Volume 38(2), pp. 203-229
http://orion.journals.ac.za

ORiON
ISSN 0259–191X (print)
ISSN 2224–0004 (online)

A column generation approach for product targeting optimisation within the banking industry

J van Niekerk* and SE Terblanche†

Received: 24 October 2022; Revised: 17 November 2022; Accepted: 17 November 2022

Abstract

Product targeting optimisation within the financial sector is becoming increasingly complex as optimisation models are being exposed to an abundance of data-driven analytics and insights generated from a host of customer interactions, statistical and machine learning models as well as new operational, business, and channel requirements. However, given the expeditious change in the data environment, it is evident that the product targeting formulation cited throughout the literature has not yet been updated to align with the realistic modeling dynamics required by financial institutions. In this paper, an enhanced product targeting formulation is proposed that incorporates a large set of new modeling constraints and input parameters to try and maximise the economic profit generated by a financial institution. The proposed formulation ensures that the correct product is offered to the desired customers at the best time of day through their preferred communication medium. To solve this formulation, a novel column generation approach is presented that is capable of reducing problem complexity and allows for significantly larger problems to be solved to global optimality within a reasonable time frame.

Key words: Product Targeting; Integer Programming; Dantzig Wolfe Decomposition; Column Generation; Optimisation; Parallel Processing.

1 Introduction

Product targeting optimisation within the banking industry is a problem that has gained significant traction throughout the last few years within the corporate ecosystem. Institutions are realising the immense financial benefit which could be unlocked by solving

*Department of Industrial Engineering, North-West University Potchefstroom, South Africa, *e-mail:* jeanpierrevn192@gmail.com

†Department of Industrial Engineering, North-West University Potchefstroom, South Africa, *e-mail:* fanie@elytica.com

http://dx.doi.org/10.5784/38-2-764

the product targeting optimisation problem. Even though it is an important problem to solve, limited research studies are available on this topic. Current exact solution approaches found throughout the literature are still not able to solve industry-sized product targeting problems to global optimality since it is considered to be an NP-hard problem [5]. Generally, researchers either reformulate the product targeting problem into a simplified version of the main problem leaving out critical business, channel, and product constraints, or apply heuristic algorithms to try and solve it [17]. The concern with these types of approaches is that sub-optimal results are obtained. To address the foregoing shortcomings a novel product targeting formulation is proposed. In addition, a column generation solution approach is presented which is capable of solving the novel formulation when applied to significantly larger product targeting problems. The novel algorithm was tested using fictitious data, which was simulated based on the characteristics of real-world problem instances.

To develop the novel product targeting strategy, the integer programming (IP) formulations of the associated problem in the literature, were augmented with complex business, operational, and channel-related constraints. Novel contributions that were added to the model formulations found in the literature, include aspects such as customer recency considerations, incorporation of channel complexity, considering the best time to call customers, and accounting for product limitations. Further aspects such as allowing customer channel preference, incorporating machine learning (ML) model output to influence model decisioning, the inclusion of marketing consent, and, lastly, accounting for product cross-sell dynamics were taken into consideration.

A novel computational framework was developed to solve the proposed product targeting problem. This computational framework utilises a novel greedy starting heuristic algorithm in conjunction with a column generation algorithm to solve significantly larger product targeting problems as compared to the normal branch-and-bound algorithm. The computational and algorithmic contributions made in the study entail the reformulation of the proposed IP formulation using both Dantzig Wolfe decomposition and column generation approaches. Parallel processing technology within the sub-problem domain is applied while incorporating dual smoothing variables to reduce the dual variable heading-in and yo-yo effects. Lastly, the methodology allows for the computation of the upper and lower bounds in order to calculate the relevant optimality and integrality gaps.

In Section 2 of this paper, a detailed literature review is provided touching on the history related to marketing strategies. In Section 3 an overview is provided of the solution methods applied by others in trying to solve the product targeting problem. Emphasis is placed on the type of modeling constraints considered in the literature and how the problem is reformulated to reduce model complexity. This review is used as a means of evaluating the advances made in the specific field of study, as well as to gain insight into the product targeting optimisation problem and how it is leveraged within the banking industry [5, 4]. The novel mathematical IP formulation proposed for solving the product targeting problem is presented in Section 4, with basic terminology and notations. The novel column generation framework is introduced in Section 4.2 with the focus on the complexities associated with combining heuristic and exact formulations in order to obtain global optimal solutions for significantly larger problem instances. Computational results generated by both the solution algorithms are discussed in Section 5. Lastly, a summarised view of the work that was conducted is represented in order to conclude the research study. To understand the complexity associated with the product targeting optimisation problem, some background regarding the technical principles on which the problem is based, is deliberated in the subsequent sections.

2 Technical background

Customers in the banking and retail industries are exposed to thousands of marketing communications on a daily basis. Organisations are competing heavily for the attention of customers to grow their clientele base and outsmart the competition. These marketing communication attempts are focused on providing information to customers about the services or products offered by the organisation. The various marketing strategies available to these organisations need to be used in full agreement to deliver an effective and holistic message to the customer that will satisfy both customer and organisational objectives. The objective of the marketing strategies is to convince customers to acquire the services or products by ensuring that each customer is receptive to the messages and that the intent to purchase is established.

Modern-day marketing techniques are based on the premise that satisfied customers will provide positive feedback to their peers and they will keep returning to make use of the organisation's offerings. It is for this reason that current marketing techniques are aimed at establishing long-term customer relationships to ultimately drive customer growth. The American Marketing Association (AMA) reformulated the definition of marketing as a function within a system or organisation that is focused on managing customer relations through developing, communicating, and providing value to customers. This definition emphasises the importance of customer relationship management within the marketing environment and how imperative it is to have a well-constructed marketing strategy aimed at meeting set out objectives [6].

Marketing strategies need to ensure that the information that is presented to the customer is relevant to the customer's current interests, needs, and situation (*i.e.*, personalised for each customer). The effect that these strategies have on customers should spur them into action to either buy a product or procure a service. There are generally two types of marketing strategies when it comes to product advertising and promotions, mass marketing and direct marketing. Mass marketing is a traditional marketing methodology that employs mass media as a way of targeting current as well as potential customers with product-related information. The channels leveraged by this type of marketing strategy include radio, television, newspapers, and magazines. It usually targets large groups of customers without discriminating between individual customers of a group. The information conveyed to these groups of customers is uniform with each customer receiving the same message. This type of marketing strategy is quite expensive in comparison to direct marketing [16].

As opposed to mass marketing, direct marketing focuses on targeting individuals or households with personalised promotions. The term direct marketing is defined by the Direct Marketing Association (DMA) as a method for a company to use data systematically in order to achieve measurable marketing objectives, by making direct contact with current or potential customers to sell individualised offers. Direct marketing classifies or clusters customers in specialised segments in order to send promotional activities or personalised advertising to these specific classes of customers [3]. Channels such as email, SMS, and voice calls have been utilised by direct marketing strategies with newer channels such as USSD, internet, and mobile push notifications being introduced in recent years [10]. Direct marketing has been receiving increasingly more traction within various industries due to its high efficiency and effectiveness in guaranteeing improved success rates on marketing campaigns. When looking at the profit margin obtained from direct marketing campaigns in recent years, it has been reported by Bose & Chen [2], that it increased from \$10 in 1999 to approximately \$12.66 in 2005 for each \$1 that was spent, with a steady increase ever since. With the introduction of the internet and other digital channels, it has significantly lowered operational costs. This means that with even low response rates on campaigns, it is frequently still sufficient in ensuring the success of the various direct marketing campaign objectives.

One of the key processes to ensure the success of direct marketing campaigns is the correct selection of target customers to which these campaigns should be distributed. Target selection is generally preceded by sophisticated customer profiling and the development of response models generated by ML algorithms [11, 12]. The data produced by the target selection process is required to be optimised to maximise economic profit and satisfy various channel, business, and product constraints before making contact with the various customers. The optimisation of direct marketing campaigns is generally achieved by implementing either exact or heuristic optimisation algorithms. These algorithms are employed to construct marketing strategies through the use of intelligent decisioning. To generate an optimal direct marketing strategy, several business, channel and product constraints need to be considered in addition to the response and clustering model outputs [17]. This includes aspects such as channel and resource limitations, channel and product exclusions, budget constraints, corporate hurdle requirements, cross-sell and upsell opportunities, campaign costs, and minimum quantity of products to sell per campaign, just to name a few [4]. The above and a multitude of other constraints need to be factored into these types of optimisation models in order to generate a direct marketing strategy, which is able to maximise a company's economic profit.

The objective of the product targeting problem is to increase the economic profit of the financial services provider. Profit maximisation can be achieved by either generating additional revenue from an existing customer base or by acquiring new customers. The former is known as "retention" whereas the latter is "acquisition". The focus of this study is justified by two aspects. Firstly, Reinartz *et, al.* [18] pointed out that firms should rather try and retain existing customers than attempting to acquire new ones. Prioritizing expenditure towards retention will result in a greater long term impact on customer profitability. Secondly, the methods and models utilised within the financial industry for data analysis are better suited for retention efforts as banks have access to the deepest veins of customer data when considering their existing customer base. Although banks have access to a significant amount of customer data, Bernstel [1] noted that many of them do not utilise these databases to their full potential when designing optimal product targeting models, which ultimately leads to sub-optimal results.

3 Related work

Nobibon *et, al.* [17] investigated and solved a simplified version of the product targeting problem. The constraints considered included aspects such as the corporate hurdle constraint, budgeting constraint, product limitations as well as curbing the number of products assigned to each customer. The authors proposed a column generation branch-and-price framework (also exploring a multitude of heuristics algorithms) capable of solving the simplified version of the product targeting problem. In their paper, problem instances of up to 10000 customers and 15 products were considered. Some conclusions that were derived from their study were that end users should use exact methods to solve small to medium-sized problems ($<$ 2000 customers and 15 products) whereas heuristics should be employed when trying to solve large product targeting problems ($>$ 10000 and 15 products).

The product targeting formulation proposed by Cohen [4] expanded slightly on the simplistic model formulation proposed by Nobibon *et, al.* [17] by adding an additional index to the optimisation problem allowing it to account for channel assignment. This addition was however very limited since it only accounted for the maximum offers which could be assigned to customers via each channel. Channels that were considered were branches, call centers, and direct mail. In this study, channel complexities such as the best time of day, probability of right party contact, the probability to answer, and a multitude of other channel-related nuances were not included. The approach followed in the paper was to aggregate customers according to their profit and cost profiles and consider a cluster or a grouping of customers as 1 data point instead of accounting for each customer individually. In doing this the input data to the problem were significantly simplified allowing the problem to be solved efficiently. The limitation of this methodology is the fact that one might not account for unique customer nuances when clustering customers into various target groups.

Delanote *et, al.* [5] accounted for the possibility of promoting bundled products to customers (cross-selling) as well as multi-channel structures. The mathematical formulation proposed was linked to the legislative framework of Belgian financial institutions. Similar to Nobibon *et, al.* [17] and Cohen [4], basic constraints, such as the corporate hurdle, campaign budgeting, and the limitation of product offers to customers, were included. Delanote *et, al.* [5] did however expand on the channel-related constraints by including aspects such as follow-up capability via multi-channels when an offer has been presented to a customer. They considered up to 15 product offers originating from 4 product categories while utilising 4 channel dimensions and 4 customer segments. The research study made use of a mixed integer programming (MIP) formulation to solve the given product targeting problem. Some important research proposals made throughout this study were to investigate the application of a Benders-base decomposition algorithm in solving the product targeting problem to global optimality or to at least provide a good approximated solution to the problem. A multitude of channel complexity constraints such as multi-number or email addresses associated with single clients and response model inputs to select the best channel option or time of day to contact a customer were not accounted for in the formulation proposed by Delanote *et, al.* [5].

Lu & Boutilier [13] proposed a new dynamic segmentation approach for linear programming product targeting problems using a modified column generation formulation as a baseline for the algorithm. Constraints related to the campaign and overall budgets were incorporated into the modeling framework, as well as, product offering guidelines and channel capacity limitations. The proposed solution was again focused on developing a customer segmentation methodology that would allow an optimisation problem to solve significantly larger product targeting problems containing millions of customers within a matter of seconds.

Savelsbergh [15] proposed a branch-and-price algorithm to solve the general assignment problem (GAP). The proposed formulation comprise of an objective function and 2 constraints. Similar research avenues were proposed by Friberg [7] where a branch-and-price algorithm was implemented to solve the opportunistic maintenance planning problem.

Work presented by Pessoa *et, al.* [14] also investigated stabilisation techniques to improve the performance of the column generation algorithm. In the mentioned research paper, techniques such as using the proximity of a stability center through implementing penalty functions were discussed. Other techniques linked to dual variable smoothing were also investigated with the last method being termed the centralized prizes approach. The work conducted by Pessoa *et, al.* [14] was however not specifically focused on the product targeting problem, but rather focused on enhancing the column generation algorithmic performance. The dual smoothing methodology was adopted in this paper to enhance the computational capability of the column generation algorithm.

4 Mathematical model formulation

When considering the modelling requirements proposed in the literature, it is imperative to note that this study is not focused on developing a segmentation methodology to allow the product targeting problem to be solved within a fraction of a few seconds. Instead, the purpose is to enhance the existing product targeting formulations to include additional complex business, operational, and channel dynamics that have not yet been accounted for in the cited mathematical formulations. A novel starting heuristic, as well as a novel column generation solution framework is proposed which is capable of solving the proposed complex formulation more efficiently compared to the standard IP formulation.

In this paper, two mathematical formulations are presented for the product targeting problem. The focus is initially on discussing the mathematical construct of the proposed IP formulation where-after the novel column generation formulation is presented. Details are provided regarding the dynamics of each formulation and the complexity associated with the respective methods. However, before discussing these mathematical formulations, some general notations are provided as an introduction.

Let $\mathcal{J}$ denote the index set of the product offers being incorporated into the product targeting problem, with $j \in \mathcal{J}$ denoting a specific product offer within the collection. The grouping of customers to which a product could be offered is denoted by the index set $\mathcal{I}$, with $i \in \mathcal{I}$ referring to a specific customer. To account for the number of channels available to contact customer i for product j, we introduce index set $\mathcal{U}$ with $u \in \mathcal{U}$ representing a unique channel within the collection of channels which could be utilised as the communication medium.

A time index set $\mathcal{T}$ is added to the product targeting problem to account for the best time $t \in \mathcal{T}$ at which a customer should be called when voice is selected as the channel of preference. Let $\mathcal{Q}$ denote the number of contact modes (*e.g.*, phone or email) linked to the collection of customers $\mathcal{I}$. The subset $\mathcal{Q}(i,u) \subseteq \mathcal{Q}$ contains the various numbers and email addresses that belong to a unique customer $i \in \mathcal{I}$.

The index set $\mathcal{C}$ contains the cross-sell options available within a given product targeting problem. The subset $\mathcal{C}(i,j) \subseteq \mathcal{C}$ delineates the cross-sell options available to each individual customer $i \in \mathcal{I}$ after being offered product $j \in \mathcal{J}$.

The last index set incorporated into the research study is only applicable to the novel column generation formulation and is given by $\mathcal{W}$. The preceding index set is representative of the total number of columns that were added to the master optimisation algorithm after a problem solution has been identified by the sub-problems. Each column added per iteration to the model is presented by $w \in \mathcal{W}$.

4.1 Novel IP formulation

The novel IP formulation objective function is represented by (1) - (5) below. In (2) variable P depicts the total monetary gain obtained by the financial institution as a result of committing products j to customers i. Variable CH is introduced into the objective function (3) to allow the model to select the best possible mobile number or email address u for a specific customer in the event where one customer might have multiple phone numbers or email addresses $q \in Q_{iu}$. Variable TM is used in the objective function (4) to allow the mathematical model to select the best time of day t to call a customer i if voice was selected as the preferred channel of contact. Lastly, variable CR is added to the objective function by incorporating (5) to account for cross-selling opportunities c which might exist for selected products j.

The monetary gain (2) obtained by the financial institution is calculated by subtracting the variable cost $c_{iju}^{(v)}$ from the potential income p_{iju}, where after it is multiplied with decision variable $x_{iju} \in$ [0,1]. Decision variable x_{iju} is utilised in the equation to govern the decisioning process as to which product j should be offered to customer i via channel u. In (2), the model also accounts for the fixed cost ($c_j^{(f)}$) associated with offering product j to customer i by incorporating parameter $c_j^{(f)}$ and decision variable $y_j \in$ [0,1]. Term r_{iju} is added to (2) to account for the probability of reaching a customer on a specific channel (*i.e.*, probability of answer). Parameter $c_{iju}^{(p)}$ in (2) enforces channel preference whereas $c_{iju}^{(s)}$ allows cross-selling opportunities to take priority over those products that do not have cross-sell options. Parameter m_1 is added to (2) to allow the assignment of an importance weighting to term P when compared to the other objective function features being considered.

Term CH is added to the product targeting objective function as seen in (3). Parameter $c_{ijuq}^{(q)}$ in (3) is the probability of right party contact (RPC) for each phone number or email address associated with a given customer i. Binary decision variable $c_{ijuq}^{(d)}$ is utilised to allow the model to select the best possible phone number or email address, where x_{iju} has been assigned a value of 1. Similar to (2), a weighting parameter m_2 is added to (3) in order to allow the end user to specify the influence of (3) on the overall objective function by assigning a constant value to m_2. Input parameter $c^{(t)}_{iuqt}$ in (4) is the probability of a customer answering at different periods of the day for a selected phone number q.

The last component that is included in the novel product targeting objective function is CR. Term $c^{(r)}_{ijc}$ in (5) is the additional financial gain that could be realized when a cross-sell option is selected. Binary decision variable o_{ijc} is multiplied with parameter $c^{(r)}_{ijc}$ in order to include the financial gain of cross-selling options to the objective function if such an option exists.

$$(M_1) \quad max \quad (P + CH + TM + CR), \tag{1}$$

$$P = \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}} \sum_{u\in\mathcal{U}} ((p_{iju} - c^{(v)}_{iju})x_{iju})(r_{iju} + c^{(p)}_{iju} + c^{(s)}_{iju})m_1 - \sum_{j\in\mathcal{J}} c^{(f)}_j y_j, \tag{2}$$

$$CH = \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}} \sum_{u\in\mathcal{U}} \sum_{q\in\mathcal{Q}_{iu}} (c^{(q)}_{ijuq} c^{(d)}_{ijuq}) m_2, \tag{3}$$

$$TM = \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}} \sum_{u=1} \sum_{q\in\mathcal{Q}_{iu}} \sum_{t\in\mathcal{T}} c^{(t)}_{iuqt} h_{jiuqt}, \tag{4}$$

$$CR = \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}} \sum_{c\in\mathcal{C}_{ij}} c^{(r)}_{ijc} o_{ijc}. \tag{5}$$

The foregoing product targeting problem is not just maximising the financial institution's monetary gain but also selecting the best conversation at the right time through the correct channel to improve customer satisfaction. The preceding is achieved by lumping (2) - (5) into the maximisation function as indicated in (1). The objective function related to the novel product targeting IP formulation is constrained using constraint sets (6) - (23) as elucidated below.

Constraint set (6), is known as the corporate hurdle, ensuring that the return on investment (ROI) for the campaigns that form part of the decisioning problem is at least R. Parameter R is calculated as $(1 + r)$ with r representing a fraction between 0 and 1 to allow for the inflation or deflation of the expected ROI by a certain margin. To account for budgeting requirements, constraint set (7) is added to the IP formulation. Parameter B_j delineates the budget linked to each product j in the campaign process. To govern the number of offers j being assigned to each customer i, constraint set (8) is added to the model. Term $(1-r^{(c)}_i)$ is introduced to constraint set (8) in order to enable the model to perform recency checks before allowing any product to be assigned to a given customer. Consideration should also be taken to limit the number of customers assigned to each product. To enforce these limitations, constraint sets (9) and (10) are added to the optimisation model enforcing the adherence to both the lower $(l^{(l)}_j)$ and upper $(l^{(u)}_j)$ bounds related to the number of customers linked to product offerings j. When considering constraint sets (6) - (10) it is apparent that these constraints are readily available throughout the product targeting literature. Some alterations were however made to the standard constraint formulations such as the addition of term u allowing the model to account for channel decisioning. The aspect of recency is also a new concept that was introduced to the overarching product targeting optimisation model in order to align the model to a more realistic representation of reality.

$$\sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} p_{iju}x_{iju} - R(\sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} c^{(v)}_{iju}x_{iju} + \sum_{j\in\mathcal{J}} c^{(f)}_j y_j) \geq 0, \tag{6}$$

$$\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} c^{(v)}_{iju}x_{iju} \leq B_j, \quad j \in \mathcal{J}, \tag{7}$$

$$\sum_{j\in\mathcal{J}}\sum_{u\in\mathcal{U}} x_{iju} \leq (1 - r^{(c)}_i), \quad i \in \mathcal{I}, \tag{8}$$

$$\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} x_{iju} \geq l^{(l)}_j y_j, \quad j \in \mathcal{J}, \tag{9}$$

$$\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} x_{iju} \leq l^{(u)}_j y_j, \quad j \in \mathcal{J}. \tag{10}$$

We introduce constraint sets (11) - (23) as novel constraints, excluding constraint set (13) which has already been cited in the literature. These constraints will allow the optimisation model to account for a multitude of new business and operational requirements which aim to align the model to real-world product targeting problems. Constraint set (11) is added into the optimisation model to limit the number of products that are allowed to be considered during the decisioning process with term $P^{(m)}$ representing a user input upper bound. As part of channel selection, constraint set (12) is added to the optimisation model in order to allow only selection of one channel u (voice ($u = 1$), SMS ($u = 2$) or email ($u = 3$)) as means of communicating offer j to customer i. The upper bound channel limiting constraint is enforced through constraint set (13) using parameter $c^{(m)}_u$. Parameter $c^{(l)}_u$ in constraint set (14) depicts the lower bound input variable which is used to manage the lead assignment to the various channels.

To enforce the upper bound limit ($l^{(m)}_{ju}$) for each product of a specific channel, constraint set (15) is introduced into the optimisation model. Parameter E_{ju} is used as an exclusion parameter, meaning that when a business unit requests that a specific product j should not be offered via a certain channel u to customer i, then the parameter will take on a value of 1. The lower bound limit for each product j being offered via a specific channel u is managed by constraint set (14) with input parameter E_{ju} being the exclusion parameter and $l^{(n)}_{ju}$ representing the lower bound input parameter. Decision variable y_j is added to constraint set (14) to only allow the offering of a product j if the model has decided to include the specific product in the solution (*i.e.*, $y_j = 1$).

The model also needs to be capable of excluding customers i from certain channels u if the customer has specified a preference towards a specific channel. This is achieved using constraint set (17) with $c^{(e)}_{iu}$ being representative of a binary input parameter used to accommodate the channel preference of customers. Parameter m_3 has been defined as a very large constant in order to ensure that when a customer has not provided a specific channel exclusion ($c^{(e)}_{iu} = 0$), the constraint will still hold true. To include channel-related marketing consent into the optimisation model, constraint sets (18) and (19) were incorporated. In constraint set (18), decision variable $c^{(d)}_{ijuq}$ keeps track of the best number or email address selection for a customer whereas input parameter $c^{(pr)}_{ijuq}$ is utilised to track marketing consent for said contact mediums. Constraint set (19) is added in order to ensure that the decisioning between variables $c^{(d)}_{ijuq}$ and x_{iju} are aligned.

When including channel as part of the decisioning process, the model is also required to decide on the best time of day to call a customer in the event that voice ($u = 1$) has been selected as the communication medium of choice. In constraint set (20), variable h_{jiuqt} is used to track the decision whether customers i should form part of the time of day selection process with no specific focus on the exact number or time to be utilised, but only aligning to x_{iju}. Constant m_4 is added to the right hand term of constraint set (20) in order to allow the model to assign a value of 1 to decision variable h_{jiuqt} where index $u = 1$ for each number q and time period t available to the interested customers.

In constraint set (21), $c^{(d)}_{ijuq}$ is used to indicate which phone number has been selected as the best number to use for a selected customer allowing variable h_{jiuqt} to only select the best time of day to call a customer for the identified number. Variable h_{jiuqt} is summated in terms of $t \in \mathcal{T}$ to ensure that only 1 time period is selected for the phone number under consideration. Decision variable o_{ijc} in constraint set (22) is used to track cross-sell option selection while $c^{(s)}_{iju}$ is a binary input parameter utilised to indicate if a cross-sell option exists for a certain product j.

$$\sum_{j\in\mathcal{J}} y_j \leq P^{(m)}, \tag{11}$$

$$\sum_{u\in\mathcal{U}} x_{iju} \leq 1, \quad j\in\mathcal{J}, i\in\mathcal{I}, \tag{12}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} x_{iju} \leq c^{(m)}_u, \quad u\in\mathcal{U}, \tag{13}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} x_{iju} \geq c^{(l)}_u, \quad u\in\mathcal{U}, \tag{14}$$

$$\sum_{i\in\mathcal{I}} x_{iju} \leq (1-E_{ju})l^{(m)}_{ju}, \quad j\in\mathcal{J}, u\in\mathcal{U}, \tag{15}$$

$$\sum_{i\in\mathcal{I}} x_{iju} \geq (1-E_{ju})l^{(n)}_{ju}y_j, \quad j\in\mathcal{J}, u\in\mathcal{U}, \tag{16}$$

$$\sum_{j\in\mathcal{J}} x_{iju} \leq m_3(1-c^{(e)}_{iu}), \quad i\in\mathcal{I}, u\in\mathcal{U}, \tag{17}$$

$$c^{(d)}_{ijuq} \leq (1-c^{(pr)}_{ijuq}), \quad j\in\mathcal{J}, i\in\mathcal{I}, u\in\mathcal{U}, q\in\mathcal{Q}_{iu}, \tag{18}$$

$$\sum_{q\in\mathcal{Q}_{iu}} c^{(d)}_{ijuq} = x_{iju}, \quad j\in\mathcal{J}, i\in\mathcal{I}, u\in\mathcal{U}, \tag{19}$$

$$\sum_{q\in\mathcal{Q}_{iu}}\sum_{t\in\mathcal{T}} h_{jiuqt} \leq m_4 x_{iju}, \quad j\in\mathcal{J}, i\in\mathcal{I}, u=1, \tag{20}$$

$$\sum_{t\in\mathcal{T}} h_{jiuqt} \leq c^{(d)}_{ijuq}, \quad j\in\mathcal{J}, i\in\mathcal{I}, u=1, q\in\mathcal{Q}_{iu}, \tag{21}$$

$$\sum_{c\in\mathcal{C}_{ij}} o_{ijc} \leq \sum_{u\in\mathcal{U};c^{(s)}_{iju}=1} c^{(s)}_{iju}x_{iju}, \quad j\in\mathcal{J}, i\in\mathcal{I}, \tag{22}$$

$$y_j, x_{ij}, h_{jiuqt}, c^{(d)}_{ijuq}, o_{ijc} \in \{0,1\}, \quad i \in \mathcal{I}, j \in \mathcal{J}, u \in \mathcal{U}, q \in \mathcal{Q}_{iu}, t \in \mathcal{T}. \tag{23}$$

The product targeting problem is considered to be NP-hard throughout the literature. Solving complexity either arise in the form of memory limitations or exponential time requirements. Adding all of the preceding complexities to an already complex product targeting problem will most definitely increase the difficulty of finding a solution to this problem when using pure exact solution algorithms. It has been noted that conventional algorithms such as the branch-and-bound algorithm are incapable of solving these complex product targeting problems for large-scale instances. It is for this reason that a column generation approach is suggested to reduce solution complexity allowing the algorithm to solve significantly larger optimisation problems within a reasonable time.

4.2 Novel column generation framework

The aim of applying the Dantzig Wolfe decomposition and Column Generation theories is to reduce the complexity of the product targeting problem and to allow optimal solutions to be computed for problem sizes where the novel IP formulation, represented in Section 4.1, fails to do so. To make use of column generation to solve the product targeting problem as defined in Section 4.1, we first need to transform the IP problem into a linear programming (LP) problem using the Dantzig Wolfe decomposition algorithm. The transformation results in a huge number of variables being generated which is attributed to the problem being written as a linear combination of its relevant extreme points and extreme rays. With the decomposed formulation containing this huge number of variables, the only way to effectively solve the reformulated optimisation problem is to make use of a column generation approach.

As part of the implementation of the column generation algorithm, the reformulated Dantzig Wolfe decomposed product targeting problem is divided into a master and multiple sub-problems. As part of the model framework, a specifically designed starting heuristic is required to generate an initial feasible solution for the initialisation of the master problem. To comprehend the complexities associated with the proposed novel product targeting column generation framework, the focus is initially set on providing detail regarding the starting heuristic algorithm where after, information is provided on the primal master problem as well as the formulation of the multiple sub-problems. Some insights are also provided into the upper bound calculation for the master problem.

4.2.1 Model starting heuristic

The starting solution which is required to initiate the column generation computational process forms an integral part of the overall product targeting solution framework. For this purpose, a greedy starting algorithm is suggested. The principle of this algorithm is to move away from the traditional optimisation techniques which require large-scale computing servers to obtain acceptable initial feasible solutions to the optimisation problem under consideration. The proposed heuristic consists of the main function which is used to manage the overall execution logic of the heuristic algorithm as well as sub-functions which is responsible for the actual inner workings and decision variable selection process of the algorithm. The outline of the heuristic algorithm is provided in both Algorithms 1 and 2 below.

Algorithm 1: Heuristic Main Function Pseudocode

Let $x^{(f)}_{piju} = 1$
Let $c^{(d)}_{ijuq} = 0$; $h_{jiuqt} = 0$; $o_{ijc} = 0$; $p^c = 120$; $obj^{(new)} = 0$; $f^f = 5$; $f^p = 2$
$funct[p^c][f^f]$ = Create 2D Matrix Constaining Number of Permutations for constraints: 4.15; 4.17; 4.21; 4.22; 4.24
$funct_1[f^p]$ = Create 1D Matrix of Constraints 4.19 and 4.14

Initialise Heuristic Function:
Send variable $x^{(f)}_{piju}$ through a series of function funnels
for $p \in p^c$ **do**
  for $f \in f^f$ **do**
    $x^{(f)}_{piju}$ = Call function $F(p, x^{(f)}_{piju}$,UB = LH of $funct[p][f]$)
  end for
  for $a \in f^p$ **do**
    $x^{(f)}_{piju}$ = Call function $F(p, x^{(f)}_{piju}$,UB = LH of $funct1[a]$)
  end for

  Ensure Corporate Hurdle is Adhered To, [4.13]:
  let $cost = 0$
  $cost+ = c^{(f)}_j$, $j \in \mathcal{J}$
  if $(((p_{iju} - (c^{(v)}_{iju} + cost)R)x^{(f)}_{piju}) \leq 0)$ **then** $(x^{(f)}_{piju} = 0)$ **end if**, $i \in \mathcal{I}, j \in \mathcal{J}, u \in \mathcal{U}$

  Determine Value for Decision Variable $y^{(heu)}$ Using simplistic Cplex Model:
  Maximise $(\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{u\in\mathcal{U}}((p_{iju} - c^{(v)}_{iju})x^{(f)}_{piju})(r_{iju} + c^{(p)}_{iju} + c^{(s)}_{iju})m_1 - \sum_{j\in\mathcal{J}} c^{(f)}_j y^{(heu)}_j)$
  Subject To Constraint 4.18
  Solve Cplex Problem and Get Value For $y^{(heu)}_j$
  Calculate New $x^{(f)}_{piju} = x^{(f)}_{piju} y^{(heu)}_j$, $i \in \mathcal{I}, j \in \mathcal{J}, u \in \mathcal{U}$
  Let $obj = 0$
  $obj = \sum_{j\in\mathcal{J}} -c^{(f)}_j y^{(heu)}_j$
  $obj = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{u\in\mathcal{U}}((p_{iju} - c^{(v)}_{iju})x^{(f)}_{piju})(r_{iju} + c^{(p)}_{iju} + c^{(s)}_{iju})m_1$

  Calculate Final x_{iju} and y_j Values:
  If $(obj \geq obj^{(new)})$ **Then** $(obj^{(new)} = obj$; $y_j = y^{(heu)}_j$; $x_{iju} = x^{(f)}_{piju})$ **end if**
end for
Provide Master Problem with x_{iju}, y_j, $c^{(d)}_{ijuq}$, h_{jiuqt} and o_{ijc} as starting solution

The main heuristic is used to initialise variable $x^{(f)}_{piju}$. This variable will be used to track the heuristic algorithm selection and update the decision variable x_{iju} as it moves through the various stages of computation. The first part of Algorithm 1 initialises the various decision variables $c^{(d)}_{ijuq}$, h_{jiuqt} and o_{ijc}, as well as variables, used to track matrix indices such as $p^{(c)}$, f^f and f^P. Note that in this starting heuristic algorithm we do not compute values for decision variables $c^{(d)}_{ijuq}$, h_{jiuqt} and o_{ijc}. The preceding decision variables are assigned zero values as a starting point for the column generation algorithm. The focus is rather on calculating starting values for decision variables x_{iju} and y_j as these variables are critical to the initialisation of the column generation algorithm. In Algorithm 1 various permutations for constraint sets (8), (10), (13), (15) and (17) are considered as well as the inclusion of constraint sets (12) and (7) to compute a starting solution for decision variable x_{iju}. An initial solution for y_j is calculated in the main heuristic by making use of (2) as the maximisation objective and (11) as constraint set.

Algorithm 2: Heuristic Sub Function Pseudocode: $F(p, x^{(f)}_{piju}, UB)$

Calculate Objective Value Using $x^{(f)}_{piju}$
Let $x^{(ft)}_{iju} = 0$; $val_{iju} = 0$; $UB = 0$; $UL = 0$; $counter = 0$
Calculate objective function: $val_{iju} = ((p_{iju} - c^{(v)}_{iju})x^{(f)}_{piju})m_1(r_{iju} + c^{(p)}_{iju} + c^{(s)}_{iju}) \quad i \in \mathcal{I}, j \in \mathcal{J}, u \in \mathcal{U}$

Sort Objective Value and Matrix Index In Descending Order
loop over the iterated index/indices of the constraint associated to UB **then do**
    vector<pair<int,int> >vp
    loop over the summated index/indices of the constraint associated to UB **then do**
        vp.push_back(make_pair(val_{iju},index))
    end loop
    sort(vp.begin(),vp.end(),sortinrev)
    loop over the summated index/indices of the constraint associated to UB **then do**
        $val_{iju} = vp$.first
        $val^{ind}_{iju} = vp$.second
    end loop
end loop

Compute Temporary $x^{(ft)}_{iju}$ Variable Until Upper Bound for Constraint Is Satisfied
loop over the iterated index/indices of the constraint associated to UB **then do**
    UB = LHS of Constraint
    loop over the summated index/indices of the constraint associated to UB **then do**
        $UL = UB - counter$
        if $(UL > 0)$ **then**
            ($x^{(ft)}_{iju} = 1$ for index val^{ind}_{iju})
            $counter$ += 1
        end if
    end loop
end loop

Update Variable $x^{(f)}_{piju}$
Update: $x^{(f)}_{piju} = x^{(ft)}_{iju} x^{(f)}_{piju} \quad i \in \mathcal{I}, j \in \mathcal{J}, u \in \mathcal{U}$
return $x^{(f)}_{piju}$

The sub-function pseudocode listed in Algorithm 2 is responsible for the sorting and selection of the variable $x^{(f)}_{piju}$ in order to determine a good combination of binary variables that would lead to a feasible solution for the product targeting problem. When the execution of the heuristic sub-function is completed, the main algorithm is provided with a proposed value for $x^{(f)}_{piju}$.

Throughout the execution of the starting heuristic, only 5 of the 8 constraints contained in the column generation master problem (Section 4.2.2) are considered, but a feasible starting solution is still obtained. The reason why not all 8 constraints are incorporated is because of the greedy algorithmic approach which was implemented. The 3 constraints that are excluded from the heuristic model are lower bound constraints. With a greedy approach, the algorithm tries to satisfy the maximum value of each constraint by assigning binary values to x_{iju}. By trying to meet the maximum criteria for each constraint, it will inherently also ensure that the proposed binary combination selected for x_{iju} will also satisfy the lower bound requirements. Therefore, leveraging the above phenomenon reduces the complexity and computational time of the starting heuristic algorithm while still ensuring a feasible initial solution is provided to the column generation method. The

Dantzig Wolfe decomposition and column generation algorithms are employed as the next step to reach global optimality. The foregoing mathematical framework is discussed in detail throughout Section 4.2.2 with the initial focus on the primal master formulation for the product targeting optimisation problem.

4.2.2 Primal Master Problem

To transform the novel product targeting IP formulation into a standard form that will support the column generation algorithm, the baseline optimisation model M_1 is reformulated using the Dantzig Wolfe Decomposition algorithm for integer programming problems.

The primal objective function takes on a similar mathematical form as the original problem formulation in Section 4.1 with some minor transformations in order to align with the Dantzig Wolfe Decomposition framework. To reformulate the objective function of Section 4.1 according to the structure of the mathematical formulation (24) - (28), the objective function is expressed as a convex combination of the extreme points for the various decision variables x_{iju}, $c_{ijuq}^{(d)}$, h_{jiuqt} and o_{ijc}. To achieve this, index w is added and the variables are transformed in order to take on predefined binary values (constants) which were either obtained from the starting heuristic solution (start of column generation algorithm) or from values generated by the sub-problem as described in Section 4.2.3. The decision variable z_{wj} is added to the problem formulation in order to track the real number weighting assigned by the optimisation problem to each extreme point combination. In order to solve the product targeting problem using column generation, we only consider a reduced master problem at the start of the column generation process with the initalisation point coinciding with the starting solution obtained from the novel heuristic algorithm. The column generation master problem objective function is represented by MP_P.

The variables utilised in the objective function (24) are identical to those considered in (1) with P_{wj} representing the monetary gain obtained from the product targeting process, CH_{wj} guiding the model to perform channel selection, TM_{wj} allowing the model to select the best time of day to call a customer and lastly, CR_{wj} which allows the model to account for cross-selling opportunities. The only difference in (24) when compared to (1) is the addition of index $w \in \mathcal{W}$ and decision variable z_{wj}.

Constraints (25) - (28) are lumped into the maximisation function (24), to guide the optimisation model in maximising the financial institution's monetary gain and allowing the selection of the best conversation at the right time through the correct channel in order to improve customer satisfaction. The decisioning performed in the master problem objective function is governed by the linking constraints defined within the product targeting problem discussed in Section 4.1. An overview of the mentioned constraints is provided in Section 4.2.2 with a detailed discussion on how the constraints had to be altered to adhere to the column generation framework.

$$(MP_P) \quad max \quad \sum_{w\in\mathcal{W}}\sum_{j\in\mathcal{J}} z_{wj}(P_{wj}+CH_{wj}+TM_{wj}+CR_{wj}) \quad (=Z), \tag{24}$$

$$P_{wj}=\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}((p_{iju}-c^{(v)}_{iju})x_{wiju})(r_{iju}+c^{(p)}_{iju}+c^{(s)}_{iju})m_1-\sum_{j\in\mathcal{J}}c^{(f)}_j y_j, w\in\mathcal{W}, j\in\mathcal{J}, \tag{25}$$

$$CH_{wj}=\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}_{iu}}(c^{(q)}_{ijuq}c^{(d)}_{wijuq})m_2, \quad w\in\mathcal{W}, j\in\mathcal{J}, \tag{26}$$

$$TM_{wj}=\sum_{i\in\mathcal{I}}\sum_{u=1}\sum_{q\in\mathcal{Q}_{iu}}\sum_{t\in\mathcal{T}}c^{(t)}_{iuqt}h_{wjiuqt}, \quad w\in\mathcal{W}, j\in\mathcal{J}, \tag{27}$$

$$CR_{wj}=\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{C}_{ij}}c^{(r)}_{ijc}o_{wijc}, \quad w\in\mathcal{W}, j\in\mathcal{J}. \tag{28}$$

In order to extract the constraint set for the column generation master problem, the linking constraints are identified from the list of constraints in Section 4.1. The criterion which is used to determine if a constraint falls within the linking constraint category is to identify if the constraint could be split up per product index j. If the relevant constraint requires a full view of the entire product list in order to maintain its original purpose (*i.e.*, the constraint requires a summation of variables across the product offerings) it would mean that the constraint can't be split up per product index j and therefore it will be classified as a linking constraint. It is however imperative to note that the preceding theory does not apply to constraint sets (15) - (16) and some modeling adjustments had to be made in order to include the mentioned constraints into the master problem formulation. The purpose of constraint sets (29) - (32) is similar to the original constraints in Section 4.1, but with the exception that column generation decision variable z_{wj} as well as index $w\in\mathcal{W}$ are included.

$$\sum_{w\in\mathcal{W}}[\sum_{j\in\mathcal{J}}z_{wj}(\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}p_{iju}x_{wiju}-R(\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}c^{(v)}_{iju}x_{wiju}+c^{(f)}_j y_j))]\geq 0, \tag{29}$$

$$\sum_{w\in\mathcal{W}}\sum_{j\in\mathcal{J}}z_{wj}(\sum_{u\in\mathcal{U}}x_{wiju})\leq(1-r^{(c)}_i), \quad i\in\mathcal{I}, \tag{30}$$

$$\sum_{w\in\mathcal{W}}\sum_{j\in\mathcal{J}}z_{wj}(\sum_{i\in\mathcal{I}}x_{wiju})\geq c^{(l)}_u, \quad u\in\mathcal{U}, \tag{31}$$

$$\sum_{w\in\mathcal{W}}\sum_{j\in\mathcal{J}}z_{wj}(\sum_{i\in\mathcal{I}}x_{wiju})\leq c^{(m)}_u, \quad u\in\mathcal{U}. \tag{32}$$

Constraint sets (15) - (16) do not fit the standard criteria used to identify constraints that should enter the master problem formulation. These constraints do not require the summation of variables across product indices j and are used for iterating through each product j to ensure that the upper and lower limits are respected. In order to add the preceding constraints into the master problem, constraint sets (15) - (16) are rewritten to be in the same form as constraint sets (33) - (34). Note however that constraint sets (33) - (34) may also have been excluded from the master problem and added to the sub-problem formulation in Section 4.2.3, however, to reduce the complexity of the sub-problems, it was decided to incorporate the said constraints into the master problem formulation instead. Given that the branch-and-bound algorithm is used to solve the sub-problem, we want to limit the complexity to prevent computational memory limitations.

$$\sum_{w\in\mathcal{W}} z_{wj}(\sum_{i\in\mathcal{I}} x_{wiju} - (1-E_{ju})l_{ju}^{(m)}) \leq 0, \quad j\in\mathcal{J}, u\in\mathcal{U}, \tag{33}$$

$$\sum_{w\in\mathcal{W}} z_{wj}(\sum_{i\in\mathcal{I}} x_{wiju} - (1-E_{ju})l_{ju}^{(n)}y_j) \geq 0, \quad j\in\mathcal{J}, u\in\mathcal{U}. \tag{34}$$

Changes involved in formulating constraint sets (17) and (11) included the addition of index w, z_{wj} and updating decision variable x_{iju} to x_{wiju}. The purpose of constraint sets (35) - (36) within the column generation context remained similar to what was discussed for the original model.

$$\sum_{w\in\mathcal{W}}\sum_{j\in\mathcal{J}} z_{wj}x_{wiju} \leq m_3(1-c_{iu}^{(e)}), \quad i\in\mathcal{I}, u\in\mathcal{U}, \tag{35}$$

$$\sum_{w\in\mathcal{W}}\sum_{j\in\mathcal{J}} z_{wj} \leq P^{(m)}. \tag{36}$$

As part of the Dantzig Wolfe Decomposition reformulation, it is required that constraint sets (37) and (38) are added to the column generation master problem to maintain model sanity. The purpose of constraint set (37) is to ensure that the fractional values assigned to z_{wj} within the master optimisation model do not exceed an overall value of 1 per product index j. In constraint set (37) a less than or equal to sign is used in the constraint instead of a strict inequality as dictated by the Dantzig Wolfe Decomposition Algorithm. This is to allow the column generation algorithm to assign fractional values to z_{wj} during the start of the computational process. As the algorithm moves through the computations, the values of z_{wj} will start to converge to a value of 1. At the point where the optimisation model terminates at a global optimal solution, z_{wj} will have been assigned a value of either 0 or 1 for each given index j.

$$\sum_{w\in\mathcal{W}} z_{wj} \leq 1 \quad j\in\mathcal{J}. \tag{37}$$

Lastly, constraint set (38) is added to the master problem to allow the optimisation model to assign real values to z_{wj} as per the requirements of the simplex algorithm used to solve the column generation master problem.

$$0 \leq z_{wj} \leq 1, \quad w\in\mathcal{W}, j\in\mathcal{J}. \tag{38}$$

By utilising the simplex method as a solution algorithm, the dual values and the reduced cost related to the primal master problem can be extracted so that the associated sub-problem objective functions can be formulated.

4.2.3 Sub-Problem

The sub-problem, also known as the pricing problem, which is used within the column generation algorithm, consists of an objective function that is derived from the dual master problem reduced cost as well as model constraints that coincide with the complicating constraints identified from the baseline model from Section 4.1. Given the structure of the product targeting column generation sub-problem, the overarching integer programming problem can be subdivided into multiple smaller integer programming problems according to index $j \in \mathcal{J}$. This allows the use of parallel processing to speed up solution time. The reduced cost is lumped into a minimisation function to complete the derivation of the objective function (45).

$$R_1 = [p_{iju}(d^{(a)} - (r_{iju} + c^{(p)}_{iju})m_1) - c^{(v)}_{iju}(Rd^{(a)} + (r_{iju} + c^{(p)}_{iju} + c^{(s)}_{iju})m_1)], \tag{39}$$

$$R_2 = \sum_{j\in\mathcal{J}}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}} x_{iju}[R_1 + [d^{(b)}_i + d^{(c)}_u + d^{(d)}_u + d^{(e)}_{ju} + d^{(f)}_{ju} + d^{(g)}_{iu}]], \tag{40}$$

$$R_3 = \sum_{j\in\mathcal{J}}[c^{(f)}_j y_j(1 - Rd^{(a)}) + d^{(h)} + d^{(k)}_j], \tag{41}$$

$$R_4 = \sum_{j\in\mathcal{J}}\sum_{u\in\mathcal{U}}[E_{ju}l^{(m)}_{ju}d^{(e)}_{ju} + E_{ju}l^{(n)}_{ju}y_j d^{(f)}_{ju} - 2], \tag{42}$$

$$R_5 = -\sum_{j\in\mathcal{J}}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}_{iu}}(c^{(q)}_{ijuq}c^{(d)}_{ijuq})m_2 - \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}}\sum_{u=1}\sum_{q\in\mathcal{Q}_{iu}}\sum_{t\in\mathcal{T}} c^{(t)}_{iuqt}h_{jiuqt}, \tag{43}$$

$$R^C = R_2 + R_3 + R_4 + R_5 - \sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}}\sum_{c\in\mathcal{C}_{ij}} c^{(r)}_{ijc}o_{ijc}, \tag{44}$$

$$(SP) \quad min \quad (R^C). \tag{45}$$

When the primal master problem is a maximisation problem as seen in Section 4.2.2, the associated sub-problem will take on the form of a minimisation problem. Note that $d^{(a)}$ - $d^{(k)}$ is representative of the dual variables which were derived from the master problem dual formulation. Parameter $d^{(a)}$ maps to constraint set (29) with the rest of the dual variables mapping to each of the successive constraint sets seen in the master problem (Section 4.2.2).

The column generation sub-problem constraints are represented by constraint sets (46) - (55). Note that these constraints are extracted from the baseline optimisation model in Section 4.1 (from the list of complicating constraints) with no alterations being made to said constraints to incorporate them into the column generation sub-problem framework. The only change required was to pair constraint sets (46) - (55) to the correct objective function (45) to generate the required sub-problem outputs. Note that the purpose of these constraints within the sub-problem also remains identical to what was discussed in Section 4.1 and therefore no additional information is provided.

$$\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} c^{(v)}_{ij}x_{iju} \le B_j, \quad j \in \mathcal{J}, \tag{46}$$

$$\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} x_{iju} \geq l_j^{(l)} y_j, \quad j \in \mathcal{J}, \tag{47}$$

$$\sum_{i\in\mathcal{I}}\sum_{u\in\mathcal{U}} x_{iju} \leq l_j^{(u)} y_j, \quad j \in \mathcal{J}, \tag{48}$$

$$c_{ijuq}^{(d)} \leq (1 - c_{ijuq}^{(pr)}), \quad j \in \mathcal{J}, i \in \mathcal{I}, u \in \mathcal{U}, q \in \mathcal{Q}_{iu}, \tag{49}$$

$$\sum_{u\in\mathcal{U}} x_{iju} \leq 1, \quad j \in \mathcal{J}, i \in \mathcal{I}, \tag{50}$$

$$\sum_{q\in\mathcal{Q}_{iu}} c_{ijuq}^{(d)} = x_{iju}, \quad j \in \mathcal{J}, i \in \mathcal{I}, u \in \mathcal{U}, \tag{51}$$

$$\sum_{q\in\mathcal{Q}_{iu}}\sum_{t\in\mathcal{T}} h_{jiuqt} \leq m_4 x_{iju}, \quad j \in \mathcal{J}, i \in \mathcal{I}, u = 1, \tag{52}$$

$$\sum_{t\in\mathcal{T}} h_{jiuqt} \leq c_{ijuq}^{(d)}, \quad j \in \mathcal{J}, i \in \mathcal{I}, u = 1, q \in \mathcal{Q}, \tag{53}$$

$$\sum_{c\in\mathcal{C}_{ij}} o_{ijc} \leq \sum_{u\in\mathcal{U}, c_{iju}^{(s)}=1} c_{iju}^{(s)} x_{iju}, \quad j \in \mathcal{J}, i \in \mathcal{I}, \tag{54}$$

$$x_{iju}, h_{jiuqt}, c_{ijuq}^{(d)}, o_{ijc} \in \{0, 1\}, \quad j \in \mathcal{J}, i \in \mathcal{I}, u \in \mathcal{U}, q \in \mathcal{Q}_{iu}, t \in \mathcal{T}, c \in \mathcal{C}_{ij}. \tag{55}$$

The purpose of the sub-problems are to determine the correct binary combination for decision variables x_{iju}, h_{jiuqt}, $c_{ijuq}^{(d)}$ and o_{ijc} that will decrease the reduced cost objective function value and in turn improve the master objective value. When an optimal binary combination is obtained while solving the sub-problem, it will be fed to the master problem variables x_{wiju}, h_{wjiuqt}, $c_{wijuq}^{(d)}$ and o_{wijc} as seen in Section 4.2.2 and be appended onto the master problem as additional columns. New columns will be appended to the master problem until the algorithm reaches its termination point. As mentioned in Section 4.2.2, the researcher used gap calculations to determine the point of termination for the column generation algorithm. To compute the various gap metrics, the algorithm first needs to calculate an upper bound for the master problem. In Section 4.2.4, the derivation of such an upper bound is provided.

4.2.4 Upper Bound Formulation

In order to compute the upper bound for the master problem, the weak duality theorem is used as a point of departure.

Theorem 1 *Consider a primal problem of the form:*

$$\textit{maximize} \quad c^T x \quad \textit{subject to} \quad Ax \leq b, \quad x \geq 0. \tag{56}$$

With the dual problem derived from the primal problem taking on the form:

$$\textit{maximize} \quad b^T y \quad \textit{subject to} \quad A^T y \geq c, \quad y \geq 0. \tag{57}$$

If $(x_1, x_2 ... x_n)$ is taken as a feasible solution to the primal maximization linear program and $(y_1, y_2 ... y_n)$ is assumed to be a feasible solution to the dual minimization linear problem, then the weak duality theorem can be stated as follow:

$$c^T x = x^T c \leq x^T A^T y \leq b^T y, \tag{58}$$

$$\sum_{j \in \mathcal{J}} c_j x_j \leq \sum_{i \in \mathcal{I}} b_i y_i. \tag{59}$$

where c_j and b_i refers to the coefficients of the respective objective functions. The above statement alludes to the fact that the objective function value for the dual optimisation problem should be either greater than or equal to the primal objective function value when computed by the solution algorithm. Simplifying the preceding function the subsequent equation is obtained:

$$\sum_{j \in \mathcal{J}} c_j x_j - \sum_{i \in \mathcal{I}} b_i y_i \leq 0. \tag{60}$$

4.2.5 Dealing with Heading-In and Yo-Yo Phenomenon

When working with column generation algorithms, aspects such as dual variable heading-in, yo-yo, and the tailing-off phenomenon could have a detrimental effect on the computational performance of the column generation algorithm [14, 19]. To reduce the initial heading-in effect experienced by the column generation algorithm, we introduce (61) into the sub-problem objective function. A large constant value (*const*) is multiplied by a dynamic fraction value $(1 - a)$ with a being initialised at a value of 0. As the algorithm progresses through the iterations, a is incremented by 0.05 to reduce the effect that (61) has on the sub-problem computations. The incremental value of 0.05 can be reduced or increased depending on the desired rate at which the end user would want (61) to influence the sub-problem objective function. After completing a certain amount of iterations, the value of a will equate to 1 (as per the incremental increases made to a) leaving the entire term $((1 - a)(const))$ to take on a value of 0. With (61) taking on a value of 0, it will no longer affect the column generation algorithm. The purpose of (61) is to try and force the algorithm out of the initial heading-in effect to allow the creation of sensible columns. After reducing the influence of the heading-in effect using (61), the algorithm moves towards the dual value yo-yo phenomenon.

$$(1 - a)(const). \tag{61}$$

The yo-yo phenomenon is caused by the sporadic changes in the dual values which are calculated from the primal master problem. A sudden abrupt change in the dual values could result in the column generation algorithm frequently changing the direction in which the solution has been steered and as a result, the model could struggle to find appropriate columns which could improve the master problem objective function. To try and counteract the preceding, we introduce a dual smoothing algorithm (dual averaging) into the sub-problem objective function to try and dampen the influence of rapidly changing dual values. The proposed solution allows for changes in the dual values, however, it is managed in a more controlled manner to allow the algorithm to generate better columns much faster. At the start of the column generation algorithm, variable $d_{ave}^{(a-k)}$ is created for each of the dual values present in the sub-problem objective function. Variables $d_{ave}^{(a-k)}$ will take on the values of the initial iteration dual values $d^{(a-k)}$ as seen in (62).

$$d_{ave}^{(a-k)} = d^{(a-k)}. \tag{62}$$

As the algorithm moves to the next iterations, the column generation method starts to leverage (63) to calculate the average dual values across the various iterations. In (63), $d_{ave}^{(a-k)}$ is updated with the average value between $d_{ave}^{(a-k)}$ and $d^{(a-k)}$.

$$d_{ave}^{(a-k)} = (d_{ave}^{(a-k)} + d^{(a-k)})/2. \tag{63}$$

After computing the average dual values for a given point in time, we formulate an equation (64) which will take into account both the current as well as the average dual values when determining the next column to enter the master problem. The model will consider the actual and average dual values in a fractional relationship to one another. The preceding is enforced by introducing fractional variable a_1 into (64). At the start of the column generation algorithm variable a_1 will be initialised at an arbitrary value of say 0.8 (a_1 can be initialised with any value between 0 and 1). The preceding would mean that when the sub-problem considers the various dual values within the objective function, the influence of the actual dual values would be weighted by 0.8 on the objective function calculations whereas the average dual values would be weighted by 0.2. Variable a_1 would maintain a constant value of 0.8 throughout the column generation iterations. Only when the integrality gap reaches a point less than 1.1 will the value of a_1 be updated to 1. When a_1 takes on a value of 1, it will result in the dual average effect induced by $d_{ave}^{(a-k)}$ to be excluded from the model. With the exclusion of $d_{ave}^{(a-k)}$, the sub-problem objective function would only be influenced by $d^{(a-k)}$. Note that the integrality gap (I^G) will only reach a point below 1.1 close to the end of the algorithm (small optimality gap) where we would not want the dual averages to induce biases on the columns being generated and for this reason, a_1 is then assigned a value of 1. However, before reaching that cut-off point of $I^G \leq 1.1$, the dual averaging technique in (64) assists with reducing the yo-yo effect within the column generation algorithm [14].

$$d_f^{(a-k)} = d^{(a-k)}(a_1) + d_{ave}^{(a-k)}(1 - a_1). \tag{64}$$

By substituting each of the dual values seen in the reduced cost function (Section 4.2.3) with it's dual smoothing equivalent calculated in (64), the computational efficiency of the column generation sub-problem algorithm is significantly improved. The foregoing dual smoothing methodology allows the column generation algorithm to reduce both computational time and memory requirements for the column generation algorithm.

5 Model results and interpretation

In the subsequent section, some comparative results are provided regarding the performance of the IP formulation versus the column generation algorithm when applied to product targeting optimisation problems. The solution framework was implemented using IBM Cplex [9], C++ and Visual Studio as a development environment, and Python for data generation and structuring. Table 1 denotes the size variations of product targeting problems that were used in the empirical tests. Both the IP formulation as well as the column generation algorithm were used to solve test instances 1 - 9 whereas for test instances 11 - 16, only the column generation algorithm was applied given the computational limitations identified for the IP formulation.

The branch-and-bound algorithm is designed in a way that requires the model to fit every combination and permutation encompassed within the optimisation problem into memory resulting in the algorithm easily exceeding the server's memory capabilities when trying to solve instances larger than test case 9. Hence, only the column generation algorithm was applied to test instances exceeding the size of problem 9 due to its superior capability in solving larger optimisation problems. No results were provided for test instance 17 given that both algorithms failed to generate any kind of solution for the problem instance due to its size and complexity. The results obtained for the aforementioned test cases were evaluated based on each algorithm's memory utilisation capability, the ability of each algorithm to reach global optimality as well as the number of columns and rows that were generated for each test instance.

Test Case	Model Type	Num Cust	Num Prod	Num Chan	Constrained
Test1	IP/DM	500	5	3	Tight
Test3	IP/DM	1500	10	3	Tight
Test5	IP/DM	3000	15	3	Tight
Test7	IP/DM	5000	20	3	Tight
Test9	IP/DM	8000	25	3	Tight
Test11	DM	10000	30	3	Tight
Test13	DM	15000	30	3	Tight
Test15	DM	20000	35	3	Tight
Test16	DM	25000	35	3	Tight
Test17	DM	30000	35	3	Tight

Table 1: IP Formulation and Column Generation Model Inputs.

5.1 Memory utilisation

Figure 1 provides a summarised view of the memory consumed by both the IP formulation as well as the column generation algorithm when applied to solving test instances 1-16. It is apparent that the IP formulation operated close to the server's memory limits (7GB) during the computation of the solution for test case 9. The IP formulation was however able to operate well below the memory limits for test cases 1 - 7 without running into any memory constraints. Consequently, the IP formulation could easily compute the desired global optimal solutions for the mentioned test instances. From the analysis, it is evident that the IP formulation follows an exponential memory consumption trend when it comes to scaling the product targeting optimisation problems to be solved. The memory utilisation profile of the column generation algorithm is also portrayed in Figure 1. When evaluating the column generation memory profile trend, it is clear that the algorithm only reached the server's memory limits during the execution of test instance 16. The column generation algorithm was however able to solve test instances 1 - 15 with ease while obtaining global optimal solutions for each.

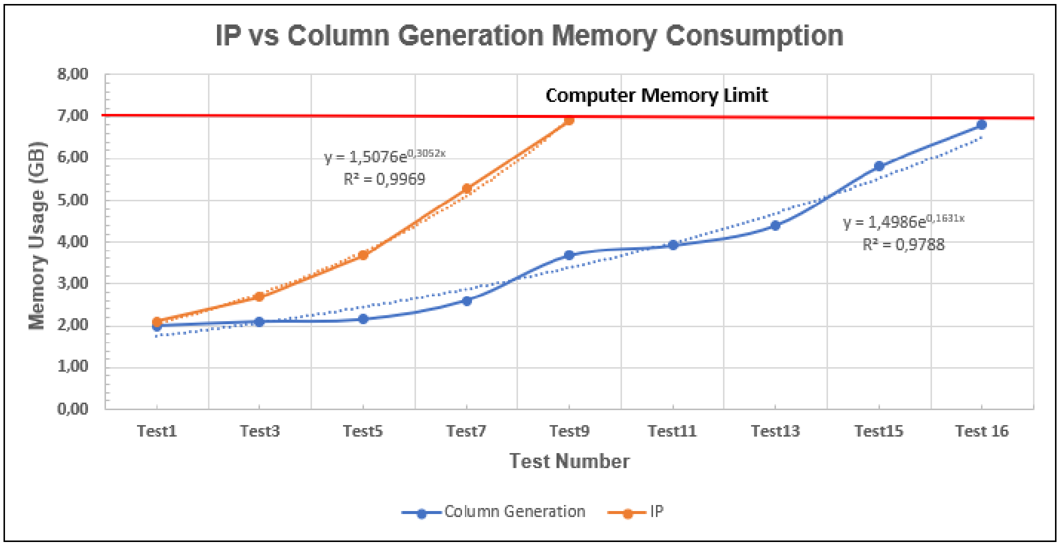

Figure 1: IP vs Column Generation Memory Consumption

When comparing the memory profile of the column generation algorithm to the IP formulation, it is evident that there is a significant difference between the two curves. By implementing the column generation algorithm, a large drop in memory consumption is achieved, allowing the algorithm to solve significantly larger product targeting optimisation problems. For example, for test instance 5 the IP formulation consumed approximately 4 GB of memory whereas the column generation algorithm only consumed 2 GB of memory. During test instance 7, 5.2 GB of memory was being used by the IP formulation whereas the memory requirements for the column generation algorithm were around the 2.7 GB mark. The preceding are just two of the examples noted in Figure 1, with the remainder examples following the same trend.

From the foregoing analysis, it can be concluded that the column generation algorithm significantly outperforms the IP formulation when it comes to server memory utilisation for all tested problem size ranges.

5.1.1 Optimality Gap

Figure 2 shows the optimality gaps computed for each of the tightly constrained test cases while leveraging both the IP formulation as well as the column generation methods as solution algorithms. For the tightly constrained test instances 1 - 7, it is clear that both the IP formulation as well as the column generation algorithm were able to compute solutions within the 0% - 10% optimality gap range. The IP formulation was not able to compute any solution for test instance 9 therefore the instance was assigned an optimality gap equal to 100%. The same is true when evaluating the computational outcomes for test instances 11 - 16. In each one of the foregoing instances, the IP formulation was not able to compute an answer and as a result, no gap could be calculated. When considering the results generated for test cases 9 - 16 when using the column generation algorithm, an optimality gap between 0% - 10% could be computed.

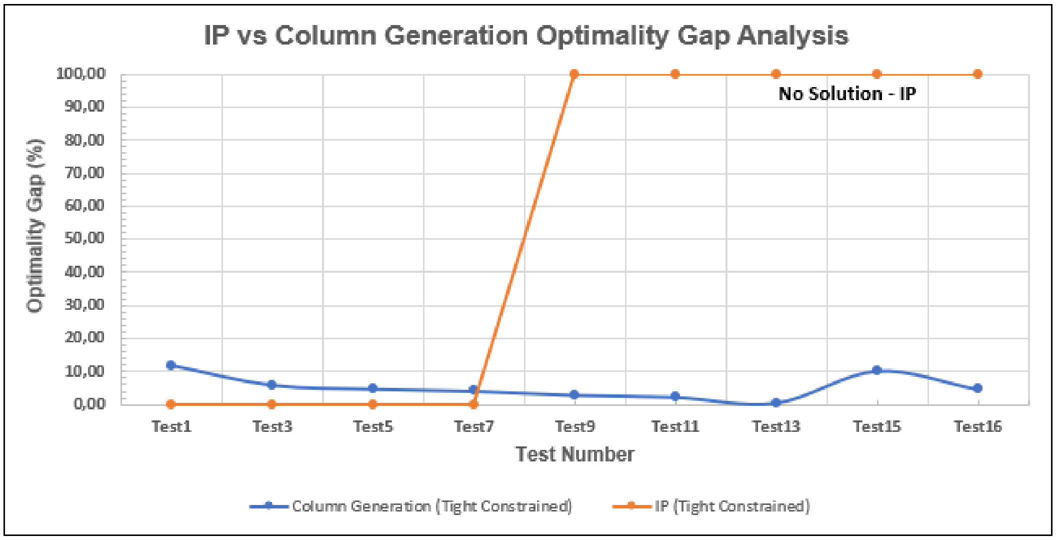

Figure 2: IP vs Column Generation Optimality GAP

Comparing the ability of the two algorithms to compute an optimality gap for tightly constrained product targeting problems, it is evident that the column generation algorithm outperforms the branch-and-bound algorithm when applied to medium or large problem sizes. The branch-and-bound algorithm, however, is better at solving small-sized tightly constrained product targeting optimisation problems, compared to the column generation algorithm.

5.2 Rows and Columns Added to Models

When applying column generation to solve large complex optimisation problems, it is stated in [8] that the computational gain achieved by the column generation algorithm can be attributed to the reduction in the number of columns and rows being considered to derive the global optimal solution. The column generation algorithm only considers a subset of the extreme points and rays for a given problem thus reducing the memory and computational requirements to solve the given problem. To confirm if the foregoing is the case, an analysis was performed on the number of columns and rows that were generated during the solutioning of test instances 1 - 16.

The data depicted in Figure 3 is representative of the number of rows and columns being generated for the tightly constrained test cases when solved using both the IP formulation as well as the column generation algorithm. Given that the number of rows and columns generated for each of the test cases varies in order of magnitude, a logarithmic scale is utilised to plot the test outcomes on the same graphing axis to ensure legibility. The top image in Figure 3 depicts the number of columns being generated by each of the mentioned algorithms whereas the bottom image is focused on the number of rows that originated from the computational process. When assessing the top image, it is apparent that the number of columns that were generated by the IP formulation was significantly more as compared to the column generation algorithm. For test 1, we note that the number of columns that were generated by the IP formulation equated to a value of 57894. For the same test case, the column generation algorithm outperformed the IP formulation to a large extent by generating only a total of 755 columns.

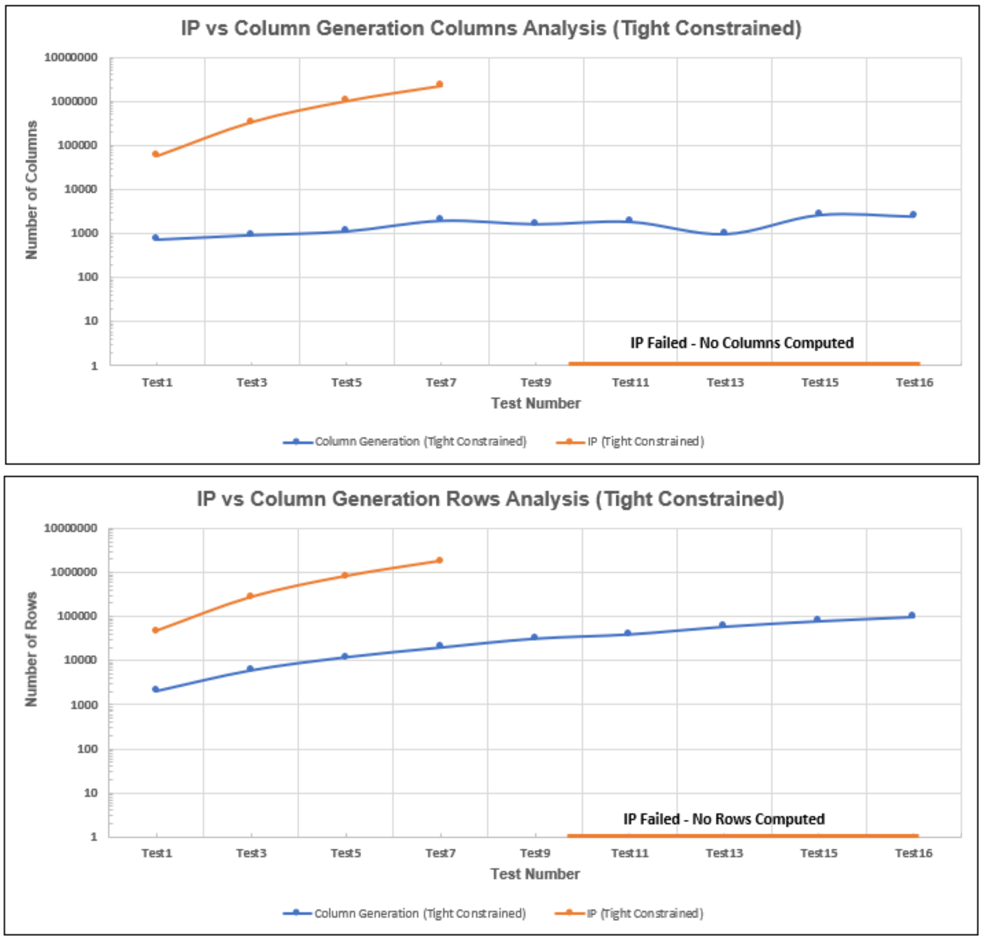

Figure 3: IP vs Column Generation Column and Row Analysis (Tightly Constrained)

A similar trend is noted for test 3 with the IP formulation generating 345425 columns while the column generation algorithm was able to compute a solution using only a minimum of 950 columns. For test cases 5 and 7, it is once again noted that the column generation algorithm generates significantly fewer columns when compared to the IP formulation. When evaluating test cases 9 -16 it is apparent that the branch-and-bound formulation was not able to compute any solution. The column generation algorithm was capable of computing a solution for each of the mentioned test cases (9 - 16) while generating a limited number of columns ranging between 1000 and 2700 columns. Given the foregoing analysis, it is apparent that the column generation algorithm is most definitely able to drastically reduce the number of columns in comparison to the columns being generated when using the IP formulation as a solution methodology. This in turn allows the column generation algorithm to reduce computational requirements which enables the algorithm to handle significantly larger optimisation problems.

The bottom image of Figure 3 shows the number of rows that were generated for each of the tightly constrained test cases. Similar to the analysis performed on the number of columns being computed, it is seen that the IP formulation produced a large amount of additional rows for each of the given test cases when compared to the rows that were generated by the column generation algorithm for the same test cases. When considering the data depicted for test 1, it is apparent that 47203 rows were generated by the IP formulation for this test instance whereas the column generation algorithm was able to keep the number of rows to a mere 2043 for the same instance. For test cases 3, 5, and 7 the IP formulation generated 276528, 826313, and 1825748 rows respectively. Evaluating the results obtained after applying the column generation algorithm, it is noted that only 6078, 12113, and 20148 rows were generated by the algorithm for the same test instances, respectively.

In the process of evaluating the results obtained for test cases 9 - 16 after applying the column generation algorithm, it is apparent that the number of rows generated for each instance was kept to a minimum with test 9 consisting of 32183 rows, test 11 having 40218, test 13 containing 60218, test 15 ending with 80253 rows and lastly test 16 not exceeding 100253 rows.

6 Summary and Conclusion

This paper proposes a novel complex product targeting problem formulation that takes into consideration a multitude of business, operation, and channel constraints not previously accounted for in the literature. This includes aspects such as customer recency, marketing consent considerations, channel preference specifications, identifying the best time of day to contact customers, selecting the best number or email address to contact customers, adding additional product limiting constraints, and lastly adding cross-sell dynamics to the model formulation. In addition to enhancing the product targeting formulation, a solution framework comprising a novel starting heuristic as well as a novel column generation formulation was proposed to allow the aforementioned complex formulation to be solved for problem sizes of up to 25000 customers, 35 products and 3 channels. The branch-and-bound algorithm was only able to solve problem instances up to a size of 5000 customers, 20 products, and 3 channels.

The results demonstrate that the proposed novel starting heuristic algorithm combined with the column generation framework is able to significantly reduce the memory requirements of the various test instances allowing the solution framework to solve significantly larger product targeting problems in comparison to standard solution methodologies. The reduction in memory requirements can be attributed to the reduced number of columns and rows being considered for each of the test cases when leveraging the column generation algorithm. The novel solution framework is able to reach global optimality for each of the test cases under consideration and solve larger test instances as compared to the instances tested in the study cited in [17].

Future work would be to investigate the feasibility of including channel dynamics such as internet, push notification, and USSD into the modeling framework to move towards a holistic formulation that is capable of accounting for all aspects of the marketing mix used by financial institutions. Other aspects to investigate would be the development of a heuristic algorithm that is capable of generating multiple column combinations that could feed into the master problem instead of only being dependent on the sub-problem formulation to generate one column per model iteration. Adding multiple columns to the master problem simultaneously would most definitely decrease the solution time required to find an optimal solution. Possibilities of including segmentation methodologies such as those discussed by Lu & Boutilier [13] into the complex product targeting framework proposed in this paper would also be worth investigating.

References

[1] Bernstel JB, 2002, *Smart move: creating intelligent database*, ABA Bank Marketing, Vol. 34, 14-19.

[2] Bose I & Chen X, 2009, *Quantitative models for direct marketing: a review from systems perspective*, European Journal of Operational Research, Vol. 195, pp 1-16.

[3] Camilleri MA, 2018, *Market segmentation, targeting and positioning*, University of Malta, (Thesis – PhD).

[4] Cohen MD, 2004, *Exploiting response models – optimizing cross-sell and up-sell opportunities in banking*, Information Systems, Vol. 29(4), pp 327-341.

[5] Delanote S, Leus R & Noninon FT, 2013 *Optimization of the annual planning of targeted offers in direct marketing*, Journal of the Operational Research Society, Vol. 64(12), 1770-1779.

[6] Du Plooy T, 2012, *A framework for the planning and integration of out-of-home advertising media in south Africa*, University of Pretoria, (Thesis – PhD).

[7] Friberg D, 2015, *An implementation of the Branch-and-Price algorithm applied to opportunistic maintenance planning*, Chalmers University of Technology and University of Gothenburg, (Thesis – Masters).

[8] Gamst, M, 2010, *Notes on Dantzig-Wolfe decomposition and column generation*, [Online], [Cited: 16 Aug 2022], Available:https://imada.sdu.dk/ jbj/DM209/Notes-DW-CG.pdf.

[9] IBM Corp, 2015, *IBM ILOG Cplex optimization studio Cplex user's manual*, [Online], [Cited: 14 July 2022], Available: https://www.ibm.com/support/knowledgecenter/SSSA5P12.6.2/ilog.odms.studio.help /pdf/usrcplex.pdf.

[10] Koopman M, 2018, *Optimizing the email marketing strategy of an airline using data modelling: a literature study*, Vrije University Amsterdam (Dissertation – Masters).

[11] Lin L, 2016, *Data mining and mathematical models for direct market campaign optimization for fred meyer jewelers*, Wright State University, (Thesis – PhD).

[12] Liu Y, Kiang M & Brusco M, 2012, *A unified framework for market segmentation and its applications*, Expert Systems with Applications, Vol. 39, pp 10292-10302.

[13] Lu T, & Boutilier C, 2014, *Dynamic Segmentation for Large-Scale Marketing Optimization*, [Online], [Cited: 14 July 2022], Available: http://www.cs.toronto.edu/ tl/papers/LuBoutilier_ICML14workshop.pdf.

[14] Pessoa A, Sadykov R, Uchoa E & Vanderbeck F, 2018, *Automation and Combination of Linear-Programming Based Stabilization Techniques in Column Generation*, Informs Journal On Computing, Vol. 30(2), pp 339 - 360.

[15] Savelsbergh M, 1997, *Branch-and-Price algorithm for the generalized assignment problem*, Operations Research, Vol. 45, pp 831 - 841.

[16] Mitik M, 2017, *Product and channel prediction for direct marketing in banking sector*, Middle East Technical University, (Dissertation – Masters).

[17] Nobibon FT, Leus R & Spieksma CR, 2011, *Models for the optimization of promotion campaigns: exact and heuristic algorithms*, [Online], [Cited: 14 July 2022], https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1290503.

[18] Reinartz W, Thomas J & Kumar W, 2005, *Balancing acquisition and retention resources to maximize customer profitability*, Journal of Marketing, Vol. 69(1), pp 63-79.

[19] Vanderbeck, F, 2005, *Implementing mixed integer column generation*, [Online], [Cited: 16 Aug 2022], Available:https://doi.org/10.1007/0-387-25486-2_12.