Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Магістерська робота
освітній ступінь – магістр

на тему: **«ІТЕРАЦІЙНІ МЕТОДИ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ З ВИКОРИСТАННЯМ МЕТОДУ ДИСКРЕТНИХ ФУНКЦІОНАЛЬНИХ ЧАСТИНОК»**

Виконав: студент 2-го року навчання,
Спеціальності
113 Прикладна математика

Авдєєнко Іван Максимович

Керівник: Дрінь С. С.
кандидат фіз.-мат. наук, ст. викладач

Рецензент:
____

Магістерська робота захищена
з оцінкою __________________________

Секретар ЕК________________________
(підпис)

«______» _________________ 20___р.

Київ - 2025

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

Освітній ступінь магістра

Спеціальність 113 Прикладна математика

Освітньо-наукова програма "Прикладна математика"

ЗАТВЕРДЖУЮ

Завідувач кафедри

____

____

"____"________________ 20__ року

ЗАВДАННЯ

ДЛЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Авдєєнко Івану Максимовичу

1. Тема роботи: «Ітераційні методи для розв'язання задач оптимізації з використанням методу дискретних функціональних частинок»

керівник роботи: Дрінь Світлана Сергіївна, кандидат фіз.-мат. наук, ст. викладач

затверджені наказом вищого навчального закладу

від "___" ________ 20__ року № ___

2. Строк подання студентом роботи:

3. План роботи:

   1. Аналіз існуючих ітераційних методів оптимізації та їх застосувань.
   2. Вивчення методу дискретних функціональних частинок.
   3. Модифікація ітераційного підходу для вирішення задач прогнозування в умовах невизначеності.
   4. Розробка математичної моделі для прогнозування оптимального набору товарів з урахуванням економічних параметрів.
   5. Проведення комп'ютерного моделювання для валідації результатів.
   6. Аналіз ефективності розробленого підходу порівняно з іншими методами оптимізації.

Графік підготовки магістерської роботи до захисту

№ з/п	**ПЕРЕЛІК РОБІТ**	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітки
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника. Узгодження календарного графіка підготовки магістерської роботи. Ознайомлення студента з критеріями оцінювання магістерської роботи.	жовтень	16.10.2024		
2.	Вивчення джерел, літератури, періодичних видань, наукових публікацій, збір та узагальнення фактів, даних.	жовтень - листопад	16.11.2024		
3.	Складання плану магістерської роботи та узгодження із науковим керівником.	листопад	22.11.2024		
4.	Постановка експерименту, аналіз отриманих результатів наукового дослідження.	листопад - березень	14.01.2025		
5.	Проміжний контроль виконання роботи.	лютий	15.02.2025		
6.	Написаня кваліфікаційної роботи в цілому, ознайомлення зі першим варіантом наукового керівника.	січень - березень	28.03.2025		
	Розділ 1 (постановка проблеми, теоретичні основи, огляд літературних джерел).	квітень	04.04.2025		
	Розділ 2 (аналітично-дослідницька частина).	квітень	15.04.2025		
	Розділ 3 (проєктно-рекомендаційна частина).	квітень	26.04.2025		
7.	Повне завершення написання кваліфікаційної роботи, оформлення її згідно з вимогами й подання на відгук науковому керівнику.	квітень - травень	25.05.2025		
8.	Подання магістерської роботи для перевірки письмових робіт студентів НаУКМА на відповідність вимогам академічної доброчесності.	початкок червня	08.06.2025		
9.	Подання на зовнішню рецензію.	початкок червня	10.05.2025		
10.	Підготовка до захисту магістерської роботи на засіданні кафедри: написання доповіді та виготовлення ілюстративного матеріалу.	**до 13 червня**	08.06.2025		
11.	Підготовка до захисту магістерської роботи на засіданні кафедри: написання доповіді та виготовлення ілюстративного матеріалу.	**до 08 червня**	13.06.2025		
12.	Подання магістерської роботи на кафедру з усіма супроводжуючими документами.	**до 08 червня**	08.06.2024		
13.	Публічний захист перед екзаменаційною комісією	згідно з розкладом роботи ЕК	13.06.2025		

Графік узгоджено «_____» _________________ 2024 р.

Науковий керівник ___________________________________
(ПІБ)

Виконавець магістерської роботи ___________________________________
(ПІБ)

Contents

List of symbols and abbreviations

ACF	Autocorrelation Function
ARIMA	Autoregressive Integrated Moving Average
CVaR	Conditional Value at Risk
DFPM	Discrete Functional Particle Method
IER	Inventory Efficiency Ratio
PACF	Partial Autocorrelation Function
SARIMAX	Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors
SKU	Stock Keeping Unit
UAH	Ukrainian Hryvnia (currency)
MAPE	Mean Absolute Percentage Error
TO	Turnover (total sales revenue)
`Q`	Matrix of sales quantities (dates $\times$ SKUs)
`P`	Matrix of sales revenue (turnover)
`L_UAH`	Matrix of end-of-month leftover value in UAH
$\mathbf{w}$	Vector of assortment weights
$\boldsymbol{\mu}$	Expected return vector (average revenues)
$\boldsymbol{\Sigma}$	Risk (covariance) matrix of IER
α	Blending factor for hybrid portfolio
$\Delta t, \eta$	DFPM time step and damping coefficient

Abstract

This master's thesis addresses the challenge of assortment planning in retail under uncertain demand and operational constraints. It develops a hybrid methodology that integrates SARIMAX time-series forecasting with the Discrete Functional Particle Method (DFPM) for optimisation, enabling both strategic (long-term) and tactical (monthly) decision support. Key elements include the preprocessing of real sales and inventory data, construction of an Inventory Efficiency Ratio as a risk metric, eigenvalue-guided tuning of DFPM parameters, and blending of mathematically optimal weights with historical baselines. The framework is implemented in Python and structured into theoretical foundations, practical application, and performance analysis. The work contributes a flexible, data-driven approach to improve assortment decisions in dynamically changing retail environments.

Анотація

Ця магістерська робота присвячена задачі планування асортименту в роздрібній торгівлі за умов невизначеності попиту та операційних обмежень. Розроблено гібридну методологію, яка поєднує часові ряди SARIMAX для прогнозування з методом дискретних функціональних частинок (DFPM) для оптимізації, забезпечуючи підтримку як стратегічних (довгострокових), так і тактичних (щомісячних) рішень. Основні елементи включають попередню обробку реальних даних з продажів та запасів, побудову коефіцієнта ефективності запасів (Inventory Efficiency Ratio) як метрики ризику, налаштування параметрів DFPM на основі спектральних властивостей матриці ризику та об'єднання математично оптимальних ваг із історичними базовими показниками. Рамковий підхід реалізовано на Python і поділено на теоретичні засади, практичну реалізацію та аналіз результатів. Запропонований підхід забезпечує адаптивну оптимізацію асортименту на основі даних, що дозволяє оперативно реагувати на швидкі зміни ринку.

Introduction

Relevance

Modern companies face immense pressure to accelerate and refine decisions related to product assortment due to rapid changes and competition in the retail landscape. Today's business and data constraints make traditional heuristic methods inadequate. At the same time, the escalation of optimization theory and forecasting enables the designing of robust and flexible intelligent systems for ancillary decision-making systems.

The practical application of Markowitz (1952) mean-variance portfolio theory in retail has proved suboptimal. This stems from the divergence of 'risk' in retail compared to finance. Risk is not volatile pricing anymore; it is operational inefficiencies, such as dormant capital in inventory.

In addition, simplistic demand forecasting can be narrow-sighted, overemphasizing short-term reactionary tactical shifts while sidelining overarching strategic seasonality. Bridging the gap between business logic and mathematical optimization requires integrated cross-functional approaches.

This thesis attempts to fill that gap by developing a new, multi-layered framework for assortment optimization. Its relevance stems from creating a hybrid approach that combines iterative optimization using sophisticated algorithms with prediction based on data and practical business heuristics. This research uses the Discrete Functional Particle Method (DFPM) in retail. Also, it proposes a new operational risk measure, the Inventory Efficiency Ratio, thus providing a more helpful tool for retailers' problems.

Creating such a model is crucial because it offers a route towards evidence-based business intelligence that can lead to profound, measurable enhancements in organizational productivity. This research directly addresses the urgent need to enhance capital productivity while providing a competitive edge within a rigorous environment by formulating a method to reduce the costs of holding stock while increasing the stock turnover rate sharply.

The object of this research is the process of managing and optimizing a product assortment within a retail business context. Specifically, the study focuses on a real-world case involving a portfolio of products from the "Antistress

Toys" category, using historical data on sales, revenue, and inventory levels.

The subject of this research is the development and application of iterative optimization methods for solving the assortment planning problem. This includes:

- The adaptation of the Discrete Functional Particle Method (DFPM) for a retail-specific objective function based on operational risk.
- The creation of a hybrid strategic framework that balances mathematical optimization with historical business data.
- The integration of forecasting models and business logic constraints to form a comprehensive, multi-stage decision-making methodology.

Research aim and objectives

The aim of this research is to develop and validate a hybrid methodology for retail assortment planning under uncertainty by integrating iterative optimization techniques — particularly the Discrete Functional Particle Method (DFPM) — with time series forecasting models, in order to improve the efficiency and robustness of strategic and tactical decision-making.

Objectives:

- To review existing iterative optimization methods and their application in unconstrained optimization problems.
- To study the Discrete Functional Particle Method (DFPM) and explore its use in solving unconstrained optimization tasks.
- To adapt the iterative approach for effective demand forecasting under uncertainty.
- To construct a mathematical model for forecasting the optimal product assortment, incorporating key economic indicators.
- To implement a computational simulation to validate the proposed model using real-world retail data.

- To analyze the performance of the developed approach in comparison with alternative optimization techniques.

Research methods

This study links forecasting with optimisation to build a practical assortment-planning tool. The work uses a compact five-step pipeline:

- **Theory scan** — a brief survey of iterative optimisation and time-series models to justify the choice of DFPM for optimisation and SARIMAX for forecasting.
- **Data prep** — reshaping raw sales / inventory tables to time-series format, handling outliers and missing values.
- **Demand forecast** — 12-month ahead SARIMAX forecasts (with exogenous drivers) for each SKU.
- **Optimisation** — DFPM minimises the covariance-based risk subject to revenue and budget constraints; the result is blended with historical weights for robustness.
- **Application of business constraints** — enforcing minimum and maximum weight bounds per category and re-normalising to ensure diversity and feasibility in the final assortment plan.

All experiments are coded in `Python 3.12` using `NumPy`, `pandas`, `statsmodels`, and `SciPy`.

Scientific novelty

This research introduces a novel hybrid framework that seamlessly integrates SARIMAX-based demand forecasting with the Discrete Functional Particle Method (DFPM). This enables end-to-end strategic and tactical assortment planning under uncertainty. By deriving DFPM's step size (Δt) and damping coefficient (η) from the spectral properties of the inventory-efficiency covariance matrix, the method achieves rapid and stable convergence in practical settings. A new risk metric—the Inventory Efficiency Ratio (IER), which combines leftover stock and revenue into a unified covariance structure—is proposed and optimised directly through DFPM. To guard against overly pessimistic forecasts, a forecast-floor mechanism enforces a minimum bound relative to recent sales, ensuring that tactical adjustments remain business-meaningful. Finally, this methodology is validated on real Ukrainian retail data, demonstrating significant improvements in inventory turnover and leftover reduction.

Dissemination of results

The core results of this master's thesis were presented by Avdieienko (2025) at the XIII All-Ukrainian scientific conference of young mathematicians, which took place on May 9, 2025, in Kyiv. The presentation covered the development of a framework that utilizes the Discrete Functional Particle Method (DFPM) to optimize retail assortment planning under uncertainty.

Structure of the thesis

This thesis is organized into four main chapters, preceded by preliminary front matter and followed by concluding remarks and references. It begins with the *Introduction*, which frames the problem of retail assortment planning under uncertainty, motivates the integration of forecasting and optimization, and lays out the research object and subject, aim and objectives, methods, and scientific novelty. A brief note on the dissemination of results completes this chapter, establishing the context and scope of the work.

The second chapter, the *Theoretical Part*, provides the mathematical and algorithmic foundations. It opens with an overview of classical and modern iterative optimization methods—gradient descent, quasi-Newton schemes, and particle-based approaches—followed by a concise treatment of time-series models, particularly ARIMA and its seasonal extension SARIMAX with exogenous regressors. Building on these foundations, the chapter formulates the product-assortment problem as a constrained quadratic programme, introduces the Discrete Functional Particle Method (DFPM) for its solution, and details how eigenvalue-based tuning of DFPM parameters ensures stable convergence. Finally, it presents the hybrid methodology that unites strategic (long-term) and tactical (monthly) planning by blending mathematically optimal weights with historical baselines and seasonality adjustments.

In the third chapter, *Practical Application and Software Implementation*, the focus shifts to data and code. It begins by describing the raw sales, inventory, exogenous-metrics datasets, preprocessing steps—reshaping to time-series format, outlier treatment, and construction of pivot matrices for revenue, quantity, and the Inventory Efficiency Ratio. The chapter then outlines the Python implementation: SARIMAX forecasting (with auto-tuned orders and external drivers), DFPM optimization, hybrid blending, and the application of business constraints. It concludes with reporting and visualization tools, detailed analysis of results, and an assessment of aggregate performance (turnover, leftovers, cost) and operational risk reduction.

The last chapter, titled *Conclusion*, compiles the main findings, considers the limitations of the current methodology, and proposes directions for future studies, including expanding the framework to encompass multi-category settings or the inclusion of non-linear business constraints. This organization guarantees a coherent flow from theoretical concepts to practical application and strategic advice.

1 Theoretical part

1.1 Analysis of existing iterative optimization methods

Optimizing a product assortment is a complicated task involving forecasting techniques and advanced optimization methods. A strong theoretical foundation is important to create a solid solution from the beginning. This chapter overviews the primary iterative methods supporting this thesis's methodology.

We will begin by examining classical approaches that form the building blocks of our solution. This includes a review of Gradient Descent, which provides the core principles for iteratively finding the minimum of an objective function. We will also examine time series forecasting models, specifically the ARIMA model and its powerful extension, SARIMAX. These methods are crucial for predicting future product demand, a key input for any data-driven optimization in retail.

Gradient Descent

Gradient Descent is an iterative optimization algorithm used to find a local minimum of a differentiable function (Boyd and Vandenberghe (2004a)). The core idea is to continuously follow the path of the steepest descent by walking in the negative direction of the slope of the objective function.

The process begins with an initial guess, $x^{(0)}$, and updates it iteratively using the rule:

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f\big(x^{(k)}\big),$$

where:

- $x^{(k)} \in \mathbb{R}^n$ is the vector of variables (e.g., product quantities) at iteration k.
- $\alpha_k > 0$ is the step size, controlling the magnitude of the update.
- $\nabla f(x^{(k)})$ is the gradient of the objective function f at the point $x^{(k)}$, which indicates the direction of the function's greatest rate of increase.

To apply this method to an assortment optimization problem, we first define an objective function that models the total operational cost. This formulation is inspired by the classic Newsvendor problem, which addresses decision-making under uncertainty (Arrow et al. (1951)). For a single product i, the cost function $f_i(x_i)$ can be formulated as:

$$f_i(x_i) \;=\; \underbrace{c_i\, x_i}_{\text{Procurement Cost}} \;-\; \underbrace{p_i\, \min\{x_i,\, d_i\}}_{\text{Revenue}} + \underbrace{h_i\, \max\{0,\, x_i - d_i\}}_{\text{Holding Cost}},$$

where c_i is the unit cost, p_i is the selling price, h_i is the holding cost for one unsold unit, and d_i is the forecasted demand for product i.

A key requirement for Gradient Descent is a differentiable objective function. The standard min and max functions are non-differentiable. To resolve this, we use their smooth approximations, known as "soft-min" and "soft-max" (Goodfellow et al. (2016)), parameterized by a small $\varepsilon > 0$:

$$\min_{\varepsilon}(a, b) \;=\; -\,\varepsilon\, \ln\!\big(e^{-a/\varepsilon} + e^{-b/\varepsilon}\big) \quad \text{and} \quad \max_{\varepsilon}(a, b) \;=\; \varepsilon\, \ln\!\big(e^{a/\varepsilon} + e^{b/\varepsilon}\big).$$

This smoothing allows for the gradient computation for each product's cost function, $\nabla f_i(x_i)$. The total objective function for the entire assortment is then the sum of these individual functions,

$$F(x) = \sum_{i=1}^{n} f_i(x_i)$$

The iterative update step for the entire vector of quantities x can then be written as:

$$x^{(k+1)} \;=\; x^{(k)} \;-\; \alpha_k\, \nabla F\big(x^{(k)}\big),$$

where the components of the total gradient $\nabla F(x^{(k)})$ are calculated based on the partial derivatives of the soft-min and soft-max functions for each product. The process terminates when a stopping criterion is met, such as the norm of the gradient being sufficiently small, $\|\nabla F(x^{(k)})\| < \delta$.

The final continuous solution x^* is then post-processed to meet practical requirements, such as rounding to non-negative integer quantities ($x_i \in \mathbb{Z}^+$) and satisfying any overall budget constraints, for instance $\sum_{i=1}^{n} c_i\, x_i \leq B$.

ARIMA (p, d, q)

The Autoregressive Integrated Moving Average (ARIMA) model, introduced by Box et al. (2015), is a fundamental method for time series forecasting. It describes the evolution of a variable based on its own past values and past forecast errors. The model is defined by three key parameters: p, d, and q.

The ARIMA modelling procedure begins with the non-stationary time series y_t, which must first be transformed through differencing to become stationary. A stationary time series is characterized by its statistical properties, such as mean and variance, which remain constant over time, unlike non-stationary time series. This is accomplished through the application of a differencing operator $(1 - B)^d$:

$$y_t^{(d)} = (1 - B)^d \, y_t,$$

where:

- y_t is the value of the time series at time t.
- B is the backshift operator, defined such that $B \, y_t = y_{t-1}$.
- d is the order of differencing, representing the number of times the series is differenced to achieve stationarity.

Once the series is stationary, its structure is modelled by the Autoregressive (AR) and Moving Average (MA) parts. The complete ARIMA model is written as:

$$\Phi(B) \, y_t^{(d)} = \Theta(B) \, \varepsilon_t,$$

where $\Phi(B) = 1 - \phi_1 \, B - \cdots - \phi_p \, B^p$ is the autoregressive polynomial of order p, which models the dependency of the current value on p previous values of the series. The term $\Theta(B) = 1 + \theta_1 \, B + \cdots + \theta_q \, B^q$ is the moving average polynomial of order q, modelling the dependency on q previous forecast errors. Finally, ε_t denotes the white noise error term at time t, which is generally presumed to be independently and identically distributed with an average of zero and a constant variance, specifically $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$.

The model's parameters, namely the autoregressive coefficients $\phi = \{\phi_j\}_{j=1}^{p}$, the moving average coefficients $\theta = \{\theta_k\}_{k=1}^{q}$, and the error variance σ^2, are unknown and must be estimated from the data. This is typically done by maximizing the log-likelihood function. Assuming Gaussian errors, the function is Hamilton (1994):

$$\ell\big(\phi, \theta, \sigma^2\big) = -\frac{T}{2}\ln\big(2\pi\sigma^2\big) - \frac{1}{2\,\sigma^2}\sum_{t=1}^{T}\big(\varepsilon_t(\phi, \theta)\big)^2,$$

where T is the number of observations and $\varepsilon_t(\phi, \theta)$ are the residuals calculated from the model's equation.

Since this function is non-linear in its parameters, maximization is performed iteratively using a numerical optimization algorithm. A common choice is a quasi-Newton method (e.g., BFGS), which uses information about the function's curvature to make more efficient steps (Nocedal and Wright, 2006). The general form of such an update at iteration m is:

$$\big(\phi, \theta, \sigma^2\big)^{(m+1)} = \big(\phi, \theta, \sigma^2\big)^{(m)} - \big[H\big(\phi, \theta, \sigma^2\big)\big]^{-1}\nabla\ell\big(\phi, \theta, \sigma^2\big),$$

where, H represents an approximation of the Hessian matrix (Boyd and Vandenberghe, 2004b) of the log-likelihood function ℓ. The Hessian matrix is a square matrix formed by the second-order partial derivatives of a function, which characterizes its local curvature. Utilizing this curvature information allows the algorithm to converge more rapidly than methods that only use first-order gradient information.

Once the optimal parameters have been identified, the model can be utilized for forecasting. The one-step-ahead forecast, $\hat{y}_{T+1}$, is calculated based on the past values of the series and the past forecast errors:

$$\hat{y}_{T+1} = \sum_{j=1}^{p}\phi_j\, y^{(d)}_{T+1-j} + \sum_{k=1}^{q}\theta_k\,\hat{\varepsilon}_{T+1-k} \quad \text{(after reversing the differencing).}$$

These forecasts can then be used as inputs for various decision-making models.

1.2 SARIMAX $(p, d, q) \times (P, D, Q)_m$ with exogenous regressors

The Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX) model is a powerful extension of the ARIMA framework, designed to handle more complex time series data. It enhances the ARIMA model by incorporating two crucial features: seasonality and the influence of external explanatory variables Box et al. (2015).

The structural equation of the SARIMAX model is written as:

$$\Phi(B)\,\Phi_s\big(B^m\big)\,(1-B)^d\,(1-B^m)^D\,y_t \;=\; \Theta(B)\,\Theta_s\big(B^m\big)\,\varepsilon_t \;+\; \beta^\mathsf{T}\,X_t,$$

where the components extend the ARIMA model as follows:

- $\Phi(B)$ and $\Theta(B)$ are the non-seasonal autoregressive (AR) and moving average (MA) polynomials of orders p and q, respectively, identical to those in the ARIMA model.

- $\Phi_s(B^m) = 1 - \Phi_1\,B^m - \cdots - \Phi_P\,B^{mP}$ is the seasonal AR polynomial of order P. It models the dependency of the current value on observations from previous seasons.

- $\Theta_s(B^m) = 1 + \Theta_1\,B^m - \cdots + \Theta_Q\,B^{mQ}$ is the seasonal MA polynomial of order Q, which models the dependency on forecast errors from previous seasons.

- m is the seasonal period (e.g., $m = 12$ for monthly data, $m = 4$ for quarterly data).

- D is the order of seasonal differencing, $(1 - B^m)^D$, used to remove seasonal trends.

- $X_t \in \mathbb{R}^K$ is a vector of K exogenous (external) regressors at time t. These can include variables like marketing expenditures, promotional activities, holidays, or macroeconomic indicators.

- $\beta \in \mathbb{R}^K$ is the vector of coefficients corresponding to the exogenous regressors, quantifying their impact on the time series y_t.

The complete set of parameters to be estimated is:

$$\Psi = \{\phi_1, \dots, \phi_p;\ \theta_1, \dots, \theta_q;\ \Phi_1, \dots, \Phi_P;\ \Theta_1, \dots, \Theta_Q;\ \beta;\ \sigma^2\},$$

where the orders (p, d, q) and $(P, D, Q)_m$ are typically determined beforehand using diagnostic tools like the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF).

Similar to ARIMA, the parameters Ψ are estimated by maximizing the log-likelihood function, which assumes the same form but now depends on the more complex SARIMAX structure:

$$\ell(\Psi) = -\frac{T}{2} \ln\bigl(2\pi\, \sigma^2\bigr) - \frac{1}{2\,\sigma^2} \sum_{t=1}^{T} \Bigl(\varepsilon_t(\Psi)\Bigr)^2,$$

where the residuals $\varepsilon_t(\Psi)$ are derived from the main structural equation. Due to the increased number of parameters, this optimization is almost always performed iteratively. A common choice for this task is the Limited-Memory BFGS (L-BFGS) algorithm, a quasi-Newton method well-suited for high-dimensional problems Nocedal and Wright (2006). The update step is given by:

$$\Psi^{(k+1)} = \Psi^{(k)} - H^{-1}\bigl(\Psi^{(k)}\bigr)\, \nabla\ell\bigl(\Psi^{(k)}\bigr),$$

where H^{-1} is an efficient approximation of the inverse Hessian matrix.

Once the parameters have converged, the model can generate forecasts. The h-step-ahead forecast $\hat{y}_{T+h}$ incorporates all components of the model: non-seasonal, seasonal, and exogenous. While the complete recursive formula is complex, its structure includes terms for each polynomial and the future values of the exogenous variables:

$$\hat{y}_{T+h} = \text{f(past } y,\ \text{past } \varepsilon,\ \text{seasonal lags},\ \text{parameters } \Psi,\ \text{and } X_{T+h}).$$

Including seasonality and external factors makes SARIMAX a significantly more powerful and flexible forecasting tool than a standard ARIMA, especially for modelling real-world business processes like retail sales, which are often subject to cyclical patterns and external influences.

1.3 Methodology for strategic portfolio optimization

The foundation of this study lies in creating a multi-phase methodology for identifying the most effective strategic distribution of resources among various product categories. This method employs an advanced iterative optimization technique known as the Discrete Functional Particle Method (DFPM). However, it modifies and incorporates it within a broader framework that guarantees both the consistency and real-world applicability of the outcomes. This section elaborates on the mathematical formulation of the issue and the particular execution of the solution approach.

1.4 Problem formulation for product assortment optimization

We begin by formulating the optimization problem based on the principles of modern portfolio theory but adapted for the specific context of retail management. We consider a category consisting of k products. Let $d_i = (d_{1i}, \ldots, d_{ki})^T$ be the k-dimensional vector of observed sales quantities for these products at time $i = 1, \ldots, N$. We assume the second moment of d_i is finite.

Let $w = (w_1, \ldots, w_k)^T$ be the vector of weights for each product in the category, where w_j denotes the share of jth product. We define $\mathbf{1} \in \mathbb{R}^k$ as the vector of ones.

A key distinction of our approach is how we define "risk" and "return." Unlike in financial markets, where risk is typically the volatility of asset prices, we define it in operational terms.

- **Return Vector μ:** The vector $\mu \in \mathbb{R}^k$ represents the expected return for each category, which we define as the historical average revenue.
- **Risk Matrix Σ:** The matrix $\Sigma \in \mathbb{R}^{k \times k}$ represents the operational risk. It is defined as the covariance matrix of the **Inventory Efficiency Ratio** ($E_i(t)$). This ratio, calculated for each category i at each time period t as:

$$E_i(t) = \frac{\text{Value of Leftovers}_i(t)}{\text{Revenue}_i(t)}.$$

A high value indicates operational inefficiency. Therefore, Σ models the volatility and interplay of these operational inefficiencies across categories.

With these definitions, we formulate the following mean-variance-style optimization problem for the assortment:

$$\min_{w} \; w^T \Sigma w \quad \text{s.t.} \quad w^T \mathbf{1} = 1, \quad w^T \mu = q, \tag{1}$$

where q is the target expected revenue for the portfolio. The objective function $w^T \Sigma w$ now represents the variance (risk) of the entire portfolio's operational efficiency. The constraint $w^T \mathbf{1} = 1$ ensures that the weights are fully allocated, and $w^T \mu = q$ enforces that the expected aggregate revenue meets a specified target level.

If Σ is positive definite, the solution to (1) is unique and can be written in closed form as:

$$w = \frac{C - q B}{A C - B^2} \Sigma^{-1} \mathbf{1} \; + \; \frac{q A - B}{A C - B^2} \Sigma^{-1} \mu, \tag{2}$$

where

$$A = \mathbf{1}^T \Sigma^{-1} \mathbf{1}, \quad B = \mathbf{1}^T \Sigma^{-1} \mu, \quad C = \mu^T \Sigma^{-1} \mu.$$

If Σ is singular (for instance, when $k > N$ or when the efficiency ratios of some products are perfectly collinear), then (1) admits infinitely many solutions. Following the approach of Pappas et al. Pappas et al. (2010), one can select a unique solution by replacing the standard inverse Σ^{-1} with the Moore–Penrose pseudoinverse Σ^{+}. This yields the minimal Euclidean norm allocation:

$$w = \frac{C - q B}{A C - B^2} \Sigma^{+} \mathbf{1} \; + \; \frac{q A - B}{A C - B^2} \Sigma^{+} \mu, \tag{3}$$

with the scalars A, B, and C recalculated using Σ^{+}.

In real-world applications, both μ and Σ must be estimated from historical observations. The standard sample estimators are used:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} p_i, \qquad \hat{\Sigma} = \frac{1}{N-1} \sum_{i=1}^{N} (e_i - \bar{e}) (e_i - \bar{e})^T, \tag{4}$$

where p_i is the vector of revenues at time i, and e_i is the vector of inventory efficiency ratios at time i. When the number of products k exceeds the number of observations N, the sample covariance matrix $\hat{\Sigma}$ will be singular, necessitating the use of methods robust to this condition, such as the iterative DFPM method, which is the focus of this work.

1.5 The Discrete Functional Particle Method (DFPM)

To solve the constrained quadratic optimization problem formulated in the previous section, we employ the Discrete Functional Particle Method (DFPM), described by Gulliksson and Mazur (2020). This iterative method is particularly well-suited for problems where the risk matrix Σ may be singular or ill-conditioned.

The core idea of DFPM is to find the minimum of a convex function $V(u)$ by treating it as a potential field for a physical system. The minimum of the function corresponds to the stationary point of a damped dynamical system, described by the second-order differential equation Bégout et al. (2015):

$$\ddot{u}(t) + \eta\, \dot{u}(t) \;=\; -\nabla V\big(u(t)\big), \qquad \eta > 0, \tag{5}$$

where $\dot{u}$ and $\ddot{u}$ are the first and second time derivatives of the position vector u, and η is a damping coefficient.

Application to the Constrained Problem

Our main optimization problem is constrained:

$$\min_{w \in \mathbb{R}^k} \; \tfrac{1}{2}\, w^\top \Sigma\, w \qquad \text{s.t. } B\, w = c, \tag{6}$$

where Σ is the operational risk matrix, and the constraints are given by:

$$B \;=\; \begin{pmatrix} \mathbf{1}^\top \\ \mu^\top \end{pmatrix} \in \mathbb{R}^{2\times k}, \qquad c = \begin{pmatrix} 1 \\ \mu_{target} \end{pmatrix} \in \mathbb{R}^2.$$

To apply DFPM, we first eliminate the linear constraints by parameterizing the solution vector w. Any feasible w that satisfies $Bw = c$ can be written as:

$$w \;=\; Z\,u \;+\; g, \qquad u \in \mathbb{R}^{\,k-2}, \tag{7}$$

where:

- $g = B^T(BB^T)^{-1}c$ is a particular solution to the constraint system.
- $Z \in \mathbb{R}^{k\times(k-2)}$ is a matrix whose columns form an orthonormal basis for the null space (kernel) of B, meaning $BZ = 0$.
- u is a new vector of variables in a lower-dimensional, unconstrained space.

Substituting this parameterization into the objective function yields an equivalent unconstrained problem in terms of u:

$$\min_{u\in\mathbb{R}^{\,k-2}} \;\Phi(u) \;=\; \tfrac{1}{2}\,(Zu+g)^{\mathsf{T}}\,\Sigma\,(Zu+g).$$

Expanding this expression and ignoring constant terms (which do not affect the position of the minimum), we obtain the function to be minimized:

$$\min_{u\in\mathbb{R}^{\,k-2}} \;\tfrac{1}{2}\,u^{\mathsf{T}}(Z^{\mathsf{T}}\Sigma Z)\,u \;+\; (Z^{\mathsf{T}}\Sigma g)^{\mathsf{T}}\,u. \tag{8}$$

Let us define $M = Z^{\mathsf{T}}\Sigma Z$ and $d = Z^{\mathsf{T}}\Sigma g$. The problem then simplifies to minimizing the potential $V(u)$:

$$V(u) = \tfrac{1}{2}\,u^{\mathsf{T}}M\,u \;+\; d^{\mathsf{T}}\,u. \tag{9}$$

The gradient of this potential is $\nabla V(u) = Mu + d$. The corresponding damped dynamical system is:

$$\ddot{u}(t) + \eta\,\dot{u}(t) \;=\; -(M\,u(t) + d). \tag{10}$$

To solve this numerically, we introduce the velocity $v(t) = \dot{u}(t)$ and apply the iterative symplectic Euler scheme with a time step Δt:

$$v_{k+1} = (I - \Delta t\, \eta)\, v_k \; - \; \Delta t\, (M u_k + d), \tag{11}$$

$$u_{k+1} = u_k + \Delta t\, v_{k+1}. \tag{12}$$

This system of equations is precisely what is implemented in the `dfpm` function in the accompanying code, where d corresponds to the variable `d_vec`. The process is initialized (typically with $u_0 = 0, v_0 = 0$) and iterated until convergence. Once the optimal u^* is found, the final weight vector is reconstructed as:

$$w^*_{\text{optimal}} = Z\, u^* + g.$$

Selection of Δt and η

The efficiency of the DFPM solver critically depends on the choice of the step size Δt and the damping coefficient η. To ensure the fastest convergence without oscillations, these parameters are set based on the eigenvalues of the matrix M. Let the smallest positive and largest eigenvalues of M be $\lambda_{\min}$ and $\lambda_{\max}$, respectively. The optimal parameters are given by:

$$\Delta t = \frac{2}{\sqrt{\lambda_{\min}} \; + \; \sqrt{\lambda_{\max}}}, \qquad \eta = 2\, \frac{\sqrt{\lambda_{\min}\, \lambda_{\max}}}{\sqrt{\lambda_{\min}} + \sqrt{\lambda_{\max}}}. \tag{13}$$

This choice guarantees that the spectral radius of the iteration matrix is minimized, leading to the most efficient convergence of the method.

1.6 A Hybrid methodology for strategic and tactical assortment planning

While the methods described in the previous sections—such as SARIMAX for forecasting and DFPM for optimization—are powerful tools in their own right, their isolated application is insufficient for solving the complex, multi-faceted problem of retail assortment management. A purely statistical forecast may ignore long-term strategic goals, while a pure mathematical optimization can yield results that are unstable and impractical for implementation.

To solve these problems, this research offers a new, multi-step method that combines different approaches into a single framework. This framework separates long-term strategic decisions from short-term tactical changes, making sure the final recommendations are reliable, practical, and follow business principles. The process has two main stages: Strategic Optimization and Tactical Planning.

Stage 1: Strategic optimization with a hybrid approach

The goal of the strategic stage is to determine a single, stable vector of foundational weights, $W_{\text{strategic}}$, that reflects a balanced view of historical performance and optimized risk. The output of the DFPM solver might lead to aggressive and volatile solutions, where categories with significant sales history might be assigned near-zero weights. To mitigate this, we use a Hybrid Strategy.

This Strategy blends the 'pure' mathematical optimum with a baseline portfolio that represents established business experience.

1. **Base Portfolio (W_{base}):** This portfolio's weights are determined by the historical revenue share of each category over the analysis window (T_{hist}, typically 24 months). It represents the 'as-is' strategy, acknowledging historically successful categories.

$$W_{\text{base},i} = \frac{\sum_{t=1}^{T_{hist}} P_{it}}{\sum_{j=1}^{k} \sum_{t=1}^{T_{hist}} P_{jt}},$$

   where P_{it} is the revenue of category i at time t.

2. **Optimal Portfolio (W_{optimal}):** This is the portfolio calculated by the DFPM, as described in Section 1.5. It is the solution to the problem of minimizing operational risk for a given target return, based on historical data.

3. **Strategic Hybrid Portfolio ($W_{\text{strategic}}$):** The final strategic weights are a weighted average of the base and optimal portfolios, controlled by a blending factor $\alpha \in [0, 1]$, which acts as a "confidence" parameter:

$$W_{\text{strategic}} = \alpha \cdot W_{\text{base}} + (1 - \alpha) \cdot W_{\text{optimal}}. \tag{14}$$

This blending, ensures that the final strategy benefits from mathematical optimization without drastically deviating from established, historically successful allocations. This $W_{\text{strategic}}$ vector serves as the foundational input for the next stage.

Stage 2: Tactical planning for future periods

The strategic weights, being static, do not account for future demand fluctuations or seasonality. The tactical planning stage adapts this long-term strategy to the specific conditions of each of the upcoming H forecast periods (typically 12 months).

Demand forecasting with safeguards

First, a demand forecast for each category, Qf_i, is generated for the next H months using the SARIMAX model, as detailed in Section 1.2. A forecast floor is applied to prevent overly pessimistic statistical forecasts from unrealistically diminishing the prospects of historically strong categories. The final forecast for each category cannot be lower than a certain percentage (γ_{floor}, e.g., 50%) of its average sales over the last 12 months.

Seasonal adjustment

To account for predictable cyclical demand, a historical seasonal index, $S_{i,m}$, is calculated for each category i and each month $m \in \{1, \ldots, 12\}$. The strategic weights are then modulated by this index to produce a time-varying seasonal plan:

$$W_{\text{seasonal},i}(t) = W_{\text{strategic},i} \cdot S_{i,m(t)},$$

where $m(t)$ is the month corresponding to time period t. The resulting weights are then re-normalized to sum to 1 for each period.

Application of business constraints

Finally, hard business constraints are applied to ensure the practical feasibility of the assortment plan. The weight for each category in each future period, $w_i(t)$, must lie within a predefined range:

$$w_{\min} \leq w_i(t) \leq w_{\max}.$$

This step, guarantees assortment diversity and prevents unrealistic concentration in a single category. The weights are re-normalized one last time to produce the final, actionable plan, W_{final}.

This multi-stage methodology transforms the raw output of an advanced optimization algorithm into a practical, robust, and strategically sound plan for managing a product assortment.

2 Practical application and software implementation

2.1 Dataset

Dataset description

The dataset used in this thesis was provided by a Ukrainian retail company and contains historical sales and inventory data. The data spans a period from January 2019 to June 2023, inclusive, and covers a wide range of products.

The dataset consists of three primary files, each structured in a 'wide' format where rows represent individual products (SKUs) and columns represent monthly data points. All three files share an identical structure, containing 4202 rows (SKUs) and 55 columns.

The three files are:

1. `Sales_UAH.xlsx`: This file contains the monthly turnover data for each product.
   - **Rows:** 4202 SKUs, where each row corresponds to a specific product.
   - **Columns:** 55 columns. The first column contains the product name (SKU identifier), and the subsequent 54 columns represent each month from January 2019 to June 2023. The values in these columns are the total turnover in Ukrainian Hryvnia (UAH) for the corresponding product and month.
2. `Sales_Qty.xlsx`: This file contains the monthly sales volume data for each product.
   - **Rows:** 4202 SKUs, identical to the turnover file.
   - **Columns:** 55 columns, with the same structure. The values represent the total number of units sold for the corresponding product and month.
3. `Leftovers_Qty.xlsx`: This file contains data on the end-of-month inventory levels for each product.

- **Rows:** 4202 SKUs, identical to the other files.
- **Columns:** 55 columns, with the same structure. The values represent the quantity of leftover units for the corresponding product at the end of each month.

4. `Antistress_Metrics.xlsx`: This file provides external factors (exogenous variables) that are used to improve the accuracy of demand forecasting. It contains aggregated metrics for the product category under analysis. The key metrics used in this work are:

   - `Stores_With_Stock`: The number of stores where at least one item from the category was in stock during the month.
   - `Cheques_Count`: The total number of transactions (checks) that included at least one item from the category during the month.

It is important to note that the raw dataset contains certain records for which all monthly data points are zero or missing. Therefore, a pre-processing stage is necessary to clean and prepare the data for analysis.

Data pre-processing

The raw dataset mentioned earlier is organized in a 'wide' format, where each row represents a product and each column corresponds to a month. While this format is useful for data storage, it does not work well for time series analysis, which requires a 'long' format. In this format, each row accounts for a single observation related to a specific product at a particular time. To address this, a pre-processing stage was conducted to transform and consolidate the data from three source files—`Sales_UAH.xlsx`, `Sales_Qty.xlsx`, and `Leftovers_Qty.xlsx`—into a unified dataset that is ready for analysis.

The pre-processing pipeline consists of two main steps: data transformation (melting) and merging.

Step 1: Data transformation from wide to long format

A dedicated function, `read_and_melt`, was developed to handle the transformation for each of the three files. This function performs the following operations:

1. **Reading the data:** The function reads an Excel file into a pandas DataFrame. The first column, containing product identifiers, is programmatically renamed to `'Category'`.

2. **Identifying date columns:** A key challenge is that column headers for dates may have inconsistent formatting. To handle this, a regular expression (`r"\d{4}-\d{2}"`) is used to robustly identify all columns that represent a month in the `YYYY-MM` format.

3. **Melting the DataFrame:** The core transformation is performed using the pandas `melt` function. This operation unpivots the DataFrame from a wide format to a long format. The `'Category'` column is kept as an identifier variable, while all the date columns are converted from columns into rows. This results in a new table with three essential columns: `'Category'`, `'Date'`, and a value column (e.g., `'Sales_Qty'`).

This process is applied independently to each of the three source files, resulting in three separate DataFrames (`df_sales_qty`, `df_sales_uah`, `df_leftovers`), all sharing a common long-format structure.

Step 2: Merging and finalization

After transforming each file, the three DataFrames are merged into a single, unified dataset.

1. **Merging:** The merge operation is performed sequentially using the pandas `merge` function. The `'Category'` and `'Date'` columns serve as the composite key for joining the tables. An `outer` join strategy is employed to ensure that no data is lost; if a record for a specific product and date exists in one file but not another, it is still included in the final dataset, with missing values represented as `NaN`.

2. **Finalization:** The columns of the merged DataFrame are reordered for clarity. Finally, the entire dataset is sorted by `'Category'` and then by `'Date'` to create a chronologically ordered time series for each product.

The resulting DataFrame contains five columns: `'Category'`, `'Date'`, `'Sales_Qty'`, `'Sales_UAH'`, and `'Leftovers_Qty'`. This clean, long-format dataset is then saved to a new file, `Merged_Table.xlsx`, and serves as the primary input for all subsequent analysis, forecasting, and optimization tasks described in this thesis.

Step 3: Data selection for case study

Given the large number of SKUs in the full dataset, a specific product category was selected for a more focused case study. The "Antistress Toys" category was chosen for this purpose. This subset initially contained data for 11 distinct products. Upon preliminary analysis, it was discovered that two of these products had no sales or inventory data (i.e., all corresponding values were zero) across the entire historical period. These two SKUs were subsequently removed from the dataset.

The final, clean dataset used for the main analysis in this work therefore consists of 9 products from the "Antistress Toys" category. This filtered dataset was saved to a new file, `data_antistress.xlsx`, to streamline the subsequent modeling and optimization stages.

2.2 Program implementation

To practically implement the multi-stage methodology for assortment optimization described in the previous chapters, a program was developed using the Python programming language. The implementation relies on a set of powerful libraries for data analysis, numerical computation, and visualization, most notably `pandas` for data manipulation, `NumPy` and `SciPy` for mathematical operations, `pmdarima` and `statsmodels` for time series forecasting, and `Matplotlib` for plotting the results.

The entire logic is encapsulated within a single script, structured into functional blocks responsible for data loading, pre-processing, optimization, forecasting, and reporting.

Global configuration and main components

The script's behavior is controlled by a global configuration dictionary named `CFG`. This allows for flexible experimentation by adjusting key parameters without altering the main codebase. The main components of the implementation are a series of functions that logically follow the stages of the developed methodology.

Data loading

The initial stage involves preparing the raw data for analysis. This is handled by a set of dedicated functions:

- `load_data(path)`: This function reads the primary dataset (`data_antistress.xlsx`) into a pandas DataFrame, converts the date column to the correct `datetime` format, and sorts the data chronologically.

- `load_metrics(path)`: This function loads the file with exogenous variables (`Antistress_Metrics_Wide.xlsx`), sets the date as the index, and prepares the data for use in the forecasting model.

- `prepare_dataframes(df)`: This is a crucial function that takes the "long" format data and transforms it into "wide" format pivot tables, which are essential for time series analysis and risk calculation. It generates several key DataFrames:

  - `Q`: A matrix of sales quantities, where rows are dates and columns are product categories.
  - `P`: A matrix of sales revenue (turnover).
  - `L_UAH`: A matrix representing the value of leftovers in UAH.
  - `Efficiency_Ratio`: A matrix calculated as `L_UAH / P`, which serves as the primary input for the operational risk calculation. Outliers in this ratio are clipped at the 95th percentile to ensure model stability.
  - `Turnover_Rate`: A matrix calculated as

    $$\texttt{Turnover\_Rate}_{t,i} = \frac{P_{t,i}}{L_{\text{UAH},t,i}},$$

    representing the ratio of revenue to leftover value for each category i at time t.

2.3 Strategic optimization stage

This is the core of the model, where the long-term strategic weights are determined. This process is executed within the `main` function.

1. **Historical window selection:** The model first selects a historical period for analysis, defined by the `WINDOW_SIGMA` parameter (typically 24 months).

2. **Base portfolio calculation (W_{base}):** The weights of the baseline portfolio are calculated based on the total revenue share of each category within this historical window. This represents the "as-is" strategy.

3. **Risk matrix calculation (Σ_{hist}):** The `calculate_sigma` function is called. Based on the `OBJECTIVE` setting (e.g., `'INVENTORY_EFFICIENCY'`), it computes the covariance matrix of the corresponding metric (the 'Efficiency_Ratio' DataFrame) over the historical window. This matrix represents the operational risk.

4. **Optimal portfolio calculation ($W_{optimal}$):**

   - First, the expected return vector (μ_{hist}) and the target return (μ_{target}) are determined.
   - Then, the constraint matrices B and c are formed using `build_constraints_for_target`.
   - Finally, the `dfpm` function is executed. It takes the risk matrix Σ_{hist} and the constraints as input and returns the vector of mathematically optimal weights, $W_{optimal}$, that minimizes the risk for the given return target. The implementation precisely follows the iterative logic of the damped dynamical system described in the theoretical section.

5. **Hybrid strategy formulation ($W_{strategic}$):** The final strategic weights are calculated by blending the base and optimal portfolios using the `BLENDING_FACTOR` (α), as per the formula $W_{strategic} = \alpha \cdot W_{base} + (1 - \alpha) \cdot W_{optimal}$. This provides a balanced and robust strategic recommendation.

2.4 Tactical planning stage

This stage adapts the static strategic weights to the dynamic conditions of the upcoming 12 months.

1. **Demand forecasting (`_forecast_one`)**

   - For each product category, a separate time series forecast is generated for the next H months.
   - The implementation uses the `pmdarima.auto_arima` function, which automatically selects the best SARIMAX model parameters based on the data. It also incorporates the exogenous variables specified in `EXOG_COLS`.
   - A "forecast floor" is applied: the predicted sales quantity for any category cannot fall below a certain percentage (`FORECAST_FLOOR_FACTOR`) of its average sales from the previous year. This acts as a business logic safeguard against overly pessimistic forecasts.

2. **Seasonal adjustment:** A historical seasonal index is calculated by averaging sales for each month of the year. The static `W_strategic` vector is then multiplied by the corresponding monthly index to produce a time-varying plan, `W_final_seasonal`.

3. **Applying business constraints:** The `apply_business_constraints` function ensures that the weights in the final plan adhere to the predefined `MIN_WEIGHT_PER_CAT` and `MAX_WEIGHT_PER_CAT` limits, guaranteeing a diversified and realistic assortment.

2.5 Reporting and visualization

The final block of the script is dedicated to presenting the results in an interpretable format.

- A detailed summary table is printed to the console, comparing various historical and forecasted metrics ("Was-Became" analysis).
- An aggregate performance block shows the overall change in key business indicators: total turnover, value of leftovers, and inventory turnover.

- A risk comparison block quantifies the effectiveness of the optimization by comparing the risk of the base, optimal, and final strategic portfolios.
- A stacked bar chart is generated using `Matplotlib`, visually representing the historical and forecasted portfolio composition over time.

This comprehensive implementation transforms the theoretical methodology into a practical and configurable tool for strategic decision-making in retail.

2.6 Analysis of results

This section presents and analyzes the results obtained from applying the developed hybrid methodology to the retail dataset. The analysis is structured to evaluate the model's effectiveness from three perspectives: the strategic allocation of weights, the impact on aggregate operational metrics, and the quantitative change in operational risk.

Strategic weight allocation: a comparison of approaches

The first block of results is a summary table that compares the weights derived from different strategies. This allows for a detailed examination of how the final recommendation is formed.

Table 2.1: Comparison of Strategic Weight Allocation (in %)

Category ID	Hist. Last 12m	Optimal (DFPM)	Strategic (Hybrid)	Final Avg.
1.1	4.1%	2.0%	2.2%	2.3%
1.2	56.4%	43.1%	42.7%	43.0%
1.3	3.9%	6.8%	6.6%	6.7%
1.4	0.4%	0.0%	0.2%	0.2%
1.5	0.4%	0.9%	1.2%	1.3%
1.6	10.6%	10.2%	10.6%	11.0%
1.7	3.6%	3.6%	3.6%	3.9%
1.8	9.8%	16.8%	16.4%	15.7%
1.9	10.9%	16.7%	16.6%	16.1%

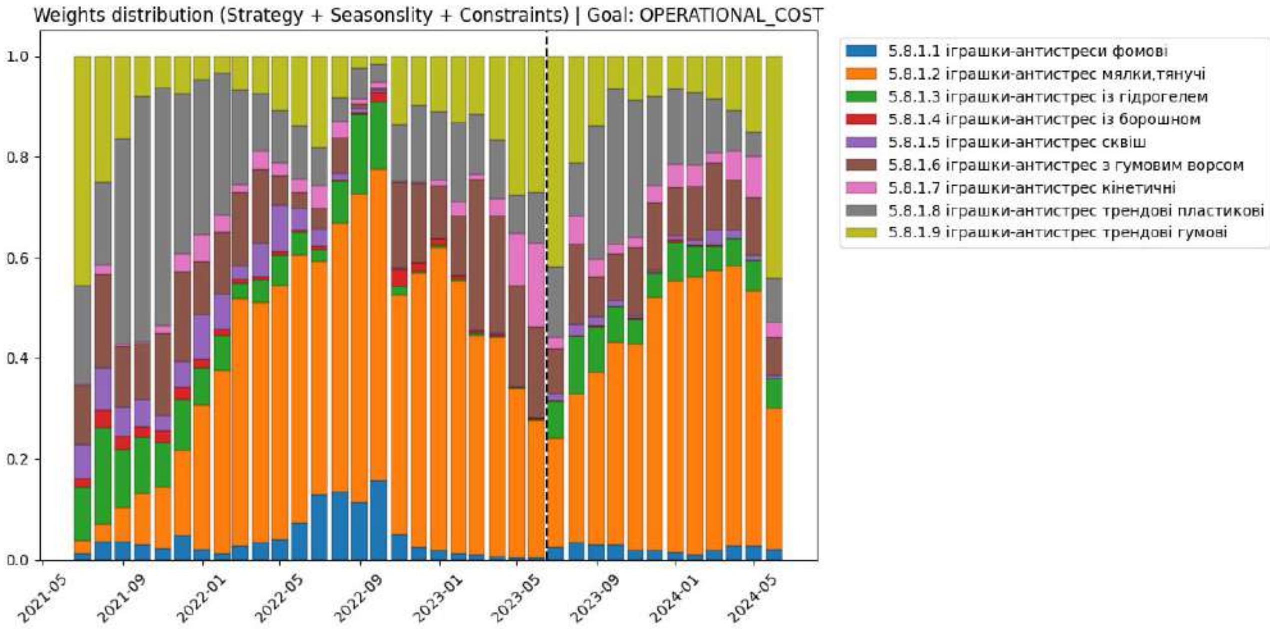

Figure 2.1: Historical monthly weights vs Optimized and forecasted monthly weights distribution for 12 month

- `w_Hist_Last12`: This column represents the actual weight distribution based on revenue from the last 12 months. It serves as our baseline, showing a clear market leader in category `1.2` (56.4%) and several other significant contributors.

- `w_Optimal`: This is the "pure" mathematical solution generated by the DFPM algorithm. We can observe its aggressive nature: it completely eliminates category `1.4` and significantly increases the weight of categories it deems most efficient (e.g., `1.8`, `1.9`). While mathematically optimal, such a strategy is often too radical for practical implementation.

- `w_Strategic`: This column presents the result of our hybrid strategy, which blends the historical baseline with the DFPM optimum using a 20/80 ratio (20% of historical /80% of DFPM). The effect is immediately apparent: the zeroed-out category `1.4` is reinstated with a small but non-zero weight (0.15%), and the weights of other categories are moderated, resulting in a more balanced and robust strategic plan.

- `w_Final_Avg`: This shows the average weights of the final tactical plan after applying seasonal adjustments with SARIMAX and business constraints (min/max limits). This is the most practical and realistic representation of the recommended assortment structure over the next 12 months.

2.7 Aggregate performance analysis: the business impact

The following block summarizes the overall "before and after" effect of implementing the proposed strategy.

Metric	Before	After
Total turnover (UAH)	54,784,003	44,614,922
Avg. monthly leftovers (UAH)	8,085,476	2,341,732
Overall turnover rate	6.78	19.05

Table 2.2: Performance metrics before and after implementing the strategy

This output highlights the main value proposition of the model.

- **Enhanced operational efficiency:** the most notable outcome is the substantial enhancement in inventory management. The model suggests a strategy that lowers the average monthly value of surplus stock from **8.1 million UAH to 2.3 million UAH**. This creates an additional 6 million UAH in available working capital.

- **Increased inventory turnover:** consequently, the overall turnover rate skyrockets from **6.78 to 19.05**, an almost threefold increase. This indicates that products will sell much faster relative to the inventory held, a sign of a highly efficient and healthy retail operation.

- **Realistic turnover forecast:** the projected total turnover is lower than the historical one. As discussed previously, this is not a model failure but rather a realistic forecast generated by the SARIMAX component, which likely detected a general downward trend in the market for this category. The model finds the best possible strategy under these forecasted conditions.

2.8 Operational cost analysis

This final block provides a quantitative assessment of the model's primary objective: managing operational costs, defined as the volatility of the inventory efficiency ratio.

Metric	**Value**
Base portfolio cost (last 12m)	0.73
Final hybrid strategy cost	0.55
Change	**-24.9%**

Table 2.3: Operational cost comparison (last 12 months)

The analysis of this table reveals a crucial insight.

The Key Finding: The main finding shows that over the past 12 months, a newly developed strategy, based on 24 months of historical data, led to a significant **24.91% reduction in operational costs** compared to the baseline. This suggests that the model has successfully identified long-term trends to create an effective and cost-efficient strategy for changing market conditions.

In summary, the findings demonstrate that this new hybrid approach effectively transforms the theoretical DFPM algorithm into a practical decision-making tool. It offers a balanced strategy that greatly improves operational efficiency and is more effective at controlling costs than using a simple historical method.

Conclusion

This thesis set out to develop and validate a hybrid framework for retail assortment planning that couples SARIMAX-based demand forecasting with the Discrete Functional Particle Method (DFPM) for optimisation under uncertainty. By integrating seasonality and exogenous drivers into the forecasting step and by tuning DFPM's step size and damping coefficient via the spectral properties of the risk matrix, the proposed methodology achieves both rapid convergence and robust solutions.

Applied to a real "Antistress Toys" dataset from a Ukrainian retailer, the framework generated a strategic portfolio that reduced operational risk by nearly 25% compared to the historical baseline while simultaneously more than tripling inventory turnover. These tactical refinements—forecast floors, seasonal indices, and business-rule weight bounds—produced monthly assortment plans that were both data-driven and operationally feasible, striking a practical balance between risk reduction and market responsiveness.

Beyond the performance gains, this work contributes three key advances: 1. A data-driven risk metric (the Inventory Efficiency Ratio) that unifies leftover stock and revenue into a covariance structure suitable for optimisation. 2. Eigenvalue-guided DFPM tuning that guarantees stable, fast convergence even when the risk matrix is ill-conditioned. 3. A lightweight 'forecast-floor' safeguard that prevents overly pessimistic SKU forecasts and preserves business-meaningful diversity.

Looking forward, there are several promising extensions to this work. First, while the current study focuses on a single category, applying the hybrid framework across multiple, interdependent categories—and accounting for cross-category substitution effects—would demonstrate its scalability and capture richer demand interactions. Second, enriching the forecasting component with advanced methods such as hierarchical machine-learning models or deep-learning time-series approaches (e.g., LSTM) could boost predictive accuracy. Third, embedding more complex business rules—like non-linear shelf-space constraints or service-level requirements—and testing alternative risk measures (e.g., Conditional Value at Risk(Rockafellar and Uryasev, 2000) or maximum drawdown) would enhance the framework's flexibility. Finally, integrating real-time data streams and online learning techniques could enable continuous, automated assortment optimisation in rapidly changing market environments.

In summary, this thesis demonstrates that tightly coupling advanced forecasting and optimisation methods yields actionable, measurable improvements in assortment planning. The hybrid framework offers practitioners a flexible, reproducible decision-support tool, while opening avenues for future extensions in multi-category and non-linear retail settings.

References

Arrow, K. J., Harris, T., and Marschak, J. (1951). Optimal inventory policy. *Econometrica*, 19(3):250–272.

Avdieienko, I. (2025). Iterative methods for solving optimization problems using the discrete functional particle method.

Bégout, P., Bolte, J., and Jendoubi, M. A. (2015). On damped second-order gradient systems. *Journal of Differential Equations*, 259(7):3115–3143.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.

Boyd, S. and Vandenberghe, L. (2004a). *Convex Optimization*. Cambridge University Press.

Boyd, S. and Vandenberghe, L. (2004b). *Convex Optimization*. Cambridge University Press.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gulliksson, M. and Mazur, S. (2020). An iterative approach to ill-conditioned optimal portfolio selection. *Computational Economics*, 56(4):773–794.

Hamilton, J. D. (1994). Time series analysis.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1):77–91.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer Science & Business Media, second edition.

Pappas, D., Kiriakopoulos, K., and Kaimakamis, G. (2010). Optimal portfolio selection with singular covariance matrix. *International Mathematical Forum*, 5(47):2305–2318.

Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value–at–risk. *Journal of Risk*, 2:21–41.

Appendix A

ІТЕРАЦІЙНІ МЕТОДИ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ З ВИКОРИСТАННЯМ МЕТОДУ ДИСКРЕТНИХ ФУНКЦІОНАЛЬНИХ ЧАСТИНОК

І.М. АВДЄЄНКО

Вступ. Метод дискретних функціональних частинок (DFPM, Discrete Functional Particle Method) є ефективним ітераційним підходом для розв'язання задач необумовленої оптимізації. Він базується на аналізі затухаючих динамічних систем другого порядку, що дозволяє уникати обмежень, пов'язаних із виродженістю задач або сингулярністю коваріаційної матриці.

DFPM вже показав свою ефективність у фінансовому моделюванні, зокрема в задачах оптимального вибору портфеля активів у випадку сингулярної коваріаційної матриці Σ, що часто виникає при малій кількості спостережень або високій колінеарності активів.

У даній роботі цей метод буде використаний для розв'язання задачі прогнозування оптимального товарного набору для певної категорії в мережі магазинів. Така задача зводиться до задачі мінімізації функціоналу, що відображає сукупний прогнозований ризик (наприклад, відсутність продажів або втрати прибутку) з урахуванням ряду обмежень (сезонність, попит, бюджет, наявність товарів тощо).

Перевага DFPM полягає в здатності ефективно працювати з недообумовленими моделями, які включають високу розмірність, пропущені дані та корельовані фактори. Це дозволяє застосовувати DFPM у практичних системах підтримки рішень у сфері торгівлі та логістики, де необхідно формувати найкращий набір товарів, адаптований до ринкових умов.

Постановка задачі для товарного набору для певної категорії. Розглянемо класичну задачу вибору портфеля з k категорій з мінімізацією ризику:

$$\min_{\mathbf{w}} \mathbf{w}^T \Sigma \mathbf{w} \quad \text{за умов} \quad \mathbf{w}^T \mathbf{1} = 1, \quad \mathbf{w}^T \mu = q, \tag{1}$$

де $\mu \in \mathbb{R}^k$ — вектор середніх прибутковостей, $\Sigma \in \mathbb{R}^{k \times k}$ — коваріаційна матриця, $\mathbf{w} \in \mathbb{R}^k$ — вектор ваг кожної категорії, q — задана очікувана дохідність.

У випадку сингулярної Σ, задача (1) стає виродженою, і класичні методи не дають єдиного розв'язку. У таких умовах застосовують ітераційні регуляризаційні підходи.

Метод DFPM як підхід до регуляризації. DFPM розв'язує задачу:

$$\min_{\mathbf{u}\in\mathbb{R}^n} V(\mathbf{u}),$$

використовуючи динамічну систему другого порядку:

$$\ddot{\mathbf{u}}(t) + \eta\dot{\mathbf{u}}(t) = -\nabla V(\mathbf{u}(t)), \quad \eta > 0.$$

Чисельна реалізація здійснюється за допомогою симплектичних методів (наприклад, Euler або Verlet), що забезпечують стабільність та швидку збіжність методу.

Література

[1] Gulliksson M., Mazur S. *An Iterative Approach to Ill-Conditioned Optimal Portfolio Selection.* Computational Economics, 2020, 56:773–794.

[2] Gulliksson M., Karlsson M., Lindskog F. *A damped dynamic systems approach for solving ill-conditioned optimization problems.* Journal of Computational and Applied Mathematics, 2019, 362: 352–363.

[3] Ding Y., Liu J. *Joint pricing strategies of multi-product retailer with reference-price and substitution-price effect.* Journal of Data, Information and Management, 2021, 3:49–63.

[4] Gulliksson M., Karlsson M. *Damped dynamical systems for solving equations and optimization problems.* In: Advances in Computational Mathematics, Springer, 2020.

Національний університет «Києво-Могилянська академія», Київ, Україна
Email address: i.avdieienko@ukma.edu.ua