UNIVERSITY OF LJUBLJANA
FACULTY OF ECONOMICS

LUKA TOMAT

ONE-DIMENSIONAL CUTTING STOCK OPTIMIZATION WITH USABLE LEFTOVER: A CASE OF LOW STOCK-TO-ORDER RATIO

DOCTORAL DISSERTATION

Ljubljana, 2013

UNIVERSITY OF LJUBLJANA
FACULTY OF ECONOMICS

LUKA TOMAT

ONE-DIMENSIONAL CUTTING STOCK OPTIMIZATION WITH USABLE LEFTOVER: A CASE OF LOW STOCK-TO-ORDER RATIO

DOCTORAL DISSERTATION

Ljubljana, 2013

AUTHORSHIP STATEMENT

The undersigned Luka Tomat, a student at the University of Ljubljana, Faculty of Economics, (hereafter: FELU), declare that I am the author of the doctoral dissertation entitled One-Dimensional Cutting Stock Optimization with Usable Leftover: A Case of Low Stock-to-Order Ratio, written under supervision of prof. dr. Mirko Gradišar.

In accordance with the Copyright and Related Rights Act (Official Gazette of the Republic of Slovenia, Nr. 21/1995 with changes and amendments) I allow the text of my doctoral dissertation to be published on the FELU website.

I further declare

- the text of my doctoral dissertation to be based on the results of my own research;
- the text of my doctoral dissertation to be language-edited and technically in adherence with the FELU's Technical Guidelines for Written Works which means that I
    - cited and / or quoted works and opinions of other authors in my doctoral dissertation in accordance with the FELU's Technical Guidelines for Written Works and
    - obtained (and referred to in my doctoral dissertation) all the necessary permits to use the works of other authors which are entirely (in written or graphical form) used in my text;
- to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Copyright and Related Rights Act (Official Gazette of the Republic of Slovenia, Nr. 55/2008 with changes and amendments);
- to be aware of the consequences a proven plagiarism charge based on the submitted doctoral dissertation could have for my status at the FELU in accordance with the relevant FELU Rules on Doctoral Dissertation.

Date of public defense:
Committee Chair: doc. dr. Peter Trkman
Supervisor: prof.dr. Mirko Gradišar
Member: prof. dr. Miroljub Kljajić
Member: prof. dr. Gerhard Wilhelm Weber

Ljubljana, 26.3.2014 Author's signature: __________________________

OPTIMIZACIJA ENO-DIMENZIONALNEGA RAZREZA Z UPORABNIM OSTANKOM: PRIMER NIZKEGA RAZMERJA MED POVPREČNO DOLŽINO NA ZALOGI IN POVPREČNO DOLŽINO NAROČIL

POVZETEK

Problem eno-dimenzionalnega razreza ima v praksi veliko pojavnih oblik. V največ primerih je definiran kot problem, pri katerem je treba v naprej znano število naročenih palic razrezati iz znane zaloge palic (različno število različnih dolžin palic). Gre za zadovoljevanje naročil kupcev, ki izhajajo iz potreb trga.

Proces optimizacije eno-dimenzionalnega razreza materiala temelji na porabi zalog z namenom izpolnjevanja naročila, pri tem pa upošteva ekonomski cilj minimiziranja stroškov. Eden glavnih razlogov, da podjetje optimizira proces razreza eno-dimenzionalnega materiala, je v minimizaciji materiala, ki predstavlja neuporaben ostanek. Neuporaben ostanek materiala je izguba, ki jo želimo zmanjšati. V nekaterih primerih, ko je ostanek dovolj velik, se ga vrne v skladišče in ponovno uporabi (Alfieri et al., 2007; Cherri et al., 2009). Oznaka problema eno-dimenzionalnega razreza z uporabnim ostankom je 1DCSPUL (angl. *One-dimensional Cutting Stock Problem with Usable Leftover*).

Pri rezanju materiala je zelo pomembno, kako velik je ostanek rezanja, saj se premajhen ostanek smatra kot odpadek in zavrže, to pa vpliva na stroške podjetja. Smotrno je, da je ostanek, ki se zavrže, čim manjši oz. tako majhen, kot je to mogoče (Gradišar & Jesenko, 1996). Odločevalci se pri reševanju 1DCSPUL odločijo, kje je meja, nad katero se ostanek vrne nazaj v skladišče, in pod katero predstavlja izgubo pri razrezu. Če upoštevamo vračanje neuporabljenega materiala v skladišče, potem minimizacija ostanka ne predstavlja nujno najboljše rešitve za podjetje (Cherri et al., 2009). Pri 1DCSPUL gre za proces, pri katerem imamo na zalogi standardne in nestandardne dolžine. Nestandardne dolžine predstavljajo uporabni ostanki prejšnjih naročil.

Poleg optimizacijske metode lahko na višino neuporabnega ostanka oziroma izgube vpliva tudi narava problema, ki jo je pri reševanju treba upoštevati. Tako je iz praktičnega primera, predstavljenega v Erjavec et al. (2009) razvidno, da je povprečna izguba pri 1DCSPUL enaka 15 %. Razlog je v nizkem razmerju med povprečno dolžino na zalogi in povprečno dolžino naročil. Razmerje med povprečno dolžino na zalogi in povprečno dolžino naročil je nizko, če je manjše od nekega praga. V praktičnem primeru, opisanem v Erjavec et al. (2009), je ta enak 3, zato je takšen prag upoštevan v doktorski disertaciji.

Predstavljena sta razvoj in podroben opis nove metode za reševanje 1DCSPUL za primere, ko je razmerje med povprečno dolžino na zalogi in povprečno dolžino naročil manjše od 3. Učinkovitost nove metode je na primerih iz literature preverjena s primerjavo s podobnimi metodami. Nova metoda je podana na način, ki omogoča enostavno integracijo v obstoječe informacijske sisteme podjetij.

V literaturi obstaja veliko metod za reševanje 1DCSPUL, vendar pa še ni jasnega odgovora na vprašanje, kako različne metode testirati in primerjati. Kljub številnim kriterijem in primerom (Cherri, Arenales, & Yanasse, 2008; Cui & Yang, 2010) metode testiranja, ki jih je mogoče zaslediti v strokovni in znanstveni literaturi, ne zagotavljajo točnih in popolnih informacij o učinkovitosti različnih metod. Glavni problem je v tem, da so posamezni primeri ustvarjeni neodvisno eden od drugega in da vsebujejo tako podatke o naročilu kot o zalogi. Slednje pa ni v skladu z lastnostjo 1DCSPUL, da se uporabni ostanki iz prejšnjih naročil vrnejo na zalogo in uporabijo za izpolnjevanje prihodnjih. Pri obstoječih metodah testiranja uporabni ostanki ne izhajajo iz prejšnjih naročil. Namesto da bi upoštevale vračanje uporabnih ostankov iz prejšnjih naročil na zalogo, obstoječe metode testiranja temeljijo na primerih, kjer je v skladišču vedno enako število uporabnih ostankov, ki so ustvarjeni z generatorjem naključnih števil.

Zato v doktorski disertaciji predlagam nov, natančen model testiranja metod za reševanje 1DCSPUL, ki na podlagi simulacij omogoča bolj natančne primerjave različnih metod s tega področja. Metoda testiranja temelji na konceptu, kjer so uporabni ostanki, ki se ustvarijo pri izpolnjevanju trenutnega naročila, uporabljeni za izpolnjevanje naslednjega, in tako dalje. Razen v Trkman in Gradišar (2007) in Cherri et al. (2012) metode v literature ne upoštevajo vračanja ostankov iz trenutnega naročila na zalogo z namenom izpolnjevanja kasnejših naročil. Ker ni mogoče zaslediti nobene natančne metode za testiranje in primerjavo rešitev, so metode v literaturi testirane napačno. Za premostitev tega problema je v doktorski disertaciji predlagana nova metoda za testiranje, kjer uporabni ostanki niso ustvarjeni naključno, temveč so simulirani na način, ki bolje odraža stanje v realnosti kot metode v literaturi. Delovanje predlagane metode testiranja je prikazano na primerih, ki imajo različno razmerje med povprečno dolžino na zalogi in povprečno dolžino naročil.

Ključne besede
Optimizacija**,** eno-dimenzionalni problem razreza, uporabni ostanek, hevristične metode, simulacije

ONE-DIMENSIONAL CUTTING STOCK OPTIMIZATION WITH USABLE LEFTOVER: A CASE OF LOW STOCK-TO-ORDER RATIO

SUMMARY

One-dimensional cutting stock problem occurs in many forms in practice. It is mostly defined as a problem, where a number of ordered items, known in advance, must be cut from a known stock of bars. The goal is to satisfy customers' orders, deriving from the market needs.

The process of optimizing the one-dimensional cutting of material makes the use of stock bars more efficient in order to fulfill the customer's order. It follows the economic goal of minimizing the costs. Companies optimize the process of cutting one-dimensional material due to minimization of material that is treated as an unusable leftover. Leftover material thus represents a loss and should be reduced. If the leftover is of enough size it can be returned back on stock and reused (Alfieri, van de Velde, & Woeginger, 2007; Cherri, Arenales, & Yanasse, 2009). Such problem is known as one-dimensional cutting stock problem with usable leftover (1DCSPUL).

The amount of leftovers is very important when cutting material. Insufficiently large amount is regarded as a waste and discarded. That affects the company's costs. It is reasonable that the discarded leftover is as small as possible (Gradišar & Jesenko, 1996). When solving 1DCSPUL the decision-makers define the threshold above which the leftover is returned back on stock and below which the leftover represents a cutting loss. When returning usable leftovers back on stock the waste minimization does not necessarily represent the best possible solution for the company (Cherri et al., 2009). The 1DCSPUL thus consists of standard and non-standard stock lengths. The non-standard lengths are leftovers from previous orders that can be reused.

The amount of unused material or loss depends on the optimization method but can be affected also by the nature of the problem, which need to be considered when solving the problem. In practical case it is shown that the average trim loss in 1DCSPUL amounts 15% (Erjavec, Gradišar, & Trkman, 2009). The cause can be found in the low ratio between average stock length and average order length. The ratio between average stock length and average order length is low if it is lower than some threshold. In practical case, as presented in Erjavec et al. (2009), the ratio equals 3, therefore the same threshold is taken into account in presented doctoral dissertation.

I introduce the development and a detailed description of the new method for solving 1DCSPUL in cases, where the ratio between average stock length and average order length is lower than 3. The efficiency of the new method is, using cases from the literature, shown by comparison with similar methods.

In the literature it is possible to find many method for solving 1DCSPUL but there is no answer to the question of testing and comparing different solutions. Despite numerous criteria and hundreds of problem instances (Cherri et al., 2009; Cui & Yang, 2010), the testing methods used in the literature do not provide accurate and complete information about the effectiveness of different methods. The main problem is that the benchmark problem instances are generated independently from one another and that they contain both order and inventory data. This is not in line with the basic characteristic of the 1DCSPUL that usable leftovers from previous orders are used in the following ones. The existing testing methods do not take usable leftovers from previous orders into account. Instead of concentrating on returning usable leftovers from previous orders back on stock, the existing methods use problem instances, where usable leftovers in stock are generated with a random number generator.

Therefore I propose a new precise model for testing solutions to 1DCSPUL, which is based on simulation. Thus it enables more accurate comparison of different methods from the relevant field. A new testing method is based on the concept in which the usable leftovers generated in the previous order are used in the current one, and the usable leftovers generated in current order in the next one, and so on. Except for two papers (Trkman & Gradišar, 2007; Cherri et al., 2012) the methods in the literature do not take into account returning of leftovers from current instance back on stock and using them in the following ones. It is also not possible to find a sufficient and accurate testing method for comparison of the solutions hence the methods in the literature are not correctly tested. To overcome this issue a new testing method is proposed where usable leftovers are not randomly generated but are simulated in a way that reflects the real situation better than the methods in the literature. The efficiency of proposed testing method is demonstrated by cases with different ratios between average stock length and average order length.

Keywords
Optimization, One-dimensional Cutting Stock Problem, Usable Leftover, Heuristics, Simulation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The introductory chapter presents the issue and provides an overview of the research area. Subsequently, I state the purpose and the goals of the doctoral dissertation, which is followed by a description of the research methods and techniques used. Thereafter, I discuss the potential research contribution. The structure of the doctoral dissertation is described in detail in the last part of the introduction.

Presentation of the issue and overview of the research area

The one-dimensional cutting-stock problem (1DCSP) comes in a variety of forms in practice in various industrial processes. It most often occurs in paper (Chauhan, Martel, & D'Amour, 2008) metal (Abuabara & Morabito, 2009; Leão, Santos, Hoto, & Arenales, 2011; Stadtler, 1990), steel (Dimitriadis & Kehris, 2009; Hajizadeh & Lee, 2007), wood (Venkateswarlu, 2001), aluminum (Stadtler, 1990), textile (Alfieri et al., 2007; Arbib, Marinelli, Rossi, & Di Iorio, 2002; Gradišar, Jesenko, & Resinovič, 1997), aerospace (Morgan, Morton, & Daniels, 2006), medicine (Aktin & Özdemir, 2009) and glass (Tsai, Hsieh, & Huang, 2009) industries.

1DCSP is largely defined as a problem in which a number of ordered items provided in advance must be cut from a known stock of bars (various numbers of bars of various lengths). This involves filling customers' orders, which arise from market needs.

The first attempts to find an optimal solution to the problem began in 1939, when Kantorovich presented a basic model for solving 1DCSP; however, the formulation of the problem as known today was introduced in 1956 (Paull, 1956). The next important step was made in 1961, when Gilmore and Gomory presented a solution method based on linear programming (Gilmore & Gomory, 1961). Later many other solutions to the 1DCSP were presented, which consequently led to the need to classify them in a way that would clearly distinguish between various types of problems. To this end, Dyckhoff proposed a classification in 1990 (Dyckhoff, 1990) based on which several interesting methods have been developed for solving the problem (Belov & Scheithauer, 2002; Scheithauer & Terno, 1997; Schilling & Georgiadis, 2002; Vanderbeck, 2000), and later on this classification was also expanded to include the time dimension (Trkman & Gradišar, 2007). The year 2007 was also important: it saw the presentation of a new classification that indicated significant progress in cutting and packaging (Wäscher, Haußner, & Schumann, 2007). Using this classification, problems can be sorted into classes in even greater detail.

The process of optimizing the one-dimensional cutting of material is based on the use of stock in order to fill the order, while taking into account the economic goal of minimizing the costs. One of the main reasons for a company to optimize the process of cutting one-dimensional material is the minimization of leftover material. Leftover material is a loss that we want to reduce. In some cases, when there is enough leftover material, it can be returned to the warehouse and reused (Alfieri et al., 2007; Cherri et al., 2009). This is known as the cutting-stock problem with usable leftover (CSPUL). With regard to one dimension it is called one-dimensional cutting stock problem with usable leftover (1DCSPUL).

Optimal solutions usually result from the application of exact methods (Alves & Valério de Carvalho, 2008), but they are only useful for small orders (the results of exact methods are the best possible solutions, but in the case of larger orders the computing time is too long due to the insufficient computing power of computers); therefore, the use of an exclusively exact method was inappropriate for my dissertation. The majority of methods developed are thus based on a heuristic approach (Ragsdale & Zobel, 2004; Vahrenkamp, 1996; Yeung & Tang, 2003). In contrast to exact methods, heuristic solutions do not ensure an optimal solution, but they do find one within a reasonable time.

When cutting material, the amount of leftovers is very important because an inappropriately large amount is considered as waste and is discarded, which affects the company's costs. It makes sense for the amount of leftovers discarded to be as small as possible (Gradišar & Jesenko, 1996). In solving the 1DCSPUL, those making decisions set the limit above which the waste is returned to the warehouse and below which the waste represents a cutting loss. Taking into account that the unused material is returned to the warehouse, waste minimization is not necessarily the best solution for the company (Cherri et al., 2009). The 1DCSPUL essentially involves a process in which there are standard and nonstandard lengths in stock. The nonstandard lengths are the useful leftovers from previous orders. In the 1DCSPUL one should also take the following assumptions into account (Cherri et al., 2009):

- the amount of the order must be known;
- the amount of the stock must be known; and
- the orders must be filled.

An extensive overview of methods used for solving 1DCSPULs is provided in Cherri et al. (2009). The most frequently used methods are heuristics. The COLA and CUT methods have an important place among the methods used for solving 1DCSPULs; they were the first methods developed for this type of problem. The COLA method was presented in Gradišar et al. (1997), and the CUT method was presented in Gradišar, Resinovič, and Kljajić (1999). Both methods are item oriented and based on exhaustive repetition.

In addition to the optimization method, the nature of the problem can also influence the amount of unused waste or loss; thus it must be taken into account when solving the problem. The practical example presented in Erjavec, Gradišar, & Trkman (2009) shows that the average loss in the 1DCSPUL is 15%. The reason for this lies in the low ratio between average stock length and average order length (henceforth, the ratio between average stock length and average order length will be referred to as *r*). The literature does not offer any definition of a low *r*; therefore this ratio will be defined in the third chapter.

Operations research is of key importance to the problem discussed. Even though the literature provides several definitions by various authors, for the purpose of this doctoral dissertation, operations research can be defined as analytical scientific method of providing decision-makers with a quantitative basis for making better decisions in organizations (Čižman, 2004).

The purpose of operations research is to find an optimal (exact) or near-optimal (heuristic) solution to complex decision-making problems. It is based on many mathematical methods and techniques such as modeling, decision analysis, and business simulations. The beginnings of operations research date back to the late 1930s, when researchers from various areas joined forces in field operations to test and evaluate solutions in order to develop radar technology for the British Royal Air Force. Later on, various operations research groups around the world analyzed military issues. After the Second World War, the use of operations research was expanded to management, with the aim of supporting the decision-making process. Today operations research is widely used around the world to aid decision-making with a significant contribution to management, business, and industry (Assad & Gass, 2011).

The use of operations research methods in companies tackles a variety of business challenges; for example (Crowder, 2000):

- the increased complexity of running a successful business;
- new opportunities in the electronic economy; and
- a large volume of information, but no decision.

The main objective of operations research in companies is to increase efficiency, which can be achieved through the following benefits, delivered by operations research projects (Robinson, 2000):

- increased revenue or return on investment; increase market share;
- decreased cost or investment;
- greater utilization from limited equipment, facilities, money and personnel;
- assessment of likely outcomes of decision alternatives and determining better alternatives;
- managed and reduced risk;
- quantified and balanced qualitative considerations;
- increased speed or throughput; decreased delays;
- greater control and turnaround;
- improved quality;

- a better basis for forecasting and planning; and
- demonstrated feasibility and workability and assistance with training.

Companies use the applications of operations research in various business functions and categories, like board room and senior executive offices, manufacturing and service operations, distribution, transportation, telecommunications, finance, control, marketing, information technology, human resources, legal services, engineering, research and development, revenue management, supply chain management, data mining, complex scheduling and intelligent real-time systems (Robinson, 2000).

One of the important areas of operations research is optimization. It appears in many disciplines, such as economics, engineering, physics, and mathematics (Antoniou & Lu, 2007). The field developed rapidly in the twentieth century and today makes it possible to solve complex problems (Alba, Blum, Asasi, Leon, & Gomez, 2009). In practice, this means that the optimization process tends to acquire the best possible solution, which is obtained through the use of various techniques, methods, procedures, and algorithms (Antoniou & Lu, 2007).

1DCSP is an optimization problem. The method presented in the doctoral dissertation will solve the posed problems automatically. This way a company can achieve a competitive advantage over other market players. The use of information technology and the automatization of production processes can significantly reduce the cost of cutting. It is therefore possible to expect an increase in methods for solving the 1DCSP in the future (Trkman & Gradišar, 2010).

Purpose and goals

The main purpose of the doctoral dissertation is to enhance the theoretical knowledge of the field researched with the development of a new method for solving the 1DCPSUL in cases in which *r* is low.

My first goal is formulated as follows:

- To develop a new method for solving 1DCSPULs with a low *r* that will provide better results than existing methods.

The second issue involves comparing and testing different methods for solving the 1DCSPUL, which should be tested by using the same criteria and benchmark instances. Because the effectiveness of different methods has not been comprehensively and thoroughly discussed in the literature, my second goal is:

- To improve the current methods of testing solutions to 1DCSPULs with the simulation method.

Research methods and techniques

The first research method I use includes a detailed and extensive overview of the research literature (both primary and secondary sources) with an emphasis on research contributions dealing with the areas discussed. I also analyze the effectiveness criteria of one-dimensional cutting.

The results of the proposed method are obtained through the use of simulations, which are one of the most frequently used research methods in the social science (Pidd, 2004). The application of the simulations technique is important for the area discussed and therefore it is presented in greater detail in the fifth chapter of the doctoral dissertation.

The concept of using simulations for testing the method—where, contrary to individual consideration of separate instances, the group of instances would be taken into account—has not yet been theoretically supported in previous studies. Nor does the literature contain any method that would take into account a low *r*, except for a working paper (Gradišar, 2008). Therefore the results of my study include important findings and thus provide a theoretical contribution to the relevant field of knowledge; in addition, the practical contribution is expressed in lower total costs for companies that apply this method.

The test data are generated using a test case generator, although some of them are also taken from practice.

To facilitate my research I use suitable software such as:

- Microsoft Excel;
- a problem generator algorithm;
- COLA;
- ECOLA;
- CUT;
- C-CUT;
- MPL/CPLEX; and
- newly developed algorithm for solving 1DCSPULs with a low *r*.

Microsoft Excel is a spreadsheet application developed by the Microsoft Corporation. It enables users to organize, format, and manipulate data using formulas and functions. In addition to spreadsheet presentation, data can also be displayed as graphs, histograms, or charts.

The problem generator algorithm PGEN for the general 1DCSP was introduced in 2002 and generates random variables through a uniform random number generator (Gradišar, Resinovič, & Kljajić, 2002). Due to its simplicity, randomness, portability, and reproducibility, PGEN uses a pseudo-random number generator presented in 1995 (Gau &

Wäscher, 1995). A slightly modified PGEN was used to generate problem instances in this dissertation. A detailed explanation of the PGEN generator and its modification is provided in Chapter 4 of the doctoral dissertation.

COLA will be used to develop a new method for testing solutions to the 1DCSPUL. The method is explained in detail in Chapter 3 of the doctoral dissertation.

ECOLA (Tomat, Gradišar & Štiglic, 2013) is a heuristic method for solving 1DCPSUL that minimizes the trim loss and the difference between usable leftovers used and produced.

CUT is an improved and generalized version of COLA (Gradišar et al., 1997) and was developed by Gradišar et al. (1998). It is a computer program for solving the 1DCSP when the stock lengths are different. It is classified as an item-oriented method and minimizes the influence of end conditions leading to a heuristic solution (Gradišar, Kljajić, Resinovič, & Jesenko, 1999).

C-CUT (Gradišar & Trkman, 2005) is an algorithm for solving 1DCSP that is based on a combination of a sequential heuristic procedure and a branch-and-bound method. It is suitable for all problem sizes.

MPL is a mathematical computer solver for integer linear problems. It includes multiple commercial optimizers to solve the models developed.

The results will be evaluated using comparative analysis and presented numerically. I will compare the results of the new method for solving 1DCSPULs with a low *r* with those of general methods for solving 1DCSPULs.

A detailed description of the methodology, techniques, and tools used is given in the corresponding chapters of the doctoral dissertation.

Research contribution

My contribution to the relevant field of knowledge is improvement of the method published in Gradišar (2008), and a new model for testing solutions to 1DCSPULs, based on simulations, that has not yet been applied in the literature dealing with item-oriented solutions. The proposed method considers the usable leftovers from the previous instance and uses them in the next one. Thus the method reflects the real situation better than the methods in the literature, which do not take into returning usable leftovers account when testing and comparing various solutions.

The main anticipated contributions include:

- an improved method for solving 1DCSPULSs problems where *r* is low; and

- development of a new method for testing 1DCSPUL solutions that will enable more accurate comparison of various methods from this area.

Limitations

The doctoral dissertation describes methods for solving the 1DCSP. These methods could also be used for solving the general cutting stock problem, but usually more suitable methods (i.e., pattern-based methods) are selected for solving such problems (Trkman & Gradišar, 2007). I consider methods for solving the 1DCSP but do not analyze such methods for solving two- or three-dimensional cutting stock problems in the doctoral dissertation.

The newly developed method is based on a heuristic approach to solve the CPSUL. Therefore an optimal solution is not provided. I analyze the possibilities for solving the problem studied with the exact method, and in Chapter 3 of the doctoral dissertation I provide an argumentation for why the use of an exact method is not suitable in my case.

Methods for solving the 1DCSPUL deal with merely part of the entire cutting processes in a company. The reduction of trim loss refers only to a small share of the total costs that arise from preparing the required number of order lengths (Arbib & Marinelli, 2005). In order to provide higher savings, a redesign of the entire cutting stock process could be more appropriate than solving only the 1DCSPUL. Constant improvement of business processes should be a continuous routine based on the development of information technology (Trkman, 2010).

A method for solving the 1DCSPUL should be easy to adjust to changing business processes and to integrate into a company's information systems, such as ERPs or decision-support systems (Čižman & Černetič, 2004; Rodríguez & Vecchietti, 2008). Thus the company would have a better decision-support environment, which could serve as a basis for making better decisions (Liu et al., 2009).

Structure

Chapter 1 of the doctoral dissertation provides a detailed and critical literature review of the 1DCSP. I highlight the terms and concepts used in this area and present the latest classification used to sort different types of problems into corresponding categories. I provide an analysis of the exact and heuristic approach and describe the difference between these two approaches. An in-depth overview of the methods and existing algorithms for solving the 1DCSP is then given. At the end of Chapter 1, I briefly discuss the role of cutting and the cutting process in companies.

Chapter 2 analyzes the 1DCSPUL, describing the concept and providing an overview of the current state of affairs. This is followed by a detailed and critical literature review of the main methods for solving the 1DCSPUL. I define and explain the low *r* at the end of the chapter.

Chapter 3 concerns the development of a new method for the 1DCSPUL in the case of low *r*. First, I define the problem. Then, I provide an experiment that justifies the use of a heuristic approach in my doctoral dissertation, rather than an exact one. After this, I present the development of a solution. I provide a detailed explanation for each step of the proposed algorithm. Next, I test the solution, whereby I also emphasize the selection and modification of a selected problem generator. An analysis of the results is presented at the end of the chapter.

Chapter 4 highlights the importance of simulation, especially computer simulation. I then present the use of computer simulation in practice, particularly in operational management. I also present the use of computer simulation in relation to cutting processes. Next I describe the issues that arise when testing algorithms for solving the 1DCSPUL. Then I propose a new method for testing solutions to the 1DCSPUL. As a testing method I use a simulation in which a group of instances is observed. The results of the proposed method are presented at the end of the chapter.

Chapter 5, the last chapter, presents the main findings of the doctoral dissertation and highlights the importance of the results for companies. I emphasize the contribution of the doctoral dissertation to the relevant field of knowledge. This is followed by an in-depth discussion of the results obtained through the experiments in the doctoral dissertation. At the end of the chapter, outlook to further research are provided. I point out the most interesting open research issues that may serve as a basis for conducting further research in the area studied.

1 THE ONE-DIMENSIONAL CUTTING STOCK PROBLEM

Various companies are involved in the process of cutting material as part of their business processes. It is thus reasonable that the method used is tailored to a company's needs as much as possible to ensure the maximum efficiency when consuming the material. One of the most important elements a company should take into account is the amount of leftover being thrown away (trim loss) that represents an unnecessary cost for the company. Therefore, it is rational to minimize the trim loss.

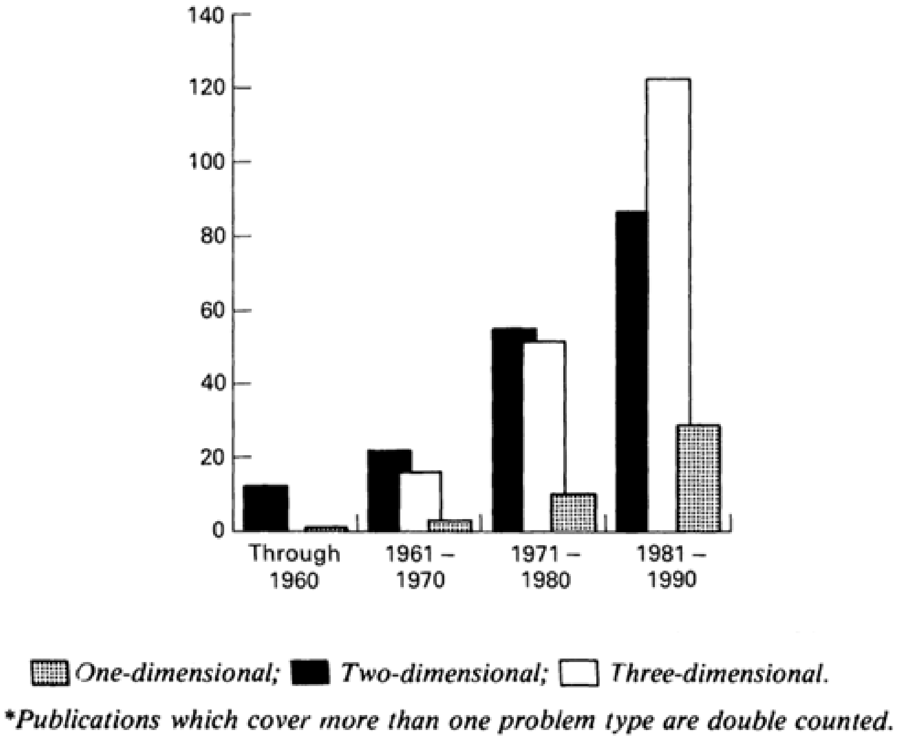
The main problem lies in the limited resources of material that have to be cut. This is usually expressed in a limited quantity or limited dimensions of material in stock due to various factors influenced by the market. Limitations regarding the dimensions of material can also derive from (Erjavec, 2011):

- the availability of material on the market;
- logistical constraints (i.e. the size and weight of the material); and
- natural limitations (i.e. maximal lengths of wooden logs dependent on the felled trees), etc.

Researchers have been engaged in resolving the cutting stock problem for more than seventy years. The first mention dates back to 1939 (Kantorovich, 1960) and since then interest in solving the cutting stock problem has been permanently on the rise. A mathematical approach was introduced in 1940 (Brooks, Smith, Stone, & Tutte, 1987) and served as a basis for further research and developments of various methods. Minimizing trim loss as a key element of cutting stock optimization emerged in 1957 (Eisemann, 1957). Great progress in solving the cutting problem was made by Gilmore and Gomory a few years later, when they presented a linear programming method for solving the 1DCSP (Gilmore & Gomory, 1961). Their method will be explained in detail in the second subchapter below.

The field of the 1DCSP problem is a well know and researched field, which can also be seen from the number of related scientific articles, published in well-cited journals. In 1992, Sweeny and Paternoster presented a study in which they analyzed more than 400 books, articles, dissertations and working papers on cutting and packing problems. They pointed out the trend of publishing scientific papers from the mentioned field. From 1960 on, when there were approximately 20 releases, the number of published scientific papers was constantly increasing. From 1961 to 1970 there were roughly 25 scientific papers, from 1971 to 1980 nearly 110, and from 1981 to 1990 around 250 (Sweeney & Paternoster, 1992). The graphical presentation is in Figure 1.

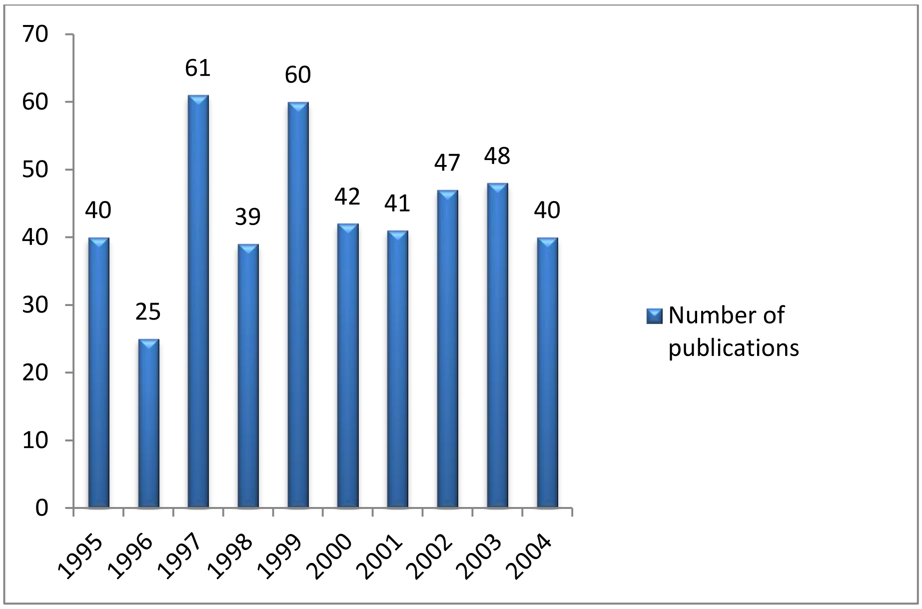
Figure 1: Publications in cutting stock problem by time periods.



Source: Sweeney & Paternoster, 1992.

Another research study of trend of published scientific papers in the field of cutting and packing problems was made by Wäscher, Haußner and Schumann in 2007. Their analysis examined the period from 1995 to 2004 in which 445 scientific papers were published. The majority of these papers were published in the field of the 1DCSP (172) (Wäscher et al., 2007). The results of the abovementioned study are presented in Figure 2.

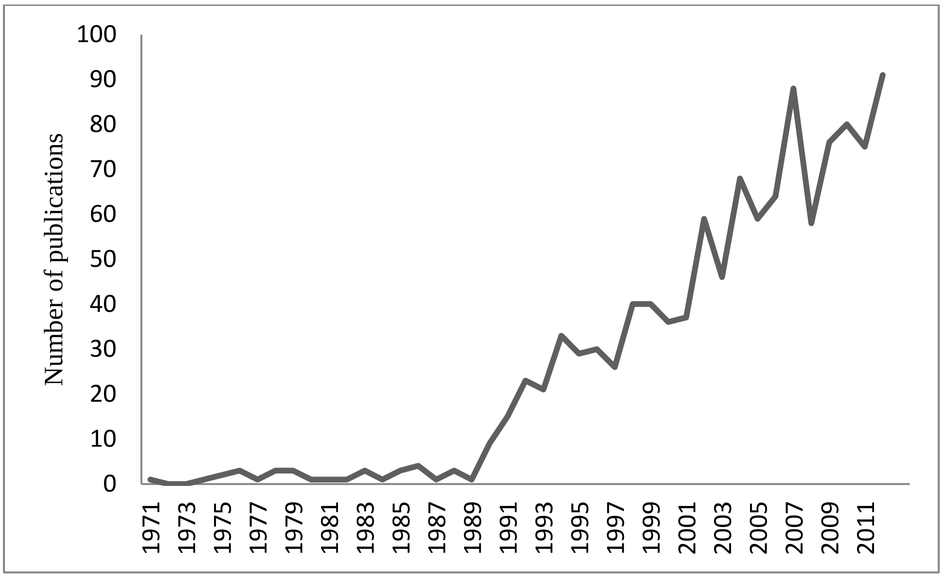
Figure 2: The number of publications in the field of cutting stock problem by year.



Source: Wäscher et al., 2007.

The most recent analysis of the number of scientific papers published in the field of cutting that could be found in the literature is presented by Cesar (2013). He conducted an analysis for the period from 1971 to 2011. The search strings consisted of cutting stock problem, nesting problem, partitioning problem, trim loss problem, guillotine problem, bin packing problem, strip packing, vector packing + cutting, knapsack + cutting, multiprocessor scheduling and usable leftovers. In addition to scientific papers written in English, Cesar also searched for publications in other languages. The search strings in the German language were *zuschnitt* (*problem*(*e*) / *-soptimierung*(*en*)) (eng. *cutting (problem(s) /optimization(s)*) and *Rucksack* + *zuschnitt* (eng. *backpack* + *cutting*), *behälterproblem* (eng. c*ontainer problem*) and *packen* (eng. *packing*), in the Russian language *задача раскроя* (eng. c*utting* problem) and *задача о ранце / рюкзаке* (eng. *problem of knapsack / backpack*), in the Spanish language *problema de la mochila* (eng. *knapsack problem*), *problema de corte* (eng. *cutting problem*) and *problema* + *guilotin* (eng. problem + guillotin) and in the Portuguese language *problema de corte de estoque* (eng. *cutting stock problem*) and *sobras aproveitáveis* + *corte* (eng. *usable leftovers* + *cutting*). He concluded that research into the problem was stagnating until 1989. From 1990 to 2007 constant growth in publishing was visible and, after some oscillations from 2007 to 2010, the highest number of published papers in history occurred in 2011. The results of this analysis are presented in Figure 3.

Figure 3: The number of scientific papers published in the field of cutting from 1971 to 2011.



Source: Cesar, 2013.

For the purposes of this doctoral dissertation, I conducted an additional analysis to narrow down the publications to the field of the cutting stock problem and, more precisely, the 1DCSP. The main difference from the abovementioned analysis is that I adjusted the search string to the cutting stock problem for the purpose of exposing only those publications that are closely related to my doctoral dissertation.

As a tool for the presented analysis I chose Web of Science™, which is an online academic citation indexing and search service provided by Thomson Reuters. The platform enables access to scientific content using tools provided for searching, tracking, and measuring. It collaborates in the sciences, social sciences, arts, and humanities. It covers more than 23,000 journals, 23 million patents, 110,000 conference proceedings, 5,500 websites, 2 million chemical structures, 100 years of back files, 40 million source items, 700 million cited references and 256 scientific disciplines. Among other things, the analytical tool I used enables the most prolific authors on a topic to be determined, the institutions that have published the most in a specific field to be identified and publication trends for a specific scientific field to be analyzed.

Four search strings were taken into account. The first was “cutting”, the second “cutting stock problem”, the third “one dimensional cutting stock problem” and the fourth “trim loss". Web of Science includes a special tool for making an analysis with an arbitrary search string, but for that the search string must be included in one of the suggested fields. I first conducted an analysis where the search strings were only included in the title of the publication. It turned out that there is a relatively small number of papers that include the strings in the title and therefore only the analysis based on the strings included in the topic of a publication is presented in this doctoral dissertation. Analysis where the search string was "cutting" had shown that there exist many papers from various research areas that are not relevant for my doctoral dissertation. Therefore the analysis was narrowed down only to papers that are classified to be from the area of operations research management science[1].

I conducted an analysis for the period from 1994 to the end of October 2013. The results are shown in Table 1.

Table 1: The number of publications by year and search string.

Year	Cutting	Cutting stock problem	Trim loss	One dimensional cutting stock problem
1994	98	20	27	4
1995	98	25	23	7
1996	109	27	32	7
1997	136	24	31	5
1998	149	31	36	5
1999	131	35	39	6
2000	137	29	33	4
2001	141	24	38	3
2002	176	40	46	14
2003	150	31	31	5
2004	179	43	42	9
2005	219	53	47	12
2006	225	55	53	11
2007	299	78	56	19
2008	299	70	54	14
2009	367	59	58	13
2010	306	55	50	11
2011	299	68	45	12
2012	324	65	56	13
2013[1]	254	50	44	14

1.Only publications that were recorded in Web of Science by the end of October 2013 are taken into account.

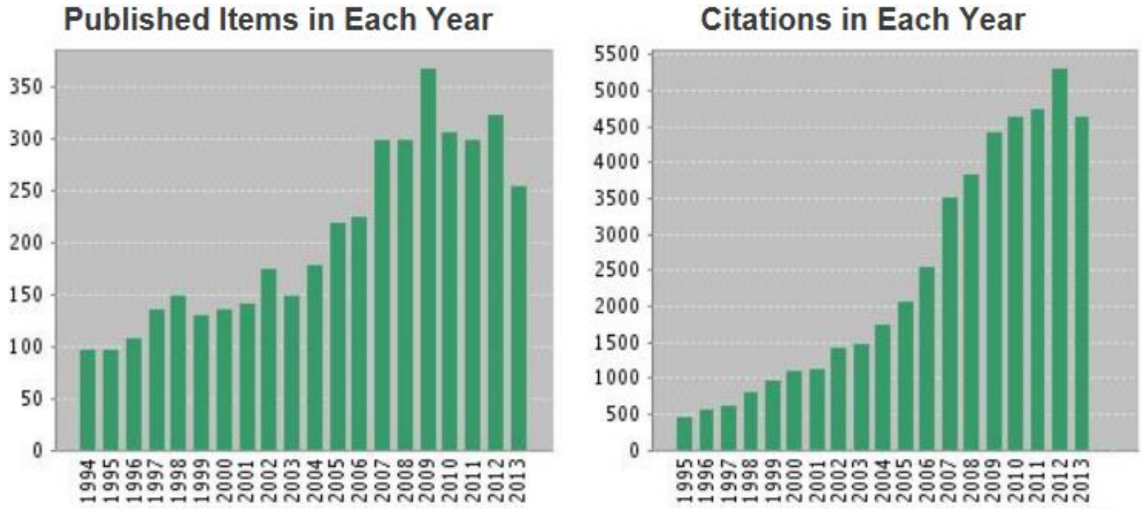
Source: Web of Science, 2013; own analysis.

A graphic presentation of the number of publications per year for the search string "cutting" with the corresponding citations indexed within Web of Science is provided in

[1] Based on the Web of Science classification of research areas.

Figure 4. Please note that the data for 2013 only include papers that had been published by the end of October 2013.

Figure 4: The number of publications and corresponding citations by year.

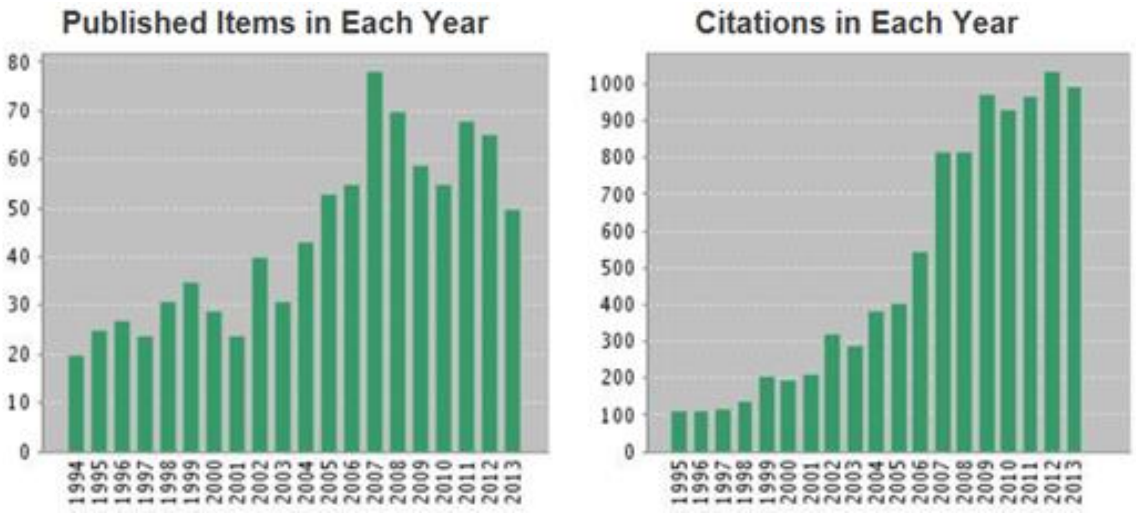


Source: Web of Science (search string "cutting"), 2013; own analysis.

The graphic presentation in Figure 4 only provides data from 1994 to 2013, although the analysis was conducted for the whole timespan. The sum of all publications was 4,614 papers that were cited 48,476 times, of which 11,721 are self-citations. Those papers were cited in 26,583 publications, 2,605 of which are self-citations. On average, every paper was cited 10.51 times.

A graphic presentation of the number of publications per year for the search string "cutting stock problem" with the corresponding citations indexed within Web of Science is provided in Figure 5. Please note that the data for 2013 only include papers that had been published by the end of October 2013.

Figure 5: The number of publications and corresponding citations by year.

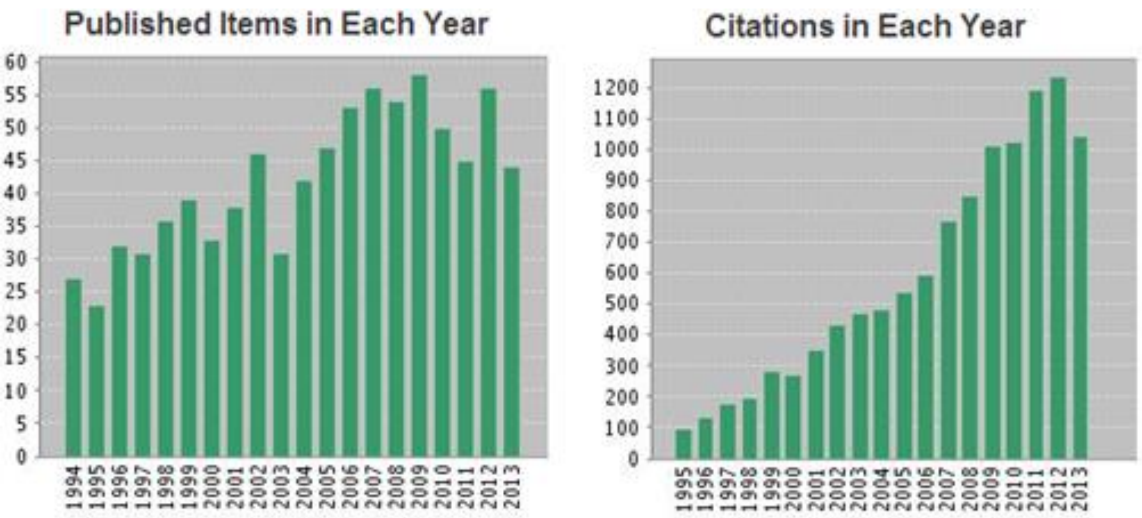


Source: Web of Science (search string "cutting stock problem"), 2013; own analysis.

The graphic presentation in Figure 5 only provides data from 1994 to 2013, although the analysis was conducted for the whole timespan. The sum of all publications was 985 papers that were cited 10,308 times, of which 4,510 are self-citations. Those papers were cited in 4,985 publications, 739 of which are self-citations. On average, every paper was cited 10.46 times.

A graphic presentation of the number of publications per year for the search string "trim loss" with the corresponding citations that are indexed within Web of Science is provided in Figure 6. Please note that the data for 2013 only include papers that had been published by the end of October 2013.

Figure 6: The number of publications and corresponding citations by year.

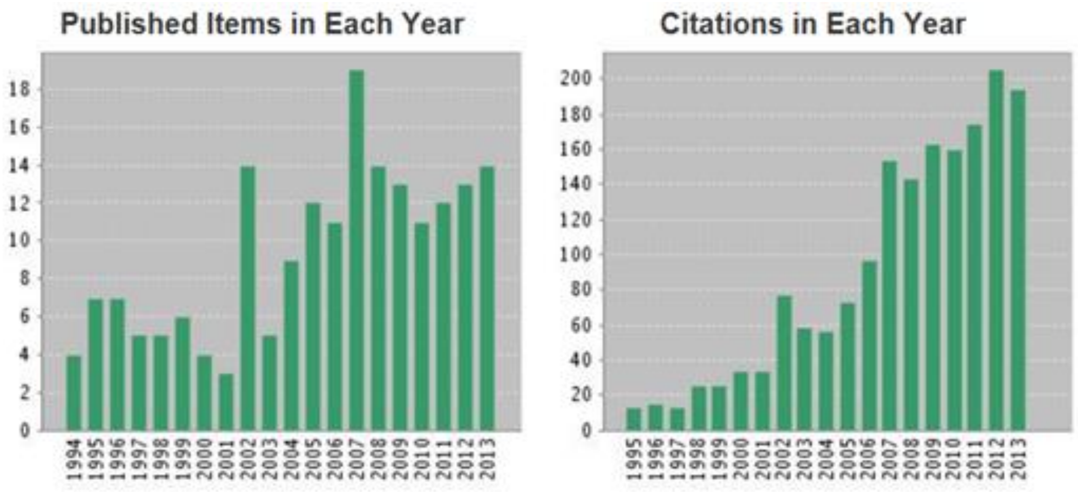


Source: Web of Science (search string "trim loss"), 2013, own analysis.

The graphic presentation in Figure 6 only shows data from 1994 to 2013, but the analysis was conducted for the whole timespan. The sum of all publications was 924 papers that were cited 11,379 times, of which 514 are self-citations. Those papers were cited in 9,984 publications, 218 of which are self-citations. On average, every paper was cited 12.31 times.

A graphic presentation of the number of publications per year for the search string "one dimensional cutting stock problem" with the corresponding citations that are indexed within Web of Science is provided in Figure 7. Please note that the data for 2013 only include papers that had been published by the end of October 2013.

Figure 7: The number of publications and corresponding citations by year.



Source: Web of Science (search string "one dimensional cutting stock problem"), 2013, own analysis.

The graphic presentation in Figure 7 only shows data from 1994 to 2013, but the analysis was conducted for the whole timespan. The sum of all publications was 197 papers that were cited 1,727 times, of which 517 are self-citations. Those papers were cited in 875 publications, 144 of which are self-citations. On average, every paper was cited 8.77 times.

It can be seen in Figure 5, Figure 6 and Figure 7 that the number of publications in the areas of the cutting, the cutting stock problem, the one-dimensional cutting stock problem and the trim loss is still increasing, even though the maturity level has probably been reached. This also holds for the number of citations. Hence, it is possible to claim that the studied area is very interesting for research.

The majority of scientific papers in the area of the cutting stock problem were published in leading journals from the field of operational research. The distribution of the publications through the journals is presented in Table 2[2].

Table 2: The number of publications in the field of cutting stock problem by journals.

Journal title	**Number of publications***	**%**
European Journal of Operational Research	123	15.36
Computers and Operations Research	60	7.50
Journal of the European Operation Research Society	49	6.13
International Journal of Production Research	31	3.88
Operations Research	29	3.63
Annals of Operations Research	24	3.00
Informs Journal of Computing	22	2.75
Computers Industrial Engineering	13	1.63
Mathematical and Computer Modeling	12	1.50
Mathematical Programming	12	1.50
International Journal of Production Economics	12	1.50
Computational Optimization and Applications	10	1.25
Engineering Optimization	9	1.13
IIE Transactions	8	1.00
Management Science	7	0.88
International Journal of Advanced Manufacturing Technology	6	0.75
Journal of Intelligent Manufacturing	6	0.75
Mathematics of Operations Research	6	0.75
…	…	
TOTAL	**800**	**100**

***** Search string “cutting stock problem” included in the topic of publication.

Source: Web of Science (search string “cutting stock problem”), 2013; own analysis.

The analysis showed that more than 15% of all publications from the cutting stock problem field were published in the European Journal of Operational Research, followed by Computers and Operations Research with more than a 7% share and the Journal of the European Operation Research Society with more than a 6% share. A large range of journals has published less than six scientific papers from the analyzed field, but they are not presented in Table 2 due to space limits. Nevertheless, it is possible to conclude that

[2] Only journals that have published six or more scientific papers concerning the field of the cutting stock problem are included in the table due to space limits.

the variety of journals from different fields indicates the prevalence of the cutting stock problem in various industries.

I also analyzed publications from the considered scientific field by individual researchers. The analysis of authors by the number of papers is presented in Table 3. The findings are provided later.

Table 3: The number of scientific papers by authors.

"Cutting stock problem"			**"One dimensional cutting stock problem"**		
Researcher[3]	**Number of publications**	**%**	**Researcher[4]**	**Number of publications**	**%**
Cui Y.D.	48	6.00	Cui Y.D.	22	12.87
Scheithauer G.	16	2.00	Scheithauer G.	12	7.02
De Carvalho J.M.V.	15	1.88	De Carvalho J.M.V.	7	4.09
Hifi M.	14	1.75	Belov G.	7	4.09
Kendall G.	13	1.63	Gradišar M.	7	4.09
Burke E.K.	11	1.38	Terno J.	7	4.09
Arenales M.N.	10	1.25	Hifi M.	5	2.92
Terno J.	10	1.25	Arenales M.N.	4	2.34
Vanderbeck F.	10	1.25	Caprara A.	4	2.34
Yanasse H.H.	10	1.25	Monaci M.	4	2.34
Alves C.	9	1.13	Trkman P.	4	2.34
Gradišar M.	9	1.13	Wäscher G.	4	2.34
Belov G.	8	1.00	Burke E.K.	3	1.75
Bennell J.A.	8	1.00	Kendall G.	3	1.75
Monaci M.	8	1.00	Kljajić M.	3	1.75
Marinelli F.	7	0.88	Resinović G.	3	1.75
Morabito R.	7	0.88	Umetani S.	3	1.75
Vasko F.J.	7	0.88	Yanasse H.H.	3	1.75
Wäscher G.	7	0.88	Rohling H.	3	1.75
…	…	…	…	…	…
TOTAL	**800**			**171**	**100**

Source: Web of Science (search strings "cutting stock problem" and "one dimensional cutting stock problem"), 2013; own analysis.

According to both search strings, Y.D. Cui is the researcher who has published the highest number of papers. He has put out 6% of all published papers concerning the "cutting stock

[3] Only researchers that have published 7 or more scientific papers are included in the presentation due to space limits.
[4] Only researchers that have published 3 or more scientific papers are included in the presentation due to space limits.

problem" search string and almost 13% with the "one dimensional cutting stock problem" search string. In both cases, Y.D. Cui is followed by G. Scheithauer (2% of all publications with the "cutting stock problem" search string and 7.02% with the "one dimensional cutting stock problem" search string) and J.M.V. De Carvalho (1.88% of all publications with the "cutting stock problem" search string and 4.09% with the "one dimensional cutting stock problem" search string). The smaller the number of papers, the higher is the number of authors. It would go well beyond the purpose of this analysis to state them all and therefore only the first 19 authors are listed.

I also analyzed the number of published papers issued in journals categorized as journals in the field of operational research by Web of Science. Analysis was carried out with the "cutting stock problem" search string being included in the topic of a publication. The results are shown in Table 4.

Table 4: Publications in the field of operational research by authors.

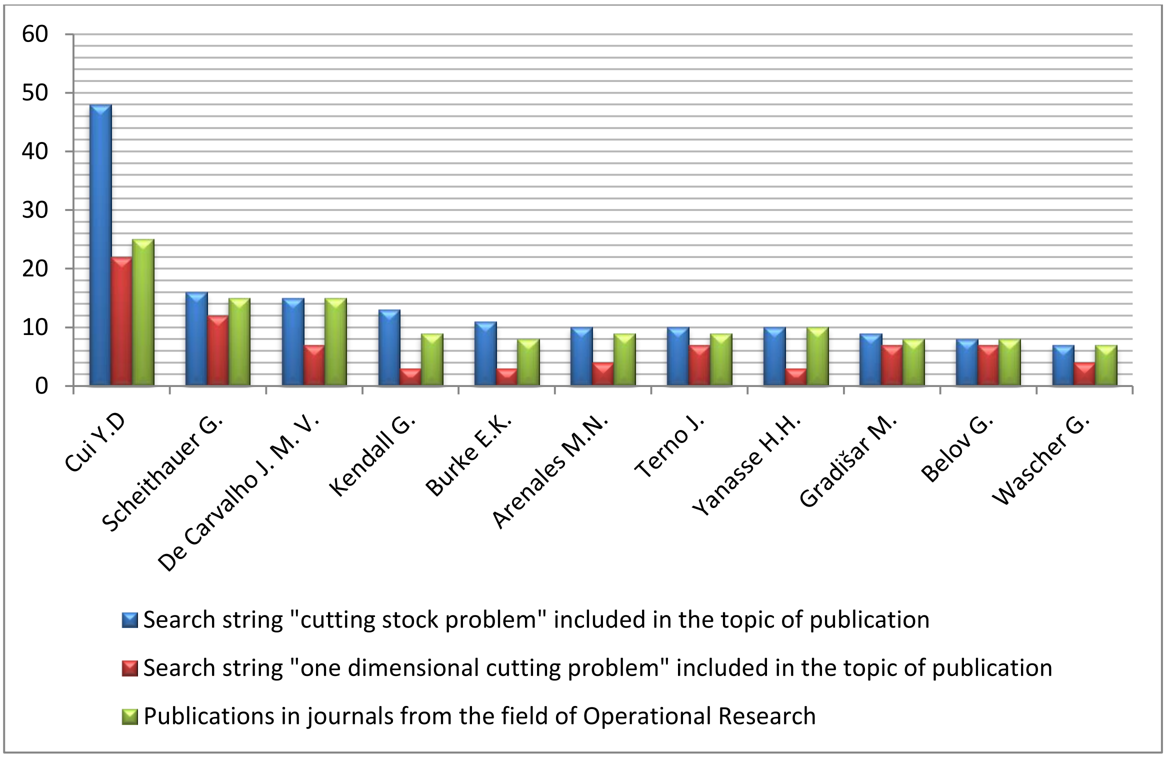
Author	Number of publications[1]	%
Cui Y.D.	25	5.07
De Carvalho J.M.V.	15	3.04
Scheithauer G.	15	3.04
Hifi M.	11	2.23
Vanderbeck F.	10	2.03
Yanasse H.H.	10	2.03
Alves C.	9	1.83
Arenales M.N.	9	1.83
Kendall G.	9	1.83
Terno J.	9	1.83
Belov G.	8	1.62
Burke E.K.	8	1.62
Gradišar M.	8	1,62
…	…	…
TOTAL	**493**	**100**

1. Search string "cutting stock problem" is included in the topic of publication.

Source: Web of Science (search string "cutting stock problem"), 2013; own analysis.

To more comprehensively represent the abovementioned analysis of researchers, I combined those authors who appear in all three categories (the search strings "cutting stock problem" and "one dimensional cutting stock problem" in the topic of the publication are the first two, while the third is the number of publications in the field of operational research). The findings are presented in Figure 8.

Figure 8: Publications by authors in dependence on search category.



Source: Web of Science, 2013; own analysis.

I also analyzed the number of times the authors have been cited. Web of Science does not provide a tool that enables searches of citations by the topic of a publication. An analysis of citations of papers that include the search strings in the topic of the publication would exceed the scope and purpose of this doctoral dissertation. Accordingly, the presented data only refer to publications with one of the search strings in the title. The results are set out in Table 5.

Table 5: The number of citations per author.

"Cutting stock problem"			"One dimensional cutting stock problem"		
Researcher[5]	**Number of citations**	**%**	**Researcher[6]**	**Number of citations**	**%**
Cui Y.D.	48	3.30	Scheithauer G.	11	4.89
Hifi M.	24	1.65	De Carvalho J.M.V.	9	4.00
Haessler R.W.	22	1.52	Alves C.	8	3.56
Jansen K.	20	1.37	Gradišar M.	8	3.56
De Carvalho J.M.V.	19	1.31	Arbib. C	7	3.11
Kendall G.	16	1.10	Arenales M. N.	7	3.11
Burke E.K.	15	1.03	Belov G.	7	3.11
Scheithauer G.	15	1.03	Cui Y. D.	7	3.11
Vanderbeck F.	14	0.96	Marinelli F.	7	3.11
Caprara A.	13	0.89	Terno J.	6	2.67
Arenales M.N.	12	0.82	Wäscher G.	6	2.67
Huang W.Q.	12	0.82	Yanasse H.H.	6	2.67
Monaci M.	12	0.82	Trkman P.	5	2.22
…	…	…	…	…	…
TOTAL	**1455**			**225**	**100**
***Only publications that include search string in the title are considered.**					

Source: Web of Science, (search strings "cutting stock problem" and "one dimensional cutting stock problem"), 2013; own analysis.

Most of the methods published by those authors who were the most cited will be presented in the second and third chapters. Papers published by those authors also serve as a theoretical background to my doctoral dissertation.

I carried out presented analysis to highlight the importance of studied area. For deeper understanding of the process of knowledge creation a systematic literature network analysis should be conducted based on existing methodologies for reviewing literature and analyzing citation network. Examples can be found, for instance, in Tranfield, Denyer and Palminder (2003), Saunders, Lewis and Thornhill (2009), Jahangirian et al. (2010), Mills, Eurepos and Wiebe (2010), Yin (2011) and Collicha and Strozzi (2012). However, such analysis would go beyond the purpose of doctoral dissertation and is therefore not carried out.

[5] Only authors that have been cited 12 times or more are included in the presentation due to spatial limitations.

[6] Only authors that have been cited at least 5 times or more are included in the presentation due to spatial limitations.

1.1 Presentation of the terms and concepts

As explained in the introductory chapter of this dissertation, the 1DCSP considers the cutting of ordered material (which usually comes in longer pieces and is denoted as items) from the material in stock (that comes in shorter pieces and is denoted as bars). The most common objective is to minimize the trim loss, but other goals can also be taken into consideration. For example, the minimization of (Trkman & Gradišar, 2010):

- costs;
- cutting time;
- number of changes in blade's settings; and
- amount of stock etc.

An illustrative presentation of the 1DCSP can be seen, for example, in Dyckhoff (1990). In stock, there are 3 bars, each 98 units long. The assortment is fixed and cannot be changed. Items demanded in an order must be cut from those bars. The orders are unpredictable and therefore the specifications, provided in an order represent a stochastic process.

An illustrative order consists of items of different lengths. The specifications are presented in Table 6.

Table 6: Order specifications.

Item length	Ordered quantity
5	4
6	2
11	1
25	2
28	1
44	1
46	3

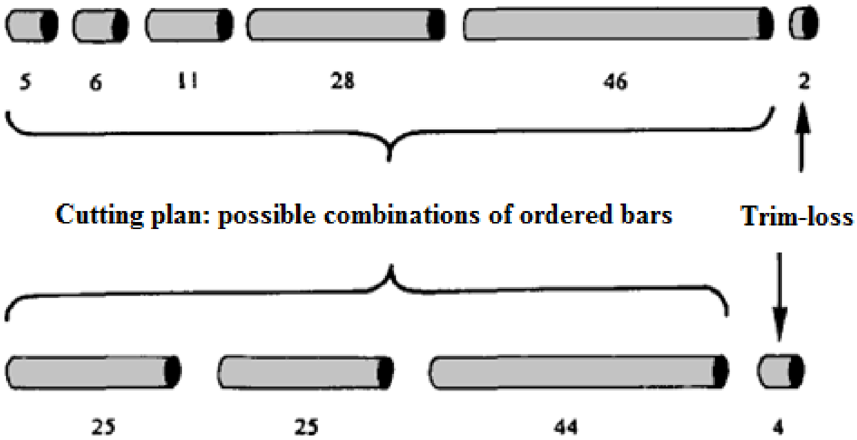
Source: Dyckhoff, 1990.

To prepare the cutting plan it is necessary to satisfy two conditions:

- the ordered items must not overlap each other; and
- the order items can only be placed in bigger objects.

The cutting plan depends on the cutting goal, which is usually minimization of the trim loss, and presents the possible ways of cutting. The cutting plan for the illustrative example is shown in Figure 9.

Figure 9: Cutting plan.



Source: Dyckhoff, 1990.

A lot of past research effort was oriented to trim loss minimization and modern methods therefore allow for very small residual pieces, also known as retails or leftovers, to be considered as waste (Cui & Yang, 2010). However, other aspects should also be taken into consideration. Trim loss minimization does not necessarily ensure the minimal possible total cost since the methods do not consider other processes associated with the cutting process, such as purchasing, warehousing, production, and sales. Researchers should keep this in mind when conducting further research. Due to the relatively small improvement possibilities, some approaches evolved that consider certain other criteria as well, e.g. (Rodríguez & Vecchietti, 2008):

- returning leftovers that are bigger than some threshold back to stock for the purpose of meeting future demands;
- energy consumption;
- An ecological effect; and
- total production cost.

The development of many solutions to cutting and packing problems consequently created a need to classify them in a way that clearly distinguishes between various types of problems and this led to Dyckhoff proposing a classification (Dyckhoff, 1990). Later on, this classification was expanded to include the possibility of a few groups of identical large objects (Gradišar et al., 2002) and the time dimension (Trkman & Gradišar, 2007). The year 2007 is also important: it saw the presentation of a new classification that indicated significant progress in cutting and packaging (Wäscher et al., 2007). Using this classification, problems can be sorted into classes in even greater detail.

An interesting approach for highlighting the importance of the field of cutting is shown by Cesar. He paid particular attention to patents for the following reasons (Cesar, 2013):

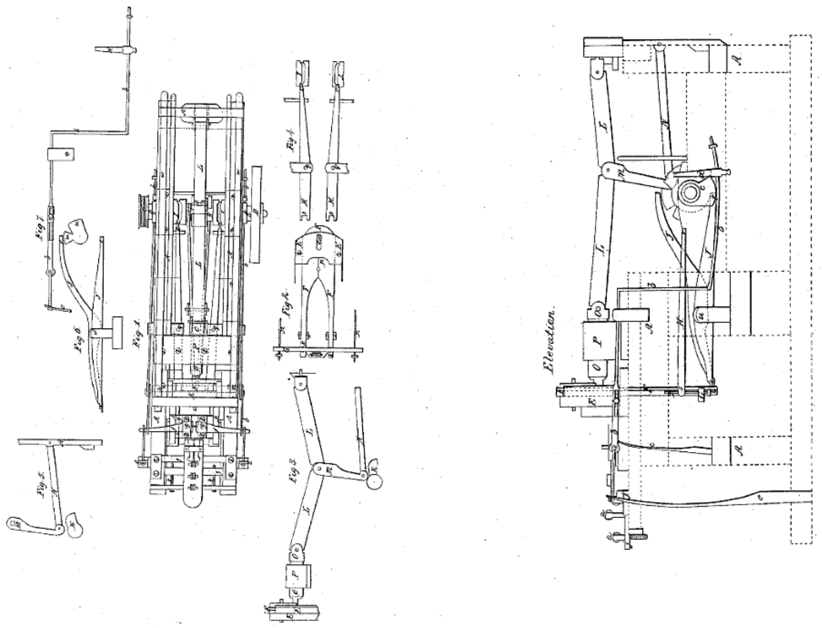
- algorithms can be patented, not just published in the journal;
- patents have appeared much earlier than scientific publications;
- patent language is not a limitation contrary to scientific publications, where papers published in the English language have a higher chance of being cited, which also affects the impact factor;
- patent application usually takes economic objectives into account; and
- in some countries patents are more respected than scientific publications (i.e. China).

In some scientific fields one can see increased growth in innovations at the expense of patents and decreased growth in innovations at the expense of scientific publications (Hicks, Breitzman, Olivastro, & Hamilton, 2001; Tansey & Stembridge, 2005).

Patents in the field of cutting are usually introduced as an answer to the needs of the industry and do not represent the result of a theoretical study. Since the presented doctoral dissertation seeks to link academic research with commercial practice I will now present some of the relevant patents.

The first patents in the field of cutting were introduced in the 19th century and consisted of various prototypes. One of the first is a machine for cutting wire for manufacturing wood screws and rivets. It was presented by Pierson (1836). The design is shown in Figure 10.

Figure 10: Patent design.

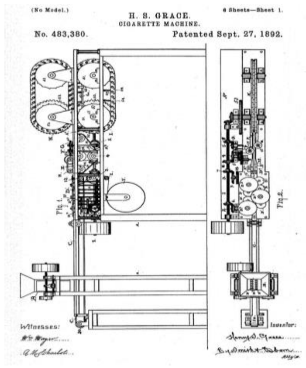


Source: Pierson, 1836.

Afterwards many machines for cutting various materials were patented. In steel industry patents were registered, i.e., by:

- Cornell (1872).
- Grace (1892).
- Carnahan (1888).
- Koehler (1890).
- Hammond (1894).
- Sheldon (1896).
- Slettengren (1950).
- …

Figure 11: Grace patent.

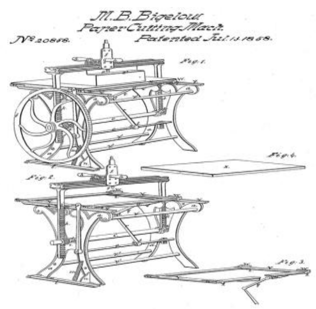


Source: Grace, 1892

In paper industry there have also been many patents. Some of the most reverberant are:

- Gage (1848).
- Bigelow (1858).
- Dougherty (1862).
- Hatch (1869).
- Aekell (1872).
- Cohek (1875).
- Bradt (1907).
- …

Figure 12: Bigelow patent.

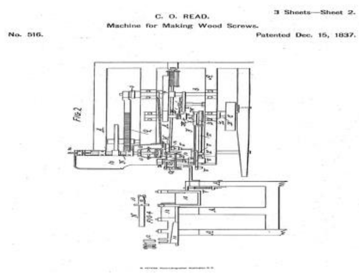


Source: Bigelow, 1958.

Also in wood industry a high number of patents emerged:

- Read (1837).
- Keane (1838).
- Bead (1846).
- Smith (1855).
- Hodge (1861).
- Wooster (1914).
- Leon (1944).
- …

Figure 13: Read patent.

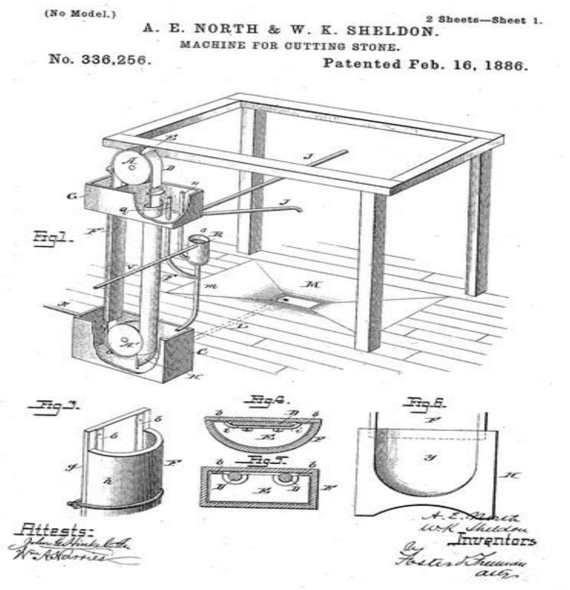


Source: Read, 1837.

The development of patents in industries other than steel or wood was not so frequent but there still exist many, i.e.:

- North (1886): the invention is related to improvements in elevating and allocating device that was especially modified for use of a mixture of sand and water material.
- Neumair (1940): the invention is proposed for cutting cigar wrappers, designed especially for a bright material such as polished metal foil blacked with a cellulosic fabric coated with elastic lacquer.
- Coddington (1943): invention is intended for cutting grass and weeds, not only on smooth terrain, but also on rough ground.
- …

Figure 14: North patent.



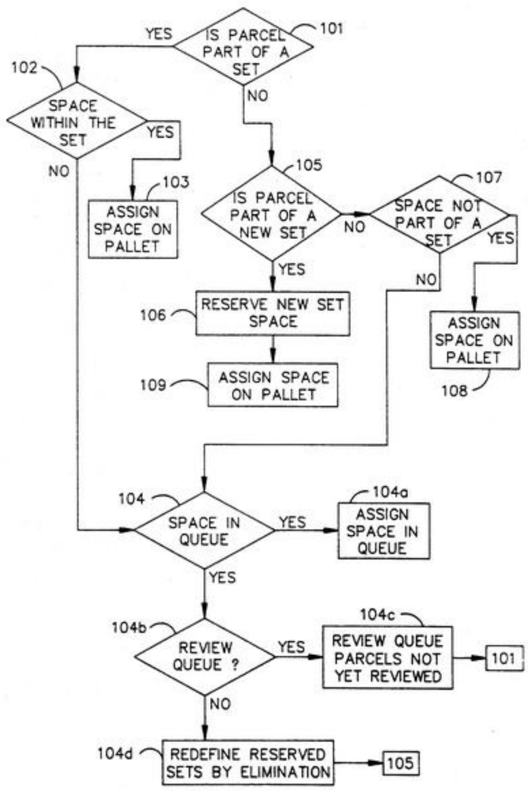
Source: North, 1886.

Patents are relevant to my dissertation since it is also possible to patent the algorithms used for cutting material. The majority of such patented algorithms come from practice and is in line with economic interests. Since the development of algorithms is closely associated with the development of computers, especially computational power, the patenting of cutting algorithms started after 1980 (Cesar, 2013).

Patented algorithms are usually not intended for general use but are adjusted to special cases. It is important to distinguish between methods and technology. Patents can relate to both, but the methods (algorithms) are typically strongly related to a specific type of machine or technology and describe the process of handling the material. Such a patent that is a combination of technology and method can be found in Singh and Jolly (1987) where the inventors had proposed a method which supervises the cutting activity with the use of high temperature. The authors suggested the use of infra-red temperature sensors to sense the surface temperature, which is a few millimeters from the width of the cut made by a saw or cutting torch and to sense the temperature in the kerf of the plate being cut. The cutting process is then optimized by a microprocessor that derives a control algorithm to sustain the recognized temperature and respective standard ranges.

The number of patents in the field of operational research in the last decade has been increasing, which also shows the importance of the field. Patented algorithms can be displayed in various forms but the most commonly used one is a presentation of the logical structure. An example can be seen in the patent by Mazouz, Slutzky, Hall, Shell, and Huston (1992). The method part is shown in Figure 15.

Figure 15: A part of the patented cutting method documentation.



Source: Mazous et al., 1992.

The proposed method combines fields of cutting and packing. The algorithm controls the cutting activity and suggests where to dispose of the cut material in the best possible place. Then the apparatus automatically places parcels of various sizes and shapes on a pallet. The material first comes into the machine randomly. It is then addressed by a computer which is based on an expert system that obtains the required information and applies predetermined rules to assign a space for the parcel on the pallet. The coordinates are then automatically communicated to a flexible material handling robot (Mazouz et al., 1992). Such methods can also be modified for the purposes of the 1DCSPUL (one-dimensional cutting stock problem with usable leftovers), but that would be beyond the limits of this dissertation.

Many algorithms have been patented for cutting material in various industries by different inventors in the last few years, for example:

- Tanida (2009): the patent consists of a device for identifying the pipe or tube before the cutting process starts, a reader to acquire information while the cutting process is performed, a management system for storing the production history and thread cutting for the removal of identifiers.
- Ylitalo (2011): the patent encompasses a method for transmitting data to a communication system.
- Borovicka, Hlavacek, Hloch, and Valicek (2012): the inventors presented a method for the design of material cutting technology based on the determination of a so-called constant of cuttability of materials. They provided four versions of the method that depend on the parameters being measured.
- Norberg Ohlsson (2012): the author presented a method and a system for a cutting machine based on beam cutting technology. The method consists of a set of controlling rules and variables for cutting patterns.

Various patents have been specially developed for solving a multi-dimensional cutting problem, e.g.:

- In 1988, a special method for the purposes of the metal industry was introduced. The invention represents a cutting tool composed of a ceramic matrix that has a distributed reinforcement comprising ceramic whiskers (Rhodes, Dziedzic, & Beatty, 1988).
- An interesting method for cutting sheet material was patented in 1988. The invention enables multiple layups of sheet material to be cut in a side-by-side or stacked connection on the cutting plate of a computer-controlled machine. It allow a different marker of a selection of pattern parts to be cut from each of the stacked layups (Pomerleau, Vivirito, & Markowitz, 1998).
- A special dimensional cutting tool for cutting wooden stock was introduced in 1988. It consists of a flat surface that can be moved to accommodate varying thicknesses of material (Rice, 1988).
- A method for manufacturing wire was presented in 1995. It is based on cutting ribbon-like material into thin rectangular strips, welding the strips together and rounding the edges (Stazaev, 1995).
- A special method of cutting material for use in medicine for implants was developed in 2011. It is grounded on a plotted laser cutting system that is controlled by a computer program and includes a motion system (Cali & Myers, 2011).

These patents represent an important driver of progress in tackling practical issues that arise from market needs. They represent the theoretical and practical development of the researched field and therefore future studies in the area of the 1DCSP should consider analyzing patents in greater detail.

1.1.1 The classification of cutting and packing problems

There are many variants of cutting and packing problems in practice due to the presence of the problem in a wide range of industries, as explained in the introductory part of the dissertation. Consequently, there are many different approaches to tackle the studied problem. Many different solutions to the problem have been presented, leading to the need for a classification that clearly distinguishes such problems. Therefore, in 1990 the Dyckhoff classification scheme for the cutting stock problem was introduced (Dyckhoff, 1990).

Beside cutting and packing problems, other problems in mathematics, operational research, logistics, production and economics have a similar logical structure. Dyckhoff therefore developed a consistent and systematic approach for a comprehensive typology integrating the various kinds of problems with the aim of standardizing the different terminology used in the literature. The problems that have a similar logical structure and appear in the literature under different names can be solved using the same or similar methods. Those problems are (Dyckhoff, 1990):

- cutting stock and trim loss problems;
- bin packing problems;
- dual packing problems;
- strip packing problems;
- vector packing problems;
- knapsack (packing) problems;
- vehicle loading, pallet loading, container loading, and car loading problems;
- assortment, depletion, design, dividing, layout, nesting, and partitioning problems; and
- capital budgeting, change making, line balancing, memory allocation, and multi-processor scheduling problems.

The proposed systematization was based on the differentiation of the main characteristics of cutting and packing problems regarding their reference to a stock of large objects, a list of small items, patterns as geometric combinations of small figures for one large figure each or for the assignment of small items to patterns as well as patterns to large objects.

The characteristic considered to be the most important is dimensionality, which is defined as the minimum number of dimensions of real numbers necessary to describe the geometry of patterns. Basic dimensionality types are one-dimensional, two-dimensional, three-dimensional and multi-dimensional (e.g., time as a fourth dimension, multi-period capital budgeting, vector packing problem…) types of the cutting and packing problem (Dyckhoff, 1990).

The next important characteristic is quantity measurement where two possibilities exist. The first considers a discrete (or integer) measurement where the frequency or number of objects or items of a certain shape is taken into account. The second one considers a continuous (or fractional) measurement where the whole length of several objects or items having the same shape is measured with regard to the relevant dimensions, the length of the objects or items being summed up with respect to a further dimension that is not essential to the geometry of the patterns (Gilmore, 1979).

Another characteristic of the discussed classification is the shape of figures. This is defined as a geometric representation in a space of relevant dimensions. The shape of figures can be unambiguously determined by its form, size and orientation. Regarding one-dimensional problems, which form an essential part of my doctoral dissertation, the size of items is the most important and can be regular (or standard) or irregular (non-standard) (Dyckhoff, 1990).

Assortment is another characteristic for use in classifying cutting and packing problems and is defined by the shapes and number of permitted figures. There are two possibilities. One is that different forms can occur and the other is that all items have the same form. The next characteristic is availability and relates to the availability of objects concerning the lower and upper bounds of their quantity, their sequence or order and the date when an object can or has to be cut or packed (Dyckhoff, 1990).

Pattern restrictions are another characteristic and can be divided into four groups when constructing geometric combinations. The first group relates to minimal or maximal distances between small figures or between cuts dividing large figures. The second group takes account of the orientation of small figures relative to each other and/or a large figure. The third group regards restrictions in the frequency of small items in a pattern, while the fourth group concerns the type and number of permitted cuts (Dyckhoff, 1990).

The seventh important characteristic concerns assignment restrictions when arranging small items into suitable patterns or when assigning patterns to appropriate large objects. The assignment restrictions relate to the kind of assignment, number of assignment stages, number and frequency or sequence of patterns and the dynamics of allocation (Dyckhoff, 1990).

The next characteristic is named objectives and means a criterion that has to be maximized or minimized. Different objectives may refer to quantities of large objects or small and residual pieces assigned to patterns, the geometry of the patterns or sequence, a combination or the number of patterns. The last characteristic relates to the status of information and variability with reference to a deterministic, stochastic or uncertain definition of problem data (Dyckhoff, 1990).

All of the abovementioned characteristics can be summed up as four main criteria that enable the classification of cutting and packing problems. The appropriate notations are given in brackets (Dyckhoff, 1990):

1. Dimensionality:
   - one-dimensional (1)
   - two-dimensional (2)
   - three-dimensional (3)
   - *n*-dimensional with $N > 3$ (N)

2. Kind of assignment:
   - all objects and a selection of items
   - a selection of objects and all items

3. Assortment of large items.
   - one object (O)
   - identical figure (I)
   - different figures (D)

4. Assortment of small items:
   - few items (of different figures) (F)
   - many items of many different figures (M)
   - many items of relatively few different (non-congruent) figures (R)
   - congruent figures (C)

The presented Dyckhoff classification has been the subject of some expansion proposals. Gradišar et al. (2002) suggested a further possibility under the third group of Dyckhoff's classification:

- (G) a few groups of identical large objects.

The next proposal for an improved typology emerged from Trkman and Gradišar (2007) due to the inability of the Dyckhoff classification to differentiate between consecutive and instantaneous solutions of cutting stock problems. The new class is called the time-span of optimization and comes in a variety of three options (Trkman & Gradišar, 2007, p. 292):

- "IN: (instantaneous) indicates the optimization/solution of cutting/packing problem for a given data in one time period.
- CD: (consecutive deterministic) optimization for two or more consecutive time periods when the demand and supply of material for all time periods are known in advance. Material unused at the end of a time period may be used at a later time period. A

specific sub-type of the CD type is an example where some predicted demand lengths from an early time period could also be held until they appear as a demanded length in a later time period.

- CS: (consecutive stochastic) optimization for two or more time periods where only the data for the first period is known with certainty. Probability distributions for future data may be known and material unused at the end of a time period may be used at a later time period".

The abovementioned expansion brings an important aspect into consideration when dealing with cutting problems – the inclusion of time periods in the classification enhances the role of returning leftovers back to stock for the purpose of being reused. Another important aspect of the proposed expansion is based on the assumption that exact demand for future orders is not known in advance (Trkman & Gradišar, 2007). Apart from a few exceptions (Alem, Munari, Arenales, & Ferreira, 2010; Krichagina, Rubio, Taksar, & Wein, 1998; Sculli, 1981), this problem has rarely been discussed in the literature so far.

Due to the increased number of papers published in the field of cutting and packing at the end of the last century, Dyckhoff's classification has been revealed to include some shortcomings, especially in terms of generality. The main deficiencies are (Wäscher et al., 2007):

- not all cutting and packing problems can be classified according to Dyckhoff classification;
- the Dyckhoff classification is inconsistent since the same problem can be classified in several subclasses; and
- each subclass does not necessarily contain homogeneous problems.

A new, improved typology was therefore introduced in which the following classification of cutting and packing problems is proposed (Wäscher et al., 2007):

1. Dimensionality:
   - one-dimensional
   - two-dimensional
   - n-dimensional with $N > 3$

2. Problem type:
   a. output maximization (not enough large objects to meet the demand for all small objects);
   b. input minimization (enough large objects to meet the demand for all small objects).

3. Distribution of order lengths:
   a. the same order lengths

b. weak heterogeneity of distribution of order lengths
c. strong heterogeneity of distribution of order lengths

4. Distribution of item lengths:
   a. one large object
   b. many large objects

5. The shape of order objects:
   a. usual shape
   b. unusual shape

1.1.2 Definition of cutting

As described in the introductory chapter, the first version of the definition of the 1DCSP was introduced in 1939 by Kantorovich, but a more significant step came in 1956 when Paull formulated the problem as it is today (Paull, 1956), and in 1961 when Gilmore and Gomory introduced a linear programming approach (Gilmore & Gomory, 1961) which led to the development of various methods for solving the 1DCSP. At this point, I shall briefly present the general definition of the 1DCSP based on Gilmore and Gomory's (1961) linear programming approach.

It is assumed that there exist L_n (n = 1,…,k) standard lengths in stock. To fulfill the demand there is an unlimited amount of pieces of each standard length in stock. The order consists of N_i pieces of length l_i (i = 1,…,m). The order can be fulfilled if there exists $L_j \geq l_i$. The material that needs to be cut to fill the order represents the total cost, which has to be minimized.

The 1DCSP can be expressed as an integer linear programming problem if we assign $x_1,\ldots,x_n$ variables to each of the possible activities that cut required order lengths from items in stock. The satisfaction of m inequalities must be achieved:

$$a_{i1}x_{i1} + a_{i2}x_2 + \cdots + a_{in}x_n \geq N_i \qquad (i = 1, \ldots, m), \tag{1}$$

where a_{ij} represents the number of pieces of l_i, created by the j-th activity. To minimize the costs the objective function is then formulated as:

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n, \tag{2}$$

where c_i represents the cost of the stock length cut by the i-th activity. Now the introduction of slack variables comes into play and the 1DCSP can be described as:

$$a_{i1}x_{i1} + \cdots + a_{in}x_n - x_{n+1} = N_i \qquad (i = 1, \ldots, m) \text{ and} \tag{3}$$

$$x_j \geq 0 \qquad (j = 1, \ldots, n+m). \tag{4}$$

To obtain a solution, the simplex computational procedure is then conducted. The variables in the basic feasible solution are x_1, $x_2, \ldots, x_n$, P_i is the vector (a_{1i}, a_{2i}, ..., a_{mi}) and c_i is the cost coefficient. A new activity that cuts the required order length from stock length L can be defined as P = (a_1, a_2, ..., a_m). Matrix A consists of the $P_1, \ldots, P_m$ columns which form a basis so that column vector U can satisfy the equation $A \cdot U = P$. An improvement of the solution with the new activity can be used if and only if $C \cdot U > c$. C is the row vector (c_1, $c_2, \ldots, c_m$). Let us say that the row vector $C \cdot A^{-1}$ has coefficients $b_1, \ldots, b_m$. One can conclude that there exists an activity cutting L that can be used over the previous solution if there exist nonnegative integers a_1, $a_2, \ldots, a_m$ that satisfy the following inequalities:

$$L \geq l_1 a_1 + \cdots + l_m a_m \quad \text{and} \tag{5}$$

$$b_1 a_1 + \cdots + b_m a_m > c. \tag{6}$$

The procedure to obtain the solution starts with the determination of m initial activities and associated costs. For each i a stock length L_j is selected. i is an activity defined to be the one cutting the required number of pieces of length l_i from L_j. The cost of the i-th activity is the cost c_j of the stock length L_j. The initial $m+1$ x $m+1$ matrix can be formed hence:

$$\left\| \begin{matrix} 1 & -c_1 & -c_2 & \ldots & -c_m \\ 0 & a_{1,1} & 0 & \ldots & 0 \\ 0 & 0 & a_{2,2} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & -a_{m,m} \end{matrix} \right\| \tag{7}$$

where a_{ii} denotes the number of pieces of length l_i that are cut in the i-th activity from the items in stock at a cost of c_i. B represents the corresponding activities for each of the m columns.

The next step concerns forming $mm+1$ dimensional column vectors $S_1, \ldots, S_2$ that correspond to the slack variables. Given $N = B^{-1} \cdot N$, it is possible to determine whether a better solution can be found. If the first element of $B^{-1} \cdot P$ is not positive, then no improvements in the results can be made and the final solution is obtained.

More information about the abovementioned formulation, including a detailed explanation with some simple examples, can be found in Gilmore and Gomory (1961).

In addition to minimizing the cost, other objectives occur in the 1DCSP, i.e. minimizing the waste, and minimizing or maximizing the total value of produced items. Growth in the number of orders influences the growth of possible cutting patterns and can lead to an inability to calculate the number of possible cutting patterns.

In general, the number of possible patterns grows exponentially as a function of m, the number of orders. As the number of orders increases, it may therefore become impractical to enumerate the possible cutting patterns. Many alternative approaches have thus emerged and will be explained in the second section of this chapter.

When solving the 1DCSP, multiple solutions that result in the same amount of, e.g., trim loss can be found, which impacts the appearance of many problems, such as (de la Banda & Stuckey, 2007; Diegel, Montocchio, Walters, Van Schalkwyk, & Naidoo, 1996; Umetani, Yagiura, & Ibaraki, 2003):

- The minimum pattern count problem: this concerns finding a minimum pattern count solution amongst minimum trim loss solutions. Any instance with n orders can have at least one minimum trim loss solution with $n+1$ patterns.
- The minimum stack problem: this regards the sequence of patterns in such a manner that there are not too many partially fulfilled orders at any point in the procedure.
- The minimum number of knife changes problem: this relates to sequencing and permuting the patterns in a way that minimizes the number of times knives need to be switched. In the literature, this problem is also denoted as the travelling salesman problem.

1.1.3 Definitions of the one-dimensional cutting stock problem and the trim loss

In this subchapter, definitions are given of the standard and general 1DCSP.

First, I present the general 1DCSP that includes the common definition of trim loss as being one of the key factors considered by methods for solving the 1DCSP. The following assumptions should be taken into account (Gradišar et al., 2002):

- there is a sufficient amount of material in stock to fill every customer’s order;
- an exact number of required order lengths needs to be cut into the demanded number of pieces;
- a shortage of material is not possible – if such a situation occurs, it is always possible to adapt the required material; and
- all lengths are considered to be integers or they can be multiplied by a factor that transforms them into integers.

The proposed formulation uses the next notation:

l_i order lengths; $I = 1,\dots, n$,

d_i required number of pieces of order length l_i,

U_j stock lenghts; $j = 1,\dots, m$,

$p_{i,j}$ number of pieces of order length l_i having been cut from stock length j.

The integer programming model is than formulated:

(minimize trim loss that is smaller or equal to max l_i)

$$\min \sum_{j=1}^{m} t_j \quad \text{(minimize trimloss that is smaller or equal to max } l_i\text{)}, \tag{8}$$

s.t.

$$U_j - \sum_{i=1}^{n} l_i p_{i,j} = \delta_j \quad \forall j \quad \text{(knapsack constraint)}, \tag{9}$$

$$\sum_{j=1}^{m} p_{i,j} = d_i \quad \forall i \quad \text{(demand constraint)}, \tag{10}$$

$$\sum_{j=1}^{m} w_j \leq 1 \quad \text{(maximum number of residual lenghts that are larger than max } l_i\text{)}, \tag{11}$$

$$p_{i,j} \geq 0, \text{ integer } \forall i, j, \tag{12}$$

$$t_j \geq 0, \ \forall j, \tag{13}$$

$$\delta_j \geq 0, \text{ integer } \forall j, \tag{14}$$

where the indication whether stock length j is used in the cutting plan is given as:

$$z_j = \begin{cases} 1 & \text{if } p_{i,j} = 0 \quad \forall i, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The indication whether the trim loss relating to stock length j is greater than the longest order lengths is functioned as:

$$w_j = \begin{cases} 1 & \text{if } z_j = 1 \land \delta_j > \max l_i, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

The indication of extent of the trim loss relating to stock length j is as following:

$$t_j = \begin{cases} \delta_j & \text{if } z_j = 1 \land \delta_j \leq \max l_i, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

All residual stock lengths that result from a demand constraint need to be considered. This is enabled by the upper bound for the trim loss being set at between 0 and max l_i. It is important to set the maximum number of residual lengths that are larger than max l_i to be equal to or smaller than 1. In the case of having more than one residual stock length, it is assumed that it is possible to combine those residuals so that only one residual piece in stock is larger than max l_i. Otherwise, it could happen that the trim loss would be avoided but the material in stock would begin to infinitely accumulate (Gradišar et al., 1997).

Due to the differences in stock lengths, the appropriate method for solving the general 1DCSP should be item-oriented. Since the heuristic approach has in several cases proven to be more efficient in terms of time complexity, such a method should be applied to find a solution.

The use of pattern-oriented approach would be more suitable to use when solving standard 1DCSP (Gau & Wäscher, 1995), which is described as follows:

l_i order lengths; $I = 1,\ldots, n$,

d_i required number of pieces of order length l_i,

L_k stock lengths; $k = 1,\ldots, p$.

p represent the amount of different standard stock lengths. Cutting patterns are being cut from different stock lengths and form a cutting plan. Another component of cutting plan are the accompanying frequencies for filling all orders. c represent the cutting pattern and can be, when being cut from standard stock length k, stated as a vector:

$$(a_{1ck}, a_{2ck}, a_{3ck}, \ldots, a_{nck}). \tag{19}$$

The upper vector satisfies the inequations:

$$\sum_{i=1}^{n} l_i a_{ick} \leq L_k, \tag{20}$$

$$a_{ick} \geq 0 \text{ and integer.} \tag{21}$$

The notation acquired for the needs of formulation is:

a_{ick} - signifies the frequency that order length l_i appears in a pattern.

x_{ck} – signifies the number of times that c are being cut from k.

t_k – represent the amount of cutting patterns that are being cut from k.

Thus the formulation of an integer programming model is:

$$\text{minimize} \sum_{k=1}^{p} \sum_{c=1}^{t_k} x_{ck} L_k \quad \text{(minimization of total stock lengths to be cut)}, \tag{22}$$

s.t.

$$\min \sum_{k=1}^{p} \sum_{c=1}^{t_k} a_{ick} x_{ck} \ \geq \ d_i \quad (i = 1, \ldots, n), \tag{23}$$

$$x_{ck} \ \geq 0 \text{ and integer} \quad (c = 1, \ldots, t_k), (k = 1, \ldots, p). \tag{24}$$

The main difference between the general 1DCSP and standard 1DCSP is that the main goal of the latter one is not to minimize the trim loss but to optimize the total stock lengths to be cut.

In practice, sometimes problems occur that cannot be solved using either the item-oriented or pattern-oriented approach. Therefore, a definition of a hybrid 1DCSP was proposed for the purpose of allowing for (Gradišar et al., 1999b):

- the cutting of order lengths into the exact required number of pieces;
- cumulating consecutive residual lengths in one piece, which can be used later; and
- using non-standard stock lengths.

1.2 Exact and heuristic approach

In this subchapter, I will introduce the most significant concepts for solving the 1DCSP. Due to the relatively large diversity of cutting problems, many different approaches have emerged that take account of various aspects (Bischoff & Wäscher, 1995):

- applicability of the research: methods for solving the 1DCSP can be used in various industries such as steel, glass and paper manufacturing.
- the diversity of real-world problems: the structures of cutting and packing problems in the real-world vary according to specific goals or constraints, e.g. the degree of integration into wider planning systems; and
- the complexity of the problems.

Also important is the time needed for an algorithm to prepare the cutting plan. The computational time mostly depends on the algorithm's time complexity, which is depends on the amount of input parameters.

The majority of cutting problems are denoted as NP-complete[7] problems (Bischoff & Wäscher, 1995). The time consumed when solving NP-complete problems increases

[7] The abbreviation NP refers to nondeterministic polynomial time.

exponentially in conjunction with the problem size and therefore bigger problems cannot be solved exactly. Depending on the computational time, the methods for solving the 1DCSP can be classified in two groups:

- Exact methods: these provide the best possible solutions but are only useful for small orders due to the exponential growth of computational time. The solutions found have exponential complexity.
- Heuristic methods: these do not ensure the optimal solution but they do find one within a reasonable time. Problems can thus be solved in polynomial time.

A problem is NP-hard if it is possible to transform any other NP problem into the original problem with a polynomial algorithm. A problem is NP-complete if it is placed in the set of NP problems and also in the set of NP-hard problems. If an efficient polynomial algorithm to solve one NP-complete problem were to be found, it would be possible to use it to solve all other NP-complete problems (Schrijver, 1998). Such an algorithm has not yet been found.

Although NP-hard problems cannot be optimally solved, various techniques can be applied to solve several large-scale instances in negligible time (Pisinger, 1995):

- branch-and-bound;
- dynamic programming;
- state space relaxation; and
- preprocessing.

NP-problems represent a set of solutions that can be defined with use of a nondeterministic algorithm in limited polynomial time. Typical NP-hard problems are knapsack problems (Caprara, Carvalho, Lodi, & Woeginger, 2013).

The knapsack problem was first mentioned already in 1896 (Mathews, 1896). Later, the number of solutions to the problem started to grow and in 1974 Horowitz and Sahni introduced a new branch and search algorithm that was significantly faster than previous algorithms (Horowitz & Sahni, 1974). Many new methods followed afterwards.

Algorithms for solving knapsack problems are some of the most widely used algorithms for solving bin-packing problems and problems related to the tree data structure (Skiena, 1999).

A knapsack problem is defined as the optimal packing of a knapsack, where the value of the content should be maximal, the empty volume minimal and the weight of the content as small as possible. Mathematical it can be defined in a following way (Scheithauer, 2008):

- p … knapsack volume;

- b_i … object volume i, where $i \in I = \{1, \dots, m\}$; $0 < b_i \le p$;
- m … number of objects of volume b_i, which are packed in the knapsack;
- c_i ... value of the object i, $c_i \ge 0$;
- z_i … sum of usage of object i at packing, where $z_i \in \mathbb{Z}_+$.

The goal is to find such set of objects that they not exceed p in the total volume and at the same time have the maximal value:

$$f(p) = \max\left\{\sum_{i \in I} c_i z_i : \sum_{i \in I} b_i z_i \le p, z_i \in \mathbb{Z}_+, \mathrm{i} \in \mathrm{I}\right\}. \quad (25)$$

As already mentioned above there exist many algorithm for solving the knapsack problem. Hereinafter is presented the main algorithm for solving hard knapsack problems (Pisinger, 1995).

It is assumed that the items are ordered according to non-increasing efficiencies $e_j = p_j / w_j$, thus

$$e_j \ge e_i \text{when } i < j. \quad (26)$$

Various orderings could satisfy the abovementioned statement but for the purpose of this presentation it is assumed that one particular ordering has been selected. For any two indices s, t, where $1 \le s \le t \le n$ the set of the partial vector $X_{s,t}$ is defined:

$$X_{s,t} = \{(x_s, \dots, x_t) \in \{0,1\}^{t-s+1}\}. \quad (27)$$

For a partial vector $\mathrm{x_i} \in X_{s,t}$ the defined corresponding profit sum π is notated as $\pi(x_i) = \pi_i = \sum_{j=s}^{t} p_j x_j$. The weight sum μ is defined as $\mu(x_i) = \mu_i = \sum_{j=s}^{t} w_j x_{i,j}$. An optimal solution is found by:

$$z = \max\ \{\pi\ (x_i) \colon \mu\ (x_i) \le c\}. \quad (28)$$

In order to enumerate the set $X_{1,n}$ it is assumed, that n is a power of two. Initially n sets of $X_{i,i}$ are constructed, each consisting of two partial vectors:

$$X_{i,i} = \{(0), (1)\} \quad (i = 1, \dots, n). \quad (29)$$

Follows the multiplication of the sets two by two, obtaining

$$X_{1,2}, X_{3,4}, \dots, X_{n-1,n} \quad (30)$$

as the Cartesian product of

$$X_{i,i+1} = X_{i,i} \times X_{i+1,i+1} \quad (i = 1,3, \dots, n-1). \quad (31)$$

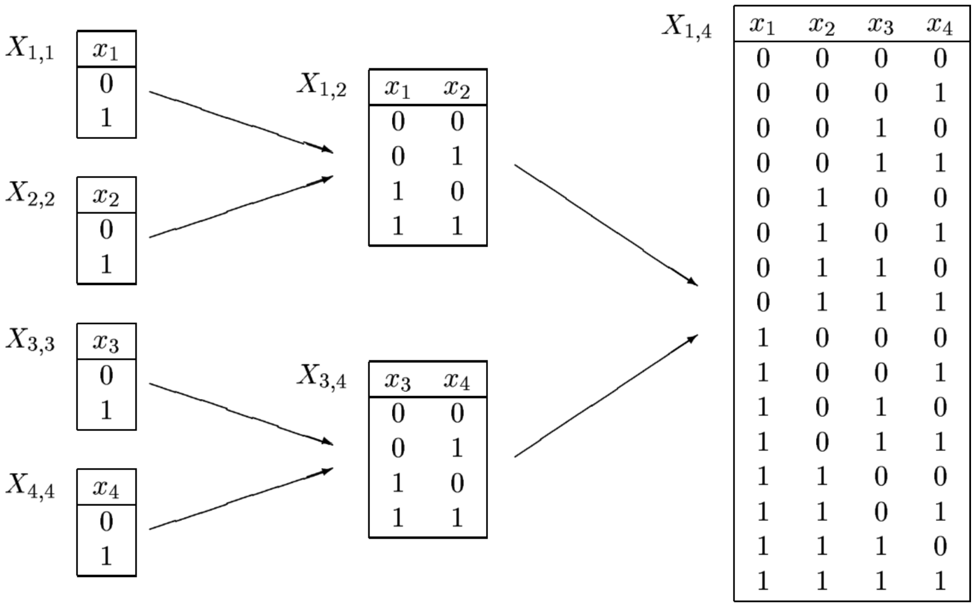
The procedure is continued in such way until the final two sets $X_{1,n/2}$ and $X_{n/2+1,n}$ are reached, which are multiplied in a similar way.

For example the following data instance is given:

J	1	2	3	4
p_j	7	5	6	3
w_j	4	5	7	4

where n = 4 and c = 13. The running of the main algorithm is presented in Figure 16.

Figure 16: Main algorithm for solving knapsack problem.



Source: Pisinger, 1995.

All such problems involve a subset of demanded items that have to be selected in such a way that the corresponding profit sum is maximized and the capacity of the knapsack is not exceeded. In accordance with the distributions of items in the knapsack many variations of knapsack problems exist, such as (Pisinger, 1995):

- the 0–1 knapsack problem: each item can only be chosen once;
- bounded knapsack problem: the amount of each item type is bounded;
- multiple-choice knapsack problem: individual items can be selected from separate classes; and
- multi-constrained knapsack problem: a general integer programming problem where coefficients are positive.

Knapsack problems are widely applicable in practice, for example in investment management, tourism, cargo loading, budget control, financial management, cryptography and cutting (Pisinger, 1995).

In terms of cutting, such examples are ideal patterns that are defined as a combination of demanded items of any lengths which are cut from a certain bar in such a way that the trim loss is minimized. A complementary problem is all of the remaining patterns that are not ideal (Cesar, 2013).

Table 7 presents the difference between polynomial and exponential algorithms. It shows the change in the problem size that can be solved using a faster CPU[8]. The size increases rapidly for algorithms with polynomial time complexity and slowly for algorithms with exponential time complexity.

Table 7: Influence of faster computer on polynomial and exponential algorithms.

Function of time complexity	The problem size with the nowadays computer	100-times faster computer	1000-times faster computer
n	N_1	100 N_1	1000 N_1
n^2	N_2	10 N_2	31,6 N_2
n^3	N_3	4,64 N_3	10 N_3
n^5	N_4	2,5 N_4	3,98 N_4
2^n	N_5	N_5 + 6,64	N_5 + 9,97
3^n	N_6	N_6 + 4,19	N_6 + 6,29

Source: Garey & Johnson, 1979.

As is evident from Table 7, it is not very likely to expect exact solutions for large problems in the future. Accordingly, the exact methods are only appropriate for solving problems of a small size. Problems of a bigger size can be solved using heuristic algorithms.

The main disadvantage of the heuristic methods used for solving large problems with low time and computational complexity is that they do not provide an optimal solution. Instead, they provide a nearly optimal solution within a reasonable time. In practice, providing a solution in a short time often has a crucial meaning for an efficient business. The desired speed of designing the cutting process can thus depend on various factors, such as (Erjavec, 2011):

- the agreed delivery period;
- the occupancy of cutting machines; or
- the number of orders.

[8] The abbreviation CPU refers to central processing unit.

Companies tend to operate at the lowest possible costs and therefore a solution obtained in a short time period and with a higher trim loss outweighs the optimal solution obtained in a longer time period with a lower trim loss (Erjavec, 2011). This is one of the main reasons that heuristic methods are more commonly used in real-world situations.

According to Dyckhoff, methods for solving the 1DCSP can be divided into two groups with regard to the basic principle that was used. These groups are (Dyckhoff, 1990):

- Pattern-oriented methods: cutting patterns and the frequency of each pattern are determined using different methods. Most pattern-oriented methods are based on the algorithm introduced by Gilmore and Gomory (1961; 1963). These methods initially construct patterns and they then assign large and small items to some of those patterns. Methods based on a pattern-oriented approach are only useful in the case of all items in stock being of the same lengths or a few different standard lengths.
- Item-oriented methods: in this group of methods patterns are not determined. Instead, the cutting plan is prepared for each individual item in stock. Methods based on an item-oriented approach are generally used as they can be applied to both standard and non-standard item lengths in stock. These methods promptly assign items to objects.

In the case of standard stock lengths, the pattern-oriented methods are more flexible and more appropriate for use in solving large problems. In the case of non-standard stock lengths, only item-oriented methods come into play (Trkman & Gradišar, 2010).

Some of the main distinctions between item-oriented and pattern-oriented approaches are presented in Table 8.

Table 8: Solution approaches.

Item-oriented		**Pattern oriented**	
Exact methods	Approximation algorithms	One-pattern	Several patterns
E.g. branch and bound, dynamic programming	Bin packing algorithms	Knapsack algorithms, various methods in more dimensions	LP-based and general heuristics

Source: Dyckhoff, 1990.

The classification presented in Table 8 is not totally whetted. For example, the type of pattern-oriented methods, denoted as one pattern, can be considered only in the case of one large object. Regarding the one-dimensional type of problem, such methods are based on combinatorial algorithms, i.e. those for solving the knapsack problem. In the presented typology, the emphasis is on combinatorial perspectives of cutting and packing problems and not on the geometric properties of objects (Dyckhoff, 1990).

When using methods based on the pattern-oriented approach to find the optimal solution, the main problem lies in the large number of possible patterns that can be used for cutting the individual item. A middle-sized problem is selected for demonstration purposes. There are 50 different order lengths and the individual item is cut into 5 pieces (5 different or same item lengths are obtained from each item in stock). The number of possible patterns for one item can be calculated using the equation for a combination with a repetition:

$$C_{n+r-1}^{r} = \binom{n+r-1}{r}, \qquad (32)$$

where *n* stands for the number of different order lengths and *r* for the number of order lengths that are used in the cutting of an individual item. Therefore, 3,162,510 different patterns can be used for an individual item. It is usually necessary to use more than one item so the number of all possible combinations of patterns and items that are going to be used in the cutting is significantly higher. It is possible to conclude that every method that attempts to obtain a solution by testing all possible patterns is doomed to fail, except in cases with very small problems (Trkman & Gradišar, 2010).

Due to the abovementioned grounds, methods based on a pattern-oriented approach do not test all possible patterns when solving large problems but only test some of them instead. The key question that arises regards the selection of patterns to be tested among all possible patterns. Hence, the most appropriate patterns are determined using various methods. The frequency of each individual pattern is determined after. As a result, the certainty of obtaining the optimal solution depends on the method's performance and the type of problem (Trkman & Gradišar, 2010).

When using methods based on an item-oriented approach each item that has to be cut is treated separately. A cutting plan is determined for each item. Where all items have different lengths such an approach is the only realistic one. Use of an item-oriented approach is in some cases also suitable when solving problems where all items have the same lengths (Trkman & Gradišar, 2010).

For the approaches presented above (item-oriented and pattern-oriented) both groups of methods, exact and heuristic, can be used to solve problems. Exact methods are based on algorithms that in a finite number of steps provide an optimal solution to the problem with a certain probability. This is also a major advantage of exact methods. The biggest disadvantage lies in the exponential growth of time needed to determine a solution when the problem size is growing due to the NP-hardness of most cutting problems. To obtain a solution the exact methods use various techniques, such as (Trkman & Gradišar, 2010):

- linear programming;
- branch and bound; and
- dynamic programming.

The heuristic method is a procedure that provides nearly optimal solutions to the optimization problem. Those methods are generally more suitable for solving bigger problems and mostly become applicable in cases where an algorithm for the optimal solution has not yet been found or it has been too computationally demanding for solving bigger problems (Eiselt & Sandblom, 2000).

Another important question arises regarding how to determine the quality of the method. A method is considered to be good quality if the obtained solution has a sufficient quality. That is, if it differs from the optimal solution by less than some acceptable value that is determined by various criteria. The proof that the obtained solution deviates by less than such a value can be mathematical or statistical on the basis of a large number of generated and solved problems (Hinxman, 1980). Mathematical proof considers the maximal possible deviation of the heuristic solution from the optimal one. Statistical proof considers a statistical analysis of the obtained solutions and their comparison with the optimal ones. Such an analysis can also point out the suitability of the exact or heuristic method for solving different types of problems.

Another disadvantage of heuristic methods is that they are usually quite specific and only suitable for solving an individual problem. A heuristic method that provides a fine solution for one specific problem given certain goals and assumptions will probably not be suitable for another problem, even if those problems are apparently similar (Trkman & Gradišar, 2010).

1.2.1 Types of exact methods

With respect to the limitations on the processing capacity of a computer there are various methods for solving the 1DCSP in an exact way. The most commonly used are the branch-and-bound method, techniques of linear programming and dynamic programming. Since descriptions of the last two approaches are presented in other parts of my doctoral dissertation, here I will only present the branch-and-bound approach.

Since the number of feasible points in bigger problems is relatively high, it is impossible to verify all of them. Therefore, the branch-and-bound method is based on a smart selection of points that are to be verified. The elementary principle is that first the linear program of the problem is solved to obtain a solution that consists of non-integer values of the variables (Trkman & Gradišar, 2010). If the values are an integer that means there is an optimal solution of the problem since the optimal solution of the linear relaxation of the problem, where all the values are integers, is also an optimal solution to the integer problem (Winston & Goldberg, 2004).

If the values are not integers, the solution of integer relaxation presents the lower bound for the solution of the minimization problem and the upper bound for the maximization problem. Therefore, it is possible to deduce that the obtained solution is not optimal (Trkman & Gradišar, 2010).

The next step is to divide a problem into two parts by branching it with the addition of a new limitation. Then one of the variables that initially does not have an integer value is selected. For example, if the value of variable y in the solution of the relaxation of the linear program equals 4.532, then the limitation $y \geq 5$ is added in the first sub-problem and $y \leq 4$ in the other. These two sub-problems do not have any common point. However, together they contain all possible integer solutions to the problem but not the optimal solution to the relaxation of the linear program that is not integer (Trkman, 2008). Created sub-problems are noted as a tree. Each individual solved sub-problem is denoted as a node. Further, one of the sub-problems is selected and the solution of its relaxation is found. This process continues until all of the nodes have been taken into account. The solution that remains is optimal. One of the key advantages of this method is that the elimination of one node excludes several feasible points that cannot provide an optimal solution.

Two relevant questions are: which nodes should be branched first, and according to which variables should the branching be carried out? In practice, there are several options. One is the "last in first out" principle where we initially process one part of the tree and acquire a possible solution and then proceed with solving the remaining tree. Another option is to always solve that part of the sub-problem which has the best value of the objective function by jumping from one part of the tree to another. By using the branch-and-bound method it is possible to find an optimal solution to the problem, but in practice application of the method usually depends on the NP-hardness of the problem; the harder a problem, the longer the time it will take to obtain a solution.

1.2.2 Types of heuristic methods

Many heuristic methods can be found in the literature. Some of the most important ones are presented in the next subchapter. At this point, only a brief categorization of heuristic methods into groups is given.

Various types of heuristic methods can be identified according to the way they operate to obtain a solution to the problem (Hinxman, 1980; Trkman & Gradišar, 2010).

- **State-space search**: partial solutions to the problem are taken as nodes in a graph. Then the best path from the initial state (a complete unsolved problem) to the final state (a final solution to the problem) is found. Individual nodes and links between them are created dynamically.
- **Problem reduction:** the main problem is divided into several smaller sub-problems. Each sub-problem is then solved separately (if those sub-problems are small enough it is also possible to apply exact methods). The final solution is presented as the union of individual solutions.

- **Cut-off**: if an exact method is based on attaining the optimal solution with iterations, it can be transformed into a heuristic method by introducing a condition when to terminate the iterations. The termination of iterations is possible when the solution obtained differs from the optimal one by less than some set value or when the costs and computational time exceed a certain, but still acceptable limit.
- **Aspiration level**: if a heuristic method is based on acquiring various possible solutions to a problem, it is possible to accept the first solution that meets some criterion that is set in advance. Such an approach is useful when a nearly optimal solution is needed instead of the optimal one.
- **Repeated exhaustion repetition**: heuristic methods of this type initially obtain some adequate pattern with a low trim loss. This pattern is then repeated as often as possible until the overproduction of an individual order length occurs. Then follows the exclusion of the last pattern usage and the determination of a new pattern that is used again as often as possible. This procedure continues until the complete fulfillment of the order has been reached.
- **Sampling**: this approach is similar as in the aspiration level. Several possible solutions are identified. Next, for each of the possible solutions the value of the objective function is calculated. Then the point with the best value of the objective function is selected.

Another classification of heuristic methods can be found in the literature. Beraldi, Bruni, and Conforti (2009) proposed a categorization of at least three different groups of heuristic approaches. These are, in particular:

- sequential heuristic procedures (SHP);
- methods that are based on linear programming; and
- metaheuristic methods, e.g. tabu search, genetic algorithms, evolution algorithms etc.

The main disadvantage of methods that use the heuristic approach is that they do not find the optimal solution on the account of the shorter computing time. But, in practice, finding the solution in the shortest possible time is often crucially important for a successful business performance. The desired speed of designing a cutting plan can therefore depend on various factors such as the date of the delivery of material, the occupancy of the cutting machines and the number of orders. Due to the smaller total cost, companies are willing to make a compromise between the amount of the trim loss and how long it takes to prepare the cutting plan (Erjavec, 2011).

1.2.3 A Combination of exact and heuristic methods

Sometimes researchers combine the exact and heuristic methods to obtain better solutions. Such an example is found in Gradišar and Trkman (2005) where the authors presented a new approach that combines the SHP and branch-and-bound algorithm. This combination enables almost optimal solutions and keeps the low time complexity. First, the bigger part of the problem is solved using the heuristic method. Second, smaller sub-problems are being solved using the exact method. The presented combination of exact and heuristic methods provides better results than can be obtained using only the heuristic approach.

1.3 Overview of the methods

In this section, I present some of the best known methods for solving the 1DCSP. In addition to the presented methods, the literature provides information about many other methods, although a detailed description of all methods would exceed the framework of this doctoral dissertation.

The best known heuristic method that obtains solution using a sampling approach was presented by Gilmore and Gomory (1961). The aim is to find a cutting plan with minimal costs. Costs are presented as the sum of costs of all items used. The stock consists of an unlimited number of pieces, each with a standard length. In the order that has to be filled there is a certain number of pieces of known lengths. Due to the unlimited amount of material in stock, the order can be met in all cases except if the longest ordered item is longer than the longest item in stock. The main problem is represented by the large number of possible patterns. The presented method deals with it in a way so that in each phase of the simplex method, when a new pattern is needed, not all possible patterns are reviewed. Instead, the relevant pattern is found by solving an auxiliary problem. Such a problem is commonly called the knapsack problem.

The knapsack problem can be explained using an example of a hitch-hiker (Martello & Toth, 1990). He has to fill his knapsack up with objects that will maximize his comfort. He can choose from among many objects but cannot select all of them due to the limited size of the knapsack. The mathematical formulation of the problem requires numbering the objects from 1 to n and introducing a vector of binary variables x_j (j = 1,...,n) with respect to:

$$x_j = \begin{cases} 1 \text{ if object } j \text{ is selected;} \\ 0 \text{ otherwise.} \end{cases} \tag{33}$$

Suppose that a measure of comfort is p_i given by object j, w_j is its size and c is the size of the knapsack. The knapsack problem is how to select, from among all binary vectors x satisfying the constraint

$$\sum_{j=1}^{n} w_j x_j \leq c, \tag{34}$$

the one that maximizes the objective function:

$$\sum_{j=1}^{n} p_j x_j. \quad (35)$$

The 1DCSP is a common application of the knapsack problem in various forms in different industries. An example can be found in the production of fiber optic cables where the problem is how to cut lengths of cable to meet a customer's demands while extracting the greatest value from each length of cable (Bartholdi III, 2008).

A similar problem can be found in the paper industry. A paper mill produces huge rolls of paper. The dimensions of these rolls are determined by the manufacturing process. With respect to filling a customer's orders these rolls are sliced into smaller rolls. Gilmore and Gomory (1963) proposed a modification of their method that is adjusted to this specific problem. They made small changes in the problem formulation and in the algorithm for the purpose of obtaining a solution in a shorter time.

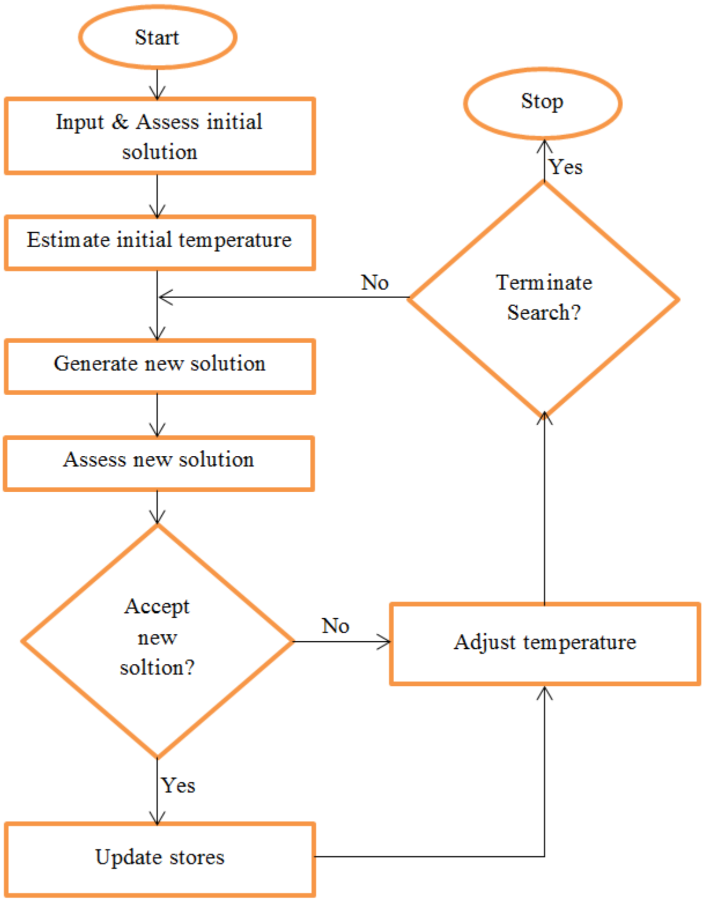
One of the first techniques for solving optimization and combinatorial problems such as the traveling salesman problem and the three machine scheduling problem was the branch-and-bound method (Little, 2012). Branch-and-bound algorithms are the most efficient for solving non-convex problems since they assure a lower and upper bound of the optimal objective value. Their main disadvantage is that they generally consume a lot of time due to the exponential growth of the required effort with the problem size (Boyd & Mattingley, 2007).

Another technique for solving complex optimization problems is called simulated annealing. It uses a random-search approach. The name comes from its similarity to a process where a metal cools and freezes into a minimum energy crystalline structure. The approach that is used to find an extreme of the function may be compared to a bouncing ball that bounces over mountains from valley to valley. The high temperature assists the ball to bounce very high – presumably over any mountain – so any of the valleys can be reached if there are enough bounces. If the temperature decreases, then the ball does not have the possibility to bounce so high anymore and is therefore stuck in relatively small arrays of valleys. In that way, an acceptance distribution is determined. It depends on the temperature. If the level of cooling is firmly controlled, then a simulated annealing can find a global optimum of the function but it consumes an infinite amount of time, which is also the main disadvantage of the presented technique. Other weaknesses refer to the non-clear compromise among the quality of the solution and the consumed time and fine-tuning of the parameters for a specific problem. The chief advantage is that simulated annealing is a flexible and robust approach that enables complex non-linear models with a lot of data and constraints to be tackled. The following elements are essential for conducting a simulated annealing procedure (Busetti, 2003):

- a representation of possible solutions;
- a generator of random changes in solutions;
- a means of evaluating the problem functions; and
- an annealing schedule – an initial temperature and rules for lowering it as the search progresses.

The structure of a simulated annealing is presented in Figure 17.

Figure 17: The structure of simulated annealing algorithm.



Source: Busetti, 2003.

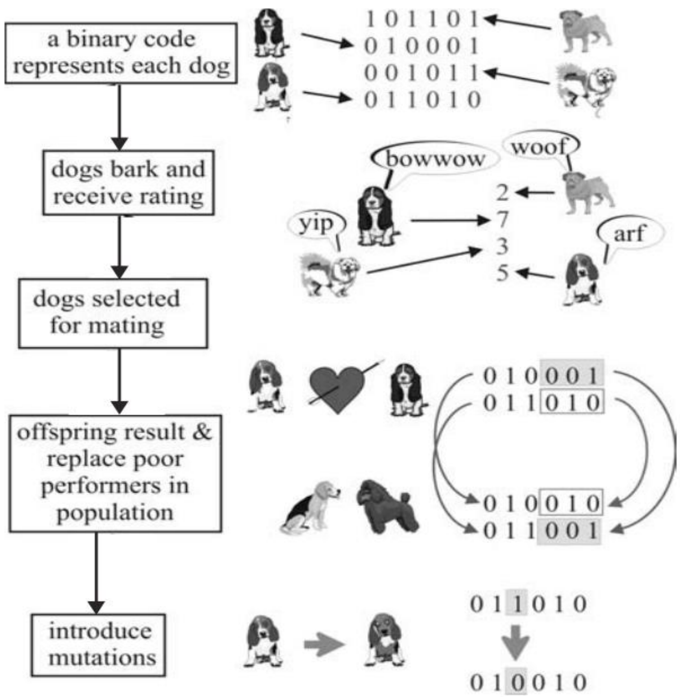
Later, many methods for solving the 1DCSP evolved that differ from Gilmore and Gomory's approach. One of the possible techniques is called a genetic algorithm (GA) which was introduced in 1975 (Holland, 1992). The core idea is that short, low-order and highly fit schemata can be recombined to form higher-order schemata and complete strings with high fitness (Rothlauf, 2006). The GA's performance is derived from the real-life situation that was first described by Darwin in 1859. Three basic principles are (Darwin, 1859):

- a population consists of individuals with different properties and abilities. There is also an upper limit on the number of individuals in a population;

- nature creates new individuals with properties that are similar to the existing individuals; and
- natural selection more often selects promising individuals for reproduction.

The analogy between biological genetics and the GA is shown in Figure 18.

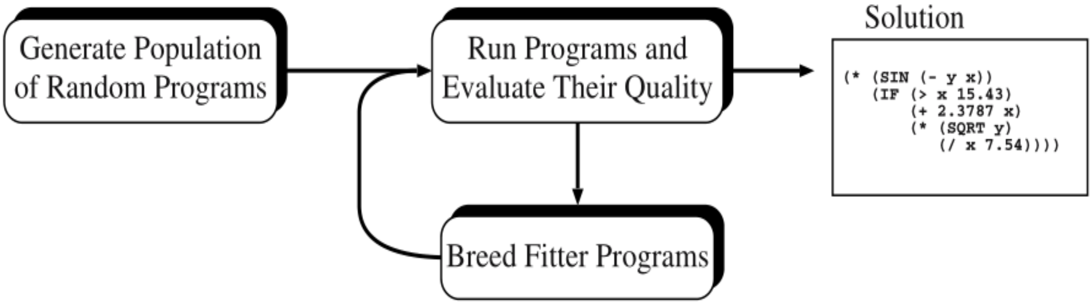
Figure 18: Analogy between biological genetics and GA.



Source: Haupt & Haupt, 2004.

The properties and abilities of an individual are determined by the genotype. Different genotypes allow that individuals with different properties exist. The number of individuals from a population is limited due to finite resources. When the number of individuals surpasses the standing upper limit, some individuals are detached from the population. Individuals mutate over generations. Descendants receive some properties of their parents. The genotype is changed by mutation. The genetic information of the parents is also recombined. Another important determination of the GA is the fitness value of an individual. Highly fit individuals are enabled to create more descendants than those who have a lower fitness value. After some generations, the poorer individuals are detached from the population. Hence they no longer have a chance of creating descendants with similar properties. In such a way the overall fitness of a population improves over generations. Accordingly, in principle the GA first randomly creates and evaluates an initial population. It then iteratively creates new generations by recombining the selected individuals with a high fitness value and applying mutation to the descendants (Rothlauf, 2006). The demonstrative main loop of the genetic algorithm is presented in Figure 19.

Figure 19: Main loop of the genetic algorithm.



Source: Poli, Langdon, McPhee & Koza, 2007.

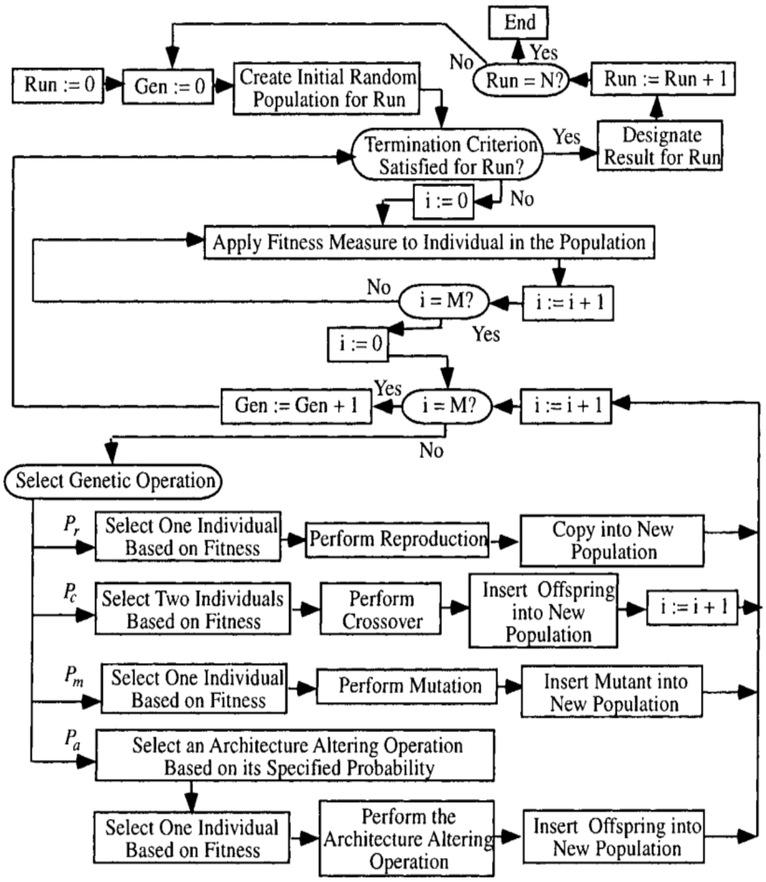
The main steps of the GA include representation, initializing the population, selection, recombination and mutation. In the representation step, the genetic programs are stated as syntax trees. Variables and constraints that are denoted as terminals are represented by leaves of the tree and the arithmetic operators are presented as internal nodes (functions). Together, they create a primitive set of the system. When the system is more complex, the trees can be joined under a special root node and become denoted as branches (Poli et al., 2007).

The phase of initializing the population is comparable with other evolutionary techniques where the individuals in the initial population are randomly generated. To create an initial population, many methods can be used such as the Full and Grow method or the Ramped half-and-half method. Common to all of the methods for the creation of the initial population is that individuals are formed with respect to a pre-determined maximum depth. Further generation of the tree depends on the selected method or a combination of them. The main limitation of such a creation of the population lies in the difficulty of controlling the statistical distributions of main properties like the size and shape of the generated tree. Those limitations can be omitted using various modifications that have been developed for specific situations. If some characteristics of the desired solution are known, the initial population does not need to be completely randomly generated. Instead, the trees with those characteristics can be used when seeding the initial population. It is also possible that such seeds are preliminary created with another GA. If the initial population is not entirely randomly generated, then it is necessary that a situation is prevented where the second generation would be dominated by offspring of a single or a few seeds. It is generally considered that a variety of population should be observed since the significant mixing of different initial trees is necessary for ensuring that the GA works efficiently (Poli et al., 2007).

As mentioned above, the phase of selection consists of genetic operators that are applied to individuals who have a higher probability of being selected on the basis of their fitness value. This means that an individual with a higher fitness value will have more children than those with a lower value (Poli et al., 2007). In this phase, two chromosomes are selected from the pool of chromosomes and produce the new offspring. This procedure runs until the stop criterion is met. Many methods for selecting an individual can be found in the literature, such as pairing from the top-to-bottom method, random pairing and weighted random pairing, but the one that is most widely used is called tournament selection. By this method, a small subgroup of chromosomes is randomly chosen from the mating pool and the chromosome with the highest value in this subgroup becomes a parent. Since many different sets of parents are created, the configuration of the next generation varies for each selection scheme (Haupt & Haupt, 2004).

Recombination and mutation reflects the main differences between the GA and other evolutionary algorithms. Those are the operators of crossover and mutation. In the sub-tree crossover process, a sub-tree crossover randomly selects a crossover point in each parent tree. Next, the offspring are created by substituting the sub-tree entrenched at the crossover point in a copy of the first parent with a copy of the sub-tree entrenched at the crossover point in the second parent. Many other methods for performing a crossover can be found in the literature, such as a one-point crossover, common region, a size-fair crossover, a uniform crossover and a context-preserving crossover. The decision on which one to select depends on the needs of the specific problem. The next step is a mutation, which is performed by randomly selecting a mutation point in a tree and substituting a sub-tree that is rooted there with a randomly generated sub-tree. The process of mutation depends on the mutation methods selected (Poli et al., 2007). After the crossover and mutation phase, the value of the offspring and mutated chromosomes is evaluated. An illustrative procedure of the GA is presented in Figure 20.

Figure 20: Flowchart of GA.



Source: Koza & Poli, 2005.

Another widely used optimization method is called dynamic programming. It transforms a complex problem into a subset of problems that are simpler. It provides a common framework for solving various types of problems. It usually follows one of two approaches (Alba et al., 2009):

- Top-down approach: initially the problem is divided into sub-problems which are then solved separately. The algorithm memorizes the mapping between sub-problems and solutions and returns to the stored solution if the problem has been preliminarily solved.
- Bottom-up approach: on the basis of complexity first the sub-problem are solved and stored. The solution to a complex problem is then built on the solutions to the sub-problems.

The main advantage of dynamic programming is the efficiency when the search tree is exponential so sub-problems that have the same properties are constantly arising. The chief limitation lies in the capacity of computational power. If the range of sub-problems is too large to fit into a computer’s memory, the dynamic programming method will not find a solution (Alba et al., 2009). The method of dynamic programming is more often used when solving two-dimensional cutting problems and since my doctoral dissertation addresses the 1DCSP I will not explain dynamic programming in detail. More information on this topic can be found, e.g. in Powell (2007).

A commonly used heuristic method is a greedy algorithm, which was first mentioned in the literature in 1971 (Edmonds, 1971). It finds the solution step by step and always selects the next step, which provides the most noticeable and direct value (Dasgupta, Papadimitriou, & Vazirani, 2006). It is used for solving various optimization problems including cutting problems. Although a greedy algorithm always takes decisions that are the best possible ones at that moment, it is not necessary that the final solution will be optimal. The performance of the algorithm depends on its design. In general, first the optimization problem is cast as one in which the decision is made and one sub-problem is left to solve. Next, the proof has to be provided that there is always an optimal solution to original problem that makes the greedy decision optimal (the greedy decision is always safe this way). Finally, the optimal sub-structure is demonstrated by showing that the remaining sub-problem has the possibility of combining the optimal solution to the sub-problem with the greedy decision so the algorithm reaches an optimal solution to the original problem. This indicates one of the biggest differences with dynamic programming, which is that the decision is based on the current situation and the sub-problems that remain are solved later (Cormen, Leiserson, Rivest, & Stein, 2009).

In the late 1990s, a new meta-heuristic approach was proposed by Glover (1986). It is called a tabu search and is still widely used. The name comes from the word “tabu” and is associated with the transmission of a social memory that is subject to adjustment over time.

The most important elements of a tabu search are adaptive memory and responsive exploration. The first one enables an efficient search for the solution space that is based on random processes that implement sampling. Such a method implies a memoryless design that includes the genetic, annealing and branch-and-bound approaches. The second element, responsive exploration, is based on the selection of bad choices since those choices produce better information than a good random choice. It combines the examination of solution characteristics and the exploration of new promising solutions. There are four fundamental dimensions of the memory structure in a tabu search, namely (Glover & Laguna, 1997):

- quality;
- influence;
- recency; and
- frequency.

The next most important elements are intensification and diversification. Intensification regards a modification of the decision rules to provide a good solution. Diversification considers the selection of the best solutions to define the best attributes for newly obtained solutions (Glover & Laguna, 1997).

1.4 Overview of existing algorithms

In this subchapter I present the best known methods for the 1DCSP. Some of them can also be used for solving the 1DCSPUL and therefore they will be explained in the second chapter and will not be mentioned here.

The first efficient algorithm for obtaining a heuristic solution was proposed by Gilmore and Gomory in 1961 and later modified for cutting paper. However, since this was already explained in the previous subchapter I will not describe it again at this point.

In the literature it is possible to find several solutions to the 1DCSP that are based on the GA or its modifications.

Vahrenkamp (1996) proposed a modified GA that only takes efficient patterns into consideration, where the trim loss is smaller than some predetermined threshold. Ragsdale and Zobel (2004) introduced a combination of the GA with other heuristic methods for solving the problem where the basic requirement is that individual orders have to be cut sequentially for the purpose of being rapidly and comprehensively filled. Another example can be found in the construction industry (Shahin & Salem, 2004) where the GA was used whereas other heuristic methods were inappropriate to apply due to various shortcomings. Quiza Sardiñas, Rivas Santana, and Alfonso Brindis (2006) presented a multi-objective optimization technique that was based on the GA with the purpose of optimizing the cutting parameters in turning processes. A similar solution with the addition of production cost under a set of machining constraints can be found in Yildiz and Ozturk (2006). In the clothing industry, a hybrid GA was proposed by Yueng and Tang (2003) where a cutting problem was transformed into a permutation problem that could be effectively solved with a combination of the GA and a novel heuristic algorithm. The number of objects and setups of cutting patterns was minimized with a symbiotic GA that enabled the generation of combinations of cutting patterns and their frequency at the same time (Golfeto, Moretti, & de Salles Neto, 2009).

In 1990 Goulimis presented an approach that provides an optimal solution for the 1DCSP. First, the possible cutting patterns are enumerated. Then the related integer program is solved by a combination of cutting plans and the branch-and-bound algorithm. The presented approach was experimentally tested with real data from a board mill. The results show that the average savings in material were up to 2.5% better than according to previously used methods. The number of cutting patterns and blade setups was also significantly reduced (Goulimis, 1990).

The modified integer round-up property of the linear integer minimization problem for the 1DCSP has been demonstrated by proposing a new non-polynomial heuristic procedure for solving the problem (Scheithauer & Terno, 1995). Antonio et al. (1999) introduced two methods based on dynamic programming. The purpose was to solve a large spectrum of industrial cutting stock problems. The main contribution is the reduced computational burden by keeping only those states that seem to be the most cost-effective in each phase of the dynamic process. The first method, which was used in the sales department, where time was more important than the excellence of a solution, favored the computational time on account of the quality of the result. The second method favored the quality of the solution (Antonio, Chauvet, Chu, & Proth, 1999).

For the purposes of the fashion industry a mixed integer programming model was presented. The algorithm hunts for an optimal range of cutting patterns. Then it allocates a combination of pieces that need to be cut to those patterns (Degraeve & Vandebroek, 1998).

Sometimes a combination of various approaches works best for solving the problem. One example is a combination of column generation and the branch-and-bound algorithm (Vance, Barnhart, Johnson, & Nemhauser, 1994). A formulation of the problem is based on general integer variables and it is thus not limited to be binary. The linear relaxation enables a strong lower bound and the cutting pattern is disintegrated into single arc variables (de Carvalho, 1998).

Degraeve and Schrage (1999) offered an interested insight into the cutting stock problem since they presented a method termed column generation which enables the use of fractional amounts of patterns when solving a problem. They proposed an extension of the Degraeve column generation procedure within the branch-and-bound method and introduced a relatively general method that can also be used for solving various situations from practice. Such situations can consist of upper bounds on the amount by which the demand could be over-satisfied, a bound on the maximum leftover acceptable in each pattern and a limit on the maximum quantity of knives in cutting patterns. They tested their method on real industrial data sets. The presented method was later improved in 2003 by reducing an average solution factor by more than 300 so that bigger problems could be solved. The improvement consisted of integration of a hybrid simplex method to find the LP relaxation and of adding a branching scheme, which centers on the enumeration tree to solve the residual problem. Some additional rules, such as a pruning rule and use of the dominance rule, were also taken into account. They also proved the efficiency of the algorithm for solving bin-packing problems (Degraeve & Peeters, 2003).

The large number of cutting patterns usually makes the cutting stock problem impossible to solve and so Suliman (2001) introduced a new approach. If the linear programming construction of a problem is free of integer variables, then the effect of the amount of cutting patterns would be diminished. The solution is a simple pattern generation procedure that uses a search tree to develop the pattern generation method.

A generalization of a combination of Chvatal-Gomory cutting planes with column generation was presented in 2002. The presented heuristic method was developed for solving the 1DCSP with various bar lengths in stock. Although the solution is based on the heuristic approach, the optimal solution is found in approximately 90% of test cases (Belov & Scheithauer, 2002).

In 2002 Holthaus introduced an algorithm that minimizes cost and was built for solving an integer 1DCSP with different types of standard stock lengths. It was based on the column-generation technique and combined some existing heuristic and exact methods (Holthaus, 2002). Another approach from the abovementioned researcher was introduced a year later. The focus was on possible savings in the total cost of material that could be achieved by using a pool with two or more different stock lengths (Holthaus, 2003).

A new model that resolved the non-linearity in the 1DCSP between pattern variables and pattern run lengths was introduced by implementing a novel use of 0-1 variables with some constraints (Johnston & Sadinlija, 2004).

A combination of the exact and heuristic methods can lead to nearly optimal solutions also when solving very big problems. Such a method for solving the 1DCSP was introduced in 2005. It enabled all stock lengths to be different and combined a sequential heuristic procedure with the branch-and-bound approach. First, the problem is solved heuristically and a good solution is obtained. Then the smaller, critical part of the problem is solved with the exact method. Such a combination gives very good solutions in a relatively short time (Gradišar & Trkman, 2005).

The general tabu search approach was improved to efficiently solve a 1DCSP that is considered as a sequence problem. A varying mixed objective function was introduced that considers incentive and the total trim loss and therefore expands the possibilities for finding a good solution. The proposed approach was experimentally tested on real data. The solutions obtained were compared with the results of methods previously used for that specific case. The modified tabu search proved to be more efficient (Yang, Sung, & Weng, 2006).

In 2006, Cui, Chen and Wu outlined a method adjusted to the needs of a silicon steel circular blanks company. The dynamic programming technique was applied. First, the sheet lengths are grouped in ascending order and the best ones are selected. The optimal solution for an individual length is then used as an initial solution for the next length. Next, the optimal strip layouts on segments are determined. The optimal segment lengths are then obtained using implicit enumeration (Cui, Chen, & Wu, 2006).

In 2006, Belov and Scheithauer presented a method that is able to tackle the 1DCSP and a two-stage, two-dimensional guillotine constrained cutting problem. They enhanced a branch-and-price scheme with general-purpose cutting plans. The solution is primarily obtained for LP relaxation and later improved with mixed-integer cuts. Unfortunately, practical experiments revealed that the method is more efficient when applied to a two-dimensional cutting problem than for the 1DCSP (Belov & Scheithauer, 2006).

A hybrid approach of metaheuristics and linear programming for solving 1DCSP where the quantity of stock rolls should be minimized due to increasing costs of switching among cutting patterns is proposed by Umetani, Yagiura, and Ibaraki (2006). The algorithm consists of two phases. For increasing the performance the local search was combined with linear programming technique.

The problem of having a large number of different cutting patterns when solving the 1DCSP was tackled by a sequential heuristic method that minimized the input of material. Additional functionality was added in order to restrict the number of open stacks to an arbitrary limit. To evaluate the solution with regard to several objectives, the Pareto criterion was used (Belov & Scheithauer, 2007).

Lee proposed an integer linear programming method based on local search heuristics. He holistically integrated the main problems and sub-problems of price-driven, pattern-oriented methods and presented a unified model which generates patterns in situ. The emphasis is on improvement of the integer linear programming over continuous linear programming (Lee, 2007).

An interesting combined method can also be found in Alves and Valério de Carvalho (2008), where the authors proposed a new branch-and-price-and-cut algorithm based on the exact approach. They researched the use of valid dual inequalities within the previous algorithms and showed that the column generation could be enhanced in all branching tree nodes with the use of those inequalities. They proved that inequalities can be stretched to the whole branch-and-bound tree to improve the conjunction of the proposed branch-and-price-and-cut algorithm.

Another sequential heuristic method was introduced by Cui, Zhao, Yang and Yu (2008). The focus was on pattern reduction. The newly presented algorithm first creates a cutting pattern from a chosen sub-set of unassigned items, then it defines the pattern usage frequency, removes those items that are given to the cutting pattern from the sub-set and repeats the procedure. It stops when all the items have been assigned. Experiments have shown that this algorithm obtains a better solution than other methods that are based on pattern reduction.

Pattern reduction is a widely studied objective when developing a method for solving the 1DCSP. Another method was introduced by Cerqueira and Yanasse (2009). Their solution was based on the evaluation of a group of patterns based on their frequencies.

A two-stage method for pattern generation and cutting plan determination is based on the integer linear programming models. First, a heuristic procedure is applied to identify all possible cutting patterns and to minimize the trim loss. Then the second integer linear programming model is applied and a cutting plan is formed. The model took various factors into account, such as (Aktin & Özdemir, 2009):

- material inputs;
- the number of setups;
- labor hours;
- overtime;
- demand requirements;
- material availability;
- regular and overtime availability; and
- due date constraints.

A rarely used technique for solving the 1DCSP is ant colony optimization. An improved model where the authors modified parts encoding, solution path, state transition probability rules and phenomena updating rules consisted of 16 steps and proved to be suitable for solving combinatorial optimization problems (Yang, Li, Huang, Tan, & Zhou, 2009).

An interesting aspect of the 1DCSP was considered by Reinertsen and Vossen (2010) where the orders must be met within a given period of time. They developed an algorithm that considers the due dates of orders. The test data were taken from practice so the aggregation of orders, multiple stock lengths and cutting of different types of material on the same machine were also studied.

When solving the 1DCSP trim loss minimization is usually the main goal, but not always. In 1990, Wäscher presented a linear programming method to simultaneously tackle several objectives, such as costs of overgrading, material, warehouse and trim loss (Wäscher, 1990).

Another linear programming method was introduced five years later and was intended for solving two-stage cutting stock problems. The outcomes of the first stage are intermediate products, which are then cut into final items in the second stage. In addition to minimizing the trim loss costs, the presented method also reduces the number of set-ups and minimizes the quantity of cutting patterns and intermediate products. The linear program is solved by a column generation technique (Valério de Carvalho & Guimarães Rodrigues, 1995).

Many methods from the literature are based on modifications of Gilmory's algorithm. Such a case can be found in Soeiro Ferreira, Antonio Neves, and Fonseca Castro (1990). The authors were dealing with a roll cutting problem in the iron and steel industry that included two phases. The first phase covered the cutting of raw material into intermediate products that were cut to the final product in the second phase. The developed method was based on automatic sequential searching.

In 1995 a project was presented where the problem involved the distribution of wooden products with different quality necessities that have to be cut from wooden panels. The main trouble was the very short time limit since the cutting plan should be made in 2 seconds, before the material is sent to the machine. The problem was solved with a combination of Lagrangean relaxation and sub-gradient optimization (Rönnqvist, 1995).

Methods for optimizing the sequence of cutting patterns evolved in 1995. The aim was to minimize the maximum queue of incompletely cut orders. The first method was intended for solving more specific cases and was based on the lower bound. The second method used an implicit exhaustive search procedure and could be used more generally (Yuen & Richardson, 1995).

Another method for determining the cutting sequence was presented in 1998 for the purpose of minimizing the queue of partially cut orders. The presented mathematical model included three phases. First, the greedy algorithm is applied and a relatively good starting solution is obtained. This solution is then improved by a tabu search in the second phase. In the third phase, an optimal solution is reached by implication of the enumeration procedure that uses the previously gained solutions (Faggioli & Bentivoglio, 1998).

The special nature of the problem was noted by Wagner (1999). He took into account trim loss minimization, stock usage and ending inventory levels. The method enabled non-integer results, while the main emphasis of his work was on rounding up those numbers. Because the traditional rounding-up procedures are not effective, he proposed a specially developed GA to solve the integer programming problem formulation.

An algorithm that was based on a hierarchical approach pursued three objectives (Vasko, Newhart, & Stott Jr, 1999):

- to satisfy the customer's order as much as possible – the number of uncut orders should be as small as possible;
- to minimize overgrading – diminishing the use of too good material when performing cutting. Such a situation occurs when a customer wants to have material of a lower quality. If the company fails to satisfy the customer's requirements, the material can be of better quality, but this at the expense of the company; and
- to minimize the trim loss.

Various objectives that should be taken into account when formulating the objective function arise from real-life situations. Minimization of the cutting time was the case in the method developed by Chu and Antonio (1999). They introduced approximation algorithms for various intentions such as a fast response to a specific customer requirement and the provision of a production plan for the next period. The proposed method is similar to the one published in Antonio et al. (1999) where the authors proposed an approach for solving large problems. The presented methods were based on dynamic programming and reduced the computational complexity by only keeping those situations that appear to be the most cost effective.

Sometimes other objectives should also be taken in to account, such as (Venkateswarlu, 2001):

- trim loss: expressed as percentage of the total material used;
- overproduction: items that are cut from the material in stock oversize the order due to the repetition of cutting patterns. It is expressed as a percentage of all items cut;
- average stock: the length of material in stock after filling a customer’s order;
- average number of different lengths of material in stock: the average of different lengths that remain in stock after cutting; and
- average number of bars in stock: the number of bars that remain in stock after filling a customer’s order.

Some other authors have also included other criteria. In 1998, an interesting method evolved for the purposes of the paper-converting industry. The aim was to minimize the bi-linear cost function when producing required paper sets from larger raw paper reels. The costs consisted of trim loss, the warehousing of partially used reels of paper and changes in the machine and blade settings. The first step of the method concerns a nonlinear procedure for generating all possible cutting patterns. In a later step, a solution is obtained with mixed integer linear programming (Westerlund, Harjunkoski, & Isaksson, 1998).

In the clothing industry, various layers of commodities must be put on a cutting table before the cutting is undertaken. This is a timewasting process that results in high setup costs, which in relation to the trim loss costs represent the total production costs. Hence, the method for minimizing such setups and at the same time generating a little or no trim loss was introduced. The proposed method uses a mixed integer programming model for searching for an optimal set of cutting patterns (Degraeve & Vandebroek, 1998).

An attention-grabbing approach can also be found in Foerster and Wascher (2000). The goal was to find a plan with a small or minimal quantity of cutting patterns instead of minimizing material costs since the quantity of cutting patterns needed to fulfill a certain set of orders can be vital for the capacity load that can be achieved for cutting equipment. The proposed method has two steps. First, the minimal input plan is generated and then the quantity of patterns is reduced. The authors focused on the development of a new method for the second step. The method may be regarded as a generalization of other methods solving such a problem.

The goal of minimizing the total cost of production was followed by Menon and Schrage (2002). The method sought to tackle the problem of allocating customers' orders to machines when producing paper. The model was formulated as a dual-angular integer problem. The authors proposed an approach that solved decomposable dual-angular integer programs on the basis of sub-problems' boundaries and applied it to solve a problem from practice. The impact of symmetry, which fortifies the difficulty of solving the decomposable integer programs, was therefore meaningfully reduced.

An interesting method which took the maximization of profit into account, based on the revenues from sales, the costs of the material, the costs of changing the cutting pattern and the costs of the discarded material, was presented by Schilling and Georgiadis (2002). The solution was grounded on a mixed integer linear programming model that was resolved with the use of standard techniques.

A method for specifying the order of cutting on various machines was introduced to solve real problems in the construction industry. The sequencing algorithm searches for the processing sequence for starting the cutting patterns arising from an integer solution and exploits the relations among orders in patterns (Armbruster, 2002).

In 2002, Zak presented two papers dealing with a multistage 1DCSP where the cutting process consists of various following steps. Since at the end of each stage (except the last stage) the result is an intermediate product, the aim was to minimize the total material used in order to meet the customer's demands. In the first paper, he developed a special procedure for row and column generation which uses a revised simplex algorithm to efficiently solve a non-linear knapsack problem (Zak, 2002a). In the second paper, he presented a generalization of the Gilmore and Gomory column generation method in order to solve large-scale linear programming models (Zak, 2002b).

As mentioned in the previous section, evolutionary algorithms (EA) present an efficient possibility for solving many optimization problems. One type of evolutionary algorithm is evolutionary programming (EA) which uses a mutation as the search operator but, unlike the GA, does not conduct the crossover phase. A method that uses the EP algorithm was introduced by Liang, Yao, Newton, and Hoffman (2002). Apart from trim loss minimization, the objectives that were considered were minimization of the quantity of stock and the quantity of partially finished items. They proposed two new mutation operators and experimentally showed that application of the EP can result in a better solution in comparison with the GA when solving the 1DCSP.

A heuristic method that followed the goal of minimizing the number of cutting patterns used in the cutting plan was introduced in 2003. The formulation of the model was based on the assumption that the relocation of the blades, as needed upon each pattern change, represents the highest cost rather than the trim loss, although the trim loss is not overlooked. The limitation about the highest quantity of trim loss in each cutting pattern is built up in the model (Umetani et al., 2003).

Another method drawn from the paper industry was presented in 2004. The costs of changes in cutting patterns and of resetting the blades were evaluated. The method was based on simulated annealing where a general regression neural network is skilled using available data to create a model that comes close to the response surface in the feasible domain. The candidates generated by the simulated annealing are then verified by estimating the objective functions (Yen, Wong, & Jang, 2004).

Another objective that is taken into account is order priority. Such an approach was studied, for example, by Ghodsi and Sassani (2005a). The purpose was to have all demanded items ready within a given time limit. The method was grounded on an adaptive fuzzy ranking technique to solve problems in real time. The orders were prioritized on the basis of the complexity of fulfillment. In the same year, those authors presented another method regarded as complementary to the one described above. The purpose was to minimize the trim loss when the items in stock are of a random size and quality. The solution enables real-time solutions and allows it to be implemented on-line (Ghodsi & Sassani, 2005b).

An interesting example can be found in Gramani and França (2006) where the minimization takes the trim loss, storage costs and setup cost into account. They considered the production planning for various periods. They combined the stock and lot-sizing problem and formulated a mathematical model based on an analogy with the network shortest path problem. The results indicated that the solution is 28% better than that obtained by solving each of the abovementioned problems separately.

Minimization of the trim loss would not be appropriate if the material in stock is infinitely long, such as in Hajizadeh and Lee (2007). Therefore, the minimization considered the sum of cutting time and pattern changing time to meet the given demand. The goal was to realize the better effectiveness of cutting machines so the whole production could have a higher performance. The model included linear formulations for presented problems using a binary expansion of the quantity of pieces of altered types in a pattern. The time available for a pattern change was formulated as a linear function of the quantity of knives. The experiments suggested that the presented approach could lead to 400% better results than those from any existing methods.

To solve the 1DCSP with limited multiple stock lengths several heuristics have been developed. Significant work has been done by Poldi and Arenales (2009). They developed a range of methods for constructive and residual heuristics based on the first fit decreasing approach and greedy procedure. They made a comparison with some other methods such as the branch-and-bound procedure developed by Belov and Scheithauer (2002).

In the paper industry it is also possible to find a method that aims to fulfill a customer's orders in the terms of the quality of material. Orders are satisfied by transforming paper mills into a high quantity of sheeted products so that huge reels of different paper grades are produced. These reels are later cut into smaller rolls and sold or sheeted into finished products. Reuse of the material is not possible. The binary linear model uses two approaches for solving. The first one is branch-and-bound based on column generation and the second is the heuristic method for marginal cost minimization (Chauhan et al., 2008).

When solving the 1DCSP an evolutionary algorithm can be very efficient. This was shown for a type of problem where different stock lengths are available but their amount is limited. Every step of EA development was developed in a way so as to be suitable for a particular problem. In the representation phase, each gene was considered as a cutting pattern and the quantity of times it is cut. Accordingly, the individuals were formed by a matrix. The initial population was generated taking the computational time and diversity into account. For the selection process the random greedy approach was used. The cooperation and self-adoption phase were based on the use of a repeated exhaustion reduction heuristic. The considered strategy for population replacement was steady-state-oriented. The stopping criterion was defined by a parameter that was signified by the maximum quantity of iterations to be made. The objective function represented the fitness function. It was the minimization of the trim loss in the described case (Araujo, Constantino, & Poldi, 2011). The presented EA method was then compared with some other heuristic methods (Belov & Scheithauer, 2002; Poldi & Arenales, 2009).

It is possible to find many papers in the literature that deal with a comparison of various methods for solving the 1DCSP. In Vance (1998), a comparison is made for two branch-and-price approaches that differ in the way that the integer programming of the column generation is formulated. In the first, the algorithm tackles the problem using 0-1 integer variables. The other uses universal integer variables. Both algorithms were tested on real-life data and the same computer. Better results were obtained with the algorithm that was based on a more compact column generation formulation since the amount of relaxations of the linear program is smaller and is thus less difficult to solve. One important finding from this study was that a compact formulation is more suitable for solving complex problems due to the increased quantity of columns. On the other hand, it cannot be used for cases where the stock items are not of standard lengths because the generalizations used to define the compact formulations do not apply. When the formulation of column generation is less compact, this influences the branch-and-bound symmetry and it is therefore not applicable to some situations (Vance, 1998).

Another evaluation of several methods was made by Gradišar et al. (2002). Until that paper, a general comparison of the various algorithms was impossible since they took many different factors into account. The authors suggested that the trim loss is the most important one and they therefore proposed a common definition of trim loss by introducing a General One-Dimensional Cutting Stock Problem type. Then two heuristic methods (CUT and COLA) were evaluated.

A comparison of three heuristic methods by trim loss was presented by Bingul and Oysu (2005). The authors evaluated three different levels of the complexity of a problem: easy, medium and hard. The approximate method was compared with the GA and simulated annealing, both with an improved bottom-left. The conclusion was that the GA performs better than the other two when solving complex problems, whereas for easy and medium-sized problems the approximate algorithm is the most suitable.

Similar methods to those described above may be found for solving two-dimensional, three-dimensional or four-dimensional cutting problems. The description of such methods is not presented here since that would go beyond the purpose of my doctoral dissertation.

1.5 The role of cutting in companies

Cutting activities in companies are closely related to elements of supply chain management (SCM), which is one of the most important areas in operational research. Many definitions of SCM can be found in the literature (Tan, 2001) but, for the purpose of this doctoral dissertation, SCM can be identified as the integration of various functional areas within an organization to enhance the flow of goods from immediate strategic suppliers through the manufacturing and distribution chain to the end user (Houlihan, 1988).

Cesar (2013) further defines the activity of cutting as a result of the planning and realization of all activities in the supply chain in the most efficient way. When emplacing the cutting process in the SCM of a company, the whole activity can be divided into three phases[9] (Cesar, 2013):

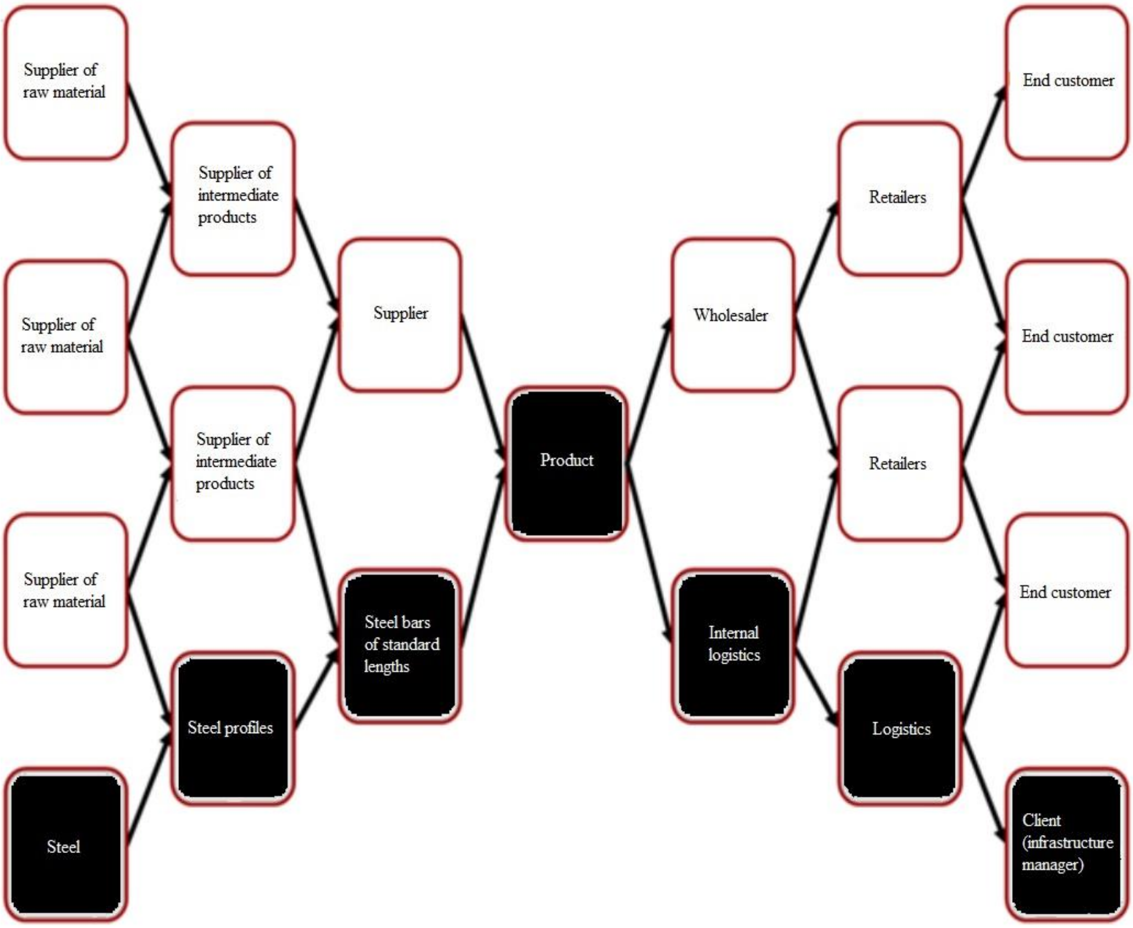
- *From raw material to product*
  This phase includes the raw material suppliers, suppliers of intermediate products and suppliers of the final product. The cutting process starts with preparation of the material needed for molding steel. The next step is to design profiles out of structural steel, which is then reworked into bars of various standard lengths. This step is usually not conducted between different suppliers but between different production units of the company.

[9] The example is taken from the steel industry.

- *Product*
  The product is derived by cutting standard bars into the number and lengths of items required in an order.
- *From product to buyer*
  Ordered items of proper lengths must be transferred from the cutting machine to the appropriate spots and prepared for their transfer to the buyer. This step is usually implemented as part of the internal logistics of the company.

The phases described above are illustrated in Figure 21, which is based on work of Wieland and Wallenburg (2011) and modified by Cesar (2013) to include individual activities related to the cutting process. The modification was undertaken on the basis of visits and personal communications with appropriate managers in the steel enterprises Štore Steel, Acroni, and Metal Ravne. Those activities are marked in black rectangles.

Figure 21: Illustration of the role of cutting activities in SCM.



Source: Wieland & Wallenburg, 2011; modified by Cesar, 2013.

The abovementioned description of the cutting activity reflects the situation in the steel industry, but does not necessary hold for other industries where the number of participants in the SCM can differ.

Beside the presented example, cutting activities can also be found in other areas, e.g.:

- transportation (Baskan, Haldenbilen, & Ceylan, 2012; Norros, 1995; Sen et al., 2012);
- stock management (Tomat, Gradišar, & Štiglic, 2013); and
- location and distribution (Iori, Salazar-González, & Vigo, 2007; Zachariadis, Tarantilis, & Kiranoudis, 2012).

2 THE ONE-DIMENSIONAL CUTTING STOCK PROBLEM WITH USABLE LEFTOVER

This chapter presents a special type of the 1DCSP in which leftovers longer than a predefined threshold are returned back to stock so that they can be used again in future orders (1DCSPUL).

First I describe the concept and present the current state of affairs, and then an overview of the methods for solving the 1DCSPUL from the literature is given. At the end of this chapter, I explain *r* and provide the definition of a low *r*.

2.1 Description of the concept and general formulation

Optimizing the one-dimensional cutting of material is based on the use of stock to fill an order with the aim of minimizing the company's costs. One of the main goals when optimizing the cutting of one-dimensional material is minimizing leftover material because it represents a loss that companies wish to reduce. If the leftover material is of adequate size, it can be returned to the stock and used for filling future orders (Alfieri et al., 2007; Cherri et al., 2009). This is known as the 1DCSPUL.

Various definitions of the 1DCSPUL can be found in the literature, but one of the most precise and comprehensive definition was given by Cherri and is therefore used for the doctoral dissertation (Cherri et al., 2009, p. 898):

"A set of pieces (items) must be produced by cutting large units (objects) of standard sizes (objects bought from suppliers) or non-standard (objects that are leftovers of previous cuts). The demand of the items and the availability of the objects are given. Demand must be met by cutting the available objects such that the leftovers are "small" (...) or "sufficiently large" (...) to return to stock, but in a reduced number."

The first attempts to evaluate usable leftovers in the 1DCSP can be found in the work of Kantorovich (1960), but this concept was barely referenced. The second occurrence of usable leftovers dates to the 1970s (Brown, 1971). The next researcher that tackled the issue of usable leftovers was Roodman (1986). He took two objectives into account: trim loss minimization and aggregation of the leftovers in several cutting patterns so they could be used again in the future.

However; the general definition was proposed by Scheithauer (1991). The presentation summarized below is based on his paper. Residual lengths are considered using a linear programming model. The order consists of pieces T_i of lengths l_i and demand b_i ($i = 1,\dots, m$). Stock lengths sizes are denoted as L_q ($q = 1,\dots, p$). It is presumed that every stock length contains a value v_q. Some residual lengths R_k of sizes r_k contain a positive valuation w_k ($k = 1,\dots, r$). The goal is to find the cutting patterns and corresponding amount in a way that the customer demand is satisfied and the costs are minimal. The costs are represented by the costs of stock lengths consumed minus the value of residual lengths attained. Another assumption is that there is always a sufficient amount of items in stock to fill an order.

Given the above the presentation of a feasible cutting pattern of stock length L_q by an (m+r)-dimensional non-negative integer vecor $a = (a_1,\ldots, a_{m+r})^T$ is of the following form:

$$\sum_{i=1}^{m} l_i a_l + \sum_{k=1}^{r} r_k a_{m+k} \leq L_q. \tag{36}$$

The j-th cutting pattern, $j = 1,\ldots, n_q$, of the q-th stock length, $q = 1,\ldots, p$, is denoted by $a^{jq} = (a_1^{jq}, \ldots, a_{m+r}^{jq})^T$. The amount of admissible cutting patterns of lengths L_q is represented by n_q. The frequency of cutting pattern a^{jq} is being cut is given by the variable x_{jq}. The proposed mathematical model is of the next form:

$$z = \sum_{q=1}^{p} v_q \sum_{j=1}^{n_q} x_{jq} - \sum\nolimits_{q=1}^{p} \sum\nolimits_{j=1}^{n_q} \sum\nolimits_{k=1}^{r} w_k a_{m+k}^{jq} x_{jk} \rightarrow min \tag{37}$$

$$= \sum_{q=1}^{p} \sum_{j=1}^{n_q} \left(v_q - \sum_{k=1}^{r} w_k a_{m+k}^{jq} \right) x_{jq} \rightarrow min. \tag{38}$$

Subject to

$$\sum_{q=1}^{p} \sum_{j=1}^{n_q} a_i^{jq} x_{jq} \geq b_i \quad (i = 1, \ldots, m), \tag{39}$$

$$x_{jq} \geq 0, \quad j = 1, \ldots, n_q \quad (q = 1, \ldots, p), \tag{40}$$

$$x_{jq} \text{ integer}, \quad j = 1, \ldots, n_q \quad (q = 1, \ldots, p). \tag{41}$$

Scheithauer suggests applying revised simplex method and column generation technique. A feasible basis B of the relaxation problem is formed by a set of m linear independent columns vectors. The vector of simplex multipliers is to be $d = (d_1, \ldots, d_m)^T$. Hence the column generation problem is

$$c^* = \min \left\{ \min_{q=1,\ldots,p} z_q,\ \min_{i=1,\ldots,m} d_i \right\}, \tag{42}$$

where

$$z_q = \min\{v_q - \sum_{k=1}^{r} v_k a_{m+k}^q - \sum_{i=1}^{m} d_i a_i^q \mid \sum_{i=1}^{m} l_i a_i^q + \sum_{k=1}^{r} r_k a_{m+k}^q \leq L_q, \tag{43}$$

$$a_i^q \geq 0, \text{integer}, i = 1, \dots, m + r\}.$$

Obtaining a new column and continuation of the simplex algorithm is repeated until $c^* \geq 0$, when the optimal basis B is found. For computation of z_q first we need to find an optimal solution to the knapsack problem

$$f(l) = \max\left\{\sum_{i=1}^{m} d_i a_i \,\middle|\, \sum_{i=1}^{m} i_i a_i \leq 1, a_i \geq 0, \text{ integer } (i = 1, \dots, m)\right\}. \tag{44}$$

Hence

$$z_q = v_q - \max\{\sum_{k=1}^{r} w_k a_{m+k} + f\left(L_q - \sum_{k=1}^{r} r_k a_{m+k}\right) \mid \sum_{k=1}^{r} r_k a_{m+k} \leq L_q, \tag{45}$$

$$a_{m+k} \geq 0, \text{integer}, k = 1, \dots, r\}.$$

Abovestated forumula value presents properties of residual lenghts but to prevent useless cuts the following assumptions have to be met:

- $w_k / r_k \leq v_q / L_q$ for all k and q with $r_k \leq L_q$,
- $w_k / r_k \leq v_t / r_t$ for all k and t with $r_k \leq r_t$.
- if r_k and r_t are residuals than $r_k + r_t$ is a residual as well.

In accordance with those assumptions an optimal solution for z_q exists with

$$\sum_{k=1}^{r} a_{m+k} \leq 1. \tag{46}$$

So the forumlation for z_q can be reduced to

$$z_q = v_q - \max\{f(L_q, w_k + f(L_q - r_k | r_k \leq L_q, k = 1, \dots r\} \quad (q = 1, \dots p). \tag{47}$$

Given the abovementioned the following origination is formed as

$$c^* = \min\left\{\min_{i=1,\dots,m} d_i, \min_{q=1,\dots,p} v_q - \max\{f(L_q, w_q + f(L_q - r_q) | r_k \leq L_q, k = 1, \dots, r\}\right\}. \tag{48}$$

Additional explanation and some trivial examples can be found in Scheithauer (1991).

A more sophisticated definition was introduced by Sinuany-Stern and Weiner (1994). Two objectives were considered when proposing the formulation. The first regards trim loss minimization and the second regards organization cutting such that the maximum quantity of leftovers is joined in the last bar to be used again. An algorithm was developed to deal with this problem. Using a heuristic approach, it is also suitable for solving large problems. For the formulation of the problem next notation is used:

L – length of the standard bars.

N – number of bars used.

n_i – the number of pieces of length l_i required, i = 1,…, I.

X_{ij} – the number of pieces of length l_i cut from bar j.

Only standard bars are available in stock.

The proposed formulation of the problem is than expressed as a linear integer problem. N and X_{ij} must be chosen in relation to:

$$min\left\{N * L - \sum_{i=1}^{I} l_i n_i\right\}, \tag{49}$$

$$max\left\{L - \sum_{i=1}^{I} l_i X_{iN}\right\}. \tag{50}$$

Subject to:

$$\sum_{i=1}^{I} l_j X_{ij} \leq L \quad (j = 1, \dots, N), \tag{51}$$

$$\sum_{j=1}^{N} X_{ij} \geq n_i \quad (i = 1, \dots, I), \tag{52}$$

$$X_{ij} \geq 0, \tag{53}$$

$$N, X_{ij} \text{ integer}. \tag{54}$$

The upper formulation holds for having no leftover. Than the number of bars is

$$N_0 = \sum n_i l_i / L. \tag{55}$$

In case N_0 is not an integer than the lower bound need to be considered:

$$N_0 = \left[\frac{\sum_{i=1}^{I} n_i l_i}{L}\right] + 1. \tag{56}$$

For the formulation of utilizing leftovers is it assumed that here are K residuals from previous orders, L_k, $k = N+1,\dots,N+K$, which are needed to use. So the previous constraints can be reformulated as

$$\sum_{i=1}^{I} l_j X_{ij+k} \leq L_k \quad (K = 1, \dots, K), \tag{57}$$

$$\sum_{j=1}^{N+K} X_{ij} \geq n_i \quad (i = 1, \dots, I). \tag{58}$$

So the new upper bound is

$$N_0 = \left[\left(\sum_{i=1}^{I} l_i n_i - \sum_{k=1}^{K} L_k \right) \Big/ L \right] + 1. \tag{59}$$

The above presented formulation and an algorithm were developed for the actual use in practice. It was only used to solve small problems but it served as a basis for the development of various more complex algorithms for solving 1DCSPUL.

The most recent general mathematical notation and formulation was presented by Cherri, Arenales, Yanasse, Poldi, and Vianna (2013):

- i: item type;
- j: cutting pattern;
- k: object type;
- K: number of types of objects in stock;
- L_k: length of object type k;
- e_k: availability of object type k in stock;
- c_k: unit cost of object type k;
- m: number of items;
- l_i: length of item type i;
- d_i: demand for item type i;
- N_k: total number of cutting patterns in relation to object type k;
- δ_k: threshold length for a retail from object type k;
- α_{ijk}: number of item type I in cutting pattern j of object type k;
- x_{jk}: number of objects type k cut in relation to cutting patterns j; and
- p_{ik}: the number of item types I cut from object type k.

Using the terminology stated above the leftover s_k from object type k is as follows:

$$s_k = L_k - \sum_{i=1}^{m} l_i p_{ik}. \tag{60}$$

2.2 Overview of methods for solving 1DCSPUL

As explained in Chapter 1 of the doctoral dissertation, optimal solutions usually result from the application of exact methods (Alves & Valério de Carvalho, 2008), but they are only useful for small orders (the results of exact methods are the best possible solutions, but in the case of larger orders the computing time is too long due to the insufficient computing power of computers). The majority of methods developed are thus based on a heuristic approach (Vahrenkamp, 1996; Cui, 2005). In contrast to exact methods, heuristic solutions do not ensure an optimal solution, but they do find one within a reasonable time. The focus of this subchapter is primarily to present heuristic methods because these are widely used across many different industries.

COLA was the first method that could generally be used to solve the 1DCSPUL on a large scale. It is based on a sequential heuristic procedure and is classified as item-oriented. It was developed by Gradišar et al. (1997) for the clothing industry. COLA is used in Chapter 5 to propose an improvement of testing solutions to 1DCSPUL with the simulation method, and so a detailed explanation is provided below.

The problem was defined as a bicriterial multidimensional knapsack problem with side constraints. To transform an exact algorithm into a heuristic one with polynomial complexity, a basic limitation was introduced. It included processing one item at a time and it stopped when all of the order requirements were met. The algorithm performed a step-by-step procedure. Because it was developed for the clothing industry, the items were denoted as rolls. When the procedure started, all of the rolls were classified as unprocessed, but were reduced by one roll for each step. The number of processed rolls therefore rose by one roll for each step. The algorithm was based on a sequential heuristic procedure. It solved a series of knapsack problems. To find the first optimal combination, the knapsack algorithm generated a sequence of vectors in lexicographically decreasing order. A PC version of the method was released for public use as well as the algorithm. The procedure that was developed was based on the following assumptions (Gradišar et al., 1997):

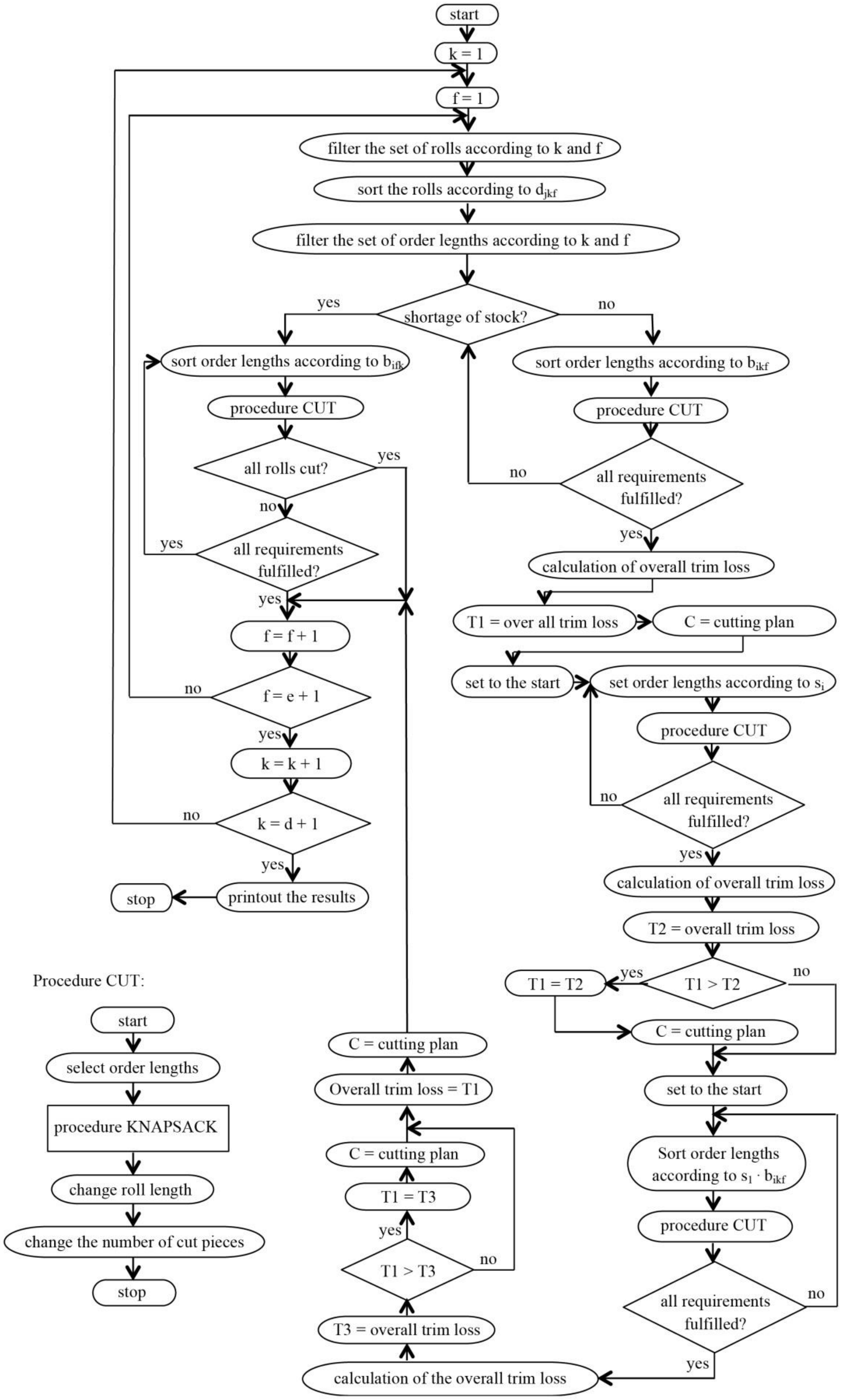
- It is easier to find a good solution by choosing a roll that is selected from the largest possible set of different rolls;
- It is easier to find a good solution for the selected roll if it is selected from the largest possible set of feasible solutions. This is possible in the following cases:
    - The roll is the longest possible;

- The order lengths are the shortest possible;
- The number of different order lengths and the number of pieces are the largest possible;

- It is easier to find a good solution for the selected roll if the differences in order lengths are the largest possible.

The COLA flowchart is presented in Figure 22. A more precise formulation can be found in Gradišar et al. (1997).

Figure 22: Flowchart of COLA.



Source: Gradišar et al., 1997.

Method COLA was than improved by CUT. It combined the pattern-oriented linear program method and the item-oriented SHP with the purpose of cutting order lengths in The COLA method was then improved with CUT. It combined the pattern-oriented linear program method and the item-oriented SHP in order to cut order lengths into the exact number of pieces required and to accumulate consecutive residual lengths into one piece that could be returned back to stock and used again later in the cutting process. The initial algorithm performed two steps. In the first step, the problem was transformed into a problem such that it could be solved with linear program methods. The second step concerned patterns. The algorithm withdraws from the solution all of the patterns that include more pieces of order lengths than required. This was the first part of the final solution. Due to the withdrawn patterns, a new problem was created and was solved with a sequential heuristic procedure, which resulted in the second part of the solution. The final solution consisted of both parts described above (Gradišar et al., 1999a). CUT was also released to the public and its algorithm is still freely available.

Based on CUT, the C-CUT method was developed by Gradišar and Trkman (2005). It combined a sequential heuristic procedure and a branch-and-bound method. The main advantage of the proposed method was that it was suitable for all sizes of the problem. First the method finds a temporary solution using CUT and then it improves the solution by solving a small, crucial part of the problem by using the exact approach. The method had some limitations. It performed better than other methods when the problem consisted of fewer than ten different lengths in stock and on demands if the time limit was set to less than one minute. However, with a small increase in the time limit, the method would perform better than others for all sizes of the problem. C-CUT consists of five steps. In the first step, it obtains the solution using CUT and stores the results. In the second step, the algorithm verifies whether the solution obtained by CUT is optimal and then stops. Otherwise it proceeds with the procedure. In the third step, it forms a sub-problem based on the results of the first step. The procedure in the third step also differentiates between cases of surplus or a lack of material. In the fourth step, the C-CUT uses the branch-and-bound exact method and finds a solution to the sub-problem defined in the third step. In the final, fifth step, the part of the cutting plan obtained in the first step is substituted with a solution to the sub-problem. The solution obtained is then compared with the solution obtained by CUT. The best one is selected as the final solution (Gradišar & Trkman, 2005).

A practical case from the metal industry was presented by Chu and Antonio (1999). They considered minimization of trim loss and cutting time in metal tube production and developed two heuristics for different purposes. The first one tackled responses for individual customer demand that needed to be as fast as possible. The second delivered the manufacturing plan for the next day. The amount of usable leftovers that are put back into stock was determined by transportation costs and storage capacity (Chu & Antonio, 1999).

A decision-support tool for solving a cutting and reuse problem was developed for use in the car industry based on a European plant producing gear belts. The purpose was to minimize trim loss, control quality, equalize workload, and minimize setup. The method was formed as a linear integer programming method and used standard packages within a column generation scheme (Arbib et al., 2002).

The next method that considered returning items back to stock and using them again was published by Kos and Duhovnik (2002). The focus was on cutting optimization with variable stock sizes and inventory status data. A hybrid genetic algorithm was proposed in order to minimize waste. The large trim loss items would be returned to the inventory and used again for subsequent optimizations. The hybrid genetic algorithm is presented as general genetic algorithm operators (crossover and mutation) combined with other heuristic algorithms. Regarding encoding, the efficiency of crossover was modified and therefore improved with a grouping method in the chromosome. The proposed framework for the genetic algorithm consisted of six sequential sub-steps. In the first step, the initial population was generated using an approximation algorithm. In the second step, the fitness of individuals was evaluated. In the third step, the selection was made. The fourth step concerned crossover, where "genes" from the parents were applied to their descendants by combining a chromosome. In the fifth step, some individuals in the newly created population were randomly substituted in order to ensure diversity in the population. The last, sixth step, stopped the procedure if the criteria had been reached regarding best members; otherwise, the procedure returned to the third step and looped again. Crossover and mutation are important operators of the genetic algorithm and therefore the paper offers a precise description of the four steps for each operator. The evaluation function is executed in the final stage of the proposed algorithm. This method minimizes waste and produces larger leftovers that can be used again (Kos & Duhovnik, 2002).

An interesting approach that can also be used to solve the 1DCSPUL is found in Trkman and Gradišar (2007). Instead of applying optimization to a single order, they considered an optimization of consecutive order sets that need to be filled. The reason is that cutting is a continuous business process and so the partly utilized stock lengths should be used again. In order to satisfy such a situation, a previously developed method for solving the 1DCSP was modified. The objective was to minimize the trim loss or production costs. Two models were developed. The testing results of the first model exposed some disadvantages of the first model in terms of a high trim loss at later time periods. Those disadvantages were then eliminated with inclusion of the costs of returning usable leftovers back to stock. The computational test of the second model suggested better performance of the proposed method in comparison to other methods (Trkman & Gradišar, 2007).

Usable leftovers were also considered by Alves and Valério de Carvalho (2007). The authors modified the column generation method to better control the dual variables, which are of key importance for fast performance of the algorithms.

Koch, König, and Wäscher (2009) presented a practical case in which an unusual algorithm was developed and implemented in a real company in the wood industry. The special features were the strong interdependency of the cutting process with a handling process and the possibility of returning leftovers back to stock to be used again in the future. The proposed decision-support application was based on an integer linear programming model. The main focus of the algorithm was on minimizing the costs of the trim loss, usable leftovers, transportation, and item manipulation (Koch et al., 2009).

For door and window production, Dimitriadis and Kehris (2009) proposed an algorithm to tackle the scheduling and planning problems at a real company. The aim was to reduce the amount of objects used and return the leftovers that were large enough back to stock. The authors modified the FFD (First-Fit Decreasing) and MBS (Minimal Bin Slack) algorithms, which are widely known heuristics for solving cutting and packing problems. First the classic CSP was solved, and the leftovers were regarded as residuals and were then considered for filling orders in subsequent periods (Dimitriadis & Kehris, 2009).

The 1DCSPUL in the aircraft industry was studied by Abuabara and Morabito (2009). They proposed two models of the problem using a mixed integer formulation with the goal of minimizing waste and considering leftovers to be usable. The first model was a special case of the model presented by Gradišar et al. (1997), transformed into a mixed integer problem. The second model was a simplification of the first one and included slight modification of some constraints to determine the leftovers that could be returned to stock. The experiments suggested substantial savings in practical situations (Abuabara & Morabito, 2009).

Poldi and Arenales (2009) developed heuristics based on exhaustive repetition to minimize trim loss when solving the 1DCSPUL. First the constructive part of the problem is solved using FFD and a greedy procedure. Then the residual heuristic is applied by a rounding-down approach and greedy rounding. The proposed algorithms were developed for special cases in which the demand is low and there are various items lengths available in stock. The procedure proved to be accurate, fast, simple, and flexible (Poldi & Arenales, 2009).

Significant work has been done by Cherri et al. (2009). In their paper, the authors presented several modifications of existing heuristic methods to take usable leftovers into account. First the exhaustive repetition heuristic (Hinxman, 1980) was adapted. The modifications concerned the FFD and greedy procedure and were denoted as the FFD_L and $Greedy_L$ procedures. Each of them consisted of several subsequent steps. Next the residual heuristics to obtain a solution to the 1DCSP from the linear relaxation of the integer programming problem (Gilmore & Gomory, 1963) was proposed. Their solution was based on the residual heuristic by greedy rounding (RGR; Poldi & Arenales, 2005), which was adjusted for solving 1DCSPUL and denoted as RGR_L. The proposed algorithm was computationally tested for solving large problems and compared with existing methods. A detailed explanation of the algorithms as well as the detailed computational results can be found in Cherri et al. (2009).

In 2010, Poldi and Arenales developed a large-scale integer linear optimization model for predicting object production. The model included waste and the warehouse cost of keeping items in stock. Linear relaxation was solved by the simplex method with column generation. In addition, the authors proposed two rounding procedures that were based on a rolling horizon scheme. The proposed solution has been used in a real company for solving practical problems (Poldi & Arenales, 2010).

Another heuristic for solving the 1DCSPUL was introduced by Cui and Yang (2010). The algorithm presented consists of two procedures. The first one is based on linear programming to satisfy the majority of the demand. The second procedure is SHP and satisfies the remaining demand. The criteria that are considered are the cost of consumed objects, profit from usable leftovers, and profit from shorter stock reduction. Computational tests showed that this algorithm outperforms other methods in reducing object costs, reducing trim loss, ensuring shorter stock, and increasing the average length of usable leftovers. A precise description of the algorithm, the procedures developed, and computational experiments can be found in Cui and Yang’s paper (2010).

New formulations and mathematical models that are modifications of previously developed models can also be found in Ravelo, de Meneses, and dos Santos (2011). The authors proposed two models with adjustments concerning constraint formulation. Some illustrative computational experiments are also presented.

An interesting heuristic that was based on the assumption that usable leftovers (non-standard stock lengths) should not stay in stock for a long time was developed in 2012. The performance of priority-in-use heuristics was tested at successive time periods. The proposed FFD_L^P heuristic consists of two steps. RGR_L^P consists of three steps. Both algorithms are a modification of FFD_L and RGR_L, which were introduced in Cherri et al. (2009). For testing the performance, orders were generated randomly. The stock consisted of remaining standard objects from previous demands and of non-standard objects that were usable leftovers from previously filled orders. Twelve time periods were simulated (Cherri et al., 2012).

2.3 Low ratio between average stock length and average order length

This subchapter is adapted from work published in Gradišar, Erjavec and Tomat (2011).

In addition to the optimization method used, the amount of trim loss can also be influenced by some specific characteristics of the problem that need to be considered when solving the problem. The average trim loss when solving 1DCSPUL is around 15%, as shown in

Erjavec et al. (2009). As already stated in the introduction, the explanation can be found in *r*.

In practice, there are various order and stock lengths for every problem, and so *r* could refer to:

1) the ratio of the longest stock to the shortest order or
2) the ratio of the average stock to average order length.

The first possibility would not reflect the real situation for all situations; for example, in the case of a very large order quantity with the longest order being only a small portion of the total order length. Therefore I employ the second meaning of the *r* for the purposes of my doctoral dissertation.

Because the literature does not provide any definition of a low *r*, it can be defined as follows:

- *r* is low if it is lower than a given threshold *t*.

In the practical example described in Erjavec et al. (2009), *r* equals 3. Thus the same threshold is taken into account in my doctoral dissertation.

3 DEVELOPMENT OF A NEW METHOD

Presented chapter is adapted from work published in Gradišar, Erjavec and Tomat (2011). Where necessary the relevant modifications are added and described.

A lower *r* means that the number of possible solutions is limited, which consequently increases the probability of a solution with a large loss (Gradišar et al., 1999a). The number of possible solutions is the lowest when *r* is between 1 and 2 because this means that in most cases only one unit of the order is cut from one unit in stock. This is also the reason why there is more trim loss produced in cases with a low *r*. It can be presumed from the above that the difference between the trim loss of an optimal solution and the trim loss arising from using the currently available methods—except for exact methods—increases if *r* decreases. Hence there is still a possibility for further improvements. For this reason a new optimization method for solving 1DCSPUL with low *r* is developed.

In relation with the subchapter 2.3 an important issue regards the configuration of the threshold *t*. In practical example that can be found in Erjavec et al. (2009) *r* is around 3 therefore *t* is set to 3 for the needs of this study.

3.1 Problem definition

For defining a problem the following assumptions are taken into account:

- a sufficiently large stock of material is available to satisfy every customer's order;
- a certain number of lengths is available in stock;
- standard stock lengths consist of the same length or of a few different standard lengths;
- nonstandard stock lengths are the leftovers from previous orders;
- leftovers that are equal or greater than minimal order length are treated as usable and are returned back on stock to be used again in the future orders;
- lengths are considered to be integers - if they are not it is presumed that it is always possible to multiply them with such factor, which transforms them to integers; and
- a given quantity of order lengths forms an order that need to be cut into a required number of pieces.

The next notation is used:

l_i - order lengths; $i = 1,...,m$, (order lengths are sorted in a decreasing order: $l_1 \geq l_2 \geq l_3$ ...),

n_i - required number of pieces of order length l_i,

L_k - stock lengths; $k = 1,...,p$,

N_k - number of pieces in stock of length L_k,

x_{jk} - frequency of cutting pattern *j* that is cut from stock length *k*,

n_k - total number of cutting patterns that are cut from stock length *k*.

The cutting plan is formed based on cutting patterns that have been cut from various stock lengths and of the frequency needed to fulfill all the orders.

Cutting pattern *j* that is cut from stock length *k* is expressed by a vector

$$(a_{1jk}, a_{2jk}, a_{3jk}, \dots, a_{mjk}), \tag{61}$$

which satisfies

$$\sum_{i=1}^{m} l_i \cdot a_{ijk} \le L_k, \tag{62}$$

$$a_{ijk} \ge 0 \land \text{integer}. \tag{63}$$

a_{ijk} signifies the number of times that order length l_i occurs in a specific pattern.

The formulation of integer programming model for minimizing sum of stock lengths to be cut is as following:

$$\min \sum_{k=1}^{p} \sum_{j=1}^{n_k} x_{jk} \cdot L_k \tag{64}$$

s.t.

$$\sum_{j=1}^{n_k} x_{jk} \le N_k \quad \forall\, k \quad \text{(stock constraints)}, \tag{65}$$

$$\sum_{k=1}^{p} \sum_{j=1}^{n_k} a_{ijk} \cdot x_{jk} = n_i \quad \forall\, i \quad \text{(demand constraints)}, \tag{66}$$

$$x_{jk} \ge 0 \text{ and integer} \quad \forall\, j, k. \tag{67}$$

3.2 Solution development

It is considered that a solution should be independent of commercial software, simple to code and easy to integrate into the company’s information systems for the reasons of constantly changing information systems that support studied processes. Therefore special care is devoted to such a development that allow for flexibility and adaptability of the method.

Nowadays the competition between supply chains instead of competition among single companies is increasingly gaining importance. In general a better integration of supply chain assures the more frequent exchange of smaller amount of material therefore the orders are becoming smaller (Trkman & McCormack, 2010). So the importance of methods for solving small to medium- size problems will increase.

Various approaches are found in the literature for solving such problems. The majority of them solve the problem heuristically and only few of them have been reported to obtain a solution using an exact approach. Exact solutions are by definition better if results are measured only by the levels of trim loss. However, in many industries solution should consider time efficiency of solution as well. Exact methods were proved to be time efficient only when solving relatively small orders. The limit is loose but nevertheless recent exact methods solve problems with 5 different stock sizes and from 40 (Belov & Scheithauer, 2002) to 100 order items (Alves & Valério de Carvalho, 2008). Problems are solved heuristically when these figures are exceeded.

Because the literature does not provide any specific information about cases with low *r*, I carried out an experiment using the C-CUT algorithm (Gradišar & Trkman, 2005) to check if the problem could be solved exactly. I was increasing the number of orders at usual *r* (more than 5) and lower *r* (less than 2) and observed whether the optimal solution was found in a reasonable time. The experiment was performed with software for developing and formulating models MPL for Windows 4.2 on a PC using 1.87 GHz Intel Pentium processor and 508 MB of RAM. The results of experiment where *r* is usual are presented in Table 9. The time that was still acceptable for finding a solution was set to 10 minutes.

Table 9: The results of C-CUT at usual r.

Number of items in the order	Time needed for finding a solution (sec)	Was the solution optimal?
15	87	Yes
18	80	Yes
19	520	Yes
20	210	Yes
20	153	Yes
21	600	No
21	600	No
24	600	No

The results of experiment where *r* is low are presented in Table 10. The time that was still acceptable for finding a solution was set to 10 minutes.

Table 10: The results of C-CUT at low r.

Number of items in the order	Time needed for finding a solution (sec)	Was the solution optimal?
15	0.9	Yes
18	0.4	Yes
21	0.05	Yes
24	7.28	Yes
27	0.72	Yes
30	12.52	Yes
33	0.53	Yes
36	17.17	Yes
36	55.63	Yes
37	600	No
37	600	No
37	600	No
38	600	No
39	600	No

The algorithm could not find an optimal solution if the number of orders was equal to or higher than 21 at usual *r* and equal to or higher than 37 at lower *r*. Therefore I can presume that a similar relationship would also hold for modern algorithms, which currently enable up to 100 orders at usual *r*. Thus the number of orders at lower *r* would be a bit lower than 200, which is less than it is enabled by the proposed algorithm, which I estimate is able to process up to 700 orders.

With respect to the above described a proposed method is a constructive heuristics that is based on the exhaustive repetition for solving small to medium-size instances of cutting problem with low *r*. It uses a combination of item-oriented and pattern-oriented approach and consists of four iterative steps, each of which fulfills a portion of demand. After all the demand is satisfied the algorithm stops. At the beginning of the procedure all stock lengths have its place in the set of unprocessed stock lengths. The amount of unprocessed pieces (UN_k) of each stock length L_k is equal to N_k. The amount of unprocessed pieces (un_i) of each order length l_i is equal to n_i. The sets of processed stock lengths and processed order lengths are both empty. The set of unprocessed pieces of stock lengths is diminished in every reiteration. The amount of pieces that are cut changes so as the processed stock lengths that are reduced to the trim loss. The specific stock lengths cannot be used in next iteration if all pieces of that stock length were used in previous iteration. Finally all un_i are equal to 0. The algorithm contains of the following iterative steps:

1. Step

The knapsack problem is solved. The optimal and second to optimal solution are found for each stock length. Only unprocessed pieces of order lengths are considered.

2. Step

Cutting patterns obtained in 1. step are considered and sorted according to a_{1jk}, a_{2jk}, a_{3jk} ... As a result sorted patterns are saved in a list. The cutting patterns that contain longer order lengths are positioned on the top of the list. The cutting patterns that contain shorter order lengths are positioned on the bottom of the list.

3. Step

A corresponding frequency of each cutting pattern is selected regarding the unprocessed pieces of order and stock lengths. First the frequency for the cutting pattern that is on the top of the list is designated. Than the algorithm sequentially moves down the list and selects the frequency for second cutting pattern and so on. The corresponding frequency is represented by the highest frequency that is not higher than the current amount of unprocessed pieces of corresponding stock length. At the same time it has to be so low that enables the prevention of an overproduction of any order length. Thus the corresponding amount of unprocessed pieces of stock and order lengths decreases instantly after the individual frequency is selected.

4. Step

The algorithm stops if all the orders are satisfied. Otherwise it returns to the 1. step.

When developing an algorithm, the next assumptions were taken into account:

- In case of low r the optimal or near to optimal solution of cutting problem compounds of optimal or second to optimal solution of knapsack problems.
- In order that a good solution would be obtained more easily the cutting patterns that consist of largest order lengths are processed earlier.

The first assumption regards to the fact that the number of possible solutions is lower in the case of low r. Thus the probability of optimal or second to optimal solution of knapsack problems being also optimal or near to optimal solution of the cutting problem is higher.

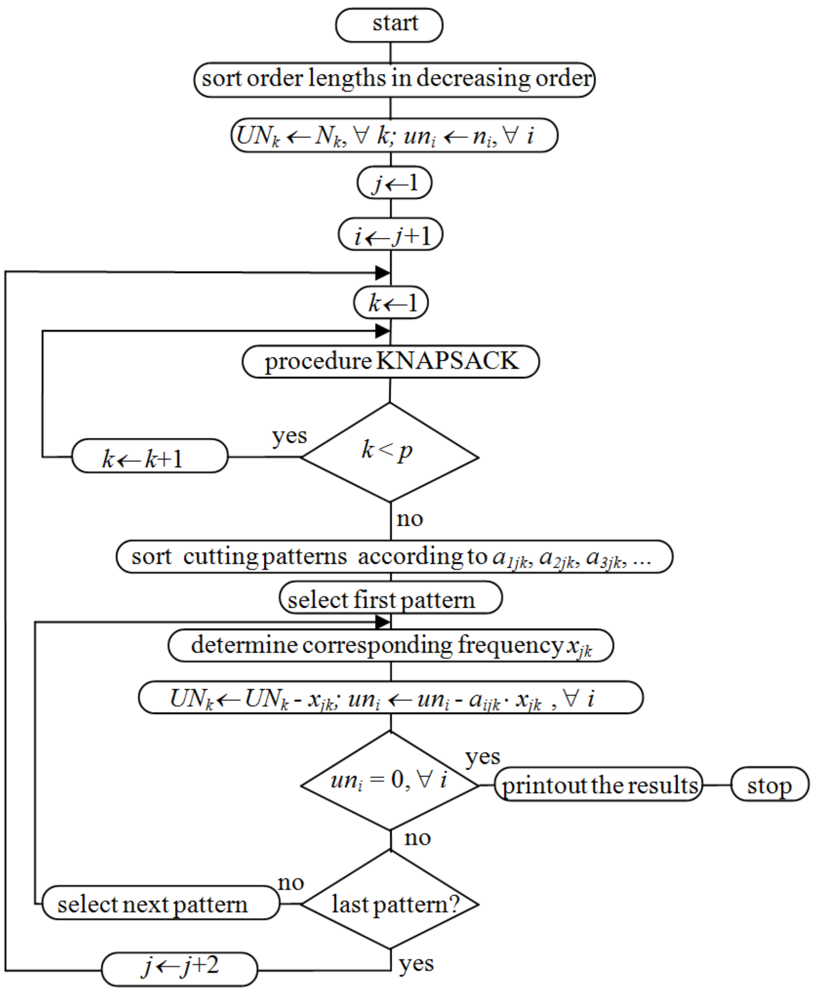
The second assumption regards to the minimization of the so-called ending conditions[10] (Haessler & Sweeney, 1991). The good solution could be obtained more easily if the set of possible solutions is larger (Gradišar et al., 1999a). So the longer order lengths are

[10] It is one of the main disadvantages of SHP, which may cause increased trim loss.

processed earlier to preserve the set of possible solutions as large as possible for the longest possible period.

The proposed algorithm that optimizes the stock length cutting can be seen on a flowchart in Figure 23.

Figure 23: Flowchart of cutting algorithm.



As it can be seen from Figure 23 a series of knapsack problems need to be solved to obtain cutting patterns. Then the corresponding frequency for each cutting pattern needs to be selected for each iteration. A sequence of vectors (*a*) is generated in a lexicographically decreasing order when applying the knapsack procedure to allow finding the optimal and second to optimal patter for L_k. *m* should be 7 or less in order to expect acceptable response time.

3.3 Testing the solution and analysis of the results

Due to very fast processing FORTRAN programming language was selected to code the algorithm. It is called LCUT and can run on a personal computer. The program contains less than 1.000 lines of code. For coding the data input and results printout the 4GL[11] was used. LCUT is stand-alone application but it can be easily integrated into existing information systems.

Following constraints are built in the LCUT:

- ratio between largest stock and shortest order length is ≤ 10;
- number of different order lengths is ≤ 7;
- number of pieces for each order length is ≤ 99;
- number of different stock lengths is ≤ 20; and
- number of pieces for each stock length is ≤ 99.

Due to those constraints less than 10 seconds is needed for creating a cutting plan on a personal computer.

To demonstrate the practical use of LCUT first an experiment with the real data was conducted. The data were obtained from the one of the big retailing companies of the technical products in south-east Europe. Among others the company is resells various metal bars that are cut in order to satisfy the customer's demand. The customer demand consists of 4 different order lengths with required sum of 34 pieces. Input data are presented in Table 11.

Table 11: Details about order lengths.

Number	Length (cm)	Pieces
1	965	12
2	780	7
3	538	10
4	430	5

Source: Retail Company, 2011.

On stock there is plenty of material available. It consists of 57 pieces of 15 different lengths with *r* being equal to 1.2. Standard and non-standard stock lengths are stored together. All lengths are in centimeters. The stock data are presented in Table 12.

[11] 4GL refers to Fourth-Generation Programming Language that consists of packages for software development. It is especially appropriate for developing heuristic solutions.

Table 12: Details of stock lengths.

Number	Length	Pieces
1	1,200	12
2	750	10
3	685	10
4	590	10
5	865	4
6	1,600	2
7	600	1
8	500	1
9	1,400	1
10	640	1
11	800	1
12	765	1
13	670	1
14	690	1
15	820	1

The question arises if the described problem is small enough to be solved optimally. As already explained in the subchapter 3.2 the limit of problem size that can be solved exactly in a reasonable time is unclear and depends on the computational power. Exact methods in the literature enable solving problems with five different stock sizes and from 40 (Belov & Scheithauer, 2002) to 100 order items (Alves & Carvalho, 2008). In addition I conducted an experiment using the C-CUT algorithm (Gradišar & Trkman, 2005) to see what sizes of the problems could be solved exactly. The experiment is described into details in subchapter 3.2. An optimal solution could not be found if the number of orders was equal to or higher than 21 at usual *r* and equal to or higher than 37 at lower *r*.

Since the presented problem consists of 34 ordered items and 57 stock pieces with 15 different stock lengths it is possible to presume that it could not be solved optimally. Even if the exact method would provide an optimal solution to this particular problem, it would not be suitable for use in practice, where the majority of the problems are of significantly higher size and thus cannot be solved exactly.

However, the presented experiment was conducted to demonstrate the practical use of LCUT. For finding the solutions less than 1 second was needed. The results are presented in Table 13.

Table 13: Results of LCUT – utilized stock lengths.

Number	Length	Pieces	Pattern	Trim loss (cm)	Trim loss (%)
1	1200	11	1 times 965[1]	235	19.58
4	590	10	1 times 538[3]	52	8.81
5	865	2	2 times 430[4]	5	0.58
5	865	1	1 times 780[2]	85	9.83
6	1600	2	2 times 780[2]	40	2.50
9	1400	1	1 times 965[1], 1 times 430[4]	5	0.36
11	800	1	1 times 780[2]	20	2.50
15	820	1	1 times 780[2]	40	4.88

Total trim loss: 3345 cm (11.98%)

Used amount of stock lengths equaled to 29 pieces. All patterns consisted of one or two pieces. The frequency of the patterns varied from 1 to 11. The total trim loss amounted 3345 cm, which represents 11.98% of total used stock lengths.

To show the efficiency of LCUT the experiment was conducted with two other methods as well. The first is RGR heuristics (Cherri et al., 2009) and the second is CUT (Gradišar et al., 1999a). The results of RGR and CUT are presented in Table 14 and Table 15.

Table 14: Results of RGR – utilized stock lengths.

Number	Length	Pieces	Pattern	Trim loss (cm)	Trim loss (%)
1	1200	11	1 times 965[1]	235	19.58
4	590	10	1 times 538[3]	52	8.81
5	865	2	2 times 430[4]	5	0.58
5	865	2	1 times 780[2]	85	9.83
6	1600	2	2 times 780[2]	40	2.50
9	1400	1	1 times 965[1], 1 times 430[4]	5	0.36
11	800	1	1 times 780[2]	20	2.50

Total trim loss: 3390 cm (12.17%)

As it is seen from the Table 14 the total trim loss amounts 3390 cm in case of RGR, which represents 12.17% of total utilized stock lengths. From comparison with the results obtained by LCUT it can be seen that the trim loss with LCUT is lower for 45 cm or 0.19 percentage points that represent savings of total utilized stock lengths.

Table 15: Results of CUT – utilized stock lengths.

Number	Length	Pieces	Pattern	Trim loss (cm)	Trim loss (%)
1	1200	12	1 times 965[1]	235	19.58
4	590	9	1 times 538[3]	52	8.81
5	865	2	2 times 430[4]	5	0.58
6	1600	2	2 times 780[2]	40	2.50
8	500	1	1 times 430[4]	70	14.00
9	1400	1	1 times 780[2], 1 times 538[3]	82	5.86
11	800	1	1 times 780[2]	20	2.50
15	820	1	1 times 780[2]	40	4.88
				Total trim loss: 3590 cm (12.75%)	

As it is seen from the Table 15 the total trim loss amounts 3590 cm in case of CUT, which represents 12.75% of total utilized stock lengths. From comparison with the results obtained by LCUT it can be seen that the trim loss with LCUT is lower for 245 cm or 0.77 percentage points that represent savings of total utilized stock lengths.

To show the performance of LCUT another experiment was conducted with more extensive series of problem instances. For testing the method a specific problem generator is used. One of the most widely known problem generator when testing the methods for standard 1DCSP is CUTGEN1, which is based on five problem parameters (Gau & Wäscher, 1995):

m - problem size;

L - standard stock length;

v_1,v_2 - lower and upper bounds for order lengths;

đ - average demand per order length.

The CUTGEN1 was then generalized to allow usage for any type of the problem in the terms of variety of large objects. Thus the new structures for determination of standard and non-standard stock lengths were added. The parameters of proposed problem generator PGEN (Gradišar et al., 2002) are:

n - number of different order lengths,

v_1,v_2 - lower and upper bounds for order lengths, $v_1 \leq l_i \leq v_2 \ (i = 1, \dots, n)$,

đ - average demand per order length,

p - number of different standard stock lengths,

s_1, s_2 - lower and upper bounds for standard stock lengths $s_1 \leq L_k \leq s_2$ $(k = 1, \dots, p)$,

m - number of non-standard stock lengths,

u_1, u_2 - lower and upper bounds for non-standard stock lengths

$u_1 \leq U_j \leq u_2$ $(j = 1, \dots, m)$.

PGEN consists of the next subsequent steps (Gradišar et al., 2002):

1. Generation of test problems:
   - defining the number of test problems,
   - defining the order lengths and demands,
   - defining the standard stock lengths,
   - defining the non-standard stock lengths.
2. Verification if the algorithm is the solution of standard 1DCSP:
   - if it does it is transformed into solution of hybrid 1DCSP
   - solution to the test problems is obtained.

For the evaluation criteria the sum of all test problems overall trim loss is used due to its simplicity. More in-depth explanation of the problem descriptors generation and seed sequences determination that is tackled by the dynamic programming scheme PROGEN can be found in Gradišar et al. (2002).

For generating problem instances a PGEN generator with slight modifications was used; in accordance with problem descriptors the input data were generated as random sample of one or more test problems. The next problem descriptors were used:

m - number of different order lengths.

v_1, v_2 - lower and upper bound for order lengths, i.e. $v_1 \leq l_i \leq v_2$ ($i = 1,\dots,n$).

n - average demand per order length.

P - number of different standard stock lengths.

s_1, s_2 - lower and upper bound for standard stock length, i.e. $s_1 \leq L_k \leq s_2$ ($k = 1,\dots,P$).

N - number of pieces of standard stock lengths.

p - number of non-standard stock length.

u_1, u_2 - lower and upper bound for non-standard stock length, i.e. $u_1 \leq L_j \leq u_2$ ($j = 1,\dots,p$).

r - number of consecutive generated problem instances.

The tenets of the parameters for generation of test problems were:

- Determination of order lengths and demands.
- Determination of standard and non-standard stock lengths.

The following values were taken into account:

- $m = 5$;
- $v_1 = 100$ and $v_2 = 200$;
- demand $n = 10$, 20 and 30;
- number of different standard stock lengths $P = 5$, 10 and 15;
- number of pieces $N = 10$; and
- number of non-standard stock lengths $p = 10$.

In line with the above description the stock consisted of 60,110 and 160 pieces. Lower and upper bound for standard stock length s_1, and s_2 had such values that allowed the variation of r from 1.25 to 3 by step 0.25. That means s_1 varied from 125 to 300 and s_2 from 250 to 600. The non-standard lengths are the usable leftovers from previous instances therefore they must not exceed maximum order lengths. Also on average they should not be greater than order lengths. Thus the u_1 and u_2 has been set to 150 and 200.

24 instances classes have been designed by combining different values of the above-mentioned problem parameters. Then 10 successive problem instances were generated for each class. The seed of random problem generator was modified by 1 from 1 to 240 so altogether 240 problem instances were generated.

Problem descriptors, seeds can be seen in Table 16, where each row represents 10 problem instances.

Table 16: Problem descriptors and seeds.

Problem descriptors					Seed	
Class instance	***n***	s_1	s_2	***P***	**from**	**to**
1	10	125	250	5	1	10
2	20	125	250	10	11	20
3	30	125	250	15	21	30
4	10	150	300	5	31	40
5	20	150	300	10	41	50
6	30	150	300	15	51	60
7	10	175	350	5	61	70
8	20	175	350	10	71	80
9	30	175	350	15	81	90
10	10	200	400	5	91	100
11	20	200	400	10	101	110
12	30	200	400	15	111	120
13	10	225	450	5	121	130
14	20	225	450	10	131	140
15	30	225	450	15	141	150
16	10	250	500	5	151	160
17	20	250	500	10	161	170
18	30	250	500	15	171	180
19	10	275	550	5	181	190
20	20	275	550	10	191	200
21	30	275	550	15	201	210
22	10	300	600	5	211	220
23	20	300	600	10	221	230
24	30	300	600	15	231	240

The results of experiment are presented in Table 17. All lengths are in centimeters. To allow comparison the solution for each problem instance was solved with both LCUT and CUT. The later was selected because it does not depend on any commercial software and the source code can be entirely controlled by user. Both LCUT and CUT are stand-alone applications. It is also possible to combine LCUT and CUT into a single application. Values of parameters B, E and W indicates how many times the results obtained with LCUT are Better, Equal or Worse than the results obtained with CUT with respect to the total waste. R1 shows the total number of usable leftovers that are returned back to stock when applying LCUT. With CUT those values are denoted as R2.

As already mentioned returning leftovers and reusing them result in additional costs. In practice those costs vary from situation to situation but in general they can be positioned somewhere between following two cases:

- In the Case 1 those costs are insignificant in comparison with the costs of saved material. In such case all leftovers that are longer or equal to the minimal order length are returned back on stock with the purpose to be used again in future orders. They are not considered as a waste.
- In the Case 2 the cost of returning leftovers back on stock are significantly high. In terms of costs it is therefore better to treat them as a waste than return them to stock.

To show the importance of both scenarios the presented experiment was carried out for both Case 1 and Case 2.

Table 17: Results of LCUT and CUT.

Class instance	Case 1					Case 2		
	B	***E***	***W***	***R1***	***R2***	***B***	***E***	***W***
1	0	9	1	0	1	1	9	0
2	1	8	1	0	0	1	8	1
3	1	8	1	0	0	1	8	1
4	1	3	6	0	5	3	3	4
5	1	2	7	0	4	5	2	3
6	1	3	6	46	19	2	2	6
7	2	0	8	0	9	8	1	1
8	5	1	5	0	6	7	1	2
9	4	0	6	0	8	8	1	1
10	0	0	10	0	9	4	0	6
11	2	0	8	4	14	7	0	3
12	4	0	6	10	16	6	0	4
13	1	0	9	4	16	6	0	4
14	0	0	10	1	9	2	0	8
15	2	1	7	6	34	5	0	5
16	0	0	10	1	15	6	0	4
17	1	0	9	0	12	6	0	4
18	3	0	7	0	14	7	0	3
19	0	0	10	0	19	9	0	1
20	3	0	7	1	17	8	0	2
21	2	0	8	0	18	5	0	5
22	0	0	10	2	16	8	0	2
23	2	0	8	8	15	5	0	5
24	0	0	10	1	20	7	0	3
Sum	**35**	**35**	**170**	**84**	**296**	**127**	**35**	**78**

The analysis of the results shows that in Case 1 LCUT performed better 35 times than CUT, 25 times the results were equal and 170 times the result obtained by LCUT was worse. In Case 1 R2 is significantly higher than R1 since the overall amount of R2 is 296 and the overall amount of R1 is 84. With LCUT usable leftovers are produced only in 13 out of 240 problem instances so the probability that one or more usable leftovers will be produced in the next problem instance is about 5%. This probability is about 84% by using CUT. In Case 2 LCUT the obtained results are 127 times better, 25 times equal and 78 times worse than with CUT. No usable leftovers were produced.

The average waste is also taken into account for better evaluation of presented algorithm. In Case 1 LCUT causes 8.4% average waste, CUT 8.0 %, and combination of LCUT and

CUT 7.9%. The combination regards to the each problem being solved with both LCUT and CUT. Then better solution is selected. The use of LCUT alone would make no sense but in combination with CUT the waste decreases for 0.1%. Such combination not only reduces waste but also diminishes the amount of usable leftovers. In Case 2 there is no usable leftover after the problem is solved. Thus the average waste is higher. LCUT produces 8.7% average waste, CUT 9.2%, and combination of LCUT and CUT 8.5%. Thus on average LCUT saves 0.5 percentage points more material than CUT. If it is combined with CUT 0.7 percentage points of material is saved. The results of experiment are summarized in the Table 18.

As it can be found in literature such decrease is not negligible as several methods provide smaller savings in comparison with other methods. For example, FFD_L consumes 0.01% less material than CUT, FFD consumes 0.03% less material than RGR_L 1, $Greedy_L$ consumes 0.05% less material than Greedy and CUT consumes 0.07% less material than RGR 1. RGR_L 3 produces 0.08% less scrap than FFD_L and RGR 3 produces 0.1% less scrap than RGR 2. Considering the amount of produced retails $Greedy_L$ generates 0.1 percentage points less retails than RGR_L 1 and RGR_L 2 and 0.3 percentage points less retails than RGR_L 3 and COLA. Regarding total loss of material constructive Greedy outperforms constructive FFD for 0.3% (Cherri et al., 2009). Several methods were also tested with problems that contain small sized items. The overall total loss of material using RGR_L^P is 0.0001 percentage points smaller than with RGR_L and 0.005 percentage points smaller than with RGR. Regarding medium sized items RGR_L and RGR_L^P produce 0.01 percentage points smaller overall trim loss than RGR (Cherri et al., 2012).

Table 18: Summarization of results.

	Case 1	**Case 2**
LCUT better than CUT	35x	127x
LCUT equal to CUT	35x	35x
LCUT worse than CUT	170x	78x
LCUT average waste	8.4%	8.7%
CUT average waste	8.0%	9.2%
LCUT+CUT average waste	7.9%	8.5%

The average percentage of waste depends on the *r* in both cases. In the first three class instances of Table 17 this ratio equals 1.25. The average waste amounts 13% of used stock. In the last three class instances *r* is 3. The average waste decreases to 3%. Regarding usable leftovers the value of R2 is significantly lower in the first half of Table 17 than in the second half. Such situation occurs due to increased number of possible solutions. Values of average demand per order length *n* in Table 16 do not affect the results since the calculation of each of 240 problem instances takes less than 1 second with both algorithms LCUT and CUT.

If the exact methods would be used the optimal solution for some problem instances could be found. Nevertheless the most of problems are of bigger size than those that could be solved exactly. Current state for solving the problem with exact methods is limited to five different stock sizes and 40 (Belov & Scheithauer, 2002) to 100 (Alves & Valério de Carvalho, 2008) order items. The experiment that took into account the cases with low *r* was already deeply explained in the subchapter 3.2.

Additional testing of the proposed algorithm has been made. 48 samples were generated with the same *r* as in Table 17. For each ratio four problem instances were generated. The number of different stock lengths equaled 5. The total number of stock units was 25. On the order side there were up to 30 items of 5 different lengths. Those 48 instances were solved with LCUT, CUT and an exact method, proposed by Gradišar & Trkman (2005). The average produced waste was observed. The results are summarized in Table 19.

Table 19: Summarized results of exact method, CUT and CUT+LCUT.

	Exact Method	**CUT**	**CUT+LCUT**
Case 1	5.6%	7.5%	7.3%
Case 2	7.3%	9.4%	8.8%

In Case 1 the average waste produced by an exact method amounted 5.6%, by CUT 7.5% and by the combination of CUT and LCUT 7.3%. In Case 2 the exact method produced 7.3%, CUT 9.4% and a combination of CUT and LCUT 8.8% of total waste on average. The presented solutions differ from those presented in Table 18 due to the reduction of a problem instances size. Therefore *r* was slightly different. The use of exact method would be more reasonable in such situation but the presented algorithm enable solving problem instances that consist of up to 693 orders and 1980 stock items of 20 different lengths, which is substantially beyond the limits of exact methods. Although there is still room for improvement it is possible to conclude that the proposed method is better than existing method for solving such kind of a problem.

4 DEMONSTRATION OF THE TESTING METHOD

In this chapter I introduce the importance of using simulation in relevant areas of my doctoral dissertation. Then I present a new simulation-based concept for testing methods for solving the 1DCSPUL. Such an approach has not yet been well researched in the literature, although it has been mentioned and suggested by Trkman and Gradišar (2007) and Cherri et al. (2012). The main idea lies in considering the usable leftovers from previous instances and using them again in subsequent instances. Thus real situations in which usable leftovers are returned back to stock would be better addressed.

Except for the above mentioned papers (Trkman & Gradišar, 2007; Cherri et al., 2012) the methods in the literature do not take into account returning of leftovers from current instance back on stock and using them in the following ones. To the best of my knowledge, none of the papers provide a sufficient and accurate testing method for comparison of the solutions. Methods for solving 1DCSPUL are thus incorrectly tested. To overcome this issue a new testing method is proposed.

The purpose of this chapter is to demonstrate the new concept for testing various solutions to 1DCSPUL. The key question is how the stock of usable leftovers would grow over time and influence the future orders. Therefore I simulate the level of usable leftovers in stock as it would grow over time in reality if the sequence of orders generated were filled with the consecutive generated shipment of standard bars.

4.1 Computer simulation

With the development and accessibility of computers, researchers and managers have been able to explore a broader variety of possible options in decision-making. They have been able to use the computer to simulate a real system and, due to its speed, they could simulate a system over a long period of time. The simulation could also be carried out without a computer, but because decision problems need to be solved as quickly as possible the computer simulation plays an important role in management science (Pidd, 2004). Today simulation represents one of the most widely used and powerful decision-making tools for the design and operation of complex processes and systems (Shannon, 1998).

There are many definitions of the simulation. For the purposes of the doctoral dissertation, a simulation can be described as a method that allows experimentation on a model of some system. It enables better understanding of the complex processes interactions within that system (Pidd, 2004).

Simulation involves a set of techniques that can show the operational aspects and relationships in models. This can be achieved through sampling and observing the evaluation parameters for the phenomena one wishes to simulate (Seila, Ceric, &

Tadikamalla, 2003). The main idea is to mimic the individual processes' operations from reality in relation to the time dimension (Banks, 2005).

In addition to the concept of simulation being easy to understand, it has many other advantages (Pidd, 2004; Shannon, 1998):

- it allows testing new designs without resources that would be needed for real implementation;
- new concepts, operations, decisions, structures, and so on can be explored without disturbing the current processes;
- it enables bottleneck identification;
- a hypothesis related to a certain phenomenon can be efficiently tested;
- it is possible to control the time dimension; slowing up the process enables in-depth analysis. Several years can be simulated in a second;
  detecting the most important variables for the system is simplified; and
- it has the ability to experiment with unknown situations and answer "what if" questions.

Like every other technique, simulation also has some disadvantages (Banks, 2005; Shannon, 1998):

- professionally skilled practitioners are needed to ensure that simulation reflects the real situation as much as possible;
- the input data should be highly reliable, which may sometimes require a lot of time;
- the results depend on the quality of the model, and so it has to be thoroughly designed;
- for a given input the simulation models produce probable output but do not "obtain" the optimal solution;
- the use of direct experiments can sometimes be cheaper;
- the cost of simulation can exceed the benefits gained from the results; and
- if the system is too complex it is very hard to define it in a way that would allow reliable simulation.

In general the simulation can be divided into two classes (Banks, 2005):

- static simulation is based on observation and determination of patterns that are then transformed into formulas and rules; and
- dynamic simulation is based on studying the system behavior over periods of time.

Two different types of simulation models exist: discrete and continuous models. Usually the independent variable is time. In discrete models, the dependent variables change across time periods. In continuous models, the state of the system is given through a functional relationship to the time variable (Gilbert & Troitzsch, 2005).

In addition, simulation models can be divided into deterministic or stochastic simulation models. In deterministic models, the performance of the system can be fully predicted in advance. In stochastic models, it can be predicted with a certain probability (Banks, 2005). The computer simulation used in the study can be denoted as stochastic because several subsequent time periods are taken into account.

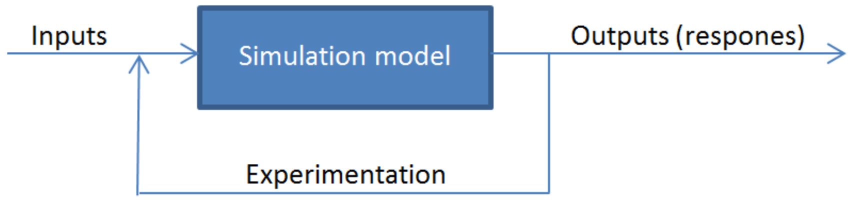
A simulation study should consist of the following steps (Shannon, 1998):

1. Problem definition
   In this step, the goals of the experiment should be defined, the purpose identified, and the questions stated.
2. Project planning
   This step considers identification of efficient and appropriate personnel, management support, and computer hardware and software resources for conducting a study.
3. Conceptual model formulation
   In this step, the preliminary model is developed (e.g., flowchart, block diagram, pseudo-code, etc.). The components, descriptive variables, and interactions among them are defined.

4. Preliminary experimental design
   The measures of effectiveness that will be used are proposed in this step. It also includes the selection of the data that should be gathered from the model.
5. Input data preparation
   This step consists of collecting and identifying the data needed for the model.
6. Model translation
   The model should be formulated using appropriate computer simulation language.
7. Verification and validation
   In this step, the model ought to be tested to confirm that it operates as it should. The output must be representative of the output of the real system.
8. Final experimental design
   Now the experiment that will produce the desired information needs to be designed. The manner of execution also needs to be identified.
9. Experimentation
   The simulation is run in order to obtain the desired information and to perform sensitivity analysis.
10. Analysis and interpretation
    The results are analyzed and the conclusions are drawn.
11. Implementation and documentation
    The results and findings are reported. The model and its use are documented.

Computer simulation is an experiment based on a computer model of a certain system. Usually various scenarios are predicted, and those with the best results are implemented into the real system. The experiment can be relatively simple; for example, trying to answer “what-if” questions. In such a case, the corresponding variables in the simulation program are set to these values and the simulation is performed. However, the models implemented are often more sophisticated and complex. Statistical design techniques are needed due to various effects that may be produced as a result of several interrelating scenarios. The basic idea is shown in Figure 24.

Figure 24: Simulation as experimentation.



Source: Pidd, 2004.

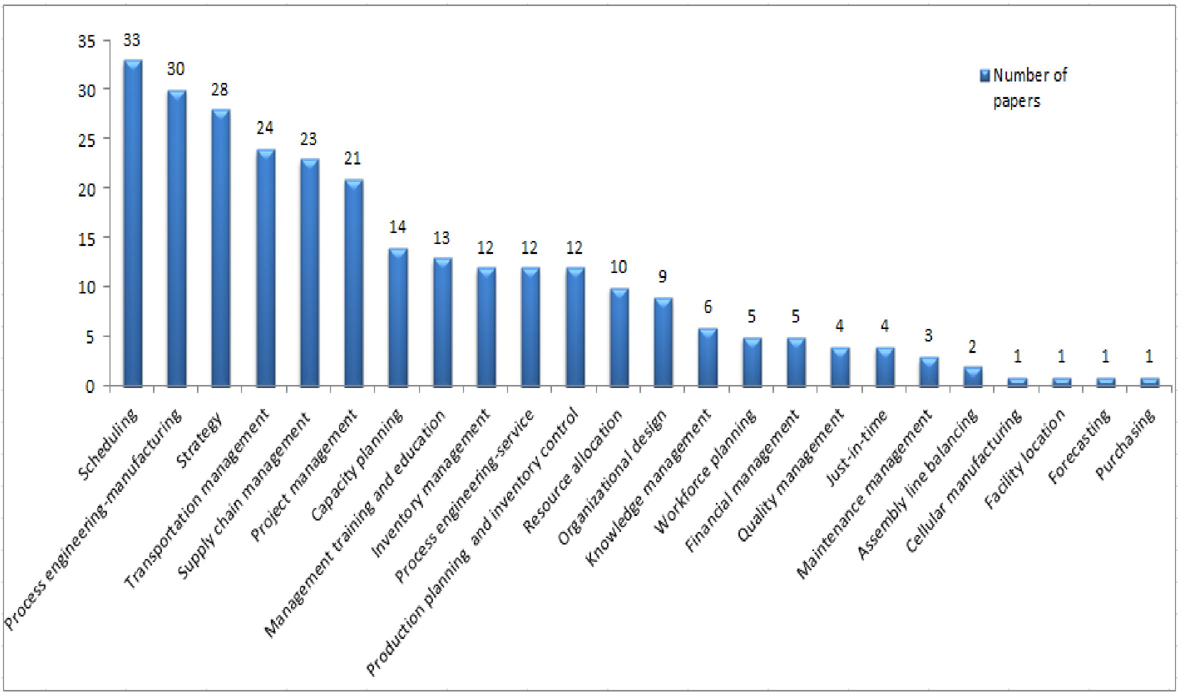
4.1.1 Simulation in practice

The applications of simulations can be found in many different sectors, such as manufacturing, services, defense, healthcare, and public services. Simulations are used in several areas; for example, design, planning and control, strategy making, resource allocation, and training. They have proved to be very useful in response to complexities involving an entire company, especially in decision-making processes. They are also an important tool for organizational learning and change management (Fowler, 1998). The simulation technique is very frequently used by decision-makers due to potential for interactive modeling, which allows for higher transparency and understandability of the models (Kljajić, Bernik, & Škraba, 2000).

Computer simulations occur in several business areas, e.g., manufacturing, services, healthcare, and public services ((Jahangirian, Eldabi, Naseer, Stergioulas, & Young, 2010). It has also attracted much attention from researchers in various fields. Since 2000, the majority of research articles that have considered the application of simulation have been related to SCM (Chan & Chan, 2005; Kleijnen & Smits, 2003; Terzi & Cavalieri, 2004; Van Der Zee & Van Der Vorst, 2005), followed by business process engineering (Jansen-Vullers & Netjes, 2006). A detailed review of research articles relating to simulation can be found in Jahangirian et al. (2010).

Simulation has various applications in practice, such as workforce planning, maintenance management, knowledge management, project management, organizational design, management training and education, financial management, quality management, assembly line balancing, capacity planning, cellular manufacturing, transportation management, facility location, forecasting, inventory management, just-in-time, process engineering-manufacturing, process engineering-service, production planning and inventory control, purchasing resource allocation, scheduling, strategy, and supply chain management (Jahangirian et al., 2010). The distribution of published research articles by the applications mentioned above is shown in Figure 25.

Figure 25: Number of papers by application.



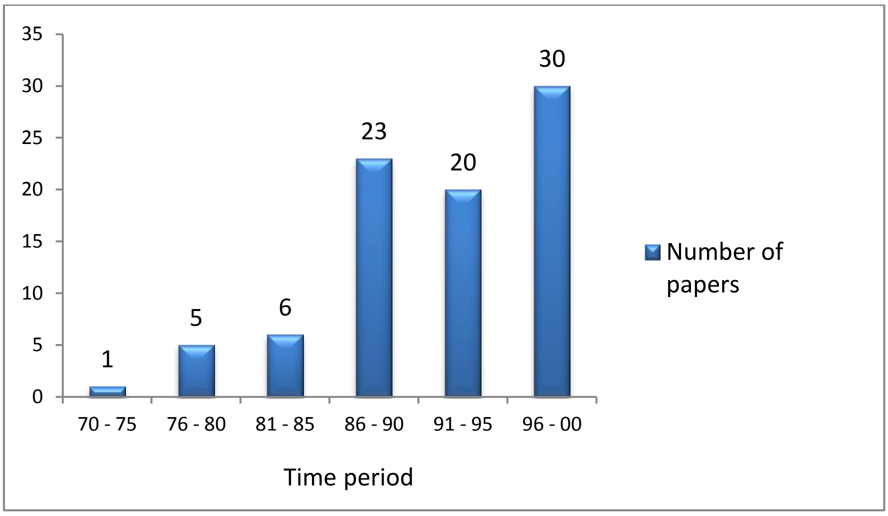
Source: Jahangirian et al., 2010.

4.1.2 Simulation in operational management

The importance of simulation in operational management can be seen from the research conducted by Shafer and Smunt (2004). They examined articles that considered simulation studies in operational management and were published in over twenty leading operations management journals between 1970 and 2000. Out of more than 600 published simulation studies, they extracted eighty-five articles that were the most empirically oriented. These articles were then classified into seventeen categories. They analyzed published empirical operational management simulation studies from 1970 to 2000 by journal and time period, by topic and time period, and by journal and topic. The detailed presentation of the findings can be found in their publication.

The trend for simulation studies in operational management is shown in Figure 26.

Figure 26: Published scientific papers that consider empirical simulation in operational management by years.



Source: Shafer and Smunt, 2004.

From the Figure 26 it is possible to see a rise in the number of publications in operational management simulation studies. Such growth may be the result of an increase in operational management research or an increase in simulation research, but the reason probably lies in a combination of both possibilities (Shafer & Smunt, 2004).

4.1.3 Simulation in relation to cutting processes

Cutting processes represent part of manufacturing processes, and this is an area in which simulation can be used for various analyses. It allow evaluation of investment in the equipment and building, such as factories, warehouses, and distribution centers (Erjavec, 2011). Sometimes simulation is used to predict the workflow of existing or planned systems and for comparing alternative solutions in production planning (Benedettini & Tjahjono, 2009). Simulation is also an appropriate technique for inventory optimization when the demand is uncertain (Kofjač, Kljajić, & Rejec, 2009). The literature includes many articles that consider simulation in relation to cutting processes.

Schulz and Bimschas (1993) conducted simulation of the cutting process to optimize precision machining. The goal of the study was to produce high-precision workpieces. In the real situation there were many errors, and so they simulated planned cutting processes and made these errors predictable. This enabled machining optimization before the start of production. The simulation application was based on persistent work involving simultaneous transmission of power caused by cutting forces onto the current workpiece geometry. The authors demonstrated the simulation of workpiece machining and the optimization processes needed to attain higher precision. This reduces the probability of the workpiece containing errors.

In industry, a finite element method is a widely used simulation technique. Usually the machining process is simulated to study various potentials of mechanism. Key techniques are considered, such as chip separation criteria, contact, self-adaptive mesh, and so on (Zhigang, Yinglin, & Litao, 2003). Another aspect that should be taken into account regards uncut chip thickness. An example can be found in Liu and Melkote (2007).

Work based on a finite element method simulation has been done by Ceretti, Lucchi, and Altan (1999). They simulated an orthogonal cutting operation in various cutting environments. They changed the tool geometry and cutting speed. To simulate ductile fracture of a material, they developed fracture criteria and built them into the model.

Modeling and simulating cutting processes can improve cutting tool designs and determine optimum conditions. This especially holds for advanced applications such as high-speed milling processes. A milling process is an operation of the highest demand because of the high temperatures and stresses that are generated on the cutting machine due to high workpiece hardness. Thus, predicting cutting forces, machine stresses, and temperature characteristics should be taken into account in high-speed end milling. Özel and Altan (2000) developed a methodology for simulating the cutting process in such operations. The proposed methodology predicts chip flow, cutting forces, device stresses, and temperatures. Their approach was based on flow stress data of the workpiece material and friction on the chip. Chip formation on the main and side cutting edges of end milling was predicted by using flat strain and axisymmetric workpiece distortion with modular representation of chip geometry. The predicted cutting forces were compared with the measured ones. The results showed an acceptable accuracy of the proposed methodology.

A finite element method was also used to study the technical parameters and chip mechanism for cutting processes in chip plastic formation and thermo-mechanical coupling. The emphasis was on determining key features; namely, contact and friction between the workpiece and tool, chip separation criteria, and residual stress and strain (Gang & Pan, 2001).

In mechanical and aerospace engineering, simulation has been used to optimize nanometric cutting of single-crystal aluminum (Komanduri, Chandrasekaran, & Raff, 2000). The authors observed the disorder of the atoms in the mechanized surface close to the cut depth in order to gain information about the variation of the cutting forces, the ratio of thrust to cutting force, the particular energy necessary for excluding the element volume of work material, the characteristics of deformation ahead of the tool, the subsurface distortion of the machined surface with crystal alignment, and cutting direction.

In the metal industry, simulation of the orthogonal cutting process based on the Lagrangian–Eulerian finite element method was proposed by Movahhedy, Gadala, and Altintas (2000). They combined formulations in order to join the advantages of both methods. Based on the proposed model, they simulated the cutting process and presented the capabilities and potential of their approach. Another simulation of cutting processes in the metal industry can be found in Li, Gao, and Sutherland (2002), which simulated cutting steel with a special machine. The objective was to identify the effects of dissimilarities of the machine rake face. They simulated four different cases in relation to three different machine face geometries. The model was validated with the experimental data.

Another example of using simulation for improving cutting processes can be found in Klocke, Raedt, and Hoppe (2001), in which the authors simulated high-speed orthogonal turning of steel with potentially endless continuous cutting simulation. They compared the results of the experiment and simulation, and evaluated the input parameters. It was shown that several assumptions in the terms of material properties need to be made when applying simulation of the cutting process.

Yun, Ko, Lee, Cho, and Ehmann (2002) introduced a virtual machining system for simulating an end milling cutting process. The cutting configuration calculation was based on a moving edge node Z-map model. They simulated four different transient cuts and predicted the cutting force variation during the process. A comparison of the results with the measured ones proved the model to be valid.

An interesting use of simulation was presented by Cus and Balic (2003). They proposed a GA-based optimization technique for the metal industry to determine the cutting parameters in machining operations. They researched the production cost and time that influences the quality of the final product with the aim of constant improvement of cutting conditions. The proposed method consists of modification of recommended cutting conditions, neural-network learning of the cutting conditions found, and replacement of better cutting conditions with those that were previously learned by the suggested GA. A comparison with the experimental results shows that the introduced GA procedure is effective and efficient. It was developed in such way that allows easy implementation into companies' existing manufacturing systems.

For research in mechanical engineering, a molecular dynamics method was developed to model and simulate the nanoscale ductile mode of cutting monocrystalline silicon wafer. The problem lay in cutting fragile small materials because a brittle-ductile transition could occur. The results indicated that when a ductile cutting mode is reached the power exerted on a cutting tool is greater than the cutting power. As the thickness of the chip increases, the compressive stress in the cutting sector decreases. Thus the chip is more likely to be cracked. The results of the study helped the company anticipate the machining mechanism and prevent unnecessary costs (Cai, Li, & Rahman, 2007).

Several simulation methods were later developed in nanometric cutting processes. For example, a three-dimensional molecular dynamics simulation was conducted to investigate the influence of uncut chip thickness on the depth of subsurface deformed layers. The relations between workpiece atoms and tool atoms were computed with a combination of embedded atom potential and Morse potential. The results obtained by simulation indicated that the depth of subsurface layers is damaged due to the atomic force microscope pin tool's rake angle. The results were experimentally verified (Zhang, Sun, Yan, Liang, & Dong, 2008). A similar method was introduced by Zhu, Hu, Ma, and Wang (2010) in which they examined the nanometric cutting process of copper using a diamond tool. They considered the tool geometry, cutting depth, cutting speed, and bulk temperature. Their findings were proven with the measurement of the real outputs.

In manufacturing companies, it is very important to plan and optimize the machining process when cutting material in order to reduce machining damage as much as possible. An example of using simulation in a helical milling process was presented by Wang, Qin, Ren, and Wang (2012). They studied the features of multi-phase milling in terms of varying chip thickness and cutting force directions. The prediction model was compared using data obtained through experimental testing. Based on a good match of the data, a tool was developed to predict the change of cutting forces under various cutting conditions.

Many other examples of using simulation to improve the cutting process can be found in the literature, but listing them would by far exceed the framework of the doctoral dissertation.

4.2 Simulation as a testing method

The efficiency of the proposed cutting method can be determined if it is compared with other methods. A comparison between several methods has already been made in Cui and Yang (2010) and the results show that in general RSHP outperforms other methods. The comparison is based on many criteria: total bar length, material utilization level, standard bar count, usable leftovers count, total length of usable leftovers, total trim loss, average usable leftovers count, average length of usable leftovers, average computation time, and number of cutting patterns. However, none of the other five methods tested best satisfied all the criteria listed.

Cherri, Junior, and da Silva (2011) pointed out that it is difficult to determine which method has the best performance. This is because individual algorithms provide different solutions regarding incompatible characteristics, such as the number of leftovers generated and trim loss. A fuzzy technique was developed to overcome this difficulty. The characteristics of individual algorithms were considered in order to allow easier classification of the method. The selection of the most appropriate method for solving a specific problem was therefore simplified (Cherri et al., 2011).

The proposed methods for solving the 1DCSPUL could be tested by using the same criteria and same benchmark instances. However, the results would not be very useful because the main issue is overlooked: when evaluating methods for solving the 1DCSPUL the group of instances should be taken into account instead considering individual instances, in which the usable leftovers for each instance are generated with a random number generator. In practice, usable leftovers from previous orders are returned back to stock and used again to fill future orders. Using the simulation method instead of randomly generated usable leftovers makes it possible to acquire information about the actual amount of usable leftovers in stock. Methods for solving 1DCSPUL should be tested and compared on the basis of the simulation; otherwise it is not possible to be in line with the basic characteristic of 1DCSPUL that usable leftovers from previous orders are used in the following ones. This is overlooked in the literature since usable leftovers for each instance are generated randomly when testing and comparing various methods. Thus the majority of methods is wrongly tested and compared.

Therefore I propose the new testing method where the usable leftovers generated in the previous order are used in the current one, and the usable leftovers generated in current order in the next one, and so on. Even though such approach can be found in Trkman and Gradišar (2007) and Cherri et al. (2012) it has not yet been sufficiently researched regarding testing and comparing the methods. In the literature methods are tested on the basis of a randomly generated constant number of usable leftovers in stock (Cherri et al. (2009; Cui & Yang 2010) instead of taking the usable leftovers from previous instance into account and using them in the current one. Hence, in my doctoral dissertation for each order the computer simulation method is used.

As an illustrative example of how testing the method should be approached, I presume that the most important criterion is the number of usable leftovers in stock because a high level of usable leftovers could result in higher trim loss and inventory costs. The level of usable leftovers in stock is simulated because in reality it would grow over time if the sequence of orders generated were filled with consecutively generated shipments of standard bars.

In presented simulations COLA is selected for solving 1DCSPUL because it consumes the highest number of usable leftovers from previous orders in comparison to other methods (Cherri et al., 2009; Cui & Yang, 2010). This means that the possibility of growth of usable leftovers in stock would be greater in comparison to selection of some other method and the real life situation would be even more reflected. COLA is written in FORTRAN programming language since it allow for very fast processing. 4GL was used for data input and printout of the results. The detailed description of COLA can be found in chapter 2.

In order to demonstrate the proposed testing method, four cases were analyzed. Among them *r* varies but it is low in all four cases. The parameters for order generation are presented in Table 20. In Sequence 1 and Sequence 4, orders are generated with the problem generator CUTGEN1 (Gau & Wäscher, 1995) by using the same parameters as in Cherri et al. (2009) and Cui and Yang (2010), and the same two standard bars are in stock (1,000 and 1,100) with 100 pieces of each. Sequences 2 and 3 are added in such a way that the gap between Sequence 1 and Sequence 4 is evenly filled. The number of consecutive orders in Cherri et al. (2009) and Cui and Yang (2010) varied between 10 and 40. In my doctoral dissertation, 30 is selected. Threshold for returning leftovers back on stock is set to min l_{s_i} in all four cases.

Table 20: Parameters for order generation.

	Sequence 1	Sequence 2	Sequence 3	Sequence 4
Number of different items	20	20	20	20
Interval in which each item is situated	[5, 83]	[6, 146]	[8, 209]	[11, 335]
Number of pieces	125	102	79	34
Number of consecutive orders	30	30	30	30

Source: Cherri et al., 2009; Cui & Yang, 2010; own analysis.

4.2.1 Results

The results of testing COLA are shown in the Table 21. Beside the trim loss the total number of usable leftovers after all previously fulfilled orders is presented as well since it influences the amount of trim loss and should be therefore taken into account.

Table 21: Results of COLA.

	Sequence 1		Sequence 2		Sequence 3		Sequence 4	
Order number	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)
1	2	2	23	2	63	20	121	2
2	1	3	134	6	181	9	443	10
3	0	2	69	10	88	36	142	49
4	1	1	10	10	105	25	160	61
5	1	2	36	23	166	19	400	59
6	20	1	51	13	316	22	452	70
7	1	2	21	23	86	38	223	80
8	1	2	22	6	70	44	115	78
9	16	2	49	3	138	71	498	68
10	0	2	73	2	426	58	460	68
11	24	2	49	6	202	60	754	63
12	2	1	46	18	117	67	240	74
13	1	2	78	11	123	95	264	126
14	1	2	25	9	133	111	162	151
15	1	1	35	20	109	118	106	165
16	0	2	93	28	410	102	458	188
17	1	2	21	20	51	104	164	188
18	1	3	16	20	78	102	40	195
19	0	1	24	19	259	97	291	198
20	3	2	21	10	85	115	206	196
21	0	2	35	9	193	100	311	195
22	3	2	21	22	63	129	160	198
23	0	2	20	11	25	129	66	185
24	0	2	43	8	210	134	150	183
25	5	3	21	6	95	153	262	191
26	0	1	51	13	78	148	236	192
27	0	2	61	13	99	156	371	188
28	0	2	60	7	215	163	263	192
29	0	2	8	8	255	154	126	213
30	5	1	53	6	122	145	355	210
Sum	**90**		**1,269**		**4,561**		**7,999**	

The most important parameter regarding trim loss is the *r*. In general, a lower *r* produces higher trim loss. From Sequence 1 to Sequence 4 this ratio decreases and therefore trim loss increases from 90 in Sequence 1 to 7,999 in Sequence 4. An additional reason for increased trim loss is the high number of usable leftovers in stock. The results in the literature are not sufficiently detailed to allow the precise determination of the contribution of usable leftovers to trim loss.

The results for Sequence 1 and Sequence 2 show that usable leftovers in stock from the previous orders do not accumulate, but are used up. The accumulated number of usable leftovers after 30 instances is one in Sequence 1 and six in Sequence 2. In Sequence 2, the results show that the amount of usable leftovers in stock initially increases, but stabilizes after five instances. This indicates that the point of using usable leftovers at least at the same pace as they are produced is reached at a low level of usable leftovers. However, in Sequences 3 and 4 there is excessive generation of usable leftovers and the point where the number of usable leftovers in stock is stabilized is most likely between instances 25 and 30 in Sequence 3 and more than 30 in Sequence 4. The reason why the number of usable leftovers in stock does not grow constantly is that at a certain level of usable leftovers the number of possible cutting plans becomes so large that finding a better solution is easier and the production of usable leftovers is lower. At that level, the number of usable leftovers used and produced becomes equal.

The dependence of results on different sequences of the same set of orders is also studied. For Sequence 1 and Sequence 3, the experiment is conducted again with the same set of orders but in the opposite direction. The results are presented in the Table 22.

Table 22: Results of COLA for Sequences 1 and 2 - calculation with opposite sequence direction.

	Sequence 1		**Sequence 3**	
Order number	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)
1	5	1	113	2
2	0	1	259	2
3	2	2	188	18
4	0	1	156	17
5	0	2	78	15
6	5	3	233	22
7	0	2	28	18
8	3	2	71	45
9	0	2	203	45
10	2	2	145	62
11	0	1	364	56
12	16	2	84	50
13	1	3	32	64
14	0	2	508	69
15	6	2	124	84
16	0	2	103	79
17	2	2	122	100
18	1	2	93	103
19	0	2	225	98
20	19	1	413	89
21	0	2	180	114
22	16	2	79	117
23	0	2	97	105
24	0	1	316	107
25	7	2	132	107
26	1	2	60	120
27	6	3	108	143
28	1	2	170	142
29	1	3	87	158
30	1	2	90	148
Sum	**95**		**4,761**	

The comparison of results shows that trim loss and the number of usable leftovers in stock are very similar in both cases. In Sequence 1 the sum of trim loss is increased from 90 to 95 and the number of usable leftovers by one. In Sequence 3 the sum of trim loss is increased from 4,561 to 4,761 and the number of usable leftovers from 145 to 148. Thus it is possible to conclude that the dependence of results on different sequences of the same set of orders is so small that it can be ignored.

To demonstrate the proposed testing method I carried out and additional experiment where I have compared COLA (Gradišar et al., 1997) with ECOLA (Tomat, Gradišar & Štiglic, 2013). 2 cases were analyzed. ECOLA is also written in FORTAN and 4GL is used for data input and printout of the results. Parameters for order generation were nearly the same as for Sequence 1 and Sequence 2 and can be found in Table 20. Different sequences for the same set of orders were taken into account and 100 consecutive instances were simulated instead of 30. For testing and comparing COLA and ECOLA 100 instances are considered since such quantity is of sufficient size to demonstrate the performance of the proposed testing method. The results are presented in Table 23.

Table 23: Comparison between COLA and ECOLA.

	Sequence 1				**Sequence 2**			
	COLA		ECOLA		COLA		ECOLA	
Order number	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)	Trim loss	Usable leftovers (number of bars)
1	2	2	2	2	23	2	23	2
2	1	3	1	3	134	6	134	6
3	0	1	0	1	69	10	69	10
4	3	1	3	1	10	10	10	10
5	1	2	1	2	36	23	36	23
6	9	2	1	1	51	13	51	13
7	0	2	0	2	21	23	21	23
8	1	2	0	2	22	6	22	6
9	17	2	17	2	49	3	49	3
10	0	2	0	2	73	2	73	2
...	…	…	…	…	…	…	…	…
46	1	4	7	2	60	49	40	86
47	1	2	2	2	41	48	359	86
48	18	2	0	2	61	43	186	81
49	13	2	0	2	36	58	27	98
50	2	2	2	2	18	57	27	94
51	1	2	0	3	54	61	47	94
52	0	1	1	4	67	56	60	93
53	4	2	0	1	94	65	67	103
54	0	2	1	2	103	65	133	110
55	1	2	0	2	141	57	170	98
…	...	…	…	…	…	…	…	…
90	12	2	12	2	16	10	31	142
91	13	3	2	1	100	19	244	130
92	2	2	6	2	49	28	66	125
93	0	2	0	2	54	32	9	129
94	0	2	1	2	137	52	124	161
95	0	2	1	3	27	55	45	180
96	1	2	4	1	29	58	18	185
97	3	2	2	2	63	62	37	197
98	6	1	5	4	89	75	92	208
99	0	1	14	3	35	43	33	178
100	0	2	0	2	17	51	22	188
Sum	**351**		**332**		**5,124**		**6,235**	

The results for Sequence 1 and Sequence 2 point out the efficiency of tested methods in dependence on *r*. In Sequence 1, where *r* is higher, both methods result in 2 usable leftovers in the stock after fulfilling 100 orders. In Sequence 2, where *r* is lower, the tested methods provide results that differ significantly. The quantity of usable leftovers in stock after 100 successive instances equals to 51 pieces with COLA and 188 pieces with ECOLA. So COLA outperforms ECOLA for 269%.

With simulation the motion of the usable leftovers in stock can be precisely observed. There is no excessive accumulation of usable leftovers in Sequence 1 although there are some minor increases in growth, i.e. in 46th instance. In Sequence 2 the amount of usable leftover starts to increase immediately after first instance. Then it slightly falls after 7th instance but it again starts to grow after 11th instance. After 100 instances there are many usable leftovers produced so their excessive accumulation in stock is not prevented.

In Sequence 1 the trim loss is 5.5% lower using ECOLA. In Sequence 2 COLA provided better results. Trim loss amounts 5,124, which is 18% lower in comparison with ECOLA, where trim loss equals 6,235.

Based on the presented demonstration of proposed testing method it is possible to conclude that the use of ECOLA is more reasonable for problems, where *r* is higher since it results in smaller trim loss. COLA is more appropriate for solving problems, where *r* is lower.

A comparison where successive instances are simulated reflects the real situation much better than if the usable leftovers would be randomly generated for each individual instance. Testing and comparing the methods on the basis of simulation enables a decision-maker better understanding of specific situation, i.e., an accumulation of usable leftovers in stock over time. Thus it is possible to make better decision when selecting the most appropriate method.

The proposed testing method can be regarded as more comprehensive than other testing methods in the literature since it provides realistic results due to taking the whole group of successive instances into account instead of considering only an individual instance. It also assures valuable additional information about the amount of usable leftovers in stock after a certain number of periods.

5 CONCLUSION AND OUTLOOK

The doctoral dissertation considers the CSP, which is a well-known problem in many industries. A special type of this problem was addressed. Leftovers that are longer than some predefined threshold are returned back to stock to be used again in the future. This is known as the 1DCSPUL.

When solving the 1DCSPUL, the most important element is usually trim loss reduction. The amount of trim loss depends on *r*. If *r* is low, then the number of possible solution is diminished and the possibility of a solution with a larger trim loss increases (Gradišar et al., 1999a). Therefore, improvement is still possible. For this reason, I developed a new optimization method for solving 1DCSPUL with low *r*.

To determine the efficiency of the method for solving 1DCSPUL, it should be compared with existing methods. Testing could be done by using the same criteria and benchmark instances, but the results would not be realistic because the main issue is ignored. Instead of individual consideration, an entire group of instances should be taken into account so that the usable leftovers from previous instances are used to fill successive orders. Therefore a new testing method is proposed based on the use of computer simulation in which usable leftovers from previous orders are used in the current one and the usable leftovers produced in the current order are used in the next one, and so on. Thus information about actual amount of usable leftovers in stock is acquired.

In this chapter I state the main findings and describe their importance for companies. Then the contribution of my doctoral dissertation to the relevant field of knowledge is presented, followed by the discussion subchapter. At the end I offer some outlook to further research.

5.1 Main findings

In presented doctoral dissertation I have addressed the problem of trim loss reduction in 1DCSPUL the case of low *r*. A new optimization heuristic method LCUT has been developed and explained into details. To compare results of LCUT with comparable existing methods a practical case was selected. In comparison with RGR and CUT the performance of LCUT was higher since the obtained results were slightly better. To conduct an additional and more extensive comparison CUT was selected due to its ability of combining it with LCUT. Because neither LCUT nor CUT depends on commercial software they can be easily coded and integrated into various computer platforms that are used in practice. Due to simple implementation into the company information system presented method also provides the opportunity for company to adjust to changing business processes.

The performance of LCUT can be seen from testing and comparison with CUT. Results obtained with LCUT consisted of significantly less usable leftovers. A larger number of usable leftovers affects the increase in the amount of shorter stock lengths, which in general leads to higher trim loss or even unsolvable problems in later time periods (Trkman & Gradišar, 2007). Therefore two extreme situations in the terms of the reprocessing usable leftovers costs were taken into account. In Case 1 those cost are negligible. LCUT obtained slightly worse results than CUT but when those two methods were combined the savings amounted 0.1% of used stock. In Case 2 the cost of preprocessing usable leftovers are too high for usable leftovers to be put back on stock. In such situation LCUT obtain significantly better results than other methods. In combination of LCUT and CUT the savings are 0.7% of used stock. The results suggest that is it useful to combine LCUT with CUT although LCUT can be also used as a stand-alone application.

Based on the above explanations the first goal, which claimed that a new method for solving the 1DCSPUL with a low *r* would be developed that will provide better results than existing methods, has been reached.

The second part of the doctoral dissertation concerns improving current methods of testing solutions to 1DCSPUL with the use of computer simulation. I proposed a new testing method based on computer simulation, in which usable leftovers from previous orders are used in the current one and the usable leftovers produced in the current order are used in the next one, and so on.

To show the effectiveness of proposed testing method COLA was selected and four cases with different *r* were analyzed. To allow comparison the parameters for conducting the experiment were partially the same as in the literature. An amount of usable leftovers in stock was observed since that is one of the most important criteria in the terms of trim loss reduction. The *r* from Sequence 1 to Sequence 4 decreases therefore the amount of usable leftovers in stock increases which influences the trim loss to be higher. The computational results are based on several tested problem instances. To verify the experiment I have also studied the dependence of the results on different sequences of the same set of orders. The results indicate that dependence is so small that can be ignored. To demonstrate the proposed testing method an additional experiment was carried out. For testing and comparing COLA and ECOLA 2 cases were analyzed and 100 consecutive instances were simulated. Such approach reflects the real situation better than the methods in the literature where usable leftovers are generated randomly instead of being taken from previous orders.

The proposed method is not limited to a single order but analyzes the criteria in consecutive orders. In the experiment, the method enabled better control of usable leftovers. The number of usable leftovers in stock is one of the most important elements that influences the amount of trim loss. A computer simulation was used in which usable leftovers from a previous order are used in the next one instead of being randomly generated.

Based on the above explanation the second goal, which claimed that the current methods for testing solutions to the 1DCSPUL can be improved with the simulation method, has been reached.

5.2 Importance of results for companies

Cutting activities are becoming an increasingly important part of SCM, especially in the steel, metal, wood, and transportation industries. To perform a cutting process in the most effective manner, it is necessary to apply methods that allow the minimal trim loss. Thus the company's cost can be reduced. With respect to trim loss, LCUT performs better than the existing method for solving the 1DCSPUL with a low *r*.

When selecting a method for solving the 1DCPSUL, it is important for the method to perform best when solving problems like those that occur at a company. In general, the selection depends on testing the potential methods. The testing method can thus help companies to select the most appropriate method for their business needs. This reduces the total cost to the company.

5.3 Contribution of the doctoral dissertation to the relevant field of knowledge

My contribution to the relevant field of knowledge is a new optimization method for solving the 1DCSPUL with low *r*. It is not possible to find any similar method in the literature that takes such an *r* into account. Therefore, the method presented represents a significant theoretical contribution. The use of the proposed method improves the possibilities for better decision-making and thus represents a contribution to the relevant field of knowledge.

A newly proposed method for testing solutions to the 1DCSPUL allows for real situations to be better replicated than possible with methods from the literature.

The doctoral dissertation contributes to further development of research in computer simulation, cutting and packing problems, optimization, and operations research.

5.4 Discussion and limitations

A few issues need to be discussed regarding the matters studied.

The proposed LCUT method for solving 1DCSPUL with low *r* has some constraints that were explained in subchapter 3.3. In relation to these constraints, a cutting plan is created in less than 10 seconds using a personal computer. If those constraints were different, that would influence the time needed for cutting plan creation. Depending on tightening or mitigating the constraints, the performance of the method in practice would change.

The time duration for which it is still acceptable to find a solution depends on the nature of the business and on the decision-maker. If the cutting process represents only a marginal activity then the overall time needed for preparing a cutting plan probably does not play such an important role. When testing whether the problems presented in the doctoral dissertation could be solved using an exact method, a certain acceptable time limit was set. If that time were prolonged it is possible that an optimal solution would be found. However, in industry the solutions should usually be obtained within a few seconds and therefore the use of an exact method for solving large problems would not be appropriate. If the problems are very small or the time limit infinite, then the decision-maker should consider using an exact approach.

When proving the efficiency of LCUT, I conducted an experiment with the same data but using two other methods (RGR and CUT). These methods were selected for comparison because it was found in the literature that they perform best for similar problems, as discussed in the doctoral dissertation (Gradišar et al., 1999a; Cherri et al., 2009). It is possible that for different situations (e.g., a different size of the problem, higher r, etc.) some other methods could perform better and should therefore be considered in comparison.

For problem generation, the PGEN generator with minor modifications was used. For a variety of large objects, some other problem generator could be applied. Thus, the results could differ from the one presented. PGEN was used due to the characteristics of the real data obtained from the retail company.

In chapter 4, the simulation method was presented as a testing method. When demonstrating the procedure, the threshold for returning leftovers back to stock was set to min ls_i in all experiments. Such a threshold was selected due to many examples from the literature (Cherri et al., 2009; Cui & Yang, 2010), but in a practical situation it depends on the decision-maker. The applicants should bear this in mind when implementing the proposed method.

When presenting the simulation approach as a testing method, a specific number of consecutive orders was taken into account. Such number was selected to be comparable with the cases presented in the literature (Cherri et al., 2009; Cui & Yang, 2010). In practice, this number depends on the frequency of cutting. That is, if the company has to fill many orders in a short time period, then that number should be higher to allow better understanding of what is happening with the amount of usable leftovers in stock.

5.5 Outlook to further research

The proposed method for solving the 1DCSPUL with low r is intended for solving one-dimensional problems. A similar approach could also be developed for solving two- or three-dimensional problems.

Next, the possibility of including some additional costs into the LCUT should be considered; for example, warehouse costs. Another possibility is to combine the minimization of trim loss with the minimization of costs, such as in Erjavec et al. (2009), in which the focus would not only be the cutting activity but also other connected activities that form the processes of procuring material (Trkman & McCormack, 2010).

Regarding the proposed method for testing solutions to the 1DCSPUL, it would be useful if other existing methods were also adapted to the sequence of orders.

As shown by the experiments, *r* influences the amount of usable leftovers in stock. It would be valuable to conduct an experiment with more successive order instances and observe the growth of usable leftovers in stock by different *r*. It is possible that in some cases those usable leftovers would begin to accumulate. This would result in ever-growing stock, which is not acceptable due to the high warehouse and logistics costs.

REFERENCES

1. Abuabara, A., & Morabito, R. (2009). Cutting optimization of structural tubes to build agricultural light aircrafts. *Annals of Operations Research, 169*(1), 149-165.
2. Aekell, J. (1872). *Improvement in pasting apparatus for paper-bag machines*: U.S. Patent No. 133,395. Washington, DC: U.S. Patent and Trademark Office.
3. Aktin, T., & Özdemir, R. G. (2009). An integrated approach to the one-dimensional cutting stock problem in coronary stent manufacturing. *European Journal of Operational Research, 196*(2), 737-743.
4. Alba, E., Blum, C., Asasi, P., Leon, C., & Gomez, J. A. (2009). *Optimization techniques for solving complex problems* (Vol. 76): Wiley.
5. Alem, D. J., Munari, P. A., Arenales, M. N., & Ferreira, P. A. V. (2010). On the cutting stock problem under stochastic demand. *Annals of Operations Research, 179*(1), 169-186.
6. Alfieri, A., van de Velde, S., & Woeginger, G. J. (2007). Roll cutting in the curtain industry, or: A well-solvable allocation problem. *European Journal of Operational Research, 183*(3), 1397-1404.
7. Alves, C., & Valério de Carvalho, J. M. (2007). Accelerating column generation for variable sized bin-packing problems. *European Journal of Operational Research, 183*(3), 1333-1352.
8. Alves, C., & Valério de Carvalho, J. M. (2008). A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Computers & Operations Research, 35*(4), 1315-1328.
9. Antonio, J., Chauvet, F., Chu, C., & Proth, J.-M. (1999). The cutting stock problem with mixed objectives: Two heuristics based on dynamic programming. *European Journal of Operational Research, 114*(2), 395-402.
10. Antoniou, A., & Lu, W. S. (2007). *Practical optimization: algorithms and engineering applications*: Springer.
11. Araujo, S. A., Constantino, A. A., & Poldi, K. C. (2011). An evolutionary algorithm for the one-dimensional cutting stock problem. *International Transactions in Operational Research, 18*(1), 115-127.
12. Arbib, C., & Marinelli, F. (2005). Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application. *European Journal of Operational Research, 163*(3), 617-630.
13. Arbib, C., Marinelli, F., Rossi, F., & Di Iorio, F. (2002). Cutting and reuse: an application from automobile component manufacturing. *Operations Research, 50*(6), 923-934.
14. Armbruster, M. (2002). A solution procedure for a pattern sequencing problem as part of a one-dimensional cutting stock problem in the steel industry. *European Journal of Operational Research, 141*(2), 328-340.

15. Assad, A. A., & Gass, S. I. (2011). *Profiles in Operations Research: Pioneers and Innovators*. New York: Springer.
16. Banks, J. (2005). *Discrete Event System Simulation, 4/e*: India: Pearson Education.
17. Bartholdi III, J. J. (2008). The Knapsack Problem *Building Intuition* (pp. 19-31): Springer.
18. Baskan, O., Haldenbilen, S., & Ceylan, H. (2012). Estimating transport energy demand using ant colony optimization. *Energy Sources, Part B: Economics, Planning, and Policy, 7*(2), 188-199.
19. Bead, C. O. (1846). I*mprovement in machines for manufacturing wood-screws*: U.S. Patent No. RE88. Washington, DC: U.S. Patent and Trademark Office.
20. Belov, G., & Scheithauer, G. (2002). A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *European Journal of Operational Research, 141*(2), 274-294.
21. Belov, G., & Scheithauer, G. (2006). A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *European Journal of Operational Research, 171*(1), 85-106.
22. Belov, G., & Scheithauer, G. (2007). Setup and Open-Stacks Minimization in One-Dimensional Stock Cutting. *INFORMS Journal of Computing, 19*(1), 27-35.
23. Benedettini, O., & Tjahjono, B. (2009). Towards an improved tool to facilitate simulation modelling of complex manufacturing systems. *The International Journal of Advanced Manufacturing Technology, 43*(1-2), 191-199.
24. Beraldi, P., Bruni, M., & Conforti, D. (2009). The stochastic trim-loss problem. *European Journal of Operational Research, 197*(1), 42-49.
25. Bigelow, M. B. (1858). *Machine fob cutting paper*: U.S. Patent No. 20,858. Washington, DC: U.S. Patent and Trademark Office.
26. Bingul, Z., & Oysu, C. (2005). Comparison of stochastic and approximation algorithms for one-dimensional cutting problems *Advances in intelligent computing* (pp. 976-985): Springer.
27. Bischoff, E. E., & Wäscher, G. (1995). Cutting and packing. *European Journal of Operational Research, 84*(3), 503-505.
28. Borovicka, A., Hlavacek, P., Hloch, S., & Valicek, J. (2012). *A method for determining technology parameters for the abrasive waterjet cutting of materials*: European Patent No. EP 2409812. Munich, Germany: European Patent Office.
29. Boyd, S., & Mattingley, J. (2007). Branch and Bound Methods *Lectures*. Palo Alto: Standford University.
30. Bradt, C. D. (1907). Machine for cutting paper: U.S. Patent No. 840,553. Washington, DC: U.S. Patent and Trademark Office.
31. Brooks, R. L., Smith, C., Stone, A., & Tutte, W. T. (1987). The dissection of rectangles into squares. *Classic Papers in Combinatorics*, 88-116.

32. Brown, A. R. (1971). *Optimum Packing and Depletion: The Computer in Space and Resource Usage Problems*. New York: American Elsevier Publishing Company.
33. Busetti, F. (2003). *Simulated annealing overview*. Retrieved April 24, 2013, from http://163.18.62.64/wisdom/Simulated%20annealing%20overview.pdf
34. Cai, M. B., Li, X. P., & Rahman, M. (2007). Study of the mechanism of nanoscale ductile mode cutting of silicon using molecular dynamics simulation. *International Journal of Machine Tools and Manufacture, 47*(1), 75-80.
35. Cali, D. S., & Myers, K. E. (2011). *Method of cutting material for use in implantable medical device*: U.S. Patent No. 20,130,055,940. Washington, DC: U.S. Patent and Trademark Office.
36. Caprara, A., Carvalho, M., Lodi, A., & Woeginger, G. J. (2013). A complexity and approximability study of the bilevel knapsack problem *Integer Programming and Combinatorial Optimization* (pp. 98-109): Springer.
37. Carnahan, M. (1888). Machine for cutting I-beams: U.S. Patent No. 392,662. Washington, DC: U.S. Patent and Trademark Office.
38. Ceretti, E., Lucchi, M., & Altan, T. (1999). FEM simulation of orthogonal cutting: serrated chip formation. *Journal of Materials Processing Technology, 95*(1–3), 17-26.
39. Cerqueira, G. R. L., & Yanasse, H. H. (2009). A pattern reduction procedure in a one-dimensional cutting stock problem by grouping items according to their demands. *Journal of Computational Interdisciplinary Sciences, 1*(2), 159-164.
40. Cesar, M. (2013). *One-dimensional Cutting Stock Optimization by Groups.* Ljubljana: Faculty of Economics.
41. Chan, F. T., & Chan, H. (2005). The future trend on system-wide modelling in supply chain studies. *The International Journal of Advanced Manufacturing Technology, 25*(7-8), 820-832.
42. Chauhan, S. S., Martel, A., & D’Amour, S. (2008). Roll assortment optimization in a paper mill: An integer programming approach. *Computers & Operations Research, 35*(2), 614-627.
43. Cherri, A. C., Arenales, M. N., & Yanasse, H. H. (2009). The one-dimensional cutting stock problem with usable leftover–A heuristic approach. *European Journal of Operational Research, 196*(3), 897-908.
44. Cherri, A. C., Arenales, M. N., & Yanasse, H. H. (2012). The usable leftover one-dimensional cutting stock problem—a priority-in-use heuristic. *International Transactions in Operational Research*.
45. Cherri, A. C., Arenales, M. N., Yanasse, H. H., Poldi, K. C., & Vianna, A. C. G. (2013). The One-dimensional Cutting Stock Problem with Usable Leftover - A Survey. *European Journal of Operational Research, In press.*

46. Cherri, A. C., Junior, D. J. A., & da Silva, I. N. (2011). Interferencia Fuzzy para o Problema de Corte de Estoque com Sobras Aproveitaveis de Material. *Pesquisa Operacional, 31*(1), 173-194.
47. Chu, C., & Antonio, J. (1999). Approximation algorithms to solve real-life multicriteria cutting stock problems. *Operations Research, 47*(4), 495-508.
48. Coddington, E. L. (1943). *Machine for cutting grass*: U.S. Patent No. 2,329,185. Washington, DC: U.S. Patent and Trademark Office.
49. Cohek, P. (1875). *Improvement in machines for cutting roll-paper*: U.S. Patent No. 167,645. Washington, DC: U.S. Patent and Trademark Office.
50. Colicchia, C., & Strozzi, F. (2012). Supply chain risk management: a new methodology for a systematic literature review. *Supply Chain Management: An International Journal, 17*(4), 403-418.
51. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3 ed.). Cambridge, Massachusetts: The MIT Press.
52. Cornell, J. M. (1872). Improvement in machines for shearing the ends of beams U.S. Patent No. RE5,097. Washington, DC: U.S. Patent and Trademark Office.
53. Crowder, H. P. (2000). The Science of Smart Decisions. *OR/MS Today, 27*(3).
54. Cui, Y. D. (2005). A cutting stock problem and its solution in the manufacturing industry of large electric generators. *Computers & Operations Research, 32*(7), 1709-1721.
55. Cui, Y. D., Chen, Y., & Wu, J. (2006). Selecting the best sheet length for the steel stock used in circular blank production. *IIE Transactions, 39*(10), 829-836.
56. Cui, Y. D., Zhao, X., Yang, Y., & Yu, P. (2008). A heuristic for the one-dimensional cutting stock problem with pattern reduction. *Journal of Engineering Manufacture, 222*(6), 677-685.
57. Cui, Y., & Yang, Y. (2010). A heuristic for the one-dimensional cutting stock problem with usable leftover. *European Journal of Operational Research, 204*(2), 245-250.
58. Cus, F., & Balic, J. (2003). Optimization of cutting process by GA approach. *Robotics and Computer-Integrated Manufacturing, 19*(1–2), 113-121.
59. Čižman, A. (2004). *Operacijske raziskave.* Kranj: Moderna organizacija.
60. Čižman, A., & Černetič, J. (2004). Improving competitiveness in veneers production by a simple-to-use DSS. *European Journal of Operational Research, 156*(1), 241-260.
61. Darwin, C. (1859). On the origins of species by means of natural selection. *London: Murray*.
62. Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. (2006). *Algorithms*: McGraw-Hill Science/Engineering/Math; 1 edition
63. de Carvalho, J. (1998). Exact Solution of Cutting Stock Problems Using Column Generation and Branch-and-Bound. *International Transactions in Operational Research, 5*(1), 35-44.

64. de la Banda, M. G., & Stuckey, P. J. (2007). Dynamic programming to minimize the maximum number of open stacks. *INFORMS Journal on Computing, 19*(4), 607-617.
65. Degraeve, Z., & Peeters, M. (2003). Optimal integer solutions to industrial cutting-stock problems: Part 2, benchmark results. *INFORMS Journal on Computing, 15*(1), 58-81.
66. Degraeve, Z., & Schrage, L. (1999). Optimal integer solutions to industrial cutting stock problems. *INFORMS Journal on Computing, 11*(4), 406-419.
67. Degraeve, Z., & Vandebroek, M. (1998). A mixed integer programming model for solving a layout problem in the fashion industry. *Management Science, 44*(3), 301-310.
68. Diegel, A., Montocchio, E., Walters, E., Van Schalkwyk, S., & Naidoo, S. (1996). Setup minimising conditions in the trim loss problem. *European Journal of Operational Research, 95*(3), 631-640.
69. Dimitriadis, S., & Kehris, E. (2009). Cutting stock process optimisation in custom door and window manufacturing industry. *International Journal of Decision Sciences, Risk and Management, 1*(1), 66-80.
70. Dougherty, A. (1862). *Improvement in paper-cutting machinces*: U.S. Patent No. 35,592. Washington, DC: U.S. Patent and Trademark Office.
71. Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research, 44*(2), 145-159.
72. Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical programming, 1*(1), 127-136.
73. Eiselt, H. A., & Sandblom, C.-L. (2000). *Integer programming and network models*: Springer.
74. Eisemann, K. (1957). The trim problem. *Management Science, 3*(3), 279-284.
75. Erjavec, J. (2011). *Analysis of the cutting stock problem as a part of a business process.* Doctoral thesis, Ljubljana: Faculty of Economics.
76. Erjavec, J., Gradišar, M., & Trkman, P. (2009). Renovation of the cutting stock process. *International Journal of Production Research, 47*(14), 3979-3996.
77. Faggioli, E., & Bentivoglio, C. A. (1998). Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research, 110*(3), 564-575.
78. Foerster, H., & Wascher, G. (2000). Pattern reduction in one-dimensional cutting stock problems. *International Journal of Production Research, 38*(7), 1657-1676.
79. Fowler, A. (1998). Operations management and systemic modelling as frameworks for BPR. *International Journal of Operations & Production Management, 18*(9/10), 1028-1056.
80. Gage, J. P. (1848). *Improvement in machinery for sanding paper*: U.S. Patent No. 5,443. Washington, DC: U.S. Patent and Trademark Office.

81. Gang, F., & Pan, Z. (2001). Advances in numerical simulation technology for cutting process [J]. *Advances In Mechanics, 3*, 006.
82. Gau, T., & Wäscher, G. (1995). CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem. *European Journal of Operational Research, 84*(3), 572-579.
83. Ghodsi, R., & Sassani, F. (2005a). Online cutting stock optimization with prioritized orders. *Assembly Automation, 25*(1), 66-72.
84. Ghodsi, R., & Sassani, F. (2005b). Real-time optimum sequencing of wood cutting process. *International Journal of Production Research, 43*(6), 1127-1141.
85. Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the social scientist*: McGraw-Hill International.
86. Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting stock problem. *Operations Research, 9*(6), 849-859.
87. Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem—Part II. *Operations Research, 11*(6), 863-888.
88. Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research, 13*, 533-549.
89. Glover, F., & Laguna, M. (1997). *Tabu Search* (1 ed.). Massachusetts: Kluwer Academic Publishers.
90. Golfeto, R., Moretti, A. C., & de Salles Neto, L. L. (2009). A genetic symbiotic algorithm applied to the cutting stock problem with multiple objectives. *Advanced Modeling and Optimization, 11*, 473-501.
91. Goulimis, C. (1990). Optimal solutions for the cutting stock problem. *European Journal of Operational Research, 44*(2), 197-208.
92. Grace, R. W. (1892). Machine for cutting right angled structural shapes U.S. Patent No. 479,066. Washington, DC: U.S. Patent and Trademark Office.
93. Gradišar, M. (2008). *One-dimensional cutting stock optimization problem: the case of low ratio between stock and order lenghts (Working paper)*. Ljubljana: Research Center of the Faculty of Economics.
94. Gradišar, M., & Jesenko, J. (1996). Optimization of one dimensional cutting in clothing industry. *Informatica(Ljubljana), 20*(2), 211-221.
95. Gradišar, M., & Trkman, P. (2005). A combined approach to the solution to the general one-dimensional cutting stock problem. *Computers & Operations Research, 32*(7), 1793-1807.
96. Gradišar, M., Erjavec, J., & Tomat, L. (2011). One-dimensional cutting stock optimization with usable leftover: A case of low stock-to-order ratio. *International Journal of Decision Support System Technology (IJDSST), 3*(1), 54-66.
97. Gradišar, M., Jesenko, J., & Resinovič, G. (1997). Optimization of roll cutting in clothing industry. *Computers & Operations Research, 24*(10), 945-953.

98. Gradišar, M., Kljajić, M., Resinovič, G., & Jesenko, J. (1999a). A sequential heuristic procedure for one-dimensional cutting. *European Journal of Operational Research, 114*(3), 557-568.
99. Gradišar, M., Resinovič, G., & Kljajić, M. (1999b). A hybrid approach for optimization of one-dimensional cutting. *European Journal of Operational Research, 119*(3), 719-728.
100. Gradišar, M., Resinovič, G., & Kljajić, M. (2002). Evaluation of algorithms for one-dimensional cutting. *Computers & Operations Research, 29*(9), 1207-1220.
101. Gramani, M. C. N., & França, P. M. (2006). The combined cutting stock and lot-sizing problem in industrial processes. *European Journal of Operational Research, 174*(1), 509-521.
102. Haessler, R. W., & Sweeney, P. E. (1991). Cutting stock problems and solution procedures. *European Journal of Operational Research, 54*(2), 141-150.
103. Hajizadeh, I., & Lee, C. G. (2007). Alternative configurations for cutting machines in a tube cutting mill. *European Journal of Operational Research, 183*(3), 1385-1396.
104. Hammond, D. (1894). Shape-metal-cutting machine: U.S. Patent No. 516,736. Washington, DC: U.S. Patent and Trademark Office.
105. Hatch, J. (1869). *Improvement in variety moulding-machine*: U.S. Patent No. 92,609. Washington, DC: U.S. Patent and Trademark Office.
106. Haupt, R. L., & Haupt, S. E. (2004). *Practical genetic algorithms*: John Wiley & Sons.
107. Hicks, D., Breitzman, T., Olivastro, D., & Hamilton, K. (2001). The changing composition of innovative activity in the US—a portrait based on patent analysis. *Research policy*, *30*(4), 681-703.
108. Hinxman, A. (1980). The trim-loss and assortment problems: a survey. *European Journal of Operational Research, 5*(1), 8-18.
109. Hodge, J. N. (1861). *Machine for sawing and cutting wood*: U.S. Patent No. 32,945. Washington, DC: U.S. Patent and Trademark Office.
110. Holland, J. H. (1992). Genetic algorithms. *Scientific american, 267*(1), 66-72.
111. Holthaus, O. (2002). Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths. *European Journal of Operational Research, 141*(2), 295-312.
112. Holthaus, O. (2003). On the best number of different standard lengths to stock for one-dimensional assortment problems. *International Journal of Production Economics*, *83*(3), 233-246.
113. Horowitz, E., & Sahni, S. (1974). Computing partitions with applications to the knapsack problem. *Journal of the ACM (JACM), 21*(2), 277-292.
114. Houlihan, J. B. (1988). International supply chains: a new approach. *Management Decision, 26*(3), 13-19.

115. Iori, M., Salazar-González, J.-J., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science, 41*(2), 253-264.
116. Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research, 203*(1), 1-13.
117. Jansen-Vullers, M., & Netjes, M. (2006). *Business process simulation–a tool survey*. Paper presented at the Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark.
118. Johnston, R. E., & Sadinlija, E. (2004). A new model for complete solutions to one-dimensional cutting stock problems. *European Journal of Operational Research, 153*(1), 176-183.
119. Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 366-422.
120. Keane, J. (1838). *Machine for cutting wood-screws*: U.S. Patent No. 830. Washington, DC: U.S. Patent and Trademark Office.
121. Kleijnen, J. P., & Smits, M. T. (2003). Performance metrics in supply chain management. *Journal of the Operational Research Society, 54*(5), 507-514.
122. Kljajić, M., Bernik, I., & Škraba, A. (2000). Simulation approach to decision assessment in enterprises. *Simulation, 75*(4), 199-210.
123. Klocke, F., Raedt, H.-W., & Hoppe, S. (2001). 2D-FEM simulation of the orthogonal high speed cutting process.
124. Koch, S., König, S., & Wäscher, G. (2009). Integer linear programming for a cutting problem in the wood-processing industry: a case study. *International Transactions in Operational Research, 16*(6), 715-726.
125. Koehler, J. (1890). Machine for cutting I-beams or channel-bars: U.S. Patent No. 438,252. Washington, DC: U.S. Patent and Trademark Office.
126. Kofjač, D., Kljajić, M., & Rejec, V. (2009). The anticipative concept in warehouse optimization using simulation in an uncertain environment. *European Journal of Operational Research, 193*(3), 660-669.
127. Komanduri, R., Chandrasekaran, N., & Raff, L. M. (2000). M.D. Simulation of nanometric cutting of single crystal aluminum–effect of crystal orientation and direction of cutting. *Wear, 242*(1–2), 60-88.
128. Kos, L., & Duhovnik, J. (2002). Cutting optimization with variable-sized stock and inventory status data. *International Journal of Production Research, 40*(10), 2289-2301.
129. Koza, J. R., & Poli, R. (2005). Genetic programming *Search Methodologies* (pp. 127-164): Springer.
130. Krichagina, E. V., Rubio, R., Taksar, M. I., & Wein, L. M. (1998). A dynamic stochastic stock-cutting problem. *Operations Research, 46*(5), 690-701.

131. Leão, A. A. S., Santos, M. O., Hoto, R., & Arenales, M. N. (2011). The constrained compartmentalized knapsack problem: mathematical models and solution methods. *European Journal of Operational Research, 212*(3), 455-463.
132. Lee, J. (2007). In situ column generation for a cutting-stock problem. *Computers & Operations Research, 34*(8), 2345-2358.
133. Leon, N. (1944). *Woodcutting machine*: U.S. Patent No. 2,361,147. Washington, DC: U.S. Patent and Trademark Office.
134. Li, K., Gao, X. L., & Sutherland, J. W. (2002). Finite element simulation of the orthogonal metal cutting process for qualitative understanding of the effects of crater wear on the chip formation process. *Journal of Materials Processing Technology, 127*(3), 309-324.
135. Liang, K.-H., Yao, X., Newton, C., & Hoffman, D. (2002). A new evolutionary approach to cutting stock problems with and without contiguity. *Computers & Operations Research, 29*(12), 1641-1659.
136. Little, J. D. C. (2012). *Branch and bound methods for combinatorial problems* Suva: Ulan Press
137. Liu, K., & Melkote, S. N. (2007). Finite element analysis of the influence of tool edge radius on size effect in orthogonal micro-cutting process. *International Journal of Mechanical Sciences, 49*(5), 650-660.
138. Liu, S., Duffy, A. H. B., Whitfield, R. I., Boyle, I. M., & McKenna, I. (2009). Towards the realization of an integrated decision support environment for organizational decision making. *International Journal of Decision Support System Technology (IJDSST), 1*(4), 38-58.
139. Martello, S., & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*: John Wiley & Sons, Inc.
140. Mathews, G. (1896). On the partition of numbers. *Proceedings of the London Mathematical Society, 1*(1), 486-490.
141. Mazouz, A.-K., Slutzky, G. D., Hall, E. L., Shell, R. L., & Huston, R. L. (1992). *Method and apparatus for palletizing randomly arriving mixed size and content parcels*: U.S. Patent No. 5,175,692. Washington, DC: U.S. Patent and Trademark Office.
142. Menon, S., & Schrage, L. (2002). Order allocation for stock cutting in the paper industry. *Operations Research, 50*(2), 324-332.
143. Mills, A. J., Eurepos, G., & Wiebe, E. (2010). *Encyclopedia of Case Study Research* (vol. 1). California: SAGE publications.
144. Morgan, L. O., Morton, A. R., & Daniels, R. L. (2006). Simultaneously determining the mix of space launch vehicles and the assignment of satellites to rockets. *European Journal of Operational Research, 172*(3), 747-760.
145. Movahhedy, M., Gadala, M. S., & Altintas, Y. (2000). Simulation of the orthogonal metal cutting process using an arbitrary Lagrangian–Eulerian finite-element method. *Journal of Materials Processing Technology, 103*(2), 267-275.

146. Neumair, J. A. (1940). *Cigar enveloping machine*: U.S. Patent No. 2,195,222. Washington, DC: U.S. Patent and Trademark Office.
147. Norberg Ohlsson, M. (2012). *Method of, system and computer program for machine cutting several parts of a piece of material using controlling rules and variables for cutting*: WIPO Patent No. 2012136262. Geneva, Switzerland: World Intellectual Property Organization.
148. Norros, I. (1995). On the use of fractional Brownian motion in the theory of connectionless networks. *Selected Areas in Communications, IEEE Journal on, 13*(6), 953-962.
149. North, A. E. (1886). Machine for cutting stone: U.S. Patent No. 336,256. Washington, DC: U.S. Patent and Trademark Office.
150. Özel, T., & Altan, T. (2000). Process simulation using finite element method — prediction of cutting forces, tool stresses and temperatures in high-speed flat end milling. *International Journal of Machine Tools and Manufacture, 40*(5), 713-738.
151. Paull, A. (1956). Linear programming: A key to optimum newsprint production. *Pulp and Paper Magazine of Canada, 57*(1), 85-90.
152. Pidd, M. (2004). *Computer Simulation in Management Science* (5 ed.). Chichester: John Wiley & Sons.
153. Pierson, J. H. (1836). *Machine for cutting and heading wire for manufacturing wood*: : U.S. Patent No. 92. Washington, DC: U.S. Patent and Trademark Office.
154. Pisinger, D. (1995). Algorithms for knapsack problems.
155. Poldi, K. C., & Arenales, M. N. (2005). Dealing with small demand in integer cutting stock problems with limited different stock lengths. *Notes of the ICMC – Series Computation, 85*, ICMC – USP, São Carlos, SP, Brazil.
156. Poldi, K. C., & Arenales, M. N. (2009). Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. *Computers & Operations Research, 36*(6), 2074-2081.
157. Poldi, K. C., & Arenales, M. N. (2010). O problema de corte de estoque unidimensional multiperíodo. *Pesquisa Operacional, 30*(1), 153-174.
158. Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2007). Genetic programming: An introductory tutorial and a survey of techniques and applications. *University of Essex, UK, Tech. Rep. CES-475*.
159. Pomerleau, R. J., Vivirito, J. R., & Markowitz, I. (1998). *Method for cutting sheet material*: U.S. Patent No. 5,727,433. Washington, DC: U.S. Patent and Trademark Office.
160. Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality* (Vol. 703): John Wiley & Sons.
161. Quiza Sardiñas, R., Rivas Santana, M., & Alfonso Brindis, E. (2006). Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering Applications of Artificial Intelligence, 19*(2), 127-133.

162. Ragsdale, C. T., & Zobel, C. W. (2004). The Ordered Cutting Stock Problem. *Decision Sciences, 35*(1), 83-100.
163. Ravelo, S. V., de Meneses, C. N., & dos Santos, M. O. (2011). Mathematical Programming Models for the one-dimensional cutting stock problem with usable leftover. *SBPO, 62*(1), 11.
164. Read, C. O. (1837). *Improved machine for manufacturing wood-screws*: U.S. Patent No. 516. Washington, DC: U.S. Patent and Trademark Office.
165. Reinertsen, H., & Vossen, T. W. (2010). The one-dimensional cutting stock problem with due dates. *European Journal of Operational Research, 201*(3), 701-711.
166. Rhodes, J. F., Dziedzic, C. J., & Beatty, R. L. (1988). *Method of cutting using silicon carbide whisker reinforced ceramic cutting tools*: U.S. Patent No. 4,789,277. Washington, DC: U.S. Patent and Trademark Office.
167. Rice, V. L. (1988). *Dimensional cutting tool for wood*: U.S. Patent No. 4,777,854. Washington, DC: U.S. Patent and Trademark Office.
168. Robinson, R. (2000). A business executive's guide to modern OR. *OR/MS Today, 27*(3).
169. Rodríguez, M. A., & Vecchietti, A. (2008). Enterprise optimization for solving an assignment and trim-loss non-convex problem. *Computers & Chemical Engineering, 32*(11), 2812-2822.
170. Rönnqvist, M. (1995). A method for the cutting stock problem with different qualities. *European Journal of Operational Research, 83*(1), 57-68.
171. Roodman, G. M. (1986). Near-optimal solutions to one-dimensional cutting stock problems. *Computers & Operations Research, 13*(6), 713-719.
172. Rothlauf, F. (2006). *Representations for genetic and evolutionary algorithms*: Springer.
173. Saunders, M., Lweis P., & Thornhill, A. (2009). *Research Methods for Business Students* (5 ed.). Harlow: Pearson Education Limited.
174. Scheithauer, G. (1991). A note on handling residual lengths. *Optimization, 22*(3), 461-466.
175. Scheithauer, G. (2008). *Zuschnitt-und Packungsoptimierung: Problemstellungen, Modellierungstechniken, Lösungsmethoden*: Springer DE.
176. Scheithauer, G., & Terno, J. (1995). The modified integer round-up property of the one-dimensional cutting stock problem. *European Journal of Operational Research, 84*(3), 562-571.
177. Scheithauer, G., & Terno, J. (1997). Theoretical investigations on the modified integer round-up property for the one-dimensional cutting stock problem. *Operations Research Letters, 20*(2), 93-100.
178. Schilling, G., & Georgiadis, M. C. (2002). An algorithm for the determination of optimal cutting patterns. *Computers & Operations Research, 29*(8), 1041-1058.
179. Schrijver, A. (1998). *Theory of linear and integer programming*: Wiley.

180. Schulz, H., & Bimschas, K. (1993). Optimization of Precision Machining by Simulation of the Cutting Process. *CIRP Annals - Manufacturing Technology, 42*(1), 55-58.
181. Sculli, D. (1981). A stochastic cutting stock procedure: cutting rolls of insulating tape. *Management Science, 27*(8), 946-952.
182. Seila, A. F., Ceric, V., & Tadikamalla, P. R. (2003). *Applied simulation modeling*: Thomson, Brooks/Cole.
183. Sen, R., Maurya, A., Raman, B., Mehta, R., Kalyanaraman, R., Vankadhara, N., Roy, S., & Sharma, P. (2012). *Kyun queue: a sensor network system to monitor road traffic queues.* Paper presented at the Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems.
184. Shafer, S. M., & Smunt, T. L. (2004). Empirical simulation studies in operations management: context, trends, and research opportunities. *Journal of Operations Management, 22*(4), 345-354.
185. Shahin, A. A., & Salem, O. M. (2004). Using genetic algorithms in solving the one-dimensional cutting stock problem in the construction industry. *Canadian Journal of Civil Engineering, 31*(2), 321-332.
186. Shannon, R. E. (1998). *Introduction to the art and science of simulation.* Paper presented at the Simulation Conference Proceedings.
187. Sheldon, G. H. (1896). Machine for cutting off I-beams: U.S. Patent No. 552,784. Washington, DC: U.S. Patent and Trademark Office.
188. Singh, G., & Jolly, W. D. (1987). *Method and apparatus for cutting metal plate*: European Patent No. EP 0129952. Munich, Germany: European Patent Office.
189. Sinuany-Stern, Z., & Weiner, I. (1994). The One Dimensional Cutting Stock Problem Using Two Objectives. *The Journal of the Operational Research Society, 45*(2), 231-236.
190. Skiena, S. (1999). Who is interested in algorithms and why?: lessons from the Stony Brook algorithms repository. *ACM SIGACT News, 30*(3), 65-74.
191. Slettengren, A. I. K. (1950). *Machine for cutting*: U.S. Patent No. 2,494,413. Washington, DC: U.S. Patent and Trademark Office.
192. Smith, S. R. (1855). *Machine for cutting wood into slivers*: U.S. Patent No. 12,424. Washington, DC: U.S. Patent and Trademark Office.
193. Soeiro Ferreira, J., Antonio Neves, M., & Fonseca Castro, P. (1990). A two-phase roll cutting problem. *European Journal of Operational Research, 44*(2), 185-196.
194. Stadtler, H. (1990). A one-dimensional cutting stock problem in the aluminium industry and its solution. *European Journal of Operational Research, 44*(2), 209-223.
195. Stazaev, V. N. (1995). *Method of manufacturing wire*: European Patent No. EP 0684092. Munich, Germany: European Patent Office.
196. Suliman, S. M. A. (2001). Pattern generating procedure for the cutting stock problem. *International Journal of Production Economics, 74*(1-3), 293-301.

197. Sweeney, P. E., & Paternoster, E. R. (1992). Cutting and packing problems: a categorized, application-orientated research bibliography. *Journal of the Operational Research Society*, 691-706.
198. Tan, K. C. (2001). A framework of supply chain management literature. *European Journal of Purchasing & Supply Management, 7*(1), 39-48.
199. Tanida, M. (2009). *Tube manufacturing history information management method and device, and manufacturing method using the management method*: European Patent No. EP 2047934. Munich, Germany: European Patent Office.
200. Tansey, M., & Stembridge, B. (2005). The challenge of sustaining the research and innovation process. *World patent information, 27*(3), 212-226.
201. Terzi, S., & Cavalieri, S. (2004). Simulation in the supply chain context: a survey. *Computers in industry, 53*(1), 3-16.
202. Tomat, L., Gradišar, M., & Štiglic, M. (2013). *One-dimensional Cutting Stock Problem: a new method for minimizing trim-loss and preventing a high amount of usable leftovers in stock*. Paper presented at the The 25th International Conference on System Research, Informatics and Cybernetics, Baden-Baden, Germany.
203. Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British Journal of Management, 30*(3), 207-22.
204. Trkman, P. (2008). *Cutting Stock Process Optimization in Consecutive Time Periods.* Ljubljana: Faculty of Economics.
205. Trkman, P. (2010). The critical success factors of business process management. *International Journal of Information Management, 30*(2), 125-134.
206. Trkman, P., & Gradišar, M. (2007). One-dimensional cutting stock optimization in consecutive time periods. *European Journal of Operational Research, 179*(2), 291-301.
207. Trkman, P., & Gradišar, M. (2010). *Optimizacija enodimenzionalnega razreza - od metod za reševanje do procesnega pregleda* (1 ed.). Ljubljana: Ekonomska fakulteta.
208. Trkman, P., & McCormack, K. (2010). Estimating the benefits and risks of implementing e-procurement. *Engineering Management, IEEE Transactions on, 57*(2), 338-349.
209. Tsai, J. F., Hsieh, P. L., & Huang, Y. H. (2009). An optimization algorithm for cutting stock problems in the TFT-LCD industry. *Computers & Industrial Engineering, 57*(3), 913-919.
210. Umetani, S., Yagiura, M., & Ibaraki, T. (2003). One-dimensional cutting stock problem to minimize the number of different patterns. *European Journal of Operational Research, 146*(2), 388-402.
211. Umetani, S., Yagiura, M., & Ibaraki, T. (2006). One-Dimensional Cutting Stock Problem with a Given Number of Setups: A Hybrid Approach of Metaheuristics

and Linear Programming. *Journal of Mathematical Modelling and Algorithms, 5*(1), 43-64.
212. Vahrenkamp, R. (1996). Random search in the one-dimensional cutting stock problem. *European Journal of Operational Research, 95*(1), 191-200.
213. Valério de Carvalho, J. M., & Guimarães Rodrigues, A. J. (1995). An LP-based approach to a two-stage cutting stock problem. *European Journal of Operational Research, 84*(3), 580-589.
214. Van Der Zee, D.-J., & Van Der Vorst, J. G. (2005). A modeling framework for supply chain simulation: Opportunities for improved decision making*. *Decision Sciences, 36*(1), 65-95.
215. Vance, P. H. (1998). Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational Optimization and Applications, 9*(3), 211-228.
216. Vance, P. H., Barnhart, C., Johnson, E. L., & Nemhauser, G. L. (1994). Solving binary cutting stock problems by column generation and branch-and-bound. *Computational optimization and applications, 3*(2), 111-130.
217. Vanderbeck, F. (2000). Exact Algorithm for Minimising the Number of Setups in the One-Dimensional Cutting Stock Problem. *Operations Research, 48*(6), 915-926.
218. Vasko, F. J., Newhart, D. D., & Stott Jr, K. L. (1999). A hierarchical approach for one-dimensional cutting stock problems in the steel industry that maximizes yield and minimizes overgrading. *European Journal of Operational Research, 114*(1), 72-82.
219. Venkateswarlu, P. (2001). The trim-loss problem in a wooden container manufacturing company. *Journal of manufacturing systems, 20*(3), 166-176.
220. Wagner, B. J. (1999). A genetic algorithm solution for one-dimensional bundled stock cutting. *European Journal of Operational Research, 117*(2), 368-381.
221. Wang, H., Qin, X., Ren, C., & Wang, Q. (2012). Prediction of cutting forces in helical milling process. *The International Journal of Advanced Manufacturing Technology, 58*(9-12), 849-859.
222. Wäscher, G. (1990). An LP-based approach to cutting stock problems with multiple objectives. *European Journal of Operational Research, 44*(2), 175-184.
223. Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research, 183*(3), 1109-1130.
224. Westerlund, T., Harjunkoski, I., & Isaksson, J. (1998). Solving a production optimization problem in a paper-converting mill with MILP. *Computers & chemical engineering, 22*(4), 563-570.
225. Wieland, A., & Wallenburg, C. M. (2011). *Supply-Chain-Management in stürmischen Zeiten*: Universitätsverlag der TU.
226. Winston, W. L., & Goldberg, J. B. (2004). *Operations research: applications and algorithms*: Thomson/Brooks/Cole Belmont.

227. Wooster, B. O. (1914). *Machine for cutting cord-wood*: U.S. Patent No. 1,123,056. Washington, DC: U.S. Patent and Trademark Office.
228. Yang, B., Li, C., Huang, L., Tan, Y., & Zhou, C. (2009). *Solving one-dimensional cutting-stock problem based on ant colony optimization.* Paper presented at the INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on.
229. Yang, C. T., Sung T. C., & Weng, W. C. (2006). An improved tabu search approach with mixed objective function for one-dimensional cutting stock problems. *Advances in Engineering Software, 37*(8), 502-513.
230. Yen, C. H., Wong, D. S. H., & Jang, S. S. (2004). Solution of trim-loss problem by an integrated simulated annealing and ordinal optimization approach. *Journal of Intelligent Manufacturing, 15*(5), 701-709.
231. Yeung, L. H. W., & Tang, W. K. S. (2003). A hybrid genetic approach for garment cutting in the clothing industry. *Industrial Electronics, IEEE Transactions on, 50*(3), 449-455.
232. Yildiz, A., & Ozturk, F. (2006). Hybrid enhanced genetic algorithm to select optimal machining parameters in turning operation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 220*(12), 2041-2053.
233. Yin, R. K. (2011). *Qualitative Research from Start to Finish*. New York: The Guilford Press.
234. Ylitalo, J. (2011). *Beam forming method, apparatus and system*: European Patent No. EP 2342834. Munich, Germany: European Patent Office.
235. Yuen, B. J., & Richardson, K. V. (1995). Establishing the optimality of sequencing heuristics for cutting stock problems. *European Journal of Operational Research, 84*(3), 590-598.
236. Yun, W.-S., Ko, J. H., Lee, H. U., Cho, D.-W., & Ehmann, K. F. (2002). Development of a virtual machining system, part 3: cutting process simulation in transient cuts. *International Journal of Machine Tools and Manufacture, 42*(15), 1617-1626.
237. Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2012). The Pallet-Packing Vehicle Routing Problem. *Transportation Science, 46*(3), 341-358.
238. Zak, E. J. (2002a). Modeling multistage cutting stock problems. *European Journal of Operational Research, 141*(2), 313-327.
239. Zak, E. J. (2002b). Row and column generation technique for a multistage cutting stock problem. *Computers & Operations Research, 29*(9), 1143-1156.
240. Zhang, J., Sun, T., Yan, Y., Liang, Y., & Dong, S. (2008). Molecular dynamics simulation of subsurface deformed layers in AFM-based nanometric cutting process. *Applied Surface Science, 254*(15), 4774-4779.
241. Zhigang, H., Yinglin, K., & Litao, W. (2003). Study on Related Techniques for the Finite Element Method Simulation in Metal Cutting. *China Mechanical Engineering, 14*(10), 846-453.

242. Zhu, P.-z., Hu, Y.-z., Ma, T.-b., & Wang, H. (2010). Study of AFM-based nanometric cutting process using molecular dynamics. *Applied Surface Science, 256*(23), 7160-7165.

APPENDICES

LIST OF APPENDICES

Appendix 1: Abbreviations

1DCSP	One-dimensional cutting stock problem
1DCSPUL	One-dimensional cutting stock problem with usable leftover
GA	Genetic Algorithm
EP	Evolutionary Programming
EA	Evolutionary Algorithm
SHP	Sequential Heuristic Procedure
SCM	Supply Chain Management
PC	Personal Computer
FFD	First Fit Decreasing
MBS	Minimal Bin Slack
RGR	Residual heuristic by Greed Rounding

DALJŠI POVZETEK DISERTACIJE V SLOVENSKEM JEZIKU

Področje operacijskih raziskav je ključno za proučevani problem. Čeprav je v literaturi mogoče zaslediti več definicij več različnih avtorjev, lahko za potrebe doktorske disertacije definiram operacijske raziskave kot uporabo analitičnih znanstvenih metod, ki zagotavljajo odločevalcem kvantitativno osnovo za boljše odločanje v organizacijah (Čižman, 2004).

Namen operacijskih raziskav je poiskati optimalno (eksaktno) ali skoraj optimalno (hevristično) rešitev za kompleksne odločitvene probleme. Operacijske raziskave temeljijo na mnogih matematičnih metodah in tehnikah, kot so npr. modeliranje, odločitvena analiza, poslovne simulacije itd. Del operacijskih raziskav predstavlja tudi optimizacija, problem eno-dimenzionalnega razreza pa sodi med optimizacijske probleme.

V operacijskih raziskavah imajo simulacije pomembno vlogo (Shafer & Smunt, 2004). V doktorski disertaciji je predstavljen nov, natančen model testiranja, temelječ na simulaciji, ki v literaturi za rešitve, ki obravnavajo posamezna naročila, še ni bil uporabljen. Metoda upošteva uporabne ostanke iz prejšnje instance v naslednji instanci. Tako se metoda bolj približa realnim situacijam, kot metode testiranja v literaturi, ki vračanja ne upoštevajo.

V teoretičnem delu doktorske disertacije podrobno raziščem področji reševanja problema eno-dimenzionalnega razreza materiala (1DCSP) in problema eno-dimenzionalnega razreza materiala z uporabnim ostankom (1DCSPUL). V praktičnem delu razvijem in opišem novo optimizacijsko metodo za reševanje 1DCSPUL za primere, kjer je razmerje med povprečno dolžino palic na zalogi in povprečno dolžino naročil (r) nizko. Nato predlagam nov model za testiranje metod za reševanje 1DCSPUL, ki temelji na simulacijah. Na koncu opišem glavne ugotovitve, prikažem prispevek doktorske disertacije k znanosti in podam predloge za nadaljnje delo.

PROBLEMATIKA IN PREDMET RAZISKOVANJA

Z 1DCSP se srečuje mnogo proizvodnih podjetij, predvsem pa tista, ki želijo optimizirati svoje poslovanje (Haessler & Sweeney, 1991). Tako optimizacija razreza materiala predstavlja pomembno raziskovalno področje širom sveta. Jedro problema je, kako izpolniti naročilo z minimalnimi stroški (Haessler & Sweeney, 1991).

1DCSP ima v praksi veliko pojavnih oblik. V splošnem je definiran kot problem, kjer je treba v naprej znano število naročenih palic razrezati iz znane zaloge, ki je sestavljena iz različnega števila različnih dolžin palic. Pri rezanju materiala je zelo pomembno, kako velik je ostanek rezanja, saj se nezadostno velik ostanek smatra kot odpadek in se zavrže, to pa vpliva na višje stroške podjetja. Smotrno je, da je ostanek, ki se zavrže, čim manjši oz. tako majhen, kot je to mogoče (Gradišar & Jesenko, 1996).

Prvi poskusi iskanja optimalne rešitve so se začeli leta 1939, ko je Leonid Kantorovich predstavil osnovni model za reševanje problema eno-dimenzionalnega razreza, vendar pa takšna formulacija problema, kot jo poznamo danes, izhaja iz leta 1956 (Paull, 1956). Naslednji pomembni korak se je zgodil leta 1961, ko sta Gilmore in Gomory, dva izmed najbolj priznanih avtorjev s področja 1DCSP, predstavila metodo reševanja, ki temelji na linearnem programiranju (Gilmore & Gomory, 1961).

Kasneje je bilo predstavljenih še veliko metod za reševanje 1DCSP, kar je posledično vodilo v potrebo po klasifikaciji, ki bi jasno razlikovala med različnimi vrstami problemov. V ta namen je leta 1990 Dyckhof predlagal klasifikacijo problemov eno-dimenzionalnega razreza materiala. Po tej klasifikaciji je 1DCSP opisan kot model 1/V/D/M, kjer 1 predstavlja eno-dimenzionalni problem, V predstavlja pogoj, da vsa naročila izhajajo iz razreza zalog, D predstavlja več različnih palic na zalogi, M pa predstavlja več različnih naročil. Klasifikacija se lahko uporablja tudi za nekatere druge, podobne primere (Dyckhoff, 1990):

- problemi razreza in problemi odpadka,
- problemi pakiranja,
- problemi nakladanja,
- problemi alokacije, razvrščanja in kategorizacije,
- alokacija kapitala,
- določanje urnikov.

Na osnovi Dyckhoffove klasifikacije je bilo razvitih več zanimivih metod za reševanje problema razreza (Scheithauer & Terno, 1997; Vanderbeck, 2000; Belov & Scheithauer, 2002). V letih med 1995 in 2002 se je pojavilo tudi več večfaznih metod, ki so opisane v Trkman in Gradišar (2010). Klasifikacija je bila kasneje razširjena tudi s časovno dimenzijo. Takšen pristop je bil prvič predstavljen v Trkman in Gradišar (2007). Nova klasifikacija temelji na več kriterijih: število različnih dolžin na zalogi, razmerje med povprečno dolžino materiala na zalogi, povprečno dolžino naročenega materiala in karakteristikah posameznih kosov materiala na zalogi in v naročilu (Trkman & Gradišar, 2007). Leta 2007 je bila predstavljena tudi nova klasifikacija, ki nakazuje pomemben napredek na področju rezanja in pakiranja (Wäscher et al. 2007). Z njeno uporabo je mogoče še bolj natančno razvrstiti probleme v razrede.

V procesu optimizacije eno-dimenzionalnega razreza materiala se zaloge porabljajo z namenom izpolnjevanja naročila, pri tem pa se sledi ekonomskemu cilju minimiziranja stroškov. Eden glavnih razlogov je v minimizaciji materiala, ki predstavlja neuporaben ostanek, saj le-ta pomeni izgubo oz. strošek za podjetje. Kadar je ostanek dovolj velik, da se ga vrne v skladišče in ponovno uporabi, govorimo o 1DCSPUL (Alfieri, 2007; Areales et al., 2009).

Optimalne rešitve so večinoma rezultat uporabe eksaktnih metod (Alves & Valério de Carvalho, 2008), vendar so te metode uporabne le pri majhnih naročilih (rezultati eksaktnih metod predstavljajo najboljše možne rešitve, v primeru večjih naročil pa je zaradi pomanjkanja računske moči računalnikov čas računanja predolg), tako da izključno eksaktne metode za uporabo v moji disertaciji niso primerne. Zato večina razvitih metod temelji na hevrističnem pristopu. Nasprotno od eksaktnih metod hevristične rešitve ne zagotavljajo optimalne rešitve, jo pa zato poiščejo v razumnem času. Nekatere izmed hevrističnih metod za reševanje 1DCSP in 1DCSPUL so predstavljene v doktorski disertaciji.

Pri reševanju 1DCSPUL je bistvenega pomena uporabni ostanek, ki se vrne v skladišče z namenom kasnejše uporabe. Pomembno je, kje je meja, nad katero se ostanek vrne nazaj v skladišče, in pod katero ostanek predstavlja izgubo pri razrezu. To mejo navadno določijo odločevalci. Tako v primeru vračanja neuporabljenega materiala v skladišče minimizacija ostanka ni nujno najboljša rešitev za podjetje (Cherri et al., 2009). Pri reševanju 1DCSPUL so na zalogi palice standardnih in nestandardnih dolžin. Slednje so uporabni ostanki iz prejšnjih naročil. Upoštevati je trebna naslednje predpostavke (Cherri et al., 2009):

- velikost naročila mora biti znana,
- velikost zaloge mora biti znana,
- naročila morajo biti izpolnjena.

V literaturi je mogoče najti mnogo metod za reševanje 1DCSPUL. Obširen pregled je npr. objavljen v Chierri et. al (2009). Metode večinoma temeljijo na hevrističnem pristopu. Pomembno mesto med metodami za reševanje 1DCSPUL imata metodi COLA in CUT, ki sta bili za ta tip problema razviti najprej. COLA je bila predstavljena leta 1997 (Gradišar et al., 1997), CUT pa leta 1999 (Gradišar et al., 1999a). Obe metodi posamično obravnavata naročila in temeljita na izčrpnem ponavljanju.

Iz praktičnega primera, predstavljenega v Erjavec et al. (2009), je razvidno, da je povprečna izguba materiala pri reševanju 1DCSPUL enaka 15 %. Vzrok pa ne tiči le v izbrani optimizacijski metodi, temveč tudi v naravi problema. Razlog je v nizkem *r*. Ker literatura ne ponuja nobene definicije nizkega *r*, omenjeno razmerje v doktorski disertaciji definiram na naslednji način:

»Razmerje med povprečno dolžino na zalogi in povprečno dolžino naročil (r) je majhno, če je manjše od nekega praga.«

Ker je v vsakem primeru problema več različnih dolžin zaloge in naročila, je lahko omenjeno razmerje:

1. razmerje med najdaljšo dolžino na zalogi in najkrajšo dolžino naročila, ali
2. razmerje med povprečno dolžino zaloge in povprečno dolžino naročila.

Prvo razmerje ne predstavlja najboljše definicije v vseh primerih (npr. kjer obstaja veliko število naročil, najdaljše naročilo pa predstavlja le zelo majhen del skupne dolžine naročil), zato je druga možnost boljša izbira. Naslednje vprašanje je povezano z definiranjem praga nizkega razmerja. V Erjavec et al. (2009) je ta enak 3, zato je takšen prag upoštevan v doktorski disertaciji.

Nižji *r* pomeni omejeno število možnih rešitev, posledično pa se poveča verjetnost rešitve z veliko izgubo (Gradišar et al., 1999a). Število možnih rešitev se najbolj zniža takrat, ko je *r* med 1 in 2, saj to pomeni, da se v večini primerov od ene palice v zalogi odreže samo ena enota naročila. To je tudi razlog, da primeri, kjer je *r* nizek, povzročajo več ostanka. Iz zgoraj napisanega je mogoče domnevati, da se razlika med izgubo optimalne rešitve in izgubo, ki je posledica uporabe trenutno razpoložljivih metod, z izjemo eksaktnih, veča, če se niža *r*. To posledično predstavlja tudi večje možnosti za izboljšave.

Stroški, povezani z razrezom, pa niso odvisni le od priprave čim boljšega načrta razreza, ampak tudi od nabave materiala in prilagoditve logističnih procesov. To pomeni, da lahko celostna prenova procesa eno-dimenzionalnega razreza materiala vodi do bistveno večjih prihrankov kot le optimizacija načrta razreza. Metode za reševanje eno-dimenzionalnega problema razreza morajo imeti torej možnost prilagoditve spreminjajočim poslovnim procesom in enostavne integracije v informacijski sistem podjetja (Alfieri, van de Velde, & Woeginger, 2007; Čižman & Černetič, 2004; Rodrigues & Vecchietti, 2007).

Vse hitrejši razvoj in večja dostopnost računalnikov v zadnjih desetletjih sta odločevalcem omogočila večje možnosti pri sprejemanju odločitev. Ker je z računalniškimi simulacijami mogoče proučevati realni sistem tudi v daljšem časovnem obdobju, predstavljajo eno izmed najbolj pogosto uporabljenih in najzmogljivejših orodij pri načrtovanju zahtevnih procesov in sistemov. Pri simulacijah gre za proces izdelave modela realnega sistema in izvajanja eksperimentov s tem modelom, z namenom boljšega razumevanja interakcij znotraj kompleksnih procesov (Pidd, 2004).

Računalniške simulacije temeljijo na računalniškem modelu določenega sistema. Običajno se simulira različne scenarije, tisti z najboljšim rezultatom pa je potem uporabljen v realnem sistemu. Takšen eksperiment je npr. iskanje odgovorov na t. i. »kaj če« vprašanja. Uporaba računalniških simulacij je pogosta v mnogih panogah, npr. v proizvodnji, obrambni industriji, zdravstvu in javnem sektorju (Jahangirial et al., 2010). Računalniške simulacije so učinkovito orodje za podporo odločanju, zato se pogosto uporabljajo za načrtovanje, kontrolo, pripravo strategij in razporeditev virov v podjetjih (Folwer, 1998). Ker omogočajo interaktivno modeliranje, zagotavljajo transparentnost in razumljivost proučevanih modelov (Kljajić et al., 2000).

Na področju operacijskih raziskav je v zadnjih 35. letih mogoče opaziti porast v številu znanstvenih študij, ki temeljijo na uporabi simulacij, saj število znanstvenih člankov s tega področja narašča (Shafer & Smunt, 2004). Tudi pri optimizaciji razreza materiala, ki predstavlja del proizvodnega procesa, se lahko simulacije uporabljajo za različne analize. Omogočajo npr. ocenjevanje investicij v opremo, stroje, skladišča in distribucijske centre (Erjavec, 2011). Številni primeri uporabe simulacij v povezavi s procesom razreza materiala so opisani v 3. poglavju doktorske disertacije.

CILJI DOKTORSKE DISERTACIJE

Cilja v moji doktorski disertaciji sta naslednja:

- Razvoj nove metode za reševanje 1DCSPUL z majhnim *r*, ki bo dala boljše rezultate od obstoječih metod.
- Izboljšanje obstoječih metod testiranja rešitev 1DCSPUL z metodo simulacij.

OPIS ZNANSTVENE METODE

Prva uporabljena znanstvena metoda zajema natančen in obsežen pregled znanstvene literature (tako primarni kot sekundarni viri) s poudarkom na znanstvenih prispevkih, ki se nanašajo na obravnavana področja. Analizirani so tudi kriteriji učinkovitosti eno-dimenzionalnega razreza.

Pri testiranju nove optimizacijske metode za reševanje 1DCSPUL z majhnim *r* so testni podatki generirani s pomočjo generatorja primerov, deloma pa so iz prakse. Za pomoč pri raziskavi je uporabljena ustrezna programska oprema (npr. MS Excel, PGEN, COLA, CUT, MPL/CPLEX). Rezultati so ocenjeni z metodo primerjalne analize. Primerjal sem rezultate nove metode, razvite za reševanje 1DCSPUL z majhnim *r*, z rezultati obstoječih metod.

Rezultati predlagane metode testiranja rešitev 1DCSPUL so pridobljeni z uporabo simulacij, ki so ene izmed najbolj pogosto uporabljenih raziskovalnih tehnik v družboslovnih znanostih (Pidd, 2004).

RAZVOJ NOVE METODE

Pričujoči razdelek je povzet po Gradišar, Tomat in Erjavec (2011). Kjer je potrebno, so dodane in opisane ustrezne prilagoditve.

Definicija problema

Pri definiciji problema so upoštevane naslednje predpostavke:

- na zalogi je na voljo zadostna količina materiala za izpolnitev vseh naročil kupca,
- na zalogi je na voljo določeno število dolžin,
- standardna zaloga sestoji iz enake dolžine ali iz več standardnih dolžin,
- nestandardna zaloga sestoji iz uporabnih ostankov iz prejšnjih naročil,

- ostanki, ki so enaki ali daljši od minimalne dolžine palic na naročilu, se štejejo kot uporabni in so vrnjeni na zalogo z namenom uporabe v prihodnjih naročilih,
- dolžine so izražene s celimi števili – če niso, se predpostavlja, da jih je vedno mogoče pomnožiti s takšnim faktorjem, ki jih zaokroži na celo število,
- število naročenih dolžin tvori naročilo, ki mora biti dosledno izpolnjeno.

Pomembno je, kako je definiran prag nizkega *r*, saj je od njegove velikost lahko odvisna višina izgube materiala pri reševanju 1DCSPUL. V praktičnem primeru, predstavljenem v Erjavec et al. (2009), je *r* enak 3, zato je takšen prag upoštevan v pričujoči doktorski disertaciji.

Natančna matematična formulacija problema je podana v 4. poglavju doktorske disertacije.

Razvoj rešitve

Obravnavani problem bi bilo mogoče rešiti z eno izmed obstoječih metod za reševanje 1DCSPUL, vendar pa lahko metoda, razvita posebej za takšen tip problema, proizvede boljše rezultate. Posebna pozornost pri razvoju metode je namenjena prilagodljivosti metode. Razvita je na način, ki omogoča enostavno integracijo v informacijske sisteme podjetij.

V literaturi je na voljo več metod za reševanje 1DCSPUL, ki večinoma temeljijo na hevrističnem pristopu. Uporaba eksaktnih metod privede do boljših rezultatov, vendar pa je običajno čas čakanja na rešitev daljši, kar v podjetjih navadno ni sprejemljivo. Uporaba eksaktnih metod je zato smiselna le za reševanje relativno majhnih problemov. Ker literatura ne ponuja posebnih informacij o primerih z nizkim *r*, sem z algoritmom C-CUT (Gradišar & Trkman, 2005) izvedel eksperiment, s katerim sem ugotavljal, ali je problem mogoče rešiti eksaktno. Postopoma sem zviševal število kosov naročil pri običajnem *r* (več kot 5) in pri majhnem *r* (manj kot 2) ter opazoval, ali je bila najdena optimalna rešitev v zglednem času. Ko je bilo število kosov naročil pri običajnem *r* enako ali večje kot 21, algoritem ni več našel optimalne rešitve. Pri nizkem *r* se je to zgodilo, ko je bilo število kosov naročil enako ali večje od 37. Iz eksperimenta je mogoče sklepati, da bi bilo razmerje podobno tudi pri sodobnih algoritmih, ki skladno z literaturo pri običajnem *r* trenutno omogočajo 100 kosov naročil. Tako bi bilo pri manjših *r* število kosov naročil nekaj manj kot 200, kar je precej manj, kot zmore obdelati predlagani algoritem, za katerega ocenjujem, da omogoča obdelavo do 700 kosov naročil.

Zato v doktorski disertaciji razvijem hevristično metodo za reševanje 1DCSPUL pri nizkem *r*, ki temelji na kombinaciji metode vzorcev in posamičnega obravnavanja. Predlagana metoda sestoji iz štirih ponavljajočih se korakov:

1. korak
Algoritem reši problem nahrbtnika[12] in poišče optimalno in drugo najbolj optimalno rešitev za vsako dolžino palic na zalogi, pri čemer upošteva le nepredelane kose naročil.

2. Korak
V 1. koraku pridobljeni vzorci razreza se razvrstijo po v naprej določenem zaporedju. Rezultat je razvrščeni seznam vzorcev. Na vrhu seznama so vzorci, ki vsebujejo daljša naročila.

3. Korak
Za vsak vzorec s seznama se določi ustrezna frekvenca[13]. Določanje frekvence poteka zaporedno, pri čemer se najprej obravnavajo vzorci, ki so na vrhu seznama.

4. korak
Če so vsa naročila izpolnjena, se algoritem ustavi, sicer se ponovno vrne na korak 1.

Podrobnejša obrazložitev in grafični prikaz predlaganega algoritma je v 3. poglavju doktorske disertacije.

Testiranje metode in analiza rezultatov

Predlagana metoda se imenuje LCUT in je napisana v programskem jeziku FORTRAN, omogoča zelo hitro procesiranje in ni namenjena splošni uporabi, temveč za primere z nizkim *r*. Vhodni podatki so generirani s pomočjo četrte generacije programskih jezikov. Program lahko teče na navadnem osebnem računalniku. LCUT ima naslednje omejitve:

- razmerje med največjo dolžino na zalogi in najkrajšo dolžino naročila mora biti manjše ali enako 10,
- število različnih dolžin naročila mora biti manjše ali enako 7,
- število kosov za posamezno dolžino naročila mora biti manjše ali enako 99,
- število različnih dolžin na zalogi mora biti manjše ali enako 20,
- število kosov za posamezno dolžino na zalogi mora biti manjše ali enako 99.

Delovanje algoritma LCUT je prikazano na realnih podatkih, pridobljenih v slovenskem podjetju, ki je eno izmed večjih ponudnikov tehničnih izdelkov v JV Evropi. Rezultati so bili pridobljeni v manj kot 1 sekundi, skupna izguba pa je znašala 11,98 % uporabljenega materiala. Učinkovitost delovanja predlagane metode je na enakem primeru prikazana s primerjavo z metodama RGR (Cherri et al., 2009) in CUT (Gradišar et al., 1999a). RGR je proizvedel 12,17 % izgube materiala, kar je 0,19 odstotne točke več kot LCUT. Tudi CUT

[12] Problem nahrbtnika je podrobno predstavljen v 1. poglavju doktorske disertacije.
[13] Ustrezna frekvenca je najvišja frekvenca, ki v danem trenutku ni višja, kot je število nepredelanih palic na zalogi. Hkrati mora biti ustrezna frekvenca dovolj nizka, da prepreči preveliko proizvodnjo katerekoli naročene dolžine.

se je odrezal slabše, saj je proizvedel 12,75 % izgube materiala, kar je 0,77 odstotne točke več kot LCUT.

Za prikaz delovanja predlagane metode sem izvedel eksperiment, ki je vseboval širok nabor problemov. Za generiranje skupno 240 problemov je bil uporabljen rahlo prilagojen generator naključnih števil PGEN (Gradišar et. al., 2002). Vsak izmed problemov je bil rešen z algoritmoma LCUT in CUT. Eksperiment je bil izveden za dva scenarija z vidika stroškov vračanja uporabnih ostankov na zalogo:

- v prvem scenariju so ti stroški zanemarljivi v primerjavi s stroški prihranjenega materiala. Zato so vsi ostanki, ki so enaki ali daljši od minimalne dolžine naročila vrnjeni na zalogo z namenom prihodnje uporabe;
- v drugem scenariju so stroški vračanja ostankov na zalogo zelo visoki. Z vidika stroškov jih je tako bolje zavreči kot vrniti na zalogo.

Opazoval sem tudi, do kakšnih rezultatov privede kombinacija obeh metod. Vsak izmed problemov je rešen z obema metodama, potema pa je izbrana boljša izmed rešitev. Tako se je skupna izguba materiala zmanjšala za 0,1 %. Poleg nižje izgube materiala, pa se z uporabo kombinacije zmanjša tudi število proizvedenih uporabnih ostankov.

Da bi prikazal učinkovitost predlagane metode, je bil opravljen še en eksperiment, ki je zajemal oba zgoraj opisana scenarija. Generiral sem 48 primerov s podobnim *r* kot pri prejšnjem eksperimentu, vendar z drugačnimi vhodnimi podatki. Ti primeri so bili rešeni z eksaktno metodo, objavljeno v Gradišar in Trkman (2005), algoritmom CUT in kombinacijo algoritmov LCUT in CUT. Velikost problemov je bila dovolj majhna, da je omogočala uporabo eksaktne metode. Rezultati so pokazali, da bi bila za proučevane primere najbolj smotrna uporaba eksaktne metode, vendar pa so v praksi problemi bistveno večji, zato uporaba eksaktne metode ne bi bila mogoča. Metoda, predlagana v doktorski disertaciji, lahko rešuje probleme, ki vsebujejo do 693 naročil in do 20 različnih dolžin z do 1980 palicami na zalogi, kar je neprimerno več, kot trenutno zmorejo reševati eksaktne metode (Belov & Scheithauer, 2002; Alves & de Carvalho, 2008).

TESTIRANJE METODE

Učinkovitost metode za reševanje 1DCSPUL je lahko prikazana, če je primerjana z ostalimi metodami. V literaturi sicer že obstajajo primerjave, ki temeljijo na več kriterijih (Cherri et al., 2009; Cui & Yang, 2010), vendar pa ne zagotavljajo točnih in popolnih informacij o učinkovitosti različnih metod.

Metode za reševanje 1DCSPUL bi bile lahko testirane z uporabo enakih kriterijev in enakih primerov, vendar pa rezultati najverjetneje ne bi bili uporabni, saj je ključni element spregledan. Pri ocenjevanju metod za reševanje 1DCSPUL je namesto opazovanja posameznega primera treba upoštevati skupino več primerov. Ključni kriterij je tako naraščanje uporabnih ostankov na zalogi skozi čas in vplivanje na izpolnjevanje prihodnjih naročil. Nova metoda za testiranje, predlagana v doktorski disertaciji, temelji na konceptu, kjer so uporabni ostanki, ki so ustvarjeni pri izpolnjevanju prejšnjega naročila, uporabljeni za izpolnjevanje trenutnega, uporabni ostanki, ustvarjeni pri izpolnjevanju trenutnega naročila, pa so uporabljeni pri naslednjem. Takšen pristop je v literaturi mogoče zaslediti le v Trkman in Gradišar (2007) in Cherri et al. (2012), ostale metode pa tega ne upoštevajo. V literaturi tudi ni moč zaslediti nobene metode za testiranje, ki bi omogočala natančno primerjavo različnih rešitev. Metode v literaturi so tako napačno testirane. Zato je v doktorski disertaciji predlagana nova metoda za testiranje rešitev 1DCSPUL.

Uporabni ostanki na zalogi niso ustvarjeni naključno, kot npr. v Cherri et al. (2009) in Cui in Yang (2010), temveč je pri vsakem naročilu uporabljena metoda računalniških simulacij. Tako je z uporabo simulacij naraščanje uporabnih ostankov v skladišču takšno kot v realnosti.

Da bi prikazal delovanje predlagane metode, sem analiziral 4 različne primere z različnim, vendar nizkim *r*. Problemi so generirani z generatorjem problemov CUTGEN1 (Gau & Wäscher, 1995). V dveh primerih so vhodni parametri in podatki o zalogi enaki kot v Cherri et al. (2009) in Cut in Yang (2010). Poleg velikosti proizvedenega neuporabnega ostanka materiala sem opazoval tudi število proizvedenih uporabnih ostankov. Za potrebe prikaza predlagane metode je bil za reševanje 1DCSPUL izbran algoritem COLA, saj v primerjavi z ostalimi metodami porablja večje število ostankov iz prejšnjih naročil. Na ta način se možnost za naraščanje uporabnih ostankov na zalogi poveča, kar bolje odraža realno situacijo. Preveril sem tudi odvisnost rezultatov od obratnega zaporedja z istim naborom naročil. Rezultati kažejo, da je razlika med izgubo materiala in količino uporabnih ostankov na zalogi pri obeh eksperimentih tako majhna, da se takšna odvisnost lahko zanemari. Z namenom demonstracije predlagane testne metode sem izvedel še dodatni eksperiment. Za primerjavo metod COLA in ECOLA sem analiziral dva primera, kjer sem simuliral 100 zaporednih časovnih obdobij. Takšen pristop odraža dejansko situacijo bolje kot metode v literaturi, saj so namesto naključno ustvarjenih uporabnih ostankov uporabljeni dejanski uporabni ostanki iz prejšnjih časovnih obdobij.

Predlagana metoda testiranja rešitev za 1DCSPUL je boljša od obstoječih metod, saj je bolj točna in zagotavlja popolnejše informacije o številu uporabnih ostankov na zalogi po več zaporednih obdobjih.

ZAKLJUČEK

V predstavljeni doktorski disertaciji proučujem področje 1DCSPUL, ki je dobro znani problem v mnogih industrijskih panogah. Ključnega pomena je zmanjšanje izgube materiala. Velikost neuporabnega ostanka je navadno odvisna od *r*. V primeru nizkega *r* se število možnih rešitev zmanjša, možnost rešitve z večjim neuporabnim ostankom pa se poveča (Gradišar et al., 1999a). Ker to predstavlja možnosti za izboljšave, je v doktorski disertaciji razvita nova optimizacijska metoda za reševanje 1DCSPUL pri nizkem *r*. Eksperimentalni rezultati so pokazali, da se predlagana metoda obnese bolje od obstoječih metod. Metoda je predstavljena na način, ki omogoča enostavno implementacijo v obstoječe informacijske sisteme podjetij.

Preverjanje učinkovitosti metode za reševanje 1DCSPUL se lahko prikaže s primerjavo z obstoječimi metodami, kjer se upošteva enake kriterije in probleme. Vendar pa na tak način rezultati niso realni. Namesto obravnavanja posameznega naročila je treba upoštevati skupino več zaporednih naročil. Zato v doktorski disertaciji predlagam novo metodo testiranja, ki temelji na računalniških simulacijah, kjer so uporabni ostanki iz prejšnjih naročil uporabljeni za izpolnjevanje prihodnjih.

Aktivnosti, povezane z razrezom postajajo vse bolj pomemben del upravljanja z oskrbovalno verigo, predvsem v kovinski, jeklarski, lesni in transportni industriji. Za čim večjo učinkovitost procesa razreza je pomembno, da podjetja implementirajo takšne metode za reševanje 1DCSPUL, ki proizvedejo čim manj neuporabnega ostanka in na ta način zmanjšajo stroške.

Podjetja se morajo pri izbiri metod odločiti za takšno, ki je najbolj primerna za reševanje njihovih problemov. Izbor je v splošnem odvisen od testiranj posameznih metod. Predstavljena nova metoda testiranja tako pomaga podjetjem pri izboru najprimernejše metode za zadovoljitev njihovih poslovnih potreb.

Teoretičen prispevek k znanosti je predstavljen z razvojem nove metode za reševanje 1DCSPUL pri nizkem *r* in nov natančen model testiranja, temelječ na simulaciji, ki v literaturi za rešitve, ki obravnavajo posamezna naročila, še ni bil uporabljen. Predstavljena doktorska disertacija tako prispeva znaten delež k nadaljnjemu razvoju obravnavanih področij.

Predlagana metoda za reševanje 1DCSPUL pri nizkem *r* je predvidena za reševanje eno-dimenzionalnih problemov, vendar bi bilo mogoče podoben pristop uporabiti tudi za reševanje dvo-dimenzionalnih ali tri-dimenzionalnih problemov. Nadaljnje raziskave bi lahko zajemale tudi vključitev različnih stroškov v metodo, kot so npr. stroški skladiščenja.

Eksperimenti kažejo, da višina *r* vpliva na količino uporabnih ostankov na zalogi. Zato bi bilo smiselno narediti eksperimente z različnimi metodami in več časovnimi obdobji ter opazovati, kako višina *r* vpliva na naraščanje števila uporabnih ostankov v skladišču.