FIXING PHASE IMBALANCES IN AN AWESENSE GRID: AN INTEGER PROGRAMMING APPROACH

EMILY AU, MARCO CAODURO, BOAZ ELAZAR, SYEDA ATIKA BATOOL NAQVI, AND VARUN NEELAMANA

ABSTRACT. Phase imbalances in an electric grid can lead to significant inefficiencies and power losses, underscoring the need for an effective method to reassign loads across phases. This report presents an Integer Programming (IP) model designed to address these imbalances by optimizing load distribution across three phases within a synthetic grid environment provided by Awesense. The model identifies the optimal clusters to reassign, minimizing phase imbalances while limiting the number of phase changes. The IP model is efficiently solved using Gurobi, a state-of-the-art optimization solver. Additionally, the report explores future enhancements, including the potential adoption of open-source solvers and model refinements to address global and local imbalances simultaneously.

1. Introduction

Awesense has been building a platform for digital energy with the goal of allowing easy access to and use of electrical grid data. This platform helps to build a myriad of applications and use cases for the decarbonized grid of the future, which will need to include more and more distributed energy resources (DERs) such as rooftop solar, batteries as well as electric vehicles (EVs) and still operate safely and efficiently. Awesense has built a sandbox environment populated with synthetic but realistic data and exposing APIs on top of which such applications can be built. It also comes with a web-based application (graphical user interface front-end) called TGI (True Grid Intelligence) that serves as a companion visual explorer for the data stored in the platform. An example screenshot of TGI is shown in Figure 1.

2. Problem description

Three-phase electric power is a common type of electric current used in electricity generation, transmission, and distribution. The current is split into three distinct phases, each travelling through its own power line. Consumers take power from the line, and in many cases, they take it only from one phase. Consumers can be factories, houses, house clusters, etc., as can be seen in Figure 2.

An imbalance appears if the distribution of the phases is not even, and some phases supply more energy than others. This may cause power loss, heating appliances, and fluctuations in power consumption. The average hourly consumption in a network over some months can be seen in Figure 3. Over that period, more power

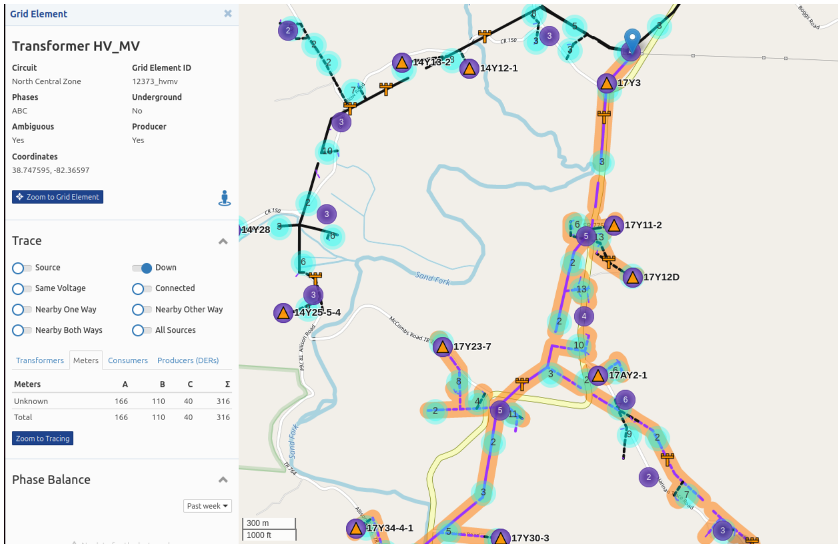

FIGURE 1. A screenshot from TGI depicting a particular section in the electrical grid.

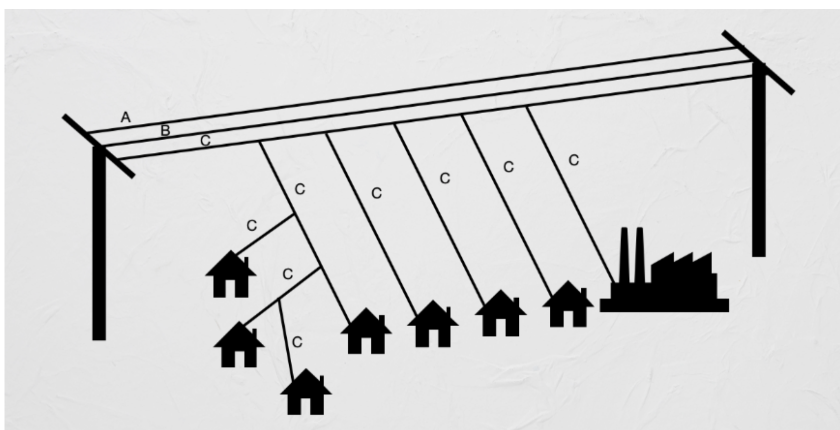

FIGURE 2. Consuming from one phase and a cluster to the left.

was taken from Phase B. The dashed line represents the average consumption over the 3 phases.

Usually, the consumers take power not directly from the ABC line but from a single-phase line attached to it, as can be seen in Figure 2 to the left. This creates clusters. As all the consumers in a cluster get the same phase, if we want to change the phase, we have to change it to the whole cluster. Changing the phase requires a physical action on the relevant electric poles. We created and improved a script that detects all the clusters in a given network, and this is usually the input to the algorithms you will see next.

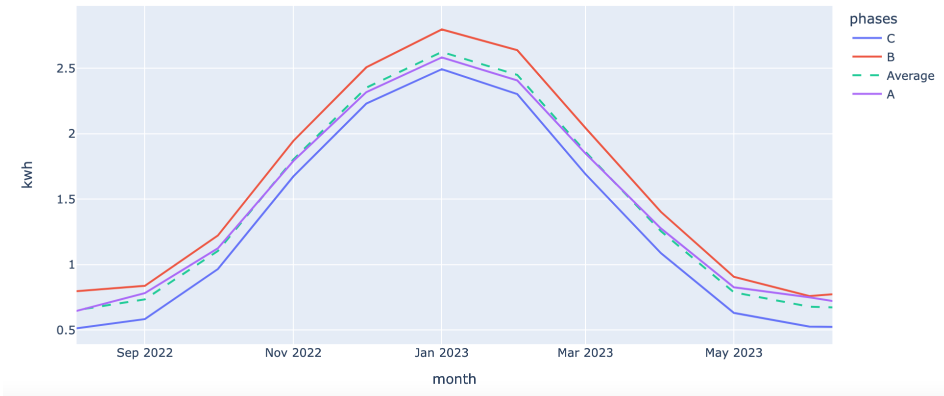

FIGURE 3. Average Hourly Consumption by Phase (w/ ABC).

The algorithm to detect these clusters is based on the following idea - Each component of the network is represented by three parameters:

- The component name
- Entry connector name
- Exit connector name

When one component's exit connector name equals a second connector's entry connector, it means the second component is connected to the first one. It is possible that several components will have the same entry connector name, which means that several different components can be connected to one predecessor. This means we have a split in the network or branching. In our case, though, we joined the tables of components to detect components with ABC phase that are connected to components with non-ABC phase (it should be noted here that a three-phase component might be connected to a two-phase component, but as those cases are less common, we decided to leave this option to future projects). Doing that will allow us to detect any branching from an ABC component to a non-ABC component, whether it is a single consumer or a cluster of consumers. Using a function given to us by Awesense, we could detect all the meters/consumers within each branch/cluster and their power consumption.

3. Methods

An Integer (linear) Program (IP) is an optimization model aimed at maximizing or minimizing a linear objective function while satisfying a set of linear constraints. In an IP, some of the decision variables are constrained to integer values. Both the objective function and constraints are represented as linear combinations of the decision variables, with each variable associated with a constant coefficient and free from exponents or nonlinear terms. This linearity is essential for preserving the problem's structure and making it solvable using specialized algorithms for IPs. We recommend the books [1] and [2] for an introduction to Integer Programming.

Given the need for discrete decisions in our problem — such as determining which clusters to adjust — we opted for an integer program model.

3.1. **Input data.** We begin by outlining the input data for our model. The first step is to define the time period under consideration, denoted as T. This period could, for instance, encompass all the hours within a year. Next, we identify a network or a specific portion of it, along with its associated clusters. We denote the set of clusters by $\mathcal{C}$. Each cluster's consumption over the defined time period is then considered. For a given cluster $c \in \mathcal{C}$ and a specific time $t \in T$, the consumption of cluster c at time t is represented by $F_{c,t}$. While $F_{c,t}$ may be derived from historical measurements, it is ideally based on predictions. Additionally, for each cluster $c \in \mathcal{C}$, we account for its initial phase $p \in P := \{A, B, C\}$. This is represented by three binary constants: $x^0_{c,p} = 1$ for the initial phase p, and $x^0_{c,p'} = 0$ for the other phases $p' \in P \setminus \{p\}$. With these inputs, the IP solver determines a set of clusters to reassign, optimizing the system's phase balance.

3.2. **IP model.** The goal of the algorithm is to determine the optimal phase to assign to each cluster. The following binary variables in the integer program represent this: For each cluster $c \in \mathcal{C}$ and phase $p \in P$, we define a variable $x_{c,p} \in \{0, 1\}$ which equals 1 if the algorithm assigns phase p to cluster c and 0 otherwise. Since each cluster can receive current only in one phase, this requirement gives rise to the first set of constraints in the IP model:

$$x_{c,A} + x_{c,B} + x_{c,C} = 1 \quad \forall\ c \in \mathcal{C} \tag{3.1}$$

To determine whether a cluster requires a phase change, we introduce a binary variable y_c for each cluster $c \in \mathcal{C}$:

$$y_c = \sum_{p \in P} (1 - x^0_{c,p}) \times x_{c,p} \quad \forall\ c \in \mathcal{C}. \tag{3.2}$$

This condition might not be immediately intuitive. The term $(1 - x^0_{c,p}) \times x_{c,p}$ is non-zero only when both $(1 - x^0_{c,p})$ and $x_{c,p}$ are non-zero. This occurs if the cluster c was not initially in phase p ($1 - x^0_{c,p}$ is non-zero) but is assigned to phase p by the algorithm ($x_{c,p}$ is non-zero). Thus, y_c equals 1 if the algorithm changes the phase of cluster c, and 0 otherwise.

As previously noted, changing the phase requires a physical operation, which can be costly. To account for this, we introduce a constraint that limits the number of phase changes the algorithm can perform:

$$\sum_{c \in \mathcal{C}} y_c \leq N_{\text{changes}}, \tag{3.3}$$

where N_{changes} is a user-defined parameter. Our experiments typically set this value to around $\frac{|\mathcal{C}|}{10}$.

We are now ready to define our objective function. In practice, phase imbalance in a grid is often measured by the maximum deviation from the average consumption across the phases. For a given time $t \in T$, this imbalance can be expressed as:

$$\text{phase-imbalance}_t = \frac{\max\left(|C_{p,t} - C_{ave,t}| \ : \ p \in P\right)}{C_{ave,t}},$$

where $C_{A,t}, C_{B,t}, C_{C,t}$ are the aggregated consumptions on phases A, B, C at time t, respectively, and $C_{ave,t}$ is their average.

While the formula to compute the phase imbalance at time t is straightforward, embedding it in an integer program requires additional work. First, we calculate the aggregated consumption for each phase at each time t based on the current configuration of x:

$$C_{p,t} = \sum_{c \in \mathcal{C}} F_{c,t} \times x_{c,p} \quad \forall\ p \in P,\ \forall\ t \in T. \tag{3.4}$$

Note that the average consumption is a constant, independent of the grid configuration, calculated simply as the total consumption divided by three:

$$C_{ave,t} = \frac{1}{3} \times \sum_{c \in \mathcal{C}} F_{c,t} \quad \forall\ t \in T. \tag{3.5}$$

The objective function of an integer program must be linear; this involves addressing two challenges: the absolute values and the max function. We handle both simultaneously by introducing the variable diff_t, which captures the maximum absolute deviation:

$$\text{diff}_t = \max\left(|C_{p,t} - C_{ave,t}| \ : \ p \in P\right) \quad \forall\ t \in T.$$

To model the above non-linear equation, we use the following linear constraints:

$$\text{diff}_t \geq C_{p,t} - C_{ave,t} \quad \forall\ p \in P,\ \forall\ t \in T, \tag{3.6}$$

$$\text{diff}_t \geq -\left(C_{p,t} - C_{ave,t}\right) \quad \forall\ p \in P,\ \forall\ t \in T. \tag{3.7}$$

With these constraints in place, we can define the IP model as follows:

$$\min \sum_{t \in T} \frac{\text{diff}_t}{C_{ave,t}}, \tag{3.8}$$

subject to the conditions (3.1)–(3.7).

3.3. **Solving the IP.** In our work, we employed Gurobi to solve the integer programming (IP) model described in the previous section. Gurobi is a state-of-the-art optimization solver renowned for its speed and robust performance. It is a popular choice in academia and industry for efficiently tackling complex optimization challenges. While Gurobi provided the computational power needed for our current study, transitioning to an open-source and free solver remains a goal for future work.

4. Results

We conducted the test on three different high-voltage (hv) transformers from two different grids, Awefice and North Central Zone. As explained in Section 2, the meters further downstream of the hv transformer were arranged in clusters. For all the tests, we looked at the hourly consumption during 2023. The future goal is to forecast the consumption for 2025 to understand which changes to make on the grid to reduce future phase imbalances. Gurobi Optimizer version 11.0.2 was used for all these tests on a CPU model: Apple M2 and the Thread count is 8 physical cores, 8 logical processors, and up to 8 threads.

4.1. **Test 1.** Grid: 'Awefice,' downstream from 'Transformer 6'. There are 5 clusters. The algorithm was run with $N_{changes} = 3$; it proposed the change reported in Table 1.

Cluster ID	Initial Phase	Changed Phase
m_21	B	C

Table 1. List of clusters with their phase changes for Test 1.

The comparison between the phase imbalances before and after the changes is displayed in Figure 4 while a histogram representing the aggregated consumption before and after the changes is in Figure 5.

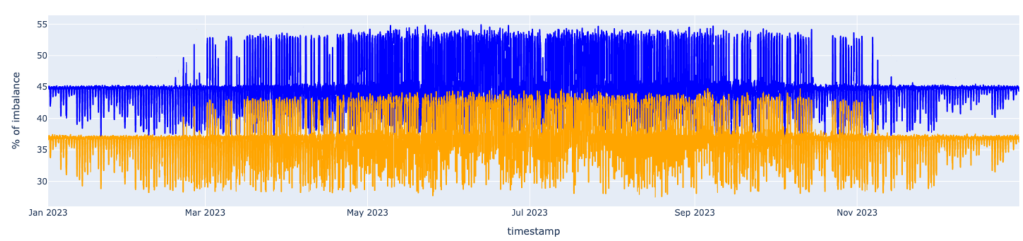

Figure 4. Before the change in blue and after the change in orange for Test 1.

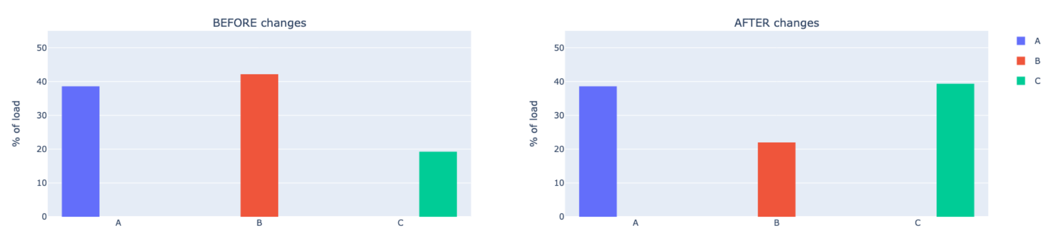

Figure 5. Aggregated consumption per phase over 2023 for Test 1.

4.2. **Test 2.** Grid: 'North Central Zone,' downstream from '12373_hvmv'. There are 40 clusters. The algorithm was run with $N_{changes} = 5$; it proposed the 5 changes reported in Table 2.

Cluster ID	Initial Phase	Changed Phase
L17-140872	C	A
L17-58026	A	C
L17-58058	C	A
L17-58268	B	A
L17-58281	A	C

TABLE 2. List of clusters with their phase changes for Test 2.

The comparison between the phase imbalances before and after the changes is displayed in Figure 6 while a histogram representing the aggregated consumption before and after the changes is in Figure 7.

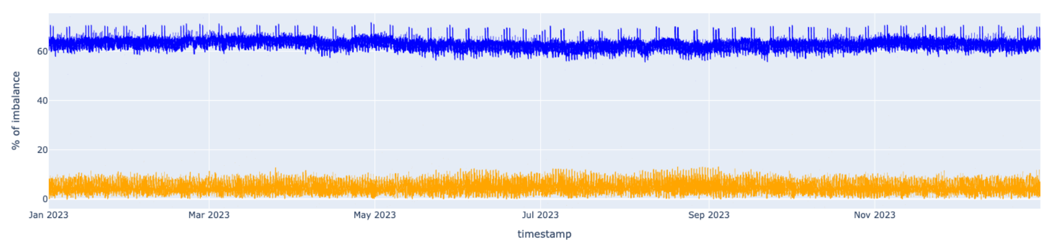

FIGURE 6. Before the change in blue and after the change in orange for Test 2.

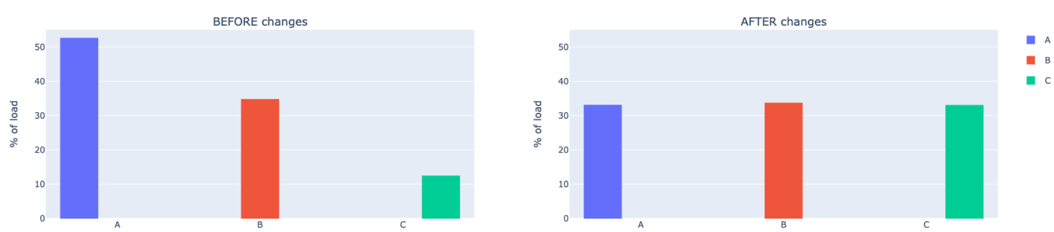

FIGURE 7. Aggregated consumption per phase over 2023 for Test 2.

4.3. **Test 3.** Grid: 'North Central Zone,' downstream from '28538_hvmv'. There are 128 clusters. The algorithm was run with $N_{changes} = 10$; it proposed the 10 changes reported in Table 3.

Cluster ID	Initial Phase	Changed Phase
13554	B	A
14Y38	C	A
46221	B	C
L14-137885	B	C
L14-138267	A	B
14-138370	B	C
L14-62934	C	A
L14-63101	C	B
L14-63752	B	A
L16-150215	C	B

TABLE 3. List of clusters with their phase changes for Test 3.

The comparison between the phase imbalances before and after the changes is displayed in Figure 8 while a histogram representing the aggregated consumption before and after the changes is in Figure 9.

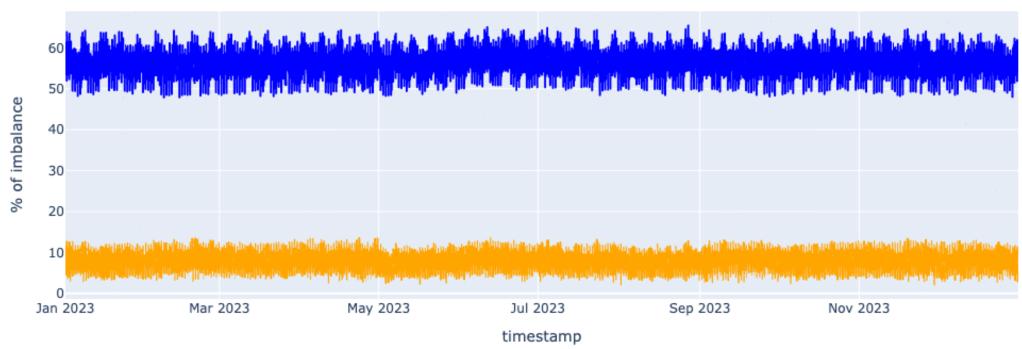

FIGURE 8. Before the change in blue and after the change in orange for Test 3.

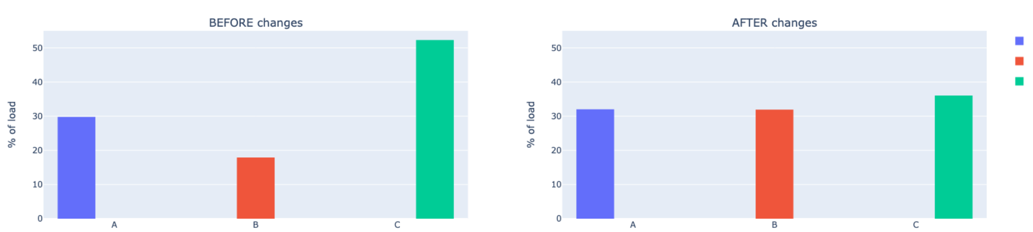

FIGURE 9. Aggregated consumption per phase over 2023 for Test 3.

5. Predictive Analysis

To enhance the solution, predictive analysis was incorporated to anticipate future load imbalances based on historical data. This predictive component allowed for proactive adjustments to the grid, ensuring ongoing balance and stability. Key efforts included using the Prophet forecasting library in Python to analyze consumption trends. For instance, in the North Central Zone grid, Cluster ID: L45-90252 was analyzed using this approach:

- Total Consumption in 2021: 1,193,688 kWh
- Total Consumption in 2022: 3,139,808 kWh
- Predicted Total Consumption for 2027: 12,870,400 kWh

Figure 10 shows the observed and predicted total consumption for Cluster L45-90252 from 2021 to 2027.

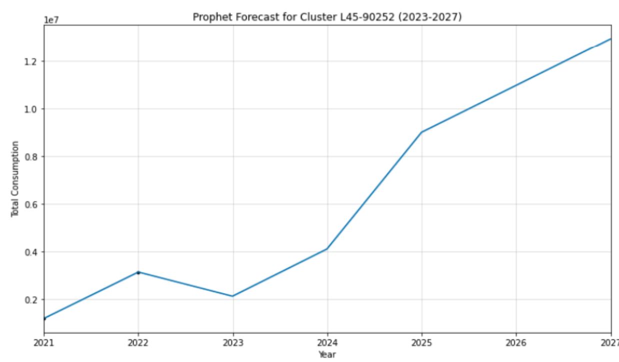

Figure 10. Total Consumption Prediction for Cluster L45-90252 (2021-2027)

6. Challenges

One of the challenges encountered is designing codes adaptable to various grid configurations. Different subgrids have different sizes and levels of complexity. It is important to generalize our code and reduce its time and space complexity to ensure accurate and efficient calculations across different electric grids. Another challenge is to ensure the forecasting model delivers accurate predictions for consumption in each phase.

7. Conclusion and Future Work

To continue our project, we aim to improve our model in four key areas.

Firstly, we aim to handle global and local imbalances simultaneously. Our initial model focused solely on global imbalance, which is the imbalance at an upper level of the electric grid. However, neglecting local imbalances can result in additional power loss (see Figure 11).

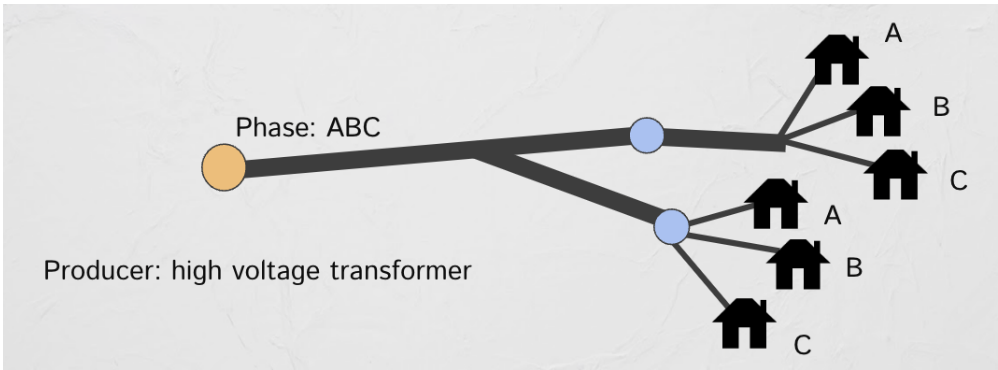

FIGURE 11. Global imbalance is assessed at the orange junction and local imbalances are assessed at the light blue junctions.

Secondly, Gurobi is used to solve the integer programming model. However, we aim to develop or identify an open-source optimizer to solve the model.

Moreover, we need to determine the optimal timing for phase changes. Since these phases requires physical actions on the electric poles, it is important to find suitable timestamps and cycles to change multiple electric poles at the same time.

Lastly, we would like to build a high-consumption model and optimize it for future uses. With expected increases in power consumption, a generalized model will simulate the power consumption in future electric grid and generate solutions for resolving phase imbalances.

Acknowledgments

We would like to express our sincere gratitude to the Pacific Institute for the Mathematical Sciences (PIMS) and the organizers of the M2PI workshop for providing the platform and support necessary for this project. Special thanks to our industry partner, Aviv Fried from Awesense, for his invaluable insights and contributions.

We would like to acknowledge the assistance of OpenAI's ChatGPT in drafting and refining parts of this report.

References

1. Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli, *Integer programming*, Graduate Texts in Mathematics, 2014.
2. Alexander Schrijver, *Theory of linear and integer programming*, Wiley-Interscience series in discrete mathematics and optimization, 1986.

E-mail address: tsznga1@ualberta.ca

E-mail address: marco.caoduro@ubc.ca

E-mail address: boazelaz@yahoo.com

E-mail address: naqvisab@myumanitoba.ca

E-mail address: vnv724@uregina.ca