



Promotional retail pricing strategies through optimisation modelling

HE Horden

 **orcid.org/ 0000-0001-7115-0063**

Dissertation accepted in fulfilment of the requirements for the degree *Master of Engineering in Industrial Engineering* at the North-West University

Supervisor: Ms C Bisset
Co-supervisor: Dr PvZ van Zyl

Graduation: May 2026

Acknowledgements

The author wishes to acknowledge the following people for their support and contribution towards the completion of this study:

- My supervisors, Ms C Bisset and Dr PVZ Venter, for their guidance, insights, feedback, and support.
- My fiancée, Megan Wallis, for her patience, encouragement, and belief in me.
- My grandparents, Bill and Laetitia Horden, who deeply valued my education and whose financial support made my studies possible.

Associated publications

- The *journal article* "Optimisation of promotions in the retail industry" proposes a simple promotion optimisation model that highlights the potential for applying mathematical optimisation to retail promotion planning. Among its future recommendations, it proposes incorporating trade spend effects into the promotion optimisation framework. Thus, this paper serves as the primary inspiration for this research; however, the problem background and data simulation sections draw notably on it (see Sections 1.1 and 5.1). The paper was presented by the first author at the South African Institute of Industrial Engineering (SAIIE) conference in 2022 and was subsequently selected for publication in The South African Journal of Industrial Engineering (SAIJE), Special Edition. DOI: <https://doi.org/10.7166/33-3-2785>
- The *conference paper* titled "Investigating the Application of Mathematical Programming Models for Retail Promotion Planning" comprises a systematic literature review of promotion optimisation models. The systematic literature review in Section 2.2 presents a summarised version of this paper while retaining its main findings. The paper was presented by the first author at the Second International Conference on Industrial Engineering, Systems Engineering and Engineering Management (ISEM) in 2023.

Abstract

During trade promotions, vendors provide trade spend—funds allocated to retailers to incentivise the promotion of specific stock, typically through the application of price discounts. However, because common approaches to promotion planning are arduous, over-reliant on managerial experience, and prone to human error, retailers often apply price promotions sub-optimally by failing to fully consider the interrelated impacts of consumer response and trade spend. While a systematic literature review (SLR) reveals that systematic approaches for promotion optimisation exist in the literature, none simultaneously accounts for own-promotional effects (the demand response of the promoted items), cross-promotional effects (the demand response of related non-promoted items), and trade spend effects.

This study aims to address the problem and gap identified in the literature by proposing a new mixed-integer nonlinear programming (MINLP) optimisation model that determines the discount depths of retail items, maximising retailer profit, while incorporating own-promotional, cross-promotional, and trade spend effects.

The model is verified in two stages. First, the model’s adherence to the requirements of the selected MINLP heuristic solver (BONMIN) is verified to ensure correct solver use. Second, the correctness of the model’s formulation and coding is verified to ensure it implements the intended conceptual design.

Next, the model is validated in three stages, each employing non-trivial simulated data. First, a benchmark analysis evaluates the optimality gap between heuristic and exact solutions across 20 datasets, spanning small to large problem sizes. The results indicate that the heuristic solutions are exact in all cases, validating the reliability of the solving approach’s solutions. Second, a sensitivity analysis is conducted in which key model parameters are varied, after which the responses are qualitatively assessed. The results indicate that the responses are internally consistent and theoretically aligned across parameter variations, validating the model’s ability to reflect realistic behaviour. Third, a scalability analysis is conducted wherein the model is solved using systematically increasing problem sizes to establish a trend in computation time. The results indicate that computation time grows exponentially with the number of variables, validating that the solving approach remains effective only for relatively small problem instances.

This study is limited by the absence of a conclusive benchmark of its profit-enhancing capability. In addition to addressing scalability and benchmarking issues, future research could extend the model by incorporating a stochastic framework to account for demand uncertainty, framing it in a multi-period setting to capture overpromotion effects (e.g., post-promotion demand dips), and expanding the representation of trade spend, cross-promotional, and inventory effects.

Despite these limitations, this study contributes to the literature by presenting the first known retail promotion optimisation model that accounts for own-promotional, cross-promotional, and trade spend effects simultaneously. In doing so, the study demonstrates that the model yields higher retailer profits than isolated or ad hoc discounting strategies. Thus, the study highlights the financial value of holistic decision-support tools in reducing reliance on intuition in retail promotion planning.

Keywords: Retail price promotion; Trade promotion; Trade spend; Cross-promotional effects; Optimisation; Mixed-integer nonlinear programming

Contents

1	Project introduction	1
1.1	Problem background	1
1.2	Research problem, aim, and objectives	4
1.3	Research methods	5
1.4	Delineation and limitations	7
1.5	Ethical considerations	7
1.6	Chapter division	8
2	Literature study	9
2.1	Theoretical foundation	9
2.1.1	Base terminology and problem classifications	9
2.1.2	Duality theory and the KKT conditions	11
2.1.3	Optimisation techniques	12
2.1.4	Selected optimisation algorithms	14
2.1.5	Selected solvers	17
2.2	Systematic literature review	19
2.2.1	SLR background and review methodology	19
2.2.2	SLR Results	19
2.3	Chapter summary	24
3	Mathematical model development	25

3.1	Conceptual problem overview and key assumptions	25
3.1.1	Base own-promotional assumptions	25
3.1.2	Base cross-promotional assumptions	26
3.1.3	Base trade spend considerations	26
3.1.4	Conceptual model representation	27
3.2	Model notation	27
3.2.1	Set and index definitions	27
3.2.2	Parameter definitions	28
3.2.3	Variable definitions	29
3.3	Supporting propositions	31
3.3.1	Proposition (1)	31
3.3.2	Proposition (2)	33
3.4	Mathematical model formulation	33
3.4.1	Objective function formulation	33
3.4.2	Constraint formulations	36
3.4.3	An estimation method for cross-promotional demand cannibalisation (α_{hik})	38
3.5	Chapter summary	39
4	Verification	40
4.1	Compliance with solver requirements	41
4.2	Correctness of the formulation and coding	42
4.2.1	Verification approach	42
4.2.2	Assessment I	42
4.2.3	Assessment II	43
4.2.4	Assessment III	43
4.2.5	Assessment IV	44

4.2.6	Assessment V	45
4.2.7	Assessment VI	46
4.2.8	Assessment VII	47
4.2.9	Formulation and coding conclusions	48
4.3	Chapter summary	48
5	Validation	49
5.1	Data simulation	50
5.2	Solving environment	55
5.3	Analysis methodologies	57
5.3.1	Benchmark analysis methodology	57
5.3.2	Sensitivity analysis methodology	57
5.3.3	Scalability analysis methodology	59
5.4	Results and discussions	60
5.4.1	Benchmark analysis results and discussion	60
5.4.2	Sensitivity analysis results and discussion	61
5.4.3	Scalability analysis results and discussion	65
5.5	Chapter summary	66
6	Conclusion	67
6.1	Chapter summaries	67
6.2	Future research	70
A	Proofs	78
A.1	Proof of proposition (1)	78
A.2	Proof of proposition (2)	81
B	Operational models	84

B.1	Verification models	84
B.2	Complete operational model	88
B.3	The fixed-margin pricing model	90
C	Datasets	92
D	Results	93
D.1	Sensitivity analysis results	93
D.2	Solver accuracy assessment results	102

List of Figures

1.1	The relationship between key promotional concepts.	2
2.1	Systematic Literature Review protocol.	20
3.1	Conceptual representation of the proposed model for optimising discounting strategies given own-promotional, cross-promotional, and trade spend effects.	27
5.1	The demand curves used in simulating own-promotional data.	50
5.2	A conceptual representation of the data simulation approach.	52
5.3	The demand models used in the sensitivity analysis	56
5.4	The variable count of the computability analysis as a function of $ N $ when $ W = 2$ and $ K = 4$	59
5.5	The computation time required to solve each instance, shown as a function of the number of items. The figure also shows the number of variables per instance.	65
5.6	The computation time required to solve each instance using a standard B&B and IPM solving configuration, shown as a function of the number of items. The figure also shows the number of variables per instance.	66
D.1	The impact of varying key parameters on profit (relative to the baseline case).	93
D.2	The impact of varying a_i on key performance indicators.	94
D.3	The impact of varying r_i on key performance indicators.	95
D.4	The impact of varying c_i on key performance indicators.	96
D.5	The impact of varying η_i on key performance indicators.	97
D.6	The impact of varying Ω_{iww} and λ_i on key performance indicators.	98

D.7	The impact of varying L_{ivw} and U_{ivw} on key performance indicators.	99
D.8	The impact of varying m on key performance indicators.	100

List of Tables

2.1	The studies identified by the review	21
2.2	The modelling approaches employed within the identified studies	23
2.3	The occurrence of common retail effects within the identified studies	24
4.1	Standard BONMIN formulations of model constraints	41
4.2	Mapping of each verification assessment to the corresponding model formulation and AMPL coding.	42
4.3	The test instances for (C4- C7) verification	46
4.4	The test instances for (3.37a - 3.37d) verification	46
5.1	The optimality gaps of the heuristic (BONMIN) solutions across the twenty simulated instances.	60
C.1	Sensitivity analysis baseline dataset	92
D.1	Numerical results from the sensitivity analysis.	101
D.2	Comparison of heuristic (BONMIN) solutions with exact (BARON) solutions for 20 simulated instances. The table reports the objective function value and selected decision variables for each solve.	102

Table of abbreviations

AMPL	A mathematical programming language
B&B	Branch-and-bound
BONMIN	Basic open-Source nonlinear mixed integer programming (solver)
Ipopt	Interior point optimizer
IPM	Interior point method
KKT	Karush-Kuhn-Tucker (conditions)
LP	Linear programming
MINLP	Mixed-integer nonlinear programming
SLR	Systematical literature review

Chapter 1

Project introduction

1.1 Problem background

Retailers must consider several trade-offs inherent in conducting price promotions, i.e., short-term discounts on retail items to influence consumer behaviour (Blattberg et al., 1995). The retailer gains from promotions as they are known to increase: the short-term sales of promoted items (Woodside and Waddle, 1975), the store traffic (Walters, 1991), the frequency of customer visits (Anderson and Fox, 2019), the spending per customer visit (Anderson and Fox, 2019), and the sales of complementary items (Cohen et al., 2021).

Furthermore, price promotions are critical for driving demand for durable consumer goods (Anderson and Fox, 2019), and evidence suggests the same holds for non-durable consumer goods. For instance, in 2004, 12%-25% of European grocery retailers' revenue came from promoted products (Ailawadi et al., 2007). Similarly, in 2009, 42% of US grocery store sales and 40.4% of US pharmaceutical retail sales were driven by promotional items (Felgate et al., 2012).

However, promotions may also reduce overall profitability if implemented ineffectively. This is because, in addition to reducing profit margins, they are also known to induce stockpiling (Ailawadi et al., 2007), alter customers' reference price framework (Bolton, 1989), and cause cross-promotional cannibalisation, i.e., decrease sales of substitute items (Cohen et al., 2021).

To mitigate these drawbacks, vendors incentivise retailers to conduct price promotions via trade promotions; trade promotions provide a rare opportunity to influence consumer behaviour, since vendors typically cannot profitably sell directly to consumers (Anderson and Fox, 2019). Trade promotions typically involve the transfer of funds, known as trade spend. For further clarity, the relationship among trade promotions, price promotions, trade spend, and discounts is visualised in Figure 1.1.

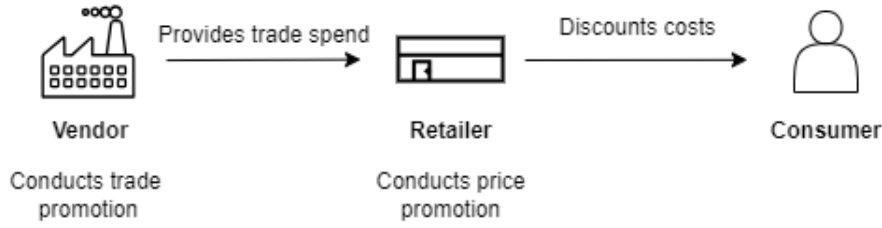


Figure 1.1: The relationship between key promotional concepts.

The value trade spend provides as a direct link to consumers is evident in the significant investment vendors allocate to this activity; following the cost of goods, trade spend accounts for the second-largest category of vendor expenditure (Gomez et al., 2007), amounting to 10%-15% of total revenue (Anderson and Fox, 2019); global trade spending amounts to approximately \$500 billion per year (Gaissmaier et al., 2012).

It follows that trade spend represents a lucrative and strategically important revenue stream for retailers. In return, retailers are required to introduce new stock, improve stock display and advertising, increase supply chain efficiency, or conduct price promotions (Anderson and Fox, 2019). Among these, price promotions, in particular, effectively increase vendor market share by directly influencing consumers to increase product consumption (Assunção and Meyer, 1993), switch brands from competitors (Anderson and Fox, 2019), and accelerate purchases to prevent competitors from acquiring future purchases (Bell et al., 1999).

Anderson and Fox (2019) describe twelve trade spend mechanisms that differ in terms of retailer performance requirements, the retailer expense being offset, and the payment characteristics. Among them, Tsao (2016) notes that off-invoice allowances, scan-backs, and buy-backs are most often used to implement price promotions. A description of each, as obtained from Tsao (2016), follows: off-invoice allowances are discounts on vendor items based on the number of units ordered, scan-backs are payments made to the retailer based on the number of units sold, and buy-backs allow retailers to return unsold stock to vendors at an agreed-upon price and, as such, are an allowance for every unsold unit of vendor stock.

The value of an off-invoice allowance is determined before the purchasing period and is therefore independent of the sales performance of the relevant items. Retailers frequently exploit off-invoice allowances by purchasing excess stock to later sell at regular margins or to resell to other retailers (Anderson and Fox, 2019). By contrast, the value of “pay-for-performance” offerings, such as scan-backs and buy-backs, depends directly on sales performance, which limits retailer exploitation (Gomez et al., 2007). While off-invoice allowances have historically been the most common trade spend mechanism, vendors have shifted toward a balance of mechanisms, including “pay-for-performance” offerings, to mitigate exploitation (Bell and Drèze, 2002); in the mid-1990s, 90% of trade spend was offered as off-invoice allowances; by the mid-2000s, this figure had decreased to 35% (Gomez et al., 2007).

To incorporate trade spend into promotion planning, retailers must evaluate the potential revenue from these mechanisms against the inventory investments required. Additionally, the promotion planning problem becomes increasingly complex as more mechanisms are considered, because their effects are interrelated. For instance, buy-backs and scan-backs represent contrasting forms of trade spend with opposing revenue-generation mechanisms: while buy-backs generate revenue from unsold stock, scan-backs generate revenue from sales. In turn, scan-backs and buy-backs affect inventory planning, thereby influencing revenue from off-invoice allowances. Cross-promotional cannibalisation further alters sales volumes, thereby indirectly affecting trade spend revenue. Consequently, own-promotional effects, cross-promotional effects, and trade spend mechanisms are inherently interdependent and cannot be considered in isolation if a comprehensive assessment of promotion profitability is to be achieved

Failure to consider the holistic impact of these effects on profitability entails financial risk; the following case study demonstrates the severity of this risk. Ailawadi et al. (2007) found that 50% of promotions conducted at a notable US pharmaceutical retailer over one year were unprofitable as profit margins were not sufficiently offset by trade spend. The authors estimate that the retailer lost \$52 million due to its inability to accurately assess promotion profitability.

Regardless of the complexity and risk involved in designing promotions, it is standard practice for retailers to use informal heuristics to plan promotions (Cohen et al., 2017), for example, by repeating discounts that were profitable in a previous period (Anderson and Fox, 2019). However, these heuristics are fundamentally limited since manual planning may not account for the simultaneous interactions among demand response, substitution effects, and the financial implications of trade spend mechanisms. Furthermore, as the number of possible promotions increases, these interactions become increasingly difficult to evaluate intuitively.

This gap between operational complexity and intuitive planning has prompted academic efforts to develop systematic approaches to designing promotions. The systematic literature review (SLR) associated with this study (see the study by Horden et al. (2023)) identifies and evaluates existing models that aim to assist retailers with promotion planning. Among the ten studies identified by the SLR, the effects of own-promotion (i.e., the expected increase in a promoted item's sales), cross-promotion, stockpiling, and trade spend are examined using profit-maximising frameworks. Within the scope of the SLR, no single model considers the effects of own-promotional, cross-promotional, and trade spend simultaneously.

While the SLR's scope is limited to studies published by 2022, contemporary literature reflects similar constraints. For instance, Phumchusri et al. (2024) propose a pure integer programming model that solves for the set of discounts maximising retailer profit over a multi-period horizon. The study notably addresses cross-promotional effects and stockpiling effects but excludes trade spend effects. The MINLP model proposed by Baardman et al. (2023) captures complex demand features—such as seasonality and social-influence trends—but omits cross-promotional and trade-spend dynamics. The pure integer Non-linear Programming (NLP) model proposed by Mohammadipour et al. (2023) accounts for own-promotional impact and cross-promotional effects; however, the latter is limited to inter-store cannibalisation, and their framework excludes trade spend effects.

Excluding cross-promotional effects risks overstating sales across the product category, whereas omitting trade spend mechanisms may understate or overstate revenue. Furthermore, treating trade spend as exogenous fails to capture the relationships among trade spend, discount depth, and sales. Consequently, existing models provide only partial representations of the retail promotion optimisation problem.

Given these limitations, there is a clear need for an optimisation model that simultaneously incorporates own-promotional, cross-promotional, and trade spend effects within a unified profit-maximising framework. A mathematical optimisation approach—such as mixed integer nonlinear programming (MINLP)—is appropriate because it provides a decision-support framework for structuring the problem inputs (e.g., item costs, permissible discounts, trade spend criteria, etc.) and outputs (e.g., the selection of discount depths of each considered item) under the possibly nonlinear behavioural relationships (e.g., own-promotional, cross-promotional, and trade spend effects, operational constraints, etc.) with a clear objective (maximising retailer profit). Therefore, the problem addressed in this dissertation is linked to the development of a MINLP model that integrates these interrelated effects within a decision-support framework.

1.2 Research problem, aim, and objectives

From the background, it is evident that retailers experience significant challenges in designing profitable promotions. This study aims to address the problem and the gap identified in the literature by proposing a new optimisation model. The model aims to maximise retailer profit while accounting for own-promotional, cross-promotional, and trade spend effects.

To achieve this aim, the study is guided by the following seven research objectives:

1. To conduct a literature study that:
 - (a) provides the theoretical foundation for this study.
 - (b) contextualises the study within existing literature and establishes its theoretical contribution.

2. To propose a new MINLP model that assists retailers with promotion planning while accounting for own-promotional, cross-promotional, and trade spend effects.
3. To verify the model, by:
 - (a) verifying the proposed model's adherence to the requirements of the solving approach.
 - (b) verifying the correctness of the formulation and coding of the proposed model.
4. To validate the model and the solving approach, by:
 - (a) validating the reliability of the solutions produced by the solving approach.
 - (b) validating the model's ability to reflect realistic behaviour.
 - (c) validating the computability of the solving approach.
5. To recommend future research and improvements based on the presented work.

1.3 Research methods

The following research methods are applied, each directly linked to a research objective formulated in Section 1.2:

1. (a) A detailed overview of optimisation theory is presented to construct the theoretical foundation of this study. The topics were selected for their relevance to the formulation and solution of the proposed model.
- (b) The results obtained from the SLR (see (Horden et al., 2023)) are evaluated and mentioned to contextualise the study within existing literature and establish its theoretical contribution. The SLR follows the methodology first outlined by Kitchenham (2004), which has since been widely adopted in the literature (Babar and Zhang, 2009).
2. A new proposed MINLP model is formulated with the objective of maximising retailer profit, while accounting for own-promotional, cross-promotional, and trade spend effects.
3. (a) Solving approaches relevant to the proposed model may require that the model's objective function and constraints are twice continuously differentiable (Wächter and Biegler, 2006; Bonami, 2023). As discussed in Section 2.1.1, the Hessian (2.4) provides a general means of verifying second-order differentiability. However, explicitly computing the model's Hessian is impractical because it involves a relatively large number of multidimensional variables whose dimensions depend on user-defined set sizes. Instead, to verify adherence, the algebraic structures of the objective function and constraints are evaluated and classified as polynomial, with all derivatives continuously differentiable to all orders. The

polynomial argument has previously been applied in the optimisation literature to verify differentiability requirements; see, for example, (Nguyen Thai and Ho, 1991) and (Nie, 2012).

- (b) Computerised model verification is a standard approach for ensuring computerised models are implemented correctly and error-free (Sargent, 2013). Computerised model verification involves dynamic testing, wherein the program is executed under different conditions while evaluating input-output relations for correctness (Sargent, 2013). The method is particularly appropriate for this study because the primary contribution is a computationally implemented model. Accordingly, the correctness of the formulation and coding of the proposed model is verified through a series of assessments that compare solver outputs with analytical values for trivial inputs. If the model's output is mathematically correct across all instances, this indicates that the model is correctly formulated and coded.
4. (a) As this study primarily employs a solver whose solution optimality is not guaranteed (Bonami, 2023), solution quality cannot be inferred from solver convergence alone. It is, thus, appropriate to validate the reliability of the solver's solutions. To this end, benchmarking against an exact solver on identical problem instances provides an objective means of assessing whether the heuristic approach consistently identifies optimal or near-optimal solutions (Beiranvand et al., 2017). A series of non-trivial datasets is simulated. Each instance is solved twice: first, using the heuristic solver, and second, using an exact solver. When the solvers produce similar solutions in terms of the solutions' objective function values and variable assignments, the reliability of the primary solving approach is validated.
 - (b) Sensitivity analysis is a standard technique in model validation (Sargent, 2013). In this approach, model parameters are varied to assess their effects on the model's output. The output is then qualitatively evaluated to confirm that the same relationships occur in the model as in the real system (Sargent, 2013). This method is particularly appropriate for the study because, in the absence of empirical data, behavioural plausibility can be evaluated through controlled variation of parameters. By demonstrating that changes in parameters produce responses that are economically intuitive, theoretically aligned, and internally consistent, the model's ability to reflect realistic behaviour is validated.
 - (c) The proposed model includes nonlinear and mixed-integer components, which are known to rapidly increase solution times as dimensionality grows. As a model's computability can be determined by analysing its computation time (Beiranvand et al., 2017), the model's computability is validated through a scalability analysis, in which problem sizes are systematically increased to establish a trend in computation time. The trend is then evaluated to determine the problem sizes for which the solution approach remains feasible.

5. Based on the presented work, insights are generated regarding future research and

improvements.

1.4 Delineation and limitations

To retain a reasonable scope, the following delineations and limitations apply:

- The model considers only price promotions and the profitability impact of the three trade spend mechanisms most often resulting from them, namely off-invoice allowances, scan-backs, and buy-backs.
- Modelling retail demand is inherently complex, as it depends on various factors, including but not limited to retailer location, product category, and item display level. Developing an accurate demand model, therefore, requires a thorough investigation and extensive historical data. Given the absence of comprehensive retail data and to maintain a reasonable scope of work, demand forecasting is excluded from this study.
- Since the proposed model is computationally expensive, owing to its high-dimensionality, nonlinear interactions, and integer requirements, it is formulated as a single-period problem rather than a multi-period problem to ensure computational tractability. The primary implication of this choice is that the model does not capture over-promotion effects (e.g., stockpiling effects) or multi-period inventory management functionality.
- The focus of this study is on the development of the model and not on its solving techniques. Accordingly, pre-existing solvers are used as required.
- The study exclusively uses simulated operational retail data.

1.5 Ethical considerations

The following ethical considerations apply:

- The study exclusively uses simulated store-level data and uses no human participants. Consequently, there are no ethical risks related to the use of industry or personal data.
- All software used is either open source or, where proprietary software has been employed, its use is permitted under academic licensing terms.
- All code and datasets used in this study have been made publicly available on GitHub to ensure transparency and facilitate reproducibility. The code is available at: <https://github.com/29922704/masters-dissertation>.

- The researcher adhered to the North-West University's code of conduct throughout this study, maintaining professionalism and upholding research integrity. This commitment includes honesty in all research activities, accountability in conduct, professional courtesy and fairness in collaborations, and responsible stewardship of research.
- There are no conflicts of interest to declare.

Following a review conducted on 13 February 2023 by the North-West University Engineering Research Ethics Committee, the study was cleared as a no-risk study. The following ethics number applies: **NWU-02035-22-A1**.

1.6 Chapter division

The study is documented within a dissertation comprising this introductory chapter, six additional chapters, and a bibliography. The chapters are organised as follows. The first chapter, Project Introduction, presents the problem background and defines the scope of the study. The second chapter, Literature Study, provides a detailed overview of optimisation theory and presents a systematic literature review (SLR) of existing models that assist retailers with promotion planning. The third chapter, Mathematical Model Development, formulates the optimisation model. The fourth chapter, Verification, describes the verification conducted in two stages, while the fifth chapter, Validation, presents validation conducted in three stages. Finally, the sixth chapter, Conclusion, summarises the dissertation's major findings and recommends directions for future work.

Chapter 2

Literature study

This chapter presents a detailed overview of optimisation theory and reports an SLR of existing models that assist retailers with promotion planning.

The chapter comprises two sections: the theoretical foundation and the SLR. The first section covers topics relevant to the study's modelling approach, solving approach, and solution interpretation. To this end, the first section covers: base terminology and problem classifications; duality theory and the Karush-Kuhn-Tucker (KKT) conditions; optimisation techniques; select optimisation algorithms; and select optimisation solvers.

2.1 Theoretical foundation

2.1.1 Base terminology and problem classifications

Let $x_1, x_2, \dots, x_n$ represent n decision variables to a problem whose values are to be determined (Hillier, 2001). The objective function, representing a performance measure to be minimised or maximised, is expressed as a function of the decision variables. Any restrictions imposed on the domain of the decision variables or on their relationships are generally expressed as inequalities, called constraints. The constant values within the constraints are called the model's parameters (Hillier, 2001). The solution to an optimisation problem is feasible if it satisfies all constraints (Hillier, 2001). Because a problem may have multiple feasible solutions, the best one with respect to the objective function is called the optimal feasible solution. A problem may have multiple optimal solutions or none at all.

Optimisation problems are classified based on the mathematical properties of the objective function, constraints, and decision variables (Hillier, 2001). A linear programming (LP) problem refers to the optimisation of a linear objective function subject to linear constraints (Hillier, 2001). The standard form of an LP problem is:

$$\text{Maximise } Z = \mathbf{c}^T \mathbf{x}, \quad (2.1a)$$

$$\text{subject to: } \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad (2.1b)$$

$$\mathbf{x} \geq \mathbf{0}, \quad (2.1c)$$

where $\mathbf{c}$ represents the vector of coefficients for the objective function, $\mathbf{x}$ represents the vector of decision variables to be optimised, $\mathbf{A}$ represents the matrix of coefficients for the constraints, and $\mathbf{b}$ represents the vector of constraint values on the right-hand side of the constraints (Hillier, 2001).

Conversely, NLP problems include cases where either the objective function, a constraint, or both are nonlinear (Hillier, 2001). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ denote the objective function, and let $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ denote the inequality- and equality-constraint functions, respectively. If either f , g , h , or some combination thereof is nonlinear, the general form of an NLP problem, as given by Hillier (2001), is:

$$\text{minimise } f(\mathbf{x}), \quad (2.2a)$$

$$\text{subject to: } g_i(\mathbf{x}) \leq 0 \quad \forall i \in \{1, \dots, m\} \quad (2.2b)$$

$$h_j(\mathbf{x}) = 0 \quad \forall j \in \{1, \dots, p\} \quad (2.2c)$$

LP and NLP problems are further classified by the domains of their respective decision variables. A mixed-integer linear programming (MILP) problem is an LP problem in which at least one decision variable is restricted to integer values (Hillier, 2001). Similarly, a MINLP problem is an NLP problem in which the domain of at least one decision variable is restricted to the set of integers (Hillier, 2001).

Finally, optimisation problems are classified according to their convexity, as it affects a problem's computability; non-convex problems require supplementary computation to address each distinct source of non-convexity and are, as such, generally harder to solve than convex problems (Hillier, 2001). Minimisation problems are convex if the objective function and inequality constraints are convex, the equality constraints are affine (linear plus a constant term), and the decision variable domains are continuous. The same applies to maximisation problems; however, their objective functions must be concave. It follows that all continuous LP problems are automatically convex, and all MILP and MINLP problems are non-convex (Hillier, 2001).

Function convexity can be determined by computing its gradient vector and Hessian matrix (Chong and Żak, 2013). If the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable, its gradient, denoted by ∇f , is:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{bmatrix} \quad (2.3)$$

Further, if ∇f is itself differentiable, the Hessian matrix of f , denoted by $\nabla^2 f$, is the square matrix of partial derivatives:

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (2.4)$$

If $\nabla^2 f$ is positive semi-definite for all $\mathbf{x}$, then f is convex. If $\nabla^2 f$ is negative semi-definite for all $\mathbf{x}$, then f is concave (Chong and Żak, 2013).

By establishing the problem's convexity and domain characteristics, the appropriate theoretical framework for finding an optimal solution can be selected. For many of the classifications discussed, the analysis of optimality is rooted in Duality theory.

2.1.2 Duality theory and the KKT conditions

Duality theory is fundamental to mathematical optimisation, providing insights into optimality conditions and algorithm design.

Every LP problem, referred to as the primal problem, is associated with a corresponding dual problem (Hillier, 2001). The dual problem is itself an LP, whose structure is derived from the primal. For instance, the dual associated with the standard primal LP problem (2.1) is:

$$\text{Minimise } W = \mathbf{y}^T \mathbf{b}, \quad (2.5a)$$

$$\text{subject to: } \mathbf{y} \mathbf{A} \geq \mathbf{c}, \quad (2.5b)$$

$$\mathbf{y} \geq \mathbf{0}, \quad (2.5c)$$

Under certain regularity conditions, two properties follow from the relationship between the primal and dual problems (Hillier, 2001). The weak duality property states that for any feasible primal solution x and dual solution y , $\mathbf{c}^T \mathbf{x} \leq \mathbf{y}^T \mathbf{b}$. This property ensures that the dual objective provides a lower bound on the primal objective for minimisation problems (and an upper bound in maximisation problems). The strong duality property guarantees that, if $\mathbf{x}^*$ and $\mathbf{y}^*$ are feasible optimal solutions for primal and dual LP problems, respectively, then $\mathbf{c}^T \mathbf{x}^* = \mathbf{y}^{*T} \mathbf{b}$ (Hillier, 2001). This property extends to convex NLP problems, provided certain constraint qualifications hold, e.g., Slater's condition, which requires the existence of a point x satisfying $h_j(x) = 0$ and $g_i(x) < 0$, following the notation of (2.2) (Nocedal and Wright, 2006).

While strong duality may not hold in non-convex problems, weak duality holds universally (Nocedal and Wright, 2006). From this property, the duality gap $\mathbf{y}^T \mathbf{b} - \mathbf{c}^T \mathbf{x}$ measures proximity to optimality and nears zero at convergence (Hillier, 2001).

Duality extends to NLPs via the Lagrangian function, which reformulates a constrained problem by incorporating constraints into the objective function via Lagrangian multipliers (Bertsekas, 2016). The Lagrangian function of the standard NLP (2.2) is defined as:

$$\mathcal{L}(x, \lambda, z) = f(x) + \sum_{j=1}^p \lambda_j h_j(x) + \sum_{i=1}^m z_i g_i(x) \quad (2.6)$$

where λ_i denotes the Lagrange multipliers for the inequality constraints (2.2b) and z_j denotes the Lagrange multiplier for the equality constraints (2.2c). In this framework, the multipliers function as dual variables, enabling the analysis of primal-dual relationships and the derivation of the KKT conditions, i.e., the necessary (but not sufficient) conditions that a solution to a constrained optimisation problem is a local optimum. The following theorem, paraphrased from Bertsekas (2016), defines these conditions.

Theorem 1. *Let x^* be a local minimum of the NLP (2.2), where f , g_i , and h_j are continuously differentiable satisfying certain regularity conditions. Then, there exist unique Lagrange multipliers $\lambda^* \in \mathbb{R}^p$ (for equality constraints) and $z^* \in \mathbb{R}^m$ for inequality constraints, such that:*

$$\nabla_x \mathcal{L}(x^*, \lambda^*, z^*) = 0 \quad (2.7a)$$

$$g_i(x^*) \leq 0, \quad h_j(x^*) = 0 \quad (2.7b)$$

$$z_i^* \geq 0 \quad (2.7c)$$

$$z_i^* g_i(x^*) = 0 \quad (2.7d)$$

where (2.6) gives the Lagrangian $\mathcal{L}(x, \lambda, z)$.

Among the required regularity conditions, the Linear Independence Constraint Qualification (LINQ) is often applied in non-convex optimisation. LINQ holds at a point x^* if the set of gradient vectors corresponding to all equality constraints and the active inequality constraints at x^* are linearly independent (Nocedal and Wright, 2006).

Given that the regularity conditions are met, and the point (solution) x^* and its corresponding Lagrange multipliers λ^* and z^* satisfy all KKT conditions, the theorem implies that the point represents a local optimum (Bertsekas, 2016). Thus, solving the KKT conditions identifies candidate solutions to the original NLP (2.2).

While duality theory provides a foundation for identifying and verifying optimal solutions, it does not prescribe the path to reach them. Consequently, various optimisation techniques have been developed to solve mathematical programming problems.

2.1.3 Optimisation techniques

Optimisation methods can be categorised into exact, heuristic, and metaheuristic approaches, each differing in their solution guarantees, computational efficiency, and applicability (Hillier, 2001).

Exact methods find provably optimal solutions by comprehensively exploring the solution space. The efficiency of these methods relies on problem convexity. Exact methods are well-suited to convex problems; for example, the Simplex method is notably efficient at finding exact solutions to LP problems (Chong and Ćak, 2013). Furthermore, as a result of the strong duality property, a zero duality gap confirms the global optimality of convex problems, providing both a stopping criterion for solvers and a guarantee of global optimality (Bertsekas, 2016).

For non-convex problems, exact methods must account for local optima, which may require exhaustive enumeration of advanced global optimisation techniques (Hillier, 2001). The computational effort for exact methods generally scales with problem size; MINLP problems, notably, scale exponentially (Naseera, 2023). Finding exact solutions to highly complex problems may not be practical. Heuristic methods do not require complete computation to find reasonably good, feasible solutions. Heuristic methods do not necessarily search the entire solution space of a given problem, and their computational effort is lower than that of exact methods (Martí and Reinelt, 2022). This makes heuristic methods particularly useful in solving large-scale optimisation problems. The trade-off for this efficiency is that heuristic methods are not guaranteed to find optimal solutions as they may converge to a local optimum (Hillier, 2001).

As heuristics may not guarantee global optimality, the weak duality property remains useful in providing optimality bounds for local optimality. However, to increase the likelihood of convergence to the global optima, higher-level strategies can be employed to escape local optima. These strategies, known as metaheuristics, include: the Tabu search method, which employs a memory feature to escape local optima (Glover, 1990), Simulated annealing, which incorporates randomness within its moves to navigate away from local optima (Bertsimas and Tsitsiklis, 1993), and Genetic algorithms, which consider a population of solutions and mimic evolution and natural selection to generate improved solutions (Holland, 1992).

Regardless of the optimisation approach, practical implementation is typically carried out using specialised solvers, software that implements exact, heuristic, or metaheuristic algorithms, or combinations thereof, to process computations efficiently (Kronqvist et al., 2019). Solver requirements and performance depend on problem class, convexity, size, and licensing restrictions, making solver selection a critical consideration in optimisation. Popular solvers include Gurobi, CPLEX, Ipopt, BARON, and BONMIN (Kronqvist et al., 2019; Nemirovski and Todd, 2008).

To interface between abstract mathematical formulations and the solver’s computational engine, modelling languages are employed. These allow the formulation of problems in concise algebraic form that closely resembles their mathematical notation (Kronqvist et al., 2019). Additionally, this separation of the model from its solution process, in conjunction with the standardisation of solver input, allows the same models to be solved using different solvers with minimal modification, enabling solver benchmarking. Commonly used modelling languages include AMPL, AIMMS, JuMP, Pyomo, and MATLAB (Kronqvist et al., 2019). While not unique among these, AMPL notably provides a Python API further enabling model building and solver interaction (AMPL Optimization Inc., 2021).

2.1.4 Selected optimisation algorithms

To provide further perspective on solving optimisation problems and to establish the theoretical background for the following section, this section reviews a selection of foundational optimisation algorithms, including Branch-and-Bound, Interior-Point Methods, and Newton’s Method.

2.1.4.1 The branch-and-bound algorithm

The standard branch-and-bound (B&B) algorithm addresses integrality by systematically exploring the solution space, recursively dividing the problem into smaller subproblems (branching) by fixing integer variables to different values (Christodoulos A., 1995). The resulting continuous relaxations are then solved to compute bounds on the objective function. If a relaxation’s bound is worse than the best-known feasible solution, the subproblem is discarded (pruned); otherwise, the algorithm continues branching. This process repeats until all subproblems are either resolved or pruned (Christodoulos A., 1995).

While the standard B&B algorithm guarantees global optimality for convex MINLP relaxations, non-convex relaxations pose computational challenges, as NLP solvers may converge to local optima, and relaxed solutions no longer provide rigorous bounds on the objective (Bonami, 2023).

2.1.4.2 The Newton-Raphson method

In the context of optimisation, the Newton-Raphson method is an efficient numerical method for finding local optima (Bertsekas, 2016). Bertsekas (2016) presents the method as follows. Consider a twice differentiable function $f(x)$ wherein an initial trial solution x^0 is given. The method iteratively computes:

$$x^{k+1} = x^k - \alpha^k [\nabla^2 f(x^k)]^{-1} \nabla f(x^k), \quad (2.8)$$

where α is the step size. Under suitable conditions, this sequence converges to a local optimum.

Newton's method also applies to systems of nonlinear equations, as Kelley (1995) describes in the following. Let $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with components $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^\top$. Then, starting from an initial vector $\mathbf{x}^0$, the roots of $\mathbf{F}$ are iteratively computed by:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \mathbf{J}_{\mathbf{F}}(\mathbf{x}^k)^{-1} \mathbf{F}(\mathbf{x}^k), \quad (2.9)$$

where α^k is the step size, and $\mathbf{J}_{\mathbf{F}}(\mathbf{x}^k)$ is the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{x}^k$, defined as:

$$\mathbf{J}_{\mathbf{F}}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \quad (2.10)$$

2.1.4.3 The primal-dual interior point method

Interior point methods (IPMs), also known as barrier methods, are a class of algorithms for solving optimisation problems by iteratively approaching an optimum from the interior of the feasible region (Nemirovski and Todd, 2008). Originally introduced by Karmarkar (1984) for linear programming, IPMs have been extended to general NLPs. A key advancement is the primal-dual framework, which treats the perturbed KKT conditions as a nonlinear system and solves it iteratively using an adapted Newton-Raphson method (Nemirovski and Todd, 2008). The following summary of the Primal-dual IPM is drawn from Wächter and Biegler (2006) and Nemirovski and Todd (2008).

First, note that the Standard IPM problem requires an augmented form of the general NLP problem, such that all constraints are expressed as equalities with associated bound conditions. The inequality constraints (2.2b) can be reformulated as equalities by introducing slack variables. Let the slack variable s_i correspond to $g_i(x)$; then, (2.2b) can be equivalently expressed as:

$$g_i(\mathbf{x}) + s_i = 0 \quad \forall i \in \{1, \dots, m\} \quad (2.11a)$$

$$s_i \geq 0 \quad \forall i \in \{1, \dots, m\} \quad (2.11b)$$

If the direction of the original inequality constraint is reversed, an equivalent expression is obtained by subtracting the slack variables within the equality constraint. Having reformulated (2.2b), the standard IPM problem is defined:

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2.12a)$$

$$\text{s.t. } c(x) = 0 \quad (2.12b)$$

$$x \geq 0 \quad (2.12c)$$

Let $\mu > 0$ denote the barrier parameter, controlling a penalty size. Next, the problem (2.12) is reformulated as a barrier problem, where the bounds (2.12c) are replaced by a logarithmic barrier term that penalises proximity to the boundary. The barrier problem associated with (2.12) is:

$$\min_{x \in \mathbb{R}^n} \phi_\mu(x) := f(x) - \mu \sum_{i=1}^n \ln(x^{(i)}) \quad (2.13a)$$

$$\text{s.t. } c(x) = 0 \quad (2.13b)$$

Here, the term $\mu \sum_{i=1}^n \ln(x^{(i)})$ acts as a barrier near $x^{(i)} = 0$; as $x^{(i)}$ approaches zero from the interior of the feasible region, the barrier term goes to infinity, penalising the objective function and retaining a strictly interior search space.

IPMs generally solve a sequence of barrier problems (2.13) for decreasing μ . However, the Primal-Dual IPM avoids explicitly solving each barrier problem, instead approximating a solution to the perturbed KKT conditions:

$$\nabla f(x) + \nabla c(x)\lambda - z = 0 \quad (2.14a)$$

$$c(x) = 0 \quad (2.14b)$$

$$XZe - \mu e = 0 \quad (2.14c)$$

Here $\lambda \in \mathbb{R}^m$ are the Lagrangian multipliers for the equality constraint (2.12b) and $z \in \mathbb{R}^n$ are the Lagrangian multipliers for the inequality constraints (2.12c). Starting from an initial point (x_0, λ_0, z_0) , the algorithm approximates a solution to the system of nonlinear equations (2.14) using an adapted Newton-Rhapson method.

Following each iteration, μ is decreased, and the barrier problem is solved again using the previous approximation as the initial trial solution. Crucially, as iterations progress: μ converges to zero, the barrier term diminishes, the perturbed KKT conditions (2.14) better approximate the true KKT conditions, and the solution is guided toward the boundary where the true optimum may lie.

The optimality error associated with each system of KKT conditions (2.14) for a given μ is given by its set of norms. The optimality error for a given barrier problem, $E_\mu(x, \lambda, z)$, is therefore:

$$E_\mu(x, \lambda, z) := \max\{\|\nabla f(x) + \nabla c(x)\lambda - z\|, \|c(x)\|, \|XZe - \mu e\|\} \quad (2.15)$$

The algorithm terminates when the KKT conditions are satisfied with a given tolerance. Let ϵ_{tol} represent the tolerance threshold. Then, the algorithm terminates when:

$$E_\mu(x, \lambda, z) \leq \epsilon_{\text{tol}} \quad (2.16)$$

2.1.5 Selected solvers

The utility of the above algorithms depends on their computational implementation. Consequently, these algorithms are implemented in software packages known as solvers. This section reviews a selection of solvers relevant to this study.

2.1.5.1 The Ipopt solver

The Interior Point Optimiser (Ipopt) is an open-source software package for solving large-scale nonlinear optimisation problems (Wächter and Biegler, 2006). Provided that the objective function and constraints are twice continuously differentiable, it can address both convex and non-convex problems; however, in the latter case, its algorithms operate heuristically (Wächter and Biegler, 2006).

Ipopt primarily implements a primal–dual IPM (Wächter and Biegler, 2006). Among other adaptations to improve efficiency and robustness, it employs a filter line-search strategy to increase the likelihood of converging to a global optimum. The filter mechanism ensures that the trial step size α^k used to solve the perturbed KKT conditions (2.14) via Newton’s method (2.9) is accepted only if it improves either optimality or feasibility without worsening the other beyond a permissible threshold. This ensures that the algorithm progresses toward both goals simultaneously, prevents cycling between iterates that improve only one criterion, and increases the probability of escaping poor local optima (Wächter and Biegler, 2006).

2.1.5.2 The BONMIN solver

Basic Open-source Nonlinear Mixed Integer programming (solver) (BONMIN) is an open-source MINLP solver distributed under the Common Public Licence on the (COIN-OR) repository (Bonami, 2023). This licence permits free use, modification, and distribution of the software for both academic and commercial purposes (Lougee-Heimer, 2003), making it suitable for retailers seeking to implement the proposed model. BONMIN addresses MINLP problems of the form:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g^L \leq g(x) \leq g^U \end{aligned} \tag{2.17a}$$

$$x^L \leq x \leq x^U \tag{2.17b}$$

$$x \in \mathbb{R}^n, x_i \in \mathbb{Z} \ \forall i \in I \tag{2.17c}$$

where $f : \{x \in \mathbb{R}^n : x^L \leq x \leq x^U\} \rightarrow \mathbb{R}$ and $g : \{x \in \mathbb{R}^n : x^L \leq x \leq x^U\} \rightarrow \mathbb{R}^m$ are required to be twice differentiable functions, and $I \subseteq \{1, \dots, n\}$. Its algorithms are exact when f and g are convex; however, they operate heuristically when either or both functions are non-convex (Bonami, 2023).

While BONMIN offers multiple algorithmic configurations, its documentation (Bonami, 2023) recommends using a B&B algorithm to address integrality within the decision domain and solving each relaxed subproblem by invoking the Ipopt solver. Crucially, however, this approach is heuristically adapted as follows to improve the solution optimality:

- Because the IPM may converge to different local optima depending on its starting point, it is restarted several times at each B&B node with different random initialisations, retaining the best solution found.
- Since the IPM only guarantees local optimality, its solutions may not provide reliable bounds for pruning in B&B. To mitigate this, BONMIN relaxes the pruning rules, forcing further branching even when a node's objective value appears worse than the current best, thereby ensuring broader exploration of the solution space.

2.1.5.3 BARON and the NEOS server

BARON (Branch-And-Reduce Optimisation Navigator) is a global optimiser capable of computing exact solutions to complex problems, including non-convex MINLP problems (Sahinidis, 2017), by iteratively invoking local nonlinear solvers within a branch-and-cut framework (employing convex relaxations of non-convex terms) (Tawarmalani and Sahinidis, 2005). BARON is a commercial solver, distributed under a proprietary licence (Sahinidis, 2017); however, it can be accessed freely via the NEOS (Network-Enabled Optimisation System) Server, a free numerical optimisation service hosted by the Wisconsin Institute of Discovery (Czyzyk et al., 1998; Dolan, 2001; Gropp and Moré, 1997). The service allows users to submit AMPL jobs that can be solved using a selection of free and commercial solvers, enabling benchmarking of solutions across solvers.

2.2 Systematic literature review

2.2.1 SLR background and review methodology

An SLR is a structured approach to defining research questions and synthesising findings. Originally developed in the healthcare industry, SLRs gained popularity in the early 1990s when health practitioners adopted evidence-based practices that emphasised integrating the best available research into clinical practice (Dang et al., 2022). Over time, evidence-based research and systematic reviews have expanded across various fields (Kitchenham et al., 2009).

This SLR follows the approach initially outlined by Kitchenham (2004), which divides the review into three stages. First, the review protocol is developed. Second, studies of appropriate quality are identified. Third, the studies are evaluated against the review questions.

Figure 2.1 presents the review protocol. As described in the review workflow (see Figure 2.1A), the final two stages of the methodology are further subdivided into four steps. In step (1), the databases are searched according to a predefined search strategy (see Figure 2.1B). Additionally, screening criteria (1) and (2) are applied (refer to Figure 2.1 for the screening criteria). In step (2), the results are filtered to remove duplicate entries. In step (3), the screening criterion (4) is applied. In step (4), the content of the remaining studies is reviewed with respect to predefined research questions (see Figure 2.1D), which aim to evaluate the modelling approaches and scopes. Finally, note that the Sankey diagram in Figure 2.1(E) visualises the systematic filtering of the results.

2.2.2 SLR Results

2.2.2.1 Identified studies

The review protocol identifies ten studies, as listed in Table 2.1. The remainder of this subsection provides a summary of each study as it pertains to the current research.

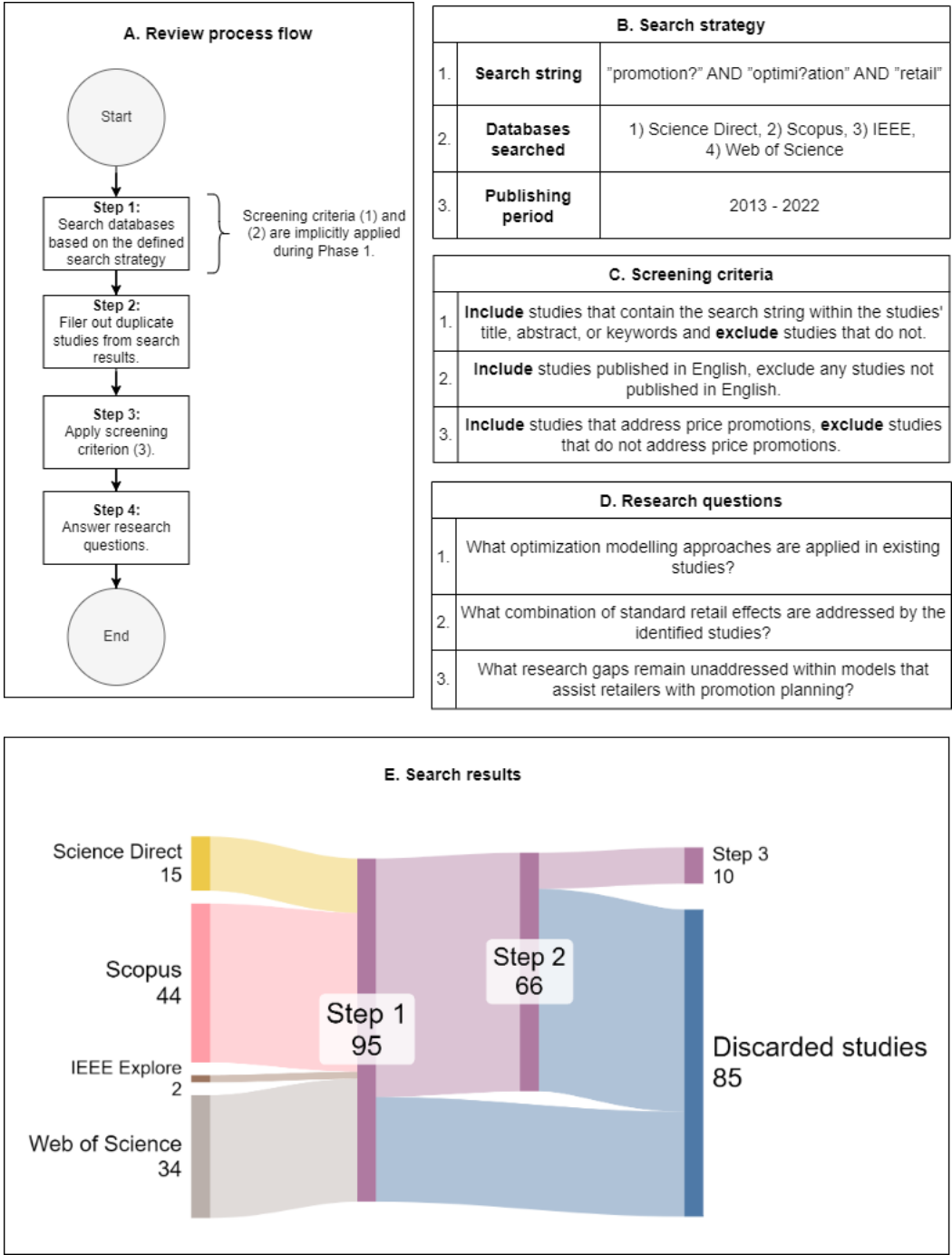


Figure 2.1: Systematic Literature Review protocol.

Table 2.1: The studies identified by the review

No.	Study title	Year	Author(s)
1	Optimising decisions of fresh-product members in daily and bourse markets considering the quantity and quality deterioration: a waste-reduction approach.	2021	Dolat-Abadi (2021)
2	Promotion Optimisation for Multiple Items in Supermarkets	2021	Cohen et al. (2021)
3	Simultaneous use of customer, product and inventory information in dynamic product promotion	2018	Mahar et al. (2018)
4	The Impact of Linear Optimization on Promotion Planning	2017	Cohen et al. (2017)
5	A retail store SKU promotions optimisation model for category multi-period profit maximisation	2017	Ma and Fildes (2017)
6	Price promotion with reference price effects in the supply chain	2016	Lin (2016)
7	Modelling of Product Sales Promotion and Price Discounting Strategy using Fuzzy Logic in a Retail Organisation	2016	Kumar et al. (2016)
8	Pricing and Ordering under Trade Promotion, Brand Competition and Demand Uncertainty	2016	Tsao (2016)
9	Redesigning promotion strategy for e-commerce competitiveness through pricing and recommendation	2015	Jiang et al. (2015)
10	Enhanced joint pricing and lot-sizing problem in a two-echelon supply chain with a logit demand function	2014	Yaghin et al. (2014)

Dolat-Abadi (2021) propose a coordinated strategy among key players in the fresh agricultural product supply chain to maximise retail promotional profit and minimise product waste. Retail price is modelled as a function of price elasticity, tax rates, product waste volume, resale price in secondary markets, stocking factor, and costs associated with wholesale, inventory shortages, and inventory surplus. The problem is formulated as a Stackelberg game, referring to a specific class of strategic games characterised by a hierarchical and sequential structure, wherein one player (the leader) makes a decision first, after which their competitor (the follower) observes the leader's decision and acts optimally (Simaan and Cruz Jr, 1973). In their implementation, Dolat-Abadi (2021) model the vendor as the leader and the retailer as the follower. The optimal retail price is determined through backwards induction, wherein each optimal decision, from the final to the initial decision node, is evaluated to identify the optimal strategy (Watson, 2002).

Cohen et al. (2021) builds on the earlier work by Cohen et al. (2017). While the predecessor focused on promoting a single fast-moving consumer good (FMCG), this study seeks to optimise retail promotions across multiple FMCGs. It develops a nonlinear integer programming model to set prices for various products, aiming to maximise retailer profit while adhering to specific business rules, including cross-item effects.

Retailers can strongly influence customer buying behaviour by employing real-time personalised promotions. This study optimises retail promotions by specifying the optimal mix and pricing of vendor items to include within real-time promotions. The optimal mix and pricing maximise retail profit, given the retailer's inventory levels and the customer's inherent preference towards a particular product. The authors formulate the problem as a mixed-integer nonlinear program.

Cohen et al. (2017) aim to optimise retail promotions by developing a nonlinear integer programming model to determine the optimal price for a single FMCG, given specific business rules.

Ma and Fildes (2017) proposes a nonlinear integer programming model to optimise retail promotions at the stock-keeping unit (SKU) level. The model aims to maximise promotional profit over a specified period by determining the optimal display level and discount depth.

Lin (2016) aims to determine the discount depth that maximises retail profit, considering the effects of trade, consumer, and vendor promotions within retail settings. A Stackelberg competition framework is used, with the vendor as the leader and the retailer as the follower. The optimal discount depth is derived through backwards induction.

Kumar et al. (2016) proposes a strategy to maximise retail profit from promotions. The authors develop a nonlinear programming model to determine the optimal order quantity for a retailer, considering the effects of trade spend. Metaheuristics are employed to determine the optimal discount rate that maximises profit.

Tsao (2016) incorporates the effects of trade spend within a nonlinear programming model that maximises retail profit during price promotions.

Electronic retailing (e-tailing) platforms often recommend products other than those listed during promotions, as customers are more likely to purchase these when the listed item is discounted. Jiang et al. (2015) proposes a nonlinear programming model that maximises overall e-tailer profit by determining the optimal discount depth for the listed product and the ideal mix of other products to recommend during promotions.

The frequency and size of orders, along with the retail price of items, significantly impact the combined profit of both the vendor and the retailer. Yaghin et al. (2014) proposes a nonlinear integer programming model in which these factors are treated as decision variables. The model aims to maximise the combined profit of the two-echelon supply chain.

2.2.2.2 Modelling approaches

Table 2.2 lists the optimisation modelling and solving approaches used within each identified study.

Table 2.2: The modelling approaches employed within the identified studies

No.	Author(s)	Modelling approach	Solving approach
1	Dolat-Abadi (2021)	Stackelberg competition	Backward induction
2	Cohen et al. (2021)	Nonlinear programming model	Linearisation
3	Mahar et al. (2018)	Nonlinear programming model	Metaheuristic
4	Cohen et al. (2017)	Nonlinear programming model	Linearisation
5	Ma and Fildes (2017)	Nonlinear programming model	Metaheuristic
6	Lin (2016)	Stackelberg competition	Backward induction
7	Kumar et al. (2016)	Nonlinear programming model	Heuristic
8	Tsao (2016)	Nonlinear programming model	Heuristic
9	Jiang et al. (2015)	Nonlinear programming model	Heuristic
10	Yaghin et al. (2014)	Nonlinear programming model	Metaheuristic

Dolat-Abadi (2021) and Lin (2016) model the problem as a Stackelberg competition, whereas the remaining studies employ nonlinear programming, indicating a clear precedence for the latter approach. Cohen et al. (2017) and Cohen et al. (2021) follow a linearisation approach to solve the problem. Kumar et al. (2016), Yaghin et al. (2014), Tsao (2016), Ma and Fildes (2017) employ metaheuristics to solve the problem. Mahar et al. (2018) employs BARON and Jiang et al. (2015) employs the NMaximise routine included in Wolfram's Mathematica software to solve the problem.

2.2.2.3 Study scopes

To maintain a holistic perspective, the discussion is limited to features that recur among the identified studies. These, in order of frequency of appearance, are: i) own-promotional effects (the effect the discount of an item has on its sales), ii) cross-promotional effects (the effect the discount of an item has on the sale of other items), iii) trade spend effects (the profit impact of trade spend from applying price promotions), and iv) stockpiling effects (the effect the discount of an item has on its future sales). Table 2.3 presents the occurrence of the features among the identified studies.

Notably, none of the identified models simultaneously address own-promotional, cross-promotional, and trade spend effects, nor do any encompass all the shortlisted features concurrently.

Table 2.3: The occurrence of common retail effects within the identified studies

No.	Author(s)	Effects			
		Own-promotional	Cross-promotional	Stockpiling	Trade spend
1	Dolat-Abadi (2021)	✓	✗	✗	✗
2	Cohen et al. (2021)	✓	✓	✓	✗
3	Mahar et al. (2018)	✓	✓	✗	✗
4	Cohen et al. (2017)	✓	✗	✓	✗
5	Ma and Fildes (2017)	✓	✓	✓	✗
6	Lin (2016)	✓	✗	✗	✓
7	Kumar et al. (2016)	✓	✗	✗	✓
8	Tsao (2016)	✓	✗	✗	✓
9	Jiang et al. (2015)	✓	✓	✗	✗
10	Yaghin et al. (2014)	✓	✗	✗	✓

2.3 Chapter summary

This chapter provides a detailed overview of optimisation theory and documents the SLR on existing models, assisting retailers with promotion planning. The chapter first describes the terminology central to mathematical optimisation and presents an overview of concepts relevant to model formulation, solution, and interpretation of solutions. The chapter then presents an SLR of existing models that assist retailers with promotion planning. It identifies 10 studies that assist retailers in planning promotions. Among them, all but two studies model the problem as a nonlinear programming model, underscoring the suitability of nonlinear programming for capturing the complexities inherent in retail promotion optimisation. The review also reveals that own-promotional, cross-promotional, and trade spend effects are commonly included within the identified studies, though no single study addresses these effects simultaneously. The next chapter formulates the optimisation model.

Chapter 3

Mathematical model development

The previous chapter provides a detailed overview of optimisation theory and documents the SLR of existing models that assist retailers with promotion planning. This chapter presents a new MINLP model that assists retailers in promotion planning while accounting for own-promotional, cross-promotional, and trade spend effects.

The chapter is organised into four sections. The first provides a conceptual overview of the problem and states key assumptions. The second provides the model notation, defining the sets, indices, parameters and variables. The third presents two original propositions to simplify formulations. The fourth presents the model formulation.

3.1 Conceptual problem overview and key assumptions

Consider a generic retailer planning for an upcoming promotional period. For strategic focus, the retailer limits its planning to a single product category, whose items may be sourced from multiple vendors. The retailer must determine i) the order quantity of each item, ii) the subset of items within the category to discount, and iii) the magnitude of each discount, such that its profit is maximised. Crucially, to ensure robust planning, the retailer must incorporate own-promotional, cross-promotional, and trade spend considerations within their decision-making. The remainder of this section states key assumptions underlying these components and presents a conceptual representation of the model.

3.1.1 Base own-promotional assumptions

To account for consumer response, the retailer must consider how pricing decisions impact demand. Given that it is standard practice for retailers to implement promotions using a discrete set of commonly used percentage discounts (Cohen et al., 2021, 2017; DelVecchio et al., 2007), the following is assumed: each item has a regular (non-discounted) retail price, and promotions are implemented by offering the item at a discounted price selected from

a discrete set of permissible fractions of that regular price. Furthermore, each discount option is associated with a distinct expected demand level and, in general, larger discounts yield greater demand (Blattberg et al., 1995; Van Heerde et al., 2001), reflecting consumer responsiveness to price reductions. Finally, as it is standard practice for retailers to order safety stock (i.e., additional inventory) to safeguard against stock-outs (Radasanu, 2016), it is assumed that each item's order quantity is inflated by a fixed percentage above its forecasted demand.

3.1.2 Base cross-promotional assumptions

The retailer must anticipate how discounts will affect consumer behaviour across the category by accounting for cross-promotional cannibalisation. As the cross-promotional cannibalisation rate depends on the degree of substitutability between items and the relative magnitude of their discounts (Gümüş et al., 2016), these factors are incorporated into the model's assumptions. Specifically, it is assumed that an item sold at its regular retail price imposes no cross-promotional effect on other items, whereas the cannibalisation effect intensifies with increasing discount depth. Additionally, cross-promotional cannibalisation rises as the substitutability between items increases.

3.1.3 Base trade spend considerations

Each item within the category is eligible for a distinct combination of three trade spend instruments offered by its vendor, namely off-invoice allowances, scan-backs, and buy-backs. For all three mechanisms, the trade spend value for a given item is calculated by multiplying its cost by the applicable trade spend rate. Although buy-back rates are predetermined and independent of retailer performance (Tsao, 2016), the assumption is that a tier-based system is used for off-invoice allowance and scan-back rate assignment. This assumption reflects the volume-dependent nature of these mechanisms, which follows directly from their definition in the problem background. Should a vendor not provide tier-based offerings, this approach remains applicable by defining a single tier. Furthermore, when no offering is made, the framework remains functional by setting the applicable rate to zero.

The applicable trade spend rates are determined as follows. Off-invoice allowances (payments based on the quantity of units ordered) follow a tiered structure, whereby the retailer qualifies for a specified allowance rate on an item only if the order quantity falls within vendor-defined thresholds. Scan-backs (providing payments based on the quantity of units sold during the promotional period) are organised according to a tiered structure, under which the retailer qualifies for a specific scan-back rate on an item only if its sales volume falls within vendor-defined thresholds. Finally, Buybacks (compensating the retailer for unsold stock returned to the vendor) apply a fixed per-item reimbursement rate.

3.1.4 Conceptual model representation

Figure 3.1 provides a conceptual representation of the proposed model. The figure links the own-promotional, cross-promotional, and trade spend considerations to the core nonlinear mixed-integer programming model, thereby providing a systematic optimisation approach for designing profit-maximising promotions. The figure lists the inputs informing the model, consisting of operational retail and trade spend data, and shows that it solves for the order quantity of each item, the subset of items within the category to discount, and the magnitude of each discount, resulting in the maximum profit being obtained.

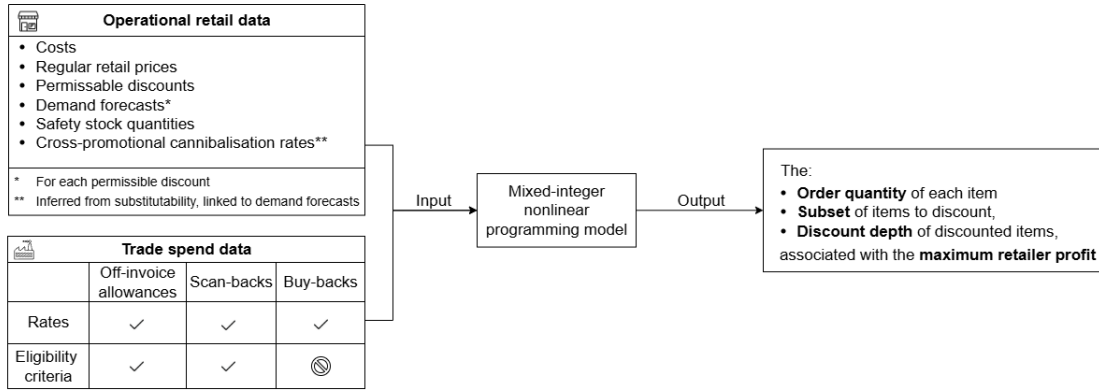


Figure 3.1: Conceptual representation of the proposed model for optimising discounting strategies given own-promotional, cross-promotional, and trade spend effects.

Note, the model variables are not limited to the outputs listed in Figure 3.1 as additional variables are introduced in Section 3.2.3. However, these serve an auxiliary role in supporting the model's internal logic and are thus non-pertinent to a conceptual problem overview.

3.2 Model notation

3.2.1 Set and index definitions

The following sets and their corresponding indices are defined:

- Let N denote the set of items stocked by the retailer within a single product category, with $i \in N$ indexing a generic item.
- Let K denote the set of allowable discounts for item i . Here, $k \in K$ indexes a generic discount.
- Let W denote the set of available off-invoice allowance and scan-back rates, with $w \in W$ indexing a generic trade spend rate.

3.2.2 Parameter definitions

The following parameters are defined:

- Let $c_i \in (0, \infty)$ denote the cost at which item i is procured from the vendor.
- Let $r_i \in (0, \infty)$ represent the regular (non-discounted) selling price of item i .
- Let $\eta_i \in [1, \infty)$ represent the order quantity ratio for item i , defined as the ratio of the ordered quantity (including safety stock) to the demand forecast. For instance, $\eta_i = 1.05$ indicates a 5% over-order relative to the forecasted demand for item i .
- The discount factor of k , i.e., the fraction of an item's regular selling price at which it is sold when discount k is applied, is represented by $q_k \in (0, 1]$. Note that the special case $q_k = 1$ implies the regular (non-discounted) selling price.
- Let $d_{ik} \in [0, \infty)$ denote the own-promotional demand for item i when discount rate k is applied.
- Let m represent the degree of substitutability between items within the product category, such that m is the number of sales of h beyond its regular demand level d_{h0} that correspond to a single lost unit of i .
- Let $\alpha_{hik} \in [0, \infty)$ represent the cross-promotional demand cannibalisation rate of item i when substitute h is offered at discount rate k . For the special case where $h = i$, let $\alpha_{h(i=h)k} = 0 \forall k \in K$ to reflect that a discounted item imposes no cross-promotional effects on itself. Note, Section 3.4.3 provides a method to estimate the α_{hik} parameters.
- Let $\lambda_i \in [0, 1]$ represent the buy-back rate offered on item i .
- Let $\Omega_{i vw} \in [0, 1]$ denote the off-invoice allowance ($v = 0$) or scan-back rate ($v = 1$) w the vendor offers on item i .
- Off-invoice allowance and scan-back rates are conditional on meeting vendor-defined quantity thresholds. For off-invoice allowances ($v = 0$), the relevant quantity is the order quantity, while for scan-backs ($v = 1$), it is the sale quantity. Each tiered rate is therefore associated with a lower and upper bound $L_{i vw}, U_{i vw} \in [0, \infty)$, such that $\Omega_{i vw}$ only applies if the quantity lies within the interval $[L_{i vw}, U_{i vw}]$. Note that the relationship between the boundary parameters implicitly requires $L_{i vw} < U_{i vw} \forall w \in W$.
- Two auxiliary parameters are defined to maintain logical consistency within the model, particularly in the assignment of trade spend rates:
 - Let Q be an arbitrary but sufficiently small value.
 - Let M be an arbitrary but sufficiently large value.

3.2.3 Variable definitions

3.2.3.1 Discount selection indicator (γ_{ik})

The binary discount selection indication $\gamma_{ik} \in \{0, 1\}$ indicates whether discount k is applied to item i , such that:

$$\gamma_{ik} = \begin{cases} 1 & \text{if item } i \text{ is discounted at } k \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

3.2.3.2 Sale price (p_i)

The sale price associated with the discount k , is $q_k r_i$. Further, all items must ultimately be sold at a single price, which implies:

$$\sum_{k \in K} \gamma_{ik} = 1 \quad \forall i \in N \quad (3.2)$$

It follows that the sale price of i is:

$$p_i = r_i \sum_{k \in K} q_k \gamma_{ik} \quad (3.3)$$

3.2.3.3 Cannibalised demand quantity (ϕ_i)

For any substitute item $h \in N \setminus \{i\}$, activating discount k (when $\gamma_{hk} = 1$) diverts a fraction α_{hik} of h 's demand d_{hk} from the demand for item i . In other words, the change in demand for i due to cross-promotional cannibalisation by h sold at k is:

$$\phi_{i \leftarrow h} = \alpha_{hik} d_{hk} \gamma_{hk} \quad (3.4)$$

Note, when item h is not sold at k , i.e., when $\gamma_{hk} = 0$, (3.4) evaluates to zero, reflecting the absence of cannibalisation effects. Given that (3.2) holds, it follows that the total demand loss for i due to cross-promotional cannibalisation is derived by aggregating (3.4) across all items $h \in N$ and all discount levels $k \in K$:

$$\phi_i = \sum_{h \in N} \sum_{k \in K} \alpha_{hik} d_{hk} \gamma_{hk} \quad (3.5)$$

3.2.3.4 Order quantity (O_i)

When item i is offered at discount k (i.e., when $\gamma_{ik} = 1$), its total demand comprises the own-promotional demand d_{ik} and the change in demand due to cross-promotional cannibalisation ϕ_i :

$$\gamma_{ik} (d_{ik} + \phi_i) \quad (3.6)$$

Note that (3.6) evaluates to zero when i is not offered at k (i.e., when $\gamma_{ik} = 0$). Given that (3.2) holds, (3.6) may be generalised across all permissible discount rates by summing over all $k \in K$:

$$\sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \quad (3.7)$$

The order quantity of i is obtained by scaling (3.7) with the order quantity ratio η_i , yielding:

$$O_i = \eta_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \quad (3.8)$$

3.2.3.5 Active trade spend rate ($\omega_{i vw}$)

Let $\omega_{i vw}$ be an active off-invoice allowance (when $v = 0$) or scan-back (when $v = 1$) rate applied to item i , such that:

$$\omega_{i vw} = \begin{cases} \Omega_{i vw} & \text{if eligible for rate } w \text{ (under instrument } v) \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

3.2.3.6 Trade spend eligibility variables ($x_{i vw}$ and $y_{i vw}$)

As noted in Section 3.2.2, the retailer qualifies for the off-invoice rate w on item i if the order quantity O_i satisfies condition (3.10). Likewise, to qualify for the scan-back rate w , the total demand for i (as defined in (3.7)) must satisfy condition (3.11).

$$L_{i0w} \leq O_i \leq U_{i0w} \quad (3.10)$$

$$L_{i1w} \leq \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \leq U_{i1w} \quad (3.11)$$

Two binary decision variables are used to model the retailer's eligibility for each rate. First, let $x_{ivw} = 1$ if and only if the retailer meets the lower trade spend requirement bound, such that:

$$x_{ivw} = \begin{cases} 1 & \text{if } \begin{cases} v = 0 \text{ and } L_{i0w} \leq O_i \\ v = 1 \text{ and } L_{i1w} \leq \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

Second, let $y_{ivw} = 1$ if and only if the retailer meets the upper trade spend requirement bound, such that:

$$y_{ivw} = \begin{cases} 1 & \text{if } \begin{cases} v = 0 \text{ and } O_i \leq U_{i0w} \\ v = 1 \text{ and } \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) \leq U_{i1w} \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

It follows that the active trade spend rate ω_{ivw} , originally defined in (3.9), may be reformulated as:

$$\omega_{ivw} = \begin{cases} \Omega_{ivw}, & \text{if } x_{ivw} = y_{ivw} = 1, \\ 0, & \text{otherwise} \end{cases} \quad (3.14)$$

3.3 Supporting propositions

MINLP models require that all relationships be expressed in algebraic form, using continuous and binary variables linked by equations and inequalities. Certain logical relationships, such as if-else conditions, cannot be directly expressed within this framework. Therefore, to model logical decisions, e.g. the variable assignments made in (3.12), (3.13), and (3.14), the relevant relationships must be reformulated into equivalent algebraic constraints. To this end, this section introduces two new propositions, i.e., logical statements that can be evaluated as true or false. Each proposition converts a logical if-else type rule into a system of mixed-integer linear constraints that the solver can process. N.B.: the proofs for propositions (1) and (2) are provided in Appendixes A.1 and A.2, respectively.

3.3.1 Proposition (1)

Proposition 1. *Let $a, b \geq 0$. Let $M > 0$ denote an arbitrarily large value and let $0 < Q < 1$ denote an arbitrarily small value, such that $Q < M$ and $a - b + Q < M$. Define the binary variable δ as follows:*

$$\delta = \begin{cases} 1 & \text{if } a \geq b \\ 0 & \text{if } a < b \end{cases} \quad (3.15)$$

Then the assignment in (3.15) is equivalently enforced by the following set of mixed-integer linear constraints:

$$a \geq b - M(1 - \delta) \quad (3.16a)$$

$$a + Q \leq b + \delta M \quad (3.16b)$$

$$\delta \in \{0, 1\} \quad (3.16c)$$

Given that $x_{ivw}, y_{ivw} \in \{0, 1\} \forall i \in N, \forall v \in V, \forall w \in W$, four key observations follow from the relationship between Proposition (1) and the variable assignments made in (3.12) and (3.13).

Observation (i)

The assignment made in (3.12), where:

$$x_{i0w} = 1 \begin{cases} 1 & \text{if } L_{i0w} \leq O_i \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, \forall w \in W$$

is an instance of (3.15) where $\delta = x_{i0w}$, $a = O_i$, and $b = L_{i0w}$.

Observation (ii)

The assignment made in (3.12), where:

$$x_{i1w} = 1 \begin{cases} 1 & \text{if } L_{i0w} \leq \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, \forall w \in W$$

is an instance of (3.15) where $\delta = x_{i1w}$, $a = \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i)$, and $b = L_{i1w}$.

Observation (iii)

The assignment made in (3.13), where:

$$y_{i0w} = \begin{cases} 1 & \text{if } O_i \leq U_{i0w} \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, \forall w \in W$$

is an instance of (3.15) where $\delta = y_{i0w}$, $a = U_{i0w}$, and $b = O_i$.

Observation (iv)

The assignment made in (3.13), where:

$$y_{i1w} = \begin{cases} 1 & \text{if } \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) \leq U_{i1w} \\ 0 & \text{otherwise} \end{cases}$$

is an instance of (3.15) where $\delta = y_{i1w}$, $a = U_{i1w}$, and $b = \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i)$.

It follows that (3.12) and (3.13) can be modelled using a system of mixed-integer linear constraints by making the appropriate substitutions into (3.16a - 3.16c), as demonstrated in Section 3.4.2.4.

3.3.2 Proposition (2)

Proposition 2. *Let $\delta_0, \delta_1 \in \{0, 1\}$, $b \geq 0$, and let $M > b$ represent an arbitrary large value. Define the decision variable a as follows:*

$$a = \begin{cases} b & \text{if } \delta_0 = \delta_1 = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.17)$$

Then, the assignment in (3.17) is equivalently enforced by the following set of mixed-integer linear constraints:

$$b - M(2 - \delta_0 - \delta_1) \leq a \leq b + M(2 - \delta_0 - \delta_1) \quad (3.18a)$$

$$-\delta_0 M \leq a \leq \delta_0 M \quad (3.18b)$$

$$-\delta_1 M \leq a \leq \delta_1 M \quad (3.18c)$$

$$\delta_0, \delta_1 \in \{0, 1\} \quad (3.18d)$$

It is observed that (3.14) is an instance of (3.17), where $\delta_0 = x_{ivw}$, $\delta_1 = y_{ivw}$, $a = \omega_{ivw}$, and $b = \Omega_{ivw}$. Thus, (3.14) can be modelled using a system of mixed-integer linear constraints by making the appropriate substitutions into (3.18a - 3.18d), as demonstrated in Section 3.4.2.5.

3.4 Mathematical model formulation

3.4.1 Objective function formulation

The objective is to maximise retail profit, defined as the difference between total income, comprising sales income I_s and trade spend income I_{ts} , and total expenses C . In other words, the objective is to maximise:

$$\text{Profit} = I_s + I_{ts} - C \quad (3.19)$$

Total sales income (I_s)

The total sales income for item i is the product of its price and its sales volume. The price associated with the k th discount is $q_k r_i$. As first established in (3.7), the total realised demand associated across $k \in K$ is $\sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$. It then follows that the total realised sales income for i across k is:

$$\begin{aligned} & \sum_{k \in K} r_i q_k \gamma_{ik} (d_{ik} + \phi_i) \\ \therefore r_i & \sum_{k \in K} q_k \gamma_{ik} (d_{ik} + \phi_i) \end{aligned} \quad (3.20)$$

To account for sale income from all items, (3.20) is summed over all $i \in N$:

$$I_s = \sum_{i \in N} r_i \sum_{k \in K} \gamma_{ik} q_k (d_{ik} + \phi_i) \quad (3.21)$$

Total trade spend income (I_{ts})

The vendor generates trade spend income through off-invoice allowances, scan-backs, and buy-backs.

First, under the vendor's buy-back programme, the retailer is refunded a fraction λ_i of the unit cost c_i , i.e., $\lambda_i c_i$ for every unsold unit of item i . The realised demand $\sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$ for i is first established in (3.7). Then, the inventory position for item i , offered at the discount level k , is determined by the difference between the order quantity and the realised demand, in other words, $O_i - \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$. It follows that the total income from buy-backs on item i is:

$$\lambda_i c_i \left(O_i - \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \right) \quad (3.22)$$

Second, under the vendor's off-invoice allowance programme, the retailer receives a discount, ω_{i0w} , on the cost c_i of each unit of i ordered. Equivalently, $c_i \omega_{i0w}$ is generated per unit of i ordered, following the full payment of i . Thus, the income generated through off-invoice allowances on i at the rate w is the product of the order quantity O_i and the off-invoice allowance per unit:

$$c_i O_i \omega_{i0w} \quad (3.23)$$

Third, the scan-back programme entitles the retailer to a rebate equal to a fraction ω_{i1w} of the unit cost c_i for each unit of item i sold, i.e., $\omega_{i1w} c_i$. The realised sales of i across all permitted discounts $\sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$ is first expressed in (3.7). Thus, the total scan-back income from i at the rate w is given by:

$$\omega_{i1w} c_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \quad (3.24)$$

Given that the total trade spend income is obtained by summing the income from each of the trade spend instruments, the total trade spend income from item i , given that it qualifies for the off-invoice allowance rate Ω_{i0w} and the scan-back rate Ω_{i1w} is obtained by summing (3.22), (3.23), and (3.24):

$$\lambda_i c_i (O_i - \gamma_{ik}(d_{ik} + \phi_i)) + c_i O_i \omega_{i0w} + \omega_{i1w} c_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \quad (3.25)$$

To account for income across all permissible off-invoice allowance and scan-back rates, (3.25) is summed over $w \in W$:

$$\begin{aligned} & \lambda_i c_i (O_i - \gamma_{ik}(d_{ik} + \phi_i)) + \sum_{w \in W} c_i O_i \omega_{i0w} + \sum_{w \in W} \omega_{i1w} c_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \\ \therefore & \lambda_i c_i (O_i - \gamma_{ik}(d_{ik} + \phi_i)) + c_i \sum_{w \in W} (O_i \omega_{i0w} + \omega_{i1w} \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)) \end{aligned} \quad (3.26)$$

Finally, to account for income across all items, (3.26) is summed over $i \in N$, obtaining the total trade spend income:

$$I_{ts} = \sum_{i \in N} \left(\lambda_i c_i (O_i - \gamma_{ik}(d_{ik} + \phi_i)) + c_i \sum_{w \in W} (O_i \omega_{i0w} + \omega_{i1w} \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)) \right) \quad (3.27)$$

Total costs (C)

Given that off-invoice allowances are already accounted for as an additional source of income following the full payment of all items (refer to (3.23), the total cost of item i is simply the product of the quantity ordered O_i , and the unit cost c_i ; in other words, $c_i O_i$. Accordingly, the total cost of all items $i \in N$ is given by the sum of the individual item costs:

$$C = \sum_{i \in N} c_i O_i \quad (3.28)$$

Complete objective function

Substituting the intermediate objective function variables (3.21), (3.27), and (3.28) into (3.19) yields the complete objective function:

$$\begin{aligned} \text{Profit} &= \sum_{i \in N} r_i \sum_{k \in K} \gamma_{ik} q_k(d_{ik} + \phi_i) \\ &+ \sum_{i \in N} \left(\lambda_i c_i (O_i - \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i)) + c_i \sum_{w \in W} (O_i \omega_{i0w} + \omega_{i1w} \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i)) \right) \\ &- \sum_{i \in N} c_i O_i \\ \therefore \text{Profit} &= \sum_{i \in N} \left(r_i \sum_{k \in K} \gamma_{ik} q_k(d_{ik} + \phi_i) + \lambda_i c_i (O_i - \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i)) \right) \end{aligned}$$

$$+ c_i \sum_{w \in W} (O_i \omega_{i0w} + \omega_{i1w} \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)) - c_i O_i \quad (3.29)$$

3.4.2 Constraint formulations

3.4.2.1 Single price assignment constraint

Recall that the binary discount selection indicator γ_{ik} equals one if and only if item i is discounted at level k , and that only one discount must ultimately be applied to each item, as established in (3.2). It follows that enforcing the sum of γ_{ik} over all $k \in K$ to equal one for each item $i \in N$, ensures that at most one discount level is selected per item. Thus, Constraint (3.30) ensures each assumes only a single discount level (and price):

$$\sum_{k \in K} \gamma_{ik} = 1 \quad \forall i \in N \quad (3.30)$$

3.4.2.2 Demand cannibalisation constraint

Equation (3.5) establishes the total demand loss for item i due to cross-promotional cannibalization is $\phi_i = \sum_{h \in N} \sum_{k \in K} \alpha_{hik} d_{hk} \gamma_{hk}$. Thus, Constraint (3.31) calculates the cross-promotional demand cannibalisation (3.5) for each item $i \in N$:

$$\phi_i = \sum_{h \in N} \sum_{k \in K} \alpha_{hik} d_{hk} \gamma_{hk} \quad \forall i \in N \quad (3.31)$$

3.4.2.3 Order quantity assignment constraint

Equation (3.8) defines the order quantity of item i as $O_i = \eta_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$. It follows that Constraint (3.32) calculates the order quantity (3.8) for each item $i \in N$:

$$O_i = \eta_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \quad \forall i \in N \quad (3.32)$$

3.4.2.4 Off-invoice allowance and scan-back eligibility constraints

Proposition (1) is utilised in formulating the off-invoice and scan-back eligibility constraints using a system of mixed-integer linear constraints. To this end, the four observations noted in Section 3.3.1 are key in applying the appropriate substitutions:

- It follows from observation (i) that the combination of (3.33a) and (3.33b) models the off-invoice allowance eligibility assignment of (3.12). In other words, the following constraints assign $x_{i0w} = 1$ if and only if the order quantity of i is sufficiently large

to qualify for the off-invoice allowance rate w ,

$$O_i \geq L_{i0w} - M(1 - x_{i0w}) \quad \forall i \in N, \forall w \in W \quad (3.33a)$$

$$O_i + Q \leq L_{i0w} + Mx_{i0w} \quad \forall i \in N, \forall w \in W \quad (3.33b)$$

- It follows from observation (ii) that the combination of (3.34a) and (3.34b) models the scan-back eligibility assignment of (3.12). In other words, the following constraints assign $x_{i1w} = 1$ if and only if the sale volume of i is sufficiently large to qualify for the scan-back rate w .

$$\sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) \geq L_{i1w} - M(1 - x_{i1w}) \quad \forall i \in N, \forall w \in W \quad (3.34a)$$

$$\sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) + Q \leq L_{i1w} + M(x_{i1w}) \quad \forall i \in N, \forall w \in W \quad (3.34b)$$

- It follows from observation (iii) that the combination of (3.35a) and (3.35b) models the off-invoice allowance rate assignment of (3.13). In other words, the following constraints assign $y_{i1w} = 1$ if and only if the order quantity of i is sufficiently small to qualify for the scan-back off-invoice allowance rate w .

$$U_{i0w} \geq O_i - M(1 - y_{i0w}) \quad \forall i \in N, \forall w \in W \quad (3.35a)$$

$$U_{i0w} + Q \leq O_i + y_{i0w}M \quad \forall i \in N, \forall w \in W \quad (3.35b)$$

- It follows from observation (iv) that the combination of (3.36a) and (3.36b) models the scan-back rate assignment of (3.13). In other words, the following constraints assign $y_{i1w} = 1$ if and only if the sale volume of i is sufficiently small to qualify for the scan-back rate w .

$$U_{i1w} \geq \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) - M(1 - y_{i1w}) \quad \forall i \in N, \forall w \in W \quad (3.36a)$$

$$U_{i1w} + Q \leq \sum_{k \in K} \gamma_{ik}(d_{ik} + \phi_i) + y_{i1w} M \quad \forall i \in N, \forall w \in W \quad (3.36b)$$

3.4.2.5 Trade spend rate assignment constraints

Proposition (2) is utilised in formulating the trade spend rate assignment constraint using a system of mixed-integer linear constraints. It follows from Section 3.3.2, that the combination of (3.37a - 3.37d) models the trade spend rate assignment (3.14). In other words, the following set of constraints assign $\omega_{iww} = \Omega_{iww}$ if and only if $x_{iww} = y_{iww} = 1$:

$$\Omega_{iww} - M(2 - x_{iww} - y_{iww}) \leq \omega_{iww} \quad \forall i \in N, \forall v \in V, \forall w \in W \quad (3.37a)$$

$$\Omega_{iww} + M(2 - x_{iww} - y_{iww}) \geq \omega_{iww} \quad \forall i \in N, \forall v \in V, \forall w \in W \quad (3.37b)$$

$$-x_{iww} M \leq \omega_{iww} \leq x_{iww}M \quad \forall i \in N, \forall v \in V, \forall w \in W \quad (3.37c)$$

$$-y_{iww} M \leq \omega_{iww} \leq y_{iww}M \quad \forall i \in N, \forall v \in V, \forall w \in W \quad (3.37d)$$

3.4.2.6 Sale price calculation constraint (3.38)

The price of item i when discounted at k , as first formulated in (3.3), is $p_i = r_i \sum_{k \in K} q_k \gamma_{ik}$. It follows that constraint (3.38) calculates the selling price of each item by calculating (3.3) for every item $i \in N$:

$$p_i = \sum_{k \in K} q_k r_i \gamma_{ik} \quad \forall i \in N \quad (3.38)$$

3.4.2.7 Binary decision variable constraints (C.10)

To ensure that γ_{ik} , x_{ivw} , and y_{ivw} are correctly modelled as binary decision variables, the following constraints are imposed:

$$\gamma_{ik} \in \{0, 1\} \quad \forall i \in N, \forall k \in K, \quad (3.39)$$

$$x_{ivw}, y_{ivw} \in \{0, 1\} \quad \forall i \in N, \forall v \in V, \forall w \in W \quad (3.40)$$

3.4.3 An estimation method for cross-promotional demand cannibalisation (α_{hik})

The increase in demand for item $h \in N, h \neq i$ when sold at a discount rate $k \in K$ compared to its regular price d_{h0} is $d_{hk} - d_{h0}$. Additionally, if every m sales of h beyond its regular demand prevent the sale of a substitute, item i , the demand reduction for i due to cross-promotional cannibalisation by h is given by (3.41), where the negative sign indicates a decrease in demand:

$$\phi_{i \leftarrow h} = -\frac{d_{hk} - d_{h0}}{m} = \frac{d_{h0} - d_{hk}}{m} \quad (3.41)$$

Equation (3.4) establishes that $\phi_{i \leftarrow h}$ may be expressed as a fraction α_{hik} of h 's demand when sold at k . In the following, (3.4) is equated to (3.41) and solved to obtain α_{hik} :

$$\begin{aligned} \phi_{i \leftarrow h} &= \alpha_{hik} d_{hk} \gamma_{hk} = \frac{d_{h0} - d_{hk}}{m} \\ \alpha_{hik} d_{hk} (1) &= \frac{d_{h0} - d_{hk}}{m} \\ \therefore \alpha_{hik} &= \frac{d_{h0} - d_{hk}}{m d_{hk}} \quad (\text{when } h \neq i) \end{aligned} \quad (3.42)$$

To generalise for the case where $h = i$, given that an item imposes no cross-promotional effects on itself, $\alpha_{hik} = 0$ when $h = k$.

In summary, the cross-promotional demand cannibalisation rate of i when substitute h is offered at k is estimated as:

$$\alpha_{hik} \approx \begin{cases} 0 & \text{when } h = i \\ \frac{d_{h0} - d_{hk}}{m d_{hk}} & \text{otherwise} \end{cases} \quad (3.43)$$

Crucially, (3.43) provides a systematic approach for estimating cross-promotional demand cannibalisation, incorporating the degree of substitutability between items and the relative magnitude of their discounts.

3.5 Chapter summary

This chapter formulates the optimisation model to assist retailers with promotion planning. Starting from a conceptual representation of the problem and base assumptions, the model notation is presented and the formulation is provided for a MINLP model that solves for i) the order quantity of each item within a given product category, ii) the subset of items within the category to discount, and iii) the size of each discount resulting in the maximum profit being obtained. The objective of the model is to maximise (3.29), subject to constraints (3.30 - 3.40). Given the findings of the SLR, the model notably contributes to the literature by incorporating own-promotional, cross-promotional, and trade spend effects simultaneously.

Chapter 4

Verification

The previous chapter proposes a model to assist retailers with promotion planning. This chapter documents the model’s verification. As discussed in Section 1.3, verification comprises two stages:

- Further analysis employs the BONMIN solver, which, as discussed in Section 2.1.5.2, requires the objective function and constraints to be twice continuously differentiable (Bonami, 2023). To verify the proposed model’s adherence to the requirements of the solving approach, its mathematical structure is examined to confirm compliance. This verification confirms the solver’s correct use.
- To verify the correctness of the formulation and coding of the proposed model, the model is assessed under different inputs to verify the correctness of its outputs. This verification confirms that the model correctly implements the intended conceptual design (including its key assumptions and simplifications) from the previous chapter.

BONMIN’s use in further analysis is motivated by three considerations. First, BONMIN can be configured to produce reasonably good heuristic solutions to possibly non-convex problems (Bonami, 2023). Second, unlike many commercial global solvers such as BARON or Gurobi, BONMIN’s licence permits free use (Bonami, 2023), third BONMIN can be invoked natively within a Python and AMPL-based input-output system (AMPL Optimization Inc., 2021), avoiding the relatively manual input-output handling associated with solvers on the NEOS server (Dolan, 2001).

The chapter is structured according to the two verification stages. The first section verifies the proposed model’s adherence to the requirements of the solving approach, and the second section verifies the correctness of the model’s formulation and implementation.

4.1 Compliance with solver requirements

As discussed in Section 2.1.5.2, the objective function and constraints must be twice continuously differentiable within the continuously relaxed domains of the branch-and-bound subproblems to comply with BONMIN's requirements (Bonami, 2023). To this end, a key foundational property of polynomial functions is exploited: multivariate polynomials are infinitely differentiable over $\mathbb{R}^n$ because their partial derivatives of all orders exist and remain polynomials (which are themselves continuous) (Stewart, 2016).

Consider the continuous relaxation of the model, wherein the variables are bound such that $\gamma_{ik}, x_{i0w}, y_{i0w} \in [0, 1]$ and $\phi_i, O_i, \omega_{i0w} \in [0, \infty)$. The objective function (3.29) is a multivariate polynomial composed of the finite sums of linear (e.g., γ_{ik}, O_i , etc.), quadratic (e.g., $\gamma_{ik}\phi_i, O_i\omega_{i0w}, \omega_{i1w}\gamma_{ik}$, etc.), and cubic terms (e.g., $\omega_{i1w}\gamma_{ik}\phi_i$). As polynomials are infinitely differentiable, the objective function (3.29) is (at least) twice continuously differentiable over the integer-relaxed decision domain.

The polynomial argument is similarly applied to the constraints. Table 4.1 expresses each constraint in its equivalent standard formulation, i.e., according to the form presented in 2.17. The table demonstrates that each constraint is either linear or quadratic, implying that all constraints are twice continuously differentiable over the integer-relaxed decision domain.

Table 4.1: Standard BONMIN formulations of model constraints

Constraint	g formulation	g_L	g_U
(3.30)	$g_i := \sum_{k \in K} \gamma_{ik} - 1$	0	0
(3.31)	$g_i := \phi_i - \sum_{h \in N} \sum_{k \in K} \alpha_{hik} d_{hk} \gamma_{hk}$	0	0
(3.32)	$g_i := O_i - \eta_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$	0	0
(3.33a)	$g_{i0w} := L_{i0w} - M(1 - x_{i0w}) - O_i$	$-\infty$	0
(3.33b)	$g_{i0w} := L_{i0w} + Mx_{i0w} - O_i - Q$	0	∞
(3.34a)	$g_{i1w} := L_{i1w} - M(1 - x_{i1w}) - \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$	$-\infty$	0
(3.34b)	$g_{i1w} := \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) + Q - L_{i1w} - Mx_{i1w}$	$-\infty$	0
(3.35a)	$g_{i0w} := O_i - M(1 - y_{i0w}) - U_{i0w}$	$-\infty$	0
(3.35b)	$g_{i0w} := U_{i0w} + Q - O_i - My_{i0w}$	$-\infty$	0
(3.36a)	$g_{i1w} := \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) - M(1 - y_{i1w}) - U_{i1w}$	$-\infty$	0
(3.36b)	$g_{i1w} := U_{i1w} + Q - \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) - My_{i1w}$	$-\infty$	0
(3.37a)	$g_{i0w} := \Omega_{i0w} - M(2 - x_{i0w} - y_{i0w}) - \omega_{i0w}$	$-\infty$	0
(3.37b)	$g_{i0w} := \omega_{i0w} - \Omega_{i0w} - M(2 - x_{i0w} - y_{i0w})$	$-\infty$	0
(3.37c)	$g_{i0w} := -\omega_{i0w} - x_{i0w}M$	$-\infty$	0
	$g_{i0w} := \omega_{i0w} - x_{i0w}M$	$-\infty$	0
(3.37d)	$g_{i0w} := -\omega_{i0w} - y_{i0w}M$	$-\infty$	0
	$g_{i0w} := \omega_{i0w} - y_{i0w}M$	$-\infty$	0
(3.37b)	$g_i := p_i - \sum_{k \in K} q_k r_i \gamma_{ik}$	0	0

To conclude, the model's objective function and constraints are twice continuously differentiable over the continuously relaxed decision domain. Thereby, the model's adherence to the solving approach is verified.

Table 4.2: Mapping of each verification assessment to the corresponding model formulation and AMPL coding.

Assessment no.	Formulation(s) assessed	AMPL listing
I	The objective function (3.29)	B.1
II	Constraints (3.30) and (3.39)	B.2
III	Constraint (3.31)	B.3
IV	Constraint (3.32)	B.4
V	Constraints (3.33a - 3.36b) and (3.40)	B.5
VI	Constraints (3.37a - 3.37d)	B.6
VII	Constraint (3.38)	B.7

4.2 Correctness of the formulation and coding

4.2.1 Verification approach

Seven AMPL models are constructed. Each isolates the functionality of the objective function, a constraint, or a set of constraints. The AMPL models are presented as listings in Appendix B.1. Table 4.2 summarises the mapping between each verification assessment, the corresponding model formulation, and its AMPL implementation. Each model is solved with BONMIN using trivial inputs. The solver output is then compared with the analytical results.

Given that the output is mathematically correct for a given assessment, the relevant formulation and coding are verified as correct. Accordingly, the formulation and coding of the complete model, obtained by aggregating the relevant formulations from the verification models, are systematically verified to be correct.

4.2.2 Assessment I

The objective function is implemented in an AMPL model, presented in Listing B.1. The instance is solved using the following input:

$$\begin{aligned}
 N = K = W &= \{0, 1\} \\
 c_i = r_i = \lambda_i = \phi_i = O_i &= 1 & \forall i \in N \\
 d_{ik} = \gamma_{ik} &= 1 & \forall i \in N, \forall k \in K \\
 \omega_{i0w} = 1 & & \forall i \in N, \forall w \in \{0, 1\}, \forall w \in W
 \end{aligned}$$

The solver computed the objective function value of 20 units, which equals the analytical result, derived in (4.1):

$$\begin{aligned}
 \text{Profit} = & \sum_{i \in N} \left(r_i \sum_{k \in K} \gamma_{ik} q_k(d_{ik} + \phi_i) + \lambda_i c_i (O_i - \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)) \right. \\
 & \left. + c_i \sum_{w \in W} (O_i \omega_{i0w} + \omega_{i1w} \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)) - c_i O_i \right)
 \end{aligned}$$

$$\begin{aligned}
\text{Profit} &= \sum_{i \in N} \left((1) \sum_{k \in K} (1) (1)(1+1) + (1) (1) \left(1 - \sum_{k \in K} (1)(1+1) \right) \right. \\
&\quad \left. + (1) \sum_{w \in W} \left((1) (1) + (1) \sum_{k \in K} (1)(1+1) \right) - (1) (1) \right) \\
\text{Profit} &= \sum_{i \in N} \left(\sum_{k \in K} 2 + 1 - \sum_{k \in K} 2 + \sum_{w \in W} \left(1 + \sum_{k \in K} 2 \right) - 1 \right) \\
\text{Profit} &= \sum_{i \in N} \left(\sum_{w \in W} \left(1 + \sum_{k \in K} 2 \right) \right) \\
\text{Profit} &= 2 \left(2(1 + (2)(2)) \right) \\
\therefore \text{Profit} &= 20
\end{aligned} \tag{4.1}$$

Since the output is correct, the formulation and coding of the objective function (3.29) are verified as correct.

4.2.3 Assessment II

Constraints (3.30) and (3.39) are implemented in an AMPL model, presented in Listing B.2. The model's objective function is constant (i.e., 1), and thus, inconsequential to the computation. As input, the sets are defined as:

$$N = K = \{0, 1, 2, 3, 4\}$$

Constraint (3.30) must ensure that only binary assignments are made for y_{ik} . Additionally, (3.39) must ensure that exactly one non-zero assignment is made for γ_{ik} for each item. The solver computed the following non-zero values for γ_{ik} :

$$\gamma_{00} = \gamma_{10} = \gamma_{20} = \gamma_{30} = \gamma_{40} = 1$$

From the output, it is evident that the solver exclusively computed binary values for γ_{ik} , and exactly one $\gamma_{ik} = 1$ was computed for each $i \in N$, matching the analytical results. Since the output is correct, the formulation and coding of (3.30) and (3.39) are verified as correct.

4.2.4 Assessment III

Constraint (3.31) is implemented in an AMPL model, presented in Listing B.3. The model's objective function is constant, and thus does not affect the computation. As input, the following definitions and assignments are made:

$$\begin{aligned}
N = K &= \{0, 1\} \\
d_{00} &= 100, \quad d_{01} = 200, \quad d_{10} = 110, \quad d_{11} = 220
\end{aligned}$$

$$\alpha_{011} = \alpha_{101} = -0.25, \quad \alpha_{000} = \alpha_{001} = \alpha_{010} = \alpha_{100} = \alpha_{110} = \alpha_{111} = 0$$

$$\gamma_{00} = \gamma_{10} = 0 \quad \gamma_{01} = \gamma_{11} = 1$$

Constraint (3.31) computes the cross-promotional demand cannibalisation quantities, ϕ_i . For the given input, the solver computed the following values for ϕ_i :

$$\phi_0 = -55$$

$$\phi_1 = -50$$

Given the input, the cross-promotional demand cannibalisation variables, as assigned by (3.5), evaluate to:

$$\begin{aligned} \phi_i &= \sum_{h \in N} \sum_{k \in K} \alpha_{hik} d_{hk} \gamma_{hk} & (3.5) \\ \phi_i &= \sum_{h \in N} (\alpha_{hi0} d_{h0} \gamma_{h0} + \alpha_{hi1} d_{h1} \gamma_{h1}) \\ \phi_i &= \alpha_{0i0} d_{00} \gamma_{00} + \alpha_{0i1} d_{01} \gamma_{01} + \alpha_{1i0} d_{10} \gamma_{10} + \alpha_{1i1} d_{11} \gamma_{11} \\ \phi_i &= \alpha_{0i0} (100) (0) + \alpha_{0i1} (200) (1) + \alpha_{1i0} (110) (0) + \alpha_{1i1} (220) (1) \\ \phi_i &= 200 \alpha_{0i1} + 220 \alpha_{1i1} \\ \therefore \phi_0 &= 200(0) + 220(-0.25) = -55 & (4.2) \\ \therefore \phi_1 &= 200(-0.25) + 220(0) = -50 & (4.3) \end{aligned}$$

Since the solver output matches the analytical results derived in (4.2) and (4.3), the formulation and coding of constraint (3.31) are verified as correct.

4.2.5 Assessment IV

Constraint (3.32) is implemented in an AMPL model, presented in Listing B.4. The model's objective function is constant and does not influence the computation. The following inputs are used:

$$\begin{aligned} N &= K = \{0, 1\} \\ \eta_0 &= \eta_1 = 1.05 \\ d_{00} &= 100, \quad d_{01} = 200, \quad d_{10} = 110, \quad d_{11} = 220 \\ \gamma_{00} &= \gamma_{10} = 0, \quad \gamma_{01} = \gamma_{11} = 1 \\ \phi_0 &= 30 \quad \phi_1 = 20 \end{aligned}$$

Constraint (3.32) computes the order quantities O_i . For the given input, the solver computed the following O_i values:

$$O_0 = 241.5$$

$$O_1 = 252.0$$

The order quantities that follow from the input, as assigned by (3.8), are calculated as follows:

$$O_i = \eta_i \sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i) \quad (3.8)$$

$$O_i = \eta_i (\gamma_{i0} (d_{i0} + \phi_i) + \gamma_{i1} (d_{i1} + \phi_i))$$

$$O_i = \eta_i (\gamma_{i0} (d_{i0} + \phi_i) + \gamma_{i1} (d_{i1} + \phi_i))$$

$$\therefore O_0 = 1.05(0 (100 + 30) + 1 (200 + 30)) = 241.5 \quad (4.4)$$

$$\therefore O_1 = 1.05(0 (110 + 20) + 1 (220 + 20)) = 252 \quad (4.5)$$

Since the output matches the analytical results derived in (4.4) and (4.5), the formulation and coding of (3.32) are verified as correct.

4.2.6 Assessment V

Constraints (3.33a – 3.36b) and (3.40) are implemented in an AMPL model, presented in Listing B.5. These constraints are verified collectively, as they jointly govern the assignment of the trade spend eligibility variables x_{ivw} and y_{ivw} . The model's objective function is constant and does not influence the computation. This model is solved across ten instances targeting edge cases. Each instance assesses the solver's output for the trade spend eligibility variables x_{ivw} and y_{ivw} , given a selection of order-quantity and sales-volume inputs. The following inputs are used across all instances:

$$N = K = W = \{0\}$$

$$M = 8000$$

$$Q = 0.1$$

$$L_{000} = L_{010} = U_{000} = U_{010} = 10$$

$$\phi_0 = 0$$

$$\gamma_{00} = 1$$

Since $\phi_0 = 0$ and $\gamma_{ik} = 1$ across all instances, the demand parameter serves as a proxy for the sales volume $\sum_{k \in K} \gamma_{ik} (d_{ik} + \phi_i)$, and as such, d_{ik} is modified to enact the appropriate instance.

Table 4.3 summarises the test instances. The left columns lists the instance-specific inputs of each instance. The middle columns presents the values of x_{ivw} and y_{ivw} derived analytically from (3.12) and (3.13) in each instance. The right column in Table 4.3 presents computed values of x_{ivw} and y_{ivw} .

From Table 4.3, it is evident that the solver output matched the analytical results across all instances, verifying the correctness of the formulation and coding of constraints (3.33a)-3.36b) and (3.40).

Table 4.3: The test instances for (C4- C7) verification

Instance definition			Analytical values				Solver output			
Instance	O_i	$\sum_{k \in K} (d_{ik} + \phi_i)$	x_{i0w}	y_{i0w}	x_{i1w}	y_{i1w}	x_{i0w}	y_{i0w}	x_{i1w}	y_{i1w}
(i)	9.0	0	0	1	0	1	0	1	0	1
(ii)	9.9	0	0	1	0	1	0	1	0	1
(iii)	10.0	0	1	1	0	1	1	1	0	1
(iv)	10.1	0	1	0	0	1	1	0	0	1
(v)	20.0	0	1	0	0	1	1	0	0	1
(vi)	0	9.0	0	1	0	1	0	1	0	1
(vii)	0	9.9	0	1	0	1	0	1	0	1
(viii)	0	10.0	0	1	1	1	0	1	1	1
(ix)	0	10.1	0	1	1	0	0	1	1	0
(x)	0	20.0	0	1	1	0	0	1	1	0

4.2.7 Assessment VI

Constraints (3.37a - 3.37d) are implemented in an AMPL model, presented in Listing B.6. The model's objective function is constant and does not influence the computation. Verification is conducted across eight instances to cover all edge cases. The following parameters are used across all instances:

$$N = K = W = \{0\}$$

$$M = 8000$$

$$\Omega_{000} = 0.1 \quad \Omega_{010} = 0.2$$

Constraints (3.37a - 3.37d) compute the values of $\omega_{i vw}$ according to (3.14). Table 4.4 summarises the test instances, showing instance-specific inputs in the left columns, analytically derived values of $\omega_{i vw}$ in the middle columns, and solver-computed outputs in the right columns. It is evident from Table 4.4 that the solver output matches the analytical

Table 4.4: The test instances for (3.37a - 3.37d) verification

Instance definition					Analytical values		Solver output	
No.	x_{i0w}	y_{i0w}	x_{i1w}	y_{i1w}	ω_{i0w}	ω_{i1w}	ω_{i0w}	ω_{i1w}
i	1	1	0	0	0.1	0	0.1	0
ii	1	0	1	0	0	0	0	0
iii	0	1	0	1	0	0	0	0
iv	0	0	1	1	0	0.2	0	0.2
v	1	1	1	1	0.1	0.2	0.1	0.2

values across all instances. Thus, the formulation and coding of (3.37a - 3.37d) are verified as correct.

4.2.8 Assessment VII

Constraint (3.38) is implemented in an AMPL model, presented in Listing B.7. The model's objective function is constant and does not influence the computation. As input, the following definitions and assignments are made:

$$\begin{aligned} N, K &= \{0, 1\} \\ q_0 &= 1, \quad q_1 = 0.3 \\ r_0 &= 10, \quad r_1 = 20 \\ \gamma_{00} &= \gamma_{11} = 1, \quad \gamma_{01} = \gamma_{10} = 0 \end{aligned}$$

Constraint (3.38) computes the prices p_i . For the given input, the solver computed the following prices:

$$\begin{aligned} p_0 &= 10 \\ p_1 &= 6 \end{aligned}$$

The values of p_i , which follow analytically from (3.3) are calculated as follows:

$$p_i = r_i \sum_{k \in K} q_k \gamma_{ik} \tag{3.3}$$

$$\begin{aligned} p_i &= r_i (q_0 \gamma_{i0} + q_1 \gamma_{i1}) \\ \therefore p_0 &= 10((1)(1) + (0.3)(0)) = 10 \end{aligned} \tag{4.6}$$

$$\therefore p_1 = 20((1)(0) + (0.3)(1)) = 6 \tag{4.7}$$

Since the solver output matches the analytical values derived in (4.6) and (4.7), the formulation and coding of (3.38) are verified as correct.

4.2.9 Formulation and coding conclusions

The solver produced the correct output across all seven assessments, thereby verifying the correctness of the objective function and constraints. This verification confirms that the model's core logic operates in accordance with the conceptual framework and the limitations specified in the previous chapter. It follows that the formulation and coding of the complete operational model, obtained by aggregating the relevant formulations from the assessment models, are also verified as correct. The complete operational, used in further analysis, is presented in Listing B.8.

4.3 Chapter summary

This chapter documents verification. First, the model's adherence to the requirements of the solving approach is verified by ensuring that the objective functions and constraints are twice continuously differentiable over the integer-relaxed decision domain. Second, the chapter verifies the correctness of the formulation and implementation of the proposed model by demonstrating that the solver's output is correct across seven assessments of the objective function and constraints. Taken together, these results provide evidence of the correctness of the solver outputs obtained in further analysis. The next chapter documents validation.

Chapter 5

Validation

The previous chapter documents verification. This chapter documents validation. As discussed in Chapter 1, validation comprises three stages:

- As stated in Section 1.3, the optimality of the heuristic solutions produced by BONMIN is not guaranteed (Bonami, 2023). To validate the reliability of the solutions produced by the solving approach, a benchmark analysis is conducted to determine the optimality of the heuristic solutions produced for a set of test problems.
- To validate the model’s ability to reflect realistic behaviour, a sensitivity analysis is conducted wherein the response in profitability is analysed in terms of theoretical alignment and internal consistency across variations in key parameters.
- To validate the computability of the solving approach, a scalability analysis is conducted wherein a relationship is established between computation time and problem size.

Each of these analyses relies on simulated data. The use of simulated data is motivated by practical and methodological considerations. While operational retail datasets are available and frequently employed in related research, no single dataset has been identified that contains all the features required for the model’s implementation, particularly detailed information on trade spend. Furthermore, empirical retail data presents the challenge of constructing a suitably accurate demand model (which is outside the scope of this study). Simulation, therefore, provides a pragmatic alternative for generating operational data for model experimentation, and has likewise been employed in prior research on the promotion planning problem (see Kumar et al. (2016); Cohen et al. (2017, 2021)).

This chapter is organised into four sections. The first three sections describe, in order, the general simulation approach, the solving environment, and the analysis methodologies. The fourth chapter presents the results and discussions from each analysis.

5.1 Data simulation

To reflect realistic consumer behaviour, the simulation relies on a general reverse S-shaped relationship between item price and consumer demand that is well established in the literature (Van Heerde et al., 2001; Guadagni and Little, 2008; Tellis and Zufryden, 1995; van Heerde et al., 2002). In this framework, a price sensitivity parameter a_i is introduced to induce item-level variation within the product category. Let a_i quantify the impact of item i 's price p_i on its demand. Then, the demand for i when sold at p_i is given by:

$$d_i = \frac{10}{0.1 + \exp(a_i p_i - 8)} \quad (5.1)$$

Figure 5.1 plots (5.1) for various a_i values, and demonstrates the key characteristics of the simulated demand data. There is a reverse S-shaped relationship between price and demand, consistent with established retail pricing theory. The shape of the demand function suggests a saturation point beyond which further discounting has minimal effect on demand, and a threshold price above which consumers are reluctant to purchase.

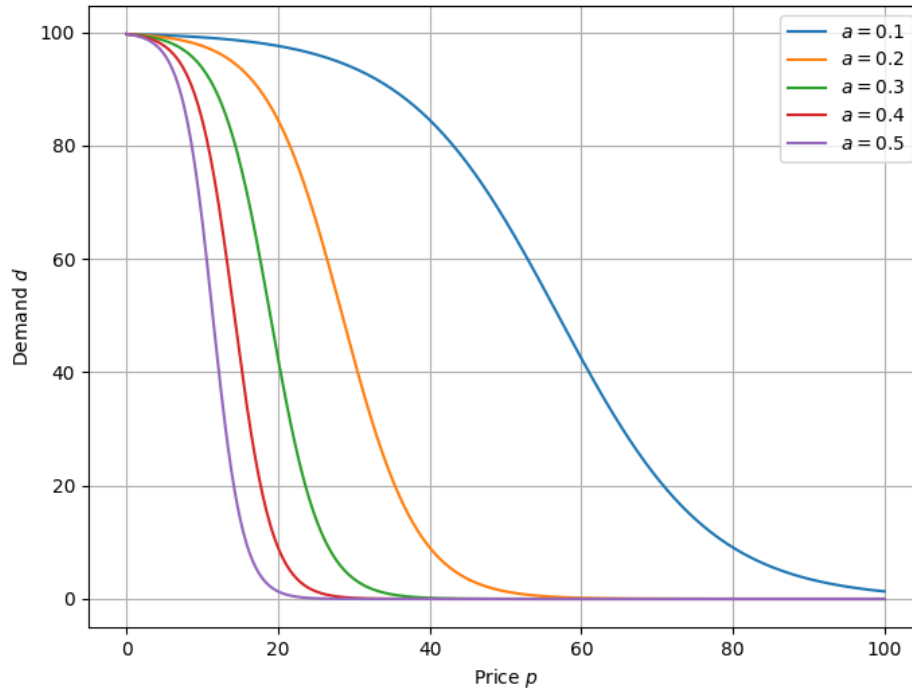


Figure 5.1: The demand curves used in simulating own-promotional data. The figure plots the demand model (5.1) across different values of price sensitivity a .

Figure 5.2 presents a conceptual overview of the simulation approach. The figure groups the model parameters into two tiers. The top-tier parameters (λ_i , η_i , a_i , q_k , m , Ω_{iww} , M , and Q) are independent of other parametric inputs and can, thus, be generated at initialisation. These parameters are simulated either by drawing from uniform distributions (e.g., λ_i , a_i) or by assigning fixed values. In contrast, the bottom-tier parameters (c_i , r_i , d_{ik} , α_{hik} , and L_{iww} and U_{iww}) depend either on these inputs or on other bottom-tier parameters, ensuring the internal consistency of the dataset. While attempts are made to calibrate parameters to reflect industry conventions, publicly available benchmark data is limited. For this reason, the simulation relies on assumptions to align the data with plausible retail ranges.

All datasets are generated programmatically using Python. The simulation proceeds in the order of the dependency tiers: first, the top-tier parameters are initialised; second, the bottom-tier parameters are generated using deterministic relationships. The complete Python implementation used to generate the datasets is publicly available on the GitHub repository: <https://github.com/29922704/masters-dissertation>

The simulation approach for each parameter is outlined below, together with the key assumptions that underpin it.

Buy-back rates (λ_i)

It is assumed that the vendor offers buy-back rates ranging from 0% to 20%. This range is selected to reflect realistic risk-sharing agreements in which the vendor provides a sufficiently strong incentive to encourage stocking (Anderson and Fox, 2019). The buy-back rate for each item is randomly selected from a uniform distribution calibrated to this range:

$$\lambda_i \sim U(0, 0.2), \quad \forall i \in N \quad (5.2)$$

Order quantity ratios (η_i)

Safety stock is commonly recommended at 10 to 20% of cycle stock (King and Bigler, 2025). Given the assumption of highly accurate demand forecasts, a conservative level of 5% is applied across the product category to balance order quantities with realised sales. The order quantity ratio is therefore defined as

$$\eta_i = 1.05 \quad \forall i \in N. \quad (5.3)$$

Price sensitivities (a_i)

Inspection of Figure 5.1 indicates that the range of 0.1-0.5 spans both relatively low and high price sensitivity. To reflect variation across items while remaining within realistic bounds, the a_i parameters are randomly selected from a uniform distribution calibrated to this range:

$$a_i \sim U(0.1, 0.5) \quad \forall i \in N \quad (5.4)$$

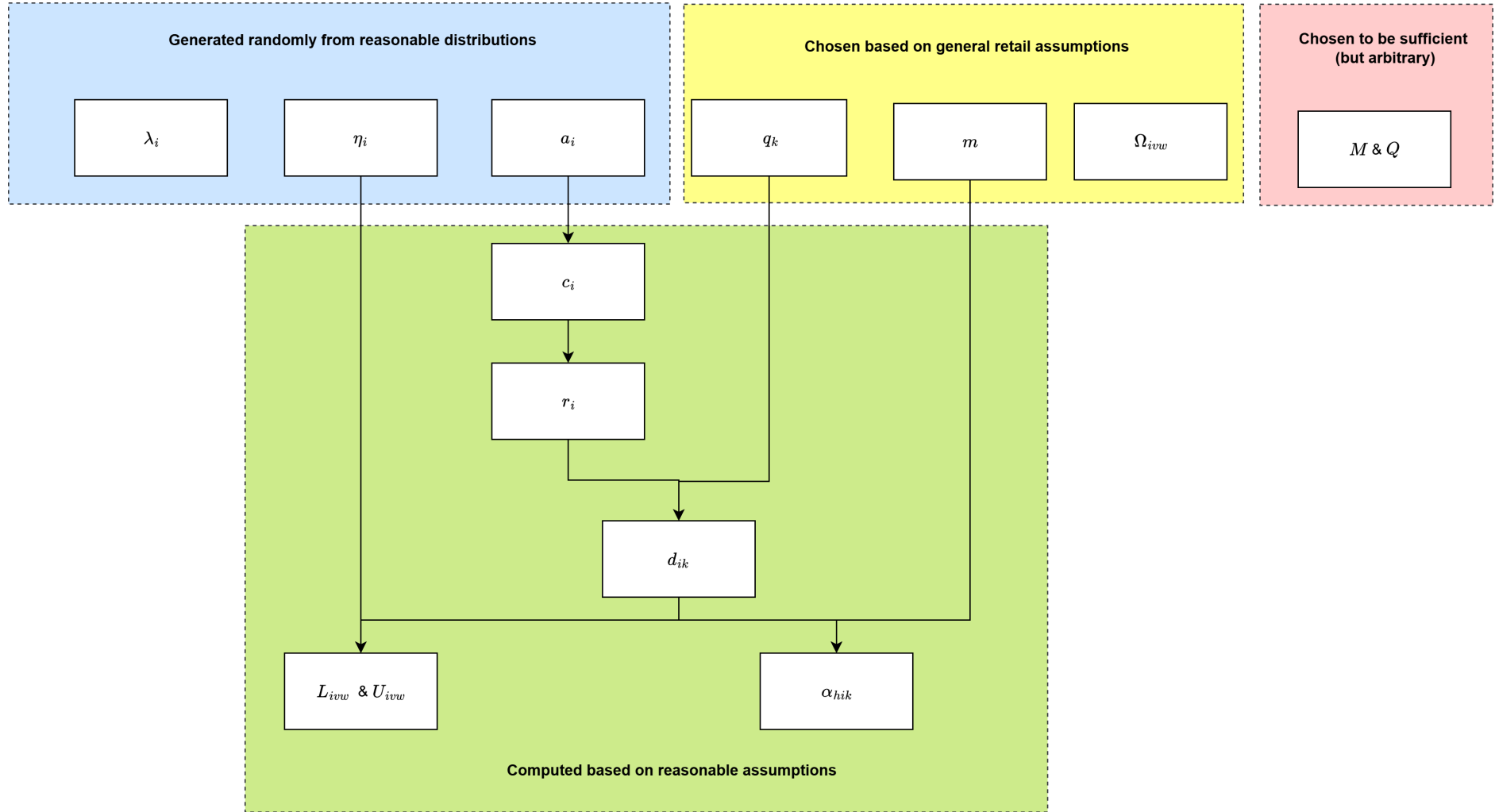


Figure 5.2: A conceptual representation of the data simulation approach. The arrows indicate dependencies among the parameters by showing the inputs used in each parameter's generation (e.g., c_i is generated from a_i). The figure also groups the parameters by their data generation methods.

Discount factors (q_k)

The retailer is assumed to apply discounts in 10% increments (e.g., the first tier applies a 10% discount, the second tier applies a 20% discount, etc.), such that:

$$q_0 = 1 > q_1 = 0.9 > \dots > q_{|K|}$$

. The discount factor can be computed programmatically via:

$$q_k = 1 - \frac{k}{10} \quad \forall k \in K \quad (5.5)$$

Substitutability parameters (m)

It is assumed that all items in the product category exhibit the same cross-promotional sensitivity, such that for every two units of an item sold above its regular demand, one unit of its substitute is displaced. Following the definition of the substitutability parameter in Section 3.43, these assumptions imply:

$$m = 2 \quad (5.6)$$

Off-invoice allowance and scan-back rates (Ω_{iww})

The vendors are assumed to offer a three-tier off-invoice allowance and scan-back programme, such that successive tiers increase the rate by 7.5%. In other words, the following assignment is made:

$$\Omega_{iww} = \begin{cases} 0 & \text{if } w = 0, \\ 0.075 & \text{if } w = 1 \\ 0.15 & \text{if } w = 2 \end{cases} \quad \forall i \in N, \forall v \in \{0, 1\}, \forall w \in W. \quad (5.7)$$

Auxiliary parameters (M and Q)

The auxiliary parameter values are arbitrary, but chosen to be sufficient:

$$M = 8000 \quad (5.8)$$

$$Q = 0.1 \quad (5.9)$$

Item costs (c_i)

Costing is aligned with the demand model by assuming that, allowing for some random market variation, an item sold at cost satisfies approximately half of its total demand. According to the demand model (5.1), each item's total market demand is 100 units. Then, the cost of each item can be expressed in terms of a_i by substituting $p_i = c_i$ and $d_i = 0.5(100)$ into (5.1) and solving for c_i :

$$\begin{aligned}
 d_i &= \frac{10}{0.1 + \exp(a_i p_i - 8)} \\
 0.5(100) &= \frac{10}{0.1 + \exp(a_i c_i - 8)} \\
 5 + 50 \exp(a_i c_i - 8) &= 10 \\
 50 \exp(a_i c_i - 8) &= 5 \\
 \exp(a_i c_i - 8) &= \frac{1}{10} \\
 \ln \frac{1}{10} &= a_i c_i - 8 \\
 \ln \frac{1}{10} + 8 &= a_i c_i \\
 \therefore c_i &= \frac{\ln 0.1 + 8}{a_i}
 \end{aligned} \tag{5.10}$$

To introduce market variation, a random adjustment factor, $\epsilon_i \sim (-0.15, 0.15)$ is added to (5.10) to generate the cost parameters, such that:

$$c_i = (1 + \epsilon_i) \frac{\ln 0.1 + 8}{a_i} \quad \forall i \in N \tag{5.11}$$

Regular retailer prices (r_i)

Regular retail prices are set by applying a 40% gross profit margin to the item's cost, consistent with typical margins in deli, bakery, produce, and frozen product categories (Nutrition Incentive Hub, 2022). In other words, the regular retail prices are assigned as follows:

$$r_i = 1.4 c_i, \quad \forall i \in N. \tag{5.12}$$

Own-promotional demand (d_{ik})

To generate the demand parameters, $p_i = q_k r_i$ is substituted into (5.1). Thus, the demand for i at k is:

$$d_{ik} = \frac{10}{0.1 + \exp(a_i q_k r_i - 8)}, \quad \forall i \in N, \forall k \in K. \tag{5.13}$$

Cross-promotional demand cannibalisation rates (α_{hik})

The cross-promotional demand cannibalisation rates α_{hik} are calculated by substituting m and d_{ik} into (3.43).

Trade spend requirement bounds (L_{i0w} and U_{i0w})

The trade spend bounds are based on the requirement to increase the regular order or sale quantity by a given percentage, calibrated to correspond to the range of discounting levels considered. If the increase in the relevant quantity is in the range:

- $[0, 1.5)$, the retailer qualifies for $\Omega_{i00} = 0$,
- $[1.5, 2.5)$, the retailer qualifies for $\Omega_{i01} = 0.075$,
- $[2.5, M)$, the retailer qualifies for $\Omega_{i02} = 0.15$.

It follows that the requirements for the off-invoice rate w can be expressed as:

$$L_{i0w} = \begin{cases} 0 & \text{if } w = 0 \\ (1.5)\eta_i d_{ik} & \text{if } w = 1 \\ (2.5)\eta_i d_{ik} & \text{if } w = 2 \end{cases} \quad U_{i0w} = \begin{cases} (1.5)\eta_i d_{ik} - Q & \text{if } w = 0 \\ (2.5)\eta_i d_{ik} - Q & \text{if } w = 1 \\ M & \text{if } w = 2 \end{cases} \quad \forall i \in N, \forall w \in W. \quad (5.14)$$

Similarly, the requirements for the scan-back rate w is given by:

$$L_{i1w} = \begin{cases} 0 & \text{if } w = 0 \\ (1.5)d_{ik} & \text{if } w = 1 \\ (2.5)d_{ik} & \text{if } w = 2, \end{cases} \quad U_{i1w} = \begin{cases} (1.5)d_{ik} - Q & \text{if } w = 0 \\ (2.5)d_{ik} - Q & \text{if } w = 1 \\ M & \text{if } w = 2 \end{cases} \quad \forall i \in N, \forall w \in W. \quad (5.15)$$

Datasets generated

To conclude this section, although it is impractical to include all 26 datasets generated by this method, the dataset used as the baseline in the sensitivity analysis is provided as an example. As noted in Chapter 1, all datasets are publicly available at <https://github.com/29922704/masters-dissertation>. Figure 5.3 shows the demand curves used in the sensitivity analysis and illustrates the relationship between each item's cost, permissible prices, and corresponding demand levels for the sample dataset wherein $N = K = W = \{0, 1, 2\}$. The complete dataset is provided in Appendix C, Table C.1.

5.2 Solving environment

Each instance is formulated using the complete AMPL model (see Listing B.8), and all inputs and outputs are managed via the AMPL Python API. Unless stated otherwise, solved via BONMIN using an approach recommended in the solver documentation for non-convex MINLPs (see Section 2.1.5.2). To this end, the solver is configured as follows:

- The IPM is restarted from five randomly selected initial points. At each subsequent

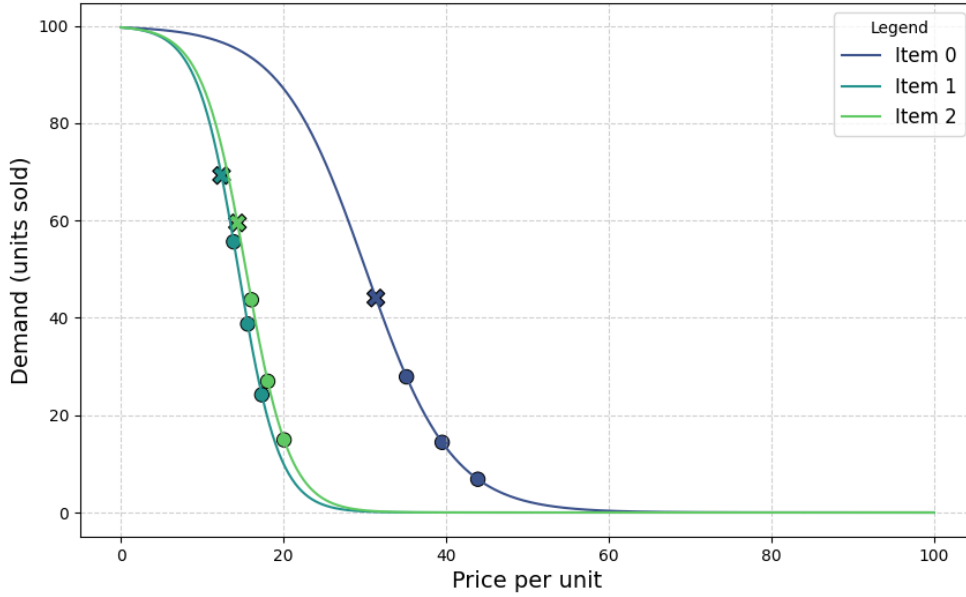


Figure 5.3: The demand models used in the sensitivity analysis. Note: each '●'-symbols is associated with a specific discount factor q_k , and each '×'-symbol represents the cost of the corresponding item.

node of the B&B algorithm, the IPM is restarted at two randomly selected points. The solution continues from the best solution obtained among these subproblems.

- The pruning rules are disabled, forcing a complete exploration of the B&B tree.

The algorithm terminates when the KKT conditions are satisfied to within a tolerance of 1×10^{-6} . In other words, following the notation of (2.16), $\epsilon_{\text{tol}} = 1 \times 10^{-6}$. This implies a very high level of numerical accuracy in the local optima identified by the IPM and, consequently, that the solution to each B&B subproblem is accepted only when an extremely stringent optimality criterion is met. However, to prevent unbounded runtimes, a CPU time limit of 3600 seconds is imposed on each solve.

Finally, as the reported computation times are hardware-dependent, it is noteworthy that the analysis is conducted on an HP Elitebook 8770w with the following specifications:

- Processor: Intel(R) Core(TM) i7-3740QM CPU 2.70GHz 2.70 GHz
- Operating system type: Windows 10 Pro, 64-bit operating system, x64-based processor
- Installed memory: 8.00 Gb

5.3 Analysis methodologies

5.3.1 Benchmark analysis methodology

A series of non-trivial datasets is simulated, varying in size. Specifically, twenty datasets were created as follows:

- The first ten datasets represent small-scale instances with:

$$N = K = W = \{0, 1, 2\} \quad (5.16)$$

- Datasets 11–15 represent medium-scale instances with:

$$N = \{0, 1, 2, 3, 4\} \quad (5.17)$$

$$K = \{0, 1, 2, 3\} \quad (5.18)$$

$$W = \{0, 1, 2\} \quad (5.19)$$

- Datasets 16–20 represent large-scale instances with:

$$N = \{0, 1, 2, 3, 4, 5, 6\} \quad (5.20)$$

$$K = \{0, 1, 2, 3, 4\} \quad (5.21)$$

$$W = \{0, 1, 2\} \quad (5.22)$$

Each instance is first solved using BONMIN, following the approach described in Section 5.2. The same instances are then solved using the BARON solver (via the NEOS server) to obtain the exact solutions.

From each solve, the objective function values and item prices (p_i) are recorded, and the optimality gap is calculated as:

$$\text{Optimality gap} = \frac{\text{Profit}_{\text{BONMIN}} - \text{Profit}_{\text{BARON}}}{\text{Profit}_{\text{BARON}}} \quad (5.23)$$

These results allow a direct comparison between the heuristic solutions (obtained from BONMIN) and the exact solutions (from BARON), enabling the assessment of the reliability of the heuristic solutions.

5.3.2 Sensitivity analysis methodology

The analysis explores how profitability responds to variations in the following key inputs:

- the price sensitivity a_i , a parameter introduced in this chapter relating to demand,
- the regular prices r_i ,

- the costs c_i ,
- the percentage safety stock, by altering the safety stock ratio η_i .
- the trade spend rates, through changes to the buy-back rate λ_i and the off-invoice and scan-back rates Ω_{ivw} ,
- the trade spend requirements, by varying the lower and upper trade spend requirement bounds L_{ivw} and U_{ivw} ,
- the cross-promotional sensitivity m .

First, a baseline dataset is simulated (following the approach defined in Section 5.1) with the set sizes: $N = K = W = \{0, 1, 2\}$. Next, the key inputs are systematically varied by a symmetric range of percentage deviations around the baseline value corresponding to: -20% , -10% , 0% (i.e., the baseline case), 10% , and 20% . Hereby, five modified datasets are created for every key input.

With each instance, an alternative pricing strategy is evaluated wherein all items are sold at their regular (and thus highest possible) price, $q_0 r_i$. To capture output under this strategy, a modified version of the operational model is used. The full formulation of the model is provided for reference as a listing in Appendix B.3; it consists of the original operational model and the following constraints to enforce the selection of the regular retail prices:

$$\gamma_{i0} = 1 \quad \forall i \in N \quad (5.24)$$

The instances are solved and the resulting profit (3.29), total cost (3.28), total sale income (3.27), and total trade spend income (3.21) is recorded, in addition to the following:

- the overall profit margin, as given by:

$$P_{\text{margin}} = \frac{P}{P + C} \quad (5.25)$$

- the trade spend efficiency, i.e., the percentage of revenue originating from trade spend as opposed to sales, as calculated by:

$$\text{eff}_{\text{ts}} = \frac{I_{\text{ts}}}{I_{\text{ts}} + I_{\text{s}}} \quad (5.26)$$

- the percentage change in profit between the two pricing strategies, as given by:

$$P_{\text{change}} = \frac{P - P'}{P'} \quad (5.27)$$

The internal consistency and theoretical alignment of the outputs are analysed qualitatively through reasoned interpretation to validate the model's ability to reflect realistic behaviour.

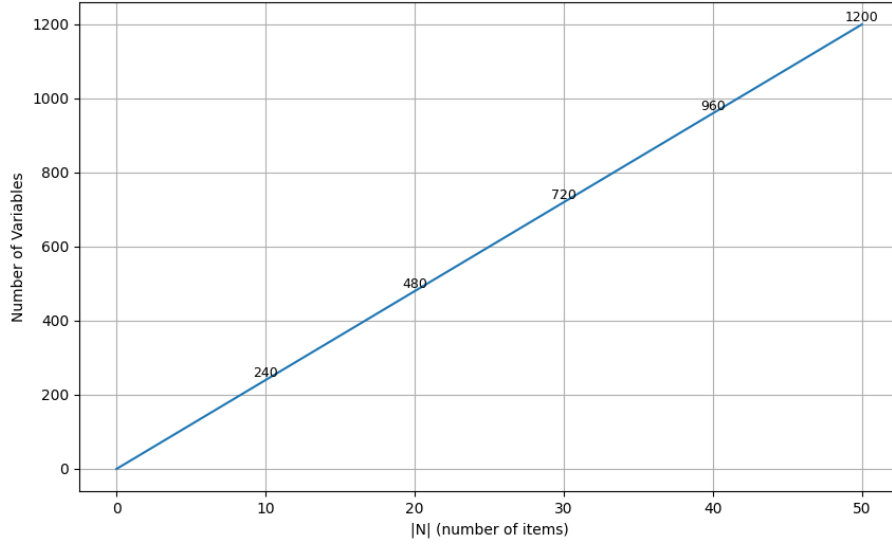


Figure 5.4: The variable count of the computability analysis as a function of $|N|$ when $|W| = 2$ and $|K| = 4$

5.3.3 Scalability analysis methodology

A series of datasets is simulated with progressively increasing sizes to reflect growing problem complexity. The datasets are simulated according to the method described in Section 5.1. The size of the decision domain primarily depends on the number of items in an instance, owing to the dominance of the $|N|$ in (5.28), which gives the number of variables as a function of the set sizes. Therefore, the analysis systematically increases $|N|$ to enact progressively larger problems while keeping the other set sizes constant, e.g., $K = \{0, 1, 2\}$ and $W = \{0, 1, 2\}$.

$$6|N||W| + |N||K| + 3|N| \quad (5.28)$$

Given this configuration, Figure 5.4 plots the number of variables as a function of $|N|$. It shows that the number of variables increases linearly, with each additional 10 items contributing approximately 200 variables to the model, supporting the use of increasing $|N|$ as an analogue of problem complexity.

Each instance is solved using BONMIN, following the approach described in Section 5.2, and computation times are recorded to assess model performance across varying problem sizes. The dataset generation and solves are terminated once a clear trend in computation time is established to avoid unreasonably prolonged computations that do not yield additional insight. Finally, the trend is assessed to validate the computability of the solving approach.

5.4 Results and discussions

5.4.1 Benchmark analysis results and discussion

The complete results from each solve are provided in Appendix D.2, Table D.2; however, Table 5.1 provides a summary thereof by reporting the optimality gap for each instance, indicating the deviation of the heuristic solutions from the exact solutions.

Table 5.1: The optimality gaps of the heuristic (BONMIN) solutions across the twenty simulated instances.

Instance	Optimality gap (%)	Instance	Optimality gap (%)
1	0.106	11	0.119
2	0.16	12	0.139
3	0.091	13	0.116
4	0.111	14	0.114
5	0.194	15	0.115
6	0.135	16	0.000
7	0.118	17	0.000
8	0.114	18	0.000
9	0.071	19	0.000
10	0.105	20	0.000

Table 5.1 demonstrates that the heuristic solutions closely approximate the exact solutions, with optimality gaps generally below 0.2%. Further inspection of Table D.2 reveals that the solved prices do not differ between the solutions, suggesting that the minor optimality gaps can be explained by differences in the computational precision between the solvers (e.g., rounding accuracy).

To generalise the findings: the average optimality gap is 0.095%. Given that minor optimality gaps can be attributed to differences in computational precision between the solvers, the heuristic solutions are considered exact across all instances. Thus, the findings suggest that the solving approach yields sufficiently good heuristic solutions, thereby validating its reliability.

5.4.2 Sensitivity analysis results and discussion

While this subsection presents a summary of the main findings, the complete results of the sensitivity analysis, including the tables and figures referenced below, are provided in Appendix D.1. The profit (3.29), trade spend efficiency (5.26), profit margin (5.25), and change in profit (5.27) relative to the baseline case from each solve are plotted in Figures D.2 - D.8. Table D.1 reports the numerical outcomes of the sensitivity analysis, providing the total cost (C), sale income (I_{ts}), trade spend income (I_{ts}), and profit (3.29) from each solve under each pricing strategy. The table also provides the percentage change in profit between the pricing strategies (5.27). Finally, to summarise the overall influence of each parameter, Figure D.1 compares the profit impacts across all variables. A summary and discussion of the results from the analysis follows.

5.4.2.1 Interpretation of parameter responses

Price sensitivity (a_i) response

Figure D.2(a) reveals a logarithmic decrease in the profit as price sensitivity increases. This response is supported by Blazenko et al. (2012) and is attributed to the inverse exponential relationship between price sensitivity and demand, as defined in (5.1). Specifically, thereby limiting the retailer's capacity to generate trade spend income through such (explaining the tapering effect present at high price sensitivity). Figure D.2(b) indicates that trade spend efficiency generally decreases with price sensitivity, with a spike at the baseline value. At low price sensitivity, targeted discounting has little effect on demand, which in turn limits the retailer's capacity to generate trade spend income through targeted discounting. Additionally, at high price sensitivity, overall demand declines, reducing the retailer's ability to secure lucrative trade spend deals. Indeed, Table D.1 indicates that at the highest price sensitivity, income is generated almost exclusively from sales. It follows that the baseline spike in trade spend efficiency can be explained by an increased ability to boost demand and accrue trade spend at moderate price sensitivity. Finally, the profit margin exhibits a near-logarithmic decline with respect to price sensitivity, as shown in Figure D.2(c). This response is attributed to the simultaneous decline in income from sales and trade spend efficiency (via decreased demand), with the relative spike at the greatest price sensitivity being ascribed to the model selecting higher margins when discounts fail to meaningfully increase demand. Overall, these responses are internally consistent with the demand model.

Regular selling price (r_i) response

Figure D.3(a) demonstrates that profit increases linearly as regular prices increase. Although this appears counterintuitive given that demand diminishes at higher prices, Nagle and Müller (2017) explain that increasing regular prices can increase total profit when higher per-unit margins outweigh lost sales. Figure D.3(c) reveals that this is indeed the case since it shows a simultaneous growth in the profit margin. Thus, as regular prices increase, losses are recouped when the overall gain from per-unit profitability exceeds the marginal gains from additional sales (at low margins). This behaviour also explains the near logarithmic decline in trade spend efficiency shown in Figure D.3(b) as the sales revenue decreases relative to trade spend revenue. Thus, the response to varying r_i aligns with standard economic intuition captured by the model.

Cost (c_i) response

Figure D.4 shows that profit and profit margins decrease linearly with costs, while trade spend efficiency increases linearly. These results are consistent with the expectation that increasing costs compress profit margins and reduce profits. Simultaneously, because trade spend income is calculated as a proportion of cost, greater costs also increase trade spend income relative to sales income and, in turn, raise trade spend efficiency.

Order quantity ratio (η_i) response

The response from varying η_i mirrors that of varying c_i . Specifically, Figure D.5 demonstrates that profit and profit margins decrease linearly when safety stock is increased, while trade spend efficiency increases linearly. Again, these results are intuitive. Increasing safety stock raises inventory costs, thereby reducing profit margins and total profit. At the same time, greater safety stock directly increases off-invoice allowances and buy-back revenue, resulting in greater trade spend income and improved trade spend efficiency.

Trade spend rate (Ω_{ivw} and λ_i) response

Figure D.6 demonstrates a linear increase in profit, trade spend efficiency, and profit margin as trade spend rates are increased. These results are logical: higher trade spend rates directly increase trade spend income, which in turn raises profit margins, profit, and trade spend efficiency.

Trade spend requirement bounds (L_{ivw} and U_{ivw}) response

Figure D.7 shows that profitability is unaffected by variations in L_{ivw} and U_{ivw} , with the observed minor fluctuations being attributable to computational limitations in output precision. However, it is emphasised that this finding is context-dependent and a consequence of the relationship between the demand model and the bounds. More simply, strictly within the analysis configuration, the marginal changes to the eligibility bounds are insufficient to effect a discrete change in the trade spend rates given d_{ik} . As such, the response is internally consistent given the problem scope.

Item substitutability (m) response

As defined in (3.43), there is an inverse relationship between cross-promotional sensitivity and the substitutability parameter. Specifically, m values correspond to reduced cross-promotional sensitivity and sale cannibalisation. Furthermore, the cross-promotional cannibalisation rate α_{hik} nears zero when m is increased without bound (owing to m being a denominator in the governing equation). This context explains the response seen in Figure D.8(a), which indicates that the profit increases logarithmically with m ; Cohen et al. (2021) support the notion that stronger substitutability among products can reduce total profitability through greater cross-promotional cannibalisation. Simultaneously, Figures D.8(b) and Figure D.8(c) show that trade spend efficiency and profit margins decrease logarithmically. The response in trade spend efficiency indicates that the increase in demand at low substitutability (high m) does not yield improved trade spend deals under the current configuration. At the same time, heightened demand increases ordering costs proportionally, reducing the profit margin. Thus, the outcomes across varying substitutability levels are consistent with the cross-promotional demand cannibalisation estimation method and are consistent with economic intuition.

5.4.2.2 The most influential parameters affecting profit

Figure D.1 indicates that price sensitivity (a_i) is the most influential parameter on profit, followed by regular item prices (r_i), and costs (c_i). Al-Mamun et al. (2014) corroborate this finding by identifying price sensitivity as a key factor impacting profitability, further signifying theoretical alignment.

5.4.2.3 Model stability across market conditions

Across the evaluated pricing strategies, Table D.1 shows that the model consistently outperformed the alternatives in profit, with the notable exception of price sensitivity, where the strategies yielded comparable profits at low and high sensitivity. This highlights the model's adaptive capacity to diverse market conditions through several compensatory mechanisms, namely:

- the model implements strategic discounts to stimulate demand recovery, when demand is reduced at elevated profit margins r_i ,
- the model offsets increased costs c_i by prioritising trade spend efficiency.
- the model secures more lucrative off-invoice allowances and buy-back agreements to mitigate inventory expenses under increased safety stock requirements. η_i .
- the model adapts to reduced trade spend rates Ω_{iww} and λ_i by shifting focus to sales volume revenue.
- the model counters strong cross-promotional effects m by moderating discount depth to minimise cannibalisation.

While reliance on simulated data and a simple benchmark strategy limits the strength of the conclusions, the model's robustness and logical consistency across parameter changes indicate that it provides a reliable foundation for further empirical applications and benchmarking against real-world retail promotion data.

5.4.2.4 Managerial insights and practical considerations

Conduct promotions in moderately price-sensitive environments

The findings indicate that price sensitivity is the most influential factor affecting profitability, and that discounting under low and high sensitivity offers little benefit relative to moderate sensitivity. Retailers should thus invest in demand forecasting to identify moderately price-sensitive environments, and plan promotions accordingly to maximise the effectiveness of strategic discounting.

Ensure regular pricing margins are sufficiently high

While trade spend can offset per-unit profitability, the findings indicate that regular pricing margins significantly impact profitability. Thus, retailers should not rely on trade spend as a primary driver of revenue and should ensure that regular margins are set appropriately high to sustain profitability; discounting may increase promotional profitability, but higher base prices support revenue even when discounts are applied.

Focus promotions on high-cost items

Retailers' control over costs is limited to their agreements with vendors. However, the analysis indicates that trade spend can effectively offset costs through a positive correlation between costs and the value of trade spend instruments. In other words, high-cost items may generate more trade spend income relative to low-cost items. Retailers should thus align promotional activities with higher-cost items to recoup costs.

Avoid multiple simultaneous promotions within a single product category

The response in varying item substitutability highlights how cross-promotional cannibalisation erodes profits. Consequently, it is generally inadvisable to run multiple simultaneous promotions within the same product category, as revenue tends to be rerouted from the more profitable items, reducing overall profit. A more effective approach is to stagger promotions across categories, ideally targeting complementary categories to leverage positive cross-promotional effects and increase revenue.

5.4.2.5 Sensitivity analysis conclusion

Across all parameter variations, the model's responses are theoretically aligned and internally consistent. Furthermore, the results indicate that the model remains stable across variations in market conditions and consistently outperforms or matches the benchmark strategy, suggesting that the optimisation logic responds coherently to structural parameter shifts. Taken together, the findings validate the model's ability to reflect realistic behaviour.

5.4.3 Scalability analysis results and discussion

The results, presented in Figure 5.5, indicate that computation time increases exponentially with the number of items included in a solve. This reveals a key limitation in the approach to solving. For relatively small problems ($|N| \leq 5$), the model remains solvable; however, the computational time required to solve larger problem instances becomes excessive, rendering the approach impractical.

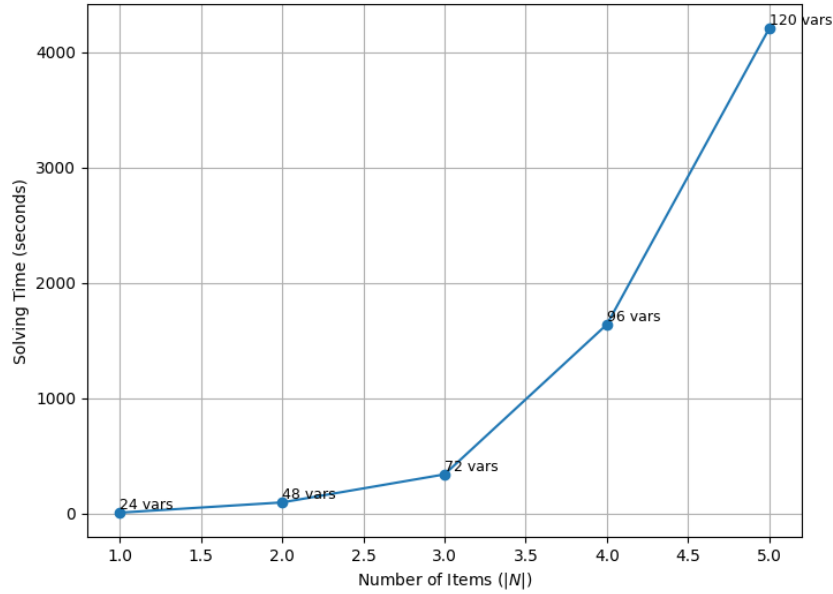


Figure 5.5: The computation time required to solve each instance, shown as a function of the number of items. The figure also shows the number of variables per instance.

The apparent approach to alleviate the computation burden is to relax the stringent heuristic modifications applied to the solver (as described in Section 5.2). While this approach reduces the likelihood of convergence to the global optimum, it should improve computational tractability by restricting the portion of the solution space explored. To this end, the experiment is repeated with the following modifications: the IPM is executed only once at each B&B node, and the pruning rules are reintroduced. The results from this experiment are presented in Figure 5.6. The figure shows that the computation time of this approach, although reduced, remains exponential in the number of items. It follows that even relaxed solver settings do not mitigate the rapid growth in computational effort for larger instances.

Thus, while the computability of the solving approach is validated for small problem instances, it presents scalability issues. Future work could therefore identify or develop more efficient solving approaches for the proposed model.

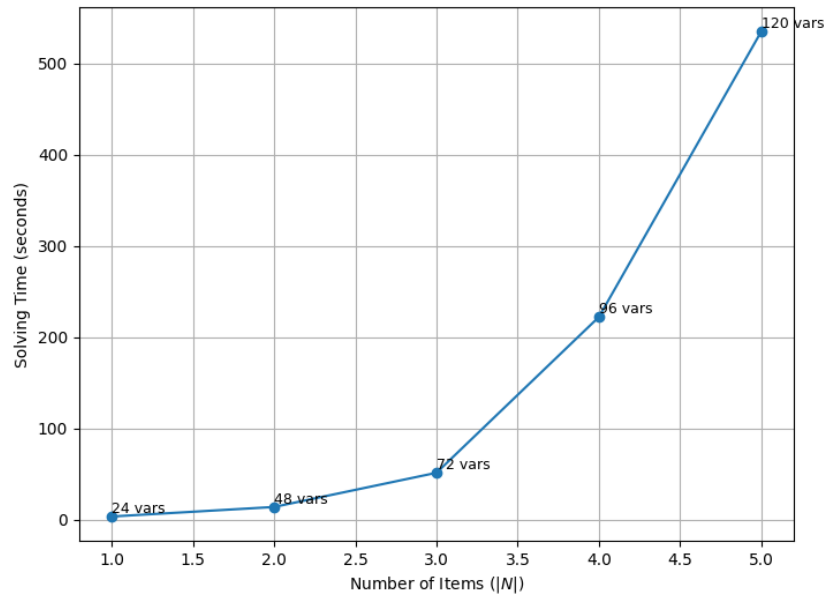


Figure 5.6: The computation time required to solve each instance using a standard B&B and IPM solving configuration, shown as a function of the number of items. The figure also shows the number of variables per instance.

5.5 Chapter summary

This chapter validates the model through three analyses employing simulated data. First, a benchmark analysis reveals that the heuristic solutions are exact in all test cases considered, validating the reliability of the solutions produced by the solving approach. Second, a sensitivity analysis reveals that each parameter responded in accordance with its governing mathematical definition and with standard economic intuition. Additionally, the model demonstrated stability across diverse market conditions. Its capacity to maintain or surpass benchmark profitability illustrates the effectiveness of the built-in compensatory mechanisms. Taken together, it is reasonable to conclude that the model is theoretically sound and internally coherent, thereby validating its ability to reflect realistic behaviour. Third, a scalability analysis systematically increases problem sizes to establish trends in computational time. The results indicate that computation time grows exponentially with the number of variables. This indicates that the solving approach remains effective only for relatively small problems, thereby validating its computability. The next chapter concludes the study.

Chapter 6

Conclusion

6.1 Chapter summaries

Chapter 1 introduces the project. The chapter outlines the problem statement, aim, five objectives and research methods. As stated in the chapter, Retailers experience significant challenges in designing profitable promotions. While an SLR identified existing models that aim to assist retailers with promotion planning, no single model considers the effects of own-promotional, cross-promotional, and trade spend simultaneously. This study aimed to address this problem and the gap identified in the literature by proposing a new optimisation model that maximises retailer profit while accounting for own-promotional, cross-promotional, and trade spend effects.

Chapter 2 first addresses objective (1a) by presenting a literature review that provides the theoretical foundation for the study. The chapter describes the terminology central to mathematical optimisation and overviews concepts relevant to model formulation, solution, and the interpretation of solutions. The chapter then addresses objective (1b) by contextualising the study within the existing literature, presenting an SLR of models that aim to assist retailers in promotion design. Within the scope of the SLR, 10 studies were identified that aimed to optimise retail price promotions. Among them, there is a preference for modelling the problem using NLP, yet none simultaneously address own-promotional, cross-promotional, and trade spend effects.

Chapter 3 addresses objective (2) by proposing a new MINLP model that assists retailers with promotion planning while accounting for own-promotional, cross-promotional, and trade spend effects. The chapter first presents the problem conceptually; given operational retail inputs (including own-promotional and cross-promotional data) and trade spend inputs (i.e., off-invoice allowance, scan-back, and buy-back rates data), the (i) order quantity of each item within a given product category, (ii) the subset of items within the category to discount, and (iii) the size of each discount must be computed, resulting in the maximum retailer profit being obtained. Starting from the base assumptions and a

description of the notation, a MINLP model is formulated in accordance with the conceptual structure. The objective of the model is to maximise (3.29), subject to constraints (3.30 - 3.37d).

Chapter 4 documents verification, consisting of two stages. As further analysis employs the BONMIN solver, which requires the problem's objective function and constraints to be twice continuously differentiable, the chapter first addresses objective (3a) by verifying that the proposed model satisfies this requirement. This is achieved by evaluating the model's mathematical structure to demonstrate that it is composed of finite sums of polynomials. Since polynomials are infinitely differentiable, the model's adherence to the requirements of the solving approach is thereby verified. The chapter then addresses objective (3b) by verifying the correctness of the proposed model's formulation and implementation through seven targeted assessments. Each assessment involves implementing specific model components in AMPL and solving simplified instances using trivial inputs to confirm that the solver output matches the analytically expected values. Since the solver outputs matched the analytical values across all instances, the model's correctness and implementation were verified. The complete operational (AMPL) model, constructed by aggregating the individually assessed components, is likewise verified for use in further analysis. Together, these verifications confirm the correct use of the solver, the soundness of the model's mathematical logic, and the absence of errors in its implementation.

Chapter 5 documents the validation process, which consists of three stages, each involving a targeted analysis using simulated data. The data simulation relies on an established relationship between price and demand. This framework introduces an additional price-sensitivity parameter to induce item-level variation in demand. The parameters are then generated randomly from appropriate distributions, selected based on general retail assumptions, or computed under reasonable assumptions. Each analysis uses BONMIN to produce heuristic solutions for the model. The first stage of validation involves benchmarking, in which 20 datasets are simulated and solved twice: first using BONMIN and second using the BARON solver, which yields exact solutions. The optimality gap of each heuristic solution is computed by comparing it with the corresponding exact solution. The results indicate that the heuristic solutions are exact across all instances, validating the reliability of the approach for solving objective (4a). The second validation stage involves a sensitivity analysis in which key parameters from a single simulated dataset are varied, and the resulting changes in profitability are qualitatively analysed. The results indicate that the model's responses are theoretically aligned and internally consistent across all parameter variations, thereby validating its ability to reflect realistic behaviour and address objective (4b). Additionally, the results identify price sensitivity as the most influential factor affecting profitability and indicate that the model remains stable across variations in market conditions, as it consistently outperforms a pricing strategy in which items are sold at their regular prices. The third validation stage involves a scalability analysis in which progressively larger instances are solved to establish a trend in computation time. The results indicate that computation time grows exponentially with the number of items included in a solve, validating the solving approach for small-scale problem instances to

address objective (4c), while revealing its computational limitations for larger problems. Taken together, the validations demonstrate that the solutions obtained are reliable, in accordance with economic intuition, and that the solving approach is effective for small problem instances.

It is concluded that the first four research objectives, as defined in the introductory chapter, are successfully addressed. The main contribution of this study is the formulation of the presented model, as no known promotion optimisation model simultaneously addresses own-promotional, cross-promotional, and trade spend effects. The model inherently simplifies the promotion planning problem by relying on assumptions such as discrete discount and trade spend tiers, fixed safety stock percentages, and highly accurate demand forecasts. Nevertheless, the verification and validation results provide confidence that the model and its solutions correctly describe the promotional pricing environment as conceptualised. Within this scope, the model is shown to effectively address the core challenge of simultaneously considering own-promotional, cross-promotional, and trade spend effects to drive retailer profit from price promotions.

The managerial insights that follow from this conclusion are clear:

- Retailers should target promotions in moderately price-sensitive markets, where strategic discounting most effectively boosts demand.
- Regular pricing margins should be set sufficiently high, as increased demand does not fully offset per-unit profitability losses.
- Promotions should be focused on higher-cost items, which generate greater trade spend benefits to offset expenses.
- Multiple simultaneous promotions within the same product category should be avoided, since cross-promotional cannibalisation can erode profits; instead, stagger promotions across complementary categories to enhance overall revenue.

It follows that holistic promotion planning yields superior profit outcomes compared with isolated or ad hoc discounting strategies. Retailers should, therefore, carefully evaluate interactions among products while strategically leveraging trade spend to maximise category gains. Simultaneously, retailers should invest in robust demand forecasting techniques to ensure that promotional decisions align with consumer behaviour. Finally, the proposed model demonstrates the tangible financial value of data-driven decision support tools, highlighting their role in improving overall profitability by reducing the retailer's reliance on intuition in promotion planning. The final research objective (5) is addressed in the next section, which provides recommendations for future work for the project.

6.2 Future research

To address research objective (7), the following recommendations on future research follow from the study:

- As the sensitivity analysis identified price as the most influential parameter affecting profit, inaccurate demand estimation poses a significant risk to the retailer's bottom line. Thus, future research could improve upon the study's demand modelling approach by incorporating more advanced forecasting techniques with quantifiable reliability. Alternatively, future work could extend the model to account for forecast uncertainty via stochastic programming, enabling it to adjust outputs based on expected demand variation and supporting more resilient decision-making under uncertainty.
- While this work deliberately models the problem in a single-period setting, future work could extend the model to a multi-period setting to better capture the long-term impact of stockpiling on profit optimisation and introduce inventory management functionalities. This modification would mitigate the erosion of long-term profitability due to overpromotion.
- While this study considers three trade spend mechanisms closely associated with price promotions, future research could include additional trade spend mechanisms to provide a more comprehensive system for optimising retailer profit from trade promotions. By evaluating a broader range of trade spend types, future models could offer greater flexibility and resilience in promotion planning under fluctuating supplier offerings, thereby addressing the third risk identified in Chapter 6, namely the potential over-reliance on trade spend.
- Although the dissertation provides a rudimentary method for estimating cannibalisation, future work could pursue a more robust methodology to capture cross-promotional dynamics to prevent demand being diverted toward lower-margin discounted items due to errors in cross-promotional cannibalisation estimates. Moreover, while this study focuses solely on cannibalisation, cross-promotional effects may also include complementary demand increases (e.g., a discount on coffee increases sales of coffee filters). Future models could therefore be enhanced by incorporating both cannibalisation and complementary effects, offering a more complete representation of cross-promotional behaviour.
- While the current model incorporates inventory costs, future research could enhance the inventory management component by integrating more sophisticated mechanisms, such as dynamic inventory control, stochastic demand modelling, or multi-period inventory balancing. This extension would enable better coordination between promotional strategies and operational capacity, thereby providing a more comprehensive representation of the retailer's finances.

- While the model performed robustly within the sensitivity analysis, its real-world effectiveness in optimising retailer profit remains a topic for future work. This could be achieved through case studies that use real-world data, experimental implementations, or comparisons with alternative discounting strategies.
- The scalability of the solving approach is a key limitation, as it limits the model's applicability to larger problems. To address this, future research could explore more scalable solution techniques to improve the model's computational efficiency. Among these, future work could explore custom metaheuristic-based approaches. In particular, the binary nature of the discount selection variables facilitates genetic encoding, making genetic algorithms a promising candidate for further investigation.

Bibliography

- Ailawadi, K. L., Gedenk, K., Lutzky, C., and Neslin, S. A. (2007). Decomposition of the sales impact of promotion-induced stockpiling. *Journal of Marketing Research*, 44(3):450–467.
- Al-Mamun, A., Rahman, M., and Robel, S. D. (2014). A critical review of consumers’ sensitivity to price: Managerial and theoretical issues. *Journal of International Business and Economics*, 2(2):1–9.
- AMPL Optimization Inc. (2021). amply: Python API for AMPL. From <https://github.com/ampl/amply>. Retrieved: 25-01-30.
- Anderson, E. T. and Fox, E. J. (2019). How price promotions work: A review of practice and theory. In Dubé, J.-P. and Rossi, P. E., editors, *Handbook of the Economics of Marketing*, volume 1, pages 497–552. North-Holland.
- Assunção, J. L. and Meyer, R. J. (1993). The rational effect of price promotions on sales and consumption. *Management Science*, 39(5):517–535.
- Baardman, L., Boroujeni, S. B., Cohen-Hillel, T., Panchamgam, K., and Perakis, G. (2023). Detecting customer trends for optimal promotion targeting. *Manufacturing & Service Operations Management*, 25(2):448–467.
- Babar, M. A. and Zhang, H. (2009). Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 1–10. IEEE.
- Beiranvand, V., Hare, W., and Lucet, Y. (2017). Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18(4):815–848.
- Bell, D., Chiang, J., and Padmanabhan, V. (1999). The decomposition of promotional response: An empirical generalization. *Marketing Science*, 18(4):504–26.
- Bell, D. and Drèze, X. (2002). Changing the channel: A better way to do trade promotions. *MIT Sloan Management Review*, 43(2):42–49.
- Bertsekas, D. P. (2016). *Nonlinear Programming*. Athena Scientific, Belmont, MA, 3rd edition.

- Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, 8(1):10–15.
- Blattberg, R. C., Briesch, R., and Fox, E. J. (1995). How promotions work. *Marketing Science*, 14(3):G122.
- Blazenko, G., Eddy-Sumeke, F., and Fu, Y. (2012). Customer price sensitivity and business profit. *SSRN Electronic Journal*, 2(6):84–96.
- Bolton, R. N. (1989). The relationship between market characteristics and promotional price elasticities. *Marketing Science*, 8(2):153.
- Bonami, P. (2023). Basic open-source nonlinear mixed integer programming. <https://www.coin-or.org/Bonmin/Intro.html>. Retrieved: 2025-06-16.
- Chong, E. K. P. and Żak, S. H. (2013). *An introduction to optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, New York.
- Christodoulos A., F. (1995). *Nonlinear and Mixed-Integer Optimization : Fundamentals and Applications*. Topics in Chemical Engineering. Oxford University Press, New York.
- Cohen, M. C., Kalas, J. J., and Perakis, G. (2021). Promotion optimization for multiple items in supermarkets. *Management Science*, 67(4):2340–2364.
- Cohen, M. C., Leung, N. H. Z., Panchangam, K., Smith, A., and Perakis, G. (2017). The impact of linear optimization on promotion planning. *Operations Research*, 65(2):446–468.
- Czyzyk, J., Mesnier, M. P., and Moré, J. J. (1998). The neos server. *IEEE Journal on Computational Science and Engineering*, 5(3):68–75.
- Dang, D., Dearholt, S. L., Bissett, K., Ascenzi, J., and Whalen, M. (2022). *Johns Hopkins evidence-based practice for nurses and healthcare professionals: Model and Guidelines*. Sigma Theta Tau International, Indianapolis.
- DelVecchio, D., Krishnan, H. S., and Smith, D. C. (2007). Cents or percent? the effects of promotion framing on price expectations and choice. *Journal of marketing*, 71(3):158–170.
- Dolan, E. D. (2001). The neos server 4.0 administrative guide. Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory.
- Dolat-Abadi, H. M. (2021). Optimizing decisions of fresh-product members in daily and bourse markets considering the quantity and quality deterioration: a waste-reduction approach. *Journal of Cleaner Production*, 283.
- Felgate, M., Fearne, A., DiFalco, S., and Garcia, M. (2012). Using supermarket loyalty card data to analyse the impact of promotions. *International Journal of Market Research*, 54:221.

- Gaissmaier, T., Heckenbach, B., and Lucht, N. (2012). Paying for performance: Trade spending for profitable growth. Technical report, Boston Consulting Group.
- Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, 20:74–94.
- Gomez, M., Rao, V., and McLaughlin, E. (2007). Empirical analysis of budget and allocation of trade promotions in the u.s. supermarket industry. *Journal of Marketing Research American Marketing Association*, 44:410–424.
- Gropp, W. and Moré, J. J. (1997). Optimization environments and the neos server. In Buhmann, M. D. and Iserles, A., editors, *Approximation Theory and Optimization*, pages 167–182. Cambridge University Press.
- Guadagni, P. M. and Little, J. D. C. (2008). A logit model of brand choice calibrated on scanner data. *Marketing Science*, 27(1):29–48.
- Gümüş, M., Kaminsky, P., and Mathur, S. (2016). The impact of product substitution and retail capacity on the timing and depth of price promotions: Theory and evidence. *International Journal of Production Research*, 54(7):2108–2135.
- Hillier, F. S. (2001). *Introduction to operations research*. McGrawHill, New York, eleventh edition.
- Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–73.
- Horden, H., Bisset, C., and Venter, P. (2023). Investigating the application of mathematical programming models for retail promotion planning. In *Proceedings of the Second International Conference on Industrial Engineering, Systems Engineering and Engineering Management*, Somerset West, South Africa. Southern African Institute for Industrial Engineering (SAIIE).
- Jiang, Y., Shang, J., Liu, Y., and May, J. (2015). Redesigning promotion strategy for e-commerce competitiveness through pricing and recommendation. *International Journal of Production Economics*, 167:257–270.
- Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395.
- Kelley, C. T. (1995). *Iterative Methods for Linear and Nonlinear Equations*, volume Sixteenth of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- King, P. L. and Bigler, C. (2025). Safety stock: A contingency plan to keep supply chains flying high. <https://www.ascm.org/ascm-insights/safety-stock-a-contingency-plan-to-keep-supply-chains-flying-high/>. Retrieved: 2025-03-18.
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.

- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15.
- Kronqvist, J., Bernal Neira, D., Lundell, A., and Grossmann, I. (2019). A review and comparison of solvers for convex minlp. *Optimization and Engineering*, 20:397–455.
- Kumar, A., Adlakaha, A., and Mukherjee, K. (2016). Modeling of product sales promotion and price discounting strategy using fuzzy logic in a retail organization. *Industrial Management & Data Systems*, 166(8):1418–1444.
- Lin, Z. (2016). Price promotion with reference price effects in supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 85:52–68.
- Lougee-Heimer, R. (2003). The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47(1):57–66.
- Ma, S. and Fildes, R. (2017). A retail store sku promotions optimization model for category multi-period profit maximization. *European Journal of Operational Research*, 260(2):680–692.
- Mahar, S., Salzarulo, P. A., and Wright, P. D. (2018). Simultaneous use of customer, product and inventory information in dynamic product promotion. *International Journal of Production Research*, 56(12):4283–4299.
- Martí, R. and Reinelt, G. (2022). *Exact and Heuristic Methods in Combinatorial Optimization: A Study on the Linear Ordering and the Maximum Diversity Problem*. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Mohammadipour, F., Amiri, M., Vanani, I. R., and Soofi, J. B. (2023). Promotion optimization in competitive environments by considering the cannibalization effect. *International Journal of Industrial Engineering*, 30(1).
- Nagle, T. T. and Müller, G. (2017). *The Strategy and Tactics of Pricing: A Guide to Growing More Profitably*. Routledge, London, UK, 6th edition.
- Naseera, S. (2023). P, np, np-hard & np-complete problems. <https://www.jntua.ac.in/gate-online-classes/registration/downloads/material/a159262902029.pdf>. Retrieved: 2025-03-14.
- Nemirovski, A. S. and Todd, M. J. (2008). Interior-point methods for optimization. *Acta Numerica*, 17:191–234.
- Nguyen Thai, P. and Ho, N. (1991). On solving a d.c. programming problem by a sequence of linear programs. *Journal of Global Optimization*, 1(4):399–408.
- Nie, J. (2012). Optimality conditions and finite convergence of lasserre’s hierarchy. *Mathematical Programming*, 146(1-2):97–121.

- Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2nd edition.
- Nutrition Incentive Hub (2022). Retail grocery pricing: A primer. <https://www.nutritionincentivehub.org/media/imihex0v/grocery-retail-pricing.pdf>. Accessed: 2025-08-19.
- Phumchusri, N., Chewcharat, T., and Kanokpongsakorn, S. (2024). Price promotion optimization model for multiperiod planning: a case study of beauty category products sold in a convenience store chain. *Journal of Revenue and Pricing Management*, 23(2):164–178.
- Radasanu, A. C. (2016). Inventory management, service level and safety stock. *Journal of Public Administration, Finance and Law*, 9(9):145–153.
- Sahinidis, N. V. (2017). BARON 21.1.13: Global Optimization of Mixed-Integer Non-linear Programs, *User's Manual*. <http://www.minlp.com/downloads/docs/baron%20manual.pdf>. Retrieved: 2025-05-17.
- Sargent, R. G. (2013). Verification and validation of simulation models. In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 404–414. IEEE.
- Simaan, M. and Cruz Jr, J. B. (1973). On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555.
- Stewart, J. (2016). *Calculus*. Cengage Learning, Australia, 8th edition.
- Tawarmalani, M. and Sahinidis, N. V. (2005). A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249.
- Tellis, G. J. and Zufryden, F. S. (1995). Tackling the retailer decision maze: Which brands to discount, how much, when and why? *Marketing Science*, 14(3):271.
- Tsao, Y.-C. (2016). Pricing and ordering under trade promotion, brand competition and demand uncertainty. *Scientia Iranica*, 23(5):2407–2415.
- van Heerde, H., Leeflang, P., and Wittink, D. (2002). How promotions work: Scan*pro-based evolutionary model building. *Schmalenbach Business Review*, 54:198–220.
- Van Heerde, H. J., Leeflang, P. S. H., and Wittink, D. R. (2001). Semiparametric analysis to estimate the deal effect curve. *Journal of Marketing Research*, 38(2):197–215.
- Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.
- Walters, R. G. (1991). Assessing the impact of retail price promotions on product substitution, complementary purchase, and interstore sales displacement. *Journal of Marketing*, 55(2):17–28.

- Watson, J. (2002). *Strategy: an introduction to game theory*, volume 3. W. W. Norton & Company, New York.
- Woodside, A. G. and Waddle, G. L. (1975). Sales effects of in-store advertising. *Journal of Advertising Research*, 15(3):29–33.
- Yaghin, R., Fatemi Ghomi, s., and Torabi, S. (2014). Enhanced joint pricing and lotsizing problem in a two-echelon supply chain with logit demand function. *International Journal of Production Research*, 52:4967–4983.

Appendix A

Proofs

A.1 Proof of proposition (1)

Proposition 1. *Let $a, b \geq 0$. Let $M > 0$ denote an arbitrary large value and let $0 < Q < 1$ denote an arbitrary small value, such that $Q < M$ and $a - b + Q < M$. Define the binary variable δ as follows:*

$$\delta = \begin{cases} 1 & \text{if } a \geq b \\ 0 & \text{if } a < b \end{cases} \quad (3.15)$$

Then the assignment in (3.15) is equivalently enforced by the following set of mixed-integer linear constraints:

$$a \geq b - M(1 - \delta) \quad (3.16a)$$

$$a + Q \leq b + \delta M \quad (3.16b)$$

$$\delta \in \{0, 1\} \quad (3.16c)$$

Proof. To prove Proposition (1), it must be shown that the set of constraints (3.15 - 3.16b) assigns i) $\delta = 1$ when $a > b$, ii) $\delta = 1$ when $a = b$, and iii) $\delta = 0$ when $a < 0$. First, δ is isolated in (3.16a):

$$a \geq b - M(1 - \delta) \quad (3.16a)$$

$$a - b \geq -M(1 - \delta)$$

$$\frac{-a + b}{M} \leq 1 - \delta$$

$$\frac{-a + b}{M} - 1 \leq -\delta$$

$$\frac{a - b}{M} + 1 \geq \delta \quad (A.2)$$

Similarly, (3.16b) is solved for δ :

$$a + Q \leq b + \delta M \quad (3.16b)$$

$$a + Q - b \leq \delta M$$

$$\frac{a + Q - b}{M} \leq \delta \quad (A.3)$$

Next, $\delta = 0$ and $\delta = 1$ are substituted into (A.2) and (A.3), and the resulting expressions are evaluated to identify valid assignments of δ when a) $a > b$, b) $a = b$, and c) $a < b$.

i) Assignment of δ when $a > b$:

If $a > b$, it first follows that $a - b > 0$, then that $0 < \frac{a-b}{M} < 1$, and finally that $1 < \frac{a-b}{M} + 1 < 2$. It is also noted that $0 < \frac{a-b+Q}{M} < 1$ given the appropriate assignment of Q and M .

As $1 < \frac{a-b}{M} + 1 < 2$, (A.2) is valid when $\delta = 0$ and $\delta = 1$ because:

- when $\delta = 0$, $\frac{a-b}{M} + 1 \geq 0$ **holds**,
- when $\delta = 1$, $\frac{a-b}{M} + 1 \geq 1$ is **holds** (as $1 < \frac{a-b}{M} + 1 < 2$).

Alone, (A.2) does not constrain the value of δ . However, given that $0 < \frac{a-b+Q}{M} < 1$, (A.3) requires $\delta = 1$ for validity, because:

- when $\delta = 1$, $\frac{a-b+Q}{M} \leq 1$ **holds**,
- but when $\delta = 0$, $\frac{a-b+Q}{M} \leq 0$ **does not hold**.

It is concluded that when i) $a > b$, collectively (3.16a - 3.16c) constrain $\delta = 1$.

ii) Assignment of δ when $a = b$:

If $a = b$, then (A.2) simplifies to (A.4):

$$\frac{a - b}{M} + 1 \geq \delta \quad (A.2)$$

$$\frac{0}{M} + 1 \geq \delta$$

$$1 \geq \delta \quad (A.4)$$

Thus, as (A.2) and (A.4) are equivalent, (A.2) holds for both $\delta = 0$ and $\delta = 1$, because:

- when $\delta = 0$, $1 \geq 0$ **holds**,
- when $\delta = 1$, $1 \geq 1$ **holds**.

As had been the case when $a > 0$, (A.2) alone does not enforce $\delta = 1$ when $a = b$, and the δ assignment is again delegated to (A.3). First, it is noted that (A.3) evaluates to (A.5)

when $a = b$:

$$\frac{a + Q - b}{M} \leq \delta \quad (\text{A.3})$$

$$\frac{Q}{M} \leq \delta \quad (\text{A.5})$$

Given the equivalence of (A.3) and (A.5), and the observation that $0 < \frac{Q}{M} < 1$, (A.3) is shown to require $\delta = 1$ for validity, because:

- when $\delta = 1$, $\frac{Q}{M} \leq 1$ **holds**,
- but when $\delta = 0$, $\frac{Q}{M} \leq 0$ **does not hold**.

Thus, when $a = b$, collectively (3.16a - 3.16c) enforces when ii) $\delta = 1$.

iii) Assignment of δ when $a < b$:

If $a < b$, then $a - b < 0$. It follows that $-1 < \frac{a-b}{M} < 0$, and $0 < \frac{a-b}{M} + 1 < 1$. Also, $-1 < \frac{a-b+Q}{M} < 0$ follows from the appropriate assignment of Q and M .

Given that $0 < \frac{a-b}{M} + 1 < 1$, (A.2) is only valid when $\delta = 0$, because:

- when $\delta = 0$, $\frac{a-b}{M} + 1 \geq 0$ **holds**,
- but when $\delta = 1$, $\frac{a-b}{M} + 1 \geq 1$ **does not hold**.

For comprehensiveness, (A.3) is shown to allow both $\delta = 0$ and $\delta = 1$ since $-1 < \frac{a-b+Q}{M} < 0$):

- when $\delta = 0$, $\frac{a-b+Q}{M} \leq 0$ **holds**.
- when $\delta = 1$, $\frac{a-b+Q}{M} \leq 1$ **holds**.

Thus, collectively, (3.16a - 3.16b) assign $\delta = 0$ when iii) $a < b$.

In conclusion, across all logical cases, the valid assignments that follow from (3.16a - 3.16c) correspond to the assignment of δ given in (3.15). Specifically:

- when $a \geq b$, (3.16a - 3.16c) enforce $\delta = 1$,
- and when $a < b$, (3.16a - 3.16c) enforce $\delta = 0$.

Therefore, the logical condition in (3.15) is equivalently enforced by the set of mixed-integer linear constraints (3.16a - 3.16c), completing the proof for proposition (1). $\square$

A.2 Proof of proposition (2)

Proposition 2. Let $\delta_0, \delta_1 \in \{0, 1\}$, $b \geq 0$, and let $M > b$ represent an arbitrary large value. Define the decision variable a as follows:

$$a = \begin{cases} b & \text{if } \delta_0 = \delta_1 = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.17)$$

Then, the assignment in (3.17) is equivalently enforced by the following set of mixed-integer linear constraints:

$$b - M(2 - \delta_0 - \delta_1) \leq a \leq b + M(2 - \delta_0 - \delta_1) \quad (3.18a)$$

$$-\delta_0 M \leq a \leq \delta_0 M \quad (3.18b)$$

$$-\delta_1 M \leq a \leq \delta_1 M \quad (3.18c)$$

$$\delta_0, \delta_1 \in \{0, 1\} \quad (3.18d)$$

Proof. To prove proposition (2), it must be demonstrated that the system of constraints (3.18a - 3.18d) enforces $a = b$ if and only if i) $\delta_0 = \delta_1 = 1$, and enforces $a = 0$ when either ii) $\delta_0 = 1$ and $\delta_1 = 0$, iii) $\delta_0 = 0$ and $\delta_1 = 1$, or iv) $\delta_0 = \delta_1 = 0$. Each of the four cases is considered separately by substituting the appropriate δ_0 and δ_1 values into (3.18a - 3.18d) and evaluating the feasible values of a .

i) Assignment of a when $\delta_0 = \delta_1 = 1$:

When $\delta_0 = \delta_1 = 1$:

- constraint (3.18a) simplifies to:

$$b - M(2 - \delta_0 - \delta_1) \leq a \leq b + M(2 - \delta_0 - \delta_1) \quad (3.18a)$$

$$b - M(2 - 1 - 1) \leq a \leq b + M(2 - 1 - 1)$$

$$b \leq a \leq b \quad (A.7)$$

- constraint (3.18b) simplifies to:

$$-\delta_0 M \leq a \leq \delta_0 M \quad (3.18b)$$

$$-(1) M \leq a \leq (1) M$$

$$-M \leq a \leq M \quad (A.8)$$

- constraint (3.18c) simplifies to:

$$-\delta_1 M \leq a \leq \delta_1 M \quad (3.18c)$$

$$-(1) M \leq a \leq (1) M$$

$$-M \leq a \leq M \quad (A.9)$$

It is noted that the only value of a for which (A.7) holds is $a = b$. Furthermore, (A.8) and (A.9) permit any $a \in [-M, M]$. Thus, (A.7) becomes the binding constraint, enforcing $a = b$ when $\delta_0 = \delta_1 = 1$.

ii) Assignment of a when $\delta_0 = 1$ and $\delta_1 = 0$ When $\delta_0 = 1$ and $\delta_1 = 0$,

- constraint (3.18a) simplifies to:

$$b - M(2 - \delta_0 - \delta_1) \leq a \leq b + M(2 - \delta_0 - \delta_1) \quad (3.18a)$$

$$b - M(2 - 1 - 0) \leq a \leq b + M(2 - 1 - 0)$$

$$b - M \leq a \leq b + M \quad (A.10)$$

- constraint (3.18b) simplifies to:

$$-\delta_0 M \leq a \leq \delta_0 M \quad (3.18b)$$

$$-(1) M \leq a \leq (1) M$$

$$-M \leq a \leq M \quad (A.11)$$

- constraint (3.18c) simplifies to:

$$-\delta_1 M \leq a \leq \delta_1 M \quad (3.18c)$$

$$-(0) M \leq a \leq (0) M$$

$$0 \leq a \leq 0 \quad (A.12)$$

Collectively, (A.10) and (A.11) permits any $a \in [-M, M]$. However, the only value of a for which (A.12) holds is $a = 0$. Thus, when $\delta_0 = 1$ and $\delta_1 = 0$, (A.12) becomes the binding constraint and assigns the value $a = 0$.

iii) Assignment of a when $\delta_0 = 0$ and $\delta_1 = 1$ When $\delta_0 = 0$ and $\delta_1 = 1$,

- constraint (3.18a) simplifies to:

$$b - M(2 - \delta_0 - \delta_1) \leq a \leq b + M(2 - \delta_0 - \delta_1) \quad (3.18a)$$

$$b - M(2 - 0 - 1) \leq a \leq b + M(2 - 0 - 1)$$

$$b - M \leq a \leq b + M \quad (A.13)$$

- constraint (3.18b) simplifies to:

$$-\delta_0 M \leq a \leq \delta_0 M \quad (3.18b)$$

$$-(0) M \leq a \leq (0) M$$

$$0 \leq a \leq 0 \quad (A.14)$$

- constraint (3.18c) simplifies to:

$$-\delta_1 M \leq a \leq \delta_1 M \quad (3.18c)$$

$$\begin{aligned} -(1) M &\leq a \leq (1) M \\ -M &\leq a \leq M \end{aligned} \quad (A.15)$$

While (A.13) and (A.15) collectively requires $a \in [-M, M]$, the only value of a that satisfies (A.14) is $a = 0$. It follows that when $\delta_0 = 0$ and $\delta_1 = 1$, (A.14) assigns $a = 0$.

iv) Assignment of a when $\delta_0 = \delta_1 = 0$ When $\delta_0 = \delta_1 = 0$:

- constraint (3.18a) simplifies to:

$$b - M(2 - \delta_0 - \delta_1) \leq a \leq b + M(2 - \delta_0 - \delta_1) \quad (3.18a)$$

$$\begin{aligned} b - M(2 - 0 - 0) &\leq a \leq b + M(2 - 0 - 0) \\ b - 2M &\leq a \leq b + 2M \end{aligned} \quad (A.16)$$

- constraint (3.18b) simplifies to (A.17):

$$-\delta_0 M \leq a \leq \delta_0 M \quad (3.18b)$$

$$\begin{aligned} -(0) M &\leq a \leq (0) M \\ 0 &\leq a \leq 0 \end{aligned} \quad (A.17)$$

- constraint (3.18c) simplifies to (A.18):

$$-\delta_1 M \leq a \leq \delta_1 M \quad (3.18c)$$

$$\begin{aligned} -(0) M &\leq a \leq (0) M \\ 0 &\leq a \leq 0 \end{aligned} \quad (A.18)$$

Here, (A.16) permits any $a \in [b - 2M, b + 2M]$. However, both (A.17) and (A.18) requires $a = 0$ for feasibility. As such, (A.17) and (A.18) are the binding constraints that assign $a = 0$ when $\delta_0 = \delta_1 = 0$.

To conclude, across all logical cases, the valid assignments that follow from (3.18a - 3.18d) correspond to the assignment of a given in (3.17). Specifically:

- when i) $\delta_0 = \delta_1 = 1$, (3.18a - 3.18d) enforce $a = b$,
- and when either ii) $\delta_0 = 1$ and $\delta_1 = 0$, iii) $\delta_0 = 0$ and $\delta_1 = 1$, or iv) $\delta_0 = \delta_1 = 0$, (3.18a - 3.18d) enforce $a = 0$.

Therefore, the logical condition in (3.17) is equivalently enforced by the set of mixed-integer linear constraints (3.18a - 3.18d), completing the proof for proposition (2). $\square$

Appendix B

Operational models

B.1 Verification models

Listing B.1: The operational model used Assessment I

```
1 # Objective function (P) verification
2
3 ## Set definitions:
4 set N;
5 set K;
6 set V;
7 set W;
8
9 ## Parameter definitions:
10 param c_i {i in N};
11 param r_i {i in N};
12 param q_k {k in K};
13 param d_ik {i in N, k in K};
14 param lambda_i {i in N};
15 param Omega_ivw {i in N, v in V, w in W};
16 param eta_i {i in N};
17 param alpha_hik {h in N, i in N, k in K};
18 param L_ivw {i in N, v in V, w in W};
19 param U_ivw {i in N, v in V, w in W};
20 param M;
21 param Q;
22
23 ## Variable declarations:
24 var gamma_ik {N,K} binary;
25 var phi_i {N};
26 var O_i {N};
27 var omega_ivw {N, V, W};
28 var x_ivw {N, V, W} binary;
29 var y_ivw {N, V, W} binary;
30 var p_i {N};
31
32 ## Objective function:
33 maximize objective:
34     sum {i in N} (
35         r_i[i] * sum{k in K} (gamma_ik[i,k] * q_k[k] * (d_ik[i,k] + phi_i[i])) #
            Sale income
```

```

36      + lambda_i[i] * c_i[i] * (O_i[i] - sum{k in K} gamma_ik[i,k] * (d_ik[i,k] +
      phi_i[i])) # Buy-back income
37      + c_i[i] * sum{w in W}(
38          O_i[i] * omega_ivw[i,0,w] + # Off-invoice allowance income
39          omega_ivw[i,1,w] * sum{k in K} gamma_ik[i,k] * (d_ik[i,k] + phi_i[i])
      # Scan-back income
40      )
41      - c_i[i] * O_i[i] # Cost
42  );
43
44  ## Constraints:
45  subj to aux_0 {i in N, k in K}: gamma_ik[i,k] = 1;
46  subj to aux_1 {i in N}: phi_i[i] = 1;
47  subj to aux_2 {i in N}: O_i[i] = 1;
48  subj to aux_3 {i in N, v in V, w in W}: omega_ivw[i,v,w] = 1;
49  subj to aux_4 {i in N, v in V, w in W}: x_ivw[i,v,w] = 1;
50  subj to aux_5 {i in N, v in V, w in W}: y_ivw[i,v,w] = 1;

```

Listing B.2: The operational model used in Assessment II

```

1  # Constraint (C1) evaluation
2
3  ## Set definition(s)
4  set N;
5  set K;
6
7  ## Variable definition(s)
8  var gamma_ik {N,K} binary; # discount selection indicator (V1)
9
10 ## Objective function
11 maximize objective: 1;
12
13 ## Constraint(s)
14 ### Single price assignment constraint
15     subj to C1 {i in N}: sum{k in K} gamma_ik[i,k] = 1;

```

Listing B.3: Operational model used Assessment III

```

1  # Constraint (C2) evaluation
2
3  ## Set definition(s)
4  set N;
5  set K;
6
7  ## Parameter definition(s)
8  param alpha_hik {h in N, i in N, k in K};
9  param d_ik {i in N, k in K};
10
11 ## Variable definition(s)
12 var gamma_ik {N,K} binary; # discount selection indicator (V1)
13 var phi_i {N}; # cannibalised demand quantity (V3)
14
15
16 ## Objective function
17 maximize objective: 1;
18
19 ## Constraint(s)
20 ### Cross-promotional sales adjustment assignment constraint (C2)
21     subj to C2 {i in N}: phi_i[i] = sum {h in N} sum {k in K} alpha_hik[h,i,k]*
      d_ik[h,k]*gamma_ik[h,k];
22
23 ### Auxiliary constraint(s)

```

```

23     subj to aux_0: gamma_ik[0,0] = 0;
24     subj to aux_1: gamma_ik[1,0] = 0;
25     subj to aux_2: gamma_ik[0,1] = 1;
26     subj to aux_3: gamma_ik[1,1] = 1;

```

Listing B.4: Operational model used in Assessment IV

```

1  # Constraint (C3) evaluation
2
3  ## Set definition(s)
4  set N;
5  set K;
6
7  ## Parameter definition(s)
8  param eta_i {i in N};
9  param d_ik {i in N, k in K};
10
11 ## Variable definition(s)
12 var gamma_ik {N,K} binary; # discount selection indicator (V1)
13 var phi_i {N}; # cannibalised demand quantity (V3)
14 var O_i {N}; # order quantity (V4)
15
16 ## Objective function
17 maximize objective: 1;
18
19 ## Constraint(s)
20 ### Order quantity assignment constraint (C3)
21     subj to C3 {i in N}: O_i[i] = eta_i[i]*(sum{k in K} gamma_ik[i,k]*(d_ik[i,k] +
22         phi_i[i]));
23 ### Auxiliary constraint(s)
24     subj to aux_0: gamma_ik[0,0] = 0;
25     subj to aux_1: gamma_ik[1,0] = 0;
26     subj to aux_2: gamma_ik[0,1] = 1;
27     subj to aux_3: gamma_ik[1,1] = 1;
28     subj to aux_4: phi_i[0] = 30;
29     subj to aux_5: phi_i[1] = 20;

```

Listing B.5: The operational model used in Assessment V

```

1  # Constraint (C4 - C7) and (C11) evaluation
2
3  ## Set definition(s)
4  set N;
5  set K;
6  set V;
7  set W;
8
9  ## Parameter definition(s)
10 param eta_i {i in N};
11 param d_ik {i in N, k in K};
12 param L_ivw {i in N, v in V, w in W};
13 param U_ivw {i in N, v in V, w in W};
14 param M;
15 param Q;
16
17 ## Variable definition(s)
18 var gamma_ik {N,K} binary; # discount selection indicator (V1)
19 var phi_i {N}; # cannibalised demand quantity (V3)
20 var O_i {N}; # order quantity (V4)
21 var x_ivw {N, V, W} binary; # trade spending rate assignment indicator (V6)
22 var y_ivw {N, V, W} binary; # trade spending rate assignment indicator (V7)

```

```

23
24 ## Objective function
25 maximize objective: 1;
26
27 ## Constraint(s)
28 ### Trade spending eligibility constraints (C4 - C7)
29 subj to C4a {i in N, w in W}: O_i[i] >= L_ivw[i,0,w] - M*(1-x_ivw[i,0,w]);
30 subj to C4b {i in N, w in W}: O_i[i] + Q <= L_ivw[i,0,w] + M*x_ivw[i,0,w];
31 subj to C6a {i in N, w in W}: O_i[i] <= U_ivw[i,0,w] + M*(1-y_ivw[i,0,w]);
32 subj to C6b {i in N, w in W}: O_i[i] >= U_ivw[i,0,w] - M*y_ivw[i,0,w] + Q;
33
34 subj to 5a {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) >=
    L_ivw[i,1,w] - M*(1-x_ivw[i,1,w]);
35 subj to 5b {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) + Q
    <= L_ivw[i,1,w] + M*x_ivw[i,1,w];
36 subj to C7a {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) <=
    U_ivw[i,1,w] + M*(1-y_ivw[i,1,w]);
37 subj to C7b {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) >=
    U_ivw[i,1,w] - M*y_ivw[i,1,w] + Q;
38
39 ### Auxiliary constraint(s)
40 ### Uncomment appropriate constraint for instance testing.
41 #subj to instance_i {i in N}: O_i[i] = 9;
42 #subj to instance_ii {i in N}: O_i[i] = 9.9;
43 #subj to instance_iii {i in N}: O_i[i] = 10.0;
44 #subj to instance_iv {i in N}: O_i[i] = 10.1;
45 #subj to instance_v {i in N}: O_i[i] = 20;
46 subj to instance_vi_x {i in N}: O_i[i] = 0;
47
48 subj to aux_5: gamma_ik[0,0] = 0;
49 subj to aux_6: gamma_ik[1,0] = 0;
50 subj to aux_7: gamma_ik[0,1] = 1;
51 subj to aux_8: gamma_ik[1,1] = 1;
52
53 subj to aux_9 {i in N}: phi_i[i] = 0;

```

Listing B.6: The operational model used Assessment VI

```

1 # Constraint (8) evaluation
2
3 ## Set definition(s)
4 set N;
5 set K;
6 set V;
7 set W;
8
9 ## Parameter definition(s)
10 param Omega_ivw {i in N, v in V, w in W};
11 param M;
12 param Q;
13
14 ## Variable definition(s)
15 var x_ivw {N, V, W} binary; # trade spending rate assignment indicator (V6)
16 var y_ivw {N, V, W} binary; # trade spending rate assignment indicator (V7)
17 var omega_ivw {N, V, W}; # trade spending rate (V5)
18
19 ## Objective function
20 maximize objective: 1;
21
22 ## Constraint(s)
23 ### Trade spend assignment constraint(s) (C8)

```

```

24  subj to C8a {i in N, v in V, w in W}: Omega_ivw[i,v,w] -M*(2-x_ivw[i,v,w] -
    y_ivw[i,v,w]) <= omega_ivw[i,v,w];
25  subj to C8b {i in N, v in V, w in W}: omega_ivw[i,v,w] <= Omega_ivw[i,v,w] + M
    *(2 - x_ivw[i,v,w] - y_ivw[i,v,w]);
26  subj to C8ca {i in N, v in V, w in W}: -x_ivw[i,v,w]*M <= omega_ivw[i,v,w];
27  subj to C8cb {i in N, v in V, w in W}: omega_ivw[i,v,w] <= x_ivw[i,v,w]*M;
28  subj to C8da {i in N, v in V, w in W}: -y_ivw[i,v,w]*M <= omega_ivw[i,v,w];
29  subj to C8db {i in N, v in V, w in W}: omega_ivw[i,v,w] <= y_ivw[i,v,w]*M;
30
31  ### Auxiliary constraint(s)
32  #   Modify the following constraints to enact specific instances
33  subj to aux_0: x_ivw[0,0,0] = 1;
34  subj to aux_1: y_ivw[0,0,0] = 1;
35  subj to aux_2: x_ivw[0,1,0] = 1;
36  subj to aux_3: y_ivw[0,1,0] = 1;

```

Listing B.7: The operational model used in Assessment VII

```

1  Constraint (C3) evaluation
2
3  ## Set definition(s)
4  set N;
5  set K;
6
7  ## Parameter definition(s)
8  param q_k {k in K};
9  param r_i {i in N};
10
11  ## Variable definition(s)
12  var gamma_ik {N,K} binary; # discount selection indicator (V1)
13  var p_i {N}; # sale price (V2)
14
15  ## Objective function
16  maximize objective: 1;
17
18  ## Constraint(s)
19  ### Sale price calculation constraint (C9)
20  subj to C9 {i in N}: p_i[i] = sum{k in K} q_k[k] * r_i[i] * gamma_ik[i,k];
21  ### Auxiliary constraint(s)
22  subj to aux_0: gamma_ik[0,0] = 0;
23  subj to aux_1: gamma_ik[1,0] = 0;
24  subj to aux_2: gamma_ik[0,1] = 1;
25  subj to aux_3: gamma_ik[1,1] = 1;
26  subj to aux_4: phi_i[0] = 30;
27  subj to aux_5: phi_i[1] = 20;

```

B.2 Complete operational model

Listing B.8: The complete operational model for the promotion optimisation problem

```

1  # The operational model for the promotion optimisation problem
2  ## Set definitions:
3  set N;
4  set K;
5  set V;
6  set W;
7
8  ## Parameter definitions:

```

```

9  param c_i {i in N}; # item cost
10 param r_i {i in N}; # regular (non-discounted) selling price
11 param eta_i {i in N}; # order quantity ration
12 param q_k {k in K}; # discount factor
13 param d_ik {i in N, k in K}; # demand quantity
14 param alpha_hik {h in N, i in N, k in K}; # Cross-promotional demand
    cannibalisation rate
15 param lambda_i {i in N}; # buy-back rate
16 param Omega_ivw {i in N, v in V, w in W}; # Off-invoice allowance or scan-back rate
17 param L_ivw {i in N, v in V, w in W}; # lower bound of trade spend eligibility
18 param U_ivw {i in N, v in V, w in W}; # upper bound of trade spend eligibility
19 param M = 9999; # sufficiently large number
20 param Q = 0.1; # sufficiently small number
21
22 ## Variable declarations:
23 var gamma_ik {N,K} binary; # discount selection indicator (V1)
24 var p_i {N}; # sale price (V2)
25 var phi_i {N}; # cannibalised demand quantity (V3)
26 var O_i {N}; # order quantity (V4)
27 var omega_ivw {N, V, W}; # active trade spending rate (V5)
28 var x_ivw {N, V, W} binary; # trade spending eligibility indicator (V6)
29 var y_ivw {N, V, W} binary; # trade spending eligibility indicator (V7)
30
31 ## Objective function:
32 maximize Total_Profit:
33     sum {i in N} (
34         sum {k in K} (q_k[k] * r_i[i] * gamma_ik[i,k] * (d_ik[i,k] + phi_i[i]))
35         + lambda_i[i] * c_i[i] * (O_i[i] - phi_i[i] - sum {k in K} gamma_ik[i,k] * (d_ik
36             [i,k] + phi_i[i]))
37         + c_i[i] * sum {w in W} (
38             O_i[i] * omega_ivw[i,0,w] +
39             omega_ivw[i,1,w] * sum {k in K} gamma_ik[i,k] * (d_ik[i,k] + phi_i[i]))
40         - c_i[i] * O_i[i]
41     ); # Total profit (P)
42
43 ## Constraint(s)
44 ### Single price assignment constraint
45     subj to C1 {i in N}: sum {k in K} gamma_ik[i,k] = 1;
46 ### Cross-promotional sales adjustment assignment constraint (C2)
47     subj to C2 {i in N}: phi_i[i] = sum {h in N} sum {k in K} alpha_hik[h,i,k]*
48         d_ik[h,k]*gamma_ik[h,k];
49 ### Order quantity assignment constraint (C3)
50     subj to C3 {i in N}: O_i[i] = eta_i[i]*(sum {k in K} gamma_ik[i,k]*(d_ik[i,k] +
51         phi_i[i]));
52 ### Trade spend eligibility constraints (C4 - C7)
53     subj to C4a {i in N, w in W}: O_i[i] >= L_ivw[i,0,w] - M*(1-x_ivw[i,0,w]);
54     subj to C4b {i in N, w in W}: O_i[i] + Q <= L_ivw[i,0,w] + M*x_ivw[i,0,w];
55     subj to C6a {i in N, w in W}: O_i[i] <= U_ivw[i,0,w] + M*(1-y_ivw[i,0,w]);
56     subj to C6b {i in N, w in W}: O_i[i] >= U_ivw[i,0,w] - M*y_ivw[i,0,w] + Q;
57     subj to 5a {i in N,w in W}: sum {k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) >=
58         L_ivw[i,1,w] - M*(1-x_ivw[i,1,w]);
59     subj to 5b {i in N,w in W}: sum {k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) + Q
60         <= L_ivw[i,1,w] + M*x_ivw[i,1,w];
61     subj to C7a {i in N,w in W}: sum {k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) <=
62         U_ivw[i,1,w] + M*(1-y_ivw[i,1,w]);
63     subj to C7b {i in N,w in W}: sum {k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) >=
64         U_ivw[i,1,w] - M*y_ivw[i,1,w] + Q;
65 ### Trade spend assignment constraint(s) (C8)
66     subj to C8a {i in N, v in V, w in W}: Omega_ivw[i,v,w] - M*(2-x_ivw[i,v,w] -
67         y_ivw[i,v,w]) <= omega_ivw[i,v,w];
68     subj to C8b {i in N, v in V, w in W}: omega_ivw[i,v,w] <= Omega_ivw[i,v,w] + M
69         *(2 - x_ivw[i,v,w] - y_ivw[i,v,w]);

```

```

61  subj to C8ca {i in N, v in V, w in W}: -x_ivw[i,v,w]*M <= omega_ivw[i,v,w];
62  subj to C8cb {i in N, v in V, w in W}: omega_ivw[i,v,w] <= x_ivw[i,v,w]*M;
63  subj to C8da {i in N, v in V, w in W}: -y_ivw[i,v,w]*M <= omega_ivw[i,v,w];
64  subj to C8db {i in N, v in V, w in W}: omega_ivw[i,v,w] <= y_ivw[i,v,w]*M;
65  ### Sale price calculation constraint (C9)
66  subj to C9 {i in N}: p_i[i] = sum{k in K} q_k[k] * r_i[i] * gamma_ik[i,k];

```

B.3 The fixed-margin pricing model

Listing B.9: The sensitivity analysis model to evaluate the fixed-margin pricing strategy

```

1  #Set definitions:
2  set N;
3  set K;
4  set V = 0 .. 1 by 1;
5  set W;
6
7  #Parameter definitions:
8  param c_i {i in N};
9  param r_i {i in N};
10 param q_k {k in K};
11 param d_ik {i in N, k in K};
12 param lambda_i {i in N};
13 param Omega_ivw {i in N, v in V, w in W};
14 param eta_i {i in N};
15 param alpha_hik {h in N, i in N, k in K};
16 param L_ivw {i in N, v in V, w in W};
17 param U_ivw {i in N, v in V, w in W};
18 param M = 9999;
19 param Q = 0.1;
20
21 #Variable declarations:
22 var gamma_ik {N,K} binary;
23 var phi_i {N};
24 var O_i {N};
25 var omega_ivw {N, V, W};
26 var x_ivw {N, V, W} binary;
27 var y_ivw {N, V, W} binary;
28 var p_i {N};
29
30 #Objective function:
31 maximize Total_Profit:
32   sum {i in N}(
33     r_i[i] * sum{k in K} (gamma_ik[i,k] * q_k[k] * (d_ik[i,k] + phi_i[i])) #
34     Sale income
35     + lambda_i[i] * c_i[i] * (O_i[i] - sum{k in K} gamma_ik[i,k] * (d_ik[i,k] +
36     phi_i[i])) # Buy-back income
37     + c_i[i] * sum{w in W}(
38       O_i[i] * omega_ivw[i,0,w] + # Off-invoice allowance income
39       omega_ivw[i,1,w] * sum{k in K} gamma_ik[i,k] * (d_ik[i,k] + phi_i[i])
40       # Scan-back income
41     )
42     - c_i[i] * O_i[i] # Cost
43   );
44
45 #Constraints:
46 #Single price assignment constraint

```

```

45 subj to C1 {i in N}: sum{k in K} gamma_ik[i,k] = 1;
46 #Cross-promotional sales adjustment assignment constraint
47 subj to C2 {i in N}: phi_i[i] = sum {h in N} sum {k in K} alpha_hik[h,i,k]*d_ik[h,
    k]*gamma_ik[h,k];
48 #Order quantity assignment constraint
49 subj to C3 {i in N}: O_i[i] = eta_i[i]*(sum{k in K} gamma_ik[i,k]*(d_ik[i,k] +
    phi_i[i]));
50 #Trade spending eligibility constraints
51 subj to C4 {i in N, w in W}: O_i[i] >= L_ivw[i,0,w] - M*(1-x_ivw[i,0,w]);
52 subj to C5 {i in N, w in W}: O_i[i] + Q <= L_ivw[i,0,w] + M*x_ivw[i,0,w];
53 subj to C6 {i in N, w in W}: O_i[i] <= U_ivw[i,0,w] + M*(1-y_ivw[i,0,w]);
54 subj to C7 {i in N, w in W}: O_i[i] >= U_ivw[i,0,w] - M*y_ivw[i,0,w] + Q;
55
56 subj to C12 {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) >=
    L_ivw[i,1,w] - M*(1-x_ivw[i,1,w]);
57 subj to C13 {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) + Q <=
    L_ivw[i,1,w] + M*x_ivw[i,1,w];
58 subj to C14 {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) <=
    U_ivw[i,1,w] + M*(1-y_ivw[i,1,w]);
59 subj to C15 {i in N,w in W}: sum{k in K} gamma_ik[i,k]*(d_ik[i,k]+phi_i[i]) >=
    U_ivw[i,1,w] - M*y_ivw[i,1,w] +Q;
60 #Trade spending rate assignment constraints
61 subj to C8 {i in N, v in V, w in W}: Omega_ivw[i,v,w] -M*(2-x_ivw[i,v,w] - y_ivw[i,
    v,w]) <= omega_ivw[i,v,w];
62 subj to C9 {i in N, v in V, w in W}: omega_ivw[i,v,w] <= Omega_ivw[i,v,w] + M*(2 -
    x_ivw[i,v,w] - y_ivw[i,v,w]);
63 subj to C10 {i in N, v in V, w in W}: -x_ivw[i,v,w]*M <= omega_ivw[i,v,w];
64 subj to C10_2 {i in N, v in V, w in W}: omega_ivw[i,v,w] <= x_ivw[i,v,w]*M;
65 subj to C11 {i in N, v in V, w in W}: -y_ivw[i,v,w]*M <= omega_ivw[i,v,w];
66 subj to C11_2 {i in N, v in V, w in W}: omega_ivw[i,v,w] <= y_ivw[i,v,w]*M;
67 #Item price calculation constraint
68 subj to AUX1 {i in N}: p_i[i] = sum{k in K} q_k[k] * r_i[i] * gamma_ik[i,k];
69
70 #NB: Constraint to force sale at regular price
71 subj to AUX2 {i in N}: gamma_ik[i,0] = 1;

```

Appendix C

Datasets

Table C.1: Sensitivity analysis baseline dataset

i	λ_i	η_i	a_i	c_i	r_i	$k = 0$	$k = 1$	$k = 2$
						d_{ik}		
0	0.178	1.050	0.189	31.369	43.902	6.833	14.923	24.189
1	0.017	1.050	0.394	12.381	17.344	14.411	26..957	38.737
2	0.084	1.050	0.371	14.333	20.076	27.876	43.708	55.616
						q_k		
						1.000	0.900	0.800

	$h \setminus i$	α_{hik}		
		0	1	2
$k = 0$	0	0.000	0.000	0.000
	1	0.000	0.000	0.000
	2	0.000	0.000	0.000
$k = 1$	0	0.000	-0.263	-0.377
	1	-0.271	0.000	-0.271
	2	-0.266	-0.266	0.000
$k = 2$	0	0.000	-0.377	-0.377
	1	-0.387	0.000	-0.387
	2	-0.381	-0.381	0.000

	$i \setminus w$	L_{iww}			U_{iww}			Ω_{iww}		
		0	1	2	0	1	2	0	1	2
$v = 0$	0	0.000	10.762	17.937	10.755	17.930	8000.000	0.000	0.075	0.150
	1	0.000	38.097	63.496	38.072	63.470	8000.000	0.000	0.075	0.150
	2	0.000	23.504	39.174	23.489	39.158	8000.000	0.000	0.075	0.150
$v = 1$	0	0.000	10.250	17.083	10.243	17.076	8000.000	0.000	0.075	0.150
	1	0.000	36.283	60.472	36.259	60.448	8000.000	0.000	0.075	0.150
	2	0.000	22.385	37.308	22.370	37.293	8000.000	0.000	0.075	0.150

Appendix D

Results

D.1 Sensitivity analysis results

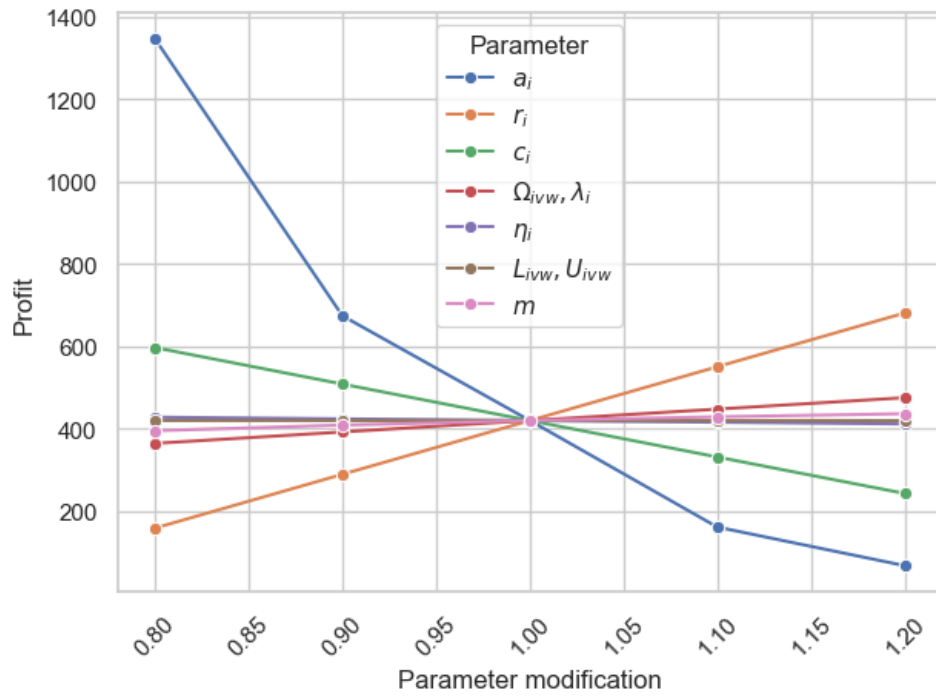


Figure D.1: The impact of varying key parameters on profit (relative to the baseline case).

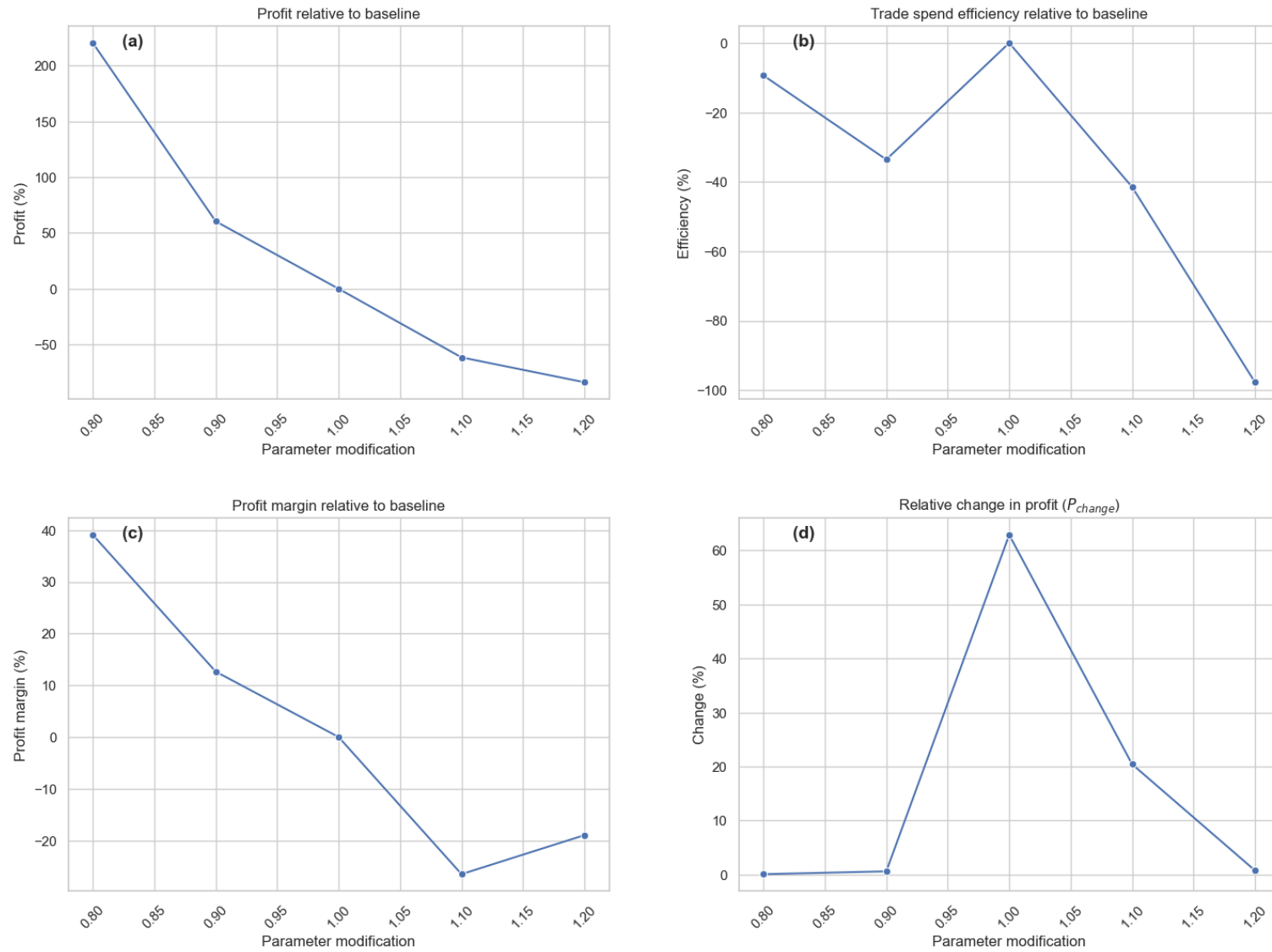


Figure D.2: The impact of varying a_i on key performance indicators. The subfigures show the effect on (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

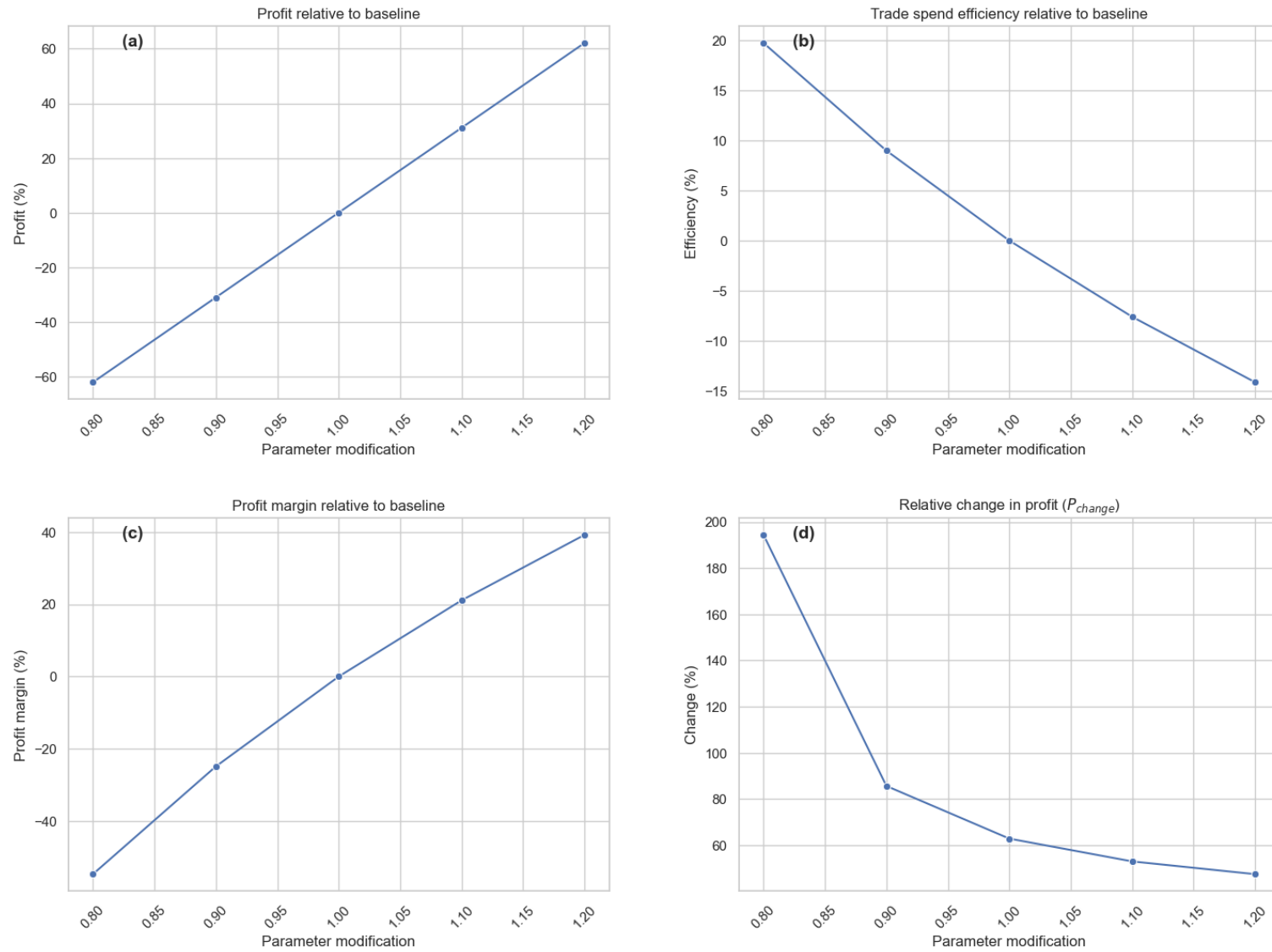


Figure D.3: The impact of varying r_i on key performance indicators. The subfigures show the effect on: (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

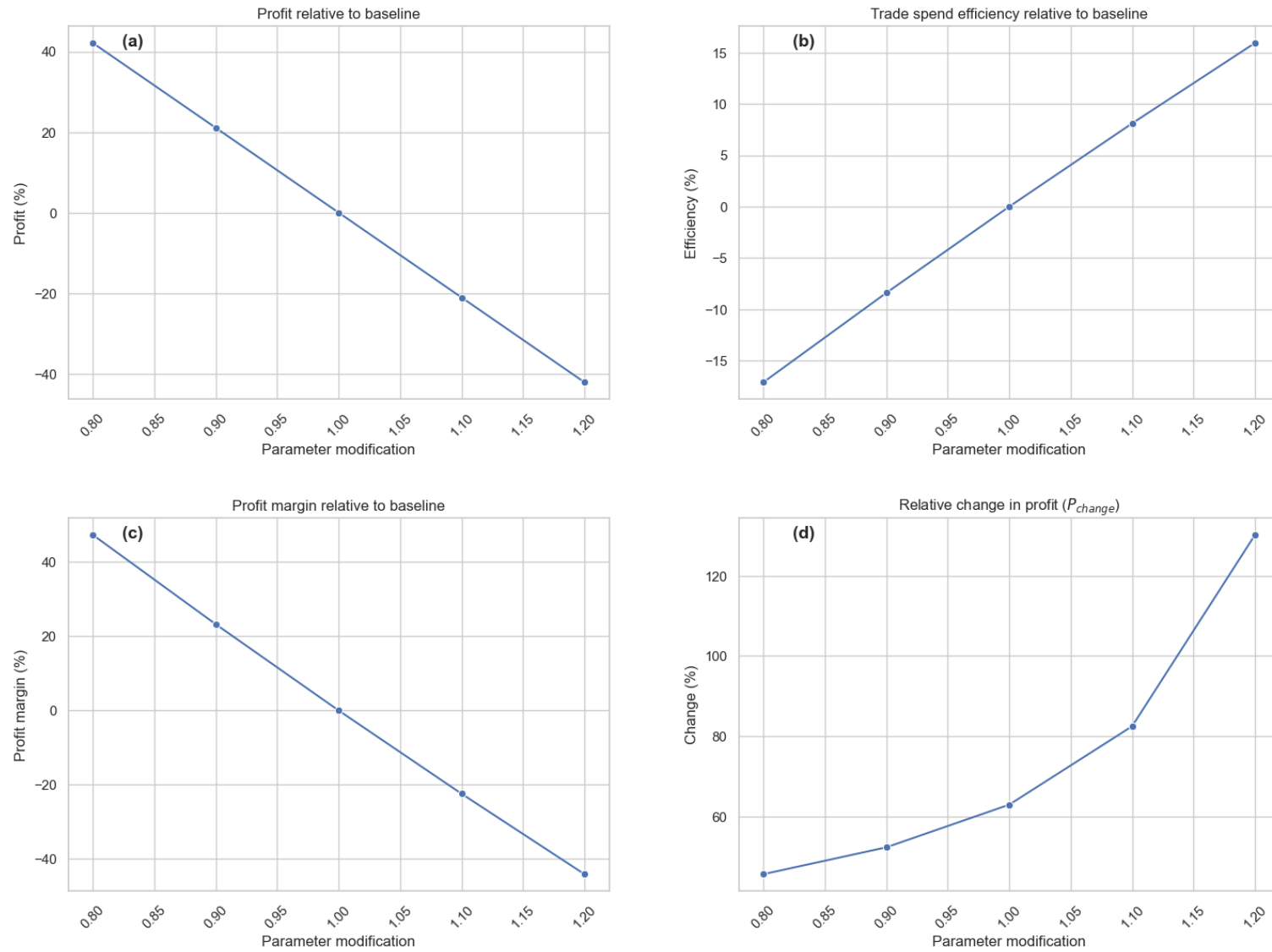


Figure D.4: The impact of varying c_i on key performance indicators. The subfigures show the effect on: (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

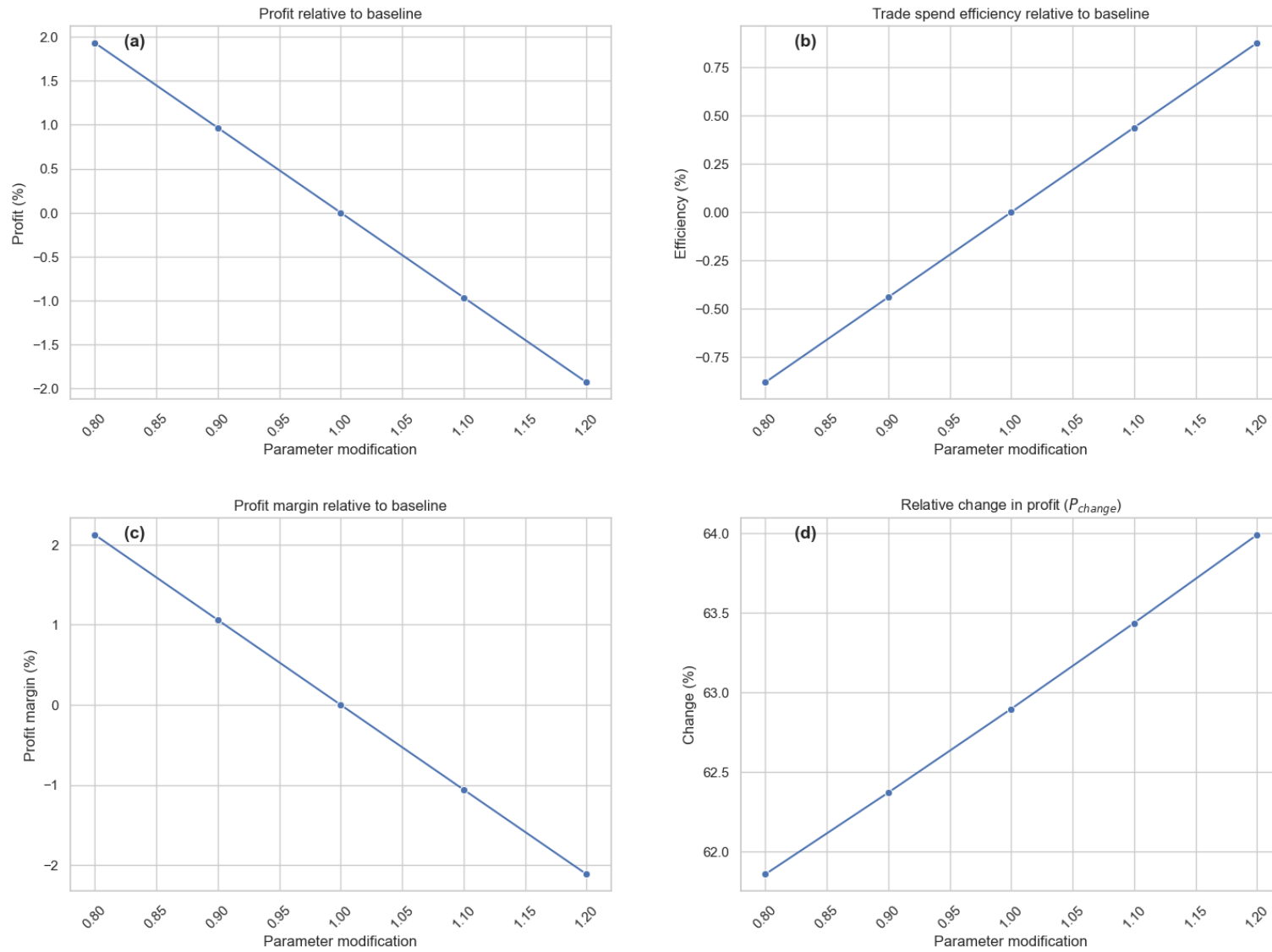


Figure D.5: The impact of varying η_i on key performance indicators. The subfigures show the effect on: (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

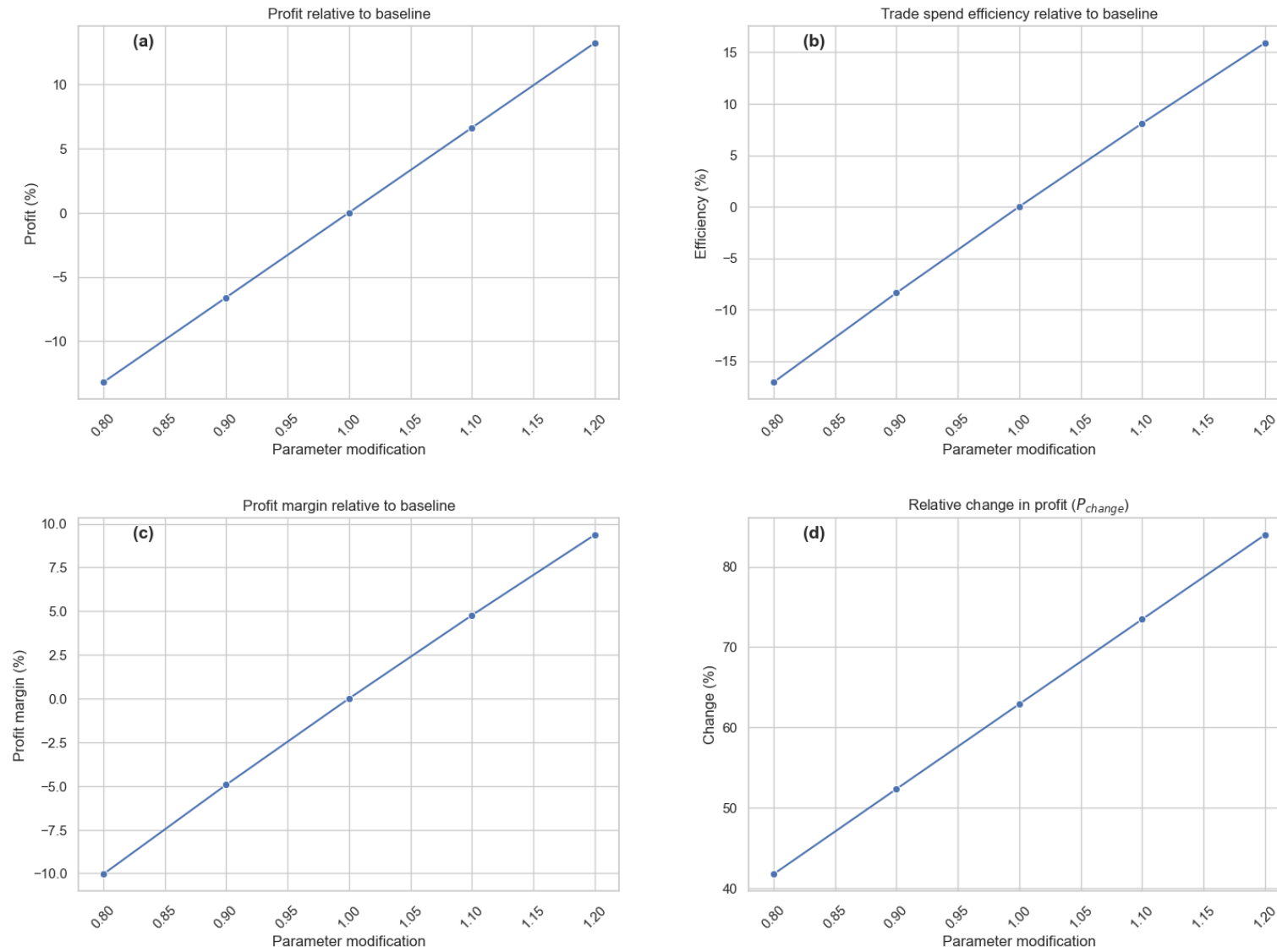


Figure D.6: The impact of varying Ω_{iww} and λ_i on key performance indicators. The subfigures show the effect on: (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

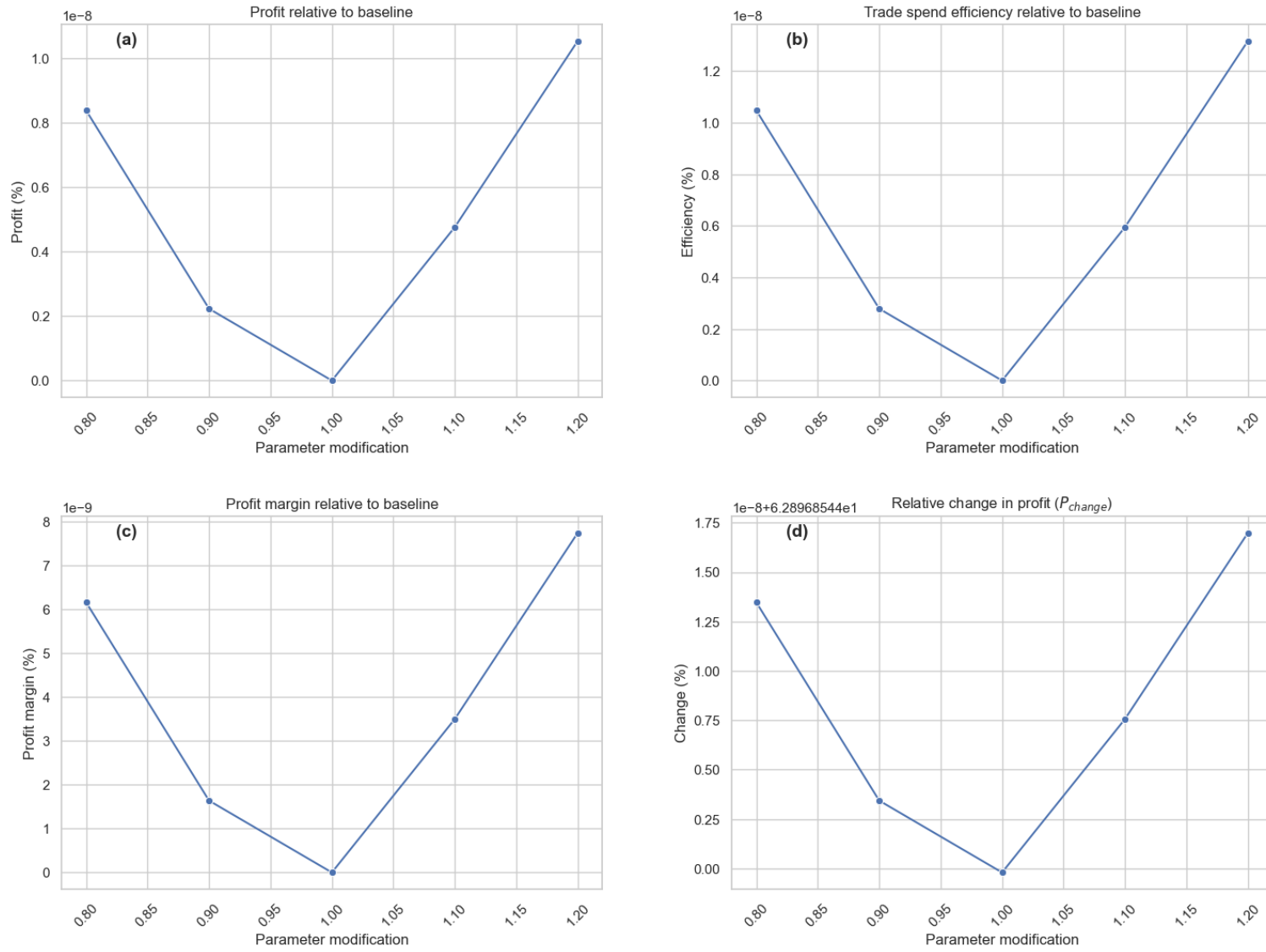


Figure D.7: The impact of varying L_{iw} and U_{iw} on key performance indicators. The subfigures show the effect on: (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

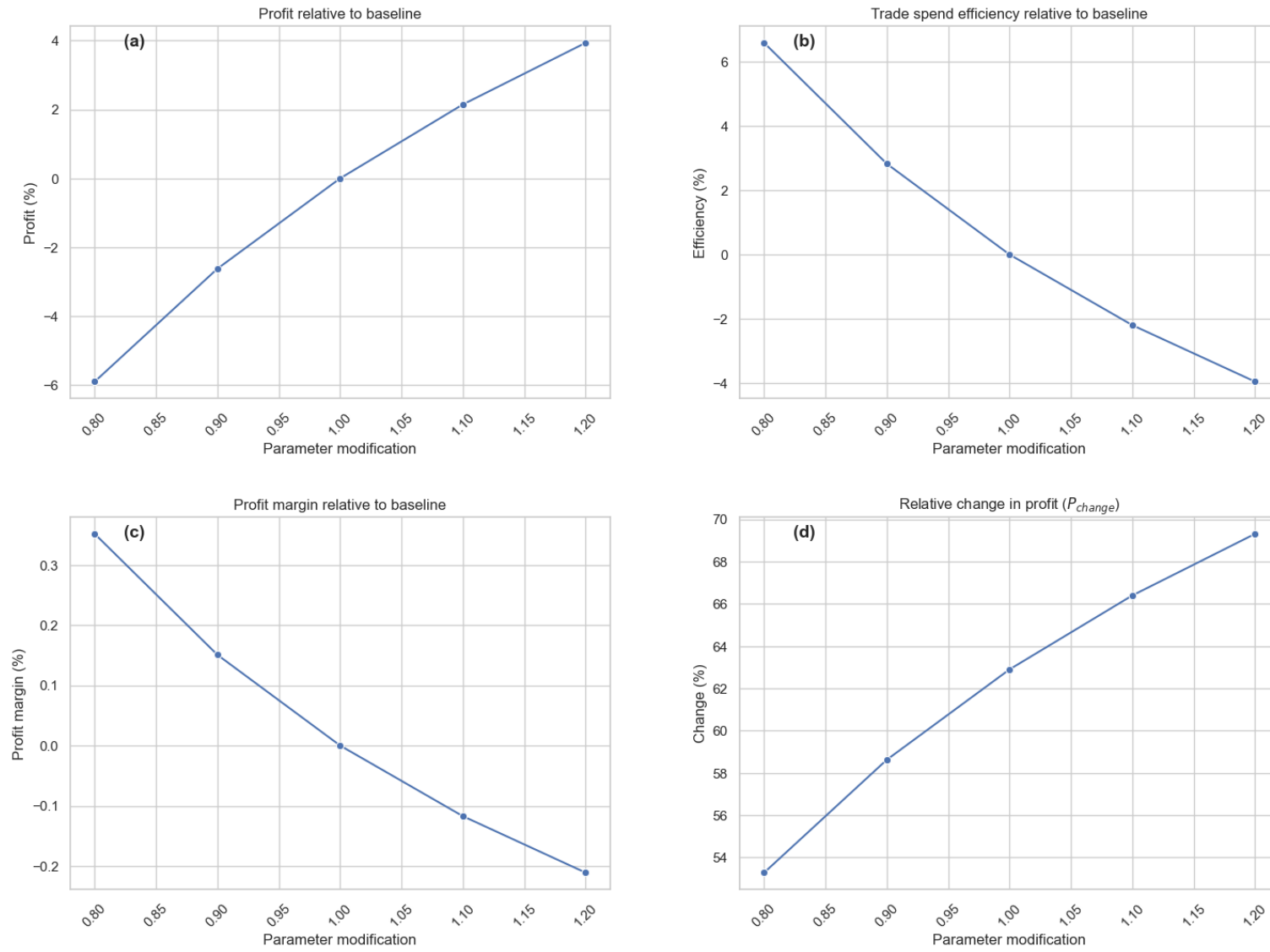


Figure D.8: The impact of varying m on key performance indicators. The subfigures show the effect on: (a) profit, (b) trade spend efficiency, (c) profit margin, and (d) the change in profit between the evaluated strategies.

Table D.1: Numerical results from the sensitivity analysis.

Parameter	Modification (%)	Model output				Alternative pricing strategy				P_{change} (%)
		C	I_s	I_{ts}	P	C	I_s	I_{ts}	P	
a_i	-20	2298.7	3064.93	579.26	1345.49	2298.7	3064.93	578.36	1344.59	0.0
a_i	-10	1579.69	1991.06	262.51	673.89	1383.77	1845.03	208.7	669.96	1.0
a_i	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
a_i	10	666.58	743.16	84.98	161.56	398.04	530.72	1.48	134.16	20.0
a_i	20	247.94	314.63	1.26	67.95	200.24	266.99	0.7	67.44	1.0
r_i	-20	1161.8	1043.44	277.38	159.02	764.05	814.98	3.07	54.01	194.0
r_i	-10	1161.8	1173.87	277.38	289.45	764.05	916.86	3.07	155.88	86.0
r_i	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
r_i	10	1161.8	1434.73	277.38	550.31	764.05	1120.6	3.07	359.63	53.0
r_i	20	1161.8	1565.16	277.38	680.74	764.05	1222.47	3.07	461.5	48.0
c_i	-20	929.44	1304.3	221.9	596.76	611.24	1018.73	2.46	409.95	46.0
c_i	-10	1045.62	1304.3	249.64	508.32	687.64	1018.73	2.77	333.85	52.0
c_i	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
c_i	10	1277.98	1304.3	305.12	331.44	840.45	1018.73	3.38	181.66	82.0
c_i	20	1394.16	1304.3	332.85	242.99	916.86	1018.73	3.69	105.56	130.0
λ_i, Ω_{ivw}	-20	1161.8	1304.3	221.97	364.47	764.05	1018.73	2.46	257.14	42.0
λ_i, Ω_{ivw}	-10	1161.8	1304.3	249.68	392.18	764.05	1018.73	2.77	257.45	52.0
λ_i, Ω_{ivw}	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
λ_i, Ω_{ivw}	10	1161.8	1304.3	305.08	447.58	764.05	1018.73	3.38	258.06	73.0
λ_i, Ω_{ivw}	20	1161.8	1304.3	332.78	475.28	764.05	1018.73	3.69	258.37	84.0
η_i	-20	1150.74	1304.3	274.42	427.99	756.77	1018.73	2.46	264.42	62.0
η_i	-10	1156.27	1304.3	275.9	423.93	760.41	1018.73	2.77	261.09	62.0
η_i	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
η_i	10	1167.33	1304.3	278.86	415.82	767.69	1018.73	3.38	254.43	63.0
η_i	20	1172.87	1304.3	280.33	411.77	771.32	1018.73	3.69	251.1	64.0
L_{ivw}, U_{ivw}	-20	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
L_{ivw}, U_{ivw}	-10	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
L_{ivw}, U_{ivw}	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
L_{ivw}, U_{ivw}	10	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
L_{ivw}, U_{ivw}	20	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
m	-20	1088.02	1205.92	277.19	395.1	764.05	1018.73	3.07	257.76	53.0
m	-10	1129.01	1260.58	277.3	408.86	764.05	1018.73	3.07	257.76	59.0
m	0	1161.8	1304.3	277.38	419.88	764.05	1018.73	3.07	257.76	63.0
m	10	1188.63	1340.07	277.45	428.89	764.05	1018.73	3.07	257.76	66.0
m	20	1210.99	1369.89	277.5	436.4	764.05	1018.73	3.07	257.76	69.0

D.2 Solver accuracy assessment results

Table D.2: Comparison of heuristic (BONMIN) solutions with exact (BARON) solutions for 20 simulated instances. The table reports the objective function value and selected decision variables for each solve.

Instance	BONMIN results								BARON results							
	Profit	p_0	p_1	p_2	p_3	p_4	p_5	p_6	Profit	p_0	p_1	p_2	p_3	p_4	p_5	p_6
1	246.68	34.59	28.22	29.72	N/A	N/A	N/A	N/A	246.42	34.59	28.22	29.72	N/A	N/A	N/A	N/A
2	324.39	28.12	29.64	30.76	N/A	N/A	N/A	N/A	323.87	28.12	29.54	30.76	N/A	N/A	N/A	N/A
3	375.00	54.16	20.84	17.53	N/A	N/A	N/A	N/A	374.66	54.16	20.84	17.53	N/A	N/A	N/A	N/A
4	632.48	21.52	23.07	48.59	N/A	N/A	N/A	N/A	631.78	21.52	23.07	48.59	N/A	N/A	N/A	N/A
5	656.78	45.34	49.95	29.65	N/A	N/A	N/A	N/A	655.51	45.34	49.95	29.65	N/A	N/A	N/A	N/A
6	111.04	17.34	17.28	18.52	N/A	N/A	N/A	N/A	110.89	17.34	17.28	18.52	N/A	N/A	N/A	N/A
7	482.34	42.41	69.85	21.75	N/A	N/A	N/A	N/A	481.77	42.41	69.84	21.75	N/A	N/A	N/A	N/A
8	623.74	46.86	22.72	33.85	N/A	N/A	N/A	N/A	623.03	46.86	22.72	33.85	N/A	N/A	N/A	N/A
9	481.87	19.17	54.19	35.59	N/A	N/A	N/A	N/A	481.53	19.17	54.19	35.59	N/A	N/A	N/A	N/A
10	813.10	17.66	59.85	23.62	N/A	N/A	N/A	N/A	812.25	17.66	59.85	23.62	N/A	N/A	N/A	N/A
11	370.46	18.97	21.60	39.80	23.59	37.76	N/A	N/A	370.02	18.97	21.60	39.80	23.59	37.76	N/A	N/A
12	309.92	23.47	37.36	29.50	22.43	19.16	N/A	N/A	309.49	23.47	37.36	29.50	22.43	19.16	N/A	N/A
13	395.45	31.67	39.85	19.74	18.93	19.98	N/A	N/A	394.99	31.67	39.85	19.74	18.93	19.98	N/A	N/A
14	543.55	38.50	53.00	44.28	22.39	47.50	N/A	N/A	542.93	38.50	53.00	44.28	22.39	47.50	N/A	N/A
15	366.06	34.78	21.50	22.95	40.82	19.88	N/A	N/A	365.64	34.78	21.50	22.95	40.82	19.88	N/A	N/A
16	2892.92	14.64	9.22	10.25	21.93	23.31	10.68	25.12	2892.92	14.64	9.22	10.25	21.92	23.31	10.68	25.12
17	4436.07	12.96	45.73	21.13	20.12	29.70	12.61	19.82	4436.07	12.96	45.73	21.13	20.12	29.70	12.61	19.82
18	4798.65	33.44	47.18	21.47	11.25	25.72	16.89	22.11	4798.65	33.44	47.18	21.47	11.25	25.72	16.89	22.11
19	3245.41	15.28	24.38	22.57	11.55	20.76	12.91	11.47	3245.41	15.28	24.38	22.57	11.55	20.76	12.91	11.47
20	5044.96	31.64	25.41	23.58	20.79	22.36	38.44	19.48	5044.96	31.64	25.41	23.58	20.79	22.36	38.44	19.48