

Poznan Monographs in Computing and Its Applications

W serii *Poznan Monographs in Computing and Its Applications* publikowane są najnowsze, wysokiej jakości wyniki badań naukowych w dziedzinie informatyki i jej praktycznych zastosowań. W skład serii wchodzi przede wszystkim rozprawy doktorskie i habilitacyjne napisane przez naukowców z Polski, ze środowiska naukowego Poznania. Najcenniejsze prace wybierane są spośród materiałów wydawanych przez naukowców z Wydziału Informatyki i Wydziału Elektroniki i Telekomunikacji Politechniki Poznańskiej (PUT) oraz z Poznańskiego Centrum Superkomputerowo-Sieciowego (PCSS). Celem cyklu jest promocja interesujących wyników badań i upowszechnienie ich wśród szerszego grona odbiorców. Przed powstaniem serii takie monografie były dzielone z bardzo ograniczonymi grupami ludzi, a dotarcie do zainteresowanych czytelników było dla nich prawie niemożliwe. Każda pozycja w serii jest starannie weryfikowana przez doświadczonych profesorów, a następnie korygowana zgodnie z ich uwagami.

Przewodniczący Komitetu Redakcyjnego

Prof. Jan Węglarz

Komitet Redakcyjny

Prof. Jacek Błazewicz

Prof. Jerzy Brzeziński

Prof. Marek Domański

Prof. Marek Figlerowicz

Prof. Andrzej Handkiewicz

Dr hab. Andrzej Jaskiewicz

Prof. Krzysztof Kozłowski

Prof. Roman Słowiński

Dr Maciej Stroiński

Prof. Krzysztof Wesołowski

Sekretarz Komitetu Redakcyjnego

Dr Szymon Wąsik

Jędrzej Musiał

**Kombinatoryczne modele i algorytmy
różnych wariantów problemu
optymalizacji zakupów internetowych**

*Combinatorial Models and Algorithms for Various Variants of the
Internet Shopping Optimization Problem*

Wydawnictwo NAKOM, Poznań, 2018 r.

Konsultacja językowa: Katarzyna Zienkowicz-Woźniak

Projekt okładki: Agata Łożykowska

Recenzja naukowa: Franciszek Seredyński

© Copyright 2018 Politechnika Poznańska

Książka, ani żadna jej część, nie może być przedrukowywana ani w jakikolwiek inny sposób reprodukowana czy powielana mechanicznie, fotooptycznie, zapisywana elektronicznie lub magnetycznie, ani odczytywana w środkach publicznego przekazu bez pisemnej zgody autora.

Wydanie 1

Wolumin 20

ISBN: 978-83-63919-28-3

Wydawca: Wydawnictwo NAKOM
ul. Krajenecka 52
60-461 Poznań
www.nakom.com.pl
wydawnictwo@nakom.com.pl
tel. 61 852-83-82

Druk: TOTEM
www.totem.com.pl
tel. +48 52 35 400 40

Nie ma żadnego powodu, aby ktokolwiek posiadał w domu komputer

Ken Olsen, 1977

Spis treści

Przedmowa	1
Extended Abstract	3
1 Wprowadzenie	25
1.1 Handel	25
1.2 Zakres materiału	26
2 Wiedza fundamentalna	29
2.1 Złożoność obliczeniowa	29
2.2 Algorytmy	33
2.3 Optymalizacja	36
2.4 Szeregowanie zadań	43
2.5 Oznaczenia, symbole, akronimy	44
3 Internet i handel elektroniczny	47
3.1 Internet	47
3.2 Handel elektroniczny	49
3.3 E-commerce w Polsce	53
3.4 Systemy rekomendujące	54
3.5 Porównywarki cen	56
4 Problem optymalizacji zakupów internetowych	59
4.1 Definicja formalna – model matematyczny	60
4.2 Złożoność obliczeniowa problemu	60
4.3 Rozwiązania wielomianowe dla specjalnych przypadków	62
4.4 Algorytmy	63
4.4.1 Model programowania całkowitoliczbowego	63
4.4.2 Algorytm MIN-MIN i jego warianty	64
4.4.3 Filozofia przetwarzania komórkowego – algorytm CELLULAR	67
4.4.4 Algorytm zachłanny GREEDY	72
4.4.5 Algorytm FORECASTING	73

4.5	Eksperymenty obliczeniowe	74
4.6	Wyniki eksperymentów obliczeniowych – analiza i dyskusja . . .	76
4.7	Podsumowanie	83
5	Problem optymalizacji zakupów internetowych z rabatami	85
5.1	Definicja formalna – model matematyczny	86
5.2	Niepełna oferta produktów	87
5.3	Pełna oferta produktów we wszystkich sklepach	89
5.3.1	Identyczne ceny standardowe	89
5.3.2	Proporcjonalne ceny standardowe z rabatami	90
5.4	Dokładny algorytm podziału i ograniczeń	92
5.5	Wstępne eksperymenty obliczeniowe	94
5.6	Algorytmy heurystyczne	100
5.6.1	Algorytm zachłanny GREEDY	100
5.6.2	Algorytm z prognozowaniem FORECASTING	101
5.6.3	Algorytm przetwarzania komórkowego CELLULAR	101
5.6.4	Algorytm MIN-MIN	103
5.7	Eksperymenty obliczeniowe	103
5.7.1	Modelowanie danych i generator instancji	104
5.7.2	Zbiór eksperymentów obliczeniowych obejmujący rozwią- zanie optymalne	105
5.7.3	Eksperymenty obliczeniowe dla dużego zestawu danych .	108
5.8	Podsumowanie	116
6	Problem optymalizacji zakupów z wieloma rabatami	117
6.1	Definicja formalna – model matematyczny	117
6.2	Algorytmy	119
6.2.1	Proste algorytmy zachłanne G1 i G2	120
6.2.2	Algorytm heurystyczny G3	121
6.3	Analiza teoretyczna algorytmu G3	123
6.3.1	Górne ograniczenie aproksymacji	123
6.3.2	Optymalność w uproszczonej formie	125
6.4	Aplikacja testowa z generatorem instancji	126
6.5	Eksperymenty obliczeniowe	128
6.6	Wyniki eksperymentów obliczeniowych – analiza i dyskusja . . .	128
6.6.1	Wpływ kolejności produktów na wyniki obliczeń algorytmu G1	128
6.6.2	Wrażliwość algorytmu G2 na dobór parametrów	128
6.6.3	Wyniki obliczeń	129
6.6.4	Skalowalność algorytmów – eksperymenty na dużych ze- stawach danych	131
6.7	Podsumowanie	133

7	Zaufanie w zakupach internetowych	135
7.1	Badanie ankietowe	136
7.2	Definicja formalna – model matematyczny	140
7.3	Algorytmy	143
7.4	Eksperymenty obliczeniowe	144
7.5	Podsumowanie	146
8	Podsumowanie i perspektywa dalszych prac	147
	Bibliografia	149

Spis rysunków

1	Average (100 instances) results for 2-10 products bought from 20 stores	12
2	Minimum required seller rating on a scale of 1-10	19
2.1	Struktura klas złożoności obliczeniowych kombinatorycznych problemów decyzyjnych	33
2.2	Problem lokalizacji fabryk – grafika pogładowa	38
3.1	Schematyczna struktura sieci szkieletowej oplatającej cały glob ziemski	48
3.2	System rekomendujący – grafika pogładowa	54
3.3	Widok strony internetowej www.ceneo.pl dla zapytania <i>e-commerce</i>	57
4.1	Reprezentacja pojedynczej komórki przetwarzającej	70
4.2	Komunikacja pomiędzy komórkami przetwarzającymi	71
4.3	Czasy uzyskania rozwiązania przy użyciu algorytmu ILP	77
4.4	Porównanie czasu działania metod ILP, MINMIN+LOCAL SEARCH i CELLULAR	78
4.5	Wyniki uzyskane przez heurystyki dla <i>średniego</i> zestawu danych	80
4.6	Wyniki uzyskane przez heurystyki dla <i>dużego</i> zestawu danych	81
4.7	Współczynnik odchylenia [%] względem rozwiązania optymalnego	82
5.1	Algorytm Cellular-ILS – schemat blokowy	102
5.2	Uśrednione (dla 100 instancji) wyniki dla 2-10 produktów kupowanych z 20 sklepów	106
5.3	Wyniki odchylenia standardowego dla eksperymentu z 20 sklepami i 2-10 produktami	107
5.4	Porównanie czasu działania algorytmów. Eksperyment przeprowadzony dla 20 sklepów	109
5.5	Wyniki działania algorytmów (eksperyment zawierający dane z 20 sklepów)	113
5.6	Wartości odchylenia standardowego (dane z 20 sklepów)	113

5.7	Czas działania algorytmów (dane z 20 sklepów)	114
5.8	Wyniki działania algorytmów (błąd względny). Eksperyment zawierający dane z 40 sklepów	114
5.9	Wartości odchylenia standardowego (dane z 40 sklepów)	115
5.10	Czasy obliczeń algorytmów (eksperyment obliczeniowy z wykorzystaniem danych z 40 sklepów)	115
6.1	Wyniki algorytmów G1 rosnący, G1 malejący i B&B dla $m = 30$ (przedział ufności = 0,95)	129
6.2	Wyniki różnych wersji (parametrów) algorytmów G2 i B&B dla $m = 30$ (przedział ufności = 0,95)	129
6.3	Wyniki działania algorytmów G1, G2, G3, PCS+, B&B dla $m = 20$ (przedział ufności = 0,95)	130
6.4	Wyniki działania algorytmów G1, G2, G3, PCS+, B&B dla $m = 30$ (przedział ufności = 0,95)	130
6.5	Wyniki działania algorytmów G1, G2, G3, PCS+, B&B dla $m = 40$ (przedział ufności = 0,95)	131
6.6	Wyniki działania algorytmów G1, G2, G3, PCS i PCS+ dla $m = 20$	131
6.7	Wyniki działania algorytmów G1, G2, G3, PCS i PCS+ dla $m = 30$	132
6.8	Wyniki działania algorytmów G1, G2, G3, PCS i PCS+ dla $m = 40$	132
7.1	Kluczowe aspekty podczas zakupów online	136
7.2	Minimalna wymagana ocena sprzedawcy w skali 1-10	137
7.3	Ważne aspekty gdy użytkownik zbiera informacje dotyczące sprzedawcy	138

Spis tabel

1	Books prices (A-E) and delivery costs offered by six online shops	6
2	Price comparison site solution – the result of the selection process	6
3	Optimal cost of purchasing A-E books in specific stores	6
4	Table of notation	20
2.1	Porównanie kilku funkcji wielomianowych i wykładniczych	31
2.2	Wpływ funkcji przyspieszenia na czas obliczeń algorytmów wielomianowych i wykładniczych	31
2.3	Przykład problemu plecakowego	40
4.1	Ceny książek (A-E) i koszty dostawy oferowane przez sześć sklepów internetowych	59
4.2	Rozwiązanie porównywarki cen – wynik procesu selekcji	60
4.3	Optymalny koszt zakupu książek A-E w określonych sklepach . .	60
4.4	Struktura cen, dla której algorytm zachłanny zaproponuje najlepsze rozwiązanie	74
5.1	Uśrednione wartości zakupów dla różnych algorytmów. Dane dla 2-15 produktów i 10,20 sklepów	96
5.2	Uśrednione wartości zakupów dla różnych algorytmów. Dane dla 2-15 produktów i 30 sklepów	97
5.3	Średnie wartości proporcjonalne (w procentach odległości od wartości optymalnej). Dane dla 2-15 produktów i 10, 20 sklepów . .	98
5.4	Średnie wartości proporcjonalne (w procentach odległości od wartości optymalnej). Dane dla 2-15 produktów i 30 sklepów	99
5.5	Porównanie dyspersji algorytmów – przedstawienie współczynnika zmienności (CV). Dane dla 20 sklepów i 2-10 produktów	108
5.6	Porównanie dyspersji algorytmów – przedstawienie współczynnika zmienności (CV). Dane z 20 sklepów dla 5-100 produktów	110
6.1	Przykładowe ceny wysyłki dla pewnego sklepu	119

7.1	Notacja dla problemu T-ISOP	141
7.2	Wyniki eksperymentu obliczeniowego	146

Przedmowa

Monografia porusza fascynujący temat zakupów internetowych, zwracając szczególną uwagę na nowatorskie, jeszcze nieodkryte możliwości zastosowania nauki w tej sferze życia. W niniejszej książce dokonano obszernego przeglądu aktualnego stanu wiedzy dotyczącego nowych technologii umożliwiających dokonanie optymalnych zakupów internetowych. Na obecnym rynku e-commerce nadal brakuje efektywnych narzędzi analizy wspomagających kupowanie online. Kluczowym problemem jest brak jakiejkolwiek analizy zakupów złożonych z kilku / wielu produktów. Natomiast takich zakupów dokonujemy zdecydowanie najczęściej. Książka zawiera zbiór najnowszej wiedzy z zakresu optymalizacji zakupów internetowych, podzielony wedle skomplikowania problemu i możliwych do zastosowania rozwiązań. Zdefiniowano formalnie problem optymalizacji zakupów internetowych [13] (ang. Internet Shopping Optimization Problem – ISOP). Zadaniem jest dokonanie zakupu złożonego z wielu produktów płacąc na końcu minimalną możliwą kwotę. Wyboru można dokonać spośród szeregu ofert sklepów internetowych. Możliwe jest również zdefiniowanie wielu dodatkowych wymagań czy ograniczeń. Problem ewoluuje przybierając nowe, ciekawsze i bardziej specjalistyczne modele (coraz bardziej skomplikowane i trudniejsze do rozwiązania). Kolejne prezentowane wersje z każdym krokiem jeszcze lepiej odzwierciedlają sytuację rynkową i odpowiadają na oczekiwania klienta. Dotyczą one w szczególności rabatów oferowanych na zakup wszystkich produktów, bardzo zróżnicowane (nierzadko skomplikowane) tabele kosztów wysyłki, gdzie cena zależy od różnych wybranych form płatności czy kwot zakupów. Zaufanie jako jeden z najważniejszych czynników zakupów nie mógł zostać pominięty w analizie dotyczącej całości problemu. Naturalnym jest, że miejscami skomplikowana matematyczna analiza nie powinna być prezentowana bez zawarcia i przedstawienia odpowiednich podstaw. Są one dostarczone na początkowych stronach książki, aby czytelnik mógł śmiało przewracać kolejne kartki bez konieczności jednoczesnego czytania innych, wspierających pozycji literatury w celu zrozumienia głównych treści. Oczywiście jest fakt, że niektóre fragmenty przedstawione są w sposób skrótowy, niezbędny i wystarczający do płynnego poruszania się w ramach niniejszej publikacji. Aczkolwiek nie wyczerpują opisywanego w danym fragmencie tematu. Całość materiału uzupełniona jest szeroką listą referencji,

do których czytelnik może się w każdej chwili odwołać.

Monografia może służyć zarówno studentom, doktorantom, jak i pracownikom naukowym z dziedziny informatyki, czy bardziej ogólnych kierunków związanych z informatyką. Może również stanowić podstawę programu odrębnego przedmiotu multidyscyplinarnego, bądź część już przygotowanych zajęć dotyczących np. algorytmiki, czy optymalizacji. Przedstawione materiały mają szerokie możliwości zastosowania i wdrożenia. Stanowi obszerny przegląd najaktualniejszej wiedzy z zakresu poruszanego w ramach problemu ISOP.

Podziękowania¹

Pragnę serdecznie podziękować osobom, które uczestniczyły w moim życiu naukowym ostatnich lat. Bez wątpienia w znacznym stopniu przyczyniły się do powstania niniejszej publikacji. Dziękuję prof. dr. hab. inż. Jackowi Błazewiczowi (Politechnika Poznańska, Polska) za wieloletnią, bezcenną opiekę naukową, wspólną pracę, liczne wskazówki dotyczące rozwoju naukowego i wsparcie w dynamicznej realizacji zawodowej. Dziękuję prof. dr. Pascalowi Bouvry (University of Luxembourg, Luksemburg) za wieloletnią, bardzo owocną współpracę, wielką otwartość i wspieranie rozwoju naukowego – szczególnie podczas bardzo wielu miesięcy spędzonych w Luksemburgu.

Chciałbym również podziękować wszystkim osobom, z którymi miałem przyjemność współpracować w ostatnich latach. Będą to w szczególności: prof. dr. Mikhail Y. Kovalyov (United Institute Of Informatics Problems, Białoruś), prof. dr. Héctor J. Fraire-Huacuja (Instituto Tecnológico de Ciudad Madero, Meksyk), prof. dr. Denis Trystram (Grenoble Institute of Technology, Francja), dr. Johnatan E. Pecero (Agency for Standardization and the Knowledge-based Economy of Luxembourg, Luksemburg), dr. Mario C. López Locés (Universidad Autónoma de Nuevo León, Meksyk), dr. Grégoire Danoy (University of Luxembourg, Luksemburg), Emmanuel Kieffer (University of Luxembourg, Luksemburg), dr. Bernabe Dorronsoro (University of Cádiz, Hiszpania), dr. Mateusz Guzek (The Goodyear Tire & Rubber Company, Luksemburg), dr. Pierre-François Dutot (University of Grenoble, Francja), dr. Nathanael Cherié (ENS Rennes, Francja), dr. inż. Adam Wojciechowski (Politechnika Poznańska, Polska). Wiele wspólnych publikacji naukowych było inspiracją do powstania niniejszej książki.

Na końcu, ale nie mniej ważne, dziękuję moim bliskim (Justyna, Izabela, Oliwia), rodzicom (Tadeusz, Małgorzata) i bratu (Bartłomiej). Zawsze byli dla mnie wsparciem i mobilizowali mnie do dalszej pracy. Ze zrozumieniem akceptowali moje liczne wyjazdy służbowe / naukowe. Dziękuję!

¹Przy tworzeniu rysunków 2.2 i 3.2 wykorzystano ikony autorstwa Freepik, monkik i Gregor Cresnar z www.flaticon.com (ang. Icons made by Freepik, monkik and Gregor Cresnar from www.flaticon.com)

Extended Abstract

Computer science is constantly developing in a very dynamic way. At the same time, the scope of its applications, solutions and learning necessary to explore the presented problems, is constantly growing. Complicated mathematical calculations, using specially designed sets of instructions (algorithms) can lead to effective solution of problems from various areas of life. Computer science is definitely applicable science - one whose results, often very complex scientific analyses, can and should be used in practice. It is a science that is ultimately supposed to change the everyday life of every person. In recent years, e-commerce has become a very important aspect not only of life, but also of information technology.

This monograph touches upon the fascinating subject of online shopping, paying particular attention to the innovative, yet unexploited possibilities of applying science in this sphere of life. One of such unresolved problems / aspects of online shopping is the situation in which we intend to buy more than one product at a time (a situation that occurs extremely often). Of course, we want to achieve everything by minimizing the final costs, while being in line with the list of additional requirements and limitations. The paper contains a number of mathematical formulations for the next presented variants of the problem of online shopping optimization, solutions are discussed and comprehensive computational experimental tests are widely commented on. The work is a collection of results of research conducted by the habilitatant on the problem of ISOP in the last 8-9 years. The monograph provides an extensive overview of the current state of knowledge concerning new technologies enabling optimal online shopping. The current e-commerce market still lacks effective analysis tools supporting online shopping. The key problem is the lack of any analysis of purchases consisting of several / many products. On the other hand, such purchases are by far the most frequent. The book contains a collection of the latest knowledge in the field of online shopping optimization, divided according to the complexity of the problem and possible solutions to be applied. The problem of online shopping optimization [13] has been formally defined as Internet Shopping Optimization Problem (ISOP). The task is to make a purchase consisting of many products paying the minimum possible amount at the end. The choice can be made from

a number of online store offers. It is also possible to define many additional requirements or limitations. The problem is evolving into new, more interesting and more specialized models (more and more complicated and more difficult to solve). Subsequent versions presented with each step reflect the market situation even better and respond to the customer's expectations. The following versions of the problem concern in particular discounts offered for the purchase of all products, very diverse (often complex) tables of shipping costs, where the price depends on different forms of payment or purchase amounts. Trust as one of the most important purchasing factors could not be omitted in the analysis of the whole problem. It is natural that sometimes complex mathematical analysis should not be presented without the inclusion and presentation of appropriate basics. They are provided on the front pages of the book so that the reader can confidently turn the pages without having to read other supportive literature items to understand the main content. It is obvious that some excerpts are briefly presented in a way that is necessary and sufficient to move smoothly within the framework of this publication, although not exhaustive in a given section of the subject matter. The whole material is supplemented with a wide list of references to which the reader can refer at any time.

The results of the conducted research may be used by students, doctoral students, as well as research workers in the field of computer science, or more general fields of study related to computer science. They may also form the basis of a separate multidisciplinary subject or part of already prepared classes on e.g. algorithmics or optimization. They can also be simply a collection of materials presented in the form of an interesting publication that describes (and solves) the real problem of almost every Internet user. They are an ideal material of knowledge, which have wide possibilities of application and implementation. They provide a comprehensive overview of the most recent knowledge of the ISOP problem. The main scientific objectives and the results achieved are presented below.

Introduction (Chapter 1)

The chapter 1 presents the subject matter of the book, as well as the objectives and scope of the material contained therein. The chapter 2, in an essential way, presents the issues necessary for the efficient and comfortable movement within the whole book. This chapter provides basic information on computational complexity, algorithms, combinatorial optimization, and task scheduling. In the chapter 3 the history of Internet development and evolution of e-commerce is presented. It also presents the ideas of recommendation systems and websites comparing prices. The following chapter 4 describes and formally defines the problem of online shopping optimization (ISOP). It contains a range of information about possible solutions (algorithms) as well as computational experiments. The chapter 5 presents an interesting variant of the ISOP problem, in which an additional factor is the variable prices of products. The chapter contains for-

mal definitions, descriptions of algorithms, as well as results of computational experiments. In the next chapter 6, even greater changes have been made to the original problem. In addition to dynamically changing commodity prices, the possibility of variable shipping costs has been proposed. These are situations that are very often used in online shops. After a formal definition of the problem, it was decided to design several solutions and examine them using exhaustive computational experiments. One of the most important factors influencing the online shopping market is trust. Therefore, the ISOP problem has been enriched with a complex analysis of trust in the chapter 7. The last chapter 8 provides a brief summary of the information contained in this book. It also provides insights into the development of optimization issues in online shopping.

Fundamental Knowledge (Chapter 2)

This chapter contains basic information that is necessary for the efficient and effective movement with the material contained in the book. The above information should be regarded as basic and sometimes abbreviated, which in no way exhaust the described topics. They are only an indispensable summoning of known fundamental knowledge in the field of computational complexity, optimization, algorithmics or task scheduling. If the reader is interested in deepening the knowledge in a given field, one should follow the widely quoted literature.

Internet and e-commerce (Chapter 3)

This chapter provides a brief introduction to the origins and evolution of the Internet. The following describes the rules of e-commerce, including a large portion of statistical data. This information makes it easy to see the enormity and strength of the current e-commerce market.

Internet Shopping Optimization Problem (Chapter 4)

Internet Shopping Optimization Problem (ISOP), where the customer wants to complete the shopping list using the online shops offer, was originally proposed by Blazewicz et al. [13]. The aim is to make purchases with possibly the lowest final invoice. Product prices and shipping costs must be taken into account.

To illustrate the problem, let's take a simple example. The customer wants to buy five books. Product prices and delivery costs in six different shops are shown in the table 1 below. The aim is to purchase all five books at a minimum total cost. The support from currently available price comparison services (hereinafter referred to as price comparison engines, PCS) allows you to build a customer basket based on the cheapest offers for particular books. The results of the selection process are presented in the table 2. Purchases based on the price engine suggestions would not be optimal (total cost: 210) because a simple price comparison does not include delivery costs, which increase with the number of selected suppliers. In order to find the cheapest solution globally, a full search can be made for all possible purchases. This is inefficient in terms of time and

Table 1. Books prices (A-E) and delivery costs offered by six online shops

Cost	A	B	C	D	E	Delivery	Sum
Shop 1	18	39	29	48	59	10	203
Shop 2	24	45	23	54	44	15	205
Shop 3	22	45	23	53	53	15	211
Shop 4	28	47	17	57	47	10	206
Shop 5	24	42	24	47	59	10	206
Shop 6	27	48	20	55	53	15	218

Table 2. Price comparison site solution – the result of the selection process

	A	B	C	D	E	Delivery	Sum
Cost	18	39	17	47	44	45	210
Shop	Shop 1	Shop 1	Shop 4	Shop 5	Shop 2		

resources, but for such a simple example you can even do it yourself on a sheet of paper. The simple example analyzed by us shows that the optimization process brings noticeable savings. In our example, the optimal purchase price for books (using shops 1 and 4) is 189 (see table 3).

Formal definition - mathematical model

A single buyer is looking for a multiset of products $N = \{1, \dots, n\}$ to buy in m shops. A multiset of available products N_l , a cost c_{jl} of each product $j \in N_l$, and a delivery cost d_l of any subset of the products from the shop to the buyer are associated with each shop l , $l = 1, \dots, m$. It is assumed that $c_{jl} = \infty$ if $j \notin N_l$. The problem is to find a sequence of disjoint selections (or carts) of products $X = (X_1, \dots, X_m)$, which we call a *cart sequence*, such that $X_l \subseteq N_l$, $l = 1, \dots, m$, $\cup_{l=1}^m X_l = N$, and the total product and delivery cost, denoted as $F(X) := \sum_{l=1}^m \left(\delta(|X_l|)d_l + \sum_{j \in X_l} c_{jl} \right)$, is minimized. Here $|X_l|$ denotes the cardinality of the multiset X_l , and $\delta(x) = 0$ if $x = 0$ and $\delta(x) = 1$ if $x > 0$. We denote this problem as ISOP (Internet Shopping Optimization Problem), its optimal solution as X^* , and its optimal solution value as F^* . It has been proven that the ISOP problem belongs to the class of strongly NP-hard problems [13].

Table 3. Optimal cost of purchasing A-E books in specific stores

	A	B	C	D	E	Delivery	Sum
Cost	18	39	17	48	47	20	189
Shop	Shop 1	Shop 1	Shop 4	Shop 1	Shop 4		

Algorithms

A number of algorithms have been proposed to solve the general problem of ISOP – including an accurate/optimal solution and various heuristics. The exact solution is based on Integer Linear Programming Model (ILP). The first approximate approach is a cellular algorithm based on calculations performed by simple processing cells that operate on their own population and share information with other cells to improve system-wide performance. The second heuristic algorithm is related to a specially developed variant of the algorithm MIN-MIN. Subsequent heuristics are built on the philosophy of greedy algorithms, as well as a more developed approach looking for local optimizations.

Computational experiments

Experiments include the evaluation of a number of different methods of solving the ISOP problem: ILP, MIN-MIN and its variants, and a cellular processing algorithm. In addition, numerous comparisons with the two simpler algorithms GREEDY and FORECASTING will be presented.

In order to prepare test data for computational experiments, a realistic model of the e-commerce market was created. The relations between the structure of competition, advertising, prices and their dispersion in online shops were examined. Books were selected as a group of representative products to be included in the computational experiments because of their wide availability in virtual (online) shops.

Some information, analyses and results of computational experiments described in [81, 84, 80, 19] literature were adopted for the model. Attention was focused mainly on the model definition of electronic bookstores (focusing mainly on books, music CDs, film DVDs), prices, acceptance factor, retailer's brand [18]. This is also extremely important for defining the ISOP problem model – on the price range. It should also be noted that consumers can choose from a large number of online bookstores. The data for our sample was collected from 32 stores and included the largest stores in the United States, including Amazon, BarnesandNoble.com, Borders.com, Buy.com, Booksamillion, as well as leading online bookstores in Poland, such as empik.com or merlin.pl. It was decided to review and update the data generation model presented in Blazewicz et al. [13]. The aim was to create a new, more sophisticated and even more realistic model than the one previously proposed in the literature. Due to the very realistic form and scope of work brought into the designed model, it was also used in research on more complex varieties of the ISOP problem - later variants containing even multiple rebate functions and additional sales conditions [11].

The working model was prepared on the basis of the data from the above-mentioned publications, as well as our own observations of many Internet shops. It is assumed that each shop has all the required books. In each instance, the following values are randomly generated for all i and j in the corresponding ranges. *Reference price (ref)* of a product j : $ref_j \in \{2, 4, \dots, 100\}$ with a percentage of

occurrence: 40% between 0 and 20, 16% between 22 and 30, 12% between 32 and 40, 16% between 42 and 60 and 16% between 62 and 100 respectively. The price of product j from shop i : $p_{ij} \in [a_{ij}, b_{ij}]$, where $a_{ij} \geq 0.75ref_j$, $b_{ij} \leq 1.36ref_j$. The detailed structure of price intervals is described in the literature [57]. In addition, each store charges a delivery fee between 0 and 20.

Using the previously described method of instance generation, we have created three sets of data of different sizes.

- A small set with three sub-collections: 3n20m, 4n20m and 5n20m, each composed of 30 cases.
- Medium set, with three subsets: 5n240m, 5n400m and 50n240m, each composed of 20 cases.
- Large set, with three sub-collections: 50n400m, 100n240m and 100n400m, each with 20 problem cases.

In order to determine the correctness of the ILP, the cases from the *small* set were solved using the calculation method and the ILP method, and the results obtained by both methods were compared in order to check whether they achieved the same solution (with the same final value).

In the second test, *medium* and *large* sets were solved with ILP without a time limit in order to achieve optimal values in all cases. Then, knowing the optimal values, the performance of the proposed methods of the solution MINMIN+LOCAL SEARCH and CELLULAR was compared with simple heuristics GREEDY and FORECASTING.

The last test was carried out on an instance consisting of real data obtained from online shops. The aim was to be able to examine the effectiveness of this method in the real-life scenario. The data set consisted of 400 different products from 57 shops presenting a total of 6387 offers – not every shop was able to offer all products.

It is worth mentioning that the test instance generator for the ISOP problem is openly available on the IShOP project website².

Results of computational experiments

For ILP, the 5n, 50n and 100n product families were selected from those available in the Small, Medium and Large Small Small, Medium and Large datasets. This kit was used to measure the time required by ILP to solve instances of different sizes.

For the 5n set, the median time used by ILP to solve each instance is close to 0.25 seconds. For the set of 50n the calculation time increases up to a minute in the case of a subset instance of 50n400m (with the median time close to 30 seconds). However, for a subset of 50n240m is less than 10 seconds. In the case

²<http://www.cs.put.poznan.pl/ishop/generator.php>

of the 100n set, the maximum time of instance resolution was 800 seconds for a subset of 100n400m, while in most cases the time needed to solve this instance was nearly 200 seconds.

The test with real data downloaded from online stores (such as prices, delivery costs and stock levels of each store) shows very good performance of the ILP model. Within 1.05 seconds, an optimal solution was achieved for 400 products and 52 stores, for example.

For larger datasets, the use of heuristics is preferred, as the increase in ILP calculation time is clearly greater than for heuristics. The CELLULAR and MIN-MIN+LOCAL SEARCH are therefore more suitable for solving larger instances with less computational resources. This last observation is critical given the context in which the ISOP problem is to be realized and implemented (thousands of simultaneous submissions of calculations to the server).

After determining the correctness of the proposed ILP model and obtaining data with optimal values from medium and large sets, an evaluation of the performance of solutions with simpler, already known algorithms GREEDY and FORECASTING was started. A computational experiment was carried out in which the four methods presented (MIN-MIN, MIN-MIN+MIDLOCAL SEARCH, MIN-MIN+LOCAL SEARCH and CELLULAR) were used to solve the problems of *medium* and *large* sets – worked until reaching their own stop conditions.

All heuristics in the Medium set present similar solutions in terms of target function values for subset 5n240m and 5n400m. The cellular processing algorithm is slower than heuristics based on the MIN-MIN algorithm.

In the case of sets of 50 and 100 products (n), we begin to see that cellular processing heuristics provide better quality solutions than heuristics based on MIN-MIN. Unfortunately, this is at the expense of more calculation time.

In the case of heuristics based on MIN-MIN, the versions that included a local search algorithm in the process also obtained better solutions. They were more consistent in terms of the quality of the solutions received compared to the basic MIN-MIN. However, there were no significant differences that would justify the inclusion of the local search algorithm in the construction process, or as an analytical final step used in the already built solution. The most important difference turned out to be the time needed by each heuristic to solve the instance. Searching MINMIN+LOCAL SEARCH took even half of the time needed to calculate the algorithm MINMIN+MIDLOCAL SEARCH.

In the case of subset 5n240m and 5n400m from the Medium set, it can be observed that in most cases all heuristics are able to reach optimal values.

In the case of instances from a large set, we observe that the differences between heuristics increase with the size of the instance. What is more, the algorithms obtaining the best results in general are the cellular processing algorithm CELLULAR and the algorithm FORECASTING, respectively. It is also worth noting that the deviation from the optimal value (PERCENTAGE ERROR) for the cellular processing algorithm consistently falls within the range from zero to ten

percent, regardless of the size of the instance set being solved. This is a very important observation showing the high stability of the algorithm.

Internet Shopping with Price Sensitive Discounts (Chapter 5)

We consider a problem in which a customer would like to buy products of a given set $N = \{1, \dots, n\}$ from Internet shops $M = \{1, \dots, m\}$ at the minimum total final price. There are the following given parameters and decision variables:

d_i - delivery price of all products from shop i to the customer;

p_{ij} - standard price of product j in shop i , $p_{ij} = p_j$ if standard prices of product j are the same in all shops;

N_i - subset of products of the set N in shop i (eligible products for shop i), $N_i \subseteq N$;

M_j - subset of shops in which product j can be bought (eligible shops for product j), $M_j \subseteq M$;

S_i - subset of products selected by the customer in shop i (basket of shop i , decision variable), $N = \cup_{i=1}^m S_i$ and $S_i \cap S_j = \emptyset$, $i \neq j$, for a feasible solution;

$T_i(S_i) = d_i + \sum_{j \in S_i} p_{ij}$ - total delivery and standard price in shop i for a given set of products $S_i \subseteq N_i$; if there is no ambiguity, notation S_i in $T_i(S_i)$ can be omitted;

$f_i(T)$ - discounting function for final price, a concave increasing differentiable or concave piecewise linear function of total delivery and standard price T in shop i at all points $T > 0$, $f_i(0) = 0$.

We denote the above problem as ISOPWD. Its mathematical program can be written as follows:

$$\min \sum_{i=1}^m f_i(d_i y_i + \sum_{j \in N_i} p_{ij} x_{ij}) \quad (1)$$

$$\text{s.t.} \quad \sum_{i \in M_j} x_{ij} = 1, \quad j = 1, \dots, n, \quad (2)$$

$$0 \leq x_{ij} \leq y_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad y_i \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (4)$$

Scientific studies on the ISOPWD have commented on a number of specific variants of the problem, including: offers with incomplete availability of products, full availability of products in all shops, identical standard prices, or proportional standard prices.

Algorithms

In order to present better results than the simple heuristics solving the problem of ISOP and ISOPwD known so far, a new set of heuristic approaches to solving the problem has been developed. The set of heuristics consists of a new, efficient metaheuristics based on a cellular optimization process (developed at a similar time for both the basic ISOP problem and for the problem containing the rebate definitions - ISOPwD), an extended version of the already known for ISOP heuristic algorithm, as well as two classic greedy algorithms. We have designed the new heuristics as a solution that balances time and calculation results, with the assumption of being close to optimal solutions. What is extremely important, the computational experimental research was carried out on a large set of real data.

The following algorithms have been proposed and defined:

- simple greedy algorithm GREEDY,
- algorithm with FORECASTING,
- CELLULAR processing algorithm,
- MIN-MIN algorithm.

Computational experiments

The developed data model is based on the model described earlier and its detailed characteristics are described in the literature [73].

The first group of experiments was the one in which the optimal solutions obtained with the use of an accurate algorithm B&B, were compared with all the presented heuristic solutions, i.e. GREEDY, FORECASTING, CELLULAR and MIN-MIN. In that experiment, it was assumed that the number of shops is $n \in \{20\}$ and $m \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ products. For each pair (n, m) 100 instances of the problem were generated. The number of instances analyzed is therefore 900.

Algorithms were compared using three measures: approximation factor, operating time and dispersion index based on standard deviation. The approximation factor is defined as $\rho = \frac{F(X)}{F(X)^*}$ where $F(X)$ represents the solution found by a given algorithm and $F(X)^*$ is the optimal solution.

Figure 1 presents averaged (for 100 instances) results were presented for a set of 2-10 products purchased from among the offer of 20 shops.

Of all heuristics, the cellular algorithm CELLULAR provides the best quality solutions (closest to optimal), regardless of the size of the instance. In the worst case, the solution proposed by this algorithm was only 1.47% more expensive than the cheapest possible. For many individual cases, the cellular algorithm calculates the optimal solution (it should be remembered that the data on the graph represent values averaged over 100 instances of the problem with the same number of shops and products). The cellular algorithm has calculated the optimal solution for 62% of instances. It should be noted that the algorithm is very stable in terms of the quality of the solutions (in most cases it is a solution

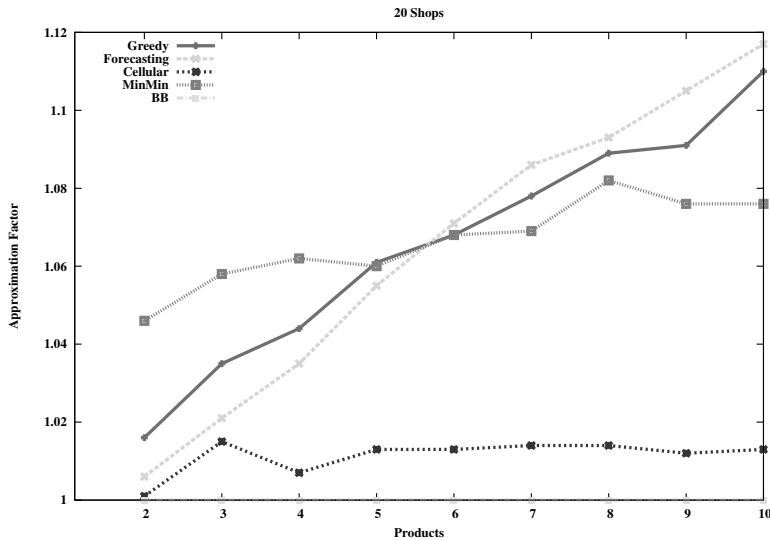


Figure 1. Average (100 instances) results for 2-10 products bought from 20 stores

between 1.24% – 1.47% worse than optimal).

Both the algorithms GREEDY and FORECASTING provide similar quality solutions. For a smaller number of products of m , the latter was better, and for a larger number of products of $m > 5$, the algorithm GREEDY turns out to be better. It is easy to see that the quality of the solution deteriorates with an increasing number of m products. The simple greedy algorithm GREEDY was able to find an optimal solution for 65% of the instance, while the algorithm FORECASTING calculated the optimal value for 75% of the instance.

The last heuristic algorithm used, MIN-MIN, provides the worst solutions for a smaller number of products. However, the algorithm is quite stable in terms of quality, so for more products, it provides better solutions than GREEDY and FORECASTING. The algorithm has found an optimal solution for 20% of the examined instances.

For a small number of products ($m \leq 5$), MIN-MIN is the fastest algorithm. It can be seen that when the number of products is greater than five ($m > 5$), the algorithm GREEDY is faster than the others. The differences between all algorithms are very significant. The running time of the exact algorithm B&B is exponentially increasing and it was not possible to prepare experiments for a larger number of products.

In the second group of experiments, a comparison was made between heuristic algorithms, solving a task consisting of a much larger problem size. In these examples, the size of the instance was increased as follows: $n \in \{20, 40\}$, $m \in$

$\{5, 10, 15, \dots, 100\}$. For each pair 100 instances were generated. The instances were generated using the same procedure. The main measure is the relative error γ . A measure is formally defined as $\gamma = \frac{F(X)}{F(X)_{best}}$, where $F(X)$ represents a solution found by heuristics, and $F(X)_{best}$ represents the best solution of all heuristics.

It can be seen that the cellular algorithm outperforms the quality of the solution of all other algorithms. In all cases of n, m where $m = \{5, 10, 15, \dots, 100\}$ (n is constant = 20), this is the best solution. Compared to MIN-MIN and FORECASTING, the GREEDY algorithm offers better scalability, providing the second best solution in terms of quality. For product quantities over $m > 15$, the relative error measure has stabilized within the limits of 10-11% over the best solution of the CELLULAR. For a small number of products $m = 5$ FORECASTING offers almost the same quality of the solution as the best algorithm CELLULAR. (0.48% more expensive). However, from $m > 15$ products becomes the worst algorithm of all tested products. Moreover, it is easy to see that the quality of the solution deteriorates with the growing number of m products. The last heuristic algorithm Min-Min provides the worst solutions for fewer products ($m < 15$). For more products, however, it provides better solutions than the FORECASTING algorithm (but worse than GREEDY). MIN-MIN provides solutions from 6% to 16% (on average) worse than the best solution.

Another calculation experiment was carried out for the process of optimization of purchases from 40 stores [73].

Given the fast lead time and the good quality of the solutions that GREEDY, FORECASTING and MIN-MIN can offer, it makes sense to use them as the starting grain for the CELLULAR algorithm. In addition, it is worth noting that a special optimization of the database system can be performed in order to shorten the calculation time [62].

Dual Discounting Functions for Internet Shopping Optimization Problem (Chapter 6)

A typical example is the following advertisement in an online shop selling books and CDs: If the value of your purchase is at least 25, we will send you products for half of the standard postage costs; 35 half of the courier delivery price; 50 free postage; 75 free courier delivery. This is just an example to illustrate the situation. Values and thresholds (and their much higher multiplicity) may vary from one vendor to another. In order to attract customers, online shops often apply discounts depending on the total price of the products we want to buy. An example is the following practice observed in an online shop selling computer parts: If the purchase value is greater than 50, then the discount is 3%, 100-7%, 150-10%, 250-15%. Note that although both additional features depend on the amount of the purchase, one does not affect the other directly.

The ISOP problem can therefore be expanded and described with two addi-

tional functions sensitive to the total price of the products purchased. Let's assume that we would like to buy several products (n). Products are offered in many shops (m) – the location of shops is not important for us. The price of each product of n in each store of m can be described as p_{ij} , where $i = 1, \dots, m$, $j = 1, \dots, n$. For each shop of m a characteristic discount function is defined (based on the total cost of purchases in this shop of i) – $f_i(T_i)$. In addition, the shipping cost of d for each store of m is variable and also depends (although the dependency characteristics may be completely different) on the amount of purchases. Our goal is to find out how to pay the lowest bill for all products. The mathematical formulation of the problem can be written as follows:

$$\min \sum_{i=1}^m f_i \left(\sum_{j \in N_i} p_{ij} x_{ij} \right) + \sum_{i=1}^m d_i \left(\sum_{j \in N_i} p_{ij} x_{ij} \right) \quad (5)$$

$$\text{s.t.} \quad \sum_{i \in M} x_{ij} = 1, \quad j = 1, \dots, n, \quad (6)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (7)$$

Both the discount function (f) and the shipping cost (d) can be aggregated to present one common function (fd) based on both criteria. The formulation can therefore be written as follows:

$$\min \sum_{i=1}^m f d_i \left(\sum_{j \in N_i} p_{ij} x_{ij} \right) \quad (8)$$

$$\text{s.t.} \quad \sum_{i \in M} x_{ij} = 1, \quad j = 1, \dots, n, \quad (9)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (10)$$

The discount function can look like this (example):

$$f_i(T_i) = \begin{cases} T_i & \text{if } 0 < T_i \leq 50; \\ 50 + 0,97(T_i - 50) & \text{if } 50 < T_i \leq 100; \\ 50 + 0,97 * 50 + 0,93(T_i - 100) & \text{if } 100 < T_i \leq 150; \\ 50 + 0,97 * 50 + 0,93 * 50 + 0,9(T_i - 150) & \text{if } 150 < T_i \leq 250. \end{cases}$$

You will notice that T_i is the total standard price of the products selected in the shop $i \in M$. The shipping cost function is more complicated because its shape depends on the customer's later choice (type of delivery - e.g. post office, courier, payment method). We can define it as $d_i(T_i, dv_1, dv_2)$, $i = 1, \dots, m$. Therefore, in this discussion we can assume that the customer always makes the same decisions dv_1, dv_2 .

Algorithms

Finding an accurate solution to the ISOPwDD problem requires exponential time, but in order to be able to compare heuristic results to an optimal solution, an accurate B&B algorithm has been implemented with several improvements leading to a faster calculation of the best possible solution.

In addition, a new advanced heuristic algorithm has been developed and proposed. Moreover, it seems important to compare the new heuristics with the existing ISOP algorithms (of course, the algorithms can be adapted/reconstructed to the specificity of the problem). The analysis will also include solutions that are used on price comparison websites. Let's call the latter PCS respectively (Price Comparison Site) and its revised version PCS+. The PCS always selects the shop where the product is cheapest, while PCS+ selects a shop with a minimum total product price and shipping cost.

In order to perform the first computational experiments for the new ISOPwDD problem, it was decided to test two simple heuristics known to be used in the ISOP problem. The study was complemented by an optimal solution, as well as two variations of the algorithm working in price comparison engines.

In the first heuristic algorithm denoted as G1, products are considered in a certain order. The algorithm is set for various product orders and the best solution found is presented to the customer. In the following example it is assumed that products are ordered $1, \dots, n$. Values of total delivery and standard price for all shops are initially set as $T_i = d_i$, $i = 1, \dots, m$. In iteration j of algorithm G1, product j is selected in its eligible shop $i \in M_j$ with minimum value $fd_i(T_i + p_{ij})$, and the corresponding T_i -value is re-set: $T_i := T_i + p_{ij}$.

The second algorithm tested is the algorithm G2, which has a mechanism to prevent the occurrence of an unfavorable situation for G1. For the problem of ISOPwDD we used an updated, differently prepared version of the algorithm G1, where the general principle of fast action and prevention of a possible bad situation occurring in the case of G1 are maintained. G2, unlike G1, starts by choosing a shop in which items should be bought: the one chosen have the lowest average price of items including delivery fees. For each shop i the valuation $V_i = (d_i + \sum_{j \in N_i} p_{ij})/|N_i|$ is calculated, then from the shop k which has the lowest valuation V_k , the set S of the $\alpha\%$ cheapest available items are taken. Then those items are removed from the shops $\forall i \in M, N_i := N_i \setminus S$ and d_k is updated to the value $d_k = 0$ since the delivery fee for shop k has already been paid.

Both of these heuristics provide efficient work (quick action) and good results. However, in order to obtain even better solutions to the ISOPwDD problem, an attempt was made to create a new, even more efficient heuristic algorithm.

The idea behind the newly proposed heuristics, marked G3, is to create product groups. The products are grouped in such a way as to maximize the V_i pricing ratio defining the savings to the maximum cost ratio in all stores.

In order to do so, for every shop, the item with the biggest ratio is put aside into a temporary basket C_i . If by adding another product this ratio can

be increased, the product is also taken into C_i and this process continues until there is no more item available in the shop or if the ratio cannot be increased any more. Then the shop k with the highest ratio V_k is chosen and all the items in C_k are placed in B_i to be bought from this shop. This step is repeated until all the products on the shopping list are bought.

The expression for V_i takes into account the basket B_i containing all what has already been chosen for this shop before and the temporary basket C_i :

$$V_i(C_i, B_i) = 1 - \frac{f_i(T_i(B_i \cup C_i)) - f_i(T_i(B_i))}{p_{max}(C_i)}$$

The G3 algorithm computational complexity is $O(n^3m)$. This is greater than the complexity of algorithms G1 and G2, but it is worth noting that G3 does not require any additional tuning and does not depend on the order of the elements. However, G3 has some interesting features such as the upper limit and optimality in some specific cases, and in general, better results can be expected [11].

Computational experiments

For the purpose of the calculation experiment, a detailed model of the data generator has been developed [11].

As most of price comparison websites only compare prices from 25 to 30 shops, m is taken in $\{20, 30, 40\}$, likewise, an average customer will only buy a few books at the same time so n is taken in $\{2, 3, 4, 5, 6, 7\}$.

These choices also take into account that the search for an optimal solution is exponential in time. In order to study how the algorithms react for bigger instances, for n from 5 to 100 by steps of 5, the heuristics G1, G2, PCS and PCS+ will later be compared to the results of G3 instead of being compared to an optimal solution. 400 instances of the problem are used for every experiment.

Results

G1 has been tested with two different strategies: taking items in the order of non-increasing reference prices and in the opposite order. Using the descending price order provides on average better solutions than the ascending order. Furthermore, G1 also provides approximated solutions within 2.5% of the corresponding optimal solution. It shows that a pure greedy algorithm, can provide a good approximation in a short time: G1 is in $O(nm)$ but sorting the products also takes at least $O(n \ln n)$.

In order to establish the sensitivity of G2 to its parameter α , it has been tested with 4 different values: 25%, 50%, 75% and 100%. The minimum value from the 4 tests has also been taken into account. It suggests that G2 needs some fine tuning to be able to provide good approximations: even the best solution out of the four parameters have a difference ratio greater than 5% for a low number

of items (for $n = 4$). The choice of the parameter α has a great impact on the performances of G2 so it has to be chosen carefully.

For each pair (n, m) , the algorithms PCS+ and G3 have been tested against the solution G1 with a declining price strategy and then compared to the optimal solution. Since G2 requires fine-tuning, the best solution provided by G2 was taken into account, using the parameters 25%, 50%, 75% i 100%. The results of the PCS algorithm have been omitted because it provided solutions that were at least 15% worse than the optimal solution.

G3 provides solutions that are no more than 0.59% worse than the optimal solution. This is by far the best of all tested heuristics. Interestingly, the simple algorithm G1 provides a much better solution than PCS+. The approximation factor for G3 increases with the increase in the number of elements, but at a much slower rate than for PCS+ and G1.

Moreover, G2 is very dependent of the number of shops: for $m = 20$, the difference ratio between G2 and an optimal solution does not exceed 7% for n in $\{2, 3, 4, 5, 6, 7\}$ but for $m = 40$ and $m = 30$, it is exceeded for $n = 6$. G3 provides good solutions close to the optimal one and better than the one provided by previous algorithms.

In a computational experiment for an instance with a larger problem size, the computational results of all heuristics were compared. It was not possible to place the results of the exact algorithm B&B due to its high computational complexity and times too long to register. Without looking for an optimal solution, the n parameter could have been much higher than in previous tests. Therefore, it was assumed that n is selected from the $\{5, 10, \dots, 95, 100\}$ set, while m remains unchanged.

G3 has better results than other heuristics even with large numbers of products. G1 produces solutions that are at least 0.66% bigger and those from PCS+ are also at least 2.5% bigger. Those gaps quickly increase, for n bigger than 20, the difference between G1 and G3 is at least 1.3%. The solutions provided by PCS become better as the impact of delivery fees on the total cost decreases when the number of items increases. Besides, PCS even provides better results than PCS+ when n increases.

G3 provides better solutions than the other algorithms even with larger number of items. Nonetheless, G3 has a higher cost in complexity ($O(mn^3)$) than G1 ($O(nm)$).

Trust in online shopping (Chapter 7)

Customers usually prefer to give up cheaper offers, which are offered by less reputable retailers, in favour of buying a more expensive product, offered by a well-known and trusted shop with greater prestige among other consumers [108, 45]. Taking into account such behavior observed among the majority of customers of online stores, it was decided to extend the problem of online shopping optimization with additional indicators related to the reputation of the stores and the trust factor.

It is worth stressing that, according to a study conducted by Nielsen [77] among more than 27,000 online consumers in 55 markets around the world, users of e-commerce platforms appreciate the opinions and reviews of previous customers. They even regard them as a decisive factor in the decision making process involving possible purchases from a given store. Zhang, Cheung and Lee [111] examined an interesting issue (question) of how inconsistent opinions play a role in consumers' online purchasing decisions.

In the work of Flanagan et al. [30] addressed the problem of using online merchant ratings and how customers use them to select more reputable online shops. The use of these indicators can help to develop a system whereby a potential buyer can buy a list of products from reputable online companies at the best price.

In preparing the ISOP model with confidence analysis, we decided to use the information contained in cc05 literature and our own research, insights and observations. As a result, a questionnaire was developed to supplement the information already mentioned. The aim was to answer the most important questions about the trust the customer can place in sellers when shopping online. We managed to collect 168 completed questionnaires. The questionnaires were then carefully analyzed [78]. The answers given in the questionnaire allow us to observe several key conclusions (details are described in the literature [72]):

- Nearly 80% of respondents stated that price is a key factor in choosing a shop when shopping online,
- For slightly less than 20% of survey participants the delivery time is an important element,
- The reputation of the retailer will be a very important / key factor for 60%,
- Customers pay attention to the ratings presented both numerically and graphically,
- For 95% of the population surveyed, shop reviews are either very important or significant,
- It is worth noting that 75% of respondents read reviews written by other customers,
- 38% of those who filled in the survey stated that the length of time a shop is an important piece of information about the shop assistant,
- It also seems that both the very wide range of shipping and payment methods are not as significant. For only 23% and 18% of people respectively, this is im-

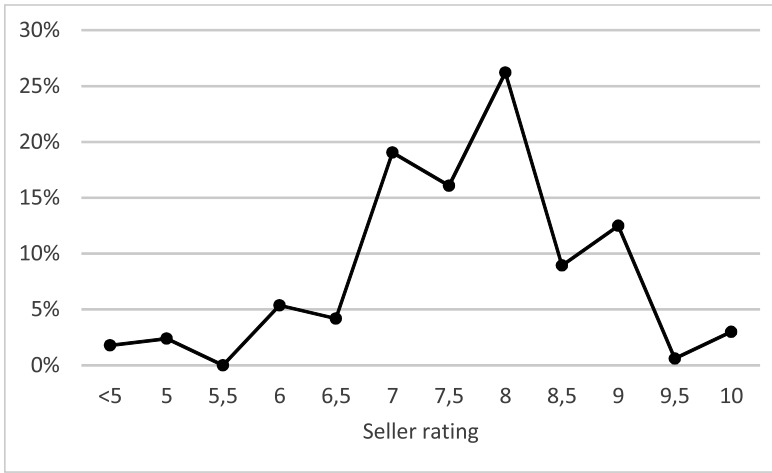


Figure 2. Minimum required seller rating on a scale of 1-10. On the Y-axis, the proportion of responses for the population is determined

portant,

- More than 69% of respondents say they will not buy if delivery prices are too high, even if the prices of the products are attractive,
- Nearly 73% of people discredit sellers with a percentage of strongly negative opinions, even if they are in the minority. This shows that one bad rating has a much higher weighted (negative) value than one positive rating,
- 83% of respondents are willing to pay more for products they buy from reputable retailers,
- 36% of respondents will give up shopping if the vendor does not support their favorite shipping method,
- 90% of people declared that their own opinion (previous experience) is either very important or important.

Price is still one of the most important factors influencing purchasing decisions. However, we are confident that customers' trust, reputation and voice are no less important. Customers are ready to pay much more (10-30%) for products from reputable (or almost impeccable) vendors with an impeccable (or almost impeccable) reputation.

Formal definition - mathematical model

The Trusted Internet Shopping Optimization Problem (T-ISOP), which describes in more detail the actual shopping situations related to an important trust factor, is presented as follows:

$$\min \sum_{i=1}^m \sum_{j=1}^n x_{ij} \frac{p_{ij}}{OPay_j} v_j + \sum_{j=1}^n y_j d_j \quad (11)$$

$$\text{s.t. } \sum_{j=1}^n x_{ij} = 1 \quad \forall i \in M, \quad (12)$$

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = m, \quad (13)$$

$$m * y_j - \sum_{i=1}^m x_{ij} \geq 0 \quad \forall j \in N, \quad (14)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N, \quad (15)$$

$$y_j \in \{0, 1\} \quad \forall j \in N, \quad (16)$$

$$OPay_j = (1, \dots, maxOp) \quad \forall j \in N, \quad (17)$$

$$v_j \in \{1, \infty\} \quad \forall j \in N. \quad (18)$$

where all variables and symbols used in patterns are listed in the table 4.

Symbol	Explanation
m	number of products to buy
n	number of shops
i	product indicator
j	shop indicator
d_j	delivery price for all products from shop j
y_j	usage indicator for delivery price from shop j
p_{ij}	cost of product i in shop j
x_{ij}	usage indicator for product i in shop j
$OPay_j$	overpay trust function for shop j
v_j	trust veto factor for shop j

Table 4. Table of notation

After analysis of the literature, the survey and gathering own experience the above general model could be presented in the following way (especially when we tackled two new functions, overpay ($OPay$) and veto (v)).

One should notice that the above equations optimize the process of shop selection. When it is done, a customer will be charged for basic prices where the trust factor does not count - $\sum_{i=1}^m \sum_{j=1}^n x_{ij} p_{ij} + \sum_{j=1}^n y_j d_j$. Both $OPay$ function and v factor can be constructed according to the specific requirements. Function ($OPay$) represents combined trust factors and their impact on the

price. Function returns values between 1 and $maxOp$ and reflects the cumulative trust overpay value - how much customer can pay additionally ($Opay * 100\%$) to buy products from a store with a very good reputation / rank. In theory $maxOp \rightarrow \infty$. However, this is obviously a very improbable situation. The Survey answers show that the $maxOp$ value for all the population was 1.4 (= 40% more expensive - and that value was acceptable just for one person).

$OPay$ function should be presented in the more detailed way. One should notice that we already described many factors that are important to customers (their general trust). General trust factor has an influence on the customers' decisions and very often they prefer to pay some extra for the desired products when shopping from trusted stores. $OPay$ function can be modeled as:

$$OPay_j = overprice_j + Others_j \quad \forall j \in N \quad (19)$$

$$Others_j = dTime_{ij} * isL_j * dur_j * payR_j * dOpt_j \quad \forall i \in M, \forall j \in N \quad (20)$$

where *overprice* factor is calculated for each store j to present what is the overpay value dedicated to the shop. *dTime* is the delivery time, *isL* answers the question is the seller is local (same country), *dur* is the seller's seniority, *payR* describes payment options range and *dOpt* is delivery options range. In the future this function (like both $OPay$ and *veto* functions) will be personalized to the user. This personalization process can be realized as a decision-aided questioner tool. For the general user purpose one can use data gathered from the survey and literature to prepare some default instance of the model values. Seeing that the most important factors are ratings and prices the *overprice* factor will be prepared as follows. One should create a matrix of rating / overpricing values pairs. All values pairs should be calculated according to the data. Moreover, average value and standard deviation will be calculated. For the normal distribution, the values more than two standard deviations from the mean ($\mu - 2\sigma$, $\mu + 2\sigma$) will be cutoff leaving for the analysis 95.45% of the population (pairs), that will be included to the model and use accordingly with their calculated normal distribution.

The *veto* factor returns just two values, a negative (no veto) one that equals 1 and a positive one (veto) equals ∞ . When it is positive, the offer should not be considered since a customers have some serious reservations to this seller. One should remember that the *v* factor could be personalized following a customer's requirements and could be modeled as:

$$v_j = rat_j * del_j * neg_j * dForm_j \quad \forall j \in N \quad (21)$$

$$rat_j, del_j, neg_j, dForm_j \in \{1, \infty\} \quad \forall j \in N \quad (22)$$

where *rat* is the seller's general rating, *del* answers the question regarding delivery options range, *neg* is the influence of negative opinions and *dFrom* is the delivery forms range.

An algorithm has been proposed, which can be used to effectively solve the problem. However, it should be emphasized that this is only the first step and the first attempt to define the T-ISOP problem. Further work and deeper analysis of the problem is necessary.

Summary (Chapter 8)

The development of the currently known Internet network began in the 60s of the last century. Along the way it has undergone many changes and evolved. The amount of information circulating in different corners of the network is overwhelming and increasingly difficult to analyze, even through the most modern machines and systems for analyzing large data sets. The Internet is not only a source of information, but practically a way of life. Thanks to it you can work, learn, communicate efficiently (including video) with distant countries, watch films, listen to music, play games, do the shopping, etc. The development of the e-commerce market has also been going on for many years. As of today, among some product ranges, online shops are beginning to supplant the traditional model of commerce. The ease of reaching a much wider group of customers or obtaining many offers for the same product stimulates the growth of the e-commerce market. The very dynamic development of e-commerce observed in recent years results in an overwhelming amount of commercial information for the customer. Therefore, solutions to support online shopping are appearing. Among the existing ones, it is worth mentioning price comparison engines or recommending systems. A wide range of possible forms of delivery or payment encourages online shopping even more. Unfortunately, according to my best knowledge, to this day there has not been an efficient solution, which will be able to analyze many products purchased by us. This solution will enable us to buy at the lowest possible price and will also be able to meet a number of requirements and additional criteria.

The problem of optimization of online shopping is illustrated by the situation where the customer intends to buy several products from online shops. In the simplest form it is supposed to get an answer where to make a purchase (it can be many shops or one) in order to pay the lowest possible amount. Subsequent versions/developments of the problem describe additional attributes and present increasingly complex (although, which should be emphasized, closer to the customer's requirements) versions of the ISOP problem. Additional issues may concern e.g. variable shipping prices, shop discounts, trust factors, the possibility of your favorite shipping method or payment. They may concern requirements for short-term products or a limited budget.

This book briefly presents the current situation / condition of the e-commerce market as an introduction to the main topic. It is a presentation of different (all its most important varieties) variants of the ISOP problem. Mathematical models, descriptions, presentations of proprietary algorithms and results of computational experiments are presented.

It is worth stressing that due to the applicability of the research, the subject matter is of great interest to the scientific and business circles. The best example of this can be the fact of obtaining a scientific and research grant for the Internet Shopping Optimization Project³ (co-funded by NCBR⁴ and FNR⁵). It was carried out with great success in 2014-2017. The results obtained exceeded the requirements set for the team implementing the project. The author of this publication was one of the main scientists involved in the research. At present, there are many conversations concerning possible implementations of the ISOP subject matter.

The ISOP problem is constantly evolving. There are many market situations which are still worth modeling. Apart from variants of the problem closely related to the subject matter of the ISOP problem, another interesting research point could be a simple simulation tool to support decision making. In addition, it could be beneficial to use biology-inspired approaches to solve the problem. In addition, many additional functions (such as those related to trust factors) can be personalized for each user. The personalization process will be implemented as a decision-making tool for the questioners. This allows to prepare an experimental analysis dedicated to each user according to their specific needs and requirements. However, all these steps require a huge amount of upcoming, exciting work.

³<http://www.cs.put.poznan.pl/ishop/>

⁴National Centre for Research and Development – <https://www.ncbr.gov.pl/en/>

⁵Luxembourg National Research Fund – <https://www.fnr.lu>

Rozdział 1

Wprowadzenie

Informatyka rozwija się niezmiennie w bardzo dynamiczny sposób. Jednocześnie zakres jej zastosowań, rozwiązań i nauki niezbędnej do zgłębiania przedstawianych problemów stale rośnie. Skomplikowane obliczenia matematyczne, przy wykorzystaniu specjalnie zaprojektowanych zbiorów instrukcji postępowania (algoritmów) mogą prowadzić do efektywnego rozwiązania problemów z najrozmaitszych dziedzin życia. Informatyka jest nauką zdecydowanie aplikowalną – taką, której wyniki, nierzadko bardzo skomplikowanych analiz naukowych, można, a nawet należy wykorzystywać w praktyce. To nauka, która docelowo ma zmieniać życie codzienne każdego człowieka. W ostatnich latach handel internetowy staje się niezmiennie ważnym aspektem nie tylko życia, ale także informatyki. Fascynujący temat zakupów internetowych zwraca szczególną uwagę ze względu na ogromną popularność i bardzo szerokie grono odbiorców. Sygnalizuje także liczne, nowatorskie, jeszcze nieodkryte możliwości zastosowania nauki w życiu.

1.1 Handel

Handel towarzyszy człowiekowi od bardzo dawnych czasów. Początkowo nie istniały żadne środki płatnicze, a handel odbywał na zasadzie barteru. Zgodnie z ustaleniami handlujących stron towar wymieniano na inny, należący do przedstawiciela drugiej strony rozmów. Była to z pewnością wygodna i stosunkowo łatwa forma pozyskiwania niezbędnych towarów i pozbywania się tych, które dostępne były w nadmiarze. Mimo, że jeden przedmiot podlegał wymianie na inny, już wtedy można mówić o różnej wartości dóbr materialnych. Im rzadszy był dany produkt, tym więcej innych produktów można było za niego otrzymać. Często jednak panowało założenie, że wymienia się jeden przedmiot na inny, również w tej samej ilości. Z czasem zorientowano się, że różne właściwości przedmiotów, trudność ich wytworzenia, czy pozyskania, powodują, że wymiana jeden-

do-jeden nie jest sprawiedliwa. Często też wymiana barterowa była uciążliwa, gdyż za każdym razem wymagane było transportowanie i dostarczanie określonych towarów. Pojawił się więc pomysł przydzielenia niektórych towarów to grup tzw. towarów pośrednich (podzielono na kilka kategorii w zależności od wartości) jak np. skóry, sól, zboże. Wspomniane produkty konsumpcyjne podlegały dwóm rodzajom transakcji, kupna i sprzedaży. Dobra konsumpcyjne z czasem zamieniano na wygodniejsze w handlu metale. Pierwsze „monety” (kawałki metalu z nabitymi nań podobiznami władców, co różnicowało wartości danych kawałków) pojawiły się najprawdopodobniej u Fenicjan. Oficjalnie jako środek płatniczy monety posłużyły gdy zaczęły być wybijane przez władze państwa, obszaru prowincji. Obecna wiedza wskazuje nam na VII w p.n.e.

2700 lat później handel i pieniądź stanowią nieodłączną, codzienną część współczesnego świata. Co więcej, wirtualne pieniądze zdecydowanie wypierają fizyczne banknoty i bilon. Reprezentowane są poprzez odpowiednie wartości wyświetlane na ekranie monitora. Całość nieodłącznie działa przy wykorzystaniu odpowiednich umów i zaufania, poświadczonego gwarancjami rządów danych państw i banków.

Handel tradycyjny również ewoluuje i od pewnego czasu coraz większą popularnością cieszą się zakupy dokonywane na odległość, w szczególności przy pomocy rozpowszechnionej od dziesięcioleci sieci Internet. Zakupy internetowe już na dobre weszły do kanonu życia codziennego. Wygoda kupowania (siedząc na przykład w swoim domowym zaciszu przed kominkiem) i bardzo szeroka oferta produktów dostępnych poprzez sieć Internet sprawia, że w niektórych branżach sklepy internetowe zaczynają stopniowo wypierać te tradycyjne. Mnogość ofert może natomiast powodować problemy z dokonaniem ostatecznego, najbardziej trafnego wyboru. Ponadto, ceny produktów, kosztów wysyłki, czy dodatkowe koszty, potrafią być bardzo zróżnicowane. Analiza rynku staje się więc coraz trudniejsza. Co więcej, przy dokonywaniu zakupu kilku produktów (co zdarza się bardzo często) wybór najlepszej oferty jest wręcz niemożliwy ze względu na ogrom czynników z tym związanych.

1.2 Zakres materiału

Rozdział 1 przedstawia tematykę książki, a także określa cele i zakres materiału w niej zawartego. W rozdziale 2, w sposób esencjonalny, przedstawiono zagadnienia niezbędne do sprawnego i wygodnego poruszania się w ramach całej książki. Rozdział ten zawiera podstawowe informacje dotyczące złożoności obliczeniowej, algorytmiki, optymalizacji kombinatorycznej, czy szeregowania zadań. W rozdziale 3 przybliżono historię rozwoju sieci Internet i ewolucję handlu elektronicznego. Zaprezentowano również idee systemów rekomendujących i serwisów internetowych porównujących ceny. Następujący rozdział 4 opisuje i formalnie definiuje problem optymalizacji zakupów internetowych (ISOP). Zawiera szereg

informacji dotyczących możliwych rozwiązań (algorytmów), jak i eksperymentów obliczeniowych. W rozdziale 5 przedstawiono ciekawą odmianę problemu ISOP, w której dodatkowym czynnikiem są zmienne ceny produktów. Rozdział zawiera formalne definicje, opisy algorytmów, a także wyniki badań eksperymentów obliczeniowych. W kolejnym rozdziale 6 dokonano jeszcze większych zmian względem oryginalnego problemu. Oprócz dynamicznie zmieniających się cen towarów, zaproponowano możliwość zmiennych kosztów wysyłki. Są to sytuacje bardzo często wykorzystywane w sklepach internetowych. Po formalnym zdefiniowaniu problemu podjęto się zaprojektowania kilku rozwiązań i przebadania ich korzystając z wyczerpujących eksperymentów obliczeniowych. Jednym z niezwykle ważnych czynników, wpływających na rynek zakupów w sieci jest zaufanie. Dlatego też, problem ISOP został wzbogacony o złożoną analizę zaufania w rozdziale 7. Ostatni rozdział 8 zawiera krótkie podsumowanie zawartych w niniejszej książce informacji. Przedstawia również przemyślenia dotyczące kierunków rozwoju optymalizacyjnych zagadnień zakupów internetowych.

Rozdział 2

Wiedza fundamentalna

Niniejszy rozdział zawiera podstawowe informacje, które są niezbędne do sprawnego i efektywnego poruszania się w dalszej części zawartego w książce materiału. Przytoczone wiadomości należy uznać za podstawowe i czasami skrótowe, które w żadnym wypadku nie wyczerpują opisywanych tematów. Są jedynie nieodzownym przywołaniem znanej wiedzy fundamentalnej z zakresu złożoności obliczeniowej, optymalizacji, algorytmiki, czy szeregowania zadań. Jeśli czytelnika interesuje pogłębienie wiedzy z danego zakresu, należy odwołać się do licznie cytowanej literatury.

2.1 Złożoność obliczeniowa

Problem optymalizacji zakupów internetowych (ang. Internet Shopping Optimization Problem, ISOP) [13] należy do klasy problemów optymalizacji kombinatorycznej. Bez wątplenia znajomość zagadnień złożoności obliczeniowej problemów kombinatorycznych jest podstawą do właściwego zdefiniowania i rozwiązania wielu problemów współczesnej informatyki, również rzeczzonego problemu ISOP. W niniejszej sekcji zaprezentowano esencję teorii złożoności obliczeniowej. Jako źródło posłużyły doskonale znane i ogromnie cenione pozycje literatury, a mianowicie książki Garey, Johnson'a [33] i Błażewicza [6].

Istnieją dwie podstawowe klasy problemów związanych ze złożonością obliczeniową – problemy decyzyjne i optymalizacyjne. Problem decyzyjny posiada wyłącznie dwa możliwe rozwiązania. Rozwiązaniem problemu jest odpowiedź *tak* lub *nie* na sformułowane w problemie pytanie. Do grupy problemów decyzyjnych można zaliczyć ogromną liczbę prawdziwie życiowych problemów (z wielu różnych dziedzin, takich jak: logika, teoria grafów, teoria liczb i wiele innych).

Przykłady:

- Problem spełnialności (ang. Boolean satisfiability problem).

Czy dla podanego wyrażenia (może zawierać zmienne, nawiasy i akcje: logiczne połączenie {AND}, logiczne wyłączenie {OR}, negacja {NOT}) istnieje przypisanie wartości TRUE (1) i FALSE (0) do zmiennych, które sprawiają, że całe wyrażenie jest prawdziwe. Problem spełnialności jest problemem NP-zupełnym.

- Cykl Hamiltona.

Za [22] możemy stwierdzić, że *Graf skierowany* G to uporządkowana para (V, E) , gdzie V to skończony zestaw wierzchołków (czasami nazywanych węzłami) w G , a E to zestaw uporządkowanych par elementów z zestawu V , zwanych łukami lub krawędziami skierowanymi. Łuk $(u, v) \in E$ to bezpośrednia krawędź od u do v . Wierzchołek u jest bezpośrednim poprzednikiem v , a wierzchołek v jest bezpośrednim następnikiem u .

Graf nieskierowany G to uporządkowana para (V, E) , gdzie V to skończony zestaw wierzchołków w G , a E to zbiór par wierzchołków, zwanych krawędziami. W grafie nieskierowanym krawędź (u, v) jest równa krawędzi (v, u) . Stopień wierzchołka określa liczba incydentnych krawędzi.

Cykl Hamiltona w grafie nieskierowanym $G = (V, E)$ jest prostym cyklem, który zawiera każdy wierzchołek w V . Mówi się, że graf jest hamiltonowski jeśli zawiera w sobie cykl Hamiltona. Możemy zatem zdefiniować problem cyklu Hamiltona jako pytanie – “Czy graf G ma cykl Hamiltona?”. Udowodniono, że problem należy do klasy problemów NP-zupełnych.

Oprócz problemów decyzyjnych, istnieje bardzo duża klasa różnych problemów optymalizacyjnych. Wymagają one ekstremalizacji (maksymalnej lub minimalnej) funkcji celu. Bardzo ważne jest to, że zarówno problemy optymalizacyjne, jak i decyzyjne mogą być badane w ten sam sposób (z punktu widzenia złożoności obliczeń). Dzięki temu można połączyć problem optymalizacji z odpowiadającym mu problemem decyzyjnym (łatwo zauważyć, że dla każdego problemu optymalizacyjnego ten drugi zawsze będzie istniał [6, 33]). Jeśli zdefiniujemy pierwotny problem optymalizacyjny i znajdziemy rozwiązanie w sposób “łatwy” (w czasie wielomianowym – jego czas działania jest górnym ograniczeniem wielomianowego wyrażenia o wielkości wejścia dla algorytmu), to problem decyzyjny (odpowiednik) nie jest trudniejszy niż pierwotna wersja problemu optymalizacyjnego. Z drugiej strony, jeśli problem decyzyjny jest trudny do rozwiązania obliczeniowo (NP-zupełny), wówczas problem optymalizacyjny odpowiednika jest również trudny obliczeniowo (NP-trudny). Ta cecha jest bardzo ważna, gdy staramy się dostarczyć i zdefiniować dowody na złożoność (przypisać do odpowiedniej klasy złożoności) obliczeniową problemów optymalizacyjnych. W wielu przypadkach powiązania pomiędzy optymalizacją, a podejmowaniem decyzji są bardziej rygorystyczne. Czasami można udowodnić, że problem decyzyjny również nie jest łatwiejszy niż problem optymalizacyjny jego odpowiednika.

Problem decyzyjny π na nieskończonym zestawie wejść odpowiada na pytanie arbitralnie *tak* lub *nie*. Przykładowa instancja I problemu π jest uzyskiwana przez przypisanie wartości do wszystkich parametrów problemu. Może być dekodowana jako ciąg parametrów za pomocą symboli na skończonym alfabetcie. Oznaczmy D_π jako zestaw instancji problemu decyzyjnego π . Jego podzbiór $Y_\pi \subseteq D_\pi$ zawiera wszystkie instancje, dla których odpowiedź brzmi *tak*. Warto zauważyć, że podobna definicja może być podana w przypadku problemów optymalizacyjnych [6, 33]. Długość tego łańcucha jest wielkością instancji. Jeżeli liczby występujące w definicji problemu są reprezentowane w bazie binarnej lub w innej bazie większej niż 1, a w danych kodowania łańcucha symboli nie ma niepotrzebnych symboli, taki schemat kodowania jest uzasadniony.

Tabela 2.1. Porównanie kilku funkcji wielomianowych i wykładniczych (źródło danych: tabela 1.1. z [6])

Funkcja złożoności obliczeniowej	n	
	10	60
n	0,00001 sekund	0,00006 sekund
n^3	0,001 sekund	0,216 sekund
n^5	0,1 sekund	13 minut
2^n	0,001 sekund	3366 stuleci
3^n	0,059 sekund	$1,3 \times 10^{13}$ stuleci

Tabela 2.2. Wpływ funkcji przyspieszenia na czas obliczeń algorytmów wielomianowych i wykładniczych (źródło danych: tabela 1.2. z [6])

Funkcja złożoności obliczeniowej	Największy problem rozwiązywalny w godzinę		
	obecny komputer	100 razy szybszy komputer	1000 razy szybszy komputer
n	N_1	$100 N_1$	$1000 N_1$
n^3	N_2	$4,64 N_2$	$10 N_2$
n^5	N_3	$2,5 N_3$	$3,98 N_3$
2^n	N_4	$N_4 + 6,64$	$N_4 + 9,97$
3^n	N_5	$N_5 + 4,19$	$N_5 + 6,29$

Klasy złożoności obliczeniowej problemów decyzyjnych

Widome jest, że nie zawsze problemy kombinatoryczne można rozwiązać za pomocą algorytmów wielomianowych. W dalszej części przedstawiono podstawowe definicje i opisy klas złożoności obliczeniowych problemów decyzyjnych i optymalizacyjnych – szersze informacje można znaleźć w uznanej literaturze [6, 33, 22].

Klasa **P** składa się z tych problemów decyzyjnych, które można rozwiązać w czasie co najwyżej wielomianowym za pomocą deterministycznej maszyny Turinga (DTM). Dokładniej mówiąc, są to problemy, które mogą być rozwiązane w czasie co najwyżej $O(n^k)$ dla pewnej stałej k , gdzie n jest wielkością wejścia

problemu.

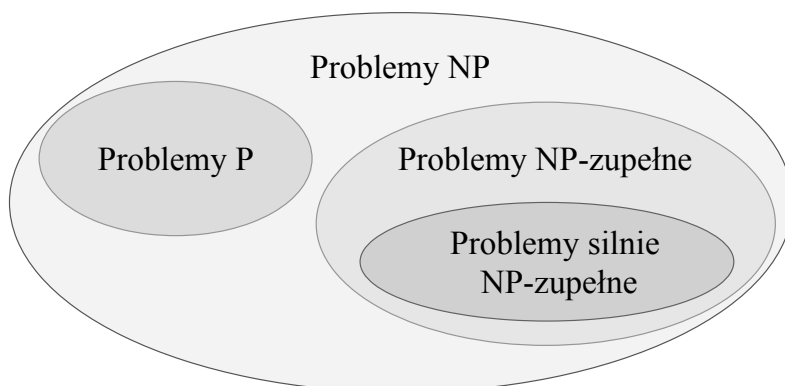
Klasa **NP** składa się z tych problemów decyzyjnych, które są “weryfikowalne” w czasie wielomianowym. Wszystkie te problemy decyzyjne mogą być rozwiązane przez niedeterministyczną maszynę Turinga w czasie co najwyżej wielomianowym. Przykładowo, w problemie cyklu Hamiltona, biorąc pod uwagę skierowany graf $G = (V, E)$, weryfikacją byłaby sekwencja $(v_1, v_2, v_3, \dots, v_{|V|})$ z $|V|$ wierzchołków. Łatwo jest sprawdzić w czasie wielomianowym, że $(v_i, v_{i+1}) \in E$ dla $i = 1, 2, 3, \dots, |V| - 1$ oraz że $(v_{|V|}, v_1) \in E$.

Każdy problem należący do klasy **P** znajduje się wewnątrz klasy **NP**, ponieważ jeśli problem jest w klasie **P**, to możemy go rozwiązać w czasie wielomianowym. Pojęcie to sformalizujemy w dalszej części tego rozdziału, ale na razie możemy przyjąć, że $P \subseteq NP$. Otwartym pytaniem jest, czy **P** jest właściwym podzbiorem **NP**.

Problem znajduje się w klasie **NPC** (odnosimy się do niego jako do **NP-zupełnego**) jeśli jest w klasie **NP** i jest tak “trudny” do rozwiązania jak każdy inny problem klasy **NP** (tj. wszystkie inne problemy przekształcone do niego poprzez transformację wielomianową). Jeżeli jakikolwiek problem **NP-zupełny** mógłby być rozwiązany w czasie wielomianowym, to każdy problem **NP-zupełny** miałby odpowiadający algorytm wielomianowy. Uważa się, że problemy **NP-zupełne** są trudne do rozwiązania, ponieważ biorąc pod uwagę szeroki zakres problemów **NP**, które zostały zbadane do tej pory (nikt nie odkrył wielomianowego algorytmu do rozwiązania któregośkolwiek z nich) byłoby naprawdę zdumiewające, gdyby wszystkie z nich mogły być rozwiązane w wielomianowym czasie. Nie możemy jednak wykluczyć, że problemy klasy **NP** są w rzeczywistości możliwe do rozwiązania w wielomianowym czasie. Problem silnie **NP-zupełny** to taki, który nawet przy ograniczeniu maksymalnych wartości występujących zmiennych pozostaje **NP-zupełny**.

Złożoność obliczeniowa problemów optymalizacyjnych

Niedeterministyczne problemy wielomianowe trudne (**NP-trudne**) to nieformalnie wszystkie problemy, które są co najmniej tak trudne jak najtrudniejsze w klasie **NP**. Problem π_2 jest **NP-trudny** wtedy i tylko wtedy gdy istnieje **NP-zupełny** problem π_1 , który jest redukowalny wielomianowo do problemu π_2 ($\pi_1 \propto_T \pi_2$). Gdy okaże się, że decyzyjna wersja problemu optymalizacji kombinatorycznej należy do klasy **NP-zupełnej**, wówczas wersja optymalizacyjna jest **NP-trudna**.



Rysunek 2.1. Struktura klas złożoności obliczeniowych kombinatorycznych problemów decyzyjnych

2.2 Algorytmy

Algorytm [22] (w Informatyce) jest specyficznym zestawem instrukcji, tj. dowolną procedurą używaną do rozwiązania problemu, zarówno decyzyjnego, jak i optymalizacyjnego. Można powiedzieć, że jest to program komputerowy napisany w określonym języku programowania. Algorytm rozwiązuje problem decyzyjny π , jeśli może rozwiązać dowolny (wszystkie) przypadek problemu $I \in D_\pi$ (odpowiedź *tak* lub *nie*). Algorytmy dla problemów optymalizacyjnych można zdefiniować w ten sam sposób, z wyjątkiem funkcji celu, która powinna być ekstremalizowana.

Złożoność czasowa

Czas obliczeń (działania) algorytmu rozwiązującego problem zależy od wielkości instancji problemu. Jeżeli funkcja złożoności czasowej algorytmu może być ograniczona wielomianem w zależności od wielkości instancji, taki algorytm nazywamy wielomianowym. Inne algorytmy, dla których funkcja czasu nie może być ograniczona wielomianem, nazywane są algorytmami wykładniczymi. Jeżeli czas obliczeniowy algorytmu może być zdefiniowany przez funkcję wielomianową zależną zarówno od wielkości instancji, jak i od największej liczby z instancji, określamy go mianem algorytmu pseudowielomianowego [6, 33].

Algorytmy rozwiązań optymalnych

Dla problemów optymalizacyjnych istnieją algorytmy, które zapewniają ich dokładne rozwiązanie. Większość podstawowych algorytmów może wykonać testy dla wszystkich instancji I danego problemu π . Jednak w przypadku nie-

których problemów (NP-trudnych) takie podejście jest zbyt złożone (złożoność czasowa jest zbyt duża). Istnieją grupy algorytmów, które mogą zapewnić optymalne rozwiązanie w krótszym czasie – jak np. podejście podziału i ograniczeń (ang. Branch and Bound).

Algorytm podziału i ograniczeń (B&B)

Podejście podziału i ograniczeń jest jedną z najbardziej popularnych metod stosowanych w celu znalezienia optymalnego rozwiązania problemów optymalizacyjnych. W 1960 r. metoda B&B została zaproponowana przez Land i Doig [53]. Ideą rozwiązania problemu optymalizacji jest rozdzielenie go na dwa lub więcej mniejszych problemów (dla danej grupy kandydatów S podzielone na zestawy $(S_1, S_2, \dots)$, których suma pokrywa zbiór S) i kierowane wyszukiwanie rozwiązań wśród mniejszych zestawów. Można zauważyć, że $\min f(x)$ na zbiorze S można zapisać jako $\min v_1, v_2, \dots$, gdzie dla każdego v_i trzeba obliczyć $\min f(x)$ na zbiorze S_i . Krok ten nosi nazwę **podziału**. Tworzona jest struktura drzewiasta (każdy węzeł jest reprezentowany przez podzbiór S_i kompletnego zbioru S). Każdy liść jest reprezentowany przez potencjalne rozwiązanie problemu.

Drugim krokiem (narzędziem) zwanym **ograniczeniem** jest procedura obliczania górnego i dolnego ograniczenia dla danego węzła w drzewie (dla podproblemu S_i). Przy założeniu minimalizacji funkcji celu, dolne ograniczenie w większości przypadków to wartość funkcji f dla aktualnego rozwiązania. Górne ograniczenie to zwykle najlepsza wartość funkcji f spośród poprzednio znalezionych rozwiązań funkcji f .

Główną ideą B&B jest zatrzymanie obliczeń algorytmu dla bieżącej gałęzi, gdy wartość dolnego ograniczenia bieżącego węzła (a) jest gorsza (dla minimalizacji funkcji f oznacza wyższą wartość, dla maksymalizacji oznacza niższą wartość) niż górne ograniczenie dla innego węzła (b). Dlatego też a można bezpiecznie odrzucić z procesu wyszukiwania.

Algorytm kończy pracę po zbadaniu lub odrzuceniu wszystkich węzłów. B&B gwarantuje znalezienie optymalnego rozwiązania, ale nie jest znana czasowa złożoność obliczeniowa – w najgorszym przypadku może być dokładnie taki sam jak w przypadku pełnego przeglądu (sprawdzi wszystkie możliwe opcje), tj. wykładniczo w czasie.

Algorytmy zachłanne

Algorytmy rozwiązujące problemy optymalizacyjne najczęściej polegają na podjęciu szeregu decyzji realizowanych w kolejnych krokach. W każdym kroku wybierana jest jedna z wielu dostępnych możliwości. Algorytm zachłanny to taki, który zawsze dokonuje najlepszego możliwego wyboru w danej chwili, z aktualnym krokiem / iteracją. Takie rozwiązanie nazywamy lokalnym optimum. Lokalnie optymalne rozwiązanie (w ramach jednego kroku) jest szansą, że doprowadzi ono do rozwiązania optymalnego całego problemu [22].

Algorytmy genetyczne

Czasami zdarza się, że obserwacja otaczającego nas świata przyrody (głównie fauny i flory) przynosi interesujące inspiracje do opracowania algorytmów. Takim przykładem mogą być algorytmy symulujące proces ewolucji (darwinowskiej) wśród populacji – tzw. algorytmy genetyczne. Holland [41] był pierwszą osobą, która zaproponowała tę metodę.

Algorytmy genetyczne [35, 90] z powodzeniem stosuje się do rozwiązania wielu różnych problemów kombinatorycznych [1, 103]. W ogólności, algorytm genetyczny działa w następujący sposób:

- Początkowa populacja osobników jest najczęściej losowana i posiada określoną liczebność. Każdy osobnik jest potencjalnym rozwiązaniem problemu. Z każdym osobnikiem związany jest zbiór właściwości / parametrów (zmiennych) nazwanych genami. Połączone geny tworzą chromosom – rozwiązanie.
- Funkcja przystosowania oznacza miarę przystosowania dla wszystkich osobników. Im wyższa wartość miary przystosowania tym dany osobnik jest lepiej dopasowany w populacji. Oznacza to również większe prawdopodobieństwo wyboru danego osobnika do procesu tworzenia nowego pokolenia. W zagadnieniach optymalizacyjnych funkcja przystosowania jest zwykle funkcją celu.
- Operatory genetyczne selekcji, krzyżowania i mutacji służą do tworzenia nowej populacji osobników.
- Operator selekcji służy do symulacji procesu selekcji darwinowskiej. Wybierane są osobniki, które będą brały udział w tworzeniu potomków (nowego pokolenia). Wyższa wartość miary przystosowania zwiększa prawdopodobieństwo wyboru.
- Dzięki operatorowi krzyżowania tworzone jest nowe potomstwo osobników. Osobniki wybrane w procesie selekcji są kojarzone w pary. Losowany jest punkt krzyżowania określający pozycję genu w chromosomie. Następnie w wyniku procesu krzyżowania (wszystkie geny znajdujące się na lewo od punktu krzyżowania są wymieniane między osobnikami danej pary) powstaje para potomków.
- Celem operatora mutacji jest zapewnienie różnorodności osobników populacji. Zgodnie z określonym prawdopodobieństwem mutacji dokonuje się zmiany wartości genu w chromosomie na przeciwną (np. 0 na 1, 1 na 0). Na ogół prawdopodobieństwo wystąpienia mutacji jest bardzo małe.
- W kolejnej nowej iteracji tworzone jest nowe pokolenie osobników w wyniku działania operatorów genetycznych.

- Liczba iteracji jest bardzo często kryterium zatrzymania (np. kolejna iteracja algorytmu bez żadnej poprawy w stosunku do wyniku).
- Jeśli wystąpił warunek zatrzymania wybiera się najlepszego osobnika / chromosom z bieżącej populacji – tego o najwyższej wartości funkcji przystosowania.

2.3 Optymalizacja

Optymalizacja [75, 21] to proces mający doprowadzić do osiągnięcia lepszego wyniku. Od strony matematycznej, czy informatycznej jest maksymalizacją (lub minimalizacją) zadanej funkcji. Funkcja na ogół zbudowana jest z wielu zmiennych i posiada szereg zdefiniowanych ograniczeń. Optymalizacje kombinatoryczna, liniowa i dyskretna definiowane są poprzez przynależności do określonych zbiorów kluczowych dla równania zmiennych.

Optymalizacja kombinatoryczna to żywo rozwijane pole matematyki stosowanej i informatyki. Dzięki łączeniu wielu technik kombinatoryki, teorii algorytmów, czy programowania liniowego, pozwala rozwiązywać problemy optymalizacyjne, prezentowane w najróżniejszych dziedzinach życia. Ogólnie przyjętym celem jest zwiększenie efektywności przy zachowaniu określonych wymagań i utrzymaniu lub nawet zmniejszeniu wymaganych surowców / składników / bytów.

Jednymi z najbardziej znanych problemów optymalizacyjnych są np.:

- Assignment problem – problem przydziału agentów do zadań,
- Closure problem – dotyczy wyboru optymalnych zadań do wykonania (zadania posiadają ograniczenia kolejnościowe),
- Constraint satisfaction problem – problem spełnialności ograniczeń,
- Cutting stock problem – dotyczy podziału materiału na zadane fragmenty, przy minimalizacji ilości odpadu – niewykorzystanego materiału,
- Integer programming – programowanie całkowitoliczbowe (wartości należą do zbioru liczb naturalnych),
- Knapsack problem – problem plecakowy [64],
- Minimum spanning tree – problem minimalnego drzewa rozpinającego – znajdujący zastosowanie np. w kwestii układania kabli teleinformatycznych,
- Nurse scheduling problem – dotyczy układania harmonogramu pracy (grafiku),

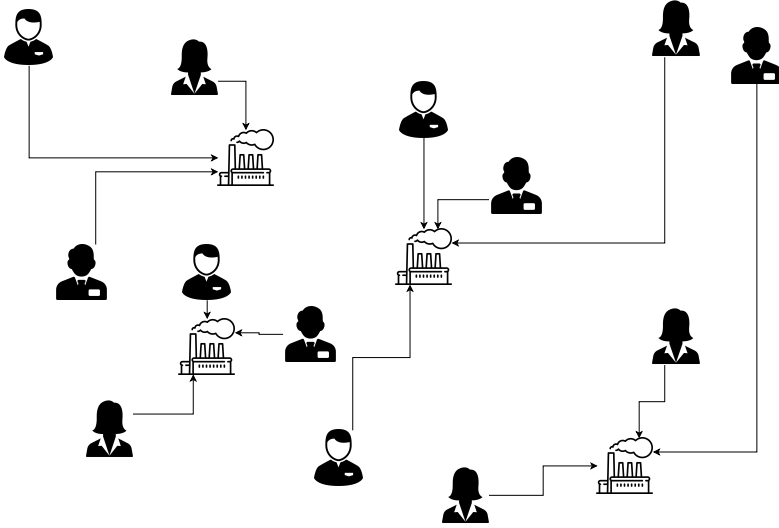
- Traveling salesman problem – problem komiwojażera opisuje sytuację obwoźnego sprzedawcy, który musi odwiedzić wszystkie zadane miejsce, pojawiając się na końcu podróży w punkcie początkowym,
- Vehicle rescheduling problem – jest stosunkowo młodym problemem dotyczącym sytuacji, w której środek transport publicznego ulega awarii i należy dokonać stosownych zmian tras pozostałych pojazdów, aby obsłużyć klientów,
- Vehicle routing problem – problem marszrutyzacji dotyczy wyboru odpowiednich tras dla określonej z góry floty pojazdów, które muszą spełnić oczekiwania klientów,
- Weapon target assignment problem – dotyczy najlepszego doboru uzbrojenia do określonych celów maksymalizując potencjalne szkody (minimalizując współczynnik ich przetrwania) [44, 3].

Poniżej przedstawiono podstawowe informacje dotyczące problemu lokalizacji fabryk (ang. Facility Location Problem, FLP), który może być interesujący ze względu na pewne podobieństwa do najprostszej wersji problemu optymalizacji zakupów internetowych.

Problem lokalizacji fabryki

Główną charakterystyką problemu FLP jest przestrzeń, miara, dane lokalizacji klientów, oraz znane lub nie lokalizacje obiektów (fabryk). Klasyczny FLP ma na celu otwarcie szeregu obiektów / fabryk w dowolnych miejscach przestrzeni (problem ciągły) lub w określonych w podzbiorze lokalizacjach (problem dyskretny). Do otwartych fabryk przydziela się klientów w taki sposób, aby suma kosztów otwarcia i kosztów związanych z odległościami pomiędzy lokalizacjami klientów i fabryk była minimalizowana.

Dyskusje na temat FLPs można znaleźć w wielu publikacjach [50, 28, 87, 66]. Tradycyjny dyskretny FLP jest silnie NP-trudny, podobnie jak problem ISOP (nawet w najprostszej postaci). Rozwiązania dla dyskretnych odmian problemu FLP zawierają się m.in. w programowaniu całkowitoliczbowym, relaksacji Lagrange, relaksacji programowania liniowego, algorytmach genetycznych, podejściach opartych o teoretyczne rozważania grafowe, czy Tabu Search [87]. Wszystkie te podejścia można wykorzystać do rozwiązania problemu ISOP w najprostszej formie, czyli bez jakichkolwiek rabatów, czy funkcji zmieniających ceny produktów, dostaw, itd.



Rysunek 2.2. Problem lokalizacji fabryk – grafika poglądowa

Stratila [96] badał *wklęsły FLP*, którego sformułowanie matematyczne można zapisać jako

$$\min \sum_{i=1}^m \left(\phi_i \left(\sum_{j=1}^n d_j x_{ij} \right) + \sum_{j=1}^n p_{ij} x_{ij} \right) \quad (2.1)$$

$$\text{p.o.} \quad \sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, n, \quad (2.2)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (2.3)$$

gdzie ϕ_i są niemalejącymi funkcjami wklęsłymi. Można zauważyć, że ten problem i problem ISOPWD nie są tożsame, podczas gdy tradycyjny dyskretny FLP jest szczególnym przypadkiem każdego z tych problemów.

Total Quantity Discount Problem

Patrząc na ISOP z uwzględnieniem rabatów cenowych można zauważyć pewne podobieństwa z Total Quantity Discount Problem (TQD) [36]. Aby pokazać podobieństwa, ale także zaznaczyć wyraźne różnice, należy przedstawić matematyczne sformułowanie problemu TQD. Zdefiniujmy G jako zbiór towarów m , indeksowanych według k , a S jako zbiór dostawców n , indeksowanych zgodnie z i . Dla każdego towaru k w G , określmy d_k jako niezbędną ilość towaru k . Z każdym dostawcą i w S kojarzymy ciąg $Z_i = \{0, 1, \dots, \max_i\}$, indeksowany według j . Ponadto, dla każdego dostawcy $i \in S$ i przedziału $j \in Z_i$, l_{ij} , u_{ij} określają od-

powiednio minimalną i maksymalną liczbę towarów, które muszą być zamówione u dostawcy i , aby móc być w dostawie w przedziale j . Wreszcie, dla każdego dostawcy $i \in S$, dla każdego przedziału $j \in Z_i$ i dla każdego towaru $k \in G$, niech c_{ijk} będzie ceną towaru k zakupionego od dostawcy i w przedziale j -tym.

Podobieństwa pomiędzy ISOP z rabatami cenowymi a problemem TQD można zauważyć, jeśli przyjmiemy, że kupowane produkty N przedstawimy jako towary G (d_k to ilość towaru k), a sklepy M jako dostawców S . Funkcja rabatująca f_i dla sklepu i będzie powiązana z ciągiem interwałów Z_i dla dostawcy i . Cena produktu c_{ij} dotyczy j -tego produktu zakupionego ze sklepu i i może być pokazana jako cena p_{ik} za jeden element towaru k zakupionego od dostawcy i . Funkcja rabatująca dla sklepu i zastosowana do ceny produktu $f_i(c_{ij})$ powinna być traktowana jako p_{ijk} – cena za jeden element towaru k zakupiony od dostawcy i w przedziale j -tym. Jednakże ISOP obejmuje wedle swej definicji także koszty wysyłki, które są specyficzne dla każdego sklepu. Ta funkcja sprawia, że ISOP staje się bardziej rozbudowanym problemem. Kolejne ewolucje problemu ISOP powodują, że jest jeszcze bardziej skomplikowany i trudniejszy do rozwiązania – aczkolwiek ciekawszy i coraz bliższy rzeczywistym wymaganiom stawianym przez klientów sklepów internetowych.

Warto zauważyć, że wersja decyzyjna problemu TQD jest silnie NP-zupełna. Ponadto dla problemu TQD nie istnieje wielomianowy algorytm aproksymacji ze stałym współczynnikiem najgorszej sytuacji (chyba że $\mathcal{P} = \mathcal{NP}$). Więcej informacji na temat wielu wariantów TQD znajdziemy w literaturze (np. dopuszczalnych opóźnień w płatnościach [51], algorytmów [61], czy innych [36, 68, 70]).

Problem plecakowy

Problem plecakowy jest jednym z najbardziej znanych problemów NP-trudnych. Pewne podobieństwa do ISOP (bardzo ogólne) zachęcają do zdefiniowania i krótkiego przedstawienia problemu plecakowego. W niniejszym fragmencie prezentacja problemu plecakowego została przedstawiona na podstawie książki autorstwa Martello i Toth [64].

Bardzo często opisując ideę problemu plecakowego przedstawia się przykład podróży autostopowicza. Podróżnik ma ze sobą plecak i musi do niego spakować najbardziej potrzebne rzeczy. Plecak ma oczywiście określoną pojemność i nie można włożyć do niego wszystkich dostępnych przedmiotów.

Problem plecakowy (odpowiadający na pytanie, co spakować do plecaka, aby uzyskać najlepszy wynik) można zdefiniować matematycznie w następujący sposób:

- Istnieje n obiektów o numeracji od 1 do n . Istnieje wektor binarny (1-0) dla zmiennych x_j ($j = 1, \dots, n$). Znaczenie jest następujące:

$$x_j = \begin{cases} 1 & \text{jeśli } j \text{ jest wybrany;} \\ 0 & \text{jeśli } j \text{ nie jest wybrany.} \end{cases}$$

- p_j – wartość przedmiotu j (im wyższa wartość, tym lepiej),
- rozmiar plecaka – c ,
- rozmiar obiektu j oznaczamy jako w_j .

Celem jest spakowanie przedmiotów pozwalających osiągnąć ich maksymalną łączną wartość – maksymalizacja *funkcji celu*

$$\text{maksymalizacja (maks)} \quad \sum_{j=1}^n p_j x_j \quad (2.4)$$

Ograniczeniem będzie wielkość plecaka

$$\text{przy ograniczeniach (p.o.)} \quad \sum_{j=1}^n w_j x_j \leq c. \quad (2.5)$$

Przykłady zastosowań są prawie nieograniczone. Obrazując problem z tematyki inwestycji i operacji biznesowych można przyjąć, że zawartość portfela wynosi c dolarów. Możemy kupować spośród j instrumentów finansowych. Każdy instrument kosztuje w_j , a oczekiwany zysk wynosi p_j . Jak widać, istnieje wiele sytuacji, które możemy opisać za pomocą modelu matematycznego problemu plecakowego.

Dla lepszego zobrazowania problemu i jego rozwiązania można się posłużyć następującym przykładem:

Nasz plecak ma rozmiar 15. Możemy wybierać spośród 5 czekoladowych batonów o rozmiarach 12, 4, 2, 1, 1, 1. Wartości (w zależności od rodzaju użytego ziarna kakaowego) w kawałeczkach są następujące: 14, 10, 3, 1, 2. Które przedmioty powinniśmy spakować, aby je sprzedać za maksymalną kwotę? Możemy oczywiście

Tabela 2.3. Przykład problemu plecakowego

n	1	2	3	4	5
w	12	4	2	1	1
p	14	10	3	1	2
c	15				

zbadać wszystkie możliwe wektory x i wybrać najlepszą opcję na końcu (spełniającą warunek). Liczba takich wektorów wynosi niestety 2^n (wzrost wykładniczy). Istnieje jednak wiele określonych algorytmów, które wymagają znacznie mniejszej ilości obliczeń. Niektóre z nich mogą dostarczyć optymalnych rozwiązań, inne bliskie optymalnym (heurystyka), ale za to w krótszym czasie.

Istnieje wiele różnych rodzajów problemu plecakowego. Poniżej zaprezentowano kilka najpopularniejszych. Przedstawiono dwie główne grupy problemów

– problemy z jednym i wieloma plecakami. Pierwszy typ charakteryzuje się istnieniem jednego pojemnika (plecaka), a drugi zawiera więcej niż jeden pojemnik (plecak). Dla zachowania spójności z literaturą skorzystano z oryginalnych, angielskich nazw problemów.

0-1 Knapsack Problem (pojedynczy) jest prototypem dla wszystkich innych problemów plecakowych.

$$\text{maks } \sum_{j=1}^n p_j x_j \quad (2.6)$$

$$\text{p.o. } \sum_{j=1}^n w_j x_j \leq c, \quad (2.7)$$

$$x_j = \{0, 1\}; j = 1, \dots, n. \quad (2.8)$$

Bounded Knapsack Problem (pojedynczy) jest uogólnieniem 0-1 KP przy założeniu, że dla każdego j ($j = 1, \dots, n$) dostępne są b_j pozycje ($b_j \leq c/w_j$). Zysk p_j i waga w_j pozostają bez żadnych zmian.

$$\text{maks } \sum_{j=1}^n p_j x_j \quad (2.9)$$

$$\text{p.o. } \sum_{j=1}^n w_j x_j \leq c, \quad (2.10)$$

$$0 \leq x_j \leq b_j; j = 1, \dots, n, \quad (2.11)$$

$$x_j \text{ integer}; j = 1, \dots, n. \quad (2.12)$$

Przypadek, w którym $b_j = \infty$ dla wszystkich j nazywamy **Unbounded Knapsack Problem**.

Subset-Sum Problem (pojedynczy) jest przypadkiem, w którym $p_j = w_j$ dla każdego j , który to często występuje w praktycznych zastosowaniach. Problemem jest znalezienie sumy (podzbioru) wag, która jest najbliższa pojemności plecaka.

$$\text{maks } \sum_{j=1}^n w_j x_j \quad (2.13)$$

$$\text{p.o. } \sum_{j=1}^n w_j x_j \leq c, \quad (2.14)$$

$$x_j = \{0, 1\}; j = 1, \dots, n. \quad (2.15)$$

Change-Making Problem (pojedynczy) pojawiający się, gdy $p_j = 1$ dla każdego j ($j = 1, \dots, n$). W takim przypadku ograniczenie pojemności plecaka

zmienia się z nierówności na równość. Nazwa problemu przypomina o sytuacji kasjerki w supermarkecie. Celem jest wydanie reszty c przy użyciu maksymalnej liczby monet.

$$\text{maks } \sum_{j=1}^n x_j \quad (2.16)$$

$$\text{p.o. } \sum_{j=1}^n w_j x_j = c, \quad (2.17)$$

$$0 \leq x_j \leq b_j; j = 1, \dots, n, \quad (2.18)$$

$$x_j \text{ integer}; j = 1, \dots, n. \quad (2.19)$$

Dodanie warunku dotyczącego wielu elementów każdego rodzaju $b_j = \infty$ dla wszystkich j sprawia, że problem określamy jako **Unbounded Change-Making Problem**.

0-1 Multiple Knapsack Problem (wielokrotny) jest ważnym uogólnieniem pojedynczego problemu 0-1 KP. Pojawiają się tutaj różne wersje pojemników o pojemności c_i ($i = 1, \dots, m$).

$$\text{maks } \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} \quad (2.20)$$

$$\text{p.o. } \sum_{j=1}^n w_j x_{ij} \leq c_i, \quad (2.21)$$

$$\sum_{i=1}^m x_{ij} \leq 1, \quad (2.22)$$

$$x_j = \{0, 1\}, \quad (2.23)$$

$$j = 1, \dots, n, i = 1, \dots, m. \quad (2.24)$$

Generalized Assignment Problem (wielokrotny) jest bardziej skomplikowany niż wyżej wymienione. Rozważmy sytuację, w której zysk p i waga w przedmiotów różnią się w zależności od pojemnika, którego używamy. Niech p_{ij} będzie zyskiem z przedmiotu j umieszczonego w kontenerze i , a w_{ij} odpowiednio

jego wagą.

$$\text{maks} \quad \sum_{i=1}^m \sum_{j=1}^n p_{ij} x_{ij} \quad (2.25)$$

$$\text{p.o.} \quad \sum_{j=1}^n w_{ij} x_{ij} \leq c_i, \quad (2.26)$$

$$\sum_{i=1}^m x_{ij} \leq 1, \quad (2.27)$$

$$x_{ij} = \{0, 1\}, \quad (2.28)$$

$$j = 1, \dots, n, i = 1, \dots, m. \quad (2.29)$$

Warto w tym momencie zauważyć, że zaprezentowany powyżej problem nie jest ściśle klasycznym problemem plecakowym, a jego odmianą (podproblemem), który to odgrywa ważną rolę ze względu na opracowane algorytmy. Jest on również bardzo podobny w sformułowaniach matematycznych do klasycznej rodziny problemów plecakowych.

Należy zaznaczyć, że niniejszy fragment dotyczący problemów plecakowych jest skrótowy i nie wyczerpuje licznych definicji i formuł. Wydaje się natomiast, że jest wystarczający, aby z odpowiednimi podwalinami kontynuować wprowadzenie do tematu optymalizacji zakupów internetowych.

2.4 Szeregowanie zadań

Szeregowanie zadań (ang. scheduling) jest niewątpliwie ważnym obszarem badań operacyjnych, czy w ogólności informatyki. Problemy związane z szeregowaniem mogą być zasadniczo opisane jako odpowiedni podział zasobów w czasie, w celu realizacji zespołu zadań, które są częścią procesów [12]. Każde zadanie wymaga do jego wykonania określonego zasobu. Charakter zasobów, zadania, powiązania i relacje między nimi można zdefiniować na wiele różnych sposobów. W przeważającej części, szeregowanie zadań jest motywowane zastosowaniami w przemyśle oraz zarządzaniu operacjami w usługach. Przykładami mogą być: planowanie transportu, linie produkcyjne w fabrykach, planowanie wydarzeń, ustalanie ogólnego harmonogramu pracy w najróżniejszych dziedzinach, transport, trasa przejazdu pojazdów, plan pracy oraz wiele więcej. Jeśli spojrzymy na problemy związane z planowaniem w sposób ogólny, to widzimy, że można je zaobserwować niemal na każdym kroku w otaczającej nas rzeczywistości.

Gospodarka elektroniczna [88] (ang. Electronic Commerce, e-commerce) jest stosunkowo młodą specjalnością informatyki dotyczącą handlu przy wykorzystaniu medium Internetu (o samym e-commerce więcej napisano w rozdziale 3). Mnogość zagadnień mieszczących się w zakresie gospodarki elektronicznej, jak

i różnorodność pojawiających się problemów, stanowią doskonale pole do zastosowania wielu innych, bardziej rozwiniętych i szerzej zbadanych specjalności informatyki.

2.5 Oznaczenia, symbole, akronimy

W niniejszym fragmencie zdefiniowano i przedstawiono znaczenie używanych w książce greckich i rzymskich liter, a także symboli, skrótów, akronimów i innych oznaczeń.

Symbole

$=$	równość (jest równe)
$\neq$	nierówność (nie jest równe)
$>$	ściśła nierówność (jest większe niż)
$<$	ściśła nierówność (jest mniejsze)
$\geq$	nierówność (jest większe lub równe)
$\leq$	nierówność (jest mniejsze lub równe)
$+$	dodawanie (plus)
$-$	odejmowanie (minus)
$*$	mnożenie (razy)
$\sqrt{}$	pierwiastek kwadratowy
$\bar{x}$	średnia
$ \dots $	wartość bezwzględna
$:=$	definicja (jest zdefiniowane jako)
$\forall$	kwantyfikator ogólny
$\exists$	kwantyfikator szczegółowy
$\{, \}$	zakres zbioru
$\emptyset$	zbiór pusty
$\in$	przynależność
$\notin$	przynależność (negacja)
$\subseteq$	podzbiór
$\supseteq$	nadzbiór
$\cup$	suma zbiorów
$\cap$	iloczyn zbiorów
$\setminus$	różnica zbiorów
$\mathbb{N}$	liczby naturalne
∞	nieskończoność
$(,)$	przedział otwarty
$\sum$	suma
$\prod$	produkt
$O(n)$	notacja złożoności obliczeniowej (tzw. duże-O)

Litery greckie

α	(alpha)
ϵ	(epsilon) bardzo mała wartość dodatnia
δ	(delta) operator różnicowania
ϕ	(phi) funkcja skumulowanego rozkładu dla rozkładu normalnego
π	(pi)
ρ	(rho) gęstość

Litery rzymskie

F	funkcja
x, p, c	zmienna (mała litera)
x_{ij}	dolny indeks zmiennych
$S, L, \dots$	zbiory

Skróty i akronimy

NP	Nondeterministic Polynomial (niedeterministyczny wielomianowy)
P	Polynomial (wielomianowy)
DTM	Deterministic Turing Machine (deterministyczna maszyna Turinga)
NDTM	Non-deterministic Turing Machine (niedeterministyczna maszyna Turinga)
IS	Internet shopping
ISOP	Internet Shopping Optimization Problem
FLP	Facility Location Problem (problem lokalizacji fabryk)
X3C	Exact Cover by 3-sets (problem dokładnego pokrycia przez 3 zbiory)
RS	Recommender Systems (system rekomendujący)
PCS	Price Comparison Site (porównywarka cen)
max	maximization (maksymalizacja)
maks	maksymalizacja
min	minimization (minimalizacja)

Rozdział 3

Internet i handel elektroniczny

Niniejszy rozdział zawiera krótkie wprowadzenie dotyczące początków i ewolucji sieci Internet. W dalszej części opisano zasady handlu elektronicznego, załączając sporą porcję danych statystycznych. Dzięki tym informacjom łatwo można zauważyć ogrom i siłę obecnego rynku e-commerce.

3.1 Internet

Przedstawioną poniżej skrótowną i ogólną drogą ewolucji sieci Internet zakończy okres przełomu wieków, kiedy Internet był na tyle dojrzały, że można zacząć świadomie wkraczać w świat z pojęciem e-commerce.

Prawie 60 lat temu, w 1960 roku Licklider [56] zaprezentował swoją ideę globalnej sieci połączonych komputerów. Opisał przy tym swoje pomysły i prognozy na symbiozę między człowiekiem a komputerem. W tym samym czasie Paul Baran, który pracował na rzecz korporacji (RAND)¹ zajmował się rozwojem rozproszonych sieci komunikacyjnych, które mają własności przetrwania nawet gdy zostaną częściowo zniszczone (przewidywane wówczas III Wojna Światowa). Pierwsze wyniki badań zostały przedstawione amerykańskim siłom powietrznym w 1961 roku. W latach 1960-1962 Baran opublikował 11 obszernych opracowań na ten temat. W ramach podsumowania, w 1964 r. ukazała się książka zatytułowana *O komunikacji rozproszonej* [7]. Autor opisywał i badał różne architektury rozproszone sieci komunikacyjnej, zdolności ich przetrwania, czy możliwości i własności pakietowych sieci komunikacyjnych (np. scentralizowanych, zdecentralizowanych, rozproszonych).

¹Rand Corporation (RAND), <https://www.rand.org/>

Kilka lat później, w 1969 roku rozgrywa się prawdziwa historia Internetu. Agencja o nazwie ARPA (Advanced Research Project Agency), przekształcona później w Agencję Zaawansowanych Projektów Badawczych Obrony, DARPA², zajmującą się koordynowaniem badań naukowych na potrzeby wojska, odpowiada za realizację pierwszych trzech węzłów sieci Arpanet. Eksperyment podążał za koncepcjami Paula Barana i miał na celu zbudowanie zdecentralizowanej sieci komputerowej zdolnej do działania nawet po zniszczeniu części jej architektury. Do końca 1971 roku do sieci podłączono 15 węzłów [38].



Rysunek 3.1. Schematyczna struktura sieci szkieletowej oplatającej cały glob ziemski (źródło (za zgodą): www.pcformat.pl, Krwiobieg internetu, PC Format 08/2016)

W 1982 r. standaryzowano pakiet protokołu internetowego (TCP/IP), co pozwoliło na ogólnosiwiatowy gwałtowny rozwój wzajemnie połączonych sieci. Rok później przyjęto protokoły TCP/IP w sprawie wspólnych norm wojskowych. Był to czas, w którym zaczęto powszechnie używać terminu Internet (połączona sieć). Publiczne korzystanie z Internetu rozpoczęło się w połowie 1989 r. (ARPANET został zlikwidowany w 1990 r.). Pierwsze międzykontynentalne szybkie połączenie T1 zostało zainstalowane w 1990 roku pomiędzy Uniwersytetem Cornell, a CERN. Krok ten umożliwił znacznie lepszą komunikację niż ta oferowana przez satelity, które wcześniej służyły do komunikacji. Pod koniec 1990 r. Tim Berners-Lee stworzył pierwsze niezbędne narzędzia do prezentacji dokumentów hipertekstowych online. Był to protokół HyperText Transfer Protocol (HTTP) (jeszcze nie w pełni opublikowana wersja – 0,9), HyperText Markup Language (HTML) i pierwsza przeglądarka WWW (World Wide Web). Pierwsze strony

²Defense Advanced Research Projects Agency (DARPA), <https://www.darpa.mil/>

www opisujące projekt zostały umieszczone na serwerze www i były dostępne przy użyciu pierwszego oprogramowania serwera HTTP.

Internet rozwijał się coraz szybciej. Ogromne ilości informacji i wiedzy dostępnej online, handel, rozrywka i sieci społecznościowe stymulowały bardzo dynamiczny wzrost. Według ostatnich raportów [47, 46] ponad połowa populacji (54,4%) ma dostęp do Internetu. Zauważyć należy wzrost liczby użytkowników do obecnych 4 156 mln w 2017 r. z 14 mln w 1993 r. (co daje prawie 300-krotny wzrost). Platforma internetowa otworzyła wiele możliwości dla nowych działań. Niewątpliwie, ważnym aspektem w tym kontekście, jest handel elektroniczny.

3.2 Handel elektroniczny

Dyskutując o początkach handlu elektronicznego (ang. e-commerce) należy się cofnąć w czasie odrobinę dalej niż by się mogło początkowo wydawać. Poniżej zaprezentowano listę wydarzeń, ważnych z punktu widzenia rozwoju e-commerce [98].

- 1960

Wprowadzenie w życie konceptu elektronicznej wymiany danych (ang. Electronic Data Interchange, EDI) umożliwiło zastąpienie pewnych tradycyjnie papierowych dokumentów nowymi w formie elektronicznej. Szczególnymi znanymi dziś doskonale przykładami są choćby transakcje elektroniczne (internetowe), czy dokumenty sprzedażowe – faktury. Przedstawiony koncept na skalę masową zaczął być realizowany wiele lat później, aczkolwiek już w latach 60' i 70' największe firmy rozpoczęły wykorzystywanie EDI.

- 1979

Michael Aldrich jest bezapelacyjnie nazywany ojcem handlu elektronicznego. Jak to zwyczajowo się zdarza, pomysł pojawił się zupełnie nagle – podczas rodzinnych zakupów. Aldrich wprowadził w życie swój pomysł realizacji zakupów na odległość poprzez proste podłączenie telewizora z domowym komputerem. W ten sposób prowadził pierwszą sprzedaż online. Sposób sprzedaży i realizacji był bardzo daleki od ideału, ale bezdyskusyjnie stanowi podwaliny pod dzisiejszy rynek e-commerce.

- 1981

Powstaje pierwszy system B2B (Business to Business). Thomson Holidays postanowił połączyć w grupę zespół 66 angielskich agentów podróży. Dzięki temu możliwa była analiza i pozyskiwanie danych online (wśród dostępnych broszur). Pozwoliło to bardzo szybko realizować zamówienia klientów.

- 1982

Powstaje Minitel – usługa sieciowa / system wideo-tekstowy (przypominający niewielki zintegrowany komputer) działający online (poprzez linię

telefoniczną – France Telecom i La Poste). Rozwiązanie zyskało dużą popularność i z pewnością należy je uznać za najbardziej udaną usługę online przed rozpowszechnieniem się Internetu. Możliwości systemu (z dzisiejszego punktu widzenia) były bardzo ograniczone, ale na owe czasy użytkownicy mieli już dostęp do zakupów online, sprawdzania książki telefonicznej, czy prowadzenia czatów.

- 1984

W tym roku firma Tesco wprowadziła pierwszy system zakupów online B2C (Business-to-Consumer). Pojawił się pomysł realizowania ich poprzez wirtualny koszyk zakupów (Online Shopping Basket), która to nazwa obowiązuje do dziś (często określana po prostu jako *koszyk* lub *koszyk zakupów*). W tym samym roku CompuServe zapowiada projekt Electronic Mall (bardzo przypominający zamysłami obecnie działające sklepy internetowe).

- 1987

Powstaje SWREG – produkt Digital River MyCommerce pozwalający na sprzedaż oprogramowania (lub licencji do takowego) online. System, w zmodyfikowanej formie, działa do dzisiaj.

- 1990

Zaprezentowano pierwszą w historii przeglądarkę internetową Dynatext, będącą integralną częścią systemu WorldWideWeb / WWW / W3 (nazwa została później zmieniona na Nexus) działającą w architekturze klient-serwer, opierającą zasadę działania na hipertekście (plus odnośniki – tzw. hiperłącza). Użytkownik czytał / przeglądał dokument, a hiperłącza pozwalały na dostęp (odniesienie) do kolejnych, powiązanych dokumentów. Autorem jest brytyjski naukowiec sir Timothy Berners-Lee, a projekt początkowo miał być niewielkim programem wykorzystywanym przez CERN (fr. Organisation Européenne pour la Recherche Nucléaire, pol. Europejska Organizacja Badań Jądrowych).

- 1991

NSF (National Science Foundation) umożliwia dostęp do sieci NSFNET (szkielet sieci Internet, kolejna generacja sieci ARPANET) dla zastosowań komercyjnych. Jest to jeden z najważniejszych kroków do obecnego e-commerce. W tym samym roku powstaje Gopher, system informacyjny (z założenia dla kampusów uczelnianych) integrujący różne protokoły (jak np. FTP, telnet, WAIS).

- 1992

Opublikowano rewolucyjną (w założeniach) książkę opisującą przyszłe założenia e-commerce i kształtu Internetu przyszłości. Książkę zatytułowano *Future Shop: How New Technologies Will Change the Way We Shop and what We Buy* [94].

- 1994

Netscape uruchamia pierwszy certyfikat szyfrujący, SSL (używany do dzisiaj). Był to niewątpliwie kamień milowy ku dzisiejszej strukturze rynku handlu elektronicznego. Czynniki zaufania (na którego wzrost wpływał bez wątpienia certyfikat SSL) do dziś pozostaje jednym z najważniejszych aspektów zakupów internetowych. W tym samym roku zostaje uruchomiony serwis Yahoo!, późniejszego giganta i potentata rynku e-commerce i Internetu.

- 1995

W roku 1995 NSF wprowadził opłaty za rejestrację domen. Od tego momentu liczba zarejestrowanych nazw zaczęła gwałtownie rosnąć od 12 tys. w roku 1995 do ponad dwóch milionów w roku 1998.

- 1998

Powstaje PayPal, który początkowo działa w innej formie niż ta, którą obecnie znamy. Google dołącza do świata e-commerce. Yahoo! tworzy swój sklep Yahoo! Stores.

- 2000

W ostatnich kilkunastu latach rozwój e-commerce przebiegał bardzo szybko. Pojawiały się setki nowych sklepów czy wiele platform aukcyjnych. Spora część z nich działa do dzisiaj. Nowe metody płatności i dostarczania towarów ułatwiły handel w sieci.

Można powiedzieć, że handel elektroniczny (e-handel) jest jedną z najszybciej rozwijających się dziedzin informatyki (korzystając garściami z badań operacyjnych, czy optymalizacji kombinatorycznej). Warto jednak zauważyć, że handel elektroniczny jest olbrzymią hybrydą zbudowaną w oparciu o dodatkowe obszary wiedzy, takie jak logistyka, gospodarka i nauki społeczne. Handel elektroniczny to branża, która koncentruje się na sprzedaży oraz kupowaniu produktów i usług online na stronach internetowych [39, 59, 102]. E-commerce w kategorii *biznes-konsument* (B2C) jest jednym z kluczowych rodzajów działalności gospodarczej oferowanej przez Internet.

Z biegiem lat liczba dokonywanych przez użytkowników zakupów online stale rośnie. Według badania przeprowadzonego przez comScore dla UPS [20], w 2013 roku osiągnęła zdumiewającą wartość 243 mld USD. Kolejne badanie przeprowadzone przez Cisco z udziałem respondentów z USA, Wielkiej Brytanii, Chin, Brazylii i Meksyku [31] wykazało, że ośmiu z dziesięciu konsumentów z rynków wyżej wymienionych krajów robi zakupy online, przy czym wiodącą pozycję zajmują grupy demograficzne w wieku 18-29 i 30-39 lat. Najnowsze badania przeprowadzone w Europie [27] wskazują, że sprzedaż na europejskim rynku e-commerce osiągnęła w 2017 r. zawrotną kwotę 602 mld € podwajając swą wartość w zaledwie 5 lat.

Tendencja dynamicznego wzrostu utrzyma się zapewne w przyszłości dzięki pojawieniu się nowych rynków internetowych, zwłaszcza w krajach azjatyckich, takich jak Indie [101], Indonezja [104], Tajwan [107], Jordania [2], Iran [25], Malezja [5], itd. Rynki te ze względu na dużą liczbę ludności i stosunkowo niski wskaźnik penetracji handlu elektronicznego, oferują bardzo interesujący potencjał wzrostu. Głównymi zaletami (dla klienta) płynącymi z zakupów online są m.in. brak konieczności dojeżdżania do centrum handlowego lub sklepu detalicznego, istnienie szerszej oferty produktowej, duża konkurencyjność prowadząca do niższych cen, pojawienie się płatności mobilnych, które oferują dodatkowy komfort i bezpieczeństwo postrzegane pozytywnie przez klientów [92]. Dzięki znacznym usprawnieniom w obsłudze dostaw, stało się możliwe zamawianie łatwo psujących się produktów, takich jak żywność i artykuły spożywcze, co może ostatecznie doprowadzić do lepszego zarządzania tego rodzaju produktami. Można również domniemywać, że zmiana na bardziej wydajne magazyny z lepszą technologią przechowywania żywności przełoży się na zmniejszenia marnotrawstwa tego rodzaju zasobów w niedalekiej przyszłości [110].

Jeśli chodzi o sprzedającego, do głównych korzyści ekonomicznych należy zaliczyć fakt, że może on znacznie obniżyć koszty związane z leasingiem i zatrudnianiem pracowników. Ze względu na otwartość działania platform internetowych istnieje możliwość dotarcia do znacznie szerszego grona odbiorców niż ma to miejsce w sklepach tradycyjnych (ang. brick-and-mortar shops).

Udogodnienia platform e-commerce działające na korzyść zarówno klientów, jak i sprzedawców, sprawiły, że liczba sklepów internetowych stale rośnie. Jeśli chodzi o sprzedawców, wygodniejsze jest przyłączenie się do istniejącego rynku online (większe sklepy zrzeszające pod swoim logo mniejsze podmioty), lub wdrożenie własnej strony internetowej, na której mogą oferować swoje produkty. Należy jednak zauważyć, że ta wygoda dla sprzedawcy może stać się niedogodnością dla klienta. Jak wspomniano wcześniej, większa konkurencja oznacza niższą cenę, ale oznacza również, że klient może zostać przytłoczony ogromną liczbą ofert i dodatkowych możliwości, co bardzo utrudnia uzyskanie najlepszej możliwej oferty. Problem ten został zamodelowany jako problem optymalizacji zakupów internetowych – przez Blazewicza et al. [13], a zainspirowany wcześniejszymi pomysłami Wojciechowskiego i Musiała [105]. Co więcej, warto zwrócić uwagę na podejście oparte na współpracy, które grupa kupujących może realizować jako strategię zakupów zbiorowych przy globalnie niższych kosztach [69].

Innym problemem dla większości użytkowników, wynikającym z handlu elektronicznego, jest niespójność czy nawet niejasność przepisów, które ułatwiają nieuczciwym lub złośliwym sprzedawcom wdrażanie własnych sklepów internetowych. Wywołują one poczucie braku bezpieczeństwa, którego doświadczają kupujący np. podczas podawania swoich danych osobowych i informacji finansowych w celu dokonania transakcji. Inną niepokojącą dla klienta sytuacją jest niemożność sprawdzenia przedmiotów przed dokonaniem zakupu [52]. Oczywiście ta sytuacja poprawia się z każdym rokiem poprzez wprowadzanie kolejnych regu-

lacji. Niemniej jednak cały czas brakuje skuteczniejszych mechanizmów obrony przed oszustami. Niestety regularnie słyszy się o fałszywych sklepach, które wyłudziły (pobrały) pieniądze za nigdy niedostarczone towary. Należy spokojnie i przemyślanie dokonywać zakupów, a także zwracać szczególną uwagę na bardzo korzystne oferty.

Mając na uwadze powyższe wiemy, że klienci zazwyczaj wolą poświęcać lepsze oferty (tańsze), które są prezentowane przez mniej renomowanych sprzedawców, aby kupić droższy produkt, oferowany przez znany i zaufany sklep z lepszym prestiżem wśród innych konsumentów [108, 45]. Biorąc pod uwagę takie zachowania zaobserwowane u większości klientów sklepów internetowych, problem optymalizacji zakupów internetowych został rozbudowany i zaprezentowano jego ciekawą i nową odmianę. Wersję, gdzie analizie poddane zostały właśnie takie aspekty jak zaufanie, prestiż i wiarygodność (patrz rozdział 7).

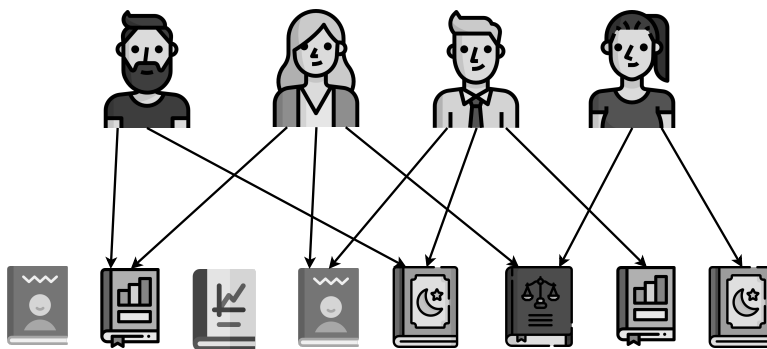
3.3 E-commerce w Polsce

Wiele bardzo ciekawych i rzetelnych informacji dotyczących rynku e-commerce w Polsce można znaleźć w najnowszym raporcie *E-commerce w Polsce* [34]. Poniżej przytoczono kilka najciekawszych i najistotniejszych faktów.

- Ponad połowa (dokładnie 54%) badanych internautów kupuje produkty online. Dane dotyczą osób we wszystkich grupach wiekowych, z różnym poziomem wykształcenia, posiadających różne dochody, czy zamieszkujących miejscowości różnej wielkości. Badana grupa jest reprezentatywna dla Polski.
- Świadomość internetowa (dotycząca zakupów) kupujących jest bardzo wysoka. Posiadają oni sporą wiedzę dotyczącą marek i serwisów internetowych e-commerce.
- Polacy zdecydowanie preferują robienie zakupów w sklepach zlokalizowanych na terenie Polski. Zakupy realizowane w zagranicznych serwisach są nadal rzadkością (statystycznie). Co ciekawe, zakupy w tych sklepach deklarują osoby z wysokimi dochodami (ponad 5000 zł miesięcznie), wykształceniem wyższym i niskim wiekiem.
- Zdecydowanie najpopularniejszymi kategoriami kupowanych produktów są odzież (dodatki, akcesoria) i książki, płyty, filmy.
- Większość kupujących dosyć aktywnie uczestniczy w procesie zakupów. Dokonują poszukiwań produktów, czytają opinie na ich temat, poszukują informacji dotyczących marek.

- Czynnikiem, które najbardziej zachęcają do zakupów są dostępność, wygoda realizacji zakupów, czy korzystniejsza cena w porównaniu ze sklepami stacjonarnymi.
- Należy zauważyć, iż bardzo ważnym czynnikiem zachęcającym do zakupów online jest możliwość zwrotu towaru bez podania przyczyny w ciągu 14 dni od jego otrzymania. Dzięki temu można towar obejrzeć, czy nawet przymierzyć odzież i dopiero wtedy podjąć finalną decyzję o pozostawieniu towaru.
- Kupujący jako jeden z głównych problemów określają wysokie koszty dostawy, czy przedłużający się czas realizacji i dostarczania zamówień.
- Jeśli chodzi o znane marki, niekwestionowanym liderem sprzedaży online w Polsce pozostaje Allegro. Następnie należy wymienić Olx, Zalando, eBay i Ceneo. Chętnie też zaglądamy do internetowych perfumerii, sklepów RTV AGD, księgarni czy aptek.

3.4 Systemy rekomendujące



Rysunek 3.2. System rekomendujący – grafika poglądowa

W ogólnym znaczeniu systemy rekomendujące (ang. Recommender Systems, RS) [89] są zbiorem metod i oprogramowaniem służącym udzielaniu sugestii dotyczących wyboru przedmiotów / obiektów właściwych dla użytkownika – najbardziej odpowiadających jego potrzebom [86, 60, 17]. Sugestie mogą dotyczyć różnych aspektów wspomagania wyboru, a w szczególności obiektów do kupienia, muzyki do słuchania, czy rekomendacji dotyczących wiadomości do przeczytania. W dobie doskonale rozwiniętego Internetu i środowiska e-commerce klient sklepu internetowego musi mierzyć się z ogromem bombardujących go informacji i ofert, a do tego nie zawsze są to informacje prawdziwe – tzw. szum informacyjny

i brudne informacje. Dlatego też rola systemów rekomendujących stała się nie do przecenienia dla użytkowników dokonujących zakupów online. W ostatnich latach opracowano wiele technik służących jak najlepszemu przygotowaniu odpowiedzi, które będą najlepiej dopasowane do konkretnej osoby, albo jej profilu. Wiele z tych rozwiązań zostało zaimplementowanych z sukcesem.

Ciekawym aspektem badań systemów rekomendujących jest fakt, że śmiało można je nazwać interdyscyplinarnymi. Łączą choćby takie aspekty jak:

- sztuczna inteligencja,
- problemy komunikacji człowiek-komputer,
- zagadnienia technologii informacyjnej,
- eksplorację danych,
- statystykę,
- badania dotyczące graficznych, adaptowalnych interfejsów użytkownika,
- systemów wspomagania decyzji,
- marketingu,
- zachowań konsumentów.

Jednym z najpopularniejszych systemów rekomendujących jest system powiadający tytuły książek, które mogą zainteresować klienta sklepu Amazon.com [48]. System jest spersonalizowany i dopasowuje swoje rekomendacje do każdego klienta z osobna. Inne systemy często analizują zachowania grup użytkowników. Istnieją także systemy, które nie dokonują spersonalizowanej analizy i proponują wszystkim odwiedzającym produkty z podobnej, wygenerowanej przez zaprojektowany algorytm listy. Mogą to być np. najlepiej sprzedające się pozycje, nowości, najwyżej oceniane produkty, czy najczęściej komentowane.

Systemy spersonalizowane w swojej najprostszej formie mogą tworzyć listę rankingową produktów, które warto polecić danemu użytkownikowi. RS próbuje zaproponować pozycje, które będą najlepiej odpowiadać wymaganiom klienta, będąc jednocześnie zgodne z zadeklarowanymi przez niego ograniczeniami. Aby to osiągnąć, RS może analizować informację o ocenach wystawianych przez użytkownika, sposobie w jaki nawiguje na stronie internetowej, czasie, który wykorzystuje na odwiedzanie konkretnych podstron, itd.

Rzecz systemów rozpoczęła dosyć oczywista i prosta obserwacja. Nowi klienci, którzy nie mają jeszcze wyrobionego swojego zdania w rzeczonym temacie, przy wyborze produktu kierują się najczęściej ocenami innych użytkowników [65]. Większym uznaniem będą obdarzane oceny wystawione przez osoby, które są godne zaufania w danym temacie – np. krytycy filmowi, dziennikarze. Naturalnym rozwinięciem tego podejścia może być fakt rekomendowania produktów,

które znalazły uznanie wśród aktywnych użytkowników o podobnym profilu (zainteresowań, wieku, zakupów). Rozwój sklepów, portali aukcji internetowych, a także znacznie poszerzenie oferowanego przez nie asortymentu (często w ramach jednego sklepu) wymusił dalszy rozwój i badania naukowe dotyczące problemów systemów rekomendujących. Należy również zauważyć wielki przyrost informacji, rozwój systemów porównujących ceny, czy serwisów kolekcjonujących opinie klientów na temat danego sklepu i/lub produktu. Ogromny przypływ informacji i gigantyczna ilość oferowanych produktów, jak i oferentów stawała się przytłaczająca dla znacznej części kupujących [91]. Obecne RS są znacznie bardziej rozbudowane. Korzystając z wyników badań naukowych, kolejnych modeli matematycznych i odkryć, potrafią dokonywać znacznie lepszych rekomendacji niż ich poprzednicy.

Ze względu na bardzo dynamiczny rozwój e-commerce również RS dotyka gigantyczny wzrost zainteresowania. Są one po prostu źródłem dodatkowej, jakże ważnej informacji. Z jednej strony są świetnym aplikowanym narzędziem, które jest z sukcesami wykorzystywane w tak znanych platformach jak choćby Amazon.com, YouTube, Netflix, Yahoo, Tripadvisor, czy IMDb. Z drugiej strony posiadają ogromny ładunek i miejsce dla badań naukowych. Organizowane są specjalistyczne konferencje, czy wydawane są światowej klasy czasopisma poświęcone w całości lub znacznej części niniejszemu tematowi.

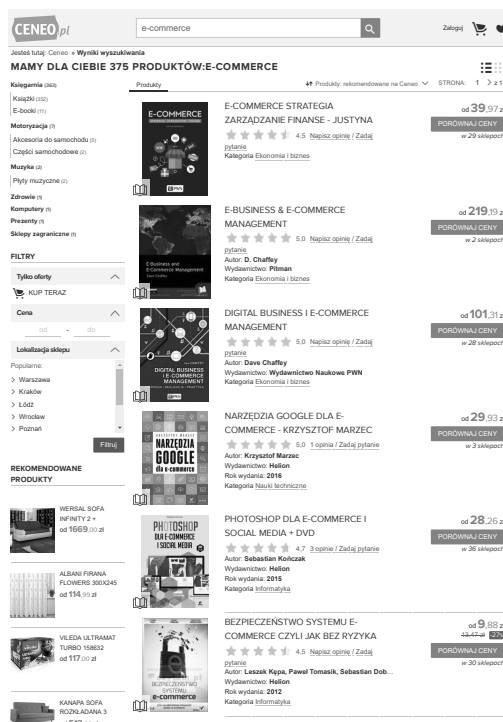
Treść tej krótkiej sekcji miała na celu jedynie przybliżenie idei systemów rekomendujących. Po szerokie i szczegółowe informacje warto odwołać się do książki *Recommender Systems Handbook* [89].

3.5 Porównywarki cen

Tzw. porównywarki cen (ang. Price Comparison Sites, PCS) to serwisy internetowe oferujące stworzenie list rankingowych ofert na dany produkt. Konkurencyjność oferentów [55] ma powodować obniżenie cen produktu [54], podobnie jak można to zaobserwować w przypadku aukcji internetowych [49]. Na prezentowanej liście rankingowej znajdują się oferty sklepów (tych, które mają podpisane umowy z właścicielami serwisu), które posiadają w sprzedaży szukany produkt. Warto zauważyć, że witryny oferujące porównywanie cen są w chwili obecnej bardzo popularne. Do ogólnie uznanych i często odwiedzanych należy zaliczyć: Google Shopping, NexTag, PriceGrabber, Shopping.com, Shopzilla. Co ciekawe, bardzo wysoką pozycję zajmuje tutaj doskonale znana na polskim rynku aplikacja www.ceneo.pl.

Przy wielu zaletach płynących z użytkowania serwisów porównujących ceny, dostrzec należy również liczne ułomności funkcjonalne. Mianowicie, największą ich wadą jest możliwość porównania ceny tylko jednego produktu w danym momencie. Jeśli mamy zamiar kupić wiele produktów (np. wiele książek, płyt audio, itd.), otrzymujemy wiele oddzielnych list rankingowych. Zakupienie wszystkich

żądanych produktów w najniższej sumarycznej cenie, okazuje się zadaniem bardzo trudnym. Nie wspominając już o dodatkowych aspektach występowania rabatów oferowanych przez sklepy, czy różnych opcjach dostawy [58].



Rysunek 3.3. Widok strony internetowej www.ceneo.pl dla zapytania *e-commerce* (odwiedzono: 22.10.2018 r.)

Rozdział 4

Problem optymalizacji zakupów internetowych

Problem optymalizacji zakupów internetowych (ang. Internet Shopping Optimization Problem, ISOP), gdzie klient chce zrealizować listę zakupów korzystając z oferty sklepów internetowych został oryginalnie zaproponowany przez Błazewicza et al. [13]. Celem jest dokonanie zakupów przy najniższym możliwym rachunku końcowym. Uwzględnić należy ceny produktów i koszty wysyłki [71].

Dla zilustrowania problemu przyjrzyjmy się prostemu przykładowi. Klient chce kupić pięć książek. Ceny produktów i koszty dostaw w sześciu różnych sklepach zostały zaprezentowane w tabeli 4.1. Celem jest zakup wszystkich pięciu książek przy minimalnym łącznym koszcie. Wsparcie z aktualnie dostępnych serwisów porównujących ceny (dalej zwane porównywarkami cen, PCS) pozwala na budowanie koszyka klienta w oparciu o najtańsze oferty dla poszczególnych książek. Wyniki procesu selekcji przedstawiono w tabeli 4.2. Zakupy dokonywane na podstawie sugestii porównywarki cen nie byłyby optymalne (koszt całkowity:

Tabela 4.1. Ceny książek (A-E) i koszty dostawy oferowane przez sześć sklepów internetowych

Koszt	A	B	C	D	E	Dostawa	Suma
Sklep 1	18	39	29	48	59	10	203
Sklep 2	24	45	23	54	44	15	205
Sklep 3	22	45	23	53	53	15	211
Sklep 4	28	47	17	57	47	10	206
Sklep 5	24	42	24	47	59	10	206
Sklep 6	27	48	20	55	53	15	218

Tabela 4.2. Rozwiązanie porównywarki cen – wynik procesu selekcji

	A	B	C	D	E	Dostawa	Suma
Cena	18	39	17	47	44	45	210
Sklep	Sklep 1	Sklep 1	Sklep 4	Sklep 5	Sklep 2		

Tabela 4.3. Optymalny koszt zakupu książek A-E w określonych sklepach

	A	B	C	D	E	Dostawa	Suma
Cena	18	39	17	48	47	20	189
Sklep	Sklep 1	Sklep 1	Sklep 4	Sklep 1	Sklep 4		

210), ponieważ proste porównanie cen nie obejmuje kosztów dostawy, które rosną wraz ze wzrostem liczby wybranych dostawców. W celu znalezienia globalnie najtańszego rozwiązania można dokonać pełnego wyszukiwania wszystkich możliwych realizacji zakupów. Jest to rozwiązanie nieefektywne czasowo i zasobowo, ale dla tak prostego przykładu można je wykonać nawet samemu na kartce papieru. Analizowany przez nas prosty przykład pokazuje, że proces optymalizacji przynosi zauważalne oszczędności. W naszym przykładzie optymalna cena zakupu książek (korzystając z oferty sklepów 1 i 4) wynosi 189 (patrz tabela 4.3).

4.1 Definicja formalna – model matematyczny

Kupujący poszukuje zestawu produktów $N = \{1, \dots, n\}$ robiąc zakupy w m sklepach. Dla dostępnych w sklepach produktów N_l , dla każdego produktu $j \in N_l$ określono jego koszt c_{jl} i cenę dostawy d_l – koszty są związane z każdym sklepem l , $l = 1, \dots, m$. Dla przejrzystości macierzy cen zakłada się, że $c_{jl} = \infty$ jeśli $j \notin N_l$. Problem polega na znalezieniu sekwencji wyłącznych zestawów (lub koszyków) produktów $X = (X_1, \dots, X_m)$, które nazywamy *sekwencją koszyka*, w taki sposób, że $X_l \subseteq N_l$, $l = 1, \dots, m$, $\cup_{l=1}^m X_l = N$, oraz całkowity koszt produktów i dostaw, oznaczony jako $F(X) := \sum_{l=1}^m \left(\delta(|X_l|)d_l + \sum_{j \in X_l} c_{jl} \right)$, jest minimalizowany.

4.2 Złożoność obliczeniowa problemu

Twierdzenie 1 *Problem ISOP należy do klasy problemów silnie NP-trudnych, nawet jeśli wszystkie ceny dostępnych produktów są równe zero, a wszystkie koszty dostaw wynoszą jeden.*

Dowód 1 *Dla odpowiadającej ISOP decyzyjnej wersji problemu (nazwijmy ją*

P1) przeprowadzimy pseudo-wielomianową transformację ze znanego powszechnie, silnie NP-zupełnego problemu EXACT COVER BY 3-SETS (X3C) [33]. Transformacja ta udowodni silną NP-trudność oryginalnej, optymalizacyjnej wersji problemu ISOP.

Exact Cover by 3-sets (X3C) można zdefiniować w następujący sposób: Przyjmując rodzinę $E = \{E_1, \dots, E_L\}$ 3-elementowych podzbiorów zbioru $K = \{1, \dots, 3k\}$ pytanie brzmi czy E zawiera dokładne pokrycie K , tj. taki podzbiór $Y \subseteq E$, że każdy $j \in K$ należy dokładnie do jednego 3-elementowego zbioru w Y ? Oczywiście jest, że jeśli Y jest rozwiązaniem dla X3C to $|Y| = k$.

Korzystając z instancji problemu X3C, konstruujemy następującą instancję problemu P1. Istnieją $m = L$ sklepy, w których znajdują się produkty określone zbiorem $N_l = E_l$, $l = 1, \dots, L$. Kupujący chciałby nabyć zestaw produktów $N = K$. Koszt każdego produktu dostępnego w każdym sklepie jest równy zeru ($c_{jl} = 0$, $j \in N_l$, $l = 1, \dots, m$), natomiast koszty dostawy z dowolnego sklepu do kupującego wynoszą jeden ($d_l = 1$, $l = 1, \dots, m$). Wartość progowa/graniczna (ang. threshold value) kryteriów wynosi $y = k$. Udowodnimy, że X3C ma rozwiązanie wtedy i tylko wtedy, gdy istnieje rozwiązanie X dla skonstruowanej instancji problemu P1 przy $F(X) \leq k$. Łatwo zaobserwować, że transformacja jest jednocześnie wielomianowa i pseudo-wielomianowa. Dlatego problem P1 należy do klasy NP.

Niech Y będzie rozwiązaniem dla X3C. Skonstruujmy rozwiązanie dla problemu P1, w którym wymagane k produkty są kupowane w określonych sklepach według zestawów produktów $N_l = E_l \in Y$, tj. $X_l = N_l$ jeśli $N_l \in Y$ i $X_l = \emptyset$ jeśli $N_l \notin Y$. Skoro Y jest dokładnym pokryciem K , wszystkie wymagane produkty są zakupione, a koszt takiego rozwiązania jest równy $F(X) = k$.

Żałóży teraz, że istnieje rozwiązanie X dla problemu P1 z kosztem $F(X) \leq k$. Z jednej strony, dla tego rozwiązania liczba sklepów o wartościach $X_l \neq \emptyset$ nie powinna przekraczać k , ponieważ w przeciwnym razie $F(X) > k$. Z drugiej strony liczba tych sklepów nie powinna być mniejsza niż k , ponieważ w przeciwnym razie co najmniej jeden produkt $j \in N$ nie będzie zakupiony. Dlatego też istnieją dokładnie k sklepy o wartości $X_l \neq \emptyset$. Ponieważ zakupione produkty tworzą zestaw K , zbiór wybranych sklepów $X_l \neq \emptyset$ stanowi rozwiązanie dla X3C. ■

Przedyskutujemy teraz możliwość aproksymacji dla problemu ISOP. Rozważmy jego szczególny przypadek, w którym koszt każdego produktu w dowolnym sklepie wynosi zero, a koszt dostawy z dowolnego sklepu do kupującego równy jest jeden. Ten szczególny przypadek jest równoważny z MINIMUM SET COVER problem [23].

MINIMUM SET COVER: Przy zbiorze C podzbiorów skończonego zbioru S , odszukaj pokrycie zbioru S , tzn. podzbiór $C' \subseteq C$ taki, że każdy element w S należy do co najmniej jednego C' , co minimalizuje moc zbioru, tj. $|C'|$.

Według literatury [85] problem MINIMUM SET COVER jest wielomianowo nieaproxymowalny w stosunku $c \cdot \ln |S|$, dla stałej $c > 0$. W związku z tym

przedstawić można następującą tezę [13].

Teza 1 *Nie istnieje wielomianowy algorytm $(c \cdot \ln n)$ -aproxymacyjny dla problemu ISOP, chyba że $\mathcal{P} = \mathcal{NP}$.*

4.3 Rozwiązania wielomianowe dla specjalnych przypadków

Należy zauważyć, że trudność rozwiązania problemu ISOP została stwierdzona na podstawie założenia, że zarówno liczba produktów n , jak i sklepów m są zmiennymi. W tej sekcji zaprezentowano dwa podejścia wielomianowe, które są możliwe do zastosowania gdy n lub m są stałymi.

Idea pierwszego algorytmu SHOP-ENUM opiera się na wyliczeniu wszystkich możliwych wyborów sklepów, w których znajdują się wszystkie wymagane produkty. Należy wybrać najlepszą sekwencję zakupów $X^{(M)}$ dla każdego zestawu wybranych sklepów M , $M \subseteq \{1, \dots, m\}$. Umożliwi to obliczenie całkowitej sumy kosztu produktów i dostawy dla każdego M , a w końcu znalezienie rozwiązania X^* przy minimalnym koszcie całkowitym F^* .

Algorytm Shop-Enum

Krok 1. Ustaw wartość $F^* = \infty$ i $X_l^* = \phi$, $l = 1, \dots, m$.

Krok 2. Rozważ wybór sklepów, w których można kupić wszystkie wymagane produkty. Może zostać wybrany każdy sklep l , $l \in \{1, \dots, m\}$ jeśli N_l nie jest pusty – tak więc do dyspozycji mamy 2^m kroków. Dla każdego zestawu wybranych sklepów M przeprowadź następujące obliczenia. Jeśli M nie zawiera wszystkich wymaganych produktów pominię ten wybór. W przeciwnym razie wykonaj co następuje. Określ sekwencję koszyka $X^{(M)} = (X_1^{(M)}, \dots, X_m^{(M)})$, $X_l^{(M)} \subseteq N_l$, $1, \dots, m$ jak poniżej. Dla każdego produktu $i \in N$, wybierz sklep $l \in M$, w którym koszt produktu i jest najtańszy. Może to być wykonywane w stałym czasie, jeśli dane kosztów c_{il} produktów i przechowujemy na stosie. Przypisz produkt i do odpowiedniego multizbioru $X_l^{(M)}$. Zauważ, że strategia przypisywania może wyeliminować niektóre z wybranych sklepów. Podejście jest optymalne dla danego doboru sklepów ze względu na fakt, że całkowite koszty produktów są minimalizowane, a całkowity koszt dostawy nie przekracza sumy kosztów dostawy dla wybranych sklepów. Oblicz całkowity koszt $F(X^{(M)})$. Jeżeli $F(X^{(M)}) < F^*$ ponownie ustaw wartość $F^* = F(X^{(M)})$ i $X^* = X^{(M)}$.

Krok 3. Optymalne rozwiązanie wyjściowe to X^* z minimalnym kosztem F^* .

Złożoność czasowa algorytmu SHOP-ENUM wynosi $O(n2^m)$, przy czym jest wielomianowa (liniowa), jeśli liczba sklepów m jest stała.

Ideą drugiego algorytmu PRODUCT-ENUM jest wyliczenie wszystkich możliwości wyboru sklepu dla każdego produktu. Niech $S = (S_1, \dots, S_n)$ będzie sekwencją sklepów taką, że $S_i \in \{1, \dots, m\}$ jest sklepem, w którym zostanie zakupiony produkt i , $i = 1, \dots, n$. Algorytm określa sekwencję sklepów S^* przy minimalnym koszcie całkowitym F^* .

Algorytm Product-Enum

Krok 1. Ustaw wartość $F^* = \infty$ oraz $S_i^* = \phi$, $i = 1, \dots, n$.

Krok 2. Rozważ sekwencję sklepów $S = (S_1, \dots, S_n)$ w taki sposób, że produkt $i \in N$ zostanie zakupiony w sklepie S_i , $i = 1, \dots, n$. Jest co najwyżej m^n takich sekwencji. Dla każdego S , oblicz koszt odpowiedniej sekwencji koszyka $F(S)$. Jeśli wartość $F(S) < F^*$ ustaw ponownie wartości $F^* = F(S)$, $S^* = S$ i przejdź do analizy kolejnego zestawu sklepów $S \in S$.

Krok 3. Optymalne rozwiązanie wyjściowe to X^* z minimalnym kosztem F^* .

Złożoność czasowa algorytmu PRODUCT-ENUM wynosi $O(nm^n)$ – jest wielomianowa, jeśli liczba produktów n jest stała.

Algorytm PRODUCT-ENUM może być stosowany w praktyce, jeżeli liczba produktów n jest niewielka, a liczba sklepów m , w których są one dostępne jest duża. Alternatywnie, jeśli liczba sklepów m jest mała, a liczba produktów n jest duża, algorytm SHOP-ENUM może być wydajny.

4.4 Algorytmy

W niniejszej sekcji zaprezentowane zostaną algorytmy rozwiązujące ogólny problem ISOP – rozwiązanie dokładne/optymalne i różne heurystyki. Dokładne rozwiązanie oparte jest o programowanie liniowe całkowitoliczbowe (ang. Integer Linear Programming Model, ILP). Pierwszym przybliżonym podejściem jest algorytm komórkowy oparty na obliczeniach wykonywanych przez proste komórki przetwarzające, które działają na własnej populacji i dzielą się informacjami z innymi komórkami w celu poprawy wydajności całego systemu. Drugi algorytm heurystyczny związany jest ze specjalnie opracowanym wariantem algorytmu MIN-MIN. Kolejne heurystyki zbudowane są na podstawie filozofii algorytmów zachłannych, jak i bardziej rozwiniętego podejścia szukającego lokalnych optimum.

4.4.1 Model programowania całkowitoliczbowego

W celu optymalnego rozwiązania problemu ISOP z kosztami dostawy, zaproponowano podejście oparte o programowanie całkowitoliczbowe [57].

Poza notacją używaną dotychczas warto nadmienić, że zmienna binarna x_{ij} wskazuje, czy produkt j został zakupiony w sklepie i , natomiast zmienna binarna y_i wskazuje, czy przynajmniej jeden produkt został zakupiony w sklepie i .

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m d_i y_i \quad (4.1)$$

$$\text{p.o. } x_{i,j} \in \{0, 1\} \quad \forall i \in M, \forall j \in N, \quad (4.2)$$

$$y_i \in \{0, 1\} \quad \forall i \in M, \quad (4.3)$$

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = n, \quad (4.4)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j \in N, \quad (4.5)$$

$$n y_i - \sum_{j=1}^n x_{ij} \geq 0 \quad \forall i \in M. \quad (4.6)$$

W niniejszym modelu ILP funkcja celu (4.1) jest przedstawiona jako całkowity koszt zakupu produktów w wybranych sklepach, łącznie z kosztami dostawy.

Lista i opis ograniczeń przedstawiają się następująco. Ograniczenie 4.4 gwarantuje, że liczba zakupionych produktów jest równa liczbie produktów na liście zakupów, natomiast ograniczenie 4.5 gwarantuje, że wybrany zostanie tylko jeden produkt każdego rodzaju. Ograniczenie 4.6 zapewnia, że zmienna y_i przyjmuje wartość 1 przy zakupie produktu ze sklepu i .

Jest to model dla liczb całkowitych, ponieważ zmienne decyzyjne x_{ij} i y_i są reprezentowane jako liczby całkowite. Dodatkowo model ma charakter liniowy, ponieważ funkcja celu i funkcje w ograniczeniach są liniowe.

Poprawność proponowanego modelu ILP została zweryfikowana za pomocą algorytmu wyliczeniowego. Metoda ta generuje wszystkie kombinacje list zakupów, co zapewnia uzyskanie optymalnego rozwiązania we wszystkich przypadkach (ale tylko dla bardzo małych instancji problemu). Rozwiązania uzyskane za pomocą ILP zostały porównane z optymalnymi rozwiązaniami algorytmu wyliczeniowego w celu wykrycia rozbieżności. Test ten potwierdził, że ILP uzyskało właściwe, optymalne rozwiązanie na zestawie danych *Small*.

4.4.2 Algorytm MIN-MIN i jego warianty

Algorytm MIN-MIN jest dobrze znany i szeroko opisany w literaturze [32, 26, 76] jako przedstawiciel statycznych algorytmów planowania zadań, gdzie każde zadanie powinno być przypisane do zasobu.

Podzielony jest na dwie fazy:

1. Pierwszym etapem jest ustalenie / obliczenie wszystkich oczekiwanych czasów

realizacji dla każdego zadania i zasobu: maczyra o wymiarach $t \times n$.

2. W drugim etapie wszystkie zadania są szeregowane w oparciu o minimalne czasy realizacji. W każdym kroku algorytmu zadanie z minimalnym czasem realizacji jest przypisywane do odpowiedniego zasobu i usuwane z listy zadań oczekujących na wykonanie.

Jedną z głównych zalet tego podejścia jest to, że generalnie jest ono w stanie uzyskać dobrej jakości rozwiązania przy stosunkowo niewielkim koszcie obliczeniowym.

Pseudokod MIN-MIN został zaprezentowany w postaci algorytmu 1. Algorytm ten rozpoczyna obliczenia od pustej sekwencji wyboru produktów X . Ze zbioru N dostępnych produktów, szukamy produktu, który ma najniższą łączną cenę po dodaniu do X , biorąc pod uwagę cenę sprzedaży produktu i koszt dostawy ze sklepu.

Element, który w najmniejszym stopniu zwiększa całkowity koszt aktualnego rozwiązania, dodawany jest do X . Następnie przypisany produkt jest usuwany z listy zakupów N oraz z wielozbioru N_i .

Powyższe kroki są powtarzane do momentu zakupu wszystkich produktów, innymi słowy, póki lista zakupów N nie będzie pusta. Algorytm MIN-MIN jest algorytmem wielomianowym o złożoności czasowej $O(mn^2)$.

Jednym z głównych mankamentów tej heurystyki jest fakt, że jest ona podatna na stagnację ze względu na swoją deterministyczną naturę. Co więcej, jego działanie może prowadzić do uzyskania lokalnie nieoptymalnych rozwiązań, ponieważ w każdym sklepie występuje dodatkowy koszt dostawy. Należy to wziąć pod uwagę jako jedną z głównych różnic pomiędzy algorytmem MIN-MIN, a algorytmem zachłannym GREEDY (opisanym w dalszej części – podsekcja 4.4.4).

Potencjalną strategią zniwelowania tego problemu jest zastosowanie jednej lub kilku metod wyszukiwania lokalnego, takich jak przykładowo przedstawiona w algorytmie 2. Algorytm rozpoczyna działanie od utworzenia wektora wartości par, który zawiera produkty j przypisane w sekwencji wyboru produktów X_{old} z odpowiadającym kosztem $cost$ i zawiera c_{ij} , a także d_i z wielozbioru N_i .

Wektor V jest sortowany rosnąco. Następnie sekwencja X_{old} jest przypisana do sekwencji X , a jej bieżąca wartość funkcji celu jest zapisana w zmiennej $oldTotalCost$.

Kolejno dla każdego produktu j w V sprawdzamy, zaczynając od najdroższych produktów, z którego sklepu moglibyśmy go kupić, aby zmniejszyć całkowity koszt wyboru produktów X .

Po zakończeniu wyszukiwania dla danego produktu, zostaje on przyporządkowany do nowego sklepu lub pozostaje w sklepie wybranym uprzednio. Proces ten trwa do momentu, aż wszystkie produkty j w V zostaną przeanalizowane.

Dodatkowo, algorytm LS może być zastosowany do stanów pośrednich rozwiązania generowanego przez algorytm MIN-MIN. Może to prowadzić do uzyskania lepszego rozwiązania końcowego. Algorytm lokalnego wyszukiwania jest algorytmem wielomianowym o złożoności czasowej $O(n^2)$.

Algorytm 1 MIN-MIN

Obliczanie rozwiązania przy użyciu algorytmu Min-Min.

Require:

N_i : Wielozbiór dostępnych produktów dla każdego sklepu.

N : Lista zakupów.

M : Lista sklepów.

Ensure:

$X = (X_1, \dots, X_m)$: Sekwencja wyboru produktów.

F : łączna suma kosztów produktów i dostaw.

```

1:  $X = (X_1 = \emptyset, \dots, X_m = \emptyset)$ 
2: while  $N \neq \emptyset$  do
3:    $min = \infty$ 
4:   for all  $i \in M$  do
5:     for all  $j \in N$  do
6:       przydzielProdukt  $j$  to  $X_i$ 
7:       if  $F(X) < min$  then
8:          $min = F(X)$ 
9:          $j' = j$ 
10:      end if
11:     usuńProdukt  $j$  z  $X_i$ 
12:   end for
13: end for
14: przydzielProdukt  $j'$  w sklepie  $i$  to  $X_i$ 
15: usuńProdukt  $j'$  z  $N$  i  $j' \in N_i$ 
16: end while
17: return  $X$ 

```

W eksperymentach porównano trzy wersje algorytmu MIN-MIN w celu określenia jakości rozwiązań i czasu potrzebnego na obliczenia: oryginalny MIN-MIN, algorytm MINMIN+LOCAL SEARCH, oraz odmianę MINMIN+MIDLOCAL SEARCH.

Pseudokod i zasada działania drugiej z wymienionych powyżej odmian została zaprezentowana w algorytmie 3. Algorytm rozpoczyna działanie od wygenerowania rozwiązania metodą MIN-MIN (algorytm 1), aby natychmiast uzyskać rozwiązanie lokalnego optimum przy użyciu metody lokalnego wyszukiwania opisanej w algorytmie 2.

Algorytm wyszukiwania MINMIN+LOCAL jest algorytmem wielomianowym o złożoności czasowej $O(mn^2)$. Złożoność czasowa lokalnego wyszukiwania jest pominięta ($O(n^2)$), ponieważ zawiera się w złożoności algorytmu MIN-MIN.

Trzecia wariacja, pokazana w algorytmie 4, stosuje algorytm wyszukiwania

Algorytm 2 Przeszukiwania lokalnego LOCAL SEARCH, LS

Oblicza lokalne optimum w zadanej przestrzeni przeszukiwania.

Require:

$X_{old} = (X_{old_1}, \dots, X_{old_m})$: Lista (sekwencja) produktów.

N_i : Wielozbiór produktów dostępnych dla każdego sklepu.

Ensure:

$X = (X_1, \dots, X_m)$: Poprawiona lista wybranych produktów.

F : suma cen produktów i kosztów wysyłki.

```

1: dodajProdukt  $j$  przy  $cost = c_{ij} + d_i$  z  $X_{old_i}$  do  $V$ 
2: posortuj  $V$  według  $cost$  w kierunku rosnącym
3:  $X = X_{old}$ 
4:  $oldTotalCost = F(X)$ 
5: for all  $j \in V$  do
6:   usuńProdukt  $j$  z  $X_i$ 
7:   for all  $i \in M$  do
8:     przydzielProdukt  $j$  do  $X_i$ 
9:     if  $F(X) < oldTotalCost$  then
10:        $oldTotalCost = F(X)$ 
11:        $i' = i$ 
12:     end if
13:   usuńProdukt  $j$  z  $X_i$ 
14:   end for
15:   przydzielProdukt  $j$  w sklepie  $i'$  do  $X_i$ 
16: end for
17: return  $X$ 

```

lokalnego do częściowych rozwiązań skonstruowanych przez algorytm MIN-MIN, aby je ulepszyć i poprowadzić do lepszego rejonu w przestrzeni wyszukiwania. Moment zastosowania metody wyszukiwania lokalnego jest kontrolowany poprzez zmienną α , która zależy od wielkości rozwiązywanej instancji. Algorytm MINMIN+MIDLOCAL SEARCH jest algorytmem wielomianowym o złożoności czasowej $O(mn^4)$. Jest to najgorszy przypadek, gdzie algorytm LOCAL SEARCH jest stosowany za każdym razem, gdy nowy sklep jest dodawany do częściowego rozwiązania.

4.4.3 Filozofia przetwarzania komórkowego – algorytm CELLULAR

Podejście obliczeń komórkowych jest opisane [93] jako filozofia projektowania algorytmów oparta na trzech powiązanych ze sobą zasadach. Zasada prostoty

Algorytm 3 MINMIN+LOCAL SEARCH

Generuje rozwiązanie pierwotne przy użyciu MIN-MIN, które jest poprawiane poprzez metodę Local Search.

Require:

N_i : Wielozbiór produktów dostępnych dla każdego sklepu.

N : Lista zakupów (zbiór produktów).

M : Lista (zbiór) sklepów.

Ensure:

$X = (X_1, \dots, X_m)$: Lista (sekwencja) wyboru produktów.

minMin: Algorytm MIN-MIN (algorytm 1).

localSearch: Algorytm LOCAL SEARCH (algorytm 2).

1: $X = \text{minMin}(N_i, N, M)$

2: $X = \text{localSearch}(X, N_i)$

3: **return** X

mówi, że komórka przetwarzająca powinna wykonywać bardzo proste zadania, które pochłaniają niewiele czasu. Następna zasada współbieżności mówi o wykorzystaniu wielu niezależnych, pojedynczych komórek w celu rozwiązania pojedynczego zadania. Trzecią jest zasada lokalizacji, która wskazuje, że biorąc pod uwagę dużą liczbę komórek, komunikacja między wszystkimi z nich jest niepraktyczna. Preferowana jest komunikacja lokalna między sąsiednimi komórkami.

Model ten jest szczególnie przydatny do stosowania w chmurze obliczeniowej lub centrach danych, ponieważ wykorzystuje dużą liczbę dostępnych procesorów.

ISOP ze swojego założenia i ze względu udowodnionej trudności obliczeniowej znajduje miejsce w tego rodzaju środowiskach. Łatwo zauważyć, że przy dużej ilości jednoczesnych zapytań o rozwiązanie centralne serwery z kilkoma jednostkami obliczeniowymi są zobowiązane do wykonywania obliczeń dla różnych użytkowników na całym świecie w jak najkrótszym czasie. Jest to główny powód, dla którego rozważano zastosowanie algorytmu przetwarzania komórkowego.

Algorytm przetwarzania komórkowego został zaproponowany [99] w celu rozwiązania, z obiecującymi wynikami, liniowego problemu zamówień z narastającymi kosztami (ang. The Linear Ordering Problem with cumulative costs).

Nietrudno zauważyć podobieństwa ze znanym podejściem hiperheurystycznym [16]. Główna różnica przy podejściu przetwarzania komórkowego polega na tym, że każda komórka ma pełną wiedzę na temat problemu, który jest rozwiązywany. Natomiast podejście hiperheurystyczne buduje swego rodzaju barierę (separuje) pomiędzy kontrolerem a heurystyką operującą na niskim poziomie. Drugą główną różnicą jest fakt, że komórki przetwarzające są przystosowywane i dopasowywane do rozwiązywanego problemu, podczas gdy podejście hiperheu-

Algorytm 4 MINMIN+MIDLOCAL SEARCH

Generuje rozwiązanie przy użyciu MIN-MIN, a następnie poprawia je poprzez LOCAL SEARCH podczas konstruowania rozwiązania.

Require:

N_i : Wielozbiór produktów dostępnych dla każdego sklepu.

N : Lista zakupów (zbiór produktów).

M : Lista (zbiór) sklepów.

Ensure:

$X = (X_1, \dots, X_m)$: Lista (sekwencja) wyboru produktów.

localSearch: Algorytm LOCAL SEARCH (algorytm 2).

F : suma cen produktów i kosztów wysyłki.

updateAlpha: określa następną iterację, w której będzie zastosowany *localSearch*.

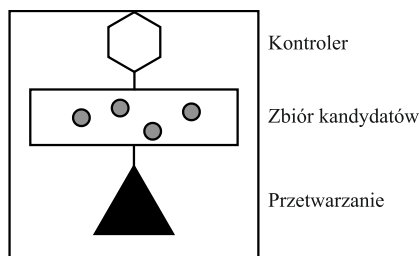
```

1:  $X = (X_1 = \emptyset, \dots, X_m = \emptyset)$ 
2: while  $N \neq \emptyset$  do
3:    $min = \infty$ 
4:   for all  $i \in M$  do
5:     for all  $j \in N$  do
6:       przydzielProdukt  $j$  do  $X_i$ 
7:       if  $F(X) < min$  then
8:          $min = F(X)$ 
9:          $j' = j$ 
10:      end if
11:      usuńProdukt  $j$  z  $X_i$ 
12:    end for
13:  end for
14:  przydzielProdukt  $j'$  w sklepie  $i$  do  $X_i$ 
15:  usuńProdukt  $j'$  z  $N$  i  $j'$  z  $N_i$ 
16:  if  $currentIteration == \alpha$  then
17:     $X = localSearch(X, N_i)$ 
18:     $updateAlpha(\alpha)$ 
19:  end if
20: end while
21: return  $X$ 

```

rystyczne zależy od ogólnej, niskopoziomowej heurystyki, która może być zastosowana do większej różnorodności problemów optymalizacyjnych [16].

Algorytm przetwarzania komórkowego opiera się na niezależnych komórkach przetwarzających, które utrzymują własną populację i przetwarzają dane za pomocą własnego mechanizmu. Składniki każdej komórki zaprezentowano na ry-



Rysunek 4.1. Reprezentacja pojedynczej komórki przetwarzającej

sunku 4.1. Zbiór zawiera rozwiązanie kandydujące lub ich zestaw, które zostaną zmodyfikowane przez komponent przetwarzający (ang. Processing). Komponent przetwarzający jest metodą lub heurystyką, która ulepsza proponowane rozwiązania kandydatów. Metoda zawarta w rdzeniu przetwarzającym może być taka sama, z niewielkimi zmianami, lub całkowicie różniąca się od metod kandydatów – w zależności od algorytmu przetwarzania komórkowego. Komponent kontrolera komórki zapewnia stabilizację puli kandydatów, będąc w stanie podjąć działania naprawcze, takie jak wygenerowanie nowego zestawu proponowanych rozwiązań w puli, przeniesienie istniejących rozwiązań do innego regionu przestrzeni wyszukiwania lub wstrzymanie wykonywania własnych działań w celu uniknięcia niewłaściwego wykorzystania zasobów obliczeniowych, takich jak czas przetwarzania danych i przestrzeń pamięci.

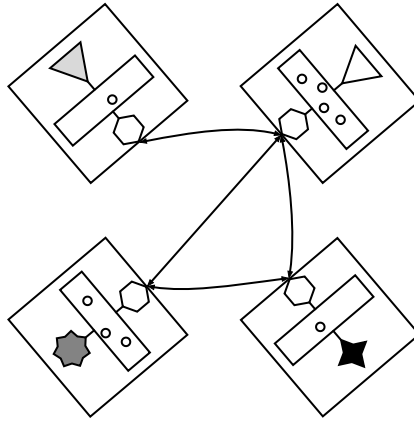
Wszystkie komórki komunikują się z innymi, pobliskimi komórkami w celu poprawy sprawności i jakości działania każdej z nich. Komunikacja może następować zarówno podczas wykonywania zadań lub po ich częściowym, czy całkowitym przetworzeniu. Proces ten ilustruje rysunek 4.2.

Cały proces zwykle kończy się, gdy większość komórek zbiega się do lokalnego optimum. Może ono być lub nie optymalnym rozwiązaniem dla danego przykładu.

W tym szczególnym przypadku zastosowania algorytmu przetwarzania komórkowego do rozwiązania problemu ISOP zdefiniowano wykorzystanie pięciu komórek przetwarzających. Jako kryterium zatrzymania przyjęto pięć iteracji bez poprawy najlepszego rozwiązania. Konfiguracja ta została opracowana poprzez obniżenie wartości zestawu parametrów, początkowo ustawionych na wysokim poziomie wartości. Redukcja następowała do momentu, w którym jakość uzyskiwanych rozwiązań dla średniego zestawu danych nie ulegała pogorszeniu przy przyporządkowaniu nowych parametrów.

Algorytm rozpoczyna działanie od skonstruowania wstępnego rozwiązania metodą MIN-MIN dla każdej komórki. Następnie wykonywane jest udoskonalenie poprzez zastosowanie lokalnego algorytmu wyszukiwania.

Implementacja wszystkich komórek w całym algorytmie przetwarzania ko-



Rysunek 4.2. Komunikacja pomiędzy komórkami przetwarzającymi

Algorytm 5 ITERATIVE LOCAL SEARCH

Poprawia dane rozwiązanie poprzez relokację wszystkich elementów sekwencji na lepszą pozycję, dopóki nie zostanie osiągnięte lokalne optimum.

Require:

N_i : Wielozbiór dostępnych produktów dla każdego sklepu.

Ensure:

$X = (X_1, \dots, X_m)$: Sekwencja wyboru produktów.

perturbSolution: Losowa relokacja części elementów w X .

localSearch: Algorytm przeszukiwania lokalnego LS (algorytm 2).

F : Łączna suma cen produktów i kosztów dostawy.

```

1:  $bestX = X$ 
2: while  $noImprov < maxNoImprov$  do
3:    $perturbSolution(X, percentage)$ 
4:    $X = localSearch(X, N_i)$ 
5:   if  $X < bestX$  then
6:      $bestX = X$ 
7:      $noImprov = 0$ 
8:   else
9:      $noImprov++$ 
10:  end if
11: end while
12:  $X = localSearch(bestX, N_i)$ 
13: return  $bestX$ 

```

mórkowego (jako element przetwarzający Iterated Local Search [ILS]) została zobrazowana w algorytmie 5.

Ustalono następujące parametry algorytmu ILS: *maxNoImprov*, który określa maksymalną liczbę iteracji bez poprawy wyniku został ustalony na 10 iteracji; zmienna *percentage*, która kontroluje procent losowo ustawianego rozwiązania została ustalona na 0,5. Algorytm znajduje najlepsze położenie dla każdego elementu w rozwiązaniu, a jeśli żadna inna pozycja nie poprawia wartości funkcji celu, element ten jest pozostawiony na swoim pierwotnym miejscu.

Komunikacja między komórkami odbywa się poprzez udostępnienie wszystkich rozwiązań sąsiednim komórkom, po każdej iteracji.

Aby nie tylko określić przydatność nowo powstałych algorytmów, ale również ocenić ich jakość względem wcześniej opracowanych, postanowiono dokonać bardziej szczegółowych badań eksperymentalnych. Takich, które miały uwzględniać implementację dwóch prostych algorytmów GREEDY [14, 106] i FORECASTING [10]. Algorytmy te zostaną krótko opisane w następującym fragmencie niniejszego rozdziału.

4.4.4 Algorytm zachłanny GREEDY

Zaimplementowanie prostego algorytmu zachłannego jest dobrym sposobem na bardzo szybkie rozwiązanie problemów NP-trudnych i w ogólności może zapewnić dobrą jakość rozwiązań poprzez zapewnienie lokalnie optymalnych rozwiązań na każdym etapie algorytmu. Dlatego też postanowiliśmy przygotować prosty algorytm zachłanny dedykowany dla problemu ISOP. Głównym celem było opracowanie metody, który może działać szybko i zapewnia lepsze wyniki niż algorytmy znane ze stron porównujących ceny. Zestawienie wyników eksperymentów obliczeniowych dla algorytmów porównywarek cen (w dwóch różnych wersjach) znajduje się w rozdziale 6 pracy Wojciechowskiego i Musiała [106].

W dalszej części zamieszczono pseudokod algorytmu zachłannego obrazujący zasadę jego działania (patrz algorytm 6). W celu uzyskania lepszych rezultatów algorytm może być uruchomiony wielokrotnie dla różnych kolejności produktów na liście N , np. $rec_n < rec_{n-1} < rec_1$, lub losowo $rec_n ? rec_{n-1} ? rec_1$, gdzie $?$ oznacza dowolną relację.

Zaobserwowano, że algorytm zachłanny GREEDY prezentuje dobrą wydajność i jakość uzyskiwanych wyników. Niemniej jednak, w bardzo specyficznych sytuacjach (mimo, że niezwykle nierealnych) może prowadzić do osiągnięcia wyniku wielokrotnie gorszego od optimum. Niniejszą sytuację obrazuje tabela 4.4.

Można łatwo zauważyć, że dla dowolnej kolejności startowej produktów (sekwencja) wszystkie zostaną zakupione w sklepie 1 z łącznym kosztem $n * W$. Optymalne natomiast będzie rozwiązanie, w którym wszystkie produkty zostały zakupione w dowolnym sklepie innym niż 1, a łączny koszt wynosi $W + \epsilon$. Pierwszy krok algorytmu heurystycznego powoduje jego "skazę" i stąd późniejsza zmiana na inny sklep dla każdego lokalnego kroku wydaje się nieopłacalna.

Algorytm 6 Zachłanny dla ISOP

Pseudokod obrazujący działanie algorytmu zachłannego, który działa poprzez wyznaczanie rozwiązań lokalnie optymalnych.

Require:

$N = \{1, \dots, n\}$ – produkty do kupienia

$M = \{1, \dots, m\}$ – sklepy

p_{ij} – cena produktu j w sklepie i

rec_j – rekomendowana cena sprzedaży produktu j

d_i – koszt dostawy dla sklepu i

$ds_i = d_i$ – koszt dostawy używany do finalnej kalkulacji

$sum = 0$ – łączna cena zakupów

$\forall_{j \in N} R_j = 0$ – wskaźnik wyboru (lista realizacji), dla każdego produktu j

- 1: Ustaw kolejność produktów $1, \dots, n$ gdzie $rec_n \geq rec_{n-1} \geq rec_1$
 - 2: Wybierz pierwszy produkt $j = 1$
 - 3: **while** $doj < n$
 - 4: Wybierz sklep i dla produktu j tak, że $\min(f_i(p_{ij}) + d_i)$; $i = 1, \dots, m$
 - 5: $R_j = i$. $d_i = 0$
 - 6: Wybierz kolejny produkt $j = j + 1$
 - 7: **end while**
 - 8: $S = \sum_{i=1}^M f_i(P) + ds_i : d_i \neq 0$
 - 9: **return** S, R
-

Pozostanie przy sklepie 1 prowadzi do bardzo niekorzystnego wyniku.

W celu zabezpieczenia przed podobnymi przypadkami opracowaliśmy kolejny algorytm heurystyczny, oznaczony jako FORECASTING.

4.4.5 Algorytm FORECASTING

W pierwszym kroku algorytmu, dla każdego zbioru N_i produktów odpowiednich dla sklepu i wyznaczana jest wartość

$$V_i = (d_i + \sum_{j \in N_i} p_{ij}) / |N_i|, \quad i = 1, \dots, n, \quad (4.7)$$

a następnie wybierane są najtańsze $\alpha\%$ produkty spośród zbioru N_i z minimalną wartością V_i . W każdym kolejnym kroku zbiór N_i jest aktualizowany poprzez usunięcie wybranych produktów. Wartość V_i jest obliczana zgodnie z powyższym wzorem (4.7) jeśli nie wybrano produktu z tego sklepu i odpowiednio $V_i := \sum_{j \in N_i} p_{ij} / |N_i|$ jeśli przynajmniej jeden produkt został zakupiony w sklepie i . Finalnie zbiór N_i zawiera tylko niewybrane produkty.

Tabela 4.4. Struktura cen, dla której algorytm zachłanny zaproponuje kiepskie rozwiązanie

	prod 1	prod 2	...	prod n	dostawa
sklep 1	W	W	...	W	0
sklep 2	ϵ	ϵ	...	ϵ	W
sklep 3	ϵ	ϵ	...	ϵ	W
...	ϵ	ϵ	...	ϵ	W
sklep m	ϵ	ϵ	...	ϵ	W

4.5 Eksperymenty obliczeniowe

Eksperymenty obejmą ocenę szeregu różnych metod rozwiązania problemu ISOP: ILP, MIN-MIN i jego wariantów (algorytmy 3 i 4), oraz algorytm przetwarzania komórkowego. Ponadto zaprezentowane zostaną liczne porównania z dwoma prostszymi algorytmami GREEDY i FORECASTING.

W celu przygotowania danych testowych dla eksperymentów obliczeniowych stworzono realistyczny model rynku e-commerce. Zbadano relacje pomiędzy strukturą konkurencyjności, reklamą, cenami i ich rozproszeniem w sklepach internetowych. Jako grupa reprezentatywnych produktów do uwzględnienia w eksperymentach obliczeniowych wybrano książki, ze względu na ich szeroką dostępność w wirtualnych (internetowych) sklepach.

Dla opracowanego modelu przyjęto niektóre informacje, analizy i wyniki eksperymentów obliczeniowych opisanych w literaturze [81, 84, 80, 19]. Skupiono uwagę głównie na modelowej definicji księgarń elektronicznych (koncentrując się głównie na książkach, płytach CD z muzyką, DVD z filmami), cenach, czynniku akceptacji, marce sprzedawcy detalicznego [18]. Co jest również niezmiernie istotne dla zdefiniowania modelu problemu ISOP – na rozpiętości cen. Należy również zauważyć, że konsumenci mogą wybierać spośród dużej liczby księgarń internetowych. Dane do naszej próby zostały zebrane z 32 sklepów i objęły największe sklepy w Stanach Zjednoczonych, w tym Amazon, BarnesandNoble.com, Borders.com, Buy.com, Booksamillion, a także czołowych sprzedawców wśród księgarń internetowych w Polsce, takich jak empik.com czy merlin.pl. Zdecydowano o przeglądzie i aktualizacji modelu generowania danych zaprezentowanego w pracy Blazewicza et al. [13]. Celem było stworzenie nowego, bardziej wyrafinowanego i jeszcze bardziej realistycznego modelu niż ten proponowany uprzednio w literaturze. Ze względu na bardzo realistyczną formę i zakres pracy wniesiony w projektowany model, został on również wykorzystywany w badaniach dotyczących bardziej skomplikowanych odmian problemu ISOP – wariantów zawierających choćby wielokrotne funkcje rabatujące i dodatkowe warunki sprzedaży [11] (patrz rozdziały 5 i 6).

Zakłada się, że każdy sklep posiada wszystkie wymagane książki. W każdym przypadku następujące wartości są losowo generowane dla wszystkich i i j w odpowiednich zakresach. *Cena referencyjna (ref)* produktu j : $ref_j \in \{2, 4, \dots, 100\}$ z procentowym prawdopodobieństwem wystąpienia: 40% między 0 a 20, 16% między 22 a 30, 12% między 32 a 40, 16% między 42 a 60 i 16% między 62 a 100. Cena produktu j ze sklepu i : $p_{ij} \in [a_{ij}, b_{ij}]$, gdzie $a_{ij} \geq 0, 75ref_j$, $b_{ij} \leq 1, 36ref_j$, a struktura interwałów pomiędzy $[a_{ij}, b_{ij}]$ jest następująca:

[8%]	<i>minimum</i>
	$minimum + \frac{(ref - minimum)}{4}$
[3%]	$minimum + \frac{(ref - minimum)}{2}$
[9%]	$minimum + \frac{(ref - minimum)}{1.25}$
[21%]	<i>ref</i>
[24%]	$ref + \frac{(maximum - ref)}{4}$
[9%]	$ref + \frac{(maximum - ref)}{2}$
[10%]	$ref + \frac{(maximum - ref)}{1.25}$
[16%]	<i>maximum</i>

Każdy sklep pobiera opłatę za dostawę wynoszącą od 0 do 20.

Przy pomocy wcześniej opisanej metody generowania instancji stworzyliśmy trzy zestawy danych różnej wielkości.

- Mały zestaw, z trzema podzbiorami: 3n20m, 4n20m i 5n20m, każdy złożony z 30 przypadków.
- Zestaw średni, z trzema podzbiorami: 5n240m, 5n400m i 50n240m, każdy złożony z 20 przypadków.
- Zestaw duży, z trzema podzbiorami: 50n400m, 100n240m i 100n400m, każdy również z 20 przypadkami problemu.

Jak już wspomniano wcześniej (podsekcja 4.4.3), w celu ustalenia poprawności ILP rozwiązano przypadki z zestawu *małego* metodą wyliczeniową i metodą ILP oraz porównano wyniki uzyskane obiema metodami w celu sprawdzenia, czy osiągnęły to samo rozwiązanie (o tej samej wartości końcowej).

W drugim teście *średnie* i *duże* zestawy zostały rozwiązane za pomocą ILP bez ograniczenia czasowego, aby uzyskać optymalne wartości we wszystkich przypadkach. Następnie, znając wartości optymalne, porównano wydajność proponowanych metod rozwiązania MINMIN+LOCAL SEARCH i CELLULAR z prostymi heurystykami GREEDY i FORECASTING.

Ostatni test przeprowadzono na instancji składającej się z rzeczywistych danych uzyskanych ze sklepów internetowych. Za cel obrano możliwość zbadania skuteczności tej metody w scenariuszu realnego użytkowania. Na zestaw danych składało się 400 różnych produktów z 57 sklepów prezentując łącznie 6387 ofert – nie każdy sklep był w stanie zaoferować wszystkie produkty.

ILP zostało zaimplementowane w języku programowania Java, wersja 1.7.0_45 przy użyciu API dostarczonego przez pakiet optymalizacji CPLEX, 12.1 opracowany przez IBM.

Algorytm zachłanny GREEDY, prognozowania FORECASTING, przetwarzania komórkowego CELLULAR oraz algorytm MIN-MIN zostały opracowane przy użyciu języka programowania skryptowego PHP, ponieważ pierwotnie został on zaproponowany w literaturze jako podstawowy język kodowania dla testowanych aplikacji internetowych ISOP.

Specyfikacja urządzenia, na którym przeprowadzono eksperymenty obliczeniowe, obejmuje procesor Intel i5 o częstotliwości 2,3 GHz z 8 GB pamięci RAM DD3 przy częstotliwości 1333 MHz, oraz dysk SSD o pojemności 256 GB.

Warto nadmienić, że generator instancji testowych dla problemu ISOP jest otwarcie dostępny na stronie internetowej¹.

4.6 Wyniki eksperymentów obliczeniowych – analiza i dyskusja

W niniejszej sekcji przedstawiono i skomentowano wyniki uzyskane w trakcie realizacji opisanych powyżej eksperymentów obliczeniowych.

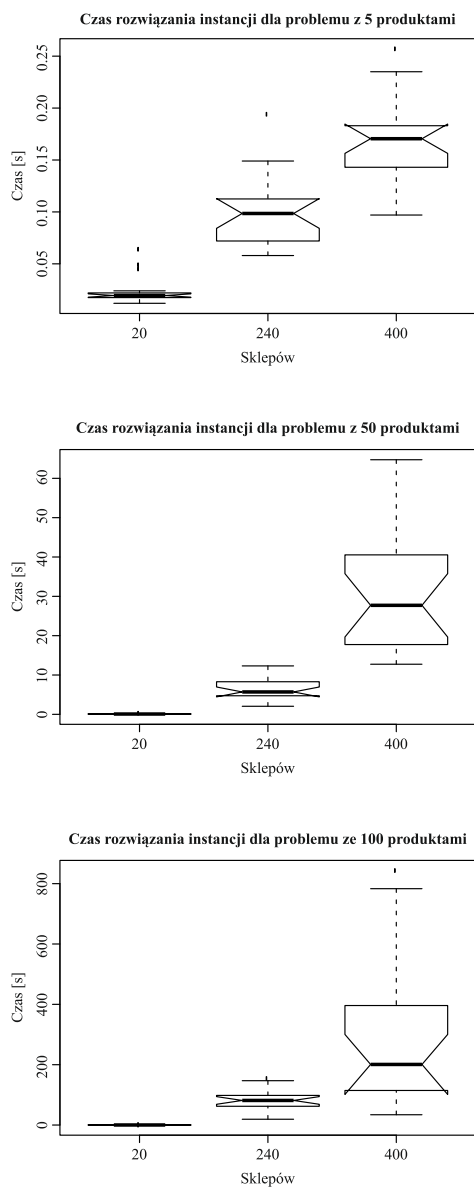
W przypadku ILP wybrano rodziny zestawów produktów 5n, 50n i 100n spośród dostępnych w modelu danych zestawów *Small*, *Medium* i *Large*. Zestaw ten został użyty do pomiaru czasu wymaganego przez ILP do rozwiązania instancji o różnych rozmiarach.

Zachowanie algorytmu ILP można prześledzić na rysunku 4.3, przedstawiającym wykres pudełkowy (ang. box plot). Na wykresie obserwacje zostały podzielone na kwartyle, w których wąsy (ang. chart whiskers) reprezentują obserwacje minimalne i maksymalne, z wyłączeniem wartości odstających.

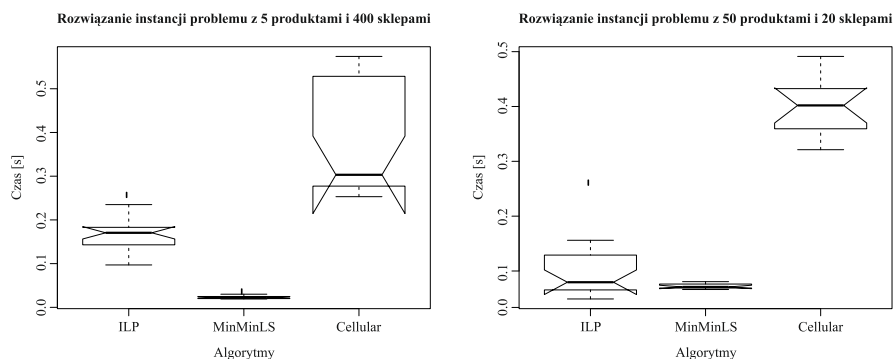
Pudełko (ang. box) przedstawia centralne kwartyle, tj. 50% występowania obserwacji, gdzie wielkość pudełka wskazuje, jak rozrzut danych zawiera się w podanym zakresie. Mediana jest wyznaczona linią poprowadzoną wewnątrz pudełka, z uwzględnieniem obserwacji między wąsami.

Nacięcia w polach zostały dodane, aby wskazać, że mediany podzbiorów są statystycznie różne. Jeśli różnice nie są znaczące, nacięcia pokrywają się z nacięciami pozostałych podzbiorów.

¹<http://www.cs.put.poznan.pl/ishop/generator.php>



Rysunek 4.3. Czasy uzyskania rozwiązania przy użyciu algorytmu ILP



Rysunek 4.4. Porównanie czasu działania metod ILP, MINMIN+LOCAL SEARCH i CELLULAR

Dla zestawu 5n mediana czasu wykorzystanego przez ILP do rozwiązywania każdej instancji jest bliska 0,25 sekundy. Dla zestawu 50n czas obliczeń wzrasta aż do minuty w przypadku instancji podzbioru 50n400m (przy czasie mediany bliskim 30 sekund). Natomiast dla podzbioru 50n240m wynosi mniej niż 10 sekund. W przypadku zestawu 100n maksymalny czas rozwiązania instancji wynosił 800 sekund dla podzbioru 100n400m, natomiast w większości przypadków czas potrzebny na rozwiązanie instancji wynosił blisko 200 sekund.

Test z danymi rzeczywistymi pobranymi ze sklepów internetowych (takimi jak ceny, koszty dostawy i stany magazynowe każdego sklepu) wykazuje bardzo dobre działanie modelu ILP. W ciągu 1,05 sekundy uzyskano optymalne rozwiązanie dla przykładu 400 produktów i 52 sklepów.

Wykorzystanie ILP w rozwiązywaniu rzeczywistych instancji ISOP przedstawiono na rysunku 4.4. Można na nim zaobserwować zestawy instancji, które są rozwiązywane przez ILP, MINMIN+LOCAL SEARCH i algorytm przetwarzania komórkowego w czasie krótszym niż jedna sekunda. Zauważono ponadto, że ILP jest w stanie dorównać (lub nawet prześcignąć) do czasu przetwarzania dwóch pozostałych proponowanych metod.

W przypadku większych zestawów danych preferowane jest wykorzystanie heurystyki, ponieważ wydłużenie czasu obliczeń ILP jest wyraźnie większe niż w przypadku heurystyk. Metody CELLULAR i MINMIN+LOCAL SEARCH są zatem bardziej odpowiednie do rozwiązywania większych instancji przy użyciu mniejszej ilości zasobów obliczeniowych. To ostatnie spostrzeżenie jest krytyczne, biorąc pod uwagę kontekst, w jakim miałyby być realizowane i wdrażane problem się ISOP (tysiące jednoczesnych zgłoszeń obliczeń do serwera).

Po ustaleniu poprawności proponowanego modelu ILP i uzyskaniu danych rozwiązań z optymalnymi wartościami ze *średnich* i *dużych* zestawów, przystą-

piono do oceny wydajności rozwiązań obliczanych przez algorytmy heurystyczne.

Przeprowadzono eksperyment obliczeniowy, w którym cztery prezentowane metody (MIN-MIN, MINMIN+MidLOCAL SEARCH, MINMIN+LOCAL SEARCH i CELLULAR) zostały użyte do rozwiązania problemów *średnich* i *dużych* zbiorów – działały aż do osiągnięcia własnych warunków stopu. Uzyskane wyniki przedstawiono na rysunku 4.5 w przypadku zestawu *średniego* oraz na rysunku 4.6 w przypadku zestawu *dużego*. Na wykresach oś x reprezentuje wielkość instancji zbioru. Na osi y po lewej stronie oznaczono funkcję celu, a czas obliczeń znajduje się po prawej stronie wykresu. Ponieważ ISOP jest problemem minimalizacji (kosztu), niższe wartości funkcji celu (podobnie jak w przypadku czasu) uważane są za lepsze.

W zestawie *Medium*, pokazanym na rysunku 4.5, wszystkie heurystyki prezentują podobne rozwiązania pod względem wartości funkcji celu dla podzbiorów 5n240m i 5n400m. Algorytm przetwarzania komórkowego jest wolniejszy niż heurystyka oparta na algorytmie MIN-MIN.

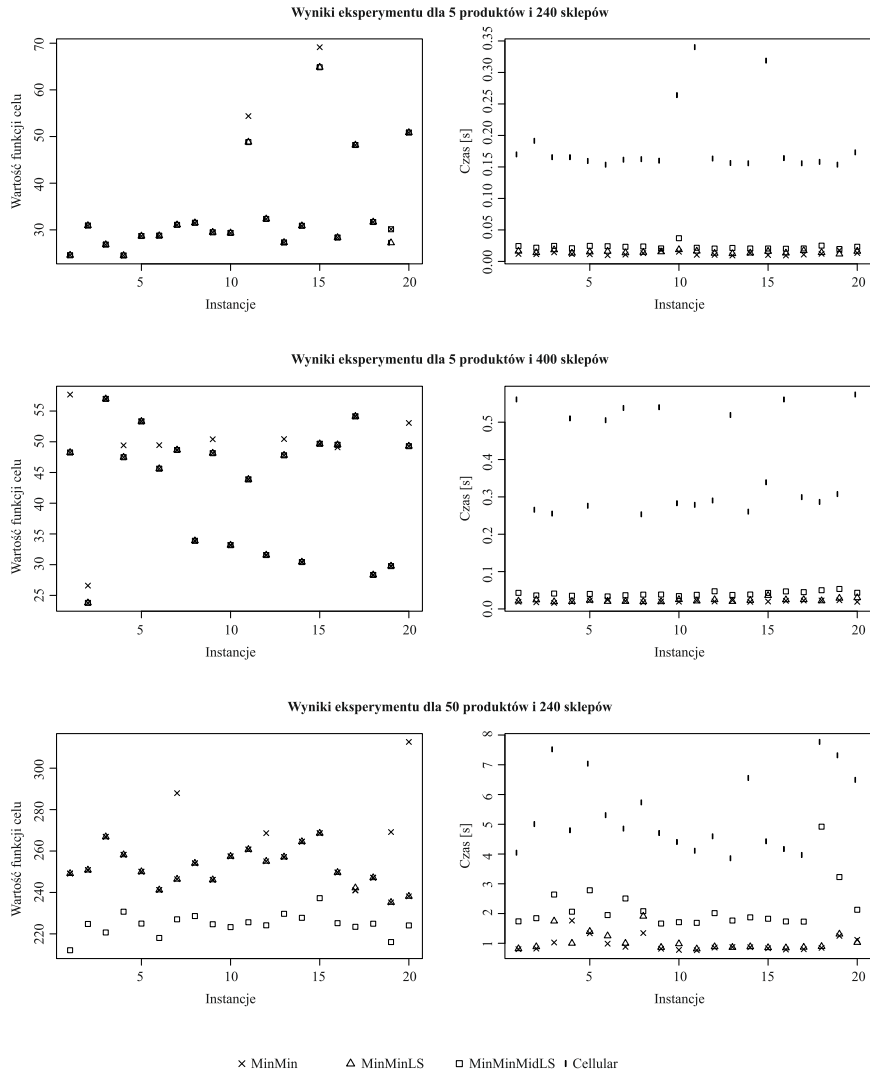
W przypadku zestawów 50 i 100 produktów zaczynamy dostrzegać, że heurystyka przetwarzania komórkowego daje lepsze rozwiązania jakościowe niż heurystyka oparta na MIN-MIN. Niestety dzieje się to kosztem większego czasu obliczeń.

W przypadku heurystyki opartej o MIN-MIN, wersje, które włączyły do procesu algorytm lokalnego przeszukiwania również uzyskały lepsze rozwiązania. Były bardziej spójne pod względem jakości otrzymywanych rozwiązań w porównaniu z podstawowym MIN-MIN. Nie zanotowano jednak istotnych różnic, które uzasadniałyby włączenie algorytmu lokalnego przeszukiwania do procesu budowy, czy też jako analitycznego kroku finalnego wykorzystywanego do już zbudowanego rozwiązania. Najistotniejszą różnicą okazał się być czas potrzebny każdej z heurystyk do rozwiązania instancji. Wyszukiwanie MINMIN+LOCAL SEARCH zajęło nawet połowę mniej czasu potrzebnego na obliczenia algorytmowi MINMIN+MidLOCAL SEARCH.

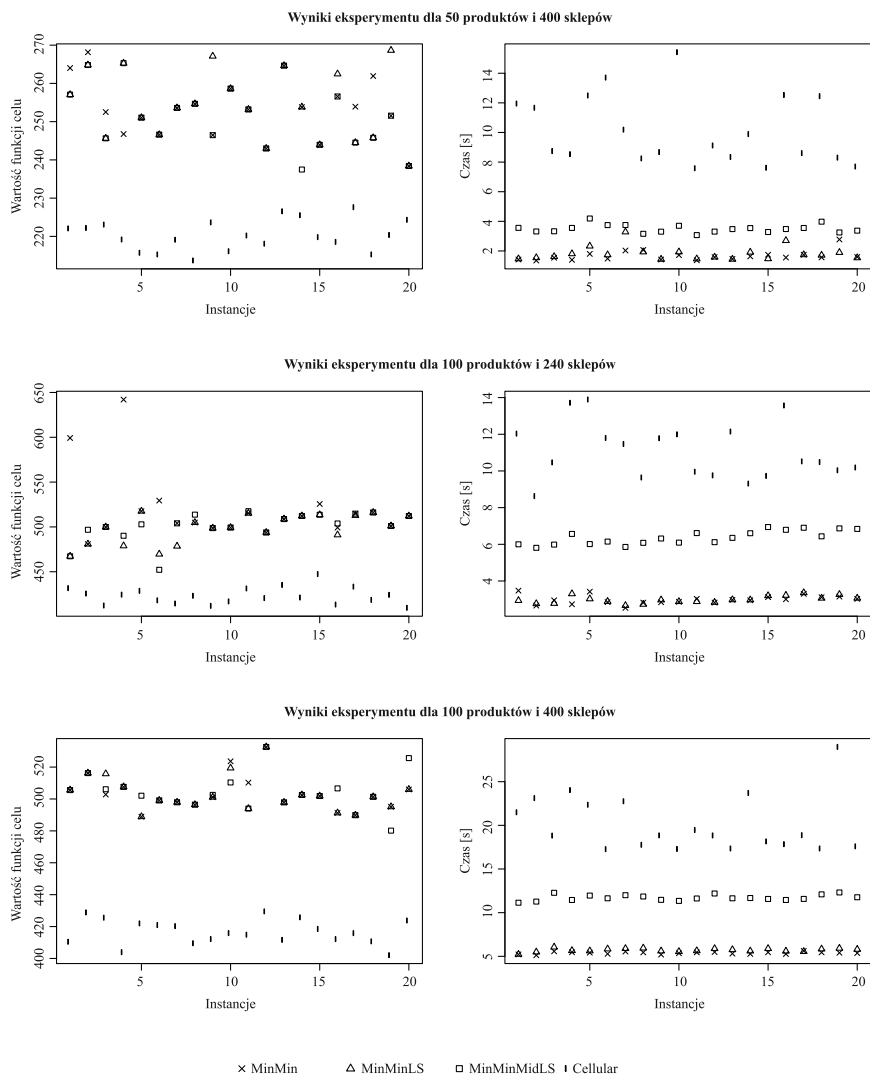
Na podstawie wszystkich przeprowadzonych badań i wyników wybrano algorytmy MINMIN+LOCAL SEARCH i CELLULAR, aby porównać je pod względem jakości rozwiązania z prostszymi rozwiązaniami GREEDY i FORECASTING. Wyniki tego eksperymentu zaprezentowano na rysunku 4.7. Oś x reprezentuje instancję problemu z zestawów *średniego* lub *dużego*. Oś y wskazuje błąd procentowy odchylenia od wartości optymalnej, reprezentowany przez linię przerywaną. Wartości bliższe zeru są lepsze (0 oznacza wartość optymalną, a np. 10% oznacza rozwiązanie 10% gorsze / droższe od optymalnego).

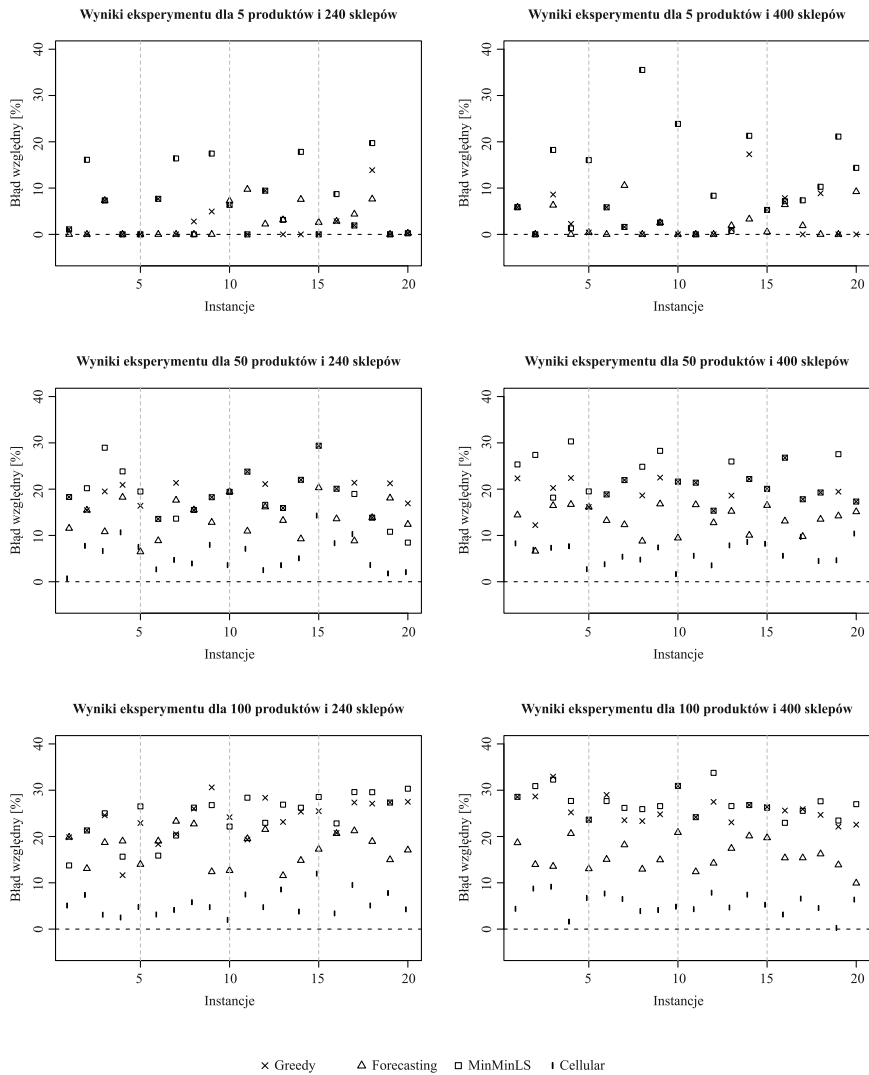
W przypadku podzbiorów 5n240m i 5n400m z zestawu *średniego* można zaobserwować, że w większości przypadków wszystkie heurystyki są w stanie osiągnąć optymalne wartości.

W przypadku instancji z dużego zbioru obserwujemy, że różnice pomiędzy heurystykami zwiększają się wraz ze wzrostem rozmiaru instancji. Co więcej, algorytmy uzyskujące w ogólności najlepsze wyniki, to odpowiednio algorytm



Rysunek 4.5. Wyniki uzyskane przez heurystyki dla *średniego* zestawu danych

Rysunek 4.6. Wyniki uzyskane przez heurystyki dla *dużego* zestawu danych



Rysunek 4.7. Współczynnik odchylenia [%] względem rozwiązania optymalnego

przetwarzania komórkowego CELLULAR i algorytm FORECASTING. Warto również zauważyć, że odchylenie od optymalnej wartości (*procentowy błąd*) dla algorytmu przetwarzania komórkowego konsekwentnie mieści się w zakresie od zera do dziesięciu procent, niezależnie od wielkości rozwiązywanego zestawu instancji. Jest to bardzo ważna obserwacja świadcząca o dużej stabilności algorytmu.

4.7 Podsumowanie

W niniejszym rozdziale przedstawiono problem optymalizacji zakupów internetowych (ang. Internet Shopping Optimization Problem), który obrazuje realistyczną sytuację, gdzie klient chce zrealizować listę zakupów korzystając z oferty sklepów internetowych. Celem jest dokonanie zakupów przy najniższym możliwym rachunku końcowym biorąc pod uwagę różne oferty, jak i zróżnicowane koszty dostaw.

W ramach prac nad problemem dowiedziono jego trudności obliczeniowej (przedstawiono dowód przynależności do klasy problemów silnie NP-trudnych). Następnie opracowano szereg algorytmów rozwiązujących ten problem. Działanie i jakość algorytmów zostały zbadane na podstawie wyczerpujących i złożonych eksperymentów obliczeniowych. Co istotne, do badań zostały użyte realistyczne modele danych (przygotowano generator oparty o złożoną analizę cen rynków). Ponadto, w niektórych przypadkach dokonano analizy używając zestawu wielu rzeczywistych danych pobranych z oferty sklepów internetowych. Eksperymenty zostały dogłębnie przeanalizowane i szeroko skomentowane.

Warto zauważyć, że problem ISOP może być jeszcze bardziej skomplikowany po dodaniu kolejnych funkcji, czy atrybutów. Będą one czynić problem jeszcze bliższym aktualnym wymaganiom użytkownika i pozwolą jeszcze lepiej odwzorować sytuację panującą na rynku zakupów internetowych.

Rozdział 5

Problem optymalizacji zakupów internetowych z rabatami

W sklepach internetowych często stosuje się zniżki / rabaty zależne od łącznej wartości zakupów. Ma to w prosty sposób przyciągnąć klienta i zachęcić do dokonania większych zakupów (nierzadko znacznie większych niż początkowo planowane). Typowym przykładem jest następująca praktyka panująca w sklepie internetowym sprzedającym części komputerowe: “Jeśli wartość twojego zamówienia wynosi co najmniej 1000 zł, to twój rabat wynosi 2%; 2000 zł – 5%; 3000 zł – 10%, 5000 zł – 15%; 10000 zł – 20%.”. Progi i kwoty są oczywiście podane jedynie w celu zobrazowania sytuacji i mogą nie odpowiadać w pełni wartościom stosowanym przez ulubiony sklep czytelnika. Ponadto, wiele sklepów internetowych oferuje tańsze lub bezpłatne dostawy, jeżeli cena zakupu przekracza próg.

W niniejszym rozdziale przedstawiono i zbadano nową odmianę oryginalnego problemu ISOP. Dotyczyć ona będzie wpływu kosztów zakupów dokonywanych w danym sklepie na rabaty jakie możemy otrzymać. Z matematycznego punktu widzenia czyni to problem znacznie bardziej skomplikowanym, gdyż jak łatwo zauważyć ceny produktów nie są stałe, a zmieniają się dynamicznie w obrębie każdego sklepu. Nazwijmy tę odmianę problemu, problemem optymalizacji zakupów internetowych ze zniżkami (ang. Internet Shopping Optimization Problem with Discounts, ISOPwD [10, 9]).

5.1 Definicja formalna – model matematyczny

Zdefiniujmy problem, w którym klient chciałby nabyć dany zestaw produktów $N = \{1, \dots, n\}$ korzystając z oferty sklepów internetowych $M = \{1, \dots, m\}$ przy minimalnej całkowitej cenie końcowej. Jako dane i odpowiednie parametry przyjmijmy:

d_i – cena dostawy wszystkich produktów ze sklepu i do klienta;

p_{ij} – podstawowa cena produktu j w sklepie i , $p_{ij} = p_j$ jeśli ceny produktu j są takie same we wszystkich sklepach;

N_i – podzbiór produktów zestawu N w sklepie i (produkty kwalifikujące się do zakupu i), $N_i \subseteq N$;

M_j – podzbiór sklepów, w których można kupić produkt j (kwalifikujące się sklepy dla produkt j), $M_j \subseteq M$;

S_i – podzbiór produktów wybranych przez klienta w sklepie i (koszyk sklepu i , zmienna decyzyjna), $N = \cup_{i=1}^m S_i$, oraz $S_i \cap S_j = \emptyset$, $i \neq j$, dla możliwego rozwiązania;

$T_i(S_i) = d_i + \sum_{j \in S_i} p_{ij}$ – łączny koszt dostawy i cen produktów w sklepie i dla określonych produktów $S_i \subseteq N_i$;

$f_i(T)$ – funkcja opisująca rabaty dla danego sklepu i . Dotyczy łącznej ceny podstawowej T w sklepie i dla wszystkich $T > 0$, $f_i(0) = 0$.

Powyższy problem oznaczamy jako ISOPwD. Sformułowanie matematyczne można zapisać w następujący sposób:

$$\min \sum_{i=1}^m f_i(d_i y_i + \sum_{j \in N_i} p_{ij} x_{ij}) \quad (5.1)$$

$$\text{p.o.} \quad \sum_{i \in M_j} x_{ij} = 1, \quad j = 1, \dots, n, \quad (5.2)$$

$$0 \leq x_{ij} \leq y_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.3)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.4)$$

$$y_i \in \{0, 1\}, \quad i = 1, \dots, m. \quad (5.5)$$

W kolejnej części rozdziału przedstawiono kilka specyficznych odmian problemu ISOPwD. Następnie zaprezentowano algorytmy rozwiązujące problem ISOPwD. Przeprowadzone zostały również rozbudowane eksperymenty obliczeniowe, a ich wyniki i przebieg zostały poddane analizie uzupełnionej o wyczerpujący komentarz.

5.2 Niepełna oferta produktów

W tej sekcji przeanalizowano ogólny przypadek problemu ISOPwD, w którym $N_i \neq N$ może być spełniony dla $i \in \{1, \dots, m\}$. Po pierwsze, przeprowadzono analizę złożoności obliczeniowej.

Twierdzenie 2 *Problem ISOPwD jest silnie NP-trudny, jeśli $N_i \neq N$ dla $i \in \{1, \dots, m\}$, $p_{ij} = 1$, $j = 1, \dots, n$, $d_i = 1$, $f_i(T) = T$, $i = 1, \dots, m$, każdy produkt pojawia się w ofercie maksymalnie trzech sklepów, z których każdy oferuje dokładnie trzy produkty.*

Dowód 2 *Wersja decyzyjna powyżej zaprezentowanej odmiany, w której pytamy o istnienie rozwiązania z wartością nie przekraczającą $4n/3$ może zostać zapisana w następujący sposób. Mając na uwadze rodzinę $C = \{C_1, \dots, C_m\}$ 3-elementowych podzbiorów (sklepów) zbioru $N = \{1, \dots, n\}$ (produkty do zakupu), gdzie $n = 3k$, czy C zawiera dokładne pokrycie zbioru N , tj. podrodzina $Y \subseteq C$ (wybór sklepów) taka, że każdy $j \in N$ należy do dokładnie jednego 3-elementowego zbioru w Y ?*

Należy zauważyć podobieństwo powyższego zapisu ze sformułowaniem problemu EXACT COVER BY 3-SETS [33], który jest NP-zupełny w silnym sensie, nawet jeśli każdy element $j \in N$ pojawia się w co najwyżej trzech podzbiorach C_i . ■

Warto pamiętać, że jeśli każdy produkt pojawia się w co najwyżej dwóch sklepach, to ten specjalny przypadek zaprezentowany w twierdzeniu 2 jest rozwiązywalny w czasie wielomianowym [33].

Problem ISOPwD można rozwiązać poprzez następujące podejście wyliczeniowe. Skonstruuj wszystkie n -wymiarowe sekwencje $(i_1, \dots, i_n)$ sklepów, gdzie $i_j \in M_j$ jest sklepem, w którym można kupić produkt j . Pomiedzy tymi sekwencjami i rozwiązaniem problemu ISOPwD istnieje bezpośrednia zależność. Liczba sekwencji nie przekracza m^n . W czasie $O(n)$ można obliczyć wartości rozwiązania (łączna suma zakupów) dla każdej sekwencji i wybrać najlepsze rozwiązanie jako optymalne.

Złożoność czasowa tego podejścia wynosi $O(nm^n)$. Podejście można określić jako wielomianowe, jeśli liczba produktów n jest stała. Może to być akceptowalne do wykorzystania w środowisku online, gdzie szybki czas odpowiedzi jest niezmiernie istotny. Obliczenia wskazują, że podejście może być odpowiednie dla niewielkich rozmiarów problemów – około dziesięciu produktów i około dziesięciu sklepów.

Identyczne ceny standardowe Rozważmy teraz odmianę problemu ISOPwD, w której $p_{ij} = p_j$, $j = 1, \dots, n$, $i = 1, \dots, m$. Oznaczamy ją jako IS-SAME-PRICES. Problem ten ma następujące własności.

Twierdzenie 3 *Istnieje optymalne rozwiązanie problemu IS-SAME-PRICES, w którym wszystkie produkty są kupowane w sklepach $M_{i_1}, M_{i_2}, \dots, M_{i_h}$ takie, że w sklepie M_{i_1} kupowane są wszystkie odpowiednie produkty zbioru N_{i_1} , w sklepie M_{i_2} wszystkie odpowiednie produkty z wyjątkiem tych wybranych w sklepie M_{i_1} i tak dalej.*

Dowód 3 *Zastanówmy się nad rozwiązaniem optymalnym. Załóżmy, że produkty są kupowane w sklepach $1, \dots, k$, $k \geq 2$ i że odpowiadające łączne ceny dostawy i produktów wynoszą $T_1, \dots, T_k$. Niech sklep i^0 ma minimalną wartość instrumentu pochodnego $f'_i(T_i)$ spośród $i = 1, \dots, k$ (w przypadku wklęsłych, fragmentarycznych funkcji liniowych f_i , minimalne nachylenie segmentu linii $[T_i, T_i + \varepsilon]$ dla arbitralnie małej wartości dodatniej $\varepsilon > 0$). Jeśli produkty z zestawu N_{i^0} wybrano nie tylko w sklepie i^0 , ale także w innych sklepach, przenieś je do koszyka sklepu i^0 . Ze względu na wklęsłość funkcji f_i , relacja $f_{i^0}(T_{i^0} + \delta) - f_{i^0}(T_{i^0}) \leq f_i(T_i) - f_i(T_i - \delta)$ jest spełniona dla każdego $\delta > 0$, oraz $i = 1, \dots, k$. W związku z powyższym przemieszczenie produktu nie zwiększa całkowitej ceny końcowej. Dedukujemy, że rozwiązanie z $S_{i^0} = N_{i^0}$ jest optymalne. Powtórzenie tego argumentu dopełnia dowód. Niewielka różnica między kolejnymi iteracjami polega na tym, że produkty wybrane w poprzednich iteracjach nie są przenoszone. Odpowiednie pochodne lub nachylenia funkcji f_i dla tych sklepów są nie większe niż te dla niewybranych sklepów z powodu założenia wklęsłości. ■*

Twierdzenie 3 jest podstawą określenia następującego podejścia do problemu IS-SAME-PRICES.

Skonstruuj wszystkie sekwencje sklepów $(i_1, \dots, i_m)$ i odpowiadające im rozwiązania $(S_{i_1}, \dots, S_{i_k})$, $k \leq m$, takie że $S_{i_1} = N_{i_1}$, $S_{i_r} = N_{i_r} \setminus (\cup_{l=1}^{r-1} S_{i_l})$, $r = 1, \dots, k$ oraz $N = \cup_{l=1}^k S_{i_l}$. Liczba tych sekwencji nie przekracza $m!$. W czasie $O(n)$ można obliczyć wartość rozwiązania (kończącą cenę całkowitą) dla każdej sekwencji i wybrać rozwiązanie najlepsze jako optymalne.

Złożoność czasowa tego podejścia wynosi $O(nm!)$, co jest lepszym rozwiązaniem w porównaniu z poprzednim rozwiązaniem o złożoności czasowej $O(nm^n)$. Podejście to jest liniowe, wyrażone w n i może być akceptowalne w życiu codziennym dla tysięcy produktów i dziesiątek sklepów. Może również służyć jako heurystyka dla bardziej ogólnego przypadku problemu ISOPwD, w którym ceny tego samego produktu nieznacznie się różnią w różnych sklepach. Można je również zaadaptować w celu rozwiązania w czasie wielomianowym następującego, realistycznego przypadku problemu IS-SAME-PRICES.

Problem IS-SAME-PRICES-SUBSETS: $p_{ij} = p_j$, $j = 1, \dots, n$, $i = 1, \dots, m$, a N jest podzielone na podzbiory $N^{(0)}$, $N^{(1)}$ i $N^{(2)}$ tak, że wszystkie produkty zbioru $N^{(0)}$ są obecne w każdym sklepie, każdy produkt zbioru $N^{(1)}$ jest dostępny w jednym sklepie, a $|N^{(2)}| \leq c$, gdzie c jest daną stałą.

Twierdzenie 4 *Istnieje optymalne rozwiązanie problemu IS-SAME-PRICES-SUBSETS, w którym wszystkie produkty ze zbioru $N^{(0)}$ są kupowane w jednym sklepie.*

Dowód 4 *Dowód jest taki sam jak w przypadku twierdzenia 3.* ■

Mając na uwadze powyższe zaproponowano następujący algorytm rozwiązujący problem IS-SAME-PRICES-SUBSETS. Rozważmy, że należy wybrać jeden sklep spośród $l = 1, \dots, m$, aby kupić wszystkie produkty ze zbioru $N^{(0)}$. Dla każdego l należy wykonać następujące obliczenia. Przypisz produkty zbioru $N^{(0)}$ do sklepu l . Przypisz każdy produkt $j \in N^{(1)}$ do odpowiadającego sklepu. Niech $N^{(2)}\{1, \dots, c\}$. Wylicz wszystkie możliwe przyporządkowania produktów $1, \dots, c$ do sklepów, co jest odpowiednikiem wyliczenia sekwencji sklepów $(i_1, i_2, \dots, i_c)$ tak, że produkt j jest przypisany do sklepu $i_j \in M_j$, $|M_j| \leq m$, $j = 1, \dots, c$. Liczba tych sekwencji nie przekracza m^c . Dla każdego l i w/w sekwencji sklepowej $(i_1, i_2, \dots, i_c)$ należy obliczyć wartość odpowiedniego rozwiązania. Wybierz rozwiązanie o minimalnej wartości jako optymalne. Czas działania tego algorytmu wynosi $O(nm^{c+1})$.

5.3 Pełna oferta produktów we wszystkich sklepach

W niniejszej sekcji studiujemy podgrupę problemu ISOPwD, w której $N_i = N$, $i = 1, \dots, m$. Oznaczmy ją jako IS-ALL-PRODUCTS. Założenie to nie oznacza, że każdy sklep oferuje ten sam asortyment. Oznacza jedynie, że każdy sklep posiada wszystkie produkty ze zbioru N wymagane przez klienta, co może się zdarzyć w wielu rzeczywistych sytuacjach. Problem jest silnie NP-trudny co demonstrują poniższe twierdzenie i dowód.

Twierdzenie 5 *Problem IS-ALL-PRODUCTS jest silnie NP-trudny, jeśli $N_i = N$, $p_{ij} \in \{1, 2\}$, $d_i = 1$, $f_i(T) = T$, każdy produkt j ma najwyżej trzy wartości $p_{ij} = 1$, a każdy sklep i ma dokładnie trzy produkty o wartości $p_{ij} = 1$, $i = 1, \dots, m$, $j = 1, \dots, n$.*

Dowód 5 *Rozważmy następujący problem: Czy istnieje rozwiązanie dla szczególnego przypadku twierdzenia 5 o wartości co najwyżej $4n/3$? Decyzyjna wersja problemu ISOPwD w zawarta w dowodzie twierdzenia 2 ewidentnie się do obecnego redukuje poprzez wybranie $p_{ij} = 1$ jeśli produkt j jest obecny w sklepie i a $p_{ij} = 2$ w przeciwnym razie.* ■

5.3.1 Identyczne ceny standardowe

Rozważmy podgrupę problemu ISOPwD, w której $N_i = N$, oraz $p_{ij} = p_j$, $j = 1, \dots, n$, $i = 1, \dots, m$. Oznaczamy ją jako IS-SAME-PRODUCTS-AND-PRICES. W

tym podproblemie całkowita cena standardowa wszystkich produktów nie zależy od rozwiązania. Oznaczmy ją jako $P = \sum_{j=1}^n p_j$. Problem IS-SAME-PRODUCTS-AND-PRICES posiada następującą właściwość.

Twierdzenie 6 *Istnieje optymalne rozwiązanie problemu IS-SAME-PRODUCTS-AND-PRICES, w którym wszystkie produkty są kupowane w jednym sklepie.*

Dowód 6 *Dowód jest taki sam jak w przypadku twierdzenia 3. Jedyna różnica polega na tym, że skoro $N_i = N$ dla każdego $i = 1, \dots, m$, za optymalne rozwiązanie z $k \geq 2$ sklepami uznajemy takie, że wszystkie produkty można przenieść do sklepu i^0 przy minimalnej wartości instrumentu pochodnego $f'_i(T_i)$ między $i = 1, \dots, k$ (w przypadku wklęsłych, fragmentarycznych funkcji liniowych f_i , minimalne nachylenie segmentu linii $[T_i, T_i + \varepsilon]$ za arbitralnie małe $\varepsilon > 0$).* ■

Problem IS-SAME-PRODUCTS-AND-PRICES można rozwiązać w następujący sposób. Wyznacz sklep $i^0 = \arg \min \{f_i(d_i + P) \mid i = 1, \dots, m\}$. Skonstruuj optymalne rozwiązanie, w którym wszystkie produkty są wybierane w sklepie i^0 . Złożoność czasowa tego podejścia wynosi $O(n + m)$.

5.3.2 Proporcjonalne ceny standardowe z rabatami

W niniejszym przypadku zakładamy, że $N_i = N$, a funkcja rabatująca: $f_i(T) = a_i^{(r)} + b_i^{(r)}T$ dla $T \in (T_{r-1}, T_r]$, $r \in \{1, \dots, k\}$, $0 = T_0 < T_1 < T_2 < \dots < T_k$. Standardowe ceny w różnych sklepach są proporcjonalne: $p_{ij} = h_i p_j$. Dodatkowo $a_i^{(r)} \geq 0$, $b_i^{(r)} > 0$ i $h_i > 0$, $r = 1, \dots, k$, $i = 1, \dots, m$. Taka struktura cen standardowych często się zdarza w rzeczywistości, gdy sklepy o ugruntowanej renomie stosują dodatkowe opłaty w stosunku do ceny produktu w porównaniu do sklepów cieszących się zwykłą renomą – problem zostanie szeroko poruszony w rozdziale 7 dotyczącym analizy czynników zaufania.

Niniejszy podproblem oznaczmy jako IS-SAME-PRODUCTS-LINEAR-CHARGES. Dla tego przypadku, końcową całkowitą cenę produktów wybranych w sklepie i można obliczyć w następujący sposób:

$$f_i(T_i(S_i)) = a_i^{(r)} + b_i^{(r)} \left(d_i + h_i \sum_{j \in S_i} p_j \right) = a_i^{(r)} + b_i^{(r)} h_i \left(d_i/h_i + \sum_{j \in S_i} p_j \right)$$

Ten wzór pokazuje, że problem IS-SAME-PRODUCTS-LINEAR-CHARGES redukuje się do problemu IS-SAME-PRODUCTS-AND-PRICES z wypukłymi liniowymi funkcjami rabatów $g_i(T) = a_i^{(r)} + b_i^{(r)} h_i T$ for $T \in (\frac{T_{r-1}}{h_i}, \frac{T_r}{h_i}]$, $r \in \{1, \dots, k\}$, a cena dostawy wynosi d_i/h_i . Dlatego można go rozwiązać w czasie $O(n + m)$ za pomocą podejścia opisanego w podsekcji 5.3.1.

Ponieważ problemy lokalizacyjne są dobrze znane od kilkudziesięciu lat, analizując literaturę warto nawiązać do niektórych sformułowań, które choćby po części mogą być powiązane z problemem ISOP.

Przed wszystkim warto zauważyć problem lokalizacji fabryk (FLP) z wklęsłymi kosztami transportu towarów [95]. Główne sformułowanie problemu jest następujące. Istnieje firma, która chce sprzedać pewną liczbę jednostek r_j danego (jednego) produktu. Przyjmuje się, że okres ten wynosi jeden rok. Klientami są miasta j , $j = 1, \dots, n$. Cała produkcja w wysokości r jednostek jest dystrybuowana z zakładów i , $i = 1, \dots, m$. x_{ij} to roczna ilość towaru, który ma być wysłany z zakładu i do miasta j , a $y_i = \sum_{j=1}^n x_{ij}$ to roczna produkcja w zakładzie i . Niech a_i , $a_i = 1, \dots, m$ będzie maksymalną pojemnością zakładu i i $t_{ij}(x_{ij})$ będzie łącznym kosztem rabatu w czasie T -lat uzyskanym dla dostawy x_{ij} . Sformułowanie matematyczne problemu można zapisać jako

$$\min \sum_{i=1}^m f_i(y_i) + \sum_{i=1}^m \sum_{j=1}^n t_{ij}(x_{ij}) \quad (5.6)$$

$$\text{p.o.} \quad \sum_{i=1}^m x_{ij} = r_j, \quad (5.7)$$

$$\sum_{j=1}^n x_{ij} = y_i \leq a_i, \quad (5.8)$$

$$x_{ij} \geq 0, y_i \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (5.9)$$

Mimo iż problem zawiera funkcję opisującą rabaty, nie można uznać, że jest tożsamy problemowi ISOP lub jeden jest wariacją na temat drugiego.

Innym ciekawym, w pewnym stopniu podobnym do ISOP, jest problem lokalizacji fabryk z kosztami stopniowymi (ang. staircase costs). Przykładowo możemy zdefiniować funkcję kosztów stopniowych jako skończoną, fragmentaryczną liniową funkcję rosnącą ze skończonym zestawem nieciągłości, z których każda odpowiada określonej wielkości fabryki. Sformułowanie problemu (patrz literaturę [42] w celu uzyskania szczegółowych informacji) jest liniowym sformułowaniem problemu MPI (ang. Mixed-Integer Programming) z ciągłymi zmiennymi $mn + mq$ i zmiennymi całkowitymi mq . W pracy Holmberga [42] autorzy badali metody rozwiązywania oparte na linearyzacji częściowo wypukłej (ang. convex piecewise linearization) i rozkładzie Bendera (ang. Benders decomposition). Przy zastosowaniu linearyzacji częściowo wypukłej wprowadza się tylko zmienne całkowite, które są potrzebne do poprawy aproksymacji. Testy obliczeniowe wykazują, że średnio potrzeba tylko kilku zmiennych. Autorzy pokazują, że dekompozycja Bendera może być użyta do rozwiązania powstałego problemu. Ponadto Holmberg i Ling [43] opracowali i porównali heurystyczne metody rozwiązywania problemów lokalizacji fabryk o dużej pojemności z funkcjami kosztów produkcji w kształcie schodowym. W tym samym artykule zaproponowano heurystykę Lagrangea, w tym relaksację i optymalizację gradientową (ang. Subgradient optimization), jako podstawę skutecznego działania heurystyki pierwotnej. Pokazali również, jak wykorzystać linearyzację częściowo wypukłą funkcji kosztowych o

kształcie schodowym, aby uzyskać dobre początkowe górne i dolne ograniczenia, jak również początkowe rozwiązania dualne. W oparciu o rozwiązanie relaksacji Lagrangea powstał problem transportowy, który daje pierwotne, dopuszczalne rozwiązania.

Widać jednak, że koszty schodowe (przedziałowe) i przedstawione problemy są nieco inne niż w przypadku problemu ISOPwD, więc potrzebne są różne podejścia.

5.4 Dokładny algorytm podziału i ograniczeń

W celu obliczenia optymalnego rozwiązania dla zadanych instancji zaimplementowano algorytm podziału i ograniczeń (Branch and Bound, B&B) [53], którego zasadę działania zobrazowano na algorytmach 7 i 8.

Algorytm 7 rozpoczyna działanie od obliczenia górnego ograniczenia (UB – upper bound) z wykorzystaniem rozwiązania zaproponowanego przez algorytm zachłanny G1 (sekcja 5.5). Następnie, przechodzi do podziału pierwszego poziomu drzewa wyszukiwania poprzez algorytm 8.

Kolejno, algorytm wybiera górny element stosu i ocenia obiektywną wartość, jaką miałyby, gdyby ten był częścią bieżącego rozwiązania. Jeśli rozwiązanie częściowe przekracza limit podany przez górne ograniczenie (UB) aktualna gałąź jest zgłębianą. W przeciwnym razie, jeśli obiekt nie był liściem, algorytm umieszcza następujące elementy w stosie. Gdyby obiekt był liściem, oznaczałoby to, że obecne rozwiązanie jest lepsze niż najlepsze globalne rozwiązanie dotychczas znalezione, tak więc zaktualizowałoby górne ograniczenie o nową, znaną wartość.

Proces ten trwa tak długo, jak długo na stosie znajdują się elementy. Oznacza to, że całe drzewo wyszukiwania zostało zbadane, a górne ograniczenie jest uważane za optymalne rozwiązanie całej instancji problemu.

Algorytm 7 *BranchandBound*

Require:*instancja*
rozwiazanie

```

1: UB = cellPILS(instancja, rozwiazanie)
2: branch(instancja, stos, 0)
3: ostatniProdukt = -1
4: repeat
5:   element = stos.pop()
6:   if ostatniProdukt ≥ element.produkt then
7:     for i = element.produkt to ostatniProdukt do
8:       odznaczSklep(instancja, rozwiazanie, i)
9:     end for
10:  end if
11:  wyznaczSklep(instancja, rozwiazanie, element.produkt, element.sklep)
12:  if element.produkt < instancja.produkt AND rozwiazanie.total <
    UB.total then
13:    if element.produkt == instancja.produkt - 1 then
14:      UB = rozwiazanie
15:    else
16:      branch(instancja, stos, element.produkt + 1)
17:    end if
18:  end if
19:  ostatniProdukt = element.produkt
20: until stos ≠ ∅
21: return UB

```

Algorytm 8 *Podzial*

Require:*instancja*
stos
int produkt
Sklepy – zbiór sklepów.

```

1: for sklep = instancja.sklepy - 1 to 0 do
2:   stos.push(sklep, produkt)
3: end for
4: return stos

```

5.5 Wstępne eksperymenty obliczeniowe

W celu wykonania pierwszych eksperymentów obliczeniowych dla problemu ISO-PwD postanowiono przetestować dwie proste heurystyki znane z wykorzystania w problemie ISOP. Badanie uzupełniono o rozwiązanie optymalne, jak i dwie odmiany algorytmu działającego w porównywarkach cen.

W pierwszym algorytmie heurystycznym oznaczonym jako G1, produkty rozpatrywane są w określonej kolejności. Algorytm jest uruchamiany dla różnych wariantów zbioru (różni się kolejność umieszczonych produktów) produktów. Niech produkty zostaną określone w kolejności $1, \dots, n$. Wartość całkowita kosztów dostawy i standardowych cen (ofert) dla wszystkich sklepów są początkowo ustalone na $T_i = d_i$, $i = 1, \dots, m$. W każdej iteracji algorytmu G1, produkt j jest wybrany w sklepie $i \in M_j$ przy minimalnej wartości $f_i(T_i + p_{ij})$, a odpowiednia wartość T_i jest ponownie przypisywana do $T_i := T_i + p_{ij}$.

Zauważono, że algorytm G1 wykazuje bardzo dobrą wydajność. Może jednak odnaleźć rozwiązanie, którego wartość jest n razy gorsza niż optymalna. Sytuacja rozstała zaprezentowana i opisana w poprzednim rozdziale (podsekcja 4.4.4).

Drugim przebadanym algorytmem jest algorytm G2, który posiada mechanizm zapobiegania niekorzystnej sytuacji występującej w przypadku algorytmu G1. Algorytm opisuje podsekcja 4.4.5.

Przeprowadzono eksperymenty obliczeniowe, w których porównano najlepsze rozwiązania uzyskane algorytmami G1 i G2 z rozwiązaniami optymalnymi oraz rozwiązaniami uzyskanymi przez algorytmy *porównywarek cen* [18, 40]. Dane dla obliczeń przygotowano na podstawie analizy ofert z branży księgarń internetowych [19]. Przyjęto, że $m \in \{10, 20, 30\}$, $n \in \{2, 3, 4, \dots, 15\}$, a funkcja rabatująca ma postać

$$f_i(P) = \begin{cases} P & \text{jeżeli } 0 < P \leq 50; \\ 50 + 0,95(P - 50) & \text{jeżeli } 50 < P \leq 100; \\ 50 + 0,95 * 50 + 0,9(P - 100) & \text{jeżeli } 100 < P \leq 150; \\ 50 + 0,95 * 50 + 0,9 * 100 + 0,85(P - 150) & \text{jeżeli } P > 150. \end{cases}$$

gdzie P oznacza łączną ceną standardową książek ze sklepu i . Przyjmuje się, że każdy sklep posiada kompletną ofertę produktów. Dla każdej pary (n, m) wygenerowano 10 instancji problemu. Dla każdej z nich utworzono następujące wartości dla każdego i i j w przedstawionych zakresach. Koszt dostawy: $d_i \in \{5, 10, 15, 20, 25, 30\}$, *rekomendowana* cena książki j : $r_j \in \{5, 10, 15, 20, 25\}$, cena książki j w sklepie i : $p_{ij} \in [a_{ij}, b_{ij}]$, gdzie $a_{ij} \geq 0,69r_j$, $b_{ij} \leq 1,47r_j$. Informacje uzupełnia tabela V zawarta w literaturze [19]. Dla każdej instancji problemu, algorytm G1 został uruchomiony dwukrotnie – produkty uszeregowano według rekomendowanej ceny rosnąco i malejąco. Natomiast algorytm G2 dokonywał obliczeń w czterech przebiegach – odpowiednio dla $\alpha = 25, 50, 75, 100$.

Tabele 5.1 i 5.2 zawierają średnie wartości rozwiązań (dla 50 instancji), które są uzyskiwane przez algorytmy G1 i G2, algorytm optymalny B&B, oraz dwa algorytmy stron porównujących ceny – podstawowy (PCS), oraz dodatkowo analizujący koszty wysyłki (PCS+).

Widać, że rozwiązanie znalezione przy wykorzystaniu algorytmu G1 było nieco gorsze od rozwiązania optymalnego dla mniejszej liczby produktów. Różnica pomiędzy G1 a B&B rośnie wraz ze wzrostem liczby produktów (patrz tabele 5.3 i 5.4 – od 101,51% dla 2 produktów poprzez 10 sklepów, do 110,55% dla 15 produktów kupowanych z 30 sklepów). Wszystkie wartości (dla każdej pary (n, m)) są reprezentowane jako wartości średnie z 50 niezależnych testów.

Algorytm G2 oblicza nieco niższe wartości niż G1 dla małej liczby produktów (do 5-8). Dla większej liczby produktów wyniki są gorsze od algorytmu G1.

Wyniki obliczeń algorytmu porównywarek cen PCS są znacznie gorsze od algorytmów G1 i G2. Algorytm PCS w rozbudowanej wersji uwzględniającej ceny dostawy (PCS+) jest znacznie lepszy niż algorytm podstawowy PCS. Jest jednak dużo gorszy od dwóch zaproponowanych algorytmów heurystycznych.

W ogólności dla uśrednionych wyników algorytmy G1 i G2 nie potrafią obliczyć optymalnych wartości rozwiązania (w pojedynczych przypadkach się to im udaje), jednak czasy obliczeń są znacznie szybsze niż dla algorytmu B&B. Dla większej liczby produktów/sklepów obliczenia dokładne są nawet 1000 razy wolniejsze. Algorytm G1 jest znacznie szybszy niż G2. Zależnie od rozmiaru instancji: od 2 do 9 razy.

Tabela 5.1. Uśrednione wartości zakupów dla różnych algorytmów. Dane dla 2-15 produktów i 10,20 sklepów

Produktów	Sklepów	G1	G2	PCS	PCS+	B&B
2	10	30,99	30,77	53,69	34,22	30,53
3	10	48,34	48,17	84,17	53,80	47,60
4	10	57,97	57,93	103,03	64,11	56,54
5	10	68,07	68,10	119,85	74,63	66,22
6	10	83,54	82,87	146,46	90,22	80,41
7	10	99,30	98,01	169,08	107,02	94,57
8	10	104,91	105,11	176,85	110,66	98,47
9	10	118,80	119,56	194,41	125,44	111,82
10	10	127,46	129,42	205,60	136,68	120,41
11	10	140,22	139,03	235,38	148,45	128,02
12	10	158,00	156,07	251,98	163,43	142,26
13	10	167,01	168,40	255,19	174,71	152,10
14	10	178,27	178,18	283,84	186,49	160,96
15	10	184,68	188,83	292,74	195,59	168,60
2	20	31,45	31,04	57,29	33,35	30,86
3	20	42,71	42,16	80,25	47,03	41,49
4	20	57,25	55,63	104,11	64,40	55,10
5	20	70,86	69,96	129,98	78,41	67,97
6	20	83,16	81,86	153,77	92,04	79,38
7	20	95,93	95,16	183,04	102,26	90,64
8	20	106,69	107,47	206,19	112,64	100,82
9	20	112,98	112,64	213,00	121,93	105,62
10	20	128,02	129,38	250,51	134,43	119,72
11	20	137,04	137,41	255,22	143,81	124,39
12	20	147,93	151,11	279,03	158,33	136,35
13	20	169,80	172,89	316,73	180,45	156,18
14	20	175,73	176,19	310,30	184,69	158,80
15	20	184,48	185,42	342,62	192,21	163,76

Tabela 5.2. Uśrednione wartości zakupów dla różnych algorytmów. Dane dla 2-15 produktów i 30 sklepów

Produktów	Sklepów	G1	G2	PCS	PCS+	B&B
2	30	35,05	34,23	59,91	37,17	34,08
3	30	43,32	42,46	79,06	48,11	42,10
4	30	55,04	53,93	109,89	60,11	52,59
5	30	65,82	65,88	131,69	71,95	63,76
6	30	80,60	77,92	148,62	85,20	76,10
7	30	93,59	91,79	184,30	101,36	87,81
8	30	103,23	102,37	211,37	111,97	96,79
9	30	116,54	118,14	240,84	127,22	109,16
10	30	128,86	130,79	241,40	139,31	120,84
11	30	139,30	141,14	271,44	150,68	129,00
12	30	149,33	150,99	301,28	156,97	136,83
13	30	169,24	170,06	329,62	180,69	152,67
14	30	176,25	177,83	349,03	183,10	158,26
15	30	179,76	183,80	357,24	193,08	162,60

Tabela 5.3. Średnie wartości proporcjonalne (w procentach odległości od wartości optymalnej). Dane dla 2-15 produktów i 10, 20 sklepów. 100% oznacza wartość optymalną

Produktów	Sklepów	$\frac{G1}{B\&B}$	$\frac{G2}{B\&B}$	$\frac{PCS}{B\&B}$	$\frac{PCS+}{B\&B}$
2	10	101,51%	100,80%	175,89%	112,10%
3	10	101,56%	101,20%	176,83%	113,02%
4	10	102,52%	102,45%	182,22%	113,38%
5	10	102,79%	102,83%	181,00%	112,70%
6	10	103,89%	103,06%	182,14%	112,19%
7	10	105,01%	103,64%	178,80%	113,17%
8	10	106,53%	106,73%	179,59%	112,37%
9	10	106,25%	106,93%	173,87%	112,19%
10	10	105,86%	107,49%	170,76%	113,51%
11	10	109,53%	108,61%	183,87%	115,97%
12	10	111,07%	109,71%	177,13%	114,88%
13	10	109,80%	110,71%	167,78%	114,87%
14	10	110,75%	110,70%	176,34%	115,86%
15	10	109,54%	112,00%	173,63%	116,01%
2	20	101,91%	100,57%	185,63%	108,07%
3	20	102,95%	101,63%	193,42%	113,35%
4	20	103,90%	100,97%	188,94%	116,88%
5	20	104,26%	102,92%	191,23%	115,36%
6	20	104,77%	103,13%	193,72%	115,95%
7	20	105,84%	104,99%	201,95%	112,82%
8	20	105,82%	106,60%	204,52%	111,72%
9	20	106,97%	106,64%	201,67%	115,44%
10	20	106,93%	108,07%	209,24%	112,28%
11	20	110,17%	110,46%	205,18%	115,61%
12	20	108,50%	110,83%	204,65%	116,13%
13	20	108,72%	110,70%	202,79%	115,54%
14	20	110,66%	110,95%	195,40%	116,30%
15	20	112,65%	113,22%	209,22%	117,37%

Tabela 5.4. Średnie wartości proporcjonalne (w procentach odległości od wartości optymalnej). Dane dla 2-15 produktów i 30 sklepów. 100% oznacza wartość optymalną

Produktów	Sklepów	$\frac{G1}{B\&B}$	$\frac{G2}{B\&B}$	$\frac{PCS}{B\&B}$	$\frac{PCS+}{B\&B}$
2	30	102,86%	100,45%	175,81%	109,08%
3	30	102,89%	100,85%	187,81%	114,29%
4	30	104,66%	102,55%	208,95%	114,29%
5	30	103,23%	103,33%	206,54%	112,85%
6	30	105,91%	102,39%	195,28%	111,95%
7	30	106,58%	104,54%	209,89%	115,43%
8	30	106,65%	105,76%	218,37%	115,68%
9	30	106,76%	108,23%	220,63%	116,54%
10	30	106,63%	108,23%	199,77%	115,28%
11	30	107,98%	109,41%	210,42%	116,80%
12	30	109,14%	110,35%	220,19%	114,72%
13	30	110,86%	111,40%	215,91%	118,36%
14	30	111,37%	112,37%	220,55%	115,70%
15	30	110,55%	113,03%	219,70%	118,75%

5.6 Algorytmy heurystyczne

Aby zaprezentować lepsze wyniki niż uzyskiwane przez znane dotychczas [74], proste heurystyki rozwiązujące problem ISOP i ISOPwD, opracowano nowy zestaw heurystycznych podejść do rozwiązania problemu. Zestaw rozwiązań składa się z nowej, wydajnej metaheurystyki opartej na procesie optymalizacji komórkowej (opracowanej w podobnym czasie zarówno dla podstawowego problemu ISOP, jak i dla problemu zawierającego definicje rabatów – ISOPwD), rozbudowanej wersji znanego już dla ISOP algorytmu heurystycznego, jak i dwóch klasycznych algorytmów, zachłannego i jego rozbudowanej wersji. Zaprojektowaliśmy nową heurystykę jako rozwiązanie, które równoważy czas i wyniki obliczeń, z założenia mające być bliskie rozwiązaniom optymalnym. Co niezmiernie istotne, obliczeniowe badania eksperymentalne przeprowadzone zostały na dużym zbiorze rzeczywistych danych. Zestaw danych został zebrany z 32 sklepów i objął największe sklepy amerykańskie, w tym: Amazon, BarnesandNoble.com, Borders.com, Buy.com i Booksamillion, oraz czołowych sprzedawców wśród księgarń internetowych w Polsce, takich jak empik.com i merlin.pl. Wyniki prezentowane przez algorytmy heurystyczne porównywane są między sobą, jak i odnoszone do wartości optymalnych. Optymalne rozwiązania dla małych problemów są obliczane za pomocą algorytmu B&B. Aby ocenić skalowalność, stabilność i zachowanie heurystyk stopniowo zwiększano rozmiary problemu [73].

ISOPwD jest silnie NP-trudny [10]. Ponadto, według naszej najlepszej wiedzy, nie jest tożsamy z żadnym ze znanych problemów. Dlatego słuszne wydaje się zaproponowanie zbioru różnych rozwiązań heurystycznych, poczynawszy od prostych efektywnych algorytmów zachłannych [22], które wykorzystują lokalną wiedzę i nie pozwalają na cofanie się w celu zwiększenia wydajności. Warto podkreślić, że algorytmy zachłanne nie zawsze obliczają optymalne rozwiązania. Mogą jednak zapewnić rozwiązanie o dopuszczalnej dla danych potrzeb jakości (a w niektórych przypadkach jest to rozwiązanie optymalne), zużywając przy tym znacznie mniej zasobów i czasu potrzebnego na dokonanie obliczeń. Ze względu na charakter problemu ISOP (opisywany szeroko w rozdziale 4) niski czas jednostkowych obliczeń jest niezmiernie istotny.

W dalsze części rozdziału przedstawiono i krótko scharakteryzowano użyte w eksperymentach algorytmy.

5.6.1 Algorytm zachłanny GREEDY

Jest to ten sam algorytm, który został przedstawiony w podsekcji 4.4.4. Została jedynie uwzględniona dodatkowa funkcja opisująca rabaty (odpowiadająca dla każdego sklepu). Dla krótkiego przypomnienia – rozważamy, że produkty są sortowane rosnąco $1, \dots, mln$. Wartości całkowitej dostawy dla każdego sklepu są wstępnie ustalone na d_j , $j = 1, \dots, n$. y_j jest wskaźnikiem wysyłki. Standardowa cena każdego produktu i w sklepie j jest ustawiona jako p_{ij} . W każdej iteracji

i produkt i jest wybierany w kwalifikującym się sklepie j o minimalnej wartości $T + f_j(p_{ij}) + d_j$, a odpowiednia wartość T jest ponownie przydzielana do $T = T + f_j(p_{ij}) + d_j$. Następnie d_j jest ustawiane na 0. Funkcja rabująca $f_j(p_{ij})$ zwraca wartość produktu i po zastosowaniu zniżki dla sklepu j .

5.6.2 Algorytm z prognozowaniem FORECASTING

Ze względu na opisywane już możliwe niekorzystne zachowanie algorytmu GREEDY (patrz podsekcję 4.4.4 i tabelę 4.4) wykorzystano również zmodyfikowaną wersję algorytmu, którą możemy nazwać zachłannym z krokiem wprzód. Jest to zmodyfikowana wersja algorytmu FORECASTING prezentowanego wcześniej (podsekcja 4.4.5). Główną ideą jest sprawdzenie sytuacji o krok do przodu (wykrywanie złych sytuacji). Od pierwszego kroku do przedostatniego, algorytm oblicza własną “ocenę”, aby wybrać kwalifikujący się sklep j do zakupu produktu i . Zamiast szukać lokalnego optymalnego rozwiązania, analizuje również krok do przodu (co zapobiega wystąpieniu opisywanego niekorzystnego przypadku) i oblicza współczynnik wyboru jako $T_j + \frac{f_j(p_{ij} + p_{ij+1}) + d_j}{2}$ dla każdego sklepu j . Wybierając ten o najniższej obliczonej wartości. W każdym kolejnym kroku brany jest pod uwagę następny produkt i , $i = i + 1$. T jest ustawiony jako $T = T + f_j(p_{ij}) + d_j$. Następnie d_j jest ustawiane na 0.

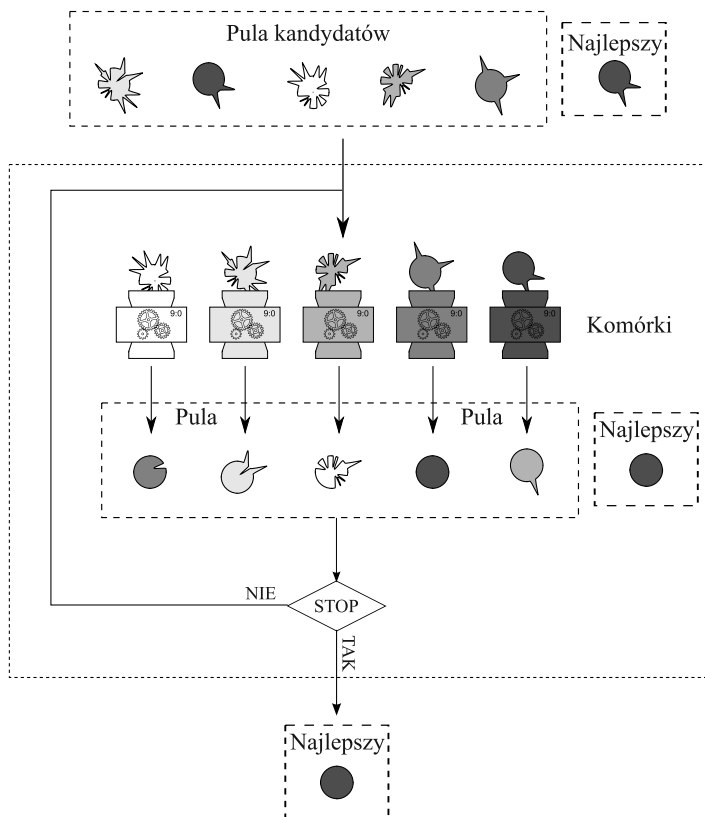
Ostatni krok algorytmu działa w inny sposób (prognoza nie może działać poza zestawem produktów i). Ostatni produkt wybierany jest zatem w kwalifikującym się sklepie j przy minimalnej wartości $T + f_j(p_{ij})$.

Funkcja rabatująca $f_j(p_{ij} + p_{ij+1})$ zwraca łączne koszty produktów i oraz $i + 1$ po zastosowaniu rabatu dla sklepu j .

5.6.3 Algorytm przetwarzania komórkowego CELLULAR

Algorytm oparty na przetwarzaniu komórkowym jest nowym pseudo-równoległym podejściem do optymalizacji [100]. Obejmuje on wiele komórek przetwarzających, które badają różne regiony przestrzeni wyszukiwania. Każda komórka przetwarzająca może być zaimplementowana przy użyciu heurystyki populacyjnej lub opartej na wyszukiwaniu albo ich hybrydyzacji. Główną ideą i zasadą algorytmu jest podzielenie algorytmu sekwencyjnego na kilka pseudo-równoległych modułów przetwarzania (tj. komórki) tak, aby każda komórka mogła badać różne rejony przestrzeni wyszukiwania. Główną cechą nowego podejścia jest to, że iteracyjna weryfikacja warunków zastoju zapobiega marnowaniu czasu na niepotrzebne zadania.

Algorytm, który został wykorzystany do eksperymentów obliczeniowych dla problemu ISOPwD został zaprezentowany w podsekcji 4.4.3.



Rysunek 5.1. Algorytm Cellular-ILS – schemat blokowy

5.6.4 Algorytm MIN-MIN

Algorytm Min-Min jest powszechnie i szeroko stosowany w kontekście planowania niezależnych zadań w rozproszonych obliczeniach w celu zminimalizowania całkowitego czasu realizacji zadań [26, 76]. Jedną z głównych zalet tego podejścia jest to, że algorytm na ogół jest w stanie uzyskać dobrej jakości rozwiązania przy stosunkowo niewielkich kosztach obliczeniowych. Z drugiej strony, ponieważ planuje te zadania lub procesy z preferencją minimalnego kosztu, może to prowadzić do niezrównoważonego rozwiązania [109]. Braun i inni [15] opisywali proces heurystycznego podejścia do optymalizacji Min-Min. Algorytm Min-Min przygotowany dla problemu ISOPwD został rozszerzony i dopasowany do wyboru listy produktów z oferty sklepów internetowych. Nasz algorytm oznaczymy jako MIN-MIN.

Proces rozpoczyna się od wyszukania produktu na liście produktów nieprzypisanych N , co minimalizuje całkowity koszt TC w koszyku różnych sklepów M , biorąc pod uwagę koszt produktu C_{ij} plus koszt dostawy D_i . W przypadku remisu wybierany jest produkt o niższym koszcie dostawy, a w przypadku, gdy oba sklepy mają ten sam koszt dostawy, wybierany jest on losowo.

Po przypisaniu, całkowity koszt koszyka jest aktualizowany o wybrany produkt i jest usuwany z nieprzypisanej listy produktów N . Proces trwa do momentu opróżnienia listy produktów, które nie zostały przypisane.

Różne warianty i możliwości algorytmu MIN-MIN zostały już szerzej opisane w podsekcji 4.4.2.

5.7 Eksperymenty obliczeniowe

Eksperymenty obliczeniowe zostały przeprowadzone i podzielone na dwie grupy (ze względu na wielkość instancji i czas obliczeń) – zestaw eksperymentów zawierający dokładny algorytm B&B, oraz zestaw eksperymentów bez optymalnego rozwiązania – jako porównanie wszystkich heurystyk w celu oceny ich skalowalności i jakości obliczeń dla dużych wartości danych.

Wszystkie algorytmy zostały zaimplementowane w języku programowania PHP w wersji 5.2. Celem wykorzystania PHP było łatwe osadzenie algorytmów na stronie internetowej tak, aby działały w warunkach podobnych do ich przyszłej wersji ostatecznej.

Eksperymenty przeprowadzono na komputerze Apple MacBook Pro w wersji 8.1 z procesorem Intel pracującym na częstotliwości 2,3 GHz, dwoma rdzeniami fizycznymi i dwoma wirtualnymi rdzeniami dla każdej pamięci głównej 4GB DDR3 pracującej z częstotliwością 1333 MHz. Eksperymenty przeprowadzono na wirtualnej maszynie z GNU/Linuxem Mint 11 na hoście Mac OS X 10.7, na której zostały przypisane dwa wirtualne rdzenie i pamięć główna o pojemności 2 GB.

5.7.1 Modelowanie danych i generator instancji

Wyzwaniem w badaniach eksperymentalnych było stworzenie modelu, który byłby jak najbliższy rzeczywistym warunkom zakupów w Internecie. Eksperyment z danymi z rzeczywistego świata byłby kolejnym etapem badań, które wymagają znacznie szerszego dialogu ze środowiskiem biznesowym. Jako badane produkty w naszym eksperymencie obliczeniowym wybraliśmy książki, ze względu na ich szeroki wybór w sklepach internetowych oraz częstotliwość zakupów przez tego typu kanał zakupowy. Wykorzystaliśmy niektóre informacje z [19, 18]. Ponadto dla usprawnienia modelu, który wcześniej wykorzystano przeanalizowaliśmy wiele sklepów internetowych (m.in. Amazon, BarnesandNoble, Borders, Buy, Booksamillion oraz czołowych sprzedawców wśród księgarni internetowych w Polsce, takich jak empik.com i merlin.pl), aby stworzyć własny model z generatorem instancji jak najbardziej zbliżony do rzeczywistości.

Model roboczy został przygotowany w następujący sposób. W eksperymentach obliczeniowych zakładamy, że liczba sklepów $n \in \{20, 40\}$, a liczba kupowanych produktów $m \in \{2, 3, \dots, 10, 15, \dots, 100\}$. Dla każdej pary n, m wygenerowano 100 instancji problemu. W każdym przypadku następujące wartości zostały wygenerowane losowo dla wszystkich i i j w odpowiednich przedziałach. Cena dostawy: $d_j \in \{5, 10, 15, 20, 25, 30\}$, rekomendowana cena książki i : $r_i \in \{5, 10, 15, 20, 25\}$, a cena książki i w księgarni j ustalona jako $p_{ij} \in [a_{ij}, b_{ij}]$, gdzie $a_{ij} \geq 0,69r_j$, $b_{ij} \leq 1,47r_j$, a struktura przedziałów $[a_{ij}, b_{ij}]$ jest przygotowana w następujący sposób:

$$\begin{aligned}
 [32\%] \quad & \text{minimum} \\
 & \text{minimum} + \frac{(\text{median} - \text{minimum})}{4} \\
 [9\%] \quad & \text{minimum} + \frac{(\text{median} - \text{minimum})}{2} \\
 [9\%] \quad & \text{minimum} + \frac{(\text{median} - \text{minimum})}{1.33} \\
 [8\%] \quad & \text{median} \\
 [13\%] \quad & \text{median} + \frac{(\text{maximum} - \text{median})}{4} \\
 [6\%] \quad & \text{median} + \frac{(\text{maximum} - \text{median})}{2} \\
 [11\%] \quad & \text{median} + \frac{(\text{maximum} - \text{median})}{1.33} \\
 [12\%] \quad & \text{maximum}
 \end{aligned}$$

gdzie $\text{minimum} = 0,69r$ i $\text{maximum} = 1,47r$.

Zaproponowano również następujący kształt funkcji opisującej rabaty dla danego sklepu.

$$f(x) = \begin{cases} x & \text{jeżeli } x \leq 25; \\ 0,95x & \text{jeżeli } 25 < x \leq 50; \\ 0,90x & \text{jeżeli } 50 < x \leq 100; \\ 0,85x & \text{jeżeli } 100 < x \leq 200; \\ 0,80x & \text{jeżeli } 200 < x. \end{cases}$$

Obowiązuje zatem prosta, realna zasada – im więcej pieniędzy wydasz, tym większe zniżki możesz uzyskać.

5.7.2 Zbiór eksperymentów obliczeniowych obejmujący rozwiązanie optymalne

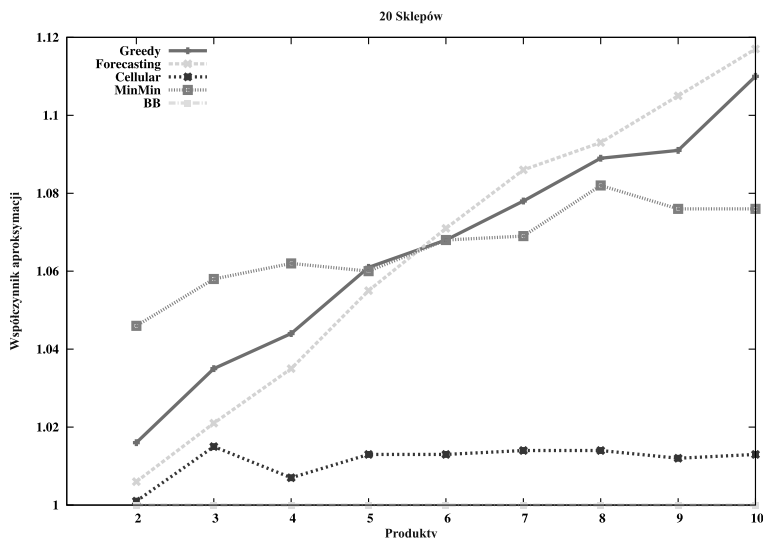
Pierwsza grupa eksperymentów to ta, w której optymalne rozwiązania otrzymane za pomocą dokładnego algorytmu B&B zostały porównane ze wszystkimi prezentowanymi rozwiązaniami heurystycznymi, tj. GREEDY, FORECASTING, CELLULAR i Min-Min. W rzeczonym eksperymencie przyjęto, że liczba sklepów $n \in \{20\}$, a liczba kupowanych produktów $m \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Dla każdej pary (n, m) wygenerowano 100 instancji według modelu opisanego w podsekcji 5.7.1. Liczba przeanalizowanych instancji wynosi zatem 900.

Algorytmy porównano przy użyciu trzech miar: współczynnika aproksymacji, czasu działania i wskaźnika dyspersji opartego na odchyleniu standardowym. Współczynnik aproksymacji jest zdefiniowany jako $\rho = \frac{F(X)}{F(X)^*}$ gdzie $F(X)$ reprezentuje rozwiązanie znalezione przez dany algorytm, a $F(X)^*$ oznacza rozwiązanie optymalne.

Na rysunku 5.2 przedstawiono uśrednione (dla 100 instancji) wyniki dla zestawu 2-10 produktów kupowanych spośród oferty 20 sklepów.

Spośród wszystkich heurystyk algorytm komórkowy CELLULAR zapewnia najlepszą jakość rozwiązań (najbardziej zbliżoną do optymalnej), bez względu na wielkość instancji. W najgorszym przypadku rozwiązanie zaproponowane przez ten algorytm było zaledwie 1,47% droższe od najtańszego możliwego. Dla wielu pojedynczych przypadków algorytm komórkowy oblicza optymalne rozwiązanie (należy przypomnieć, że dane na wykresie reprezentują wartości uśrednione dla 100 instancji problemu przy tej samej liczbie sklepów i produktów). Algorytm komórkowy obliczył optymalne rozwiązanie dla 62% instancji. Należy zauważyć, że algorytm jest bardzo stabilny pod względem jakości rozwiązań (w większości przypadków jest to rozwiązanie między 1,24% – 1,47% gorsze od optymalnego).

Zarówno algorytmy GREEDY, jak i FORECASTING zapewniają podobną jakość rozwiązań. Dla mniejszej liczby produktów m , ten drugi był lepszy, a dla większej liczby produktów $m > 5$, lepszy okazuje się algorytm GREEDY. Łatwo zauważyć, że jakość rozwiązania pogarsza się wraz z rosnącą liczbą produktów m . Prosty algorytm zachłanny GREEDY był w stanie znaleźć optymalne rozwiązanie dla



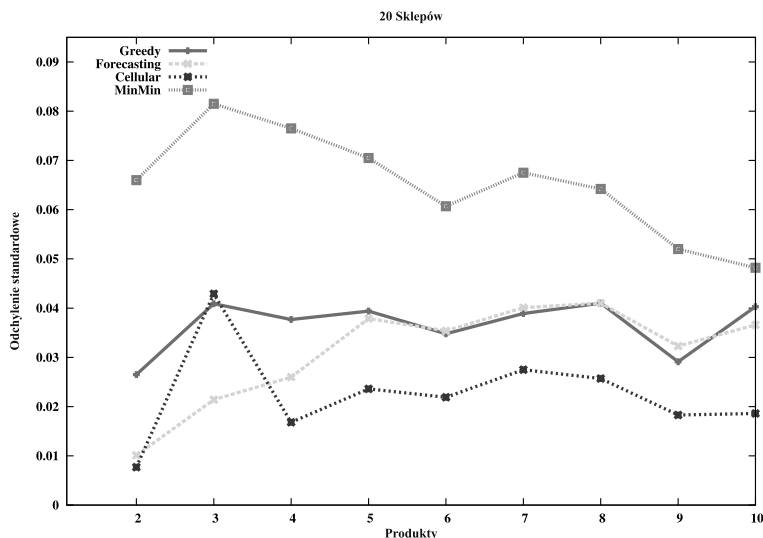
Rysunek 5.2. Uśrednione (dla 100 instancji) wyniki dla 2-10 produktów kupowanych z 20 sklepów

6% instancji, podczas gdy algorytm FORECASTING obliczył wartość optymalną dla 7% instancji.

Ostatni wykorzystany algorytm heurystyczny MIN-MIN dostarcza najgorszych rozwiązań dla mniejszej liczby produktów. Algorytm jest jednak dość stabilny pod względem jakościowym, dlatego dla większej liczby produktów zapewnia lepsze rozwiązania niż GREEDY i FORECASTING. Algorytm znalazł optymalne rozwiązanie dla 20% badanych instancji.

Jeśli zależy nam wyłącznie na jakości rozwiązań, niekwestionowanym liderem wśród algorytmów heurystycznych jest algorytm komórkowy CELLULAR. Korzystając z tego algorytmu klient jest w stanie zaoszczędzić odpowiednio 5,88%, 6,30% i 6,64% w stosunku do rozwiązań proponowanych przez algorytmy MIN-MIN, GREEDY i FORECASTING.

Rysunek 5.3 przedstawia wyniki rozważań nad dyspersją (łącznie z rozwiązaniem optymalnym) zobrazowane za pomocą odchylenia standardowego. Każdy punkt reprezentuje wartość odchylenia standardowego (odchylenie od wartości optymalnej) ze 100 testów obliczeniowych dla 20 sklepów, gdzie każda wartość dla każdego testu jest przedstawiana jako korelacja wyniku z wartością optymalną ($\frac{Greedy}{BB}$, $\frac{Forecasting}{BB}$, $\frac{Cellular}{BB}$, $\frac{Min}{BB}$). Wyższe odchylenie standardowe oznacza bardziej niestabilną pracę (duża różnica odległości od rozwiązania optymalnego). Jako wartość referencyjną wykorzystano wyniki algorytmu B&B, który zawsze oblicza rozwiązanie optymalne. Wykres dostarcza bardzo przydatnych



Rysunek 5.3. Wyniki odchylenia standardowego dla eksperymentu z 20 sklepami i 2-10 produktami

informacji na temat stabilności wyników algorytmów.

Dla uzupełnienia informacji zawartej na skomentowanym rysunku 5.3 warto odwołać się również do tabeli 5.5, która zawiera dane reprezentowane przez współczynnik zmienności (CV) znormalizowanej miary (wartość procentowa). Każda komórka reprezentuje współczynnik zmienności (CV) ze 100 testów obliczeniowych dla 20 sklepów, gdzie każda wartość dla każdego testu jest przedstawiana jako korelacja wyniku z wartością optymalną. Wyższa wartość CV oznacza bardziej niestabilną pracę (duża różnica odległości od optymalnego rozwiązania) dla ponad 100 testów danego przypadku (czasami optymalnych lub bliskich, czasami dość odległych). Jako wartość referencyjną wykorzystaliśmy wyniki algorytmu B&B. W każdej komórce prezentowana jest wartość CV dla tych samych ilości n sklepów i m produktów – przy 100 testach obliczeniowych.

Ważnym czynnikiem, który należy wziąć pod uwagę w kontekście zakupów internetowych, jest czas potrzebny każdemu algorytmowi na obliczenie rozwiązania. Rysunek 5.4 zawiera zobrazowane wyniki dotyczące porównania czasu działania algorytmów, wyrażone w milisekundach [ms]. Każda komórka reprezentuje średnią wartość ze 100 testów obliczeniowych dla 20 sklepów.

Dla niewielkiej liczby produktów ($m \leq 5$) MIN-MIN jest najszybszym algorytmem. Można zauważyć, że gdy liczba produktów jest większa niż pięć ($m > 5$) algorytm GREEDY jest szybszy niż pozostałe. Różnice pomiędzy wszystkimi algorytmami są bardzo znaczące.

Tabela 5.5. Porównanie dyspersji algorytmów – przedstawienie współczynnika zmienności (CV). Dane dla 20 sklepów i 2-10 produktów

produktów	GREEDY	FORECASTING	Cellular	MIN-MIN	B&B
2	2,62%	1,02%	0,78%	6,33%	0,00%
3	3,96%	2,11%	4,23%	7,67%	0,00%
4	3,63%	2,53%	1,68%	7,21%	0,00%
5	3,72%	3,61%	2,33%	6,55%	0,00%
6	3,27%	3,31%	2,17%	5,69%	0,00%
7	3,62%	3,70%	2,71%	6,32%	0,00%
8	3,78%	3,81%	2,54%	5,93%	0,00%
9	2,67%	2,93%	1,82%	4,84%	0,00%
10	3,64%	3,28%	1,84%	4,48%	0,00%

Czas działania algorytmu dokładnego B&B rośnie wykładniczo i nie było możliwe przygotowanie eksperymentów dla większej liczby produktów. Z drugiej strony, wszystkie heurystyki są bardzo szybkie, więc postanowiono wykonać dalsze testy, znacznie zwiększając liczbę produktów m .

5.7.3 Eksperymenty obliczeniowe dla dużego zestawu danych

W drugiej grupie eksperymentów dokonano porównania pomiędzy algorytmami heurystycznymi, rozwiązującymi zadanie złożone z dużo większego rozmiaru problemu. W tych przykładach zwiększono rozmiar instancji w następujący sposób: $n \in \{20, 40\}$, $m \in \{5, 10, 15, \dots, 100\}$, natomiast funkcja rabatująca pozostała zgodna z opisem w podsekcji 5.7.1.

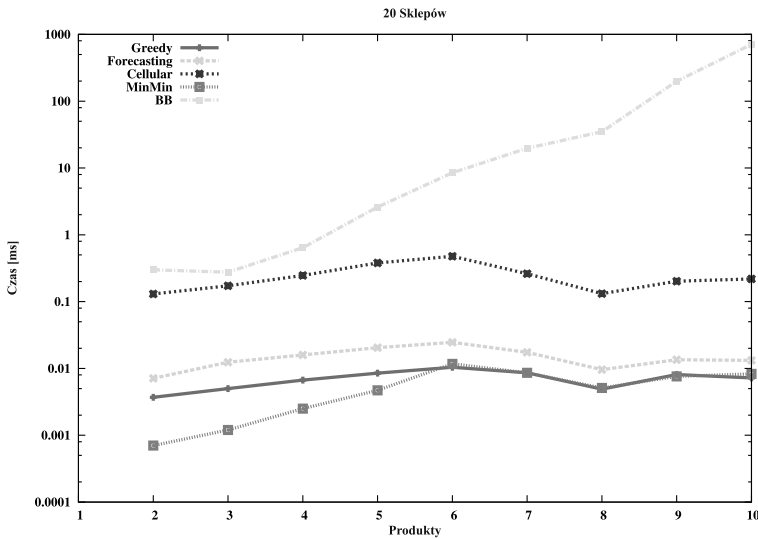
Dla każdej pary wygenerowano 100 instancji. Instancje zostały wygenerowane przy użyciu tej samej procedury, która została opisana w sekcji 5.7.2.

Ponieważ nie jesteśmy w stanie obliczyć optymalnej wartości rozwiązania w rozsądnym czasie, biorąc pod uwagę wielkość instancji, jako główną miarę wyznaczamy błąd względny γ . Miara jest formalnie zdefiniowana jako $\gamma = \frac{F(X)}{F(X)_{best}}$, gdzie $F(X)$ reprezentuje rozwiązanie znalezione przez heurystykę, a $F(X)_{best}$ oznacza najlepsze rozwiązanie spośród wszystkich heurystyk.

Zestaw eksperymentów zawierających dane z 20 sklepów

Rysunek 5.5 przedstawia średnie wartości błędu względnego dla 100 instancji problemu.

Można zauważyć, że algorytm komórkowy przewyższa jakością rozwiązania wszystkie pozostałe algorytmy. We wszystkich przypadkach n, m gdzie $m = \{5, 10, 15, \dots, 100\}$ (n jest stałe = 20), jest to najlepsze rozwiązanie.



Rysunek 5.4. Porównanie czasu działania algorytmów. Eksperyment przeprowadzony dla 20 sklepów

Algorytm GREEDY w porównaniu do MIN-MIN i FORECASTING oferuje lepszą skalowalność, zapewniając drugie co do jakości rozwiązanie. Dla ilości produktów ponad $m > 15$ miara błędu względnego ustabilizowała się w granicach 10-11% ponad najlepsze rozwiązanie CELLULAR.

Dla niewielkiej liczby produktów $m = 5$ FORECASTING proponuje prawie taką samą jakość rozwiązania jak najlepszy algorytm CELLULAR (0,48% droższy). Od $m > 15$ produktów staje się jednak najgorszym algorytmem spośród testowanych. Ponadto, łatwo zauważyć, że jakość rozwiązania pogarsza się wraz z rosnącą liczbą produktów m .

Ostatni algorytm heurystyczny MIN-MIN dostarcza najgorszych rozwiązań dla mniejszej liczby produktów ($m < 15$). Dla większej liczby produktów zapewnia jednak lepsze rozwiązania niż algorytm FORECASTING (ale gorsze niż GREEDY). MIN-MIN dostarcza rozwiązania od 6% do 16% (średnio) gorsze niż najlepsze rozwiązanie.

Rysunek 5.6 zawiera porównanie rozproszenia uzyskiwanych wyników dla algorytmów heurystycznych. Każdy punkt reprezentuje wartość odchylenia standardowego ze 100 testów obliczeniowych dla 20 sklepów. Każda wartość dla każdego testu jest przedstawiana jako korelacja pomiędzy aktualnym wynikiem algorytmu, a najlepszym wynikiem dla tego obliczenia ($\frac{Greedy_{best}}{Greedy_{best}}$, $\frac{Forecasting_{best}}{Forecasting_{best}}$, $\frac{Cellular_{best}}{Cellular_{best}}$, $\frac{Min-Min_{best}}{Min-Min_{best}}$). W każdym punkcie prezentowana jest wartość odchylenia standardowego dla tego samego przykładu n sklepów i m produktów.

Tabela 5.6. Porównanie dyspersji algorytmów – przedstawienie współczynnika zmienności (CV). Dane z 20 sklepów dla 5-100 produktów

produktów	GREEDY	FORECASTING	Cellular	Min-Min
5	4,45%	2,92%	2,62%	6,58%
10	3,75%	3,73%	0,94%	5,24%
15	2,79%	3,17%	0,00%	3,86%
20	3,18%	3,19%	0,00%	3,93%
25	2,65%	2,65%	0,00%	3,74%
30	2,71%	3,09%	0,00%	3,26%
35	2,36%	2,47%	0,00%	3,50%
40	3,42%	2,36%	0,00%	3,82%
45	3,04%	2,38%	0,00%	3,62%
50	3,09%	2,40%	0,00%	3,61%
55	2,69%	2,06%	0,00%	3,29%
60	2,51%	2,27%	0,00%	3,40%
65	2,10%	2,17%	0,00%	3,72%
70	3,77%	2,12%	0,00%	3,62%
75	2,57%	2,35%	0,00%	3,37%
80	2,73%	1,98%	0,00%	3,23%
85	3,48%	2,36%	0,00%	3,56%
90	2,06%	1,99%	0,00%	3,05%
95	3,47%	1,98%	0,00%	3,49%
100	2,71%	3,09%	0,00%	3,26%

Tabela 5.6 zawiera informacje w postaci znormalizowanej miary współczynnika zmienności.

Rysunek 5.7 przedstawia informacje dotyczące porównania czasu działania algorytmów [ms]. Każda komórka reprezentuje średnią wartość ze 100 testów obliczeniowych dla danych pochodzących z 20 sklepów. Można zauważyć, że dla małej liczby produktów $m = 5$ najszybszym algorytmem jest MIN-MIN. W innych przypadkach (liczba produktów $m > 5$) algorytm GREEDY jest najszybszy. Różnice pomiędzy wszystkimi algorytmami są bardzo znaczące.

Zestaw eksperymentów zawierających dane z 40 sklepów

Przy dwukrotnym zwiększeniu liczby sklepów (do wartości 40) algorytm komórkowy ponownie zapewniał najlepszą jakość rozwiązań. Oznacza to, że w każdym przypadku n, m było to najlepsze rozwiązanie.

Rysunek 5.8 przedstawia uśrednione wyniki dla błędu względnego. Zauważmy, że charakterystyka wyników nie jest daleka od wyników eksperymentu wykorzystującego dane z $m = 20$ sklepami.

Algorytm GREEDY zapewnia drugą najwyższą jakość rozwiązań. Dla $m > 25$ produktów ustabilizował się w granicach 9,6-10,8% od najlepszego rozwiązania.

Dla niewielkiej liczby produktów $m = 5$ FORECASTING prezentuje prawie taką samą jakość rozwiązań jak najlepszy algorytm CELLULAR (jest 1,55% droższy). Od $m > 10$ produktów staje się najgorszym algorytmem spośród testowanych. Ponadto łatwo zauważyć, że jakość rozwiązania pogarsza się wraz z rosnącą liczbą produktów m od 1,55% do 15,83%.

MIN-MIN oferuje najgorsze rozwiązania dla mniejszej liczby produktów ($m = 5$). Dla większej liczby produktów zapewnia jednak lepsze rozwiązania niż algorytm FORECASTING (ale gorsze niż GREEDY). Algorytm zapewnia również lepszą stabilność niż w poprzednim eksperymencie dla mniejszej liczby sklepów internetowych.

Rysunek 5.9 zawiera porównanie rozproszenia wyników (prezentując wartości odchylenia standardowego).

W przypadku doświadczenia z użyciem danych pochodzących z 40 sklepów internetowych obserwacje ogólne są podobne do tych z eksperymentu przeprowadzonego dla danych z 20 sklepów. Jednak pewne różnice są znaczące i warte zauważenia. Wszystkie odchylenia standardowe i wartości CV są niższe w przypadku eksperymentu z użyciem danych z 40 sklepów. Algorytm GREEDY zapewnia niższy poziom rozproszenia niż FORECASTING (nieznacznie, ale wciąż). Obserwacje dla algorytmów komórkowego CELLULAR i MIN-MIN są podobne do tych z eksperymentu dla danych z ofertami 20 sklepów internetowymi.

Jeśli przeanalizujemy czas obliczeń (patrz rysunek 5.10), dla wszystkich instancji ($n = 40$ sklepów), algorytm GREEDY jest najszybszy. Różnice pomiędzy wszystkimi algorytmami są bardzo znaczące. Czas obliczeń jest tym ważniejszy, ponieważ aplikacja docelowa będzie działać jako strona internetowa. Istnieje wiele badań pokazujących, że użytkownicy nie chcą czekać na pojawienie się zawartości strony internetowej dłużej niż kilka sekund, ponieważ szybko tracą zainteresowanie [63].

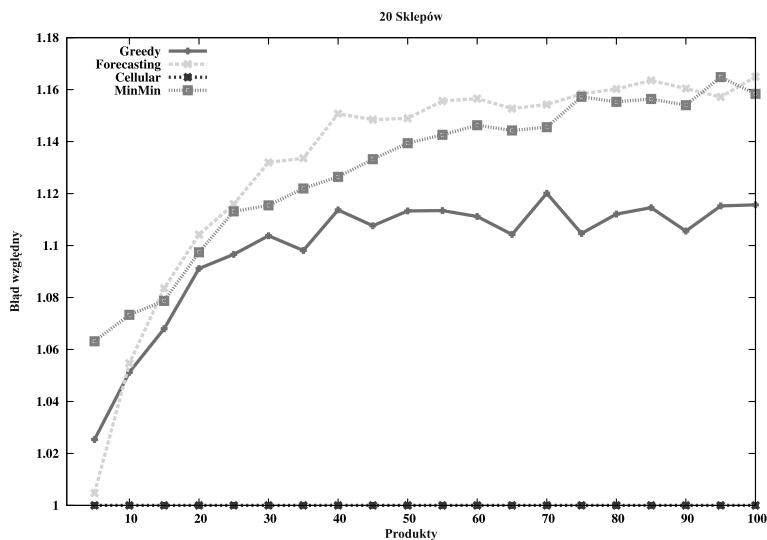
W odniesieniu do wszystkich wyników zebranych w eksperymencie obliczeniowym można stwierdzić, że algorytm CELLULAR zapewnia najlepsze rozwiązania spośród wszystkich algorytmów heurystycznych. Dostarcza on również bardziej stabilne rozwiązania niż pozostałe porównywane algorytmy. Z drugiej strony, jest od pozostałych wolniejszy. Warto zauważyć, że może on być znacznie bardziej wydajny, gdy będzie zaimplementowany współbieżnie na pięciu maszynach (każda komórka może przetwarzać na innej maszynie). Innym ważnym aspektem jest to, że w prawdziwej sytuacji rynku e-commerce algorytmy będą wykonywane na wydajnych urządzeniach serwerowych (np. w serwerze centrum danych), które zwykle są znacznie potężniejsze (pod kątem mocy obliczeniowej) od pojedynczych komputerów klientów.

Algorytm GREEDY może wzbudzić duże zainteresowanie ze względu na bardzo krótki czas obliczeń, w którym może zapewnić całkiem dobrą jakość rozwiązań. Oba algorytmy GREEDY i FORECASTING mogą być implementowane

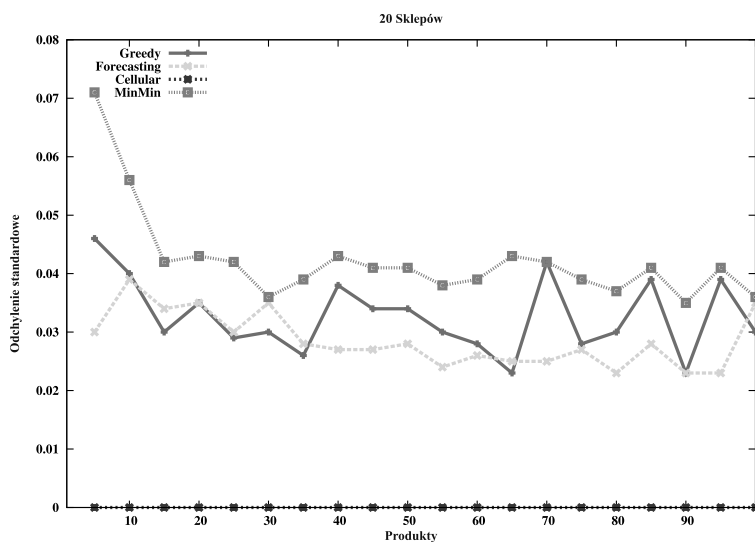
współbieżnie na dwóch maszynach / procesorach / niezależnych wątkach. Oba algorytmy tworzą dwa oddzielne wykonania i wybierają najlepsze z nich na ostatnim etapie.

Wydażność MIN-MIN można poprawić za pomocą lokalnego algorytmu wyszukiwania [82]. Aby usprawnić wykonanie czasu pracy, MIN-MIN można łatwo zrównoleglić. Istnieją różne równoległe implementacje algorytmu [82, 29]. Są to implementacje zarówno procesora CPU, jak i procesora układu graficznego.

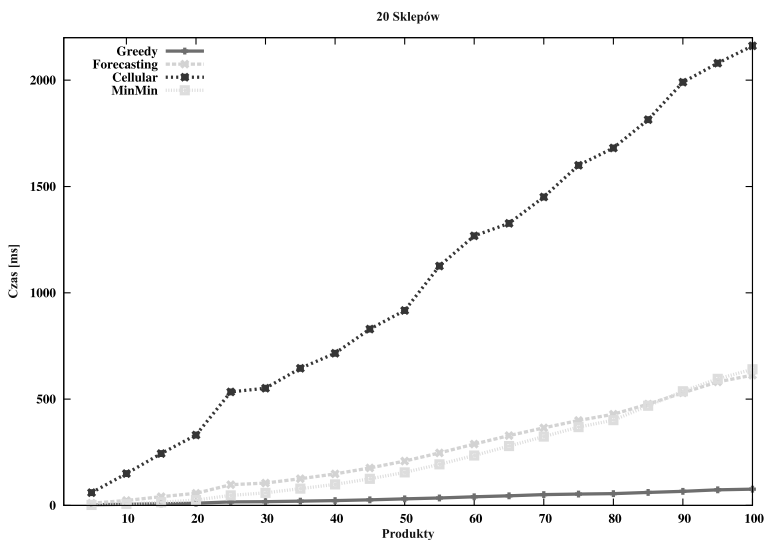
Biorąc pod uwagę szybki czas realizacji i dobrą jakość rozwiązań, które GREEDY, FORECASTING i MIN-MIN mogą zaoferować, sensowne wydaje się ich wykorzystanie jako początkowego ziarna dla algorytmu CELLULAR. Ponadto, warto zwrócić uwagę, że może być wykonana specjalna optymalizacja systemu baz danych w celu skrócenia czasu obliczeń [62].



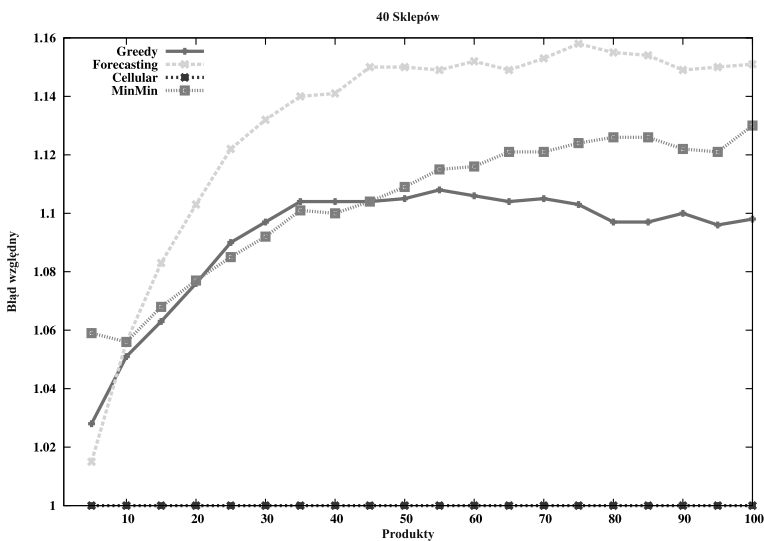
Rysunek 5.5. Wyniki działania algorytmów (eksperyment zawierający dane z 20 sklepów)



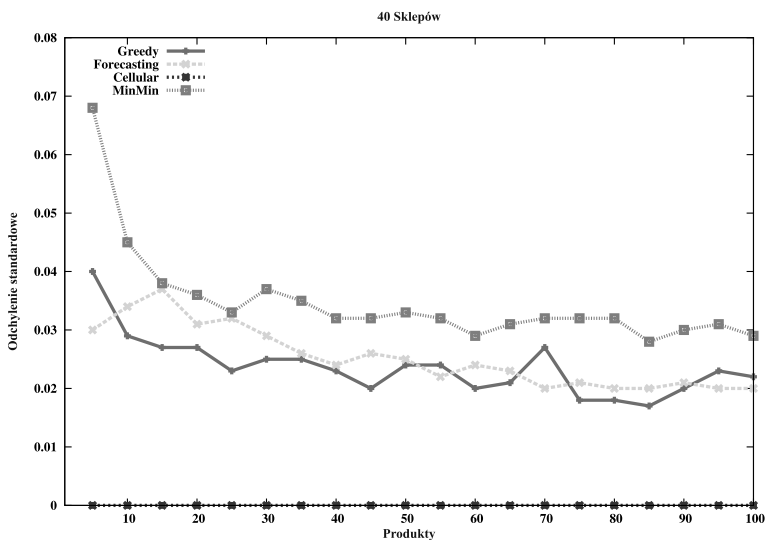
Rysunek 5.6. Wartości odchylenia standardowego (dane z 20 sklepów)



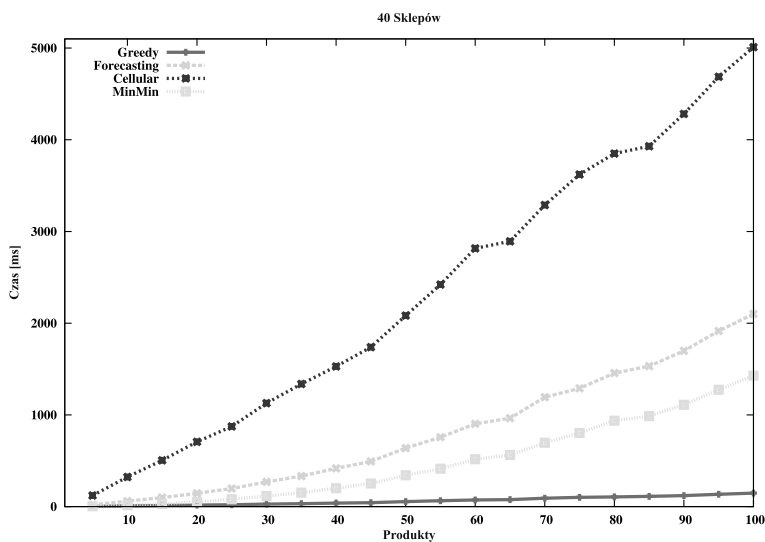
Rysunek 5.7. Czas działania algorytmów (dane z 20 sklepów)



Rysunek 5.8. Wyniki działania algorytmów (błąd względny). Eksperyment zawierający dane z 40 sklepów



Rysunek 5.9. Wartości odchylenia standardowego (dane z 40 sklepów)



Rysunek 5.10. Czasy obliczeń algorytmów (eksperyment obliczeniowy z wykorzystaniem danych z 40 sklepów)

5.8 Podsumowanie

W niniejszym rozdziale przedstawiono bardziej rozbudowaną wersję problemu optymalizacji zakupów internetowych. Oryginalny problem obrazował sytuację, w której klient chce zrealizować listę zakupów korzystając z oferty sklepów internetowych. Celem jest dokonanie zakupów przy najniższym możliwym rachunku końcowym biorąc pod uwagę różne oferty, jak i zróżnicowane koszty dostaw. Zdecydowano się przedstawić bardziej rozbudowaną, bardziej realistyczną i jeszcze ciekawszą wersję problemu ISOP. Dodano mianowicie funkcję określającą rabaty cenowe, jakie mogą zostać zaproponowane przez sprzedawców. Jest to często stosowana praktyka w sklepach internetowych, mająca zachęcić klientów do dokonania większych zakupów w ramach ich sklepu internetowego. Ma to tym większe znaczenie, że kupując większą ilość produktów z jednego sklepu oszczędzamy na kosztach wysyłki – jak to było podkreślone już przy definicji oryginalnego, podstawowego problemu ISOP. Nowa wersja problemu obejmująca rabaty została oznaczona jako ISOPwD (ang. Internet Shopping Optimization Problem with Price Sensitive Discounts).

W ramach prac nad problemem dowiedziono jego trudności obliczeniowej (przedstawiono dowód przynależności do klasy problemów silnie NP-trudnych). Następnie opracowano szereg algorytmów rozwiązujących problem. Działanie i jakość algorytmów zostały zbadane na podstawie wyczerpujących i złożonych eksperymentów obliczeniowych. Problem został rozwiązany ponad 22,5 tys. razy. Co istotne, do eksperymentów zostały użyte realistyczne modele danych (przygotowano generator oparty o złożoną analizę cen rynków). Eksperymenty zostały dogłębnie przeanalizowane i szeroko skomentowane. Doświadczenia były na tyle rozbudowane, że dla porządku podzielono ich opis na kilka sekcji w zależności od użytych algorytmów, jak i wielkości problemu.

Warto zauważyć, że problem ISOP (a także wynikający z niego ISOPwD) może być jeszcze bardziej skomplikowany po dodaniu kolejnych funkcji czy atrybutów. Będą one czynić problem jeszcze bliższym aktualnym wymaganiom użytkownika i pozwolą jeszcze lepiej odwzorować sytuację panującą na rynku zakupów internetowych. Każdy kolejny krok przy badaniach nad problemem ISOP pozwala czynić go jeszcze bardziej realnym i jeszcze bliższym oczekiwaniom klientów.

Rozdział 6

Problem optymalizacji zakupów z wieloma rabatami

Kolejnym ważnym rozszerzeniem problemu optymalizacji zakupów internetowych jest problem, w którym mamy do czynienia z dwoma niezależnymi funkcjami zmieniającymi koszty zakupów. W niniejszym rozdziale chcemy przedstawić problem, który nazwano Dual Discounting Functions for Internet Shopping Optimization Problem [11]. Oznaczmy go jako ISOPwDD. Poza powszechną i ciekawą praktyką stosowania przez sklepy rabatów (sytuacja, która została przedstawiona w poprzednim rozdziale 5) często wykorzystuje się również zróżnicowane koszty wysyłki. Różnice wynikają najczęściej z kwoty, którą mamy zamiar zapłacić. Im więcej wydanych pieniędzy, tym tańsza (lub darmowa) wysyłka. Praktyka ma na celu przyciągnąć klientów i zachęcić ich do kupowania większej ilości produktów w ramach jednego sklepu.

W kolejnych sekcjach niniejszego rozdziału zaprezentowany zostanie model matematyczny problemu. Opracowane i zaimplementowane algorytmy opisane zostaną w dalszej części. Obszerne testy (eksperymenty obliczeniowe) pozwolą poddać ocenie proponowane rozwiązania. Analiza i dyskusja dopełnią rozdział.

6.1 Definicja formalna – model matematyczny

Typowym przykładem jest następująca reklama w sklepie internetowym sprzedającym książki i płyty CD: Jeśli wartość Twojego zakupu wynosi co najmniej 125 zł, wówczas wyślemy Ci produkty za połowę standardowych kosztów wysyłki pocztą; 200 zł – połowę ceny dostawy przez kuriera; 250 zł – bezpłatna przesyłka pocztą; 350 zł – bezpłatna przesyłka kurierska. Jest to tylko przykład dla zobrazowania sytuacji. Wartości i progi (a także ich znacznie większa mnogość) mogą się różnić w zależności od sprzedawcy. Aby przyciągnąć klientów, w

sklepach internetowych często stosuje się również rabaty uzależnione od łącznej ceny produktów, które chcemy zakupić. Przykładem jest następująca praktyka zaobserwowana w sklepie internetowym sprzedającym części komputerowe: Jeśli wartość zakupu jest większa niż 50 zł, wówczas rabat wynosi 3%; 100-7%; 150-10%, 250-15%. Zauważmy, iż mimo że obie dodatkowe właściwości zależą od kwoty zakupów, to jedna nie wpływa bezpośrednio na drugą.

Problem ISOP można zatem rozbudować i opisać dwoma dodatkowymi funkcjami wrażliwymi na łączną cenę kupowanych produktów. Przyjmijmy sytuację, w której chcielibyśmy kupić kilka produktów (n). Produkty oferowane są w wielu sklepach (m) – lokalizacja sklepów nie jest dla nas istotna. Cena każdego produktu n w każdym sklepie m może być zapisana jako p_{ij} , gdzie $i = 1, \dots, m$, $j = 1, \dots, n$. Dla każdego sklepu m zdefiniowana jest charakterystyczna dla niego funkcja rabatowa (na podstawie całkowitego kosztu zakupów w tym sklepie i) – $f_i(T_i)$. Ponadto, koszt wysyłki d dla każdego sklepu m jest zmienny i również zależy (choć charakterystyka zależności może być zupełnie inna) od kwoty zakupów. Naszym celem jest dowiedzieć się, jak zapłacić najniższy rachunek za wszystkie produkty. Matematyczne sformułowanie problemu może być zapisane w następujący sposób:

$$\min \sum_{i=1}^m f_i \left(\sum_{j \in N_i} p_{ij} x_{ij} \right) + \sum_{i=1}^m d_i \left(\sum_{j \in N_i} p_{ij} x_{ij} \right) \quad (6.1)$$

$$\text{p.o.} \quad \sum_{i \in M} x_{ij} = 1, \quad j = 1, \dots, n, \quad (6.2)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Zarówno funkcja rabatowa (f), jak i koszt wysyłki (d) mogą być agregowane w celu zaprezentowania jednej wspólnej funkcji (fd) w oparciu o oba kryteria. Sformułowanie można zapisać więc w następujący sposób:

$$\min \sum_{i=1}^m f d_i \left(\sum_{j \in N_i} p_{ij} x_{ij} \right) \quad (6.3)$$

$$\text{p.o.} \quad \sum_{i \in M} x_{ij} = 1, \quad j = 1, \dots, n, \quad (6.4)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Funkcja rabatowa może wyglądać następująco (przykład):

$$f_i(T_i) = \begin{cases} T_i & \text{jeżeli } 0 < T_i \leq 50; \\ 50 + 0,97(T_i - 50) & \text{jeżeli } 50 < T_i \leq 100; \\ 50 + 0,97 * 50 + 0,93(T_i - 100) & \text{jeżeli } 100 < T_i \leq 150; \\ 50 + 0,97 * 50 + 0,93 * 50 + 0,9(T_i - 150) & \text{jeżeli } 150 < T_i \leq 250. \end{cases}$$

Dostawa	Płatność	$T_i < 25$	$25 \leq T_i < 50$
Poczta	Karta	10	5
Poczta	Pobranie	15	10
Kurier	Karta	15	15
Kurier	Pobranie	20	20
Dostawa	Płatność	$50 \leq T_i < 75$	$T_i \geq 75$
Poczta	Karta	0	0
Poczta	Pobranie	5	0
Kurier	Karta	5	0
Kurier	Pobranie	10	0

Tabela 6.1. Przykładowe ceny wysyłki dla pewnego sklepu

Można zauważyć, że T_i to całkowita standardowa cena produktów wybranych w danym sklepie $i \in M$. Funkcja kosztów przesyłki jest bardziej skomplikowana, gdyż jej kształt zależy dodatkowo od późniejszego wyboru klienta (rodzaj dostawy – np. poczta, kurier, sposób płatności). Możemy ją zdefiniować jako $d_i(T_i, dv_1, dv_2)$, $i = 1, \dots, m$. Dlatego też, w niniejszej dyskusji możemy przyjąć, że klient zawsze podejmuje te same decyzje dv_1, dv_2 . Przykładowy cennik kosztów wysyłki znajduje się w tabeli 6.1.

Do rozwiązania problemu ISOPwDD można wykorzystać już istniejące heurystyki opracowane dla prostszych odmian problemu ISOP. Jednak przed zakończeniem eksperymentów obliczeniowych nie można stwierdzić, że będą one równie efektywne jak w przypadku prostszych wersji problemu ISOP. Ponadto, można przypuszczać, że dla tak rozbudowanego problemu jakim się przedstawia ISOPwDD, rozsądne wydaje się zaproponowanie zupełnie nowej, autorskiej, wyrafinowanej heurystyki. Odpowiednio opracowana ma szansę umożliwić uzyskiwanie jeszcze lepszych rezultatów obliczeń.

6.2 Algorytmy

Znalezienie dokładnego rozwiązania problemu ISOPwDD wymaga czasu wykładniczego, ale aby móc porównać wyniki heurystyk do rozwiązania optymalnego, zaimplementowano dokładny algorytm B&B z kilkoma jego ulepszeniami prowadzącymi do szybszego obliczenia najlepszego, możliwego rozwiązania.

Ponadto, opracowany i zaproponowany zostanie nowy, zaawansowany algorytm heurystyczny. Co więcej, ważne wydaje się porównanie nowej heurystyki z istniejącymi już algorytmami ISOP (oczywiście algorytmy mogą być dostosowane / przebudowane do odpowiadającej specyfiki problemu). Do analizy włączone zostaną również rozwiązania, które są wykorzystywane na stronach inter-

netowych porównywarek cen. Te ostatnie nazwijmy odpowiednio PCS (od Price Comparison Site) i jego poprawioną wersję PCS+. PCS zawsze wybiera sklep, w którym produkt jest najtańszy, natomiast PCS+ wybiera sklep z minimalną ceną łączną produktu i kosztu wysyłki.

Algorytm 9 G1

Require:

Liczba sklepów m
 Liczba produktów n
 Zbiór dostępnych produktów dla sklepu i , N_i
 Funkcja kolejności produktów σ
 Funkcja fd_i
 Ceny p_{ij}
 Koszty wysyłek d_i

Ensure: Tablica koszyków zakupów S

```

1:  $N := \{1..n\}$ 
2: for  $i := 1$  to  $m$  do
3:    $S[i] := \emptyset$ 
4: end for
5: for  $j := 1$  to  $n$  do
6:   Wybierz  $k \in M$  takie, że  $\sum_{i \in M} fd_i(T_i(S[i]) + \delta_{ik}p_{k\sigma(j)})$  jest minimalne i  $\sigma(j) \in N_k$ 
7:    $S[k] := S[k] \cup \{\sigma(j)\}$ 
8: end for
9: return  $S$ 

```

6.2.1 Proste algorytmy zachłanne G1 i G2

W celu wykonania pierwszych eksperymentów obliczeniowych dla nowego problemu ISOPwDD postanowiono przetestować dwie proste heurystyki znane z wykorzystania w problemie ISOP. Badanie uzupełnione o rozwiązanie optymalne, jak i dwie odmiany algorytmu działającego w porównywarek cen.

W naszym pierwszym algorytmie heurystycznym oznaczonym jako G1, produkty rozpatrywane są w określonej kolejności. Algorytm jest uruchamiany dla różnych wariantów zbioru (różni się kolejność umieszczonych produktów) produktów. Niech produkty zostaną określone w kolejności $1, \dots, n$. Wartość całkowita kosztów dostawy i standardowych cen (ofert) dla wszystkich sklepów są początkowo ustalone na $T_i = d_i$, $i = 1, \dots, m$. W każdej iteracji algorytmu G1 produkt j jest wybrany w sklepie $i \in M_j$ przy minimalnej wartości $f_i(T_i + p_{ij})$, a odpowiednia wartość T_i jest ponownie przypisywana do $T_i := T_i + p_{ij}$. Algorytm został początkowo opisany w podsekcji 4.4.4, natomiast jego zaktualizowaną wersję z zaprezentowanym pseudokodem obrazuje algorytm 9.

Drugim przebadanym algorytmem jest algorytm G2, który posiada mechanizm zapobiegania występowania niekorzystnej dla G1 sytuacji. Ogólną ideę pierwszej wersji algorytmu opisuje podsekcja 4.4.5. Natomiast dla problemu ISOPwDD wykorzystaliśmy zaktualizowaną, inaczej przygotowaną wersję tego algorytmu, gdzie ogólna zasada szybkiego działania i zapobiegania możliwej złej sytuacji występującej w przypadku G1 są zachowane (patrz algorytm 10). G2, w odróżnieniu od G1, rozpoczyna działanie od wyboru sklepu, w którym należy kupić produkty. Wybrany ma najniższą średnią cenę produktów łącznie z opłatami za dostawę. Dla każdego sklepu wyliczana jest wycena $V_i = (d_i + \sum_{j \in N_i} p_{ij}) / |N_i|$. Następnie ze sklepu k , który ma najniższą wycenę V_k , wybierany jest zbiór S produktów zgodnie z zasadą $\alpha\%$ najtańszych dostępnych produktów. Następnie, przedmioty te są usuwane ze sklepów $\forall i \in M, N_i := N_i \setminus S$, a d_k jest aktualizowane do wartości $d_k = 0$, ponieważ opłata za dostawę do sklepu k została już uiszczona.

Obie wymienione heurystyki zapewniają wydajną pracę (szybkie działanie) i dobre wyniki. Jednak, aby uzyskać jeszcze lepsze rozwiązania dla problemu ISOPwDD podjęto próbę stworzenia nowego, jeszcze wydajniejszego algorytmu heurystycznego.

6.2.2 Algorytm heurystyczny G3

Ideą nowo proponowanej heurystyki, oznaczonej jako G3 (algorytm 11), jest stworzenie grup produktów. Produkty są tak pogrupowane, aby maksymalizować stosunek wyceny V_i określający relacje oszczędności do maksymalnego kosztu we wszystkich sklepach.

W celu realizacji zadania dla każdego sklepu wybierany jest towar o największym wskaźniku V_i i jest odkładany do koszyka tymczasowego C_i . Jeśli przez dodanie innego produktu istnieje możliwość zwiększenia wskaźnika, produkt jest również brany pod uwagę w C_i . Proces ten trwa do momentu, gdy w sklepie nie ma już dostępnego produktu lub gdy wskaźnik nie może być poprawiony (zwiększony). Następnie wybierany jest sklep k o najwyższej wartości V_k i wszystkie pozycje w C_k umieszczane są w B_i – liście produktów do kupienia w tym sklepie. Krok ten jest powtarzany do momentu zakupu wszystkich produktów znajdujących się na liście zakupów.

Wyrażenie V_i uwzględnia koszyk B_i zawierający wszystko to, co zostało już wcześniej wybrane dla sklepu i , oraz tymczasowy koszyk C_i :

$$V_i(C_i, B_i) = 1 - \frac{f_i(T_i(B_i \cup C_i)) - f_i(T_i(B_i))}{p_{max}(C_i)}$$

G3 posiada złożoność obliczeniową $O(n^3m)$. Jest ona większa od złożoności algorytmu G1 (algorytm 9) i G2 (algorytm 10), ale warto zauważyć, że G3 nie wymaga żadnego dodatkowego strojenia i nie zależy od kolejności elementów. G3 ma natomiast kilka interesujących właściwości, takich jak górne ograniczenie i

Algorytm 10 G2

Require:

Liczba sklepów m
 Liczba produktów n
 Zbiór dostępnych produktów dla sklepu i , N_i
 Funkcja fd_i
 Ceny p_{ij}
 Koszty wysyłek d_i
 Parametr α

Ensure: Tablica koszyków zakupów S

```

1: while  $N \neq \emptyset$  do
2:   for  $i \in M$  do
3:     if  $S[i] = \emptyset$  then
4:        $V[i] := (d_i + \sum_{j \in N_i \cap N} p_{ij}) / |N_i \cap N|$ 
5:     else
6:        $V[i] := (\sum_{j \in N_i \cap N} p_{ij}) / |N_i \cap N|$ 
7:     end if
8:   end for
9:   Wybierz  $k$  takie, że  $V[k] = \min_{\{i \in M \mid N_i \cap N \neq \emptyset\}} V[i]$ 
10:  for  $c := 1$  to  $\max(1, \alpha * |N_k \cap N|)$  do
11:    Wybierz  $l$  takie, że  $p_{il} = \min_{j \in N_i \cap N} p_{ij}$ 
12:     $S[k] := S[k] \cup \{l\}$ 
13:     $N := N \setminus \{l\}$ 
14:  end for
15: end while
16: return  $S$ 

```

optymalność w niektórych szczególnych przypadkach, a w ogólności można się spodziewać lepszych wyników - co zostanie dokładnie sprawdzone w ramach eksperymentów obliczeniowych.

Algorytm 11 G3

Require:

Liczba sklepów m
 Liczba produktów n
 Zbiór dostępnych produktów dla sklepu i , N_i
 Funkcja f_i
 Ceny p_{ij} i $p_{max}(j)$
 Koszty wysyłek d_i

Ensure: Tablica koszyków zakupów S

```

1:  $N := \{1, \dots, n\}$ 
2: while  $N \neq \emptyset$  do
3:   for  $i = 1$  to  $m$  do
4:      $C_i := \emptyset$ 
5:      $val_i := 0$ 
6:      $\forall j \in N \cap N_i, r[j] := V_i(\{j\}, S[i])$ 
7:     Wybierz  $k$  takie, że  $r[k] = \max_{j \in N_i \cap N} r_j$ 
8:     repeat
9:        $C_i := C_i \cup \{k\}$ 
10:       $val_i := r[k]$ 
11:       $\forall j \in N \cap N_i \setminus C_i, r[j] := V_i(C_i \cup \{j\}, S[i])$ 
12:      Wybierz  $k$  takie, że  $r[k] = \max_{j \in N_i \cap N \setminus C_i} r_j$ 
13:    until  $N \cap N_i \setminus C_i = \emptyset$  or  $r[k] < val_i$ 
14:   end for
15:   Wybierz  $k$  takie, że  $val_k = \max_{i \in \{1..m\} | C_i \neq \emptyset} val_i$ 
16:    $S[k] := S[k] \cup C_k$ 
17:    $N := N \setminus C_k$ 
18: end while
19: return  $S$ 

```

6.3 Analiza teoretyczna algorytmu G3

W tej sekcji zostaną przeanalizowane najbardziej interesujące właściwości algorytmu G3.

6.3.1 Górne ograniczenie aproksymacji

Twierdzenie 1 Dla zadanej instancji problemu ISOPwDD, gdzie istnieje n produktów i m sklepów, z optymalnym rozwiązaniem c^* , $\forall i \in M$, $\forall x \in [0, \sum_{i \in M} \max_{j \in N_i} p_{ij}]$, $f_i(x) \geq \beta x$, koszt rozwiązania sugerowany przez G3 to c_{G3} , gdzie:

$$c_{G3} \leq (n - 1) \max_{i \in M} d_i + \frac{c^*}{\beta}$$

Dowód 7 Niech S^* będzie rozwiązaniem optymalnym.

Rozważmy k -tą iterację algorytmu G3; dla każdej poprzedniej iteracji l produkty $C^{(l)}$ zostały wybrane ze sklepu $i^{(l)}$, pozostało więc wybrać miejsce zakupu produktów $N^{(k)} = N \setminus \bigcup_{1 \leq l < k} C^{(l)}$.

Niech $B_i^{(<k)} = \bigcup_{q < k | i^{(q)} = i} C^{(q)}$ będzie koszykiem produktów wybranych w sklepie i przed k -tą iteracją.

$C_i^{(k)}$ jest zestawem przedmiotów wybranych w sklepie i w momencie k -tej iteracji. $C^{(k)}$ określa produkty wybrane po zakończeniu tej iteracji w sklepie $i_{\max(k)}$. Dla uproszczenia notacji po zakończeniu k -tej iteracji, (k) jest pomijane:

$$\begin{aligned} C_i^{(k)} &= C_i \\ B_i^{(<k)} &= B_i \\ i_{\max(k)} &= i_{\max} \end{aligned}$$

$$\forall i \in M, V_{i_{\max}}(C_{i_{\max}}, B_{i_{\max}}) \geq V_i(C_i, B_i)$$

Dla danego $i \in M$, niech x_1 będzie pierwszym elementem budowy C_i :

- $V_{i_{\max}}(C_{i_{\max}}, B_{i_{\max}}) \geq V_i(\{x_1\}, B_i)$
- $\forall x \in N_i \cap N^{(k)}, V_i(\{x_1\}, B_i) \geq V_i(\{x\}, B_i)$

Niech $i^*(x)$ będzie sklepem, w którym x jest kupowane dla osiągnięcia rozwiązania optymalnego, tj. $x \in S_{i^*(x)}$

Tak więc,

$$\forall x \in C_{i_{\max}}, V_{i_{\max}}(C_{i_{\max}}, B_{i_{\max}}) \geq V_{i^*(x)}(\{x\}, B_{i^*(x)}) \quad (6.5)$$

i

$$\begin{aligned} V_{i_{\max}}(C_{i_{\max}}, B_{i_{\max}}) &\geq \frac{\sum_{x \in C_{i_{\max}}} p_{\max}(x) V_{i^*(x)}(\{x\}, B_{i^*(x)})}{\sum_{x \in C_{i_{\max}}} p_{\max}(x)} \quad (6.6) \end{aligned}$$

tj.

$$\begin{aligned} f_{i_{\max}}(C_{i_{\max}} \cup B_{i_{\max}}) - f_{i_{\max}}(B_{i_{\max}}) &\leq \sum_{x \in C_{i_{\max}}} f_{i^*(x)}(x) \quad (6.7) \end{aligned}$$

Niech q oznacza łączną liczbę iteracji algorytmu G3, dla danej instancji,

$$\sum_{k=1}^q f_{i_{max(k)}} \left(C_{i_{max(k)}}^{(k)} \cup B_{i_{max(k)}}^{(<k)} \right) - f_{i_{max(k)}} \left(B_{i_{max(k)}}^{(<k)} \right) \leq \sum_{k=1}^q \sum_{x \in C_{i_{max(k)}}^{(k)}} f_{i^*(x)}(x) \quad (6.8)$$

$$\begin{aligned} c_{G3} &\leq \sum_{k=1}^q \sum_{x \in C^{(k)}} f_{i^*(x)}(x) \\ c_{G3} &\leq \sum_{k=1}^q \sum_{x \in C^{(k)}} d_{i^*(x)} + p_{i^*(x)}x \\ c_{G3} &\leq \sum_{x \in N} d_{i^*(x)} + p_{i^*(x)}x \\ c_{G3} &\leq (n-1) \max_{i \in M} d_i + \frac{1}{\beta} c^* \end{aligned}$$

Warto scharakteryzować przypadek struktury najgorszego rozwiązania. Ponieważ każdy koszyk jest skonstruowany iteracyjnie, mogą wystąpić przypadki gdzie zakup dwóch produktów nie poprawia wartości wskaźnika V_i . W takich przypadkach, algorytm może na każdym etapie wybrać jeden obiekt w nowym sklepie, w którym cena tego obiektu jest niższa niż wszędzie indziej (łącznie z kosztem dostawy).

Optymalną strategią może być kupno wszystkiego w jednym sklepie, w którym ceny pojedynczych przedmiotów są nieco wyższe niż te wybrane przez algorytm, ale za to sklep proponuje bardzo duże rabaty przy zakupach o większej wartości. Prowadzi to do ograniczenia, w którym jego lewa strona wyrażona jest faktem, że algorytm mógłby korzystać z wielu różnych sklepów, zaś prawa strona pokazuje, że optymalne rozwiązanie mogłoby korzystać z wielkiego rabatu jednego sklepu, inaczej niż to proponuje algorytm.

6.3.2 Optymalność w uproszczonej formie

W przypadku, gdy wszystkie sklepy posiadają wszystkie produkty i oferują je w tej samej cenie, problem można rozwiązać w czasie wielomianowym ($O(n + m)$) za pomocą określonego algorytmu [10]. Przy tych założeniach optymalnym rozwiązaniem jest również to wyliczone przez G3.

Twierdzenie 2 *Jeśli $\forall i \in M, N_i = N$ i $\forall j \in N$ i $(j, j') \in M^2, p_{ij} = p_{ij'}$ to G3 dostarcza rozwiązania optymalnego.*

Dowód 8 Dla danego $i \in M$, $E \subset N$ i $j \in N \setminus E$, niech zbiór $s = \sum_{k \in E} p_{ik}$, zatem, korzystając z faktu, że g_i jest wklęsłe i niemalejące, a także, że h_i jest malejące:

$$\begin{aligned} V_i(E) - V_i(E \cup \{j\}) \\ \leq \frac{g_i(T_i(E) + p_{ij})}{s + p_{ij}} + \frac{g_i(T_i(E))}{s} + \frac{h_i(s + p_{ij})}{s + p_{ij}} - \frac{h_i(s)}{s} \end{aligned} \quad (6.9)$$

$$\begin{aligned} V_i(E) - V_i(E \cup \{j\}) \\ \leq \frac{g_i(T_i(E)) + \frac{p_{ij}}{s} * g_i(s) - \frac{s+p_{ij}}{s} * g_i(T_i(E))}{s + p_{ij}} \end{aligned} \quad (6.10)$$

tj.

$$V_i(E) - V_i(E \cup \{j\}) \leq 0$$

Tak więc, dla wszystkich podzbiorów E należących do N , dla każdego elementu $x \in N \setminus E$, $V_i(E \cup \{x\}) \geq V_i(E)$. Analizując sposób przetwarzania algorytmu G3, w pierwszej iteracji wybrane zostają wszystkie produkty spośród oferty wszystkich sklepów aby maksymalizować V_i : $\forall i, C_i = N$.

Niech i_{max} będzie wartością i która maksymalizuje $V_i(C_i)$.

$$\forall i \in M, V_{i_{max}}(C_{i_{max}}) \geq V_i(C_{i_{max}})$$

$$\forall i \in M, V_{i_{max}}(N) \geq V_i(N)$$

$$\forall i \in M, f_{i_{max}}(N) \leq f_i(N)$$

Przy tak zdefiniowanym problemie udowodniono, że wszystkie produkty są kupowane tylko w jednym sklepie [10]. Tak więc G3 oblicza rozwiązanie optymalne.

6.4 Aplikacja testowa z generatorem instancji

Wyzwaniem w badaniach eksperymentalnych było stworzenie modelu, który byłby jak najbliższy rzeczywistym warunkom zakupów w Internecie. Jako podstawę użyto pomysłów, które służyły budowaniu modeli dla problemów ISOP i ISO-PwD. Dokonano jednak pewnych znaczących modyfikacji struktury cen dla jeszcze lepszego odwzorowania sytuacji rynkowych. Dla naszego nowego modelu przyjęliśmy pewne informacje i wyniki obliczeń zawarte w literaturze [81, 84, 80, 19]. Dane do naszej próby zostały zebrane z trzydziestu dwóch sklepów i objęły największe sklepy amerykańskie, w tym Amazon, BarnesandNoble.com, Borders.com, Buy.com i Booksamillion, oraz czołowych sprzedawców wśród księgarń internetowych w Polsce, takich jak empik.com i merlin.pl.

Zakładamy, że wszystkie testy powinny być wykonane dla kilku zbiorów sklepów $m \in \{20, 30, 40\}$, z określoną liczbą produktów w koszyku. Zakłada się, że każdy sklep posiada wszystkie wymagane produkty. W każdym przypadku następujące wartości są losowo generowane dla wszystkich i i j w odpowiednich zakresach. *Cena referencyjna (ref)* produktu j : $ref_j \in \{2, 4, \dots, 100\}$ z prawdopodobieństwem wystąpienia: 40% między 0 a 20, 16% między 22 a 30, 12% między 32 a 40, 16% między 42 a 60 i 16% między 62 a 100. Cena produktu j w sklepie i wynosi $p_{ij} \in [a_{ij}, b_{ij}]$, gdzie $a_{ij} \geq 0,75ref_j$, $b_{ij} \leq 1,36ref_j$, a struktura interwałów $[a_{ij}, b_{ij}]$ wygląda następująco:

[8%]	<i>minimum</i>
	$minimum + \frac{(ref - minimum)}{4}$
[3%]	$minimum + \frac{(ref - minimum)}{2}$
[9%]	$minimum + \frac{(ref - minimum)}{1.25}$
[21%]	<i>ref</i>
[24%]	$ref + \frac{(maximum - ref)}{4}$
[9%]	$ref + \frac{(maximum - ref)}{2}$
[10%]	$ref + \frac{(maximum - ref)}{1.25}$
[16%]	<i>maximum</i>

Każdy sklep pobiera opłatę za dostawę, ustaloną z przedziału od 0 do 20. Funkcja określająca rabaty jest generowana wedle poniższej obserwacji:

- Na niektórych stronach internetowych możemy zauważyć, że np. jeśli dokonamy zakupów o wartości większej niż 150 zł, to zapłacimy połowę kosztów wysyłki. Jeśli natomiast łączna kwota zakupów przewyższy 250 zł, to dostawa jest bezpłatna. Aby symulować podobne sytuacje, opłatę za dostawę można przykładowo podzielić na trzy progi: jeśli koszt przesyłki jest podany jako wielokrotność k , to istnieją progi k i każdorazowo odliczana jest jedna kwota k od opłaty za przesyłkę.
- Ponadto, powszechnie stosowane są rabaty, które zależą od łącznej sumy wartości produktów w koszyku. Przykładowo sklep udziela 5% rabatu jeśli wydamy więcej niż 200 zł, 10%, jeśli produkty są warte ponad 400 zł, itd., itp.

Obie strategie mogą być łączone, a wszystkie progi wykorzystywane do generowania danych są wielokrotnością 25, aby najlepiej odzwierciedlić obserwacje prawdziwych stron (sklepów) internetowych.

6.5 Eksperymenty obliczeniowe

Dla bardzo wielu grup produktów serwisy porównujące ceny przedstawiają oferty dla około 25 – 30 sklepów. Dlatego też do naszych testów wykorzystaliśmy trzy zbiory sklepów m o licznosci $\{20, 30, 40\}$. Podobnie, przeciętny klient kupi tylko kilka książek w tym samym czasie, więc przyjęto, że n należy do zbioru $\{2, 3, 4, 5, 6, 7\}$. Wybory te uwzględniają również fakt, że poszukiwanie optymalnego rozwiązania jest wykładnicze w czasie.

Aby zbadać, jak algorytmy działają na znacznie większych zbiorach danych, za n przyjęto wartości od 5 do 100 rosnące o 5. Dla dużego zestawu danych heurystyki G1, G2, PCS i PCS+ będą porównywane z wynikami G3. Dla każdego eksperymentu przeprowadzono badania oparte na 400 wygenerowanych instancjach problemu.

6.6 Wyniki eksperymentów obliczeniowych – analiza i dyskusja

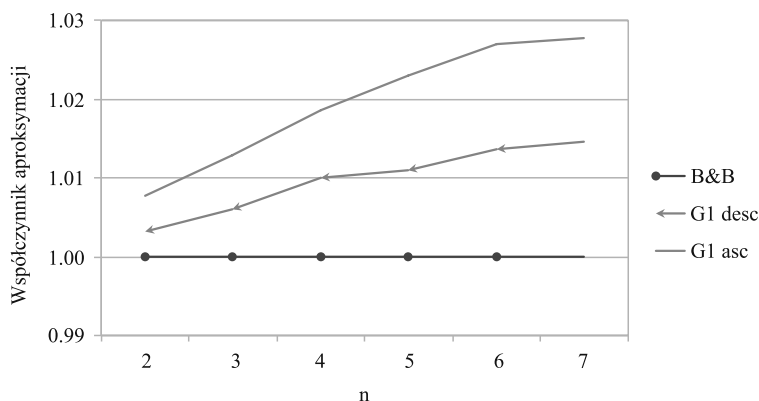
Niniejsza sekcja zawiera informacje dotyczące wyników eksperymentów obliczeniowych. Przedstawiono analizę wpływu kolejności produktów przy korzystaniu z algorytmu G1, wrażliwość algorytmu G2 na dobór parametrów. Następnie zaprezentowano wyniki eksperymentu wraz z komentarzem.

6.6.1 Wpływ kolejności produktów na wyniki obliczeń algorytmu G1

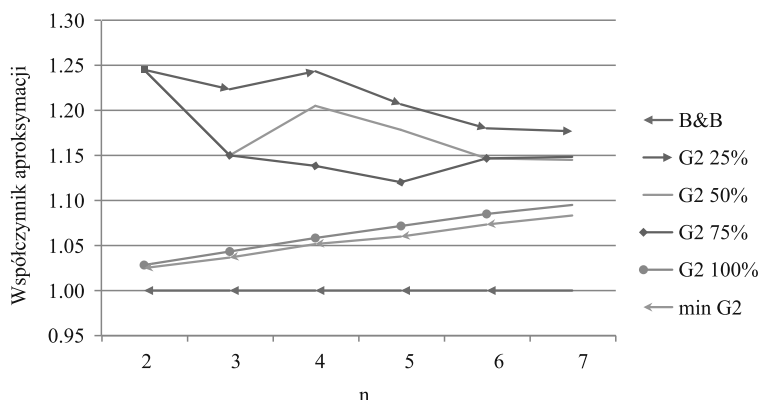
G1 został przetestowany z dwiema różnymi strategiami uszeregowania produktów w koszyku wedle cen referencyjnych rosnących i malejących. Strategia wykorzystująca ceny malejące daje w ogólności lepsze wyniki (rysunek 6.1). Ponadto, G1 dostarcza przybliżone rozwiązania w zasięgu 2,5% wartości optymalnych. Obrazuje to fakt, że prosty algorytm zachłanny może przedstawić dobre rozwiązania w bardzo krótkim czasie. Złożoność obliczeniowa algorytmu G1 wynosi $O(nm)$, ale sortowanie produktów również zajmuje co najmniej $O(n \ln n)$.

6.6.2 Wrażliwość algorytmu G2 na dobór parametrów

W celu ustalenia wrażliwości G2 na parametr α , przeprowadzono badanie z 4 różnymi wartościami: 25%, 50%, 75%, oraz 100%. Uwzględniono również minimalną wartość z 4 testów. Można zatem wnioskować, że G2 wymaga strojenia, aby móc zapewnić dobre przybliżenia (rysunek 6.2). Nawet najlepsze rozwiązanie wykazuje wynik o 5% gorszy od wartości optymalnej i to dla małej liczby elementów ($n = 4$). Wybór parametru α ma duży wpływ na wydajność G2 – należy go zatem dobrać bardzo starannie.



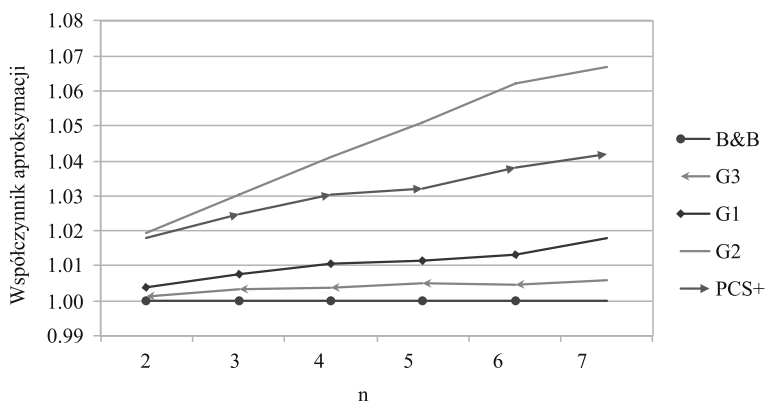
Rysunek 6.1. Wyniki algorytmów G1 rosnący, G1 malejący i B&B dla $m = 30$ (przedział ufności = 0,95)



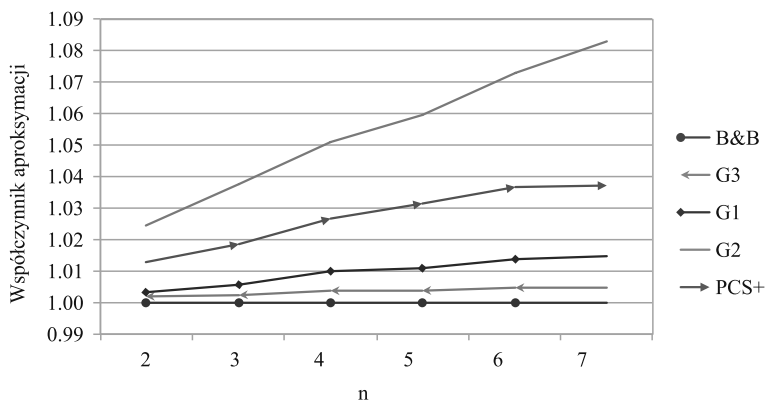
Rysunek 6.2. Wyniki różnych wersji (parametrów) algorytmów G2 i B&B dla $m = 30$ (przedział ufności = 0,95)

6.6.3 Wyniki obliczeń

Dla każdego zestawu par (n, m) algorytmy PCS+ i G3 zostały przetestowane względem rozwiązania G1 przy strategii cen malejących, a następnie porównane do optymalnego rozwiązania. Ponieważ G2 wymaga dostrojenia, wzięto pod uwagę najlepsze rozwiązanie dostarczone przez G2 przy użyciu parametrów 25%, 50%, 75% i 100%. Pominęto wyniki algorytmu PCS, ponieważ dostarczył rozwiązania, które były o co najmniej 15% gorsze od rozwiązania optymalnego.



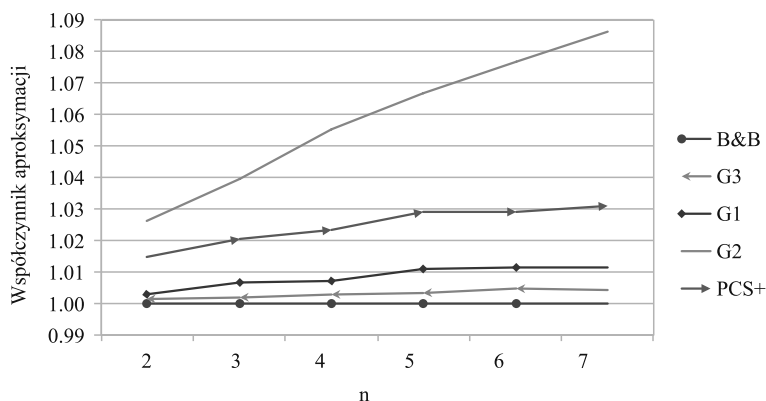
Rysunek 6.3. Wyniki działania algorytmów G1, G2, G3, PCS+, B&B dla $m = 20$ (przedział ufności = 0,95)



Rysunek 6.4. Wyniki działania algorytmów G1, G2, G3, PCS+, B&B dla $m = 30$ (przedział ufności = 0,95)

G3 dostarcza rozwiązań, które są co najwyżej o 0,59% gorsze od rozwiązania optymalnego. To zdecydowanie najlepsza ze wszystkich testowanych heurystyk (patrz rysunki 6.3, 6.4 i 6.5). Co ciekawe, prosty algorytm G1 zapewnia dużo lepsze rozwiązania niż PCS+. Współczynnik aproksymacji dla G3 wzrasta wraz ze wzrostem liczby elementów, ale znacznie wolniej niż to ma miejsce w przypadku PCS+ i G1.

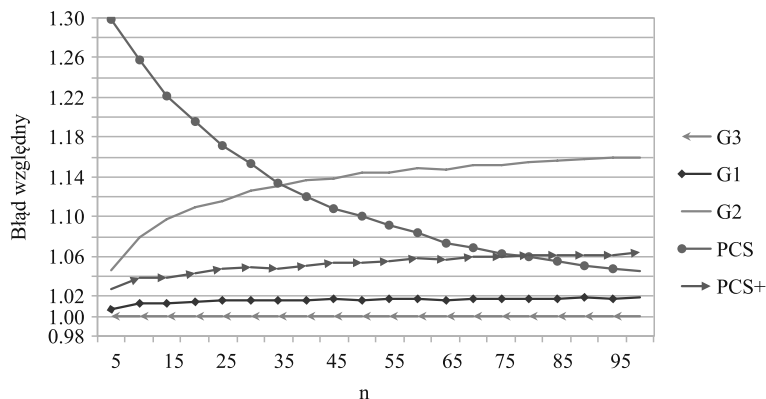
Warto podkreślić, że G2 jest bardzo zależny od liczby sklepów: dla $m = 20$, wskaźnik aproksymacji G2 nie przekracza 7% dla n w $\{2, 3, 4, 5, 6, 7\}$. Natomiast



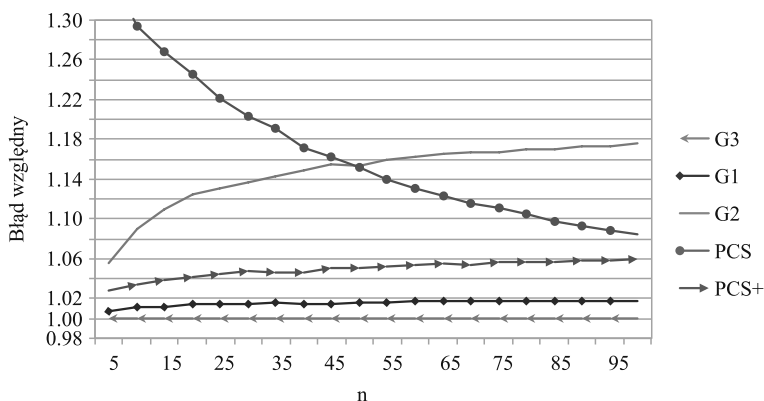
Rysunek 6.5. Wyniki działania algorytmów G1, G2, G3, PCS+, B&B dla $m = 40$ (przedział ufności = 0,95)

dla $m = 40$ i $m = 30$, szybko przekracza tę wartość i rośnie dla $n > 5$. G3 zapewnia dobre rozwiązania zbliżone do optymalnych, a jednocześnie lepsze od wszystkich pozostałych algorytmów heurystycznych.

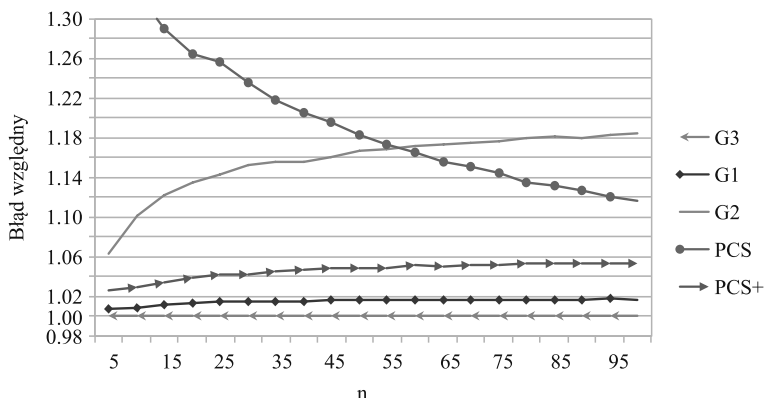
6.6.4 Skalowalność algorytmów – eksperymenty na dużych zestawach danych



Rysunek 6.6. Wyniki działania algorytmów G1, G2, G3, PCS i PCS+ dla $m = 20$



Rysunek 6.7. Wyniki działania algorytmów G1, G2, G3, PCS i PCS+ dla $m = 30$



Rysunek 6.8. Wyniki działania algorytmów G1, G2, G3, PCS i PCS+ dla $m = 40$

Przy eksperymencie obliczeniowym dla instancji o większych rozmiarach problemu porównano wyniki obliczeniowe wszystkich heurystyk. Zamieszczenie wyników działania algorytmu dokładnego B&B było niemożliwe ze względu na jego dużą złożoność obliczeniową i czasy zbyt duże do zarejestrowania. Bez poszukiwania optymalnego rozwiązania, parametr n mógł być znacznie większy niż w poprzednich testach. Przyjęto zatem, że n jest wybierany ze zbioru $\{5, 10, \dots, 95, 100\}$, natomiast m pozostaje bez zmian.

G3 uzyskuje lepsze wyniki niż inne heurystyki, nawet przy bardzo dużej

liczbie produktów (rysunki 6.6, 6.7 i 6.8). G1 oblicza rozwiązania, które są co najmniej 0,7% gorsze, a te z PCS+ co najmniej 2,5%. Różnice szybko rosną, dla n większego od 20. Różnica między G1 i G3 wynosi co najmniej 1,3%.

Rozwiązania oferowane przez PCS stają się coraz lepsze wraz ze wzrostem liczby produktów, gdyż wpływ opłat za dostawę na koszt całkowity maleje wraz ze wzrostem liczby pozycji w koszyku.

G3 zapewnia lepsze rozwiązania niż inne algorytmy, nawet przy większej liczbie elementów. Należy podkreślić, że różnice mogą sięgać nawet znacznie ponad 30% na korzyść algorytmu G3. Ponadto, lepiej radzi sobie w sytuacjach, gdy nie wszystkie produkty są dostępne, co czyni go najbardziej wszechstronną heurystyką. Niemniej jednak, należy zauważyć, że G3 ma wyższy stopień złożoności obliczeniowej ($O(mn^3)$) niż choćby G1 ($O(nm)$).

6.7 Podsumowanie

W niniejszym rozdziale przedstawiono problem ISOPwDD, który jest znacznie bardziej skomplikowaną odmianą problemu optymalizacji zakupów internetowych. Zdecydowano się przedstawić bardziej rozbudowaną, bardziej realistyczną i jeszcze ciekawszą wersję problemu ISOP. Dodano dwie funkcje zmieniające dynamicznie koszt zakupu. Pierwsza z nich określa rabaty cenowe, jakie mogą zostać zaproponowane przez sprzedawców. Jest to często stosowana praktyka w sklepach internetowych, mająca zachęcić klientów do dokonania większych zakupów (nierzadko większych niż początkowo zakładali) w ramach ich sklepu internetowego. Ma to tym większe znaczenie, że kupując większą ilość produktów z jednego sklepu oszczędzamy na kosztach wysyłki – jak to było podkreślone już przy definicji oryginalnego, podstawowego problemu ISOP. Kolejną, niezależną od pierwszej, funkcją jest ta opisująca koszty wysyłki. W zależności od sposobu wysyłki, jak i kwoty przeznaczonej na zakupy w danym sklepie, wartość wysyłki może być dowolnie modyfikowana. Powstały problem zbudowany na bazie ISOP został zatem wzbogacony o dwie niezależne funkcje wpływające na finalną kwotę zakupów.

W ramach prac opracowano zupełnie nowy algorytm oznaczony jako G3, a także przygotowano nowe przebudowane wersje algorytmów G1 i G2. Szczegółnej analizie poddano oczywiście nową heurystykę G3. Określono jej górne ograniczenie oraz opisano optymalność w szczególnym przypadku, gdy problem można rozwiązać za pomocą algorytmu wielomianowego. Przeprowadzono ponadto szereg eksperymentów obliczeniowych (dla samego “dużego” zestawu danych przeprowadzono 120 tys. pomiarów) z wykorzystaniem wielu heurystyk i rozwiązania optymalnego. Warto zauważyć, że zaproponowano nowy, bardziej realistyczny model danych. Wyniki z eksperymentów wykazały, że G3 zapewnia lepsze rozwiązania inne niż istniejące heurystyki, nawet jeśli produkty mogą być niedostępne.

Rozdział 7

Zaufanie w zakupach internetowych

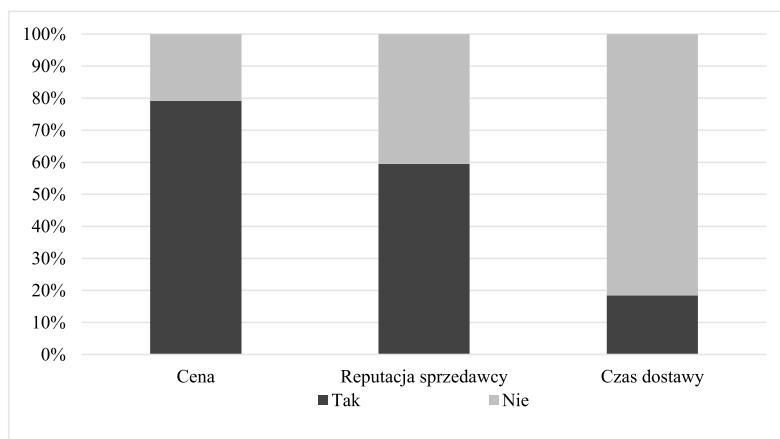
Prowadzenie sklepu internetowego może skutkować uzyskaniem wielu korzyści względem tradycyjnego lokalnego modelu sprzedaży (ang. brick-and-mortar shops). Jeśli chodzi o sprzedającego, główne korzyści ekonomiczne obejmują to, że może on znacznie zmniejszyć koszty związane z leasingiem i wynajmem lokali, czy zatrudnianiem osób niezbędnych do efektywnego prowadzenia swojej działalności handlowej. Ponadto, ponieważ zjawisko handlu elektronicznego jest stosunkowo nowe i zazwyczaj transakcje zawierane są w oparciu o inne lokalizacje niż miejsce zamieszkania klienta, mają oni większą swobodę działania w porównaniu do ich fizycznych odpowiedników. Te udogodnienia zarówno dla klientów, jak i sprzedawców, które wynikają z handlu elektronicznego, sprawiły, że liczba sklepów internetowych stale i dynamicznie rośnie. Dla sprzedawców wygodniejsze jest dołączenie do istniejącego rynku internetowego (aukcje lub inne istniejące platformy), lub wdrożenie własnej strony internetowej. Jak wspomniano wcześniej, większa konkurencja oznacza niższe ceny, ale oznacza również, że klient może zostać przytłoczony ogromną liczbą ofert, co sprawia, że bardzo trudno jest mu dokonać odpowiedniej analizy przed-zakupowej. Szczególnie jeśli zakupy dotyczą nie jednego, a większej ilości produktów. Problem ISOP (rozdział 4) rozważa opisywane kwestie. Należy jednak pamiętać, że poza analizą cen towarów pojawiają się inne obawy większości użytkowników. Wynikają one z charakteru działania rynku e-commerce. Można do nich zaliczyć m.in. brak poczucia bezpieczeństwa klienta, którego doświadcza przy podawaniu danych osobowych i informacji finansowych w celu realizacji transakcji, czy np. niemożność sprawdzenia przedmiotów lub wypróbowania ich przed dokonaniem zakupu [52].

Co niezmiennie istotne, klienci zwykle wolą rezygnować z tańszych ofert, które

są proponowane przez mniej renomowanych sprzedawców, na rzecz zakupu droższego produktu, oferowanego przez znany i zaufany sklep o większym prestiżu wśród innych konsumentów [108, 45]. Mając na uwadze takie zachowanie obserwowane wśród większości klientów sklepów internetowych, zdecydowano o rozszerzeniu problemu optymalizacji zakupów internetowych o dodatkowe wskaźniki związane z renomą sklepów i czynnikiem zaufania.

Warto podkreślić, że według badania przeprowadzonego przez firmę Nielsen [77] wśród ponad 27 000 konsumentów internetowych z 55 rynków na całym świecie, użytkownicy platform e-commerce doceniają opinie i recenzje innych klientów. Traktują je nawet jako decydujący czynnik w procesie decyzyjnym dotyczącym ewentualnych zakupów z danego sklepu. Zhang, Cheung i Lee [111] zbadali interesujący problem (pytanie), w jaki sposób niespójne opinie odgrywają rolę w podejmowaniu przez konsumentów decyzji o zakupach online.

W pracy Flanagin et al. [30] odniesiono się do problemu wykorzystywania ocen sprzedawców internetowych oraz sposobu, w jaki klienci używają ich do wybierania sklepów internetowych o lepszej reputacji. Wykorzystanie tych wskaźników może pomóc w opracowaniu systemu, w ramach którego potencjalny nabywca może kupić listę produktów od renomowanych firm internetowych po najlepszej cenie.

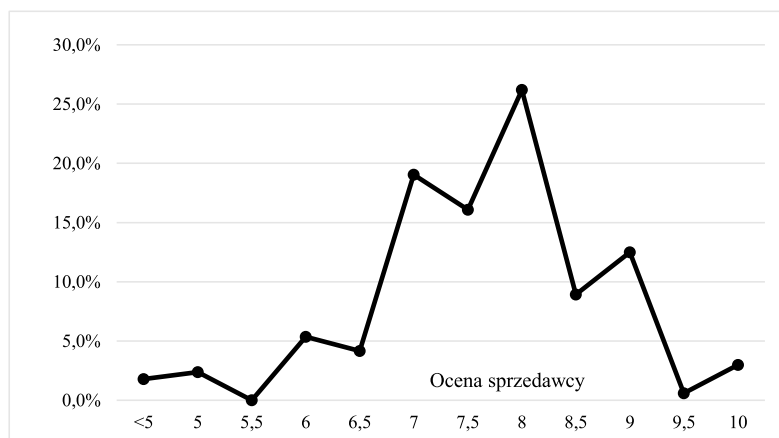


Rysunek 7.1. Kluczowe aspekty podczas zakupów online

7.1 Badanie ankietowe

Przygotowując model ISOP z analizą zaufania zdecydowaliśmy się wykorzystać informacje zawarte w literaturze [18] oraz własne badania, spostrzeżenia i obserwacje. W związku z tym opracowano ankietę uzupełniającą wspomniane już in-

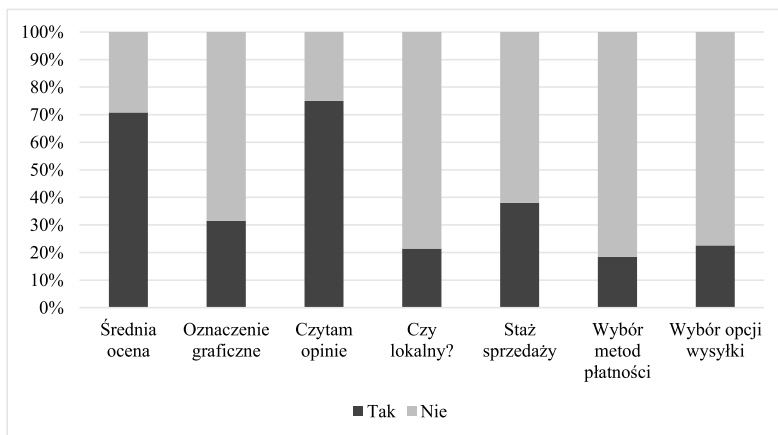
formacje. Celem było uzyskanie odpowiedzi na najważniejsze pytania dotyczące zaufania, jakim klient może obdarzyć sprzedawców podczas zakupów internetowych. Udało nam się zebrać 168 wypełnionych ankiet. Następnie zostały one dokładnie przeanalizowane [78]. Odpowiedzi udzielone w ankiecie pozwalają na zaobserwowanie następujących wniosków:



Rysunek 7.2. Minimalna wymagana ocena sprzedawcy w skali 1-10. Na osi Y oznaczono udział odpowiedzi dla populacji

Prawie 80% respondentów stwierdziło, że cena jest kluczowym czynnikiem przy wyborze sklepu podczas dokonywania zakupów internetowych. Powszechnie wiadomo, że bez względu na to, z jakim rodzajem produktów mamy do czynienia, cena jest bardzo ważna. Zwłaszcza gdy patrzymy na rynek zakupów internetowych, gdzie z pozycji naszego komputera dostępnych jest wiele ofert z różnych sklepów. Dla nieco mniej niż 20% uczestników badania czas dostawy jest ważnym elementem. Z jednej strony obserwacja ta może być zaskakująca. Bardzo często jednak stawiamy na jakość, oceny i cenę w zamian za jeden lub dwa dodatkowe dni spędzone w oczekiwaniu na dostawę zamówienia. Dlatego 137 osób odpowiedziało, że czas dostawy nie jest ważnym czynnikiem. Dla 60% reputacja sprzedawcy będzie bardzo ważnym / kluczowym czynnikiem. Rysunek 7.1 przedstawia procent odpowiedzi *tak* / *nie*, gdy pytamy o kluczowe czynniki podczas zakupów online.

Czy ocena sklepu przedstawiona w postaci numerycznej jest ważna? Czy klienci zwracają na to uwagę? 71% powiedziało *tak*. Wyniki te mogą być związane z kolejnym pytaniem o ogólną opinię / renomę sprzedawców. Czy to ważne i jak bardzo? Dokładnie dla 159 osób (co stanowi ogromną grupę 95% badanej populacji) opinie o sklepie są albo bardzo ważne, albo znaczące. Tylko jedna osoba nie zwraca na to uwagi. W sumie te dwa pytania pokazały, że podczas zakupów w sklepach internetowych całkowicie polegamy (i cenimy) na opiniach



Rysunek 7.3. Ważne aspekty gdy użytkownik zbiera informacje dotyczące sprzedawcy

innych ludzi, których nawet nie znamy. Uważamy to za jeden z najważniejszych wniosków płynących z tego badania.

Bardzo często oceny sklepów są wyróżniane graficznie (np. tzw. gwiazdki). Tylko dla 32% respondentów czynnik ten jest bardzo istotny. Jak można to wytłumaczyć? Oceny graficzne są znacznie mniej dokładne niż liczby. Ponadto nie ma ścisłych zasad dotyczących sposobu, w jaki sprzedający otrzymują inną ocenę (kolejna “gwiazdka”, czy oznaczenie innego koloru) i na jakich warunkach. Ogólnie rzecz biorąc, jeśli coś jest niezrozumiałe, to jest dla nas mniej interesujące. Jednak 32% nie jest zupełnie bez znaczenia, dlatego też oceny graficzne mogą być bardzo dobrym uzupełnieniem najważniejszych ocen liczbowych i opinii (należy pamiętać, że znaczenie tego czynnika zostało podkreślone przez wszystkie osoby). Jako wynik uzupełniający można zauważyć, że 75% respondentów czyta opinie napisane przez innych klientów. Oznacza to, że wolimy spędzić trochę więcej czasu, aby uzyskać bardziej szczegółowe informacje o sklepach i sprzedawcach. Okazuje się również, że ludzie nie są szczególnie mocno związani z lokalnymi sklepami (znajdującym się w ich kraju). Tylko dla 21% osób lokalizacja sklepów jest ważna i wolą dokonywać zakupy w sklepach lokalnych (w perspektywie kraju). Może to być związane z pewnym wsparciem dla lokalnych inicjatyw, albo z rodzajem strachu / obawy – dostawy z innego kraju, dodatkowymi podatkami, aspektami gwarancji itp. Nieco zaskakujące jest to, że 38% osób, które wypełniły ankietę stwierdziło, że długość czasu działania sklepu jest ważną informacją dotyczącą sprzedawcy. Nie spodziewaliśmy się, że tak wiele osób doceni ten fakt. Oczywiście jest jednak, że doświadczenie zawodowe i obecność sklepu na rynku zwiększa jego wartość / reputację (oczywiście, jeśli reputacja ta jest poparta pozytywnymi opiniami). Wydaje się również, że zarówno bardzo szeroki wachlarz

metod wysyłki, jak i płatności nie są aż tak znaczące. Tylko dla odpowiednio 23% i 18% osób jest to istotne. Na rysunku 7.3 przedstawiono zestawienie najważniejszych opisywanych aspektów.

Z obserwacji życia codziennego, jak również z niektórych znanych faktów z literatury i ogólnych opinii wynika, że ludzie nie cenią sprzedawców, którzy pobierają wysokie opłaty za dostawę (zauważalnie wyższe stawki niż reszta rynku). Tezę tę potwierdzają odpowiedzi udzielone w ankietach. Ponad 69% respondentów twierdzi, że zrezygnuje z zakupów w przypadku zbyt wysokich cen dostawy, nawet jeśli ceny produktów są atrakcyjne. Uważamy to za bardzo ważny wynik i spostrzeżenie. Jak można to wytłumaczyć? Jeśli trzeba zapłacić za dostawę znacznie więcej niż wynosi jej cena rynkowa, czujemy się oszukani lub “naciągani”. Mamy wrażenie kontaktu z osobą nieszczerą. Klient natychmiast traci zainteresowanie, a nawet niskie ceny produktów nie zachęcają go do zakupów.

Jaka powinna być minimalna ocena sprzedawców, aby pozostać w spektrum zainteresowania klientów? Ocena poniżej 5, w skali 1-10 (precyzja pomiaru 0,5) jest wystarczająca tylko dla 3 osób ($\approx 2\%$). Ocena na poziomie równym 5 wystarczy tylko dla bardzo małej grupy (2%). Kolejnym progiem jest ocena równa 6. Sklepy z co najmniej 6 punktami wzbudzają zainteresowanie 5% respondentów (nie liczymy tu wcześniej wspomnianych osób, a jedynie nowe). Kolejne 4% stwierdziło, że wskaźnik powinien wynosić co najmniej 6,5. Zatrzymajmy się na chwilę i sporządźmy podsumowanie. Sklep, który uzyskuje prawie $\frac{2}{3}$ maksymalnej oceny (widzimy, że np. w systemach rezerwacji hoteli nie jest to już zły wynik) jest interesujący tylko dla 13 osób (tj. 8% klientów). Kontynuując analizę można zauważyć, że 19% respondentów wymaga oceny numerycznej 7, a ponadto 16% deklaruje, że ocena sprzedawców powinna wynosić co najmniej 7,5. Największa grupa respondentów (dokładnie 26%) chciałaby zobaczyć wskaźnik na poziomie 8 lub wyższym. Dla około 9% klientów wynik co najmniej 8,5 będzie godny zaufania. Bardzo wysoka ocena, wynosząca co najmniej 9 punktów, jest obowiązkowa dla 12% respondentów. Oceny na poziomach 9,5 i 10 są wymaganiem odpowiednio dla 0,5% i 3%.

Rozproszenie (dystrybucja) powyższych wyników jest bardzo klarowne i stabilne. Należy zauważyć, że 3 największe grupy popierają trzy sąsiednie wartości ratingowe (7; 7,5; 8), co obejmuje ponad 61% badanych osób. Warto wspomnieć o dwóch dodatkowych uwagach. Sklepy posiadające ocenę na poziomie $\frac{2}{3}$ skali maksymalnej (co daje wynik 6,5) są interesujące tylko dla 14% klientów. Następnie, ponad 16% akceptuje tylko najlepsze sklepy (z oceną 9 lub wyższą). Generalnie obrazuje to fakt, że ludzie mają bardzo wysokie wymagania w stosunku do reputacji / rankingów sklepów internetowych. Rysunek 7.2 ilustruje aktualne wyniki, które przybierają kształt rozkładu normalnego.

Prawie 73% osób zdyskredytuje sprzedawców posiadających odsetek zdecydowanie negatywnych opinii, nawet jeśli stanowią oni mniejszość. Pokazuje to, że jedna zła opinia ma znacznie większą wartość ważoną (ujemną) niż jedna opinia pozytywna.

Bardzo ciekawym aspektem było kluczowe pytanie dotyczące relacji ceny do wskaźnika oceny sklepu. 83% respondentów chętnie płaci więcej za produkty kupowane od cieszących się dobrą reputacją sprzedawców. Największa grupa (powyżej 36%) może dodatkowo zapłacić 10%. Nieco mniejsza grupa 31% osób deklaruje za ten komfort przepłacić 5%. Natomiast 12% populacji może zapłacić 20% więcej, a niewiele ponad 2% osób zadeklarowało, że wyda nawet 30% więcej, aby mieć pewność zakupu u renomowanego sprzedawcy. Należy jednak zauważyć, że 17% badanych nie chce płacić żadnej dodatkowej kwoty.

36% respondentów zrezygnuje z zakupów, jeśli sprzedawca nie obsługuje ich ulubionej formy wysyłki. Jednak na obecnym rynku zakupów internetowych wiele sklepów oferuje podobny, szeroki wybór metod wysyłki.

Ostatnie pytanie dotyczyło własnej opinii klientów (na podstawie poprzednich zakupów lub po prostu wrażenia o sklepie). Dystrybucja odpowiedzi jest prawie taka sama, jak w przypadku pytania o ogólną opinię sklepu – ogromna grupa 90% zadeklarowała, że ich opinia jest albo bardzo ważna, albo istotna. Tylko dla 7% jest to w ogóle nieważne (dla ostatnich 3% jest mniej ważne).

Cena jest nadal jednym z najważniejszych czynników wpływających na decyzje zakupowe. Jesteśmy jednak pewni, że zaufanie, reputacja i głos klientów są nie mniej ważne. Klienci są gotowi zapłacić znacznie więcej (10-30%) za produkty od dobrze ocenionych sprzedawców o nienagannej (lub prawie nienagannej) reputacji.

Dane demograficzne

Poniżej zamieszczono dane demograficzne respondentów.

- Płeć: 99 mężczyzn, 69 kobiet.
- Wykształcenie: 73 średnie, 95 wyższe.
- Przedział wiekowy: liczba przedstawicieli; <18: 2; 18-25: 92; 26-35: 58; 36-45: 13; 56-65: 2; >65: 1.
- Miejsce zamieszkania: liczba przedstawicieli; miejscowość: 18; małe miasteczko: 10; miasto 10-100 tys.: 29; miasto 100-500 tys.: 36; miasto 500-1000 tys.: 50; miasto > 1 milion: 26.

7.2 Definicja formalna – model matematyczny

Problem optymalizacji zakupów internetowych pojawia się, gdy klient chce kupić pewną liczbę produktów w sklepach internetowych (patrz rozdział 4).

Zdefiniowano nowy, bardziej wyrafinowany model, który dokładniej oddaje rzeczywiste sytuacje zakupowe związane z ważnym czynnikiem zaufania. Pro-

blem optymalizacji Trusted Internet Shopping Optimization Problem (T-ISOP) przedstawiony jest w następujący sposób:

$$\min \sum_{i=1}^m \sum_{j=1}^n x_{ij} \frac{p_{ij}}{OPay_j} v_j + \sum_{j=1}^n y_j d_j \quad (7.1)$$

$$\text{p.o.} \quad \sum_{j=1}^n x_{ij} = 1 \quad \forall i \in M, \quad (7.2)$$

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = m, \quad (7.3)$$

$$m * y_j - \sum_{i=1}^m x_{ij} \geq 0 \quad \forall j \in N, \quad (7.4)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in M, \forall j \in N, \quad (7.5)$$

$$y_j \in \{0, 1\} \quad \forall j \in N, \quad (7.6)$$

$$OPay_j = (1, \dots, maxOp) \quad \forall j \in N, \quad (7.7)$$

$$v_j \in \{1, \infty\} \quad \forall j \in N. \quad (7.8)$$

gdzie wszystkie zmienne i użyte we wzorach symbole zostały zestawione w tabeli 7.1.

Tabela 7.1. Notacja dla problemu T-ISOP

Oznaczenie	Opis
m	liczba produktów
n	liczba sklepów
i	wskaźnik produktu
j	wskaźnik sklepu
d_j	łączne koszty wysyłki dla sklepu j
y_j	wskaźnik użycia wysyłki ze sklepu j
p_{ij}	cena produktu i w sklepie j
x_{ij}	wskaźnik wyboru produktu i w sklepie j
$OPay_j$	funkcja nadpłaty dla sklepu j
v_j	wskaźnik veto dla sklepu j

Po przeprowadzeniu analizy literatury, ankiety i zebraniu własnych doświadczeń, powyższy ogólny model można zaprezentować w następujący sposób. Wprowadzono dwie nowe funkcje: nadpłata ($OPay$) i veto (v). Należy zwrócić uwagę, że powyższe równania optymalizują proces wyboru sklepu dokonując zmian w cenach produktów, niezbędnych z punktu widzenia obliczeń. Kiedy proces się

zakończy i zostanie wyznaczone rozwiązanie (przydział produktów do sklepów), klient zostanie obciążony opłatą za ceny podstawowe, w przypadku których czynnik zaufania nie ma znaczenia – $\sum_{i=1}^m \sum_{j=1}^n x_{ij}p_{ij} + \sum_{j=1}^n y_j d_j$. Zarówno funkcja $OPay$, jak i współczynnik v mogą być konstruowane zgodnie z konkretnymi wymaganiami określonymi dla danego typu odbiorcy. Funkcja $OPay$ reprezentuje połączone czynniki zaufania i definiuje ich wpływ na cenę. Przyjmuje wartości od 1 do $maxOp$ i odzwierciedla skumulowaną wartość nadpłaconej kwoty – ile klient może dodatkowo zapłacić ($Opay * 100\%$) za zakup produktów w sklepie o bardzo dobrej reputacji. Teoretycznie $maxOp \rightarrow \infty$. Jest to jednak oczywiście zupełnie nierealna sytuacja. Z analizy przeprowadzonej ankiety wynika, że wartość $maxOp$ dla całej populacji powinna wynosić 1,4 (t.j. 40% nadpłaty – i wartość ta była akceptowalna tylko dla jednej osoby). Funkcję $OPay$ należy jednak opisać bardziej szczegółowo. Funkcja $OPay$ może być zamodelowana jako:

$$OPay_j = overprice_j + Others_j \quad \forall j \in N \quad (7.9)$$

$$Others_j = dTime_{ij} * isL_j * dur_j * payR_j * dOpt_j \quad \forall i \in M, \forall j \in N \quad (7.10)$$

gdzie dla każdego sklepu j obliczany jest współczynnik $overprice$, aby przedstawić, jaka jest wartość nadpłacenia dedykowana sklepowi. $dTime$ to czas dostawy, isL odpowiada na pytanie, czy sprzedający jest sklepem lokalnym (ten sam kraj), dur to staż działania na rynku, $payR$ opisuje zakres opcji płatności, a $dOpt$ to zakres opcji dostawy. W przyszłości funkcje $OPay$ i $veto$ będą spersonalizowane dla każdego użytkownika. Proces personalizacji może być realizowany jako dodatkowe narzędzie decyzyjne udostępnione dla klientów. Do ogólnych celów przedstawienia problemu (zwłaszcza, że tematyka i całe zagadnienie problemu są bardzo innowacyjne) można wykorzystać dane zebrane z ankiety i literatury w celu przygotowania początkowych wartości dla prezentowanego modelu. Biorąc pod uwagę, że najważniejszymi czynnikami są wskaźniki oceny sklepów i ceny, czynnik $overprice$ zostanie przygotowany w następujący sposób. Należy stworzyć zbiór par wartości ratingowych i nadpłaty. Wszystkie pary wartości należy obliczyć na podstawie zebranych informacji. Ponadto, obliczona zostanie wartość średnia i odchylenie standardowe. Dla normalnej dystrybucji, wartości większe niż dwa odchylenia standardowe od średniej ($\mu - 2\sigma$, $\mu + 2\sigma$) zostaną usunięte jako wartości progowe i nieistotne. Do finalnej analizy pozostawiając 95,45% populacji (par), które zostaną włączone do modelu i będą wykorzystane zgodnie z ich obliczonym rozkładem.

Czynnik $veto$ zwraca tylko dwie wartości: negatywną (bez veto), która wynosi 1 i pozytywną (veto), która wynosi ∞ . W przypadku pozytywnego wyniku oferta nie powinna być brana pod uwagę, ponieważ klienci mają poważne zastrzeżenia do tego sprzedawcy. Należy pamiętać, że czynnik v może być spersonalizowany zgodnie z wymaganiami klienta i może być zamodelowany jako:

$$v_j = rat_j * del_j * neg_j * dForm_j \quad \forall j \in N \quad (7.11)$$

$$rat_j, del_j, neg_j, dForm_j \in \{1, \infty\} \quad \forall j \in N \quad (7.12)$$

gdzie *rat* jest ogólnym wskaźnikiem oceny sprzedawcy, *del* dotyczy kosztu wysyłki, *neg* określa wpływ negatywnych opinii, a *dFrom* oznacza wachlarz dostępnych form wysyłki.

Problem 1 *Problem ISOP [13] można łatwo przetransformować do problemu T-ISOP. Tym samym można przeprowadzić dowód zawarty w sekcji 4.2 przedstawiający transformację pseudo-wielomianową ze znanego powszechnie silnie NP-zupełnego problemu EXACT COVER BY 3-SETS (X3C) [33].*

Twierdzenie 7 *Problem T-ISOP jest silnie NP-trudny.*

Dowód 9 *Założmy sytuację, w której obie funkcje związane z zaufaniem ($OPay$) i (v) wynoszą 1 dla każdego sklepu j ($\forall_{j=1}^n OPay_j = 1, \forall_{j=1}^n v_j = 1$). Istnieje zatem prosta bezpośrednia transformacja z problemu ISOP do problemu T-ISOP. Tym samym można przeprowadzić taki sam dowód jak w twierdzeniu 1. Tak więc problem T-ISOP jest silnie NP-trudny, a dodatkowo nie istnieje wielomianowe ($c \cdot \ln n$)-aproxymacyjne rozwiązanie dla problemu T-ISOP, chyba że $\mathcal{P} = \mathcal{NP}$.*

Wyniki przeprowadzonego badania ankietowego wykazały, że w większości przypadków użytkownicy cenią sobie, poza finalnym niskim kosztem, większą renomę. Obok ocen i opinii klientów zaufanie opiera się na solidnych zabezpieczeniach metod płatności, spójności jakości produktów prezentowanych w sklepie internetowym z tymi otrzymanymi przez użytkownika, oraz efektywnych możliwościach dostawy.

Wcześniej opisaną sytuację można zauważyć, gdy sklepy o niższej renomie starają się przyciągać nowych klientów poprzez redukcję kosztów. W przeciwieństwie do bardziej renomowanych i zaufanych sprzedawców, którzy mogą sobie pozwolić na utrzymanie lub nawet podniesienie cen oferowanych produktów. W przypadku podobnych ofert, sklep o niższej renomie mógłby zostać zawetowany lub wybrane zostaną sklepy posiadające wyższe wskaźniki ocen.

7.3 Algorytmy

Początkowym podejściem, które można zastosować do rozwiązania problemu T-ISOP, uwzględniającego czynnik zaufania i opinie klientów, może być użycie algorytmu genetycznego (GA). Metoda zainspirowana jest naturalnym procesem ewolucji w wyniku selekcji naturalnej opisanej przez Charlesa Darwina w 1858 roku. Obecna postać GA została sformalizowana przez Johna Hollanda w latach 70 [41]. Aktualny zapis algorytmu genetycznego wykorzystywanego do celów niniejszych badań jest zobrazowany poprzez algorytm 12.

GA rozpoczyna działanie od wygenerowania *populacji* przypadkowych losowych rozwiązań, a następnie rozpoczyna wykonywanie pętli, która kończy się, gdy warunek zakończenia jest spełniony – tutaj maksymalna liczba wygenerowanych pokoleń. Wewnątrz pętli, dwa osobniki *p1* i *p2* z *populacji* są wybierane

metodą *selekcji*. Wybrane osobniki biorą udział w etapie tworzenia nowego pokolenia poprzez *krzyżowanie* – powstają *potomkowie*. Nowo powstałe osobniki są poddawane *mutacji*, a następnie wyznaczona zostaje wartość *funkcji przystosowania*. Jeśli dla danego *potomka* wartość *funkcji przystosowania* jest lepsza niż dla najgorszego rozwiązania w *populacji*, to dane potomstwo je zastępuje (strategia z częściową reprodukcją, elityzm). Po spełnieniu warunku zakończenia, najlepsze rozwiązanie z populacji jest zwracane, a GA kończy swoje działanie.

Algorytm 12 Genetic Algorithm (GA)

```

1: Inicjalizacja population o rozmiarze populationSize rozwiązań losowych
2: while currentGeneration < maxGenerations do
3:   Wybierz p1 and p2 z population poprzez selection()
4:   Uzyskaj offspring poprzez zastosowanie crossover() do p1 i p2
5:   Zastosuj mutation() na offspring
6:   Oceń fitness() dla offspring
7:   if offspring jest lepszy niż worst then
8:     Usuń worst z population
9:     Dodaj offspring do population
10:  end if
11: end while
12: return best z population

```

7.4 Eksperymenty obliczeniowe

Dane eksperymentalne składają się z dwóch zestawów:

- 10 produktów i 20 sklepów 10n20m (10_20_1125, 10_20_1343, 10_20_1729, 10_20_2094, 10_20_3596, 10_20_4069, 10_20_4186, 10_20_4590, 10_20_5339, 10_20_5721),
- 20 produktów i 20 sklepów 20n20m (20_20_1085, 20_20_1207, 20_20_1413, 20_20_1535, 20_20_1817, 20_20_2321, 20_20_2329, 20_20_3361, 20_20_3869, 20_20_4207).

Ostatnia wartość przy każdej nazwie instancji została użyta jako ziarno do wygenerowania losowo rozłożonych ocen liczbowych dla każdego sklepu, z średnią wartością 7 i odchyleniem standardowym 2. Odpowiada to wynikom wcześniej przeprowadzonych analiz.

Do eksperymentów zastosowano trzy wartości veta (3,6,9). Sklepy o wartości wyższej lub równej wartości veta są uwzględniane przy zakupie produktów znajdujących się na liście zakupów. Wartość veta wynosząca 0 została wykorzystana

jako kontrolna, w ramach której uwzględniono wszystkie sklepy, niezależnie od ich wskaźników oceny.

Funkcja nadpłaty T została zdefiniowana za pomocą następujących wartości:

$$T(x) = \begin{cases} \text{prodCost}(x) * 1.0, & \text{jeżeli } 0 \leq \text{rating}(x) \leq 1; \\ \text{prodCost}(x) * 1.05, & \text{jeżeli } 1 < \text{rating}(x) \leq 4; \\ \text{prodCost}(x) * 1.1, & \text{jeżeli } 4 < \text{rating}(x) \leq 7; \\ \text{prodCost}(x) * 1.2, & \text{jeżeli } 7 < \text{rating}(x) \leq 9; \\ \text{prodCost}(x) * 1.3, & \text{jeżeli } 9 < \text{rating}(x) \leq 10. \end{cases}$$

gdzie x reprezentuje sklep, $\text{prodCost}(x)$ produkty z listy zakupów, które zostały zakupione w sklepie x i $\text{rating}(x)$ jako wskaźnik oceny sklepu x .

Populacja GA została ustalona na 50 osobników, z maksymalną liczbą 25 000 pokoleń. Operatorem *selekcji* był Binary Tournament Selection [67]. Zastosowanym operatorem *krzyżowania* był Integer Simulated Binary Crossover (SBX [8, 24]) z prawdopodobieństwem wymiany 0,9. Wykorzystanym operatorem *mutacji* była mutacja wielomianowa (Integer Polynomial Mutation [79]) o prawdopodobieństwie mutacji $p_m = 0,05$. Nie było limitu czasowego na rozwiązanie instancji w eksperymencie obliczeniowym.

W tabeli 7.2 zaprezentowano wyniki eksperymentu obliczeniowego. W pierwszej kolumnie podano różne, brane pod uwagę wartości veta. Dla każdego zestawu wartości veta obliczono skumulowany czas w milisekundach, który wykorzystał GA na rozwiązanie wszystkich instancji w zestawie. Ponadto, zaznaczono średnią ocenę sklepów wybranych do ostatecznych rozwiązań, oraz średnią obiektywną wartość (OW) lub całkowity koszt listy zakupów każdego zestawu.

Można zauważyć, że łączny koszt listy zakupów wzrasta wraz ze średnią oceną wybranych sklepów. Wskazuje to, że T-ISOP można w przyszłości modelować jako wieloobiektowy problem optymalizacyjny (dwie funkcje celu). Eksperyment powinien być postrzegany jako ważny pierwszy krok otwierający szerokie możliwości dla przyszłych badań. Rzeczywiste dane zgromadzone w procesie dogłębnej analizy rynku (niezmiernie trudne zadanie) mogą być wykorzystane do udoskonalenia projektu algorytmu, procesu optymalizacji i wykorzystania bardziej szczegółowego modelu T-ISOP.

Tabela 7.2. Wyniki eksperymentu obliczeniowego

		10n20m	20n20m
veto = 9	<i>Czas(ms)</i>	2279	2613
	<i>Średnia ocena</i>	9.49	9.42
	<i>OW</i>	133.73	330.49
veto = 6	<i>Czas(ms)</i>	2085	2493
	<i>Średnia ocena</i>	7.65	7.71
	<i>OW</i>	100.51	155.18
veto = 3	<i>Czas(ms)</i>	2049	2567
	<i>Średnia ocena</i>	5.84	6.72
	<i>OW</i>	83.43	157.89
veto = 0	<i>Czas(ms)</i>	2209	2594
	<i>Średnia ocena</i>	6.35	6.76
	<i>OW</i>	87.59	172.58

7.5 Podsumowanie

W niniejszym rozdziale przeanalizowano rozszerzenie ISOP formułując problem optymalizacji Trusted Internet Shopping Optimization Problem (T-ISOP). Czynniki zaufania zostały dokładnie przeanalizowane z punktu widzenia klienta. Analiza opiera się na specjalnie przygotowanej ankiecie, uzupełnionej o informacje z literatury oraz własne spostrzeżenia. W ten sposób możliwe było zaproponowanie definicji nowego modelu matematycznego problemu i udowodnienie jego przynależności do klasy problemów silnie NP-trudnych. W dalszej części rozdziału przedstawiono wiele interesujących spostrzeżeń i dyskusji dotyczących zarówno natury problemu, jak i jego możliwych rozwiązań.

Zaproponowany został algorytm, który może być wykorzystany do skutecznego rozwiązania problemu. Należy podkreślić, że jest to dopiero pierwszy krok i pierwsza próba zdefiniowania problemu T-ISOP. Przyszłe prace będą polegały na dogłębnej analizie czynnika zaufania i połączeniu go z optymalizacją problemu (także z bardziej technicznego punktu widzenia [62]). Interesującym pomysłem badawczym na przyszłość jest przygotowanie eksperymentów obliczeniowych na podstawie rzeczywistych danych. Ponadto naturalne i inspirujące wydaje się wzbogacenie problemu Cloud Brokeringu [37] o analizę czynnika zaufania. Cloud Brokering jest niezwykle interesującym zagadnieniem i zgodnie z naszą najlepszą wiedzą cierpi z powodu braku analizy zaufania i wpływu czynników reputacji na rynek. Są to głosy pojawiające się zarówno wśród naukowców, jak i wielu firm działających na rynku e-commerce w zakresie Cloud Brokeringu.

Rozdział 8

Podsumowanie i perspektywa dalszych prac

Rozwój obecnie znanej sieci Internet rozpoczął się w latach 60-tych ubiegłego wieku. Po drodze przeszedł wiele zmian i ewoluował, aby osiągnąć znaną przez nas, dzisiejszą formę. Znacznie trudniej jest obecnie powiedzieć, czego nie znajdziemy w Internecie. Ilość informacji krążących w różnych zakątkach sieci jest wręcz przytłaczająca i coraz trudniejsza do przeanalizowania. Internet jest nie tylko źródłem informacji, ale praktycznie sposobem na życie. Można dzięki niemu pracować, uczyć się, sprawnie komunikować się (również wideo) z odległymi krajami, oglądać filmy, słuchać muzyki, grać w gry, robić zakupy, itd. Rozwój rynku e-commerce postępuje również od wielu lat. Na dzień dzisiejszy, wśród niektórych asortymentów produktów, sklepy internetowe zaczynają wypierać tradycyjny model handlu. Łatwość dotarcia do znacznie szerszego grona odbiorców, czy pozyskania wielu ofert na ten sam produkt stymuluje wzrost rynku handlu elektronicznego. Obserwowany w ostatnich latach bardzo dynamiczny rozwój e-commerce powoduje pojawianie się przytłaczającej dla klienta ilości informacji handlowych. Pojawiają się zatem rozwiązania mające wspomagać zakupy internetowe. Wśród istniejących, warto na pewno wymienić porównywarki cen, czy systemy rekomendujące. Szeroka oferta możliwych form dostawy, czy płatności jeszcze bardziej zachęca do zakupów online. Niestety, według mojej najlepszej wiedzy, do dziś nie powstało sprawnie działające rozwiązanie, które będzie w stanie dokonać analizy kupowanych przez nas wielu produktów. Takiej, aby możliwy był zakup przy najniższej możliwej cenie i mógł spełniać ponadto szereg wymagań i dodatkowych kryteriów.

Problem optymalizacji zakupów internetowych obrazuje sytuację, gdzie klient ma zamiar zakupić kilka produktów ze sklepów internetowych. W najprostszej formie ma uzyskać odpowiedź gdzie należy dokonać zakupów (może to być wiele

sklepów, bądź jeden), aby zapłacić najniższą możliwą kwotę. Kolejne wersje / rozwinięcia problemu opisują dodatkowe atrybuty i przedstawiają coraz bardziej skomplikowane (aczkolwiek, co należy podkreślić coraz bliższe wymaganiom klienta) wersje problemu ISOP. Dodatkowe kwestie mogą dotyczyć np. zmiennych cen wysyłki, rabatów udzielanych przez sklepy, czynników zaufania, możliwości ulubionej formy wysyłki, czy płatności. Mogą również obejmować wymagania na produkty krótkotrwałe, czy ograniczony budżet kupującego.

W niniejszej książce skrótkowo przedstawiono obecną sytuację / kondycję rynku e-commerce, jako wprowadzenie do głównego tematu. Jest nim prezentacja różnych (wszystkich jego najważniejszych odmian) wariantów problemu ISOP. Przytoczono modele matematyczne, opisy, prezentacje autorskich algorytmów i wyniki eksperymentów obliczeniowych.

Warto podkreślić, że ze względu na aplikowalny charakter badań tematyka znajduje spore zainteresowanie wśród środowisk naukowych czy biznesowych. Przykładem może być fakt uzyskania grantu naukowo-badawczego na międzynarodowy projekt Internet Shopping Optimization Project¹ (współfinansowanego przez NCBR² i FNR³), który był z wielkim sukcesem realizowany w latach 2014-2017. Uzyskane wyniki znacznie przekroczyły wymagania postawione zespołowi realizującemu zadania projektowe. Autor niniejszej publikacji był jednym z głównych naukowców biorących udział w badaniach. Obecnie trwają rozmowy dotyczące możliwych wdrożeń tematyki ISOP.

Problem ISOP cały czas ewoluuje. Istnieje wiele sytuacji rynkowych, które nadal warto zamodelować. Poza odmianami problemu ściśle związanymi z tematyką problemu ISOP, innym interesującym punktem badawczym mogłoby być proste narzędzie symulacji wspomagające podejmowanie decyzji. Ponadto korzystne mogłoby być zastosowanie podejść inspirowanych biologią, wykorzystanych w tym zakresie modeli [83] i wyników obliczeń [97, 4]. Ponadto, wiele funkcji dodatkowych (jak np. te dotyczące czynników zaufania) mogą być spersonalizowane dla każdego użytkownika. Proces personalizacji może być realizowany jako pomocnicze narzędzie decyzyjne dla kupujących. Pozwoli to na przygotowanie analizy eksperymentalnej dedykowanej każdemu użytkownikowi zgodnie z jego szczegółowymi potrzebami i wymaganiami. Wszystkie wspomniane kroki wymagają jednak ogromnej ilości nadchodzącej, ekscytującej pracy.

Zdecydowanie należy też wspomnieć o szerokiej tematyce Cloud Brokreingu [37]. Problemy pośredników uczestniczących w handlu usług chmurowych są w wielu sytuacjach zbieżne z problemami opisanymi w niniejszym wydawnictwie. Z pewnością warto zgłębić tematykę Cloud Brokeringu z punktu widzenia problemów ISOP i przedsiębiorstw uczestniczących w szeroko pojętym handlu internetowym. Może to dać pole do nowych, bardzo ciekawych badań naukowych, jak i szeroko zakrojonej współpracy na linii nauka-biznes.

¹<http://www.cs.put.poznan.pl/ishop/>

²Narodowe Centrum Badań i Rozwoju – <http://www.ncbr.gov.pl>

³Fonds National de la Recherche – <https://www.fnr.lu>

Bibliografia

- [1] E. Aarts, J. K. Lenstra. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, 1997.
- [2] O. A. al-Qudah, K. Ahmad. The roles of culture in online shopping to enhance e-commerce in Jordan. In *2014 International Conference on Information Technology and Multimedia (ICIMU)*, s. 113–117. IEEE, 2014.
- [3] R. K. Ahuja, A. Kumar, K. C. Jha, J. B. Orlin. Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem. *Operations Research*, 55(6), s. 1136–1146, 2007.
- [4] M. Antczak, M. Kasprzak, P. Lukasiak, J. Blazewicz. Structural alignment of protein descriptors – a combinatorial model. *BMC Bioinformatics*, 17(1), s. 383, 2016.
- [5] M. S. M. Ariff, M. Sylvester, N. Zakuan, K. Ismail, K. M. Ali. Consumer Perceived Risk, Attitude and Online Shopping Behaviour; Empirical Evidence from Malaysia. In *IOP Conference Series: Materials Science and Engineering*, volume 58, s. 012007. IOP Publishing, 2014.
- [6] J. Błażewicz. *Złożoność obliczeniowa problemów kombinatorycznych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1988.
- [7] P. Baran. *On Distributed Communications*. The Rand Corporation, Memorandum RM-3420-PR, 1964.
- [8] R. B. Agrawal, K. Deb. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9(2), s. 115–148, 1995.
- [9] J. Blazewicz, P. Bouvry, M. Y. Kovalyov, J. Musial. Erratum to: Internet shopping with price-sensitive discounts. *4OR*, 12(4), s. 403–406, 2014.
- [10] J. Blazewicz, P. Bouvry, M. Y. Kovalyov, J. Musial. Internet shopping with price sensitive discounts. *4OR*, 12(1), s. 35–48, 2014.

- [11] J. Blazewicz, N. Cheriére, P.-F. Dutot, J. Musiał, D. Trystram. Novel dual discounting functions for the Internet shopping optimization problem: new algorithms. *Journal of Scheduling*, 19(3), s. 245–255, 2016.
- [12] J. Blazewicz, K. Ecker, E. Pesch, G. Schmidt, J. Weglarz. *Handbook on Scheduling - From Theory to Applications*. Springer-Verlag Berlin Heidelberg, 2007.
- [13] J. Blazewicz, M. Kovalyov, J. Musiał, A. Urbanski, A. Wojciechowski. Internet shopping optimization problem. *International Journal of Applied Mathematics and Computer Science*, 20(2), s. 385–390, 2010.
- [14] J. Błażewicz, J. Musiał. E-Commerce Evaluation – Multi-Item Internet Shopping. Optimization and Heuristic Algorithms. In B. Hu, K. Morasch, S. Pickl, M. Siegle (eds.), *Operations Research Proceedings 2010*, s. 149–154, Springer Berlin Heidelberg, 2011.
- [15] T. Braun, H. Siegel, N. Beck, L. Bölöni, M. Maheswaran, A. Reuther, J. Robertson, M. Theys, B. Yao, D. Hensgen, et al. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, 61(6), s. 810–837, 2001.
- [16] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, S. Schulenburg. Hyper-Heuristics: An Emerging Direction in Modern Search Technology. In F. Glover and G. A. Kochenberger (eds.), *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, s. 457–474. Springer, Boston, MA, 2003.
- [17] R. Burke. Hybrid Web Recommender Systems. In P. Brusilovsky, A. Kobsa, W. Nejdl (eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, s. 377–408. Springer Berlin Heidelberg, 2007.
- [18] W. Chu, B. Choi, M. Song. The Role of On-line Retailer Brand and Infomediary Reputation in Increasing Consumer Purchase Intention. *International Journal of Electronic Commerce*, 9(3), s. 115–127, 2005.
- [19] K. Clay, R. Krishnan, E. Wolff. Prices and Price Dispersion on the Web: Evidence from the Online Book Industry. NBER Working Papers 8271, National Bureau of Economic Research, Inc, 2001.
- [20] comScore. UPS Pulse of the Online Shopper, A customer experience study, 2014. <http://www.ups.com/media/en/2014-UPS-Pulse-of-the-Online-Shopper.pdf>.
- [21] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, 1998.

- [22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.
- [23] K. Crescenzi, V. Kann. A compendium of NP optimization problems, 2005. <http://www.nada.kth.se/viggo/wwwcompendium/>
- [24] K. Deb, H.-G. Beyer. Self-Adaptive Genetic Algorithms with Simulated Binary Crossover. *Evol. Comput.*, 9(2), s. 197–221, 2001.
- [25] M. H. Dehnavi, A. Shirazi, M. Zhourian. The Influence of Cultural variables on Online Shopping (A survey of cultural effects on online users' behavior in Iran). *Asian Journal of Research in Marketing*, 3(2), s. 168–183, 2014.
- [26] C. O. Diaz, J. E. Pecero, P. Bouvry. Scalable, low complexity, and fast greedy scheduling heuristics for highly heterogeneous distributed computing systems. *The Journal of Supercomputing*, 67(3), s. 837–853, 2014.
- [27] Ecommerce Foundation. European Ecommerce Report 2017, 2017.
- [28] H. Eiselt, C. Sandblom. *Decision Analysis, Location Models, and Scheduling Problems*. Springer-Verlag Berlin Heidelberg, 2004.
- [29] P. Ezzatti, M. Pedemonte, A. Martin. An efficient implementation of the Min-Min heuristic. *Computers & Operations Research*, 40(11), s. 2670–2676, 2013.
- [30] A. J. Flanagan, M. J. Metzger, R. Pure, A. Markov, E. Hartsell. Mitigating risk in ecommerce transactions: perceptions of information credibility and the role of user-generated ratings in product quality and purchase intention. *Electronic Commerce Research*, 14(1), s. 1–23, 2014.
- [31] L. Fretwell, J. Stine, H. Sethi, A. Noronha. 'Catch and Keep' Digital Shoppers How To Deliver Retail Their Way, 2013.
- [32] R. Freund, M. Gherrity, S. Ambrosius, M. Campbell, M. Halderman, D. Hensgen, E. Keith, T. Kidd, M. Kussow, J. Lima, F. Mirabile, L. Moore, B. Rust, and H. Siegel. Scheduling resources in multi-user, heterogeneous, computing environments with SmartNet. In *Proceedings of the Seventh Heterogeneous Computing Workshop*, s. 184–199, 1998.
- [33] M. Garey, D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, 1979.
- [34] Gemius dla e-Commerce Polska. E-commerce w Polsce. Technical report, Gemius, Izba Gospodarki Elektronicznej, 2017.

- [35] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- [36] D. R. Goossens, A. J. T. Maas. Exact algorithms for procurement problems under a total quantity discount structure. *European Journal of Operational Research*, 178(2), s. 603–626, 2007.
- [37] M. Guzek, A. Gniewek, P. Bouvry, J. Musiał, J. Blazewicz. Cloud Brokering: Current Practices and Upcoming Challenges. *IEEE Cloud Computing*, 2(2), s. 40–47, 2015.
- [38] K. Hafner, M. Lyon. *Where Wizards Stay Up Late: The Origins of the Internet*. Simon & Schuster, 1st edition, 1996.
- [39] J. Hagel. Net gain: Expanding markets through virtual communities. *Journal of Interactive Marketing*, 13(1), s. 55–65, 1999.
- [40] R. Hof. More ways to price-shop online. *BusinessWeek*, 3851:14, 2003.
- [41] J. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, 1975.
- [42] K. Holmberg. Solving the staircase cost facility location problem with decomposition and piecewise linearization. *European Journal of Operational Research*, 75(1), s. 41–61, 1994.
- [43] K. Holmberg, J. Ling. A Lagrangean heuristic for the facility location problem with staircase costs. *European Journal of Operational Research*, 97(1), s. 63–74, 1997.
- [44] P. A. Hosein, M. Athans. Some analytical results for the dynamic weapon-target allocation problem. *MIT, Laboratory for Information and Decision Systems*, s. 1–28, 1990.
- [45] M.-H. Hsu, L.-W. Chuang, C.-S. Hsu. Understanding online shopping intention: the roles of four types of trust and their antecedents. *Internet Research*, 24(3), s. 332–352, 2014.
- [46] internet live stats. Internet Users, 2016. odwiedzono: lipiec 2017, <http://www.internetlivestats.com/internet-users/>
- [47] Internet World Stats. Internet Usage and World Population Statistics, 2017. odwiedzono: lipiec 2018, <http://www.internetworldstats.com/stats.htm>
- [48] D. Jannach. Finding Preferred Query Relaxations in Content-based Recommenders. *2006 3rd International IEEE Conference Intelligent Systems*, s. 355–360, 2006.

- [49] S. Klein. The Emergence of Auctions on the World Wide Web. In M. Shaw, R. Blanning, T. Strader, A. Whinston (eds.), *Handbook on Electronic Commerce*, s. 627–645. Springer Berlin Heidelberg, 2000.
- [50] J. Krarup, D. Pisinger, F. Plastriab. Discrete location problems with push-pull objectives. *Discrete Applied Mathematics*, 123(1-3), s. 363–378, 2002.
- [51] S. Krichen, A. Laabidi, F. B. Abdelaziz. Single supplier multiple cooperative retailers inventory model with quantity discount and permissible delay in payments. *Computers & Industrial Engineering*, 60(1), s. 164–172, 2011.
- [52] S. Kumar, S. Maan. Status and Scope of Online Shopping: An Interactive Analysis through Literature Review. *International Journal of Advance Research in Computer Science and Management Studies*, 2(12), s. 100–108, 2014.
- [53] A. Land, A. Doig. An Automatic Method for Solving Discrete Programming Problems. *Econometrica*, 28(3), s. 497–520, 1960.
- [54] H.-G. Lee. Do Electronic Marketplaces Lower the Price of Goods? *Commun. ACM*, 41(1), s. 73–80, 1998.
- [55] T.-P. Liang, J.-S. Huang. An empirical study on consumer acceptance of products in electronic markets: a transaction cost model. *Decision Support Systems*, 24(1), s. 29–43, 1998.
- [56] J. C. R. Licklider. Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1:4–11, 1960.
- [57] M. Lopez-Loces, J. Musial, J. Pecero, H. Fraire-Huacuja, J. Blazewicz, P. Bouvry. Exact and heuristic approaches to solve the Internet shopping optimization problem with delivery costs. *International Journal of Applied Mathematics and Computer Science*, 26(2), s. 391–406, 2016.
- [58] P. Maes, R. H. Guttman, A. G. Moukas. Agents That Buy and Sell. *Commun. ACM*, 42(3), s. 81–ff., 1999.
- [59] B. Mahadevan. Business Models for Internet-Based E-Commerce: An Anatomy. *California Management Review*, 42(4), s. 55–69, 2000.
- [60] T. Mahmood, F. Ricci. Improving Recommender Systems with Adaptive Conversational Strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, s. 73–82, ACM New York, NY, 2009.
- [61] D. Manerba, R. Mansini. An exact algorithm for the Capacitated Total Quantity Discount Problem. *European Journal of Operational Research*, 222(2), s. 287–300, 2012.

- [62] J. Marszałkowski, J. M. Marszałkowski, J. Musiał. Database Scheme Optimization For Online Applications. *Foundations of Computing and Decision Sciences*, 36(2), s. 121–129, 2011.
- [63] J. Marszałkowski, J. M. Marszałkowski, M. Drozdowski. Empirical Study of Load Time Factor in Search Engine Ranking. *Journal of Web Engineering*, 13(1&2), s. 114–128, 2014.
- [64] S. Martello, P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, 1990.
- [65] F. McSherry, I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, s. 627–636, ACM New York, NY, 2009.
- [66] M. Melo, S. Nickel, F. Saldanha-da-Gama. Facility location and supply chain management – A review. *European Journal of Operational Research*, 196(2), s. 401–412, 2009.
- [67] B. L. Miller, D. E. Goldberg. Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems*, 9(3), s. 193–212, 1995.
- [68] S. H. Mirmohammadi, S. Shadrokh, F. Kianfar. An efficient optimal algorithm for the quantity discount problem in material requirement planning. *Computers & Operations Research*, 36(6), s. 1780–1788, 2009.
- [69] D. D. Monica, M. Napoli, M. Parente. On a Logic for Coalitional Games with Priced-Resource Agents. *Electronic Notes in Theoretical Computer Science*, 278:215–228, 2011.
- [70] C. Munson, J. Hu. Incorporating quantity discounts and their inventory impacts into the centralized purchasing decision. *European Journal of Operational Research*, 201(2), s. 581–592, 2010.
- [71] J. Musiał. *Applications of Combinatorial Optimization for Online Shopping*, volume 4 of *Poznan Monographs in Computing and Its Applications*. NAKOM, Poznan, 2012.
- [72] J. Musiał, M. C. Lopez-Loces. Trustworthy Online Shopping with Price Impact. *Foundations of Computing and Decision Sciences*, 42(2), s. 121–136, 2017.
- [73] J. Musiał, J. Pecero, M. Lopez-Loces, H. Fraire-Huacuja, P. Bouvry, and J. Blazewicz. Algorithms solving the Internet shopping optimization problem with price discounts. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 64(3), s. 505–516, 2016.

- [74] J. Musial, J. E. Pecero, M. C. Lopez, H. J. Fraire, P. Bouvry, and J. Blazewicz. How to efficiently solve Internet Shopping Optimization Problem with price sensitive discounts? In *2014 11th International Conference on e-Business (ICE-B)*, s. 209–215, 2014.
- [75] G. L. Nemhauser, L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, 1999.
- [76] S. Nesmachnow, B. Dorronsoro, J. E. Pecero, P. Bouvry. Energy-aware scheduling on multicore heterogeneous grid computing systems. *Journal of grid computing*, 11(4), s. 653–680, 2013.
- [77] Nielsen Company. Global Online Shopping Report. Technical report, Nielsen Company, 2010. <http://www.nielsen.com/us/en/insights/news/2010/global-online-shopping-report.html>
- [78] T. U. of Reading Statistical Services Centre | Biometrics Advisory and S. S. to DFID. Approaches to the Analysis of Survey Data. Technical report, The University of Reading Statistical Services Centre | Biometrics Advisory and Support Service to DFID, 2001. <http://archive.acehresearch.org/items/show/69>
- [79] G. Onwubolu, B. Babu. *New Optimization Techniques in Engineering*. Studies in Fuzziness and Soft Computing. Springer Berlin Heidelberg, 2013.
- [80] X. Pan, B. T. Ratchford, V. Shankar. The Evolution of Price Dispersion in Internet Retail Markets. *Advances in Applied Microeconomics*, 12:85–105, 2003.
- [81] B. Pathak. Comparison Shopping Agents and Online Price Dispersion: A Search Cost based Explanation. *J. Theor. Appl. Electron. Commer. Res.*, 7(1), s. 64–76, 2012.
- [82] F. Pinel, B. Dorronsoro, P. Bouvry. Solving very large instances of the scheduling of independent tasks problem on the GPU. *Journal of Parallel and Distributed Computing*, 73(1), s. 101–110, 2013.
- [83] T. Prejzendanc, S. Wasik, J. Blazewicz. Computer representations of bioinformatics models. *Current Bioinformatics*, 11(5), s. 551–560, 2016.
- [84] B. T. Ratchford, X. Pan, V. Shankar. On the Efficiency of Internet Markets for Consumer Goods. *Journal of Public Policy & Marketing*, 22(1), s. 4–16, 2003.
- [85] R. Raz, S. Safra. A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP. In *29th Annual ACM Symposium on Theory of Computing*, s. 475–484, 1997.

- [86] P. Resnick, H. R. Varian. Recommender Systems. *Commun. ACM*, 40(3), s. 56–58, 1997.
- [87] C. Revelle, H. Eiselt, M. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3), s. 817–848, 2008.
- [88] J. Reynolds, R. Mofazali. *The Complete E-Commerce Book: Design, Build, and Maintain a Successful Web-Based Business*. C M P Books, 1st edition, 2000.
- [89] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (eds.). *Recommender Systems Handbook*. Springer US, 2011.
- [90] D. Rutkowska, M. Piliński, L. Rutkowski. *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*. Wydawnictwo Naukowe PWN, 1997.
- [91] B. Schwartz. *The Paradox of Choice: Why More Is Less*. Ecco, New York, NY, 2004.
- [92] S. Shin, W.-J. Lee, D. O. Odom. A Comparative Study Of Smartphone User’s Perception And Preference Towards Mobile Payment Methods In The US And Korea. *Journal of Applied Business Research*, 30(5), s. 1365–1376, 2014.
- [93] M. Sipper. The emergence of cellular computing. *Computer*, 32(7), s. 18–26, 1999.
- [94] J. Snider, T. Ziporyn. *Future Shop: How New Technologies Will Change the Way We Shop and what We Buy*. St. Martin’s Press, 1992.
- [95] R. M. Soland. Optimal Facility Location with Concave Costs. *Operations Research*, 22(2), s. 373–382, 1974.
- [96] D. Stratila. *Combinatorial optimization problems with concave costs*. Massachusetts Institute of Technology, Sloan School of Management, Operations Research Center, 2009.
- [97] N. Szostak, S. Wasik, J. Blazewicz. Hypercycle. *PLoS Computational Biology*, 12(4), s. 1–13, 2016.
- [98] W. E. Team. The History of Online Shopping – From the 1960’s to the 1990’s, 2016. odwiedzono: kwiecień 2018, <https://1stwebdesigner.com/history-of-online-shopping/>
- [99] J. D. Terán-Villanueva, H. J. Fraire Huacuja, J. M. Carpio Valadez, R. A. Pazos Rangel, H. J. Puga Soberanes, J. A. Martínez Flores. A heterogeneous cellular processing algorithm for minimizing the power consumption

- in wireless communications systems. *Computational Optimization and Applications*, 62(3), s. 787–814, 2015.
- [100] J. D. Terán-Villanueva, H. J. Fraire Huacuja, J. M. Carpio Valadez, R. A. Pazos Rangel, H. J. Puga Soberanes, J. A. Martínez Flores. Cellular Processing Algorithms. In P. Melin, O. Castillo (eds.), *Soft Computing Applications in Optimization, Control, and Recognition*, volume 294 of *Studies in Fuzziness and Soft Computing*, s. 53–74. Springer Berlin Heidelberg, 2013.
- [101] R. Thakur, M. Srivastava. A study on the impact of consumer risk perception and innovativeness on online shopping in India. *International Journal of Retail & Distribution Management*, 43(2), s. 148–166, 2015.
- [102] P. Timmers. Business Models for Electronic Markets. *Electronic Markets*, 8(2), s. 3–8, 1998.
- [103] S. Voss, S. Martello, I. Osman, C. Roucairol (eds.). *Meta-Heuristics - Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers, Boston, MA, 1998.
- [104] W. Warayuant, A. Suyanto. The Influence of Lifestyles and Consumers Attitudes on Product Purchasing Decision via Online Shopping in Indonesia. *European Journal of Business and Management*, 7(8), s. 74–80, 2015.
- [105] A. Wojciechowski, J. Musiał. A customer assistance system: Optimizing basket cost. *Foundations of Computing and Decision Sciences*, 34(1), s. 59–69, 2009.
- [106] A. Wojciechowski, J. Musiał. Towards Optimal Multi-item Shopping Basket Management: Heuristic Approach. In R. Meersman, T. Dillon, P. Herrera (eds.), *On the Move to Meaningful Internet Systems: OTM 2010 Workshops*, volume 6428 of *Lecture Notes in Computer Science*, s. 349–357. Springer Berlin Heidelberg, 2010.
- [107] M. Wong. Evaluating Online Shopping Service Quality—SERVEGAL in Taiwan Tourism Industry. *Open Journal of Social Sciences*, 2(09), s. 213–216, 2014.
- [108] L.-Y. Wu, K.-Y. Chen, P.-Y. Chen, S.-L. Cheng. Perceived value, transaction cost, and repurchase-intention in online shopping: A relational exchange perspective. *Journal of Business Research*, 67(1), s. 2768–2776, 2014.

- [109] M. Wu, W. Shu, H. Zhang. Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems. In *Proceedings 9th Heterogeneous Computing Workshop (HCW 2000)*, s. 375–385. IEEE, 2000.
- [110] X. Xiang, L. Jiashi, G. Zhongliang, K. Xinsheng. Fresh Food Online Supermarket Development Study. *Journal of Electronic Commerce in Organizations*, 12(2), s. 14–30, 2014.
- [111] K. Z. Zhang, C. M. Cheung, M. K. Lee. Examining the Moderating Effect of Inconsistent Reviews and Its Gender Differences on Consumers' Online Shopping Decision. *Int. J. Inf. Manag.*, 34(2), s. 89–98, 2014.