e \in E_n(A)} z_e - (\eta - 2) \sum_{j \in A} z_{ij} \geq 0$$

is valid for Z. Furthermore, for $\eta = 3$ it is facet-defining if and only if $n \geq 5 + |A|$. For $\eta \geq 4$ it is facet-defining if and only if $n \geq \eta + 1 + |A|$.

At this point, the theory block of the document finishes. This does not mean this block is absolutely closed in the project. This means, in fact, that all the results with theoretical proofs for the Pricing Problem achieved are included in the previous pages and summarised above.

With the results already achieved the project has carried on more on practical aspects. The idea is to obtain a first implementation of the Column Generation strategy on CPLEX to test whether the procedure performs well and also test how the facet-defining inequalities obtained might improve CPLEX performance on the Pricing Problem.

Part 4

Implementation of Column Generation & Future Research

Chapter 9
Code

Additionally to the strategy proposed to face microaggregation with a Column Generation scheme and the different theoretical results obtained in relation to the facial structure of our Pricing Problem, a preliminary code has been implemented and tested. This code mainly performs the Column Generation scheme. It uses CPLEX to perform all the optimizations for both the Master and the Pricing problems. All the code is done in *C++*, and the library used for CPLEX is the Callable Library for *C*. Our code also implements the Heuristic solution with the Maximum Distance to Average Vector (MDAV) described in Part 1. This solution has been used to initialize the Master Problem with the first feasible clusters and also provides a SSE value for the heuristic solution. With our code the idea is to perform the Column Generation. It terminates when there are no more feasible clusters with negative reduced cost to be added. The code implements the scheme found on the section *Chapter 7. Section 1.*

The main goals of this code are to test the performance of this scheme in terms of speed, test whether the eventual solutions are too fractional or not and provide with an actual accurate lower bound on the SSE for non approximate Microaggregation. With this lower bound we can for example compute an accurate GAP on the heuristic solution provided by the MDAV on real data. This is one of the main contributions of this project apart from the theory results for the Pricing Problem and the Column Generation scheme itself.

The code is mainly based on classes, inheritance, polimorphism and CPLEX. In fact those are the main issues I have been able to learn and apply in matter of programming. Follows a brief description of the structure:

- ***MAClass:*** A main general class that contains the problem itself. It loads the data, the parameters (minimal size of the clusters or anonymity desired, number of attributes) and allocates memory for the solution. Also includes a member to solve the problem. This member is however virtual and non defined in this class. It will be defined in the inherited classes and used in the main depending on the inherited class (polimorphism).
- ***H_MAClass:*** A directly inherited class of *MAClass*. It defines the member for solving the problem implementing the heuristical solution based on the MDAV algorithm.
- ***MAMasterProblemClass:*** A class that constructs a Master Problem in CPLEX environment. It constructs the environment of CPLEX for this problem and also the object. The problem is initialized with a function to load the clusters from the heuristic and thus populating the partition constraints of the Master Problem for every node.
  Its most relevant members are the procedure of solving and the procedure to add a column. The procedure of solving simply solves the problem at its stage as an LP and provides with both the primal and dual variables solution. Those dual variables will be loaded to the Pricing Problems. The procedure to add a column simply adds a column to the constraints for every node. It is simply adding a new variable in the rows corresponding to the nodes of the cluster added.
- ***MAPricingProblemClass:*** A class that constructs a Pricing Problem in CPLEX environment. It constructs the environment of CPLEX and also the object. An important feature on the code is that the environment for all the Pricing Problems declared is the same, thus minimizing the number of

environments called by CPLEX in the execution. This is done by using an static environment and certain tricks for the destructor of this class. When constructing the Pricing Problem it only does it in cases when enumerating is not a useful solution ($5 \leq \eta \leq n-5$). In this cases when the CPLEX object is constructed we initially add the fixed size constraint, *Upper bound pair inequalities*, *Node-to-Node inequalities* and *Node-to-Set inequalities* for sets of size two. As stated previously this is a preliminary testing code and a separation routine is yet to be done.
The solving member of this class solves the problem with CPLEX as a BIP when the problem is constructed as a CPLEX object ($5 \leq \eta \leq n-5$). In the remaining cases it solves the problem by enumeration.
Given that this is a general description some technical details on the loading of the dual pricing coefficients, the construction of the objective function... are avoided.

- ***CG_MAClass:*** A directly inherited class of *MAClass*. It defines the member for solving the problem implementing the Column Generation scheme. In fact, this is not an actual solution because at the current stage our code terminates after the Column Generation without performing Branching once the solution is fractional.
  This class includes objects of the classes *H_MAClass*, *MAMasterProblemClass*, *MAPricingProblemClass*. The heuristic solution from *H_MAClass* is used to load the first clusters to the Master Problem. The classes *MAMasterProblemClass*, *MAPricingProblemClass* allows us to consider the different problems as elements in our Column Generation Code. The code in the solve of *CG_MAClass* only implements the scheme of the Column Generation but does not use CPLEX at all given that all CPLEX tools are executed internally in the classes for the problems. This makes the code very understandable. It basically consists on executing a heuristic solution, creating a Master Problem as an element of the corresponding class included and similarly creating k Pricing Problems (recall k is the minimal cluster size). Then it runs the Column Generation simply as a loop adding columns to the Master Problem every time a Pricing Problem gives an optimal solution with a negative objective value. Eventually once this loop is over it solves the Master Problem, provides with this solution (fractional or not), the lower bound on the SSE (objective value for the Master Problem) and computes the GAP of the original heuristic solution finally. The GAP is computed as percentage in the following way:

$$GAP\ (\%) = 100 \cdot \frac{SSE_H - SSE_{MP}}{SSE_H}$$

  where SSE_H is the SSE for the MDAV algorithm and SSE_{MP} is the objective value for the Master Problem after the Column Generation.

Follows and scheme representing the dependance relations of the classes implemented to perform our code, see Figure 1.

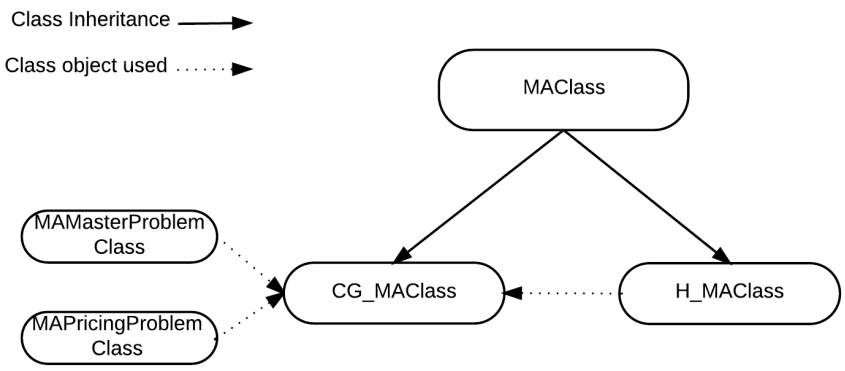

FIG. 1. Diagram describing the class dependance in our code.

Chapter 10
Computational Results

The code has been tested with data sets extracted from actual data widely used in the field of *Statistical Disclosure Control* methods. In particular the data is extracted from the *CASC* project [**2**]. In particular two microdata sets from this project were considered, the "Tarragona" data set and the "Census" data set, which are widely used in the literature. The "Census" data set was obtained on July 27, 2000 using the Data Extraction System of the U. S. Bureau of the Census. Includes 1080 records with 13 attributes each. The "Tarragona" data set comprises figures of 834 companies in the Tarragona area. It was collected during the year 1995 and includes 13 attributes for company too. Details about both data sets can be found in [**2**].

Given that our code is preliminary and at this stage intends to test the Column Generation performance with CPLEX as well as the GAP for the MDAV heuristic algorithm, we consider using subsets of individuals in both microdata sets. In particular we extract the 30, 40 and 50 individuals with less norm from our microdata sets. This is just to make a choice of individuals to test the code.

Recall that in the state of art for CPPMIN the approximate model, Mitchell and Ji in [**10**], [**11**], work with data sets with only two attributes and 103 individuals at most. It seems reasonable for us, then, to work with such a low amount of data given.

Thus we are considering testing our code for subsets of 30, 40 and 50 individuals from both real data sets "Tarragona" and "Census". For each of those subsets of individuals we also test our code considering three different minimal cluster sizes k. The sizes chosen are amongst the typical k-anonymity parameters used in the state of art for microaggregation, those are $k = 3$, 4 and 5.

In Table 1. the results of our several testings are summarised. Follows a brief description of the table variables:

- **Dat. Set:** Indicates the original microdata set from where the data tested was extracted.
- **Ind.:** Indicates the number of individuals extracted. Recall that those are the ones with less norm from the original microdata.
- **Att.:** Indicates the number of attributes of the data. In any case they are 13 all the attributes for the individuals considered from both original microdata sets.
- **Min. Size:** Indicates k the minimal cluster size for our microaggregation, or anonymity desired.
- **Cl. H.:** Indicates the number of clusters generated by the heuristic solution with MDAV algorithm. Recall that those are the clusters that initialize our Master Problem.
- **Cl. Add.:** Indicates the number of clusters added throughout all our Column Generation scheme. Every time a Pricing Problem with fixed size $\eta = k, \ldots, 2k-1$ solved optimally gave a cluster solution with negative objective function it was added to the Master Problem.
- **Cl. Tot.:** Indicates the total number of clusters eventually. This is actually the number of variables of the Master Problem after the Column Generation, when there are no more clusters with negative reduced cost to be added. In fact it is simply the sum of *Cl. H.* and *Cl. Add.*.
- **GAP$_H$ (%):** Indicates the GAP of the heuristic solution in percentage. Recall that after the Column Generation, when there exist no more clusters with negative reduced cost to be added, the Master

Problem is solved and it provides a solution (fractional or not) and a SSE_{MP} its objective value. This objective value SSE_{MP} must be less or equal to the SSE provided by the heuristic solution (SSE_H) given that the Master Problem at the end includes the cluster variables for the heuristically generated clusters at the beginning. Actually this objective value SSE_{MP} is a lower bound on the SSE for non-approximate microaggregation like we stated in previous chapters.
Recall again that this GAP is computed like

$$GAP_H\ (\%) = 100 \cdot \frac{SSE_H - SSE_{MP}}{SSE_H}$$

- **Int. Sol.:** Indicates with a "YES" or a "NO" if the eventual solution of the Master Problem is integer or not. In case it is integer then the solution corresponds to an actual clustering and optimal microaggregation has been achieved simply by Column Generation. If it is not then some Branching must be performed considering the objective value obtained for this Master Problem solution as a lower bound on the SSE.

TABLE 1

Dat. Set	**Ind.**	**Att.**	**Min. Size**	**Cl. H.**	**Cl. Add.**	**Cl. Tot.**	**GAP_H (%)**	**Int. Sol.**
Tarragona	30	13	3	10	132	142	14.25	NO
Tarragona	30	13	4	7	286	293	22.96	NO
Tarragona	30	13	5	6	411	417	23.94	YES
Tarragona	40	13	3	13	161	174	8.49	NO
Tarragona	40	13	4	10	341	351	18.38	NO
Tarragona	40	13	5	8	588	596	16.75	NO
Tarragona	50	13	3	16	206	222	12.35	NO
Tarragona	50	13	4	12	465	477	15.55	NO
Tarragona	50	13	5	10	756	766	12.77	YES
Census	30	13	3	10	132	142	28.54	NO
Census	30	13	4	7	226	233	26.73	NO
Census	30	13	5	6	258	264	7.34	YES
Census	40	13	3	13	175	188	15.97	NO
Census	40	13	4	10	302	312	16.22	YES
Census	40	13	5	8	432	440	8.66	YES
Census	50	13	3	16	214	230	16.61	NO
Census	50	13	4	12	393	405	24.7	NO
Census	50	13	5	10	609	619	26.13	NO

An issue not detailed in Table 1. however is the time performance. It has not been included given that the code is preliminary and it must be deeply optimised in future work. However, we would like to outstand the quality of the code in the sense of simplicity and structure, as it allows us to develop features in parallel through the different classes involved. For example we outstand the polimorphism use that allows to consider the Heuristic and the Column Generation procedures by separate as solution algorithms for the same microaggregation problem. We also remark the implementation of the Master Problem and the Pricing Problem as classes itself, each of them with a static environment for CPLEX what optimizes CPLEX use and calls (for each environment in future coding we could develop a *callback* procedure with new inequalities to be added, branching, etc). Also CPLEX is only executed inside these classes so the code in the Column Generation is simplified as much as possible. In both classes for both problems we can also include heuristic procedures to obtain cliques with negative reduced cost but not optimal in the Pricing Problem or feasible solutions with low SSE in case of the Master Problem. In summary, our code structure allows us to comfortably develop future implementations.

In relation to our computational results we face them with optimism. Although for practical limitations we can not perform yet in greater cases (n and η greater) we have observed that the Column Generation scheme does not generate too many clusters. This probably means our line of research might have future. Besides the GAPs obtained for the heuristic solution of MDAV in our cases are themselves a result which is certainly

new, given that it corresponds to non-approximate microaggregation. This gives content and substance to the project at this stage. Finally we have observed that at some cases (concretely 5 out of 18) the eventual Master Problem solution after the Column Generation is certainly integer. This means that for those cases including real data we have achieved the optimal microaggregation.

This had never been achieved before: it is the first ever developed procedure that can be optimally solve -certificating the optimality- some nontrivial instances of the microaggregation problem.

In the following chapter we include some of the lines our research could follow from now on. Recall that this document is a report of the process, experience and results obtained during the research collaboration until the date of release on early January. In fact, I am carrying on the research collaboration during the following months until March 2016. There is more to be done yet, of course, and probably it follows the lines stated in the following chapter for future research, both on theoretical and implementation aspects.

Chapter 11
Future Research

We can clearly distinguish between work done in paper or theoretical, and work done in implementation. In fact, this two lines of work have split the stay into two in matter of time dedication to the project.

Theoretical aspects

Considering all the results obtained until now, in our opinion the theoretical aspects related to the polyhedral structure of the Pricing Problem are pretty developed. Considering all our results and specially the ones for the families of the *δ-Set inequalities* and the *Node-to-Set inequalities*, we think that there is a strong basis of material in terms of facet-defining inequalities. This does not mean there are no more important inequalities to be explored. Using Porta software for low cases the inequalities stated only cover totally the inequalities observed for n up to six. For the case $n = 7$ some other inequalities have been observed. Some of them have a complex structure and depend on cycles for example. Finding proofs for their validity and facets they could define could be left for future research on the polyhedral and facial structure of the Pricing Problem.

In relation to both the implementation of the Column Generation and the theoretical aspects there also lacks to construct a cutting plane method with an efficient separation routine to add smartly the facet-defining inequalities already known and the ones to introduce yet if necessary. Currently in our preliminary implementation we are adding directly to the Pricing Problem the *fixed size equation*, the *Upper bound pair inequalities*, the *Node-to-Node inequalities* and the *Node-to-Set inequalities* for sets of size two. Obviously it must be improved a lot even to make our code consider smartly the theoretical results on the polyhedral structure of the Pricing Problem that we already have now.

Implementation aspects

There are plenty of features to modify in our preliminary code. We have noticed the Column Generation scheme performs well given that is does not generate too many clusters. We have also seen that, in case the solution of the Master Problem is eventually fractional (which does not always occur in our tests), the solution includes plenty of zeros. This encourages us to use this solution to work on possible heuristics to solve microaggregation with lower SSE than MDAV. This actually is a line of future work on implementation explained afterwards.
Another immediate line of work is to improve the time efficiency of the current code. This does not necessarily imply to implement immediately a separation routine for the Pricing Problem. Instead we can work on heuristics not to solve the Pricing Problem to optimality but to obtain cliques with simply negative objective function, i.e., negative reduced cost for the non-approximate microaggregation. Especially at the begining of the execution there are plenty of existent clusters with a negative reduced cost and it is not absolutely necessary to obtain every time the minimal one. This type of heuristics could be very useful specially to reduce the time expense at this initial stage of the execution. In fact we might possibly call the BIP solution of the Pricing Problem only in case the heuristics fail to obtain clusters with negative objective value.

Another issue to be tested is whether the enumeration solution of the Pricing Problem for lower η is useful or not in comparison with CPLEX optimization. This might only involve some execution testing.

Finally, we must recall that after all the Column Generation our Master Problem eventually provides with a solution that might be non integer. Also provides with a lower bound on the SSE for microaggregation. Considering this lower bound and the fractional solution the Master Problem provides there are two possible strategies left for future work on the Master Problem.

(1) Think of a branching procedure to pursue the optimal solution. This might involve some difficulties. In any case this Branching for the Master Problem might include inequalities which can be simply inequalities for a typical Branch & Bound or useful inequalities that we could come up with for the Master Problem. In any case adding an inequality in the Master Problem and then solving again the whole Column Generation scheme adds a relevant difficulty. A new inequality in the Master Problem means a new dual variable or pricing coefficient for the Pricing Problem and this should introduced in the Pricing Problem objective function. The objective function in the Pricing Problem might be able to include the coefficients of the new cluster to be added in all the different inequalities added by the Branching in the Master Problem, all this with respect to the variables modelling for the Pricing Problem. It should be studied how to treat this computationally which is not trivial a priori.
(2) Think on a heuristic strategy that using the fractional result on the Master Problem after the Column Generation scheme obtains a non optimal but useful solution. A preliminary idea could be treating the coefficients in the fractional solution for the clusters considered in the Master Problem as a probability that the nodes inside this cluster are aggregated together in the eventual solution. Conflicts occur however when two different clusters contain the same pair of nodes.
The main idea is however to use the fractional coefficients as weights to consider the clusters in the eventual solution. For example we could consider the Master Problem with the clusters that have a certain minimal weight on this solution. We could solve the Master Problem as a BIP using those clusters and check the SSE of this solution. In fact, we could solve the Master Problem as a BIP using all the clusters in the final stage. At the end the Master Problem includes also the clusters from the MDAV solution (the Master Problem is initialised with them) so this BIP solution would have a SSE lower or equal to the heuristic SSE from MDAV. Actually during the whole Column Generation scheme it might occur that at some point an integer solution is found. For the same reason this integer solution is a microaggregation solution with less or equal SSE than the heuristic one, despite it might not be optimal neither. Those are just ideas to be tested in the near future.

References

[1] J. M. Abowd, J. Domingo-Ferrer, V. Torra *Using Mahalanobis distance-based record linkage for disclosure risk assessment*. Lecture Notes in Computer Science, 4302, 233-242, 2006.

[2] R. Brand, J. Domingo-Ferrer, J. M. Mateo-Sanz, *Reference data sets to test and compare sdc methods for protection of numerical microdata*. European Project IST-2000-25069 CASC, http://neon.vb.cbs.nl/casc, 2002.

[3] S. Bujalance, J. Domingo-Ferrer, A. Martínez-Ballesté, J. M. Mateo-Sanz, A. Solanas, *Métodos de microagregación para k-anonimato: privacidad en bases de datos*. Dept. Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain.

[4] T. Cristof, S. Schenker, *POlyhedron Representation Transformation Algorithm, PORTA*. http://comopt.ifi.uni-heidelberg.de/software/PORTA/index.html, 2008.

[5] J. Domingo-Ferrer, L. Franconi, S. Giessing, A. Hundepool, E. Schulte Nordholt, K. Spicer, P.-P. de Wolf, *Handbook on Statistical Disclosure Control*. Chichester, Wiley, 2012.

[6] J. Domingo-Ferrer, J. M. Mateo-Sanz, *Practical Data-Oriented Microaggregation for Statistical Disclosure Control*. IEEE Transactions on knowledge and data engineering, VOL. 14, NO. 1, January/February 2002.

[7] J. Domingo-Ferrer, A. Oganian, *On the complexity of optimal microaggregation for statistical disclosure control*. Dept. Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain.

[8] M. Grötschel, Y. Wakabayashi, *Facets of the Clique Partitioning Polytope*. Mathematical Programming, 47: 367-387, 1990.

[9] M. Grötschel, Y. Wakabayashi, *A cutting plane algorithm for a clustering problem*. Mathematical Programming, 45: 59-96, 1989.

[10] X. Ji, J. E. Mitchell, *The Clique Partitioning Problem with Minimum Clique Size Requirement*. Technical Report, Department of Mathematics Sciences, Rensselaer Polytechnic Institute, Troy, NY, USA, 2005.

[11] X. Ji, J. E. Mitchell, *Branch-and-Price-and-Cut on the Clique partitioning Problem with Minimum Clique Size Requirement*. Discrete Optimization, Vol. 4 Issue 1, 87-102, 2007.

[12] S. Lee Hansen, S. Mukherjee, *A Polynomial Algorithm for Optimal Univariate Microaggregation*. IEEE Transactions on knowledge and data engineering, VOL. 15, NO. 4, July/August 2003.

[13] A. Martínez-Ballesté, A. Solanas, *V-MDAV: Variable group size multivariate microaggregation*. CRISES Research Group. Department of Computer Engineering and Mathematics. Rovira i Virgili University. Spain.

[14] K. Muralidhar, R. Sarathy, *Data Shuffling: A New Masking Approach for Numerical Data*. Management Science, 52, 658–570, 2006.

[15] G. Sande, *Exact and approximate methods for data directed microaggregation in one or more dimensions*. International Journal of Uncertainty, Fuzziness and Knowledge-Base Systems, Vol. 10, No. 5, 459-476, 2002.

[16] L. A. Wolsey, *Integer Programming*. Wiley Series in Discrete Mathematics and Optimization, John Wiley & Sons, Inc, NY, September 1998.