

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

PETER TRKMAN

OPTIMIZACIJA PROCESA ENODIMENZIONALNEGA RAZREZA
V ZAPOREDNIH ČASOVNIH OBDOBJIH

DOKTORSKA DISERTACIJA

LJUBLJANA, 2008

OPTIMIZACIJA PROCESA ENODIMENZIONALNEGA RAZREZA V ZAPOREDNIH ČASOVNIH OBDOBJIH

Povzetek

Enodimenzionalni razrez materiala se pogosto pojavlja v praksi. Običajno je potrebno iz materiala, ki je trenutno na zalogi (na zalogi imamo torej različno število palic različnih dolžin), narezati zahtevano število palic vnaprej določenih dolžin za potrebe proizvodnje ali nadaljnje prodaje. Najpogostejši cilj priprave načrta razreza je zmanjšanje izgube materiala, včasih pa želimo doseči še zmanjšanje stroškov ali časa razreza, števila menjav nastavitev rezila, zmanjšanje zaloge in podobne cilje.

Večina različic problema razreza je NP težkih, tako da algoritem polinomske časovne zahtevnosti za pridobitev optimalne rešitve najverjetneje ne obstaja. V praksi so zato za večje probleme uveljavljene različne hevristične metode.

V doktoratu se ukvarjam s splošnim problemom eno-dimenzionalnega razreza, kjer so lahko vse dolžine palic različnih dolžin. Tak problem je posplošitev standardnega problema, kjer je na razpolago le ena ali nekaj standardnih dolžin palic – za take probleme se običajno uporablja metoda vzorcev na podlagi Gilmore-Gomoryjeve metode. Do različnih dolžin palic v skladišču lahko pride zaradi različnih razlogov: zaradi vračanja delno narezanih palic v skladišče, namernega naročanja različnih dolžin zaradi večjega števila kombinacij pri razrezu ali naravnih danosti.

Poleg zmanjševanja izgube v danem časovnem trenutku, pokažem tudi na pristope k razreševanju pomembne pomanjkljivosti večine obstoječih metod za različne vrste razreza. Le-te namreč obravnavajo samo razrez v nekem danem trenutku – iščejo torej dober ali optimalen načrt razreza za dani material in dane potrebe. To je le približna slika realnosti v večini podjetij, ki se z razrezom ukvarjajo kontinuirano in potrebujejo načrte razreza za več zaporednih, med seboj povezanih obdobjih. Želijo izpolnjevanje ciljev (denimo majhne izgube) za skupek vseh obdobjih in ne samo za posamezno obdobje. Lokalni optimumi ne vodijo nujno do dobrega načrta razreza za celotno obdobje. Zato v disertaciji posebno pozornost namenim preučevanju ustreznosti metod za trenutno optimizacijo v daljšem časovnem obdobju.

Učinkovit razrez pa lahko h konkurenčnosti podjetja največ prispeva, če je ustrezno vključen v druge procese v podjetju, predvsem v proces preskrbe z materialom, s katerim je tesno povezan. Zagotavljanje potrebnega materiala za razrez je lahko učinkovitejše in cenejše ob uspešnem upravljanju odnosov z dobavitelji v oskrbovalni verigi. Uspešno poslovanje z dobavitelji oziroma kupci pa je lažje ob večji procesni usmerjenosti (McCormack, Johnson, 2000; Lockamy, McCormack, 2004) in posledični integriranosti procesov. Omenjeno področje pridobiva na pomenu, vendar razrez do sedaj še ni bil obravnavan kot eden od procesov v oskrbovalni verigi.

V disertaciji zato najprej predstavim vlogo razreza kot enega od (pod)procesov v podjetju oziroma celi oskrbovalni verigi. Obravnavam vlogo modeliranja poslovnih procesov pri integraciji procesa razreza v oskrbovalni verigi. Na koncu identificiram ključne probleme in neodgovorjena raziskovalna vprašanja pri problemu razreza.

V tretjem poglavju najprej podajam obširen pregled literature s področja razreza. Pri tem najprej navedem osnovne probleme pri iskanju dobrih rešitev ter nekatere pristope za odpravo teh problemov. Predstavim splošno klasifikacijo za probleme razreza. Nato se osredotočim predvsem na obširen prikaz predhodno razvitih metod za reševanje problema enodimenzionalnega razreza.

V četrtem poglavju podam besedno in matematično definicijo splošnega enodimenzionalnega problema razreza ob upoštevanju vračanja in ponovne uporabe materiala. Z različicami tega problema se ukvarjam v večjem delu disertacije. Nato predstavim eksaktno rešitev problema razreza z uporabo celoštevilskega linearnega programiranja. Predstavim obsežno eksperimentalno testiranje metode, ki približno pokaže velikostni razred problema, za katerega je ta metoda primerna.

V šestem poglavju najprej predstavim obstoječe hevristične metode za splošni problem razreza. Nato primerjam te rešitve z rezultati iz prejšnjega poglavja ter razvijem pristop na podlagi odločitvenih dreves za izbor primerne metode za posamezen problem. Boljšo od hevrističnih in eksaktno metodo kombiniram v novo metodo, ki vodi do bistveno boljših rezultatov od katere koli od obstoječih metod. V sedmem poglavju se osredotočim na primere, ko razpoložljiv material ne zadošča za izpolnitev potreb, in razvijem pristop, ki omogoča, da poleg izgube materiala pri odločitvah o načrtu razreza upoštevamo tudi druge relevantne dejavnike.

V osmem poglavju se ukvarjam s pristopi za optimizacijo razreza v več zaporednih časovnih obdobjih. Iščem torej dobro rešitev za daljše obdobje namesto trenutnih, lokalnih optimumov. Obstoječo metodo ustrezno prilagodim tako, da je primerna tudi za kontinuiran razrez. Razvite pristope obsežno eksperimentalno testiram.

Disertacija tako predstavlja pomemben znanstveni napredek pri razvoju novih metod, ki vodijo do nižjih izgub, nekatere pa omogočajo tudi upoštevanje dodatnih kriterijev. Predstavljen je pristop za izbor ustrezne metode. Pomemben prispevek k temu področju pa je tudi inovativno obravnavanje razreza kot kontinuiranega poslovnega procesa v podjetju.

Ključne besede: problem razreza, kontinuiran razrez, prenova poslovnih procesov, hevristične metode

CUTTING STOCK PROCESS OPTIMIZATION IN CONSECUTIVE TIME PERIODS

Summary

A one-dimensional cutting-stock problem often arises in practice. The usual problem is to use the material in stock (a predetermined number of stock lengths with a certain length) to cut a requested number of order lengths for either sale or further production. The most frequent goal is a reduction of trim loss, while other goals such as a reduction in cost and cutting time, a smaller number of changes in cutting patterns, inventory etc., may also be followed.

It is known that stock-cutting problems are generally NP-complete and a solution can mostly be found by employing approximate methods and heuristics. The use of exact methods is limited to very small problems.

The thesis deals with a general one-dimensional cutting-stock problem (G1D-CSP) where all stock lengths can differ. The problem is a generalization of the standard problem where all stock lengths are equal – for these problems a pattern-oriented approach is usually applied. The difference in stock lengths can arise for various reasons – the return of partly used stock lengths to the warehouse, the ordering of different stock lengths in order to increase the number of possible combinations or due to natural conditions.

In addition to reducing the trim loss in a given time period where optimisation is not limited to a single order, the thesis identifies the problem of most methods developed previously that only optimise cutting at a given moment in time. Stock cutting can namely be treated as a permanent business process in a company in which consecutive order sets need to be fulfilled either for production needs or for its customers. The exact demand for future orders is often not known in advance. The unutilised and partly utilised stock lengths left after fulfilling current order sets can be stored and used later. The goal is to reduce the trim loss and costs over a broader timespan. A new approach is suggested in the thesis whereby a previously developed method for G1D-CSP is modified.

Further, efficient cutting can make the biggest contribution to a company's overall competitiveness if it is properly connected to other processes in the company, especially the procurement process. The fulfilment of material needs can be more efficient and cheaper if the relationship with suppliers in a supply chain is properly managed. Better co-operation with suppliers is more easily achieved in companies that are more process-oriented (McCormack, Johnson, 2000; Lockamy, McCormack, 2004). Nevertheless, in previous research cutting stock has rarely been treated as one of the processes in a company or a supply chain.

Therefore, the thesis firstly analyses the role of cutting stock as one of the (sub)processes in a company. The role of business process modelling in integration projects is shown. The main problems and new research questions are identified at the end of the chapter.

Then the previously developed methods for optimising the cutting stock are thoroughly reviewed. First, the main problems in the development of new algorithms (mostly arising from NP completeness) and some approaches to limiting them are examined. Two previously developed classifications for cutting-stock problems are presented. The rest of the thesis deals mainly with G1D-CSP.

The studied version of the cutting-stock problem is then defined. The role of the re-use of partly used material is emphasised. The rest of the thesis studies various versions of the defined problem. First, integer linear programming is used to develop the exact method. The method is experimentally tested to determine the approximate problem size for which it is suitable.

Then the existing heuristic methods for G1D-CSP are reviewed. The heuristic solution is compared with the results of the exact method. Decision trees are used to develop a new approach to selecting the appropriate method for each problem size. The better of the heuristic methods is subsequently combined with the exact method to develop a new method, C-CUT, that produces considerably better results than any of the previously developed methods. After that I focus on problems where the available material does not suffice to cover all needs. The newly developed approach enables the inclusion of other relevant factors in preparation of the cutting plan.

Finally, the focus shifts to optimisation over consecutive time periods. I seek a good solution for a longer time period instead of a search for local optimums. I point to the problems of several existing methods that only offer good results for a given moment in time. The existing method is properly adapted to make it suitable for continuous cutting. The method is thoroughly experimentally tested.

In such a way the thesis makes an important scientific contribution to the development of new methods that lead to lower losses and costs, while enabling the inclusion of other criteria and optimisation over a longer time period. An approach to the selection of the appropriate method is presented. Another significant contribution is the innovative analysis of cutting stock as one of the continuous processes in a company or a supply chain.

Keywords: cutting stock problem, continuous cutting, business process renovation, heuristic methods

KAZALO

1	UVOD	1
1.1	Predstavitev problematike.....	1
1.2	Namen in cilji dela.....	2
1.3	Teza in podteze doktorske disertacije.....	3
1.4	Metodologija	4
1.5	Potencialni prispevek k razvoju znanosti.....	5
1.6	Omejitve dela	5
1.7	Struktura disertacije.....	6
2	RAZREZ KOT MEDORGANIZACIJSKI PROCES	7
2.1	Opredelitev pojmov.....	7
2.2	Pomen obravnavanja razreza kot poslovnega procesa	8
2.3	Osnovne značilnosti managementa medorganizacijskih procesov.....	14
2.3.1	Modeliranje poslovnih procesov	15
2.3.2	Procesne sheme	16
2.4	Študija primera preskrbe in razreza materiala.....	18
3	PREDSTAVITEV METOD ZA RAZREZ	27
3.1	Opredelitev razreza.....	27
3.2	Metode za dvodimenzionalni razrez/pakiranje	31
3.3	Enodimenzionalni razrez	33
3.3.1	Splošni pristopi k reševanju.....	33
3.3.2	Metode za rešitev enodimenzionalnega problema razreza	36
3.3.3	Večkriterijske metode za razrez	40
4	DEFINICIJA PROBLEMA RAZREZA	47
5	EKSAKTA REŠITEV.....	53
5.1	Pristop razveji in omeji.....	53

5.2	Rezultati z eksaktno metodo.....	56
6	HEVRISTIČNE REŠITVE	67
6.1	Obstoječe hevristične rešitve	67
6.2	Primerjava eksaktne in hevristične rešitve	70
6.2.1	Izbor metode.....	73
6.2.2	Predstavitev odločitvenih dreves	73
6.2.3	Uporaba odločitvenih dreves	80
6.3	Kombinacija hevristične in eksaktne metode (C-CUT)	90
6.3.1	Rezultati C-CUT-a	98
7	MODEL OD UPOŠTEVANJU STROŠKOV NEIZPOLNITVE.....	106
7.1	Model za reševanje.....	108
7.2	Rezultati	110
8	OPTIMIZACIJA KONTINUIRANEGA RAZREZA.....	117
8.1	Kontinuiran razrez – model 1	121
8.2	Rezultati – Model 1	123
8.2.1	Rezultati model 1 – zaporedna časovna obdobja.....	124
8.2.2	Analiza rezultatov.....	128
8.3	Model ob stroških vračanja palic v skladišče.....	130
8.3.1	Rezultati modela s stroški vračanja	132
8.3.2	Nadaljnje razširitve modela	137
9	SKLEP.....	138
9.1	Ključna spoznanja.....	138
9.1.1	Metode za razrez	139
9.1.2	Metode z dodatnimi kriteriji	139
9.1.3	Izbor metode.....	140
9.1.4	Razrez kot kontinuiran poslovni proces	140
9.2	Nadaljnje delo.....	141
10	LITERATURA	142

KAZALO TABEL

Tabela 1: Število objavljenih člankov o razrezu	12
Tabela 2: Vpliv hitrejšega računalnika na polinomske in eksponentne algoritme	31
Tabela 3: Parametri za 27 testnih primerov	62
Tabela 4: Rezultati za $UB = \min s_i$ ob različnih časovnih limitih.....	65
Tabela 5: Rezultati za $UB = \max s_i$ ob različnih časovnih limitih	66
Tabela 6: Primerjava rezultatov doseženih z eksaktno metodo in rezultatov CUT	71
Tabela 7: Prvih 150 generiranih kombinacij parametrov in njihove rešitve.....	83
Tabela 8: Izbira ustrezne metode na podlagi odločitvenih dreves	87
Tabela 9: Korelacija med vhodnimi spremenljivkami in optimalnostjo rešitve	89
Tabela 10: Primerjava rezultatov med metodo CUT in metodo C-CUT	104
Tabela 11: Podatki o dolžinah palic	126
Tabela 12: Podatki o dolžinah in številu naročil.....	127
Tabela 13: Izgube materiala, število palic in čas reševanja po posameznih obdobjih ob različnih nastavitvah zgornje meje.....	127
Tabela 14: Osnovne prednosti in slabosti posamezne metode.....	130
Tabela 15: Izgube materiala, skupni stroški, število palic in čas reševanja po posameznih obdobjih ob različnih nastavitvah zgornje meje	134
Tabela 16: Stroški razreza, diskontirani na 1. obdobje (mesec)	136

KAZALO SLIK

Slika 1: Prikaz procesa razreza in povezanih procesov	9
Slika 2: Osnovni elementi tehnike procesnih shem	17
Slika 3: »Kot-je« model dobavitelja	19
Slika 4: »Kot-je« model proizvajalca	20
Slika 5: Kot-bo integrirani model	24
Slika 6: Izboljšana rešitev problema iz (Gradišar et al., 1999).....	56
Slika 7: Prikaz sheme procedure PROGEN, uporabljene za generiranje testnih problemov	61
Slika 8: Diagram poteka za eksperiment	63
Slika 9: Shema procedure PROGEN, uporabljene za generiranje testnih problemov za izdelavo odločitvenega drevesa	81
Slika 10: Odločitveno drevo za izbiro ustrezne metode za rešitev problema razreza glede na vrednosti parametrov	85
Slika 11: Prikaz algoritma metode C-CUT	96
Slika 12: Shema procedure PROGEN, uporabljena za pridobitev 270 podprimerov ..	99
Slika 13: Začetna rešitev, pridobljena s CUT algoritmom	100
Slika 14: Rešitev podproblema v drugi fazi algoritma	102

Slika 15: Rešitev problema enodimenzionalnega razreza ob upoštevanju stroškov neizpolnitve v 1. obdobju.....	111
Slika 16: Rešitev problema enodimenzionalnega razreza ob upoštevanju stroškov neizpolnitve v 2. obdobju.....	112
Slika 17: Podatki za primer.....	115
Slika 18: Rešitev brez upoštevanja stroškov neizpolnitve (model za premalo materiala)	115
Slika 19: Rešitev ob upoštevanju stroškov neizpolnitve.....	116
Slika 20: Rešitev problema razreza z Modelom 1	123
Slika 21: Rešitev problema ob upoštevanju stroškov vračanja.....	133

1 Uvod

1.1 Predstavitev problematike

Enodimenzionalni razrez materiala se pogosto pojavlja v praksi (Wäscher et al., 2007). Običajno je potrebno iz materiala, ki je trenutno na zalogi (na zalogi imamo torej različno število palic različnih dolžin), narezati zahtevano število palic vnaprej določenih dolžin za potrebe proizvodnje ali nadaljnje prodaje. Najpogostejši cilj priprave načrta razreza je zmanjšanje izgube materiala, včasih pa želimo doseči še zmanjšanje stroškov ali časa razreza, števila menjav nastavitev rezila, zmanjšanje zaloge in podobne cilje.

Večina različic problema razreza je NP težkih, tako da algoritem polinomske časovne zahtevnosti za pridobitev optimalne rešitve najverjetneje ne obstaja. V praksi so zato za večje probleme uveljavljene različne hevristične metode.

Na tem mestu od predhodno razvitih metod omenjam le nekatere zanimivejše, kot so denimo različne metode na podlagi kombinacije generacije stolpcev s pristopom razveji in omeji (Degraeve, Vandebroek, 1998), metoda za razrez palic različnih dolžin (Belov, Scheithauer, 2002) ter materiala različnih kakovosti (Carnieri et al., 1993), možnost minimiziranja števila uporabljenih vzorcev rezanja (Umetani et al., 2003; Vanderbeck, 2000) ter stroškov menjave vzorcev (Shilling, Georgadis, 2002), metoda razreza na osnovi zaokroževanja neceloštevilskih rezultatov (Scheithauer, Terno, 1995) ter več večfaznih metod (Westerlund et al., 1998; Faggioli, Bentivoglio, 1998; Carvalho, Rodrigues 1995; Zak, 2002; Zak, 2002a). Slednje uporabljajo različne pristope za razrez polizdelkov ter morebitno kasnejše izboljšanje rezultatov. Zanimive so tudi metode, prilagojene za posamezno panogo, denimo za jeklarsko (Vasko et al., 1999; Armbuster, 2002; Ferreira et al., 1990), lesarsko (Rönnqvist, 1995; Wagner, 1999; Čížman, Černetič, 2004; Čížman, Urh, 2006), kovinarsko (Chu, Antonio, 1999) in avtomobilsko industrijo (Dowsland et al., 2007). Podrobneje predhodno razvite metode predstavljam v tretjem poglavju.

Čeprav so bile torej v preteklih letih razvite mnoge metode za optimizacijo različnih vidikov razreza, pa na obravnavanem področju ostaja še precej neraziskanega. Že pri iskanju osnovnega cilja, torej čim manjše izgube ali porabe materiala, obstoječe metode sicer vodijo do dobrih rezultatov, ki pa jih je mogoče še izboljšati. V doktoratu se ukvarjam s splošnim problemom enodimenzionalnega razreza, kjer so lahko vse dolžine palic različnih dolžin (Gradišar et al., 1997). Tak problem je posplošitev standardnega problema, kjer je na razpolago le ena ali nekaj standardnih dolžin palic – za take probleme se običajno uporablja metoda vzorcev na podlagi Gilmore-Gomoryjeve metode (Gilmore, Gomory, 1961; 1963).

Do različnih dolžin palic v skladišču lahko pride zaradi različnih razlogov: zaradi vračanja delno narezanih palic v skladišče, namernega naročanja različnih dolžin zaradi večjega števila kombinacij pri razrezu ali naravnih danosti.

Poleg zmanjševanja izgube v danem časovnem trenutku, pokažem tudi na pristope k razreševanju pomembne pomanjkljivosti večine obstoječih metod za različne vrste razreza. Le-te namreč obravnavajo samo razrez v nekem danem trenutku – iščejo torej dober ali optimalen načrt razreza za dani material in dane potrebe. To je le približna slika realnosti v večini podjetij, ki se z razrezom ukvarjajo kontinuirano in potrebujejo načrte razreza za več zaporednih, med seboj povezanih obdobj. Želijo izpolnjevanje ciljev (denimo majhne izgube) za skupek vseh obdobj in ne samo za posamezno obdobje. Lokalni optimumi ne vodijo nujno do dobrega načrta razreza za celotno obdobje. Zato v disertaciji posebno pozornost namenim preučevanju ustreznosti metod za trenutno optimizacijo v daljšem časovnem obdobju.

Učinkovit razrez pa lahko h konkurenčnosti podjetja največ prispeva, če je ustrezno vključen v druge procese v podjetju, predvsem v proces preskrbe z materialom, s katerim je tesno povezan. Zagotavljanje potrebnega materiala za razrez je lahko učinkovitejše in cenejše ob uspešnem upravljanju odnosov z dobavitelji v oskrbovalni verigi. Uspešno poslovanje z dobavitelji oziroma kupci pa je lažje ob večji procesni usmerjenosti (McCormack, Johnson, 2000; Lockamy, McCormack, 2004) in posledični integriranosti procesov. Omenjeno področje pridobiva na pomenu, vendar razrez do sedaj še ni bil obravnavan kot eden od procesov v oskrbovalni verigi.

Zato v uvodnem delu prikažem vlogo in mesto razreza kot enega od medorganizacijskih poslovnih procesov. V njem običajno sodelujeta dve podjetji – dobavitelj materiala, ki izpolnjuje naročilo kupca po vhodnem materialu, ter naročnik, ki mora iz naročenega materiala narezati želeno število kosov. Razvoj in uporaba ustreznih metod za razrez v danem trenutku je vsekakor predpogoj za uspešno izvajanje procesa razreza materiala, vendar ni hkrati tudi zadosten pogoj. V drugem poglavju tako pokažem, kako lahko učinkovit management oskrbovalne verige in medorganizacijskih poslovnih procesov dodatno prispeva k boljšemu doseganju ciljev podjetja oziroma celotne verige.

1.2 Namen in cilji dela

Namen doktorske disertacije je predvsem nadgraditev teoretičnih spoznanj o zniževanju izgube materiala ter celotnih stroškov pri razrezu materiala v zaporednih časovnih obdobjih ter teoretična obravnavo možnosti za učinkovitejšo in uspešnejšo uporabo enodimenzionalnega razreza kot enega od kontinuiranih procesov v podjetju.

Pri tem izboljšam do sedaj razvite metode za optimizacijo enodimenzionalnega razreza, omogočim širše obravnavanje problema razreza ter pokažem na nekatera pomembna raziskovalna vprašanja, ki zahtevajo nadaljnjo znanstveno obravnavo

Osnovni cilji doktorske disertacije so naslednji:

1. analizirati vpliv razreza kot enega od procesov v podjetju ter odprta raziskovalna vprašanja pri obravnavi procesa razreza,
2. identificirati trenutno stanje pri raziskovanju razreza s posebnim poudarkom na splošnem enodimenzionalnem razrezu ter kritično ovrednotiti obstoječe metode,
3. predstaviti eksaktno metodo (Trkman, Gradišar, 2002) za rešitev splošnega enodimenzionalnega problema razreza, ki vodi do optimalnih rešitev za probleme manjšega obsega,
4. predstaviti pristop za ugotavljanje približne meje (Trkman, Gradišar, 2003a), za katere je eksaktna metoda primerna, ter kriterije, ki bodo omogočali izbor ustrezne metode za posamezen problem,
5. razvito eksaktno metodo uporabiti v kombinaciji z obstoječo hevristično metodo za predstavitev in analizo kombinirane metode (Gradišar, Trkman, 2005), ki vodi do skoraj optimalnih rešitev tudi za večje probleme. Osnovni cilj je, da taka kombinirana metoda nudi boljše rešitve od hevrističnih metod in hkrati za razliko od eksaktne metode ohranja tudi nizko časovno zahtevnost za velike probleme,
6. predstaviti pristop za optimizacijo razreza za primere s premalo materiala (Trkman, Gradišar, 2003), kjer bodo poleg izgube materiala pri odločitvi o tem, kateri kosi naročila bodo ostali neizpolnjeni, upoštevani tudi drugi kriteriji,
7. razviti in testirati pristop za kontinuiran razrez materiala, kjer bo metoda vodila do dobrih rezultatov v daljšem obdobju in ne le v posameznem primeru,
8. predlagati utemeljeno razširitev obstoječe Dyckhoffove klasifikacije (Dyckhoff, 1990) za probleme razreza in pakiranja, ki bo omogočala ustrezno umestitev novo razvite metode.

1.3 Teza in podteze doktorske disertacije

Osnovna teza doktorske disertacije je:

Prilagoditev optimizacijske metode za enodimenzionalni razrez, ki kombinira eksaktne in hevristične pristope, za uporabo v zaporednih časovnih obdobjih vodi do nižje izgube materiala in posledično nižjih stroškov izvajanja poslovnega podprocesa razreza.

Iz osnovne teze izhajajo naslednje podteze:

- eksaktni pristop pri splošnem problemu enodimenzionalnega razreza omogoča znižanje izgube materiala za računsko manj zahtevne probleme,
- kombinacija eksaktnega in hevrističnega pristopa pri enodimenzionalnem razrezu omogoča znižanje izgube materiala ne glede na računsko zahtevnost problema,
- optimizacija izgube materiala/stroškov v danem trenutku lahko vodi do bistveno slabših rezultatov v daljšem časovnem obdobju,

- prilagoditev optimizacijskih metod za uporabo v zaporednih časovnih obdobjih omogoča manjšo izgubo materiala in nižje stroške na daljši rok,
- prenova poslovnih procesov v podjetju lahko prispeva k boljšemu odvijanju podprocesa razreza.

1.4 Metodologija

Obstoječe stanje na danem znanstvenem področju bom ugotovil z obsežnim pregledom literature, pri čemer bom podrobneje analiziral metode za primerljive probleme enodimenzionalnega razreza.

Za eksaktno reševanje manjših problemov enodimenzionalnega razreza bom uporabil celoštevilsko linearno programiranje in metodo razveji in omeji (angl. branch & bound). Enak pristop bom uporabil za reševanje problema s premalo materiala ob upoštevanju stroškov neizpolnitve.

Primere za testiranje različnih razvitih metod bom pridobil z uporabo predhodno razvitega generatorja naključnih problemov razreza PGEN. Zato bom lahko ustreznost vseh razvitih in predstavljenih metod obsežno testiral.

Za izbiro metode bom uporabil eno od tehnik rudarjenja po podatkih, in sicer odločitvena drevesa (Quinlan, 1993), s pomočjo katerih bom izdelal pravila za izbor metode za posamezen problem. Vpliv posameznih dejavnikov na rešljivost problema z eksaktno metodo bom dodatno analiziral z uporabo korelacijske analize.

Pri kontinuiranem razrezu bom za reševanje problema v posameznem obdobju ravno tako uporabil celoštevilsko linearno programiranje in metodo razveji in omeji.

Pri modeliranju poslovnih procesov bom uporabil tehniko procesnih shem (angl. process maps) za prikaz sedanjega in želenega bodočega stanja ter njuno medsebojno primerjavo.

Pri uporabi zgoraj navedenih metodoloških pristopov bom uporabil naslednjo programsko opremo:

- generator naključnih števil PGEN,
- reševalec (angl. solver) za celoštevilске linearne programe MPL/CPLEX,
- program za delo s preglednicami MS Excel,
- programska jezika Visual Basic for Applications ter Fortran,
- program za izdelavo odločitvenih dreves C5/See5,
- program za statistične analize SPSS,
- program za modeliranje in simulacije poslovnih procesov iGrafx Process 2006.

Metodologija posameznega dela doktorske disertacije je podrobneje predstavljena v ustreznem poglavju.

1.5 Potencialni prispevek k razvoju znanosti

Doktorska disertacija pomeni pomemben napredek na naslednjih področjih:

1. izboljšava obstoječih metod za splošni enodimenzionalni razrez:
 - a. razvoj nove metode C-CUT, ki ob nizki časovni kompleksnosti privede do bistveno boljših rezultatov od obstoječih metod (Gradišar et al., 1997; Gradišar et al., 1999). Metoda je uporabna v vseh praktičnih situacijah, kjer se srečujemo s splošnim enodimenzionalnim problemom razreza, ne glede na velikost problema,
 - b. prikaz uporabe odločitvenih dreves za izbor ustrezne metode: omogoča verjetni izbor boljše metode za posamezen problem,
 - c. razvoj pristopa, ki omogoča upoštevanje stroškov neizpolnitve: v primerih, ko material ne zadošča za pokritje vseh potreb po narezanem materialu, dosedanje metode kot edini cilj želijo doseči čim nižjo izgubo materiala. Razviti pristop omogoča tudi upoštevanje drugih dejavnikov pri odločitvi, katera naročila bodo ostala nenarezana. Na ta način se lahko bolje doseže cilj podjetja v danem trenutku.
2. Razvoj metod za optimizacijo pri kontinuiranem razrezu: pristop, ki vključuje vračanje palice v skladišče ter analizira vpliv razreza v več zaporednih časovnih obdobjih, omogoča podjetju doseganje ciljev v celotnem obdobju razreza namesto samo pri enem načrtu razreza v danem trenutku.
3. Povezava razreza s celotnim procesom preskrbe z materialom in preostalimi procesi v oskrbovalni verigi: pričujoča disertacija je prva, ki razrez obravnava v širšem kontekstu. Analiziramo vpliv uspešnejšega upravljanja z oskrbovalno verigo na proces razreza in konkurenčnost verige kot celote.

1.6 Omejitve dela

Najprej je potrebno poudariti, da se v doktorski disertaciji ukvarjamo s splošnim problemom enodimenzionalnega razreza (Gradišar et al., 2002). Razvite metode je možno sicer uporabiti tudi pri splošnem problemu razreza (Dyckhoff, 1990), vendar so običajno za take probleme primernejše metode na podlagi vzorcev (Gilmore, Gomory, 1961; 1963). Pristop, razvit v tej disertaciji, je zato primeren predvsem za primere z različnimi dolžinami palic.

Planiranje razreza v različnih podjetjih je zelo različno, ravno tako so različni tudi ključni dejavniki uspeha posameznega podjetja. Zato vključeni kriteriji in razvite metode niso nujno ustrezni za podjetja v vseh panogah in na vseh nivojih oskrbovalne verige. Poleg tega hevristične

metode (kot so z izjemo ene tudi vse v tej disertaciji razvite) lahko niso več ustrezne že ob manjših spremembah vhodnih podatkov ali kriterijske funkcije.

Reševanje z linearnim programiranjem je primerno samo za manjše primere. Omejitve uporabe eksaktne metode v disertaciji temeljito analiziram in pokažem na možne rešitve.

V disertaciji eksplicitno ne obravnavam različnih modelov za naročanje materiala, ki ga je potrebno razrezati. Konkretno modele za naročanje je mogoče razmeroma enostavno vključiti v predstavljeno metodologijo za analizo koristi upravljanja z oskrbovalno verigo ter z njimi razširiti tudi metode za optimizacijo kontinuiranega razreza.

V disertaciji pri obravnavi stroškov razreza ne vključujem stroškov skladiščenja, razen implicitno pri upoštevanju stroškov vračanja palic v skladišče. Morebitna nadaljnja razširitev predstavljenega raziskovalnega dela bi bila torej vključitev tovrstnih stroškov pri optimizaciji kontinuiranega razreza ter iskanje optimalne ravni zaloge za doseganje ciljev podjetja ali oskrbovalne verige pri razrezu.

Primer procesa razreza, ki je predstavljen v drugem poglavju, je pripravljen na podlagi poznanega odvijanja procesa naročanja in razreza v podjetju in ne predstavlja realne študije razreza v posameznem podjetju.

1.7 Struktura disertacije

V drugem poglavju najprej predstavim vlogo razreza kot enega od (pod)procesov v podjetju oziroma celi oskrbovalni verigi. Obravnavam vlogo modeliranja poslovnih procesov pri integraciji procesa razreza v oskrbovalni verigi. Na koncu identificiram ključne probleme in neodgovorjena raziskovalna vprašanja pri problemu razreza.

V tretjem poglavju najprej podajam obširen pregled literature s področja razreza. Pri tem najprej navedem osnovne probleme pri iskanju dobrih rešitev ter nekatere pristope za odpravo teh problemov. Predstavim splošno klasifikacijo za probleme razreza. Nato se osredotočim predvsem na obširen prikaz predhodno razvitih metod za reševanje problema enodimenzionalnega razreza.

V četrtem poglavju podam besedno in matematično definicijo splošnega enodimenzionalnega problema razreza ob upoštevanju vračanja in ponovne uporabe materiala. Z različicami tega problema se ukvarjam v večjem delu disertacije.

V petem poglavju problem rešim eksaktno z uporabo celoštevilskega linearnega programiranja. Predstavim obsežno eksperimentalno testiranje metode, ki približno pokaže velikostni razred problema, za katerega je ta metoda primerna.

V šestem poglavju najprej predstavim obstoječe hevristične metode za splošni problem razreza. Nato primerjam te rešitve z rezultati iz prejšnjega poglavja ter razvijem pristop na podlagi odločitvenih dreves za izbor primerne metode za posamezen problem. Boljšo od hevrističnih in eksaktno metodo kombiniram v novo metodo, ki vodi do bistveno boljših rezultatov od katere koli od obstoječih metod.

V sedmem poglavju se osredotočim na primere, ko razpoložljiv material ne zadošča za izpolnitev potreb, in razvijem pristop, ki omogoča, da poleg izgube materiala pri odločitvah o načrtu razreza upoštevamo tudi druge relevantne dejavnike.

V osmem poglavju se ukvarjam s pristopi za optimizacijo razreza v več zaporednih časovnih obdobjih. Iščem torej dobro rešitev za daljše obdobje namesto trenutnih, lokalnih optimumov. Obstoječo metodo ustrezno prilagodim tako, da je primerna tudi za kontinuiran razrez. Razvite pristope obsežno eksperimentalno testiram.

Ključne ugotovitve, posledice za teorijo in prakso ter najzanimivejša nadaljnja raziskovalna vprašanja povzamem v sklepu.

2 Razrez kot medorganizacijski proces

2.1 Opredelitev pojmov

Uvodno poglavje disertacije pričnem z opredelitvijo nekaterih ključnih pojmov, ki so potrebni za razumevanje konceptov, predstavljenih v tem poglavju.

Poslovni proces (angl. business process) opredelimo kot zaporedje logično povezanih opravil, ki so potrebne za doseganje zelenega poslovnega učinka (Srivardhana, Pawlowski, 2007; Davenport, Short, 1990). Poslovni proces lahko opredelimo kot takšno sestavo logično med seboj povezanih izvajalskih in nadzornih aktivnosti, katerih posledica je proizvod ali storitev (Kovačič et al., 2004). Podproces je zbirka logično povezanih aktivnosti znotraj večjega procesa (BPR Glossary of Terms, 2008). Aktivnost je največje tako opravilo v poslovnem procesu, ki ga več ni smiselno deliti naprej.

Razrez lahko v podjetju nastopa kot (temeljni) proces ali pa kot podproces v katerem od temeljnih procesov. Kot proces nastopa predvsem pri podjetjih, ki se ukvarjajo predvsem z razrezom (npr. žaga). Kot podproces nastopa v marsikaterem industrijskem podjetju. Ker za koncepte, predstavljene v tej disertaciji, ni bistveno ali razrez nastopa kot proces ali podproces, ta dva izraza v nadaljevanju uporabljam izmenjujoče.

Proces razreza pa je sestavljen iz več aktivnosti. Smiselnost nadaljnjega deljenja je odvisna predvsem od namena modeliranja. Aktivnosti v ožjem (pod)procesu razreza so lahko denimo

sprejem naročila, prenos materiala na mesto razreza, priprava načrta razreza¹, vračanje neuporabljenega materiala in podobne aktivnosti.

Če želimo bolje spoznati izvajanje procesa razreza, običajno uporabljamo modeliranje poslovnih procesov. Modeliranje poslovnih procesov vključuje vse aktivnosti povezane s spremembo znanja o poslovnem sistemu v modele, ki opisujejo procese, ki se izvajajo v organizaciji (Scholz-Reiter, Stickel, 1996; Giaglis, 2001). Procese običajno modeliramo tudi z namenom njihove prenove.

Prenova poslovnih procesov (angl. business process renovation) je strategija, ki kritično pregleda obstoječe poslovne politike in postopke, jih analizira in po potrebi spremeni/izboljša (Prasad, 1999). Cilj prenove procesov so bistvene izboljšave v kritičnih dejavnikih uspeha, denimo stroških, kakovosti, pretočnih časih (Hammer, Champy, 1994). Management poslovnih procesov pa je širši pojem od prenove, saj je to zbirka metodologij, tehnik in orodij, ki podpirajo analizo in izboljšavo poslovnih procesov (Wang, Wang, 2006; Melao, Pidd, 2000). Kombinira prenovo procesov z avtomatizacijo aktivnosti in delovnih procesov. Je skupek managementa procesov, integracije aplikacij in uporabe orodij za management delovnih procesov (Smith, Fingar, 2003).

Oskrbovalna veriga (angl. supply chain) je povezan skupek virov in procesov, ki se začne z zagotavljanjem osnovnih surovin in konča z dostavo dokončanih izdelkov ali storitev končnemu kupcu (Bridgefeld Group ERP/Supply chain Glossary, 2004). Gre za pretok in transformacijo materialov preko proizvodnih in distribucijskih kapacitet do izdelka za končnega kupca (McCormack, Johnson, 2000).

V današnjem svetu si delovanja oskrbovalnih verig ne moremo več predstavljati brez njihove medsebojne informacijske povezave. Medorganizacijski informacijski sistem je splet informacijske in telekomunikacijske tehnologije, aplikacij, standardov in človeških sposobnosti. Je okvir za elektronsko sodelovanje med podjetji, saj omogoča procesiranje, deljenje in posredovanje informacij (Williamson et al., 2004). Tak sistem lahko opredelimo kot sistem za omogočanje operativnega sodelovanja (angl. operational cooperation) (Hong, 2002).

2.2 Pomen obravnavanja razreza kot poslovnega procesa

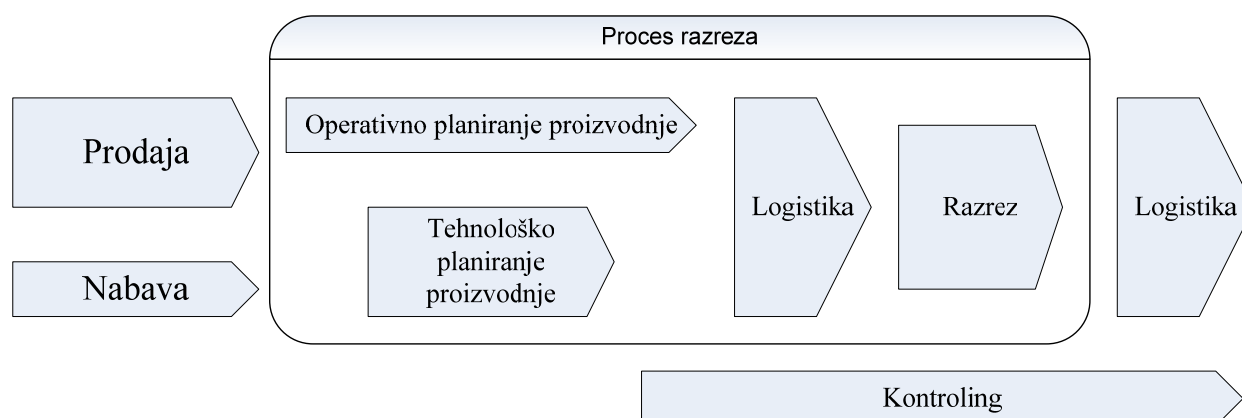
Proces razreza je eden od številnih procesov v okviru podjetja, ki so potrebni za zadovoljevanje pričakovanj strank podjetja in posledično zagotavljanje uspešnosti poslovanja. Običajno pa seveda podjetje ni samostojno pri doseganju omenjenih ciljev, ampak sodeluje z drugimi podjetji v oskrbovalni verigi. Preden se podrobneje posvetim razvoju novih metod za razrez, zato predstavljam vlogo in pomen razreza kot enega od poslovnih medorganizacijskih procesov. Razrez je praviloma (ne pa nujno ali izključno) proces, ki se odvija pretežno na začetku

¹ Večina predhodno razvitih metod (tudi metode v 4. in 5. poglavju tega doktorskega dela) se ukvarja z optimizacijo izključno te aktivnosti.

oskrbovalne verige pri proizvajalcu. Na njegovo izvajanje pa ključno vplivajo zahteve podjetij nižje v verigi, ki morajo zadovoljevati zahteve končnega kupca.

Na sliki 1 je prikazan proces razreza in njegova povezava z preostalimi procesi v podjetju (slika je bila pripravljena na podlagi študije primera večjega slovenskega podjetja; natančnejša metodologija študije primera je v Erjavec et al., 2008). V proces razreza smo v tem primeru uvrstili pripravo načrta razreza, premestitev zalog na mesto razreza in samo rezanje. V modele na slikah 3-5 pa so vključeni tudi povezani procesi, ki so potrebni za razumevanje procesa razreza – to so predvsem naročanje materiala in prodaja narezanih (pol)izdelkov.

Slika 1: Prikaz procesa razreza in povezanih procesov



Vir: (Erjavec et al., 2008)

Zato lahko uspešno usklajevanje poslovnih procesov in management oskrbovalne verige (angl. supply chain management) povratno pomembno vplivata na nekatere dejavnike, ki so ključni za boljše doseganje ciljev pri samem razrezu. Zgolj za ilustracijo navajam nekaj možnih primerov:

- strošek naročila in pretočni čas se lahko zmanjša - denimo ob uspešni uvedbi elektronskega naročanja² – zato lahko podjetje naroča in/ali dobavlja manjše količine materiala. Posledično pa so potrebne ali zaželenе tudi drugačne metode za razrez (denimo povečanje možnosti uporabe eksaktne metode ali pa kontinuiranega razreza);
- v osmem poglavju disertacije v modelu za kontinuiran razrez predpostavljam, da je poznana samo približna razporeditev števila in dolžine naročil. Uspešno izmenjevanje informacij s kupci (oziroma dobro upravljanje s poslovnimi procesi znotraj podjetja, če gre za razrez za potrebe proizvodnje) in uporaba teh informacij lahko zmanjšata negotovost glede prihodnjih potreb. V skrajnem, hipotetičnem, primeru bi lahko celo natančno poznali zahteve po razrezanih naročilih, kar bi omogočilo natančno

² Uvedba e-poslovanja lahko zmanjša stroške naročila v povprečju za približno polovico. Različne raziskave so pokazale zmanjšanje med 30 in 70% transakcijskih stroškov (Gansler et al., 2003, Boulianne, 2005; Puschmann, Alt, 2005; Davila et al., 2003; Presutti, 2003). Pretočni časi se lahko skrajšajo za cca. 20-30% (Presutti, 2003).

načrtovanje števila in dolžine palic ter bi vodilo do razrezov s stalno ničelnimi izgubami ter zelo nizkimi zalogami;

- podobno je s stroški neizpolnitve naročila, ki jih obravnavam v sedmem poglavju: dobro sodelovanje z dobaviteljem in naročniki lahko zmanjša stroške neizpolnitve naročila ter omogoča hitrejši in boljši odziv na nastalo situacijo;
- uspešno usklajevanje procesa razreza v oskrbovalni verigi lahko olajša množično prilagajanje (angl. mass customization) kupcem. Gre torej za preseganje navideznega paradoksa - ponujanje različnih dolžin palic ob ohranjanju prednosti množičnega razreza in majhnih skupnih izgubah materiala. Množično prilagajanje pomeni razvoj, proizvodnjo, trženje in dobavo prilagojenih izdelkov ali storitev na množični osnovi. To je proces, pri katerem morajo biti vsi vidiki organizacije, torej procesi, organizacijske strukture, ljudje in tehnologija usmerjeni k zagotavljanju izdelkov ali storitev, ki jih želi posamezni kupec (Pine, 1992). Predpogoj za to je ustrezna proizvodna strategija, ki omogoča agilno proizvodnjo. Povečano prilaganje zahteva reorganizacijo aktivnosti v proizvodnji in distribuciji za povečanje fleksibilnosti procesa (Fredriksson, Gade, 2005). To je možno doseči samo s kombinacijo znanja, tehnologije, procesov in povezav s partnerji (Brown, Bessant, 2003). Osnovna naloga je torej razvoj procesa, ki lahko dobavlja številne različice po izboru kupcev ob hkratnih kratkih pretočnih časih in stroških (Ahlstrom, Westbrook, 1999). Z možnostmi za skrajševanje pretočnih časov se ukvarjam v tem poglavju, medtem ko se z zmanjševanjem stroškov pri različnih verzijah problema razreza ukvarjam v vseh naslednjih poglavjih. Podrobnejši pregled vloge razreza kot enega od procesov pri masovnem prilagajanju in zahtev, ki jih masovno prilagajanje razreza predstavlja za podjetje, je v (Trkman, Gradišar, 2005d). Tovrstne prilagoditve so toliko pomembnejše, ker se povečuje pomen masovnega prilagajanja ter proizvodnje po naročilu (Krajewski et al., 2005).

Kot je razvidno, večina od zgornjih primerov izhaja prav iz boljšega managementa medorganizacijskih procesov v celotni oskrbovalni verigi, zato temu v tem poglavju posvečam posebno pozornost. Prihranki pri tem so v povezavi s prihranki zaradi boljših metod za razrez lahko zelo pomembni za doseganje konkurenčne prednosti podjetja in verige kot celote. Za predstavljene koncepte ni ključno, ali se razrez odvija pri dobavitelju materiala ali pri kupcu/proizvajalcu končnih izdelkov ali polizdelkov.

Zato v uvodnem delu najprej predstavljam nekatere nove izzive pri optimizaciji razreza kot enega od procesov ter nekatera osnovna spoznanja in načine za uspešnejši management oskrbovalne verige, ki lahko prinese navedene prednosti. Pri tem se še posebej osredotočam na pomen, možnosti in vpliv učinkovitega izmenjevanja informacij o prihodnjih potrebah po narezanem materialu znotraj oskrbovalne verige.

Čeprav se večina člankov s področja optimizacije razreza ukvarja predvsem z izboljšanjem metod za razrez, so se namreč v preteklih letih pojavile trditve, da sama izboljšava v metodi za razrez ne vodi nujno do bistveno povečane učinkovitosti razreza ali večje konkurenčnosti podjetja ali celotne oskrbovalne verige. Kot bom dokazal v zadnjem poglavju te disertacije, lahko trenutna rešitev problema vodi do bistveno slabših rezultatov v daljšem obdobju. Poleg tega izboljšane metode za samo pripravo razreza obravnavajo le nekatere od težav, povezanih s procesom razreza. Zmanjšanje izgube materiala je namreč le del stroškov, ki so povezani s pripravo želene količine naročil določene dolžine.

Sodobne metode namreč vodijo že do skoraj optimalnih rezultatov oz. minimalne izgube materiala (npr. Gradišar et al., 2002), zato se je pozornost raziskovalcev nekoliko preusmerila k metodam z mešanimi cilji (denimo Trkman, Gradišar, 2003; Yang et al., 2006). Tradicionalno pa so glavni kriteriji za ocenjevanje ustreznosti rešitev: izguba materiala, prevelika proizvodnja, povprečna raven zalog, povprečno število različnih dolžin palic in povprečno število palic v skladišču (Venkateswarlu, 2001). Samo tretji kriterij lahko delno povežemo s poslovnim ozadjem procesa razreza, medtem ko preostali kriteriji optimizirajo le eno od aktivnosti v procesu. To je problematično, saj se poleg večkriterijskih metod za razrez v zadnjem času poudarja tudi pomen večih kriterijev za ocenjevanje uspešnosti oskrbovalne verige ter potrebne razširitve modelov oskrbovalnih verig, ki bi vključevali tako proizvodnjo kot dobavitelje (Meixell, Gargeya, 2006). Primer takega pristopa predstavim v poglavju 2.4.

Kot pokažem v nadaljevanju, so obstoječe metode za trenutno optimizacijo lahko neprimerne pri uporabi v daljšem časovnem obdobju. Podobno sta Gramani in Franca (2006) ugotovila, da optimalna rešitev za povezan problem razreza in določanja velikosti proizvodne serije vsebuje neoptimalne rešitve za vsak posamezen problem. Chauhan s soavtorji (2008) predstavlja zanimiv vpogled v različne točke naročila razrezanih materialov v oskrbovalni verigi v papirni industriji. Čeprav v članku poudarijo pomen študije pretočnih časov in možnosti za njihovo skrajševanje pri zagotavljanju narezanega materiala, pa predstavljena metoda minimizira samo vsoto stroškov zalog in izgube materiala.

Kot bom dokazal v nadaljevanju, sodobne metode za optimizacijo razreza vodijo do skoraj optimalnih rezultatov z zelo nizko izgubo za trenutno rešitev problema razreza. Zato nadaljnje izboljšave metod za pripravo razreza same po sebi verjetno ne morejo voditi do bistvenih izboljšav v poslovanju podjetja. To velja še toliko bolj, ker v sodobnem svetu ni več dovolj preučevanje lokalne optimizacije za posamezno operacijo in nato združevanje lokalnih optimumov v skupni plan (Kobayashi et al., 2003).

Čeprav lahko v literaturi opazimo trditve o povečevanju pomena razreza in objavljenih člankov³ s tega področja (Wäscher et al, 2007, Gradišar et al., 1999a), pa lahko v tabeli 1 opazimo, da je

³ Ključne članke, objavljene v zadnjih letih, predstavljam v tretjem poglavju.

število objavljenih člankov v vodilnih svetovnih revijah v zadnjih letih bolj ali manj konstantno. To bi lahko kazalo na zrelost tega raziskovalnega področja.

Tabela 1: Število objavljenih člankov o razrezu

leto	št. člankov ¹	št. člankov ²
1995	40	13
1996	25	11
1997	61	8
1998	39	10
1999	60	13
2000	42	8
2001	41	8
2002	47	20
2003	48	14
2004	40	15
2005	/	18
2006	/	22
2007	/	25

Vir: ¹Wäscher et al, 2006; ²Web of Science, 2008 (iskalni niz "cutting stock problem"); lastna analiza

Velika večina člankov o problemu razreza je objavljenih v vodilnih revijah s področja operacijskih raziskav. Čeprav so se raziskave pojavile tudi v revijah, kot so denimo Naval Research Logistics (Klamroth, Wiecek, 2000), Journal of Intelligent Manufacturing (Onwubolu, Mutingi, 2003), International Journal of Production Economics (Sulliman, 2001) in Forrest Products Journal (Carnieri et al., 1993), pa obravnave problema razreza kot enega od poslovnih procesov v podjetju v svetovni literaturi nisem zasledil (delne izjeme so navedene v nadaljevanju). Še več: medtem ko je pojav konceptov managementa oskrbovalne verige imel minimalen vpliv na teorijo zalog (Buxey, 2006), se zdi, da je ta vpliv na problem razreza tako rekoč neobstoječ. Zato večina novo razvitih metod za optimizacijo razreza še vedno kaže temeljne pomanjkljivosti kvantitativnih pristopov, ki so bile opisane že pred desetimi leti (Fowler, 1998) – metode so preveč abstraktne, oddaljene od odločevalcev, ob tem da so modeli preveč enostavni. Idealno bi bilo, če bi bile metode za razrez integrirane v celovito programsko rešitev (angl. enterprise resource planning – ERP) podjetja (Hoffman, 2000). Le tako lahko pričakujemo povečanje vpliva metod operacijskih raziskav na poslovanje podjetij.

Večina metod za razrez tudi predpostavlja, da ima odločevalec na razpolago točne in popolne informacije, kar pa v praktičnih primerih le redko drži, saj je pretok informacij v podjetjih ali oskrbovalni verigi pogosto počasen in z nepopolnimi informacijami (Trkman et al., 2005c). Zato lahko trdim, da raziskave na področju razreza pogosto trpijo za tako imenovanim »problemom relevantnosti« - raziskovalci ne obravnavajo problemov relevantnih za prakso in odločevalce v praktičnih situacijah. Znanstvena spoznanja v praksi pogosto niso testirana in uporabljena (Bertrand, Fransoo, 2002). Ena od zanimivih izjem, ki ponuja ne samo optimizacijo razreza

ampak tudi njegovo povezavo s planiranjem proizvodnje in zalog, je denimo študija lesarske industrije, predstavljena v (Rönnqvist, 2003).

Obravnavanje razreza kot enega od procesov v podjetju je skladno s spoznanjem, da lahko podjetja izboljšajo svoje poslovne rezultate s privzetjem procesnega pogleda. Predhodne raziskave so pokazale, da podjetja na višji ravni procesne zrelosti konstantno dosegajo boljše rezultate (glej (Lockamy, McCormack, 2004; Škrinjar et al., 2007) za statistično analizo ali (Jaklič et al., 2006; Indihar Štemberger et al., 2006) za dve študiji primerov). V tem podpoglavju zato navajam nekatere študije na področju razreza, ki predstavljajo premik proti usmeritvi k izboljšanju procesov (Burgess et al., 2006). Prenova procesa razreza je še toliko pomembnejša pri izboljšanju storitve za kupca, saj sta pomembni značilnosti pri kakovosti storitve dolžina in zanesljivost časa izvedbe razreza (Denton et al., 2003).

Zanimiva študija izvedbe razreza in pakiranja v električni industriji je v (Cochran, Ramanujam, 2007). Študija poudari medsebojno povezanost različnih odločitev v oskrbovalni verigi, kot so problem pakiranja, izbor embalaže ter vključitve dodatnih storitev z dodano vrednostjo za kupca. Potrebno je torej izbrati najboljšo kombinacijo alternativ, ki jih ima na razpolago proizvajalec, za zmanjšanje celotnih stroškov. Podobno kot v tem poglavju tudi (Weng, Sung, 2008) poudarita, da sama optimizacija razreza ni dovolj in da je potrebno optimizirati celotno proceduro⁴ izvedbe procesa. Njun članek nato predstavi metodo za optimizacijo vsote izgube materiala in učinkovitosti dela, medtem ko celotne »procedure« (oz. poslovnega procesa) ne obravnavata ali optimizirata. Podobno je bil razrez materiala v papirni industriji opredeljen kot ena od odločitev pri terminiranju proizvodnje papirja (Keskinocak et al., 2002).

Povezava problema planiranja in razreza v papirni oskrbovalni verigi je predstavljena v (Rodriguez, Vecchietti, 2008), ki v veliki meri sledi ugotovitvam tega poglavja. Poudarjen je pomen zmanjševanja izgube materiala – tako zaradi zniževanja stroškov verige kot zaradi varovanja okolja. Razviti optimizacijski model za zmanjševanje izgube materiala je implementiran kot modul poslovne inteligence in je povezan z ERP sistemom podjetja, ki omogoča delitev informacij s kupci in dobavitelji v oskrbovalni verigi. Na ta način se hkrati upošteva izguba materiala, zahteve po energiji, vpliv na okolje in stroški proizvodnje. Avtorja posebej poudarita tudi izboljšano ravnanje z zalogami kot posledico bolj natančnih in ustrezno statistično obdelanih informacij, ki jih omogoča tak sistem ter posebej poudarita pomen uporabe tovrstnega sistema v oskrbovalni verigi. Njun prispevek je tako eden prvih, ki sledi izhodiščem in možnostim, predstavljenim v začetku tega podpoglavja

⁴ Avtorja v članku, ki delno gradi na mojem delu, sicer uporabita termin procedura, vendar je iz konteksta članka očitno, da obravnavata koncept, ki je v svetovni literaturi običajno poimenovan kot »poslovni proces«.

2.3 Osnovne značilnosti managementa medorganizacijskih procesov

Pomen managementa poslovnih procesov je toliko večji, ker čedalje bolj osnovno težišče konkurence ni več med posameznimi podjetji ampak običajno tudi med posameznimi oskrbovalnimi verigami (Li et al., 2005; Trkman et al., 2007; Ketchen, Hult, 2007). Zato tudi za področje razreza materiala velja, da podjetje svoje cilje bistveno lažje dosega ob ustrezni usklajenosti v celotni oskrbovalni verigi. Ker je zadovoljstvo končnega kupca ključnega pomena za uspešnost celotne verige, je izjemno pomemben učinkovit management vseh procesov v verigi. Pri tem je proces zagotavljanja razpoložljivega materiala v pravem obsegu in pravih dolžinah lahko eden od pomembnejših poslovnih procesov.

V preteklih letih so se za management oskrbovalne verige razvile mnoge nove tehnološke rešitve in organizacijski koncepti. Večina teh idej je tako ali drugače povezana z razvojem informacijske tehnologije (IT) in učinkovitejših transportnih poti (Das, Abdel-Malek, 2003). Vendar le redka podjetja te možnosti uporabljajo strateško v oskrbovalni verigi za doseganje polne konkurenčne prednosti, medtem ko mnoge razvijajo in uvajajo neustrezne rešitve (Cox et al., 2001), zaradi česar se celo pojavljajo dvomi o učinkovitosti uporabe managementa oskrbovalne verige kot orodja za izboljševanje operativne učinkovitosti podjetja (Hsu, 2005). Praktične izkušnje kažejo, da temeljni razlogi za neuspešnost niso tehnološki, ampak so posledica organizacijskih in procesnih težav (Jaklič et al., 2003).

Računalniško implementirane metode operacijskih raziskav, predstavljene v nadaljevanju, lahko bistveno prispevajo k manjši izgubi materiala pri razrezu, če so ustrezno integrirane v sistem za podporo odločanja v organizaciji (Čižman, Černetič, 2004). Vendar so te izgube le manjši del celotnih stroškov zagotavljanja materiala, zato lahko učinkovit management oskrbovalne verige, ki je predstavljen v tem poglavju, deluje bistveno širše. Za polno doseganje možnih prednosti pa je seveda potrebna povezava vsega omenjenega, kar je predstavljeno v tej disertaciji – učinkovitih metod za razrez materiala v danem trenutku; metod, ki omogočajo minimizacijo stroškov pri kontinuiranem razrezu materiala; metod za zmanjšanje stroškov ob primanjkljaju materiala; ter metod za uspešen management oskrbovalne verige, ki zmanjšuje stroške in povečuje učinkovitost v celotni verigi.

Zato je temeljni namen tega poglavja disertacije pokazati, da rešitve IT in operacijskih raziskav same po sebi ne morejo v celoti prispevati k izboljšanju celotnega procesa zagotavljanja materiala. Strateško prednost lahko prinesejo samo, če upoštevamo, da je:

- ustrezno modeliranje poslovnih procesov predpogoj za učinkovit management in prenovno poslovnih procesov (angl. Business Process Management oziroma Business Process Reengineering),
- uspešno poslovanje in management celotne oskrbovalne verige je možno predvsem ob učinkovitem managementu poslovnih procesov v vseh podjetjih v verigi.

Medtem ko so lahko pristopi k managementu in prenovi poslovnih procesov v praksi različni, pa noben od njih ni izvedljiv brez predhodnega natančnega poznavanja notranjih (internih)⁵ in zunanjih (eksternih) poslovnih procesov. Modeli poslovnih procesov tako igrajo pomembno vlogo v različnih fazah vzpostavljanja ali prenove poslovnih procesov ne glede na dejansko uporabljeno metodologijo (Desel, Ervin, 2000).

Management poslovnih procesov naj se ne bi uporabljal samo na ravni enega podjetja, ampak tudi na ravni celotne verige. Le tako ravnanje lahko privede do polnih prednosti tudi pri razrezu materiala (nekatero od teh prednosti sem navedel tudi na začetku tega poglavja). Številne od teh sprememb so neposredno ali posredno povezane s pretokom in uporabo informacij. Tu lahko e-poslovanje (predvsem internet) igra pomembno vlogo z omogočanjem poceni, hitrega in učinkovitega prenosa informacij. Uporaba informacij pa je odvisna od ustreznih in učinkovitih poslovnih procesov, kot je prikazano v nadaljevanju tega poglavja.

2.3.1 Modeliranje poslovnih procesov

Predpogoj za uspešno prenovu poslovnih procesov v oskrbovalni verigi je, da so ključni poslovni procesi v vseh podjetjih v verigi dobro poznani in razumljeni. To je še toliko pomembnejše, ker lahko pomanjkanje razumevanja teh procesov po celi verigi povzroči motnje in nihanja tako v povpraševanju kot v ponudbi. »Vidnost« procesov in povpraševanja kupcev je predpogoj za sinhronizacijo oskrbovalne verige (Holweg, Bicheno, 2002). Osredotočenost na SCM pa ni možna, če podjetje prej ne identificira in izboljša svojih notranjih procesov (Motwani et al., 1998).

Modeliranje procesov lahko tako opredelimo kot uporabo modelov za razumevanje in spremembo poslovanja skupaj z IT. Pomembnejše so poslovne potrebe in posledice informacijskih sistemov kot pa sama analiza informacijskih tokov. Informacijska podpora postane integriran del operacij in v mnogih primerih ključen del poslanstva podjetja (Nilsson, 2005). Modeliranje procesov lahko uporabimo v različnih fazah tovrstnih projektov: za olajšanje interakcije z informacijskim sistemom, za analizo obstoječih procesov in pripravo predlogov izboljšav ter za podporo avtomatski ali polavtomatski izvedbi procesov (Pernici, Weske, 2006). V tem poglavju modeliranje uporabljam predvsem za analizo in pripravo predlogov izboljšav.

Orodje za modeliranje procesov mora omogočati prikaz povezav med aktivnosti in razstavljanje procesov na aktivnosti. Omogočati mora izvedbo kaj-če (angl. what-if) analiz ter identificiranje in označevanje nepotrebnih aktivnosti, stroškov in učinkovitosti procesov (predvsem analizo ozkih grl). Ravno tako mora omogočiti razvoj »kot-je« (angl. AS-IS)⁶ in »kot-bo« (angl. TO-BE)⁷ modelov. Novi modeli morajo biti pred vpeljevanjem potrjeni in testirani. Uporabljajo se

⁵ Razrez praviloma uvrščamo med notranje procese v podjetju.

⁶ Gre za trenutno stanje v podjetju oziroma oskrbovalni verigi.

⁷ Gre za želeno stanje – torej ciljno stanje po prenovi poslovnih procesov.

lahko za napovedovanje značilnosti, ki jih ne moremo direktno meriti ter za predvidevanje poslovnih kazalcev uspešnosti – pridobivanje teh ocen bi lahko bilo drugače oteženo ali predrago.

Za modeliranje poslovnih procesov z namenom boljšega razumevanja in možnih scenarijev za izboljšavo procesov, obstajajo številne metodologije in tehnike (Ould, 1995). Nekatere od bolj uporabljenih v praksi so denimo IDEF0, IDEF3, Petrijeve mreže (angl. Petri nets), sistemska dinamika (angl. System Dynamics), diskretne simulacije (angl. Discrete-Event Simulation) (Eatock et al., 2000). Glede na to da v zadnjih letih modeliranje poslovnih procesov pridobiva na pomembnosti in popularnosti v praksi, se število različnih metod in orodij, ki te metode podpirajo, hitro povečuje (Hommes, Van Reijswoud, 2000). Že več kot 50 orodij poleg modeliranja podpira tudi preverjanje modelov s simulacijami (Hommes, 2001). Vse to seveda otežuje izbiro ustreznega orodja za modeliranje. Empirični pregled obstoječih metodologij, orodij in tehnik za spreminjanje poslovnih procesov je v (Kettinger et al., 1997). Podoben pregled s kriteriji za izbiro ustreznega orodja za posamezno praktično situacijo je tudi v (Kirikova, Makna, 2005).

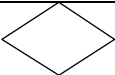
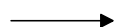
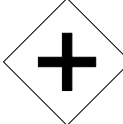
2.3.2 Procesne sheme

V tej disertaciji uporabljam za modeliranje poslovnih procesov tehniko procesnih shem (angl. process maps). Procesne sheme se pogosto uporabljajo v organizacijah, predvsem za modeliranje in analizo poslovnih procesov. Predstavljajo standardno metodo za modeliranje in analizo pri prenovi organizacije ter omogočajo uporabo simulacij. Ena glavnih prednosti te tehnike je, da so zaposleni že po krajšem usposabljanju sposobni kreiranja in ocenjevanja modelov poslovnih procesov (Chen, 1999).

Nadaljnja pomembna prednost je, da pomaga pri identifikaciji prehodov med posameznimi oddelki v podjetju (ali različnimi podjetji) in tako pokaže, kateri oddelek znotraj podjetja je odgovoren za posamezno aktivnost. S tem omilimo osnovno težavo pri uporabi modelov procesov za ocenjevanje ustreznosti projektov – namreč čas, napor in stroške, ki so potrebni za izdelavo takih modelov (Dutta, Roy, 2004). Procesne sheme so torej primerne za namene, za katere jih uporabljam v tej disertaciji; predstavljene koncepte pa bi lahko ilustriral tudi s kako drugo tehniko.

Tehnika procesnih shem je hkrati tudi v pomoč pri posredovanju informacij o aktivnostih, ki se dogajajo med izvajanjem procesov – tako denimo vidimo, katera skupina ljudi ali organizacija opravlja posamezno aktivnost v procesu. Posamezni elementi so povezani s povezavami, ki opišejo tok procesa. Na sliki 2 so prikazani posamezni modelni elementi te tehnike.

Slika 2: Osnovni elementi tehnike procesnih shem

	Simbol	Predstavlja	Primeri
1		Začetek/Konec	Prejem zahtevka Prihod stranke
2		Aktivnost	Priprava načrta razreza Izdaja računa
3		Odločitev	Odobritev/zavrnitev
4		Čakanje	Čakanje na dostavo materiala
5		Podproces	Razrez materiala
6		Oddelek	Oddelek računovodstva Proizvodnja
7		Tok procesa	
8		vzporedno izvajanje (angl. AND)	Hkratno izvajanje dveh ali več aktivnosti (denimo pošiljanje blaga in knjiženje posla)

Vir: povzeto in dopolnjeno po (Popovič et al., 2006)

Omenjeni simboli so uporabljeni na vseh slikah poslovnih procesov v tem poglavju. V zadnjem času se kot industrijski standard uveljavlja tehnika BPMN (angl. business process management notation), ki uvaja nekatere nove ali delno spremenjene simbole (White, 2004), vendar je v vsebinskem smislu praktično enaka pristopu, uporabljenem v tem poglavju doktorske disertacije.

Koncepte uporabe managementa oskrbovalne verige za zmanjševanje stroškov in povečevanje uspešnosti, bi lahko prikazal z uporabo različnih orodij in tudi z različnimi tehnikami za modeliranje. V disertaciji je uporabljeno orodje iGrafx Process. Izbira konkretnega orodja niti ni tako pomembna, ključno je, da ima zgoraj navedene lastnosti. IGrafx Process omogoča vse naštetu in je dovolj preprost za uporabo tudi za nestrokovnjake. iGrafx je integrirano okolje več procesno orientiranih aplikacij, ki omogočajo dokumentiranje, analiziranje in optimizacijo procesov. Ponuja možnosti za analizo, izvajanje, simuliranje in spremljanje procesov (diToro, 2007). Uporabniki in analitiki lahko izvajajo simulacije in kaj-če analize brez poznavanja skriptnih jezikov za simulacije (iGrafx, 2007). iGrafx je eno od najpogostejše uporabljenih orodij za izvajanje analize in prenove poslovnih procesov (Davies et al., 2006). Primeren je tako za statično (izdelavo procesnih shem) kot dinamično (simulacije procesov) modeliranje (Popovič, 2005).

Model procesa je v tem orodju pripravljen kot zaporedje aktivnosti, ki so uvrščene v enega ali več oddelkov (torej organizacijske enote, ki opravljajo te aktivnosti). Vsaka aktivnost ima lahko podatke o vhodih, potrebnih virih za izvedbo, nalogah in izhodih. Dodatno lahko vsaki aktivnosti določimo posamezne attribute, kot so tipi in število potrebnih virov, čas trajanja (konstanten ali stohastično razporejen po določeni funkciji) in različne tipe stroškov (urne in nadurne postavke, strošek uporabe in podobno). Delovni urniki za posamezne vire so poljubno nastavljivi, ravno tako generatorji za kreiranje novih transakcij pri simulacijah. Naštete značilnosti nudijo možnost za podrobno stroškovno in časovno analizo poslovnih procesov (Indihar Štemberger et al., 2004). To vključuje analizo obremenjenosti virov, stroškov izvedbe procesov, pretočnih časov in njihove variabilnosti.

Praktične izkušnje pri uporabi različnih orodij za modeliranje in simulacijo poslovnih procesov (ARIS, Income, iGrafx Process) so pokazale, da so ključne prednosti te tehnike modeliranja visoka raven in intenzivnost komunikacije z zaposlenimi, enostavnost in razumljivost. Te prednosti so še toliko pomembnejše pri modeliranju in prenovi poslovnih procesov v celi verigi, saj je pomembno, da vsi vpleteni iz vseh podjetij dobro razumejo celoten obravnavani poslovni proces (Indihar Štemberger et al., 2004).

Procesne sheme, ki jih uporablja orodje iGrafx Process, imajo grafični uporabniški vmesnik za modeliranje realnega sistema, ki ne zahteva predznanja programiranja. Tudi ljudje brez znanja ali praktičnih izkušenj s področja modeliranja lahko enostavno razumejo in uporabljajo omenjeno tehniko (Bosilj-Vukšić et al., 2002).

Predpogoj za uspešen management medorganizacijskih procesov je nedvomno ustrezna ureditev lastnih procesov in sistemov (Feller, 2000). V tem poglavju disertacije pa se predvsem posvetim možnostim za uporabo modeliranja poslovnih procesov za boljšo pojasnitev in uvedbo medorganizacijske integracije poslovnih procesov (več o prenovi procesov znotraj enega podjetja je denimo v (Bosilj-Vukšić et al., 2002)).

2.4 Študija primera preskrbe in razreza materiala

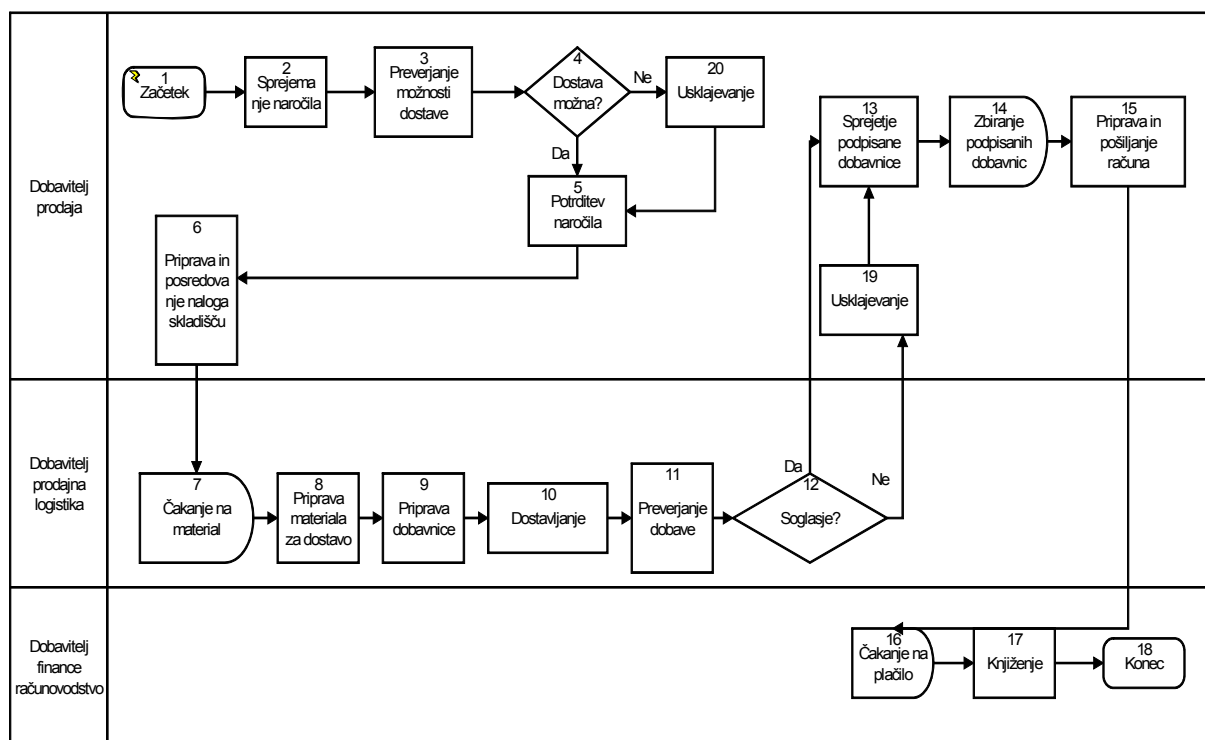
Osnovni koncept preнове poslovnih procesov, s posebnim poudarkom na možnostih za problem razreza, lahko prikažem z naslednjo študijo primera. Študija primera je poenostavljen prikaz odvijanja procesa razreza, prirejen po (Trkman et al., 2005a); (Trkman et al., 2005b). Metodološki pristop avtorja doktorata je bil uspešno uporabljen za modeliranje in analizo koristi preнове procesa razreza v velikem slovenskem podjetju (Erjavec et al., 2008) ter tudi v francoski letalski industriji (Carver et al., 2007).

V tej študiji obravnavamo dve podjetji: proizvajalca/naročnika materiala (denimo žago ali drugo industrijsko podjetje, ki skrbi za razrez) in dobavitelja (ki dobavlja material). Procesi za zagotavljanje materiala so namerno nekoliko poenostavljeni, kar omogoča poudarek najpomembnejših vidikov pri upravljanju verige (po potrebi se lahko uporabi tudi natančnejše

modeliranje, primer le-tega je denimo v Bosilj-Vukšić et al., 2002). Preučevanje alternativnih načrtov za oskrbovalno verigo namreč ne zahteva tako natančnega planiranja kot denimo optimizacija proizvodnih procesov (Persson, Olhager, 2002). Prav zato v nadaljevanju pri optimizaciji razreza uporabim bolj podroben pristop k preučevanju.

Predstavljeni modeli se brez težav razširijo z vključitvijo dodatnih podjetij ali procesov in se nato analizirajo z enako metodologijo (glej denimo Trkman et al., 2007 ali Rajagopal, Rajagopal, 2008).

Slika 3: »Kot-je« model dobavitelja



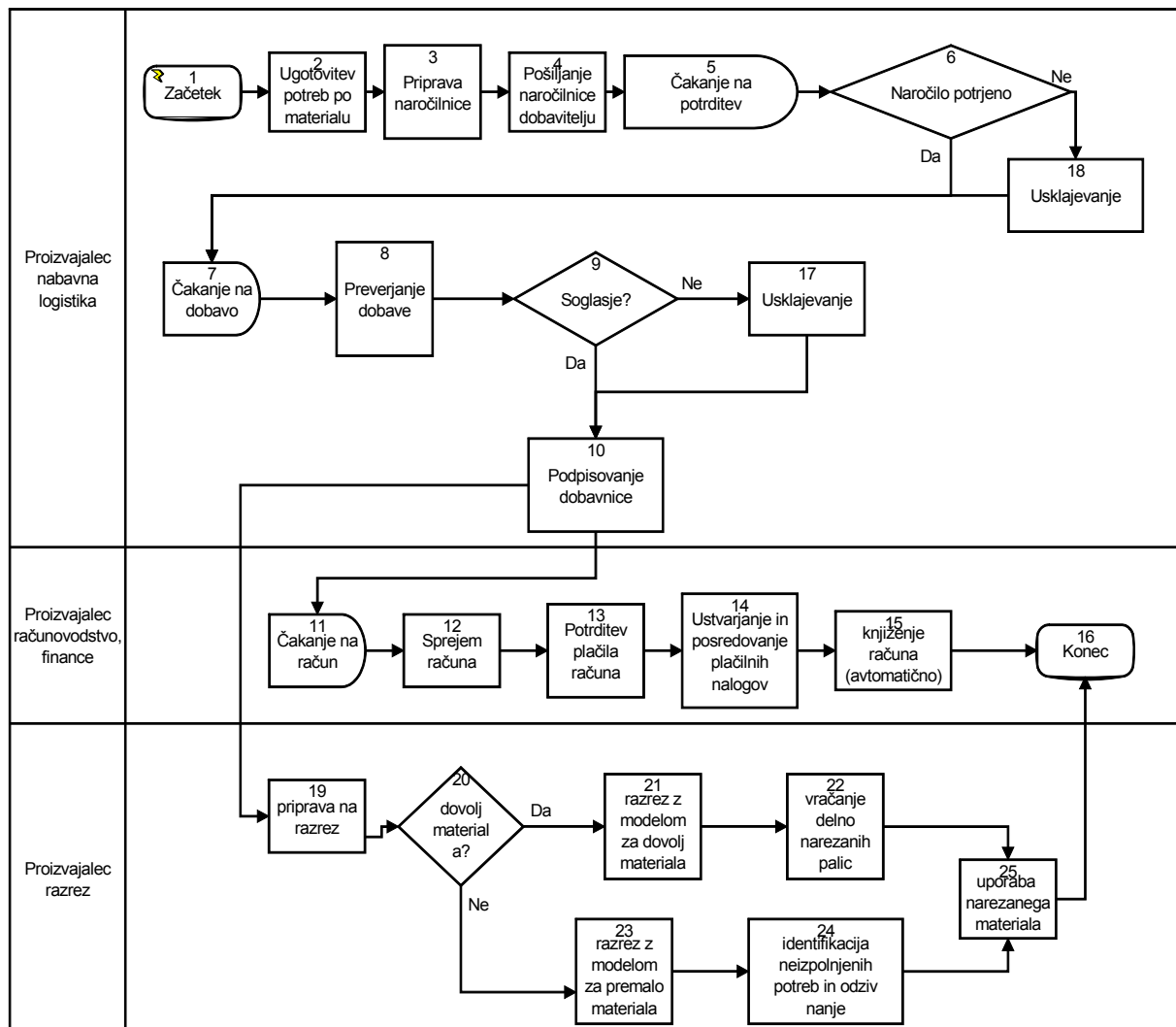
Vir: prirejeno po (Trkman et al., 2005a); (Trkman et al., 2005b).

Slika 3 tudi jasno kaže, kako je razrez vključen v proces zagotavljanja materiala. V modelih procesa so namreč neposredno s samim razrezom povezane samo aktivnosti 19-24 v modelu proizvajalca. Za uspešno izvajanje teh aktivnosti so ključne v naslednjih poglavjih predstavljene metode; za uspešno izvajanje celotnega procesa pa upoštevanje spoznanj tega poglavja. Za doseganje ciljev podjetja in verige je torej pomembno oboje.

Kot je razvidno s slik 3 in 4, je trenutno »kot-je« model sestavljen iz dveh ločenih procesov – procesa zagotavljanja materiala pri proizvajalcu in procesa izpolnjevanja naročila pri dobavitelju. Čeprav sta oba procesa seveda medsebojno povezana (denimo aktivnost št. 12 – sprejem računa - v modelu proizvajalca je neposredna posledica aktivnosti št. 15 – priprava in pošiljanje računa - v modelu dobavitelja) in čeprav obstaja delno izmenjevanje informacij med obema podjetjema oziroma obema procesoma, je sedanje stanje iz spodaj navedenih razlogov nezadovoljivo. To

seveda lahko vodi do večjih skupnih stroškov zagotavljanja materiala in tudi razreza samega. Kot je prikazano v (Erjavec et al., 2008) se lahko razliko med sedanjim stanjem in želenim, prenovljenim izvajanjem procesov oceni z uporabo simulacij poslovnih procesov.

Slika 4: »Kot-je« model proizvajalca



Vir: prirejeno po (Trkman et al., 2005a); (Trkman et al., 2005b).

Nekateri izmed opaženih problemov so denimo:

- v procesih je možnih precej nepotrebnih zamud in zaostankov: tipičen, še zdaleč pa ne edini, primer je dejstvo, da se lahko proces razreza in dobave materiala pri dobavitelju začne šele po koncu četrte aktivnosti pri proizvajalcu,
- ključne informacije niso pravočasno na voljo. Posledica tega so številna dodatna čakanja in nepotrebne aktivnosti. Proizvajalec mora tako na primer čakati na potrditev dobavitelja, ali so izdelki v želeni količini in dolžini na voljo. To je posledica omejenih informacij (proizvajalec ne pozna trenutnih možnosti dobave in ravni zaloga)

pri dobavitelju) in lahko vodi do številnih zaostankov. To še posebej velja, če dobavitelj ne more pravočasno dostaviti zelenega materiala in je potreben popravek naročila. Popravki ali kasnejša dobava materiala pa seveda vodijo do manjše ravni zaloge od želene pri proizvajalcu⁸ in pogosto do tega, da proizvajalec ne more pravočasno razrezati potrebnih količin (rešujemo torej v nadaljevanju opredeljen problem s premalo materiala). To pa vodi do stroškov neizpolnitve naročil, opisanih v sedmem poglavju,

- hiter in fleksibilen odziv na morebitne spremembe v povpraševanju končnega kupca ni možen ali pa je zelo drag, ker ni možno hitro spreminjati zelene količine (zaradi dolžine in nepreglednosti procesov). To lahko pomeni večje stroške skladiščenja, razreza in večje potencialne oportunitetne stroške. Fleksibilnost je zelo pomembna in jo velika večina vodij proizvodnje navaja kot ključno strategijo za doseganje konkurenčne prednosti (Abdel-Malek et al., 2000),
- posamezno podjetje nima možnosti za vpliv ali izboljšavo procesov drugega podjetja, čeprav sta nedvomno medsebojno odvisni,
- težko merimo stroške in celotne čase v celotni oskrbovalni verigi, ampak se moramo ukvarjati predvsem s stroški v posameznem podjetju (torej je možna samo lokalna optimizacija, ki pa ne vodi nujno do globalnega optimuma).

Običajen odgovor na zgoraj opisane probleme je povečanje ravni zalog, ki pa vodi do dobro znanih problemov in dodatnega porasta stroškov.

Z vidika razreza taka postavitev procesov pomeni, da sicer lahko uporabljamo v nadaljevanju disertacije predstavljene metode za optimizacijo razreza v danem trenutku. Zaradi razmeroma visoke ravni zalog, ki praviloma znižuje izgube, lahko pričakujemo tudi načrte razreza z majhno izgubo. Vendar zaradi zamud in nezanesljivosti prihodnjih dobav nastajajo višji stroški naročanja in zalog; ravno tako je težko uporabljati metode kontinuiranega razreza. Cilj integracije pa je torej, da dobre načrte razreza zagotovimo s pretokom informacij – v bistvu torej zamenjamo zaloge z informacijami (Daugherty et al., 1999).

Prikazane modele lahko uvrstim še po hierarhiji procesne zrelosti (Lockamy, McCormack, 2004), ki vključuje 5 nivojev:

- trenuten (angl. ad-hoc): procesi so nestrukturirani in slabo definirani, merjenje procesne uspešnosti ni uvedeno,
- opredeljen (angl. defined): osnovni procesi so dokumentirani, vendar je organizacija in delo zaposlenih še vedno klasično,

⁸ Metodologija za izbor primerne ravni zalog za minimizacijo celotnih stroškov (vsota stroškov zalog, odpadka in neizpolnjenih naročil) pri izvajanju razreza je podrobneje predstavljena v (Erjavec et al., 2008a)

- povezan (angl. linked): uvede se management procesov s strateškimi cilji. Sodelovanje med zaposlenimi in podjetji je predvsem na procesnem nivoju,
- integriran (angl. integrated): celotna koordinacija, organizacijske strukture in delovna mesta so postavljene na procesnem nivoju. Kazalniki merjenja uspešnosti so vključeni na procesnem nivoju za celotno oskrbovalno verigo,
- razširjen (angl. extended): konkurenčnost temelji na mreži več podjetij. Medsebojno zaupanje omogoča sproten prenos odgovornosti brez pravnih dogovorov.

»Kot-je« model obeh podjetij lahko uvrstimo na drugi nivo, saj so vsi procesi sicer že dokumentirani (kot je prikazano na predhodnih slikah), vendar ne obstaja integracija ali pravo posredovanje informacij med obema podjetjema.

Prenovljeni model (na sliki 5) pa je na četrti stopnji (integriran proces), saj obe podjetji sedaj dejansko sodelujeta na procesnem nivoju. Prehod na peti nivo bo verjetno možen šele sčasoma, saj je potreben pogoj za to globoko medsebojno zaupanje. Zaupanje se lahko delno poveča z investicijami v sredstva/informacijski sistem, ki so specifična za posamezno verigo. Čeprav je medorganizacijsko zaupanje pomembno področje in potencialen problem pri uspešnem sodelovanju podjetij, bi natančnejši pregled presegel okvire te disertacije. Temeljit pregled obstoječe literature in nadaljnja spoznanja o vplivu medsebojnega zaupanja na trajanje ciklov in učinkovitost verige je denimo v (Handfield, Bechtel, 2002).

Potreba po ustrezni prenovi poslovnih procesov za doseganje večje uspešnosti in učinkovitosti je torej jasno razvidna. Prenovljeni poslovni procesi so tesno podprti z ustrezno uporabo IT, kot je prikazano v nadaljevanju.

Kot-bo integrirani model, predstavljen na sliki 5, kaže integracijo obeh poslovnih procesov v en proces, ki sega preko različnih oddelkov obeh podjetij. Osnovne spremembe, ki omogočajo to integracijo, so:

- sklenejo se dolgoročne pogodbe in povezave na podlagi možnosti e-poslovanja med dobaviteljem in proizvajalcem – namera za dolgoročno partnerstvo je nedvomno predpogoj za zavezo k investicijam v prenovi in integracijo poslovnih procesov ter nove rešitve, saj so lahko te investicije precej drage,
- za podporo celotnega procesa (od ugotovitve potrebe do razreza materiala) se uvede medorganizacijski informacijski sistem. Sistem je na voljo vsem vpletenim oddelkom pri proizvajalcu in dobavitelju,

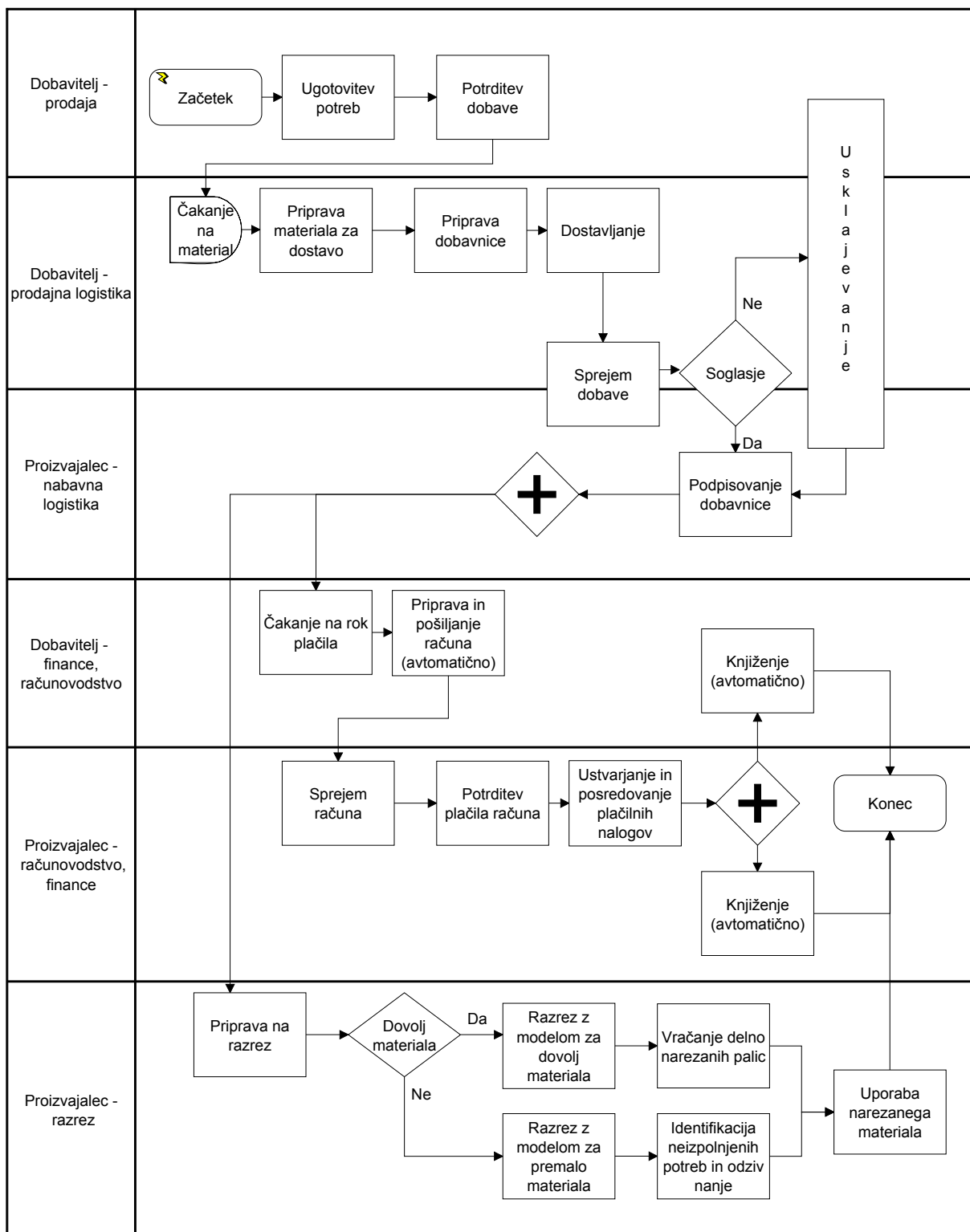
- uvede se upravljanje zalog s strani dobavitelja⁹. Dobavitelj ima sedaj popolne informacije o stanju zalog pri proizvajalcu in tudi najboljše možne informacije o prihodnjih potrebah¹⁰ in je zadolžen za pravočasno ugotavljanje potreb in dostavo zahtevanega materiala. Ravno tako pa dobavitelj tudi takoj reagira v primeru, da za izvedbo razreza ni bilo dovolj materiala (z novim pristopom se sicer ta verjetnost precej zmanjša) in po potrebi ponovno sproži proces dobavljanja. Na ta način se zaradi hitrega odziva zmanjšajo stroški neizpolnitve naročila ob pomanjkanja materiala,
- posledično je tudi začetna točka v procesu drugačna in je v podjetju dobavitelja. Dobavitelj ugotovi potrebo po materialu in začne opisani proces, ki je potreben za zadovoljitev te potrebe – tako je seveda pretočni čas, potreben za dokončanje procesa, bistveno krajši, z vsemi s tem povezanimi prednostmi,
- integracija procesov tudi omogoča dobavitelju boljše upravljanje lastnih procesov, izogibanje ali vsaj zmanjševanje nevarnosti ozkih grl in zmanjševanje varnostnih zalog, ker so informacije o bodočem povpraševanju prej dostopne. Vse naštetu seveda pomeni precejšnjo korist tudi za dobavitelja. Še več: zmanjševanje varnostne zaloge pri dobavitelju pomeni tudi večjo fleksibilnost v prihodnje – tako bo lahko dobavitelj prilagajal dolžine in število palic v zalogi, kar bo vodilo do nadaljnjega zmanjševanja izgube pri razrezu in s tem povezanih stroškov za proizvajalca.

Navedene spremembe vodijo do bistvene izboljšave procesov s precejšnjimi prednostmi za vsa (obe) vpletena podjetja in povečujejo dodano vrednost za končnega kupca in posledično tudi konkurenčnost verige kot celote. Poleg tega tudi bistveno zmanjšujejo primanjkljaj informacij in omogočajo boljšo koordinacijo. To pa poleg ostalih navedenih prednosti vodi tudi do boljšega načrtovanja razreza in tako da manjših izgub in drugih stroškov, neposredno povezanih z razrezom materiala.

⁹ Podobni pristopi kot omenjeni so tudi skupno upravljanje zalog (angl. co-managed inventory (CMI)), porazdeljeno načrtovanje potreb (angl. distribution requirements planning (DRP)), neprestano/učinkovito načrtovanje materiala (angl. continuous/efficient replenishment planning (CRP/ERP)), sočasen odziv na povpraševanje kupca (angl. synchronized consumer response (SCR)) – čeprav so med temi pristopi določene razlike, pa so osnovni koncepti in ideje pri vseh omenjenih pristopih zelo podobni. Kot že omenjeno - za koncepte v tej disertaciji, konkreten uporabljen pristop ni ključen.

¹⁰ Zaradi variabilnosti in negotovosti pri povpraševanju končnih kupcev seveda informacije o prihodnjih potrebah nikoli ne morejo biti povsem natančne, uspešen management medorganizacijskih procesov pa lahko (kot je za primer razreza razvidno tudi iz nekaj primerov na začetku tega poglavja) precej prispeva k zmanjševanju variabilnosti.

Slika 5: Kot-bo integrirani model



Vir: prirejeno po (Trkman et al., 2005a); (Trkman et al., 2005b).

Poleg tega je možnih tudi precej izboljšav na bolj taktičnem in operativnem nivoju, kot so denimo:

- avtomatično izvajanje nekaterih aktivnosti (kot denimo priprava računa, dobavnice oziroma naročilnice), kar dodatno zmanjšuje stroške, potreben čas, obremenjenost zaposlenih in število napak,
- zagotavljanje elektronskega sledenja naročilom, ki dodatno povečuje razpoložljive informacije za vsa podjetja v oskrbovalni verigi – informacije so na razpolago prej in ceneje, možen je tudi hitrejši odgovor na nepredvidene spremembe - denimo, da so potrebe po narezanem materialu večje/manjše od prej predvidenih ali da bomo potrebovali drugačno razporeditev po dolžinah, kot smo pričakovali,
- vpeljava elektronskega sistema za plačevanje računov, ki bistveno zmanjša čas in napor, potreben za poravnavanje računov.

Kot-bo model omogoča krajše čase ciklov in nižje stroške transakcij, poleg tega pa pomeni tudi zmanjšanje ravni zalog (predvsem na račun manjših potrebnih varnostnih zalog) za vsa (obe) podjetja v verigi brez povečanja nevarnosti za pomanjkanje materiala (angl. stock-out)¹¹. Ker smo uspeli izločiti precej aktivnosti iz procesa, številne preostale pa lahko bistveno skrajšamo, se precej zmanjša količina dela za zaposlene, tako da se lahko bolj osredotočijo na bolj strateška opravila ter opravila z večjo dodano vrednostjo.

Poleg teh prednosti na taktičnem/operativnem nivoju pa lahko uresničimo tudi nekatere strateške prednosti. Hitrejša prepoznavanje in odziv na dolgoročne spremembe v vzorcih povpraševanja (tako glede skupne količine kot razporeditve dolžin naročil oziroma potreb), izboljšan nivo storitev za kupce, boljši in hitrejši odziv na nepričakovane dogodke in tudi vpeljava novih proizvodov ali storitev so bistveno enostavnejši ob prenovljenem modelu poslovnih procesov.

V tem poglavju sem torej pokazal, kako lahko deljenje informacij, ki jih omogočajo storitve e-poslovanja ter ustrezno prenovljeni procesi, izboljšuje poslovne procese in posledično uspešnost tako posameznega podjetja kot tudi oskrbovalne verige kot celote. Modeliranje poslovnih procesov je primerna tehnika tako za analiziranje sedanjega stanja kot za načrtovanje prihodnjega razvoja.

Omenjene ugotovitve sem prikazal na primeru študije primera oskrbovalne verige, sestavljene iz dveh podjetij (dobavitelja materiala ter proizvajalca, ki izvaja razrez). Prenovljeni kot-bo model omogoča hitrejšo, bolj učinkovito izvedbo enega poslovnega procesa (torej nabave in razreza materiala), ki je ključen za uspešnost obeh podjetij.

¹¹ Za primer uporabe modeliranja in simulacij poslovnih procesov za ocenjevanje skupnih stroškov zalog (tako držanja kot primanjkljaja) glej denimo (Trkman et al., 2007) ali (Erjavec et al., 2008a).

Ne smemo pa pozabiti, da tudi odlično zastavljen kot-bo model sam po sebi ni zadnji korak pri razvoju oskrbovalne verige. Tudi po uspešni uvedbi takega modela morajo biti vsa podjetja neprestano pozorna in proaktivno reagirati na spremembe v poslovnem okolju z neprestanimi izboljšavami. Vse metode managementa poslovnih procesov namreč poudarjajo neprestano izboljševanje procesov (Lee, Dale, 1998).

Ob koncu tega poglavja strnjeno povzemam ključne probleme in zahteve pri razrezu, ki izhajajo iz zgoraj napisanega, in s katerimi se bom ukvarjal v nadaljevanju disertacije:

- poznavanje razreza in metod za optimizacijo v različnih praktičnih situacijah v oskrbovalni verigi – ker so zahteve in omejitve od primera do primera zelo različne, je bilo razvitih veliko število različnih metod za optimizacijo različic problema razreza. Te metode lahko prispevajo k boljšemu izvajanju posameznih aktivnosti v procesu, zato njihov obširen pregled podajam v tretjem poglavju,
- povečevanje konkurenčnosti oskrbovalne verige z razvojem boljših metod za razrez, ki bodo vodile do manjših izgub materiala in nižjih stroškov razreza (ustrezno metodo za manjše probleme predstavljam v poglavju 5.1., za večje pa v poglavju 5.3.),
- v kolikor za isti problem obstaja več metod za reševanje, vendar zaradi potrebe po hitri rešitvi ni mogoče uporabiti vseh, lahko izbira prave metode prispeva k nižji izgubi materiala (tak pristop predstavljam v poglavju 5.2.),
- v primeru, da zaradi težav pri preskrbi v oskrbovalni verigi ali nihanj v povpraševanju, razpoložljiv material ne zadošča za pokritje vseh potreb, je pomembno vprašanje, katera naročila naj ostanejo nenarezana in s kakšnim načrtom razreza naj se narežejo preostala (metodo ob upoštevanju stroškov neizpolnitve predstavljam v 6. poglavju),
- pri neprestanem izpolnjevanju zahtev kupca (zaradi uporabe enega od zgoraj predstavljenih pristopov, denimo upravljanje zalog s stran dobavitelja (angl. vendor-managed inventory - VMI) je potrebna metoda, ki bo dobavitelju omogočala kontinuirano izpolnjevanje zahtev po narezanem materialu ob hkratnem ohranjanju nizkih stroškov (tak pristop predstavljam v 7. poglavju).

Ključno nadaljnje raziskovalno delo je potrebno pri prilagoditvi in uporabi predstavljene metodologije za merjenje in primerjavo stroškovnih prihrankov pri prenovi procesa razreza. Prvi rezultati (Erjavec et al., 2008) kažejo, da tovrstni pristop omogoča primerjavo in analizo stroškov procesa razreza, razdeljenih na stroške opreme in stroške dela, ter stroškov same izgube materiala. Ti rezultati v povezavi s teoretično utemeljitvijo v tem poglavju torej dokazujejo, da prenova poslovnih procesov v podjetju lahko prispeva k boljšemu odvijanju podprocesa razreza.

Nekatera raziskovalna vprašanja, ki so potrebna nadaljnje znanstvene obravnave, predstavljam v zaključku disertacije.

3 Predstavitev metod za razrez

3.1 Opredelitev razreza¹²

Po širši predstavitvi vloge razreza v okviru poslovanja podjetja in predvsem oskrbovalne verige v tem poglavju najprej predstavljam obstoječe metode za razrez. Kot že omenjeno, lahko njihova uporaba pomembno zmanjša stroške pri preskrbi z materialom.

Problem razreza je bil v taki obliki prvič formuliran pred skoraj natančno 50 leti (Paull, 1956) in je v vsem tem obdobju, predvsem pa v zadnjih letih, pritegnil veliko pozornost raziskovalcev, ki so se ukvarjali z različnimi oblikami tega problema ter različnimi pristopi k reševanju. Nekatere od teh oblik in pristopov predstavljam v nadaljevanju tega poglavja disertacije.

Tak problem se pojavlja v različnih panogah, kot so na primer papirna, tekstilna, kovinska, električna, lesarska itd. Osnovni problem je razrez materiala na zalogi v željeno količino in dolžino naročil, pri čemer je potrebno minimizirati izgubo, ki nastane pri tem razrezu. Na razpolago imamo torej material določene dolžine in širine. Razrez lahko opravimo na več različnih strojih.

Nože na vsakem stroju je možno nastaviti na poljubno kombinacijo dolžin naročil, tako da skupna dolžina naročil ne presega dolžine palice. Naročila so določena kot število kosov posamezne dolžine ali kot skupna dolžina, ki jo rabimo za posamezno naročilo. Običajno se izgubi ne da povsem izogniti. Osnovni problem je torej v tem, kako naročila razporediti na posamezne kose materiala in/ali stroje tako, da bo izguba materiala čim manjša (Gass, 1985).

V praksi je seveda veliko različic tega problema, saj se razrez pojavlja v številnih industrijskih panogah – probleme iz nekaterih od teh panog tudi predstavljam v nadaljevanju doktorske disertacije. Za material na zalogi zaradi poenostavitve uporabljam v celotni disertaciji izraz »palice«¹³, ker se material pogosto pojavlja v obliki palic. Seveda pa je lahko v posamezni praktični situaciji material tudi v obliki rol, zvitkov, navitkov, tuljav, kartonov, plošč in podobno.

V tej doktorski disertaciji se sicer ukvarjam s problemom enodimenzionalnega razreza (angl. one-dimensional cutting stock problem), vendar je potrebno poudariti, da ima poleg problemov razreza tudi veliko problemov v matematiki, operacijskih raziskavah, logistiki, proizvodnji in ekonomiji podobno logično strukturo. Dyckhoff (1990) tako navaja naslednje probleme s podobno strukturo, ki se tako lahko rešujejo z istimi ali vsaj zelo podobnimi metodami:

- problemi razreza in problemi zmanjševanja odpadka,
- problemi pakiranja (kako določene predmete tako razporediti v določen prostor, da bo kar najmanjši del prostora ostal neizkoriščen),

¹² Poglavje je razširjeno, dopolnjeno in prilagojeno po (Trkman, Gradišar, 2005c); (Trkman, 2002).

¹³ V angleščini se ravno tako uporabljajo različni izrazi, najpogostejši pa je stock length.

- problemi nakladanja npr. na palete, v kontejnerje ali na vozila,
- problemi razporeditve, sortiranja, razvrščanja,
- določanje urnika pri večprocesorskih računalnikih, razporejanje kapitala, dodeljevanje pomnilnika.

Kot že omenjeno, so ti problemi logično podobni in se lahko rešujejo s podobnimi metodami. Med različnimi tipi so nekatere vsebinske razlike – tako denimo (Vahrenkamp, 1996) navaja, da pri problemih razreza običajno potrebujemo več kosov posamezne dolžine naročila, medtem ko je to število pri problemih pakiranja manjše (običajno kar 1). Ravno tako je običajno cilj problemov razreza minimiziranje izgube materiala, pri problemih pakiranja pa običajno minimiziramo število uporabljenih posod/kontejnerjev oziroma druge embalaže, ki jo uporabljamo za pakiranje materiala. Kljub tem razlikam pa so metode za različne probleme medsebojno dovolj podobne, da jih navajam skupaj, pri čemer se posebej osredotočim na probleme razreza.

Za razporeditev posameznih problemov v kategorije je zelo pomembna Dyckhoffova klasifikacija (Dyckhoff, 1990), ki so jo pozneje razširili Gradišar et al. (1999b; 2002). Ta klasifikacija vsak problem razdeli po 4 kriterijih (v oklepaju je originalna angleška oznaka za posamezno kategorijo):

1. število dimenzij (gre za verjetno najpomembnejši kriterij za delitev problemov na različne kategorije) :
 - a. ena dimenzija (1),
 - b. dve dimenziji (2),
 - c. tri dimenzije (3),
 - d. n dimenzij, kjer je $n > 3$ (N): 4-dimenzionalni problem pride v poštev, če vključimo še čas – denimo, da morajo biti 3-dimenzionalne škatle spravljene v zabojniku za določeno časovno obdobje.
2. Tip problema
 - a. vsi veliki predmeti in izbor manjših predmetov (B): gre za t. i. probleme nahrbtnika (angl. knapsack problem), kjer moramo v enega ali več večjih predmetov spraviti izbor manjših predmetov – izbor izvedemo glede na relativno pomembnost posameznega predmeta,
 - b. izbor velikih predmetov in vsi manjši (V): tak primer je običajen pri problemih razreza in pakiranja – vse manjše predmete moramo razporediti v izbor večjih predmetov, tako da dosežemo čim manjšo izgubo ali dosežemo kakšen drug cilj.

3. Vrsta velikih predmetov

- a. en predmet (O): imamo samo en velik predmet,
- b. več identičnih predmetov (I): imamo več velikih predmetov, ki pa so vsi enaki,
- c. nekaj skupin identičnih predmetov (G) – to je omenjena razširitev v (Gradišar et al., 2002; Gradišar et al., 1999b) – imamo več velikih predmetov, ki so v nekaj standardnih velikostih,
- d. različni predmeti (D) – vsi veliki predmeti so medsebojno različni.

4. Vrsta majhnih predmetov

- a. nekaj predmetov (različnih oblik) (F) – imamo razmeroma majhno število medsebojno različnih predmetov,
- b. veliko predmetov različnih oblik (M) – imamo večje število medsebojno različnih predmetov,
- c. veliko predmetov, ki niso enake oblike (R) – npr. pakiranje 1000 predmetov velikosti med 0 in 1 v zabojnike velikosti 1,
- d. predmeti ene oblike (C): npr. nakladanje palet z enakimi kosi.

Poleg te klasifikacije se ponekod (npr. Hinxman, 1980) omenja še t. i. 1 ½ dimenzionalne probleme, kjer lahko pri vsakem razrezu ali pakiranju uporabimo samo določeno število osnovnih načrtov. Ravno tako bi lahko med take probleme uvrstili metode, ki se sicer ukvarjajo z 2-dimenzionalnim razrezom, vendar pri tem (običajno zaradi zahtev ali omejitev v proizvodnji) lahko upoštevajo samo manjše število standardnih dvodimenzionalnih oblik. Takšna je recimo metoda za razrez pri proizvodnji električnih generatorjev (Cui, 2005; Cui, 2005a).

V zadnjem času pa je bilo opozorjeno na še več pomembnih pomanjkljivosti omenjene klasifikacije, kot so predvsem (povzeto po Wäscher et al., 2007):

- vseh problemov razreza in pakiranja ne moremo razporediti v Dyckhoffovo klasifikacijo;
- njegova klasifikacija je nekonsistentna, isti problem lahko razvrstimo v več podrazredov;
- posamezni podrazred ne vsebuje nujno homogenih problemov: glej denimo (Gradišar et al., 2002) za podrobnejšo obrazložitev tega problema.

Zato v novi klasifikaciji predlagajo delitev problemov razreza po naslednjih kriterijih:

- *število dimenzij*: ena, dve ali tri; problemi z več kot tremi dimenzijami se štejejo kot različica tri-dimenzionalnega problema,

- *tip naloge*: maksimiranje outputa ali minimiranje inputa/vrednosti, pri čemer pa se tako input kot output oz. njuna vrednost obravnava le kot količina in vrednost večjih ali manjših kosov materiala ter njuna morebitna razlika¹⁴.
- *razporeditev dolžin naročil*:
 - o *enake dolžine naročil*,
 - o *šibko heterogena razporeditev dolžine naročil*: nekaj različnih dolžin naročil, pri čemer je praviloma povpraševanje po več kosih vsake dolžine,
 - o *močno heterogena razporeditev dolžine naročil*: praviloma povsem različne dolžine naročil; povpraševanje po posamezni dolžini naročila je torej 1,
- *razporeditev dolžine objektov na zalogi*:
 - o *en velik objekt*: ki ima lahko vse dimenzije fiksne ali pa je ena ali več od dimenzij spremenljivih,
 - o *več velikih objektov*: analogno kot pri razporeditvi dolžin naročil tudi tu avtorji ločijo enake velikosti objektov ter šibko in močno heterogeno razporeditev objektov.

Taka klasifikacija sedaj praviloma omogoča razporeditev vseh verzij problema razreza v ustrezni razred. V sedmem poglavju disertacije pokažem na pomembno pomanjkljivost tako Dykchoffove kot Wäscherjeve klasifikacije, ki je posledica naraščanja pomena obravnavanja razreza kot enega od procesov v podjetju, in predlagam ustrezno razširitev obeh klasifikacij.

Velika večina problemov razreza v obeh klasifikacijah je NP polna (angl. NP-complete), iz česar sledi, da večjih problemov običajno ni možno rešiti eksaktno. Probleme v operacijskih raziskavah lahko namreč v splošnem razvrstimo v dve vrsti:

- problemi, za katere obstajajo algoritmi, ki jih rešijo v polinomskem času, torej v $O(n^r)$ zahtevnosti, kjer je r konstanten (Kozak, 1986). Gre torej za probleme, ki jih je možno rešiti v polinomskem času na determinističnem Turingovem stroju¹⁵ (NP Complete, 2002). Ta razred problemov označimo s črko P,
- problemi, za katere smo uspeli najti le rešitve eksponentne zahtevnosti. Ti problemi sodijo v razred NP. Ta oznaka označuje probleme, ki jih lahko rešimo z nedeterminističnim algoritmom v polinomskem času (Kozak, 1986).

Problem je NP težek, če je možno vsak drug problem v NP prevesti v ta problem s polinomskim algoritmom. Problem je NP poln, če je v razredu NP ter je NP težek. Če bi torej našli učinkovit (polinomski) algoritem za rešitev enega NP polnega problema, bi to pomenilo, da lahko s

¹⁴ Interdisciplinarnega pristopa ter povezave s širšim obravnavanjem razreza avtorji pri obravnavanju ciljev ne predlagajo.

¹⁵ Predstavitev Turingovega stroja in njegovega delovanja je denimo v (Gradišar, 2003).

polinomskim algoritmom rešimo tudi vse ostale NP polne probleme (Schrijver, 1986, str. 21). Do sedaj tak algoritem še ni bil najden, splošno (sicer nedokazano) mnenje je, da ne obstaja (NP Complete, 2002; Cook, 2000).

Razlika med polinomskimi in eksponentnimi algoritmi je razvidna iz tabele 2, kjer je prikazana sprememba v velikosti problema, ki ga lahko rešimo ob uporabi hitrejšega računalnika. Za algoritme s polinomsko časovno zahtevnostjo (prve štiri vrstice v tabeli) se ta velikost hitro povečuje, za oba algoritma z eksponentno časovno zahtevnostjo pa bistveno počasneje.

Tabela 2: Vpliv hitrejšega računalnika na polinomske in eksponentne algoritme

Funkcija časovne zahtevnosti	Velikost problema s sedanjim računalnikom	100-krat hitrejši računalnik	1000-krat hitrejši računalnik
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	$N_6 + 4,19$	$N_6 + 6,29$

Vir: (Garey, Johnson, 1979)

Iz te tabele je razvidno, da za večino NP polnih problemov razreza tudi v prihodnosti ne moremo pričakovati eksaktnih rešitev za velike probleme. Kot je predstavljeno v nadaljevanju te disertacije, so eksaktne metode primerne predvsem za reševanje majhnih problemov, pri večjih problemih pa si pomagamo z različnimi hevrističnimi metodami, ki ob razmeroma nizki časovni in računski zahtevnosti vodijo do skoraj optimalnih rešitev. Kot kaže metoda, razvita v poglavju 6. 3. te disertacije, pa je možna tudi kombinacija eksaktnih in hevrističnih metod, ki vodi do še boljših rešitev ob bistveno večji časovni zahtevnosti reševanja kot pri hevrističnih metodah (in bistveno nižji kot pri eksaktnih metodah).

3.2 Metode za dvodimenzionalni razrez/pakiranje

V tem poglavju na kratko predstavimo še ostale tipe razreza in metode, ki so se v zadnjih letih uveljavile za njihovo reševanje. V nadaljevanju se posvetimo izključno enodimenzionalnemu razrezu.

Dvodimenzionalno pakiranje (kot že omenjeno, gre pri razrezu za analogen problem) pomeni, da imamo n pravokotnih predmetov z določeno dolžino in širino ter neomejeno količino zabojnikov ravno tako z neko dolžino in širino. Cilj je razporeditev teh predmetov v zabojnike brez prekrivanja, tako da so robovi predmetov vzporedni z robovi zabojnikov – predmetov torej ne moremo obračati. Če imajo vsi predmeti enako širino, gre za enodimenzionalen problem

pakiranja (*IBP*). Tako enodimenzionalno kot dvodimenzionalno pakiranje sta NP težka (Lodi et al., 2002). Različni pristopi za reševanje tega problema so predstavljeni v (Lodi et al., 2002). Zgoraj opisan problem pakiranja je seveda le ena od različic tega problema (in sicer razmeroma enostavna). Druge variante problema, kjer lahko denimo predmete tudi obračamo ali pa predmeti ter zabojniki niso pravih, pravokotnih oblik, so predstavljene v (Lodi et al., 1999). Tudi ti primeri so NP težki.

Ena od predstavljenih rešitev je denimo eksaktna rešitev, ki s kombinacijo uporabe metode razveji in omeji in dinamičnega programiranja eksaktno reši problem, kjer je potrebno serijo pravokotnikov razporediti v pas fiksne širine in neomejene dolžine (Hifi, 1998). Isti avtor je kasneje razvil še nekaj heurističnih algoritmov za reševanje nekoliko težje različice istega problema, pri čemer je glavni cilj maksimizacija doseženega dobička z narezanimi kosi (Hifi, Hallah, 2006)

Z oddelčnim (angl. compartmentalised) problemom nahrbtnika se ukvarjajo v (Hoto et al., 2007), kjer je predstavljena študija primera razreza jeklenih palic. Razrez se izvede v dveh fazah: najprej se pridobijo rezultate prve faze (to so tehnično gledano posamezni oddelki). Polizdelki se nato razrežejo v končne izdelke – jeklene pasove določene debeline. Avtorji dobre rezultate pridobijo s kombinacijo različnih metod, ki temeljijo na klasičnem pristopu Gilmoreja in Gomoryja. Identičen problem je obravnavan v (Marques, Arenales, 2007), kjer avtorja maksimizirata razliko med celotno vrednostjo in stroški posameznih oddelkov. Kot omenjeno, je tovrstni problem pogost pri razrezu v jeklarstvu ali papirni¹⁶ industriji. Za reševanje razvijeta celoštevilski nelinearni optimizacijski model ter ustrezno heuristično metodo za njegovo reševanje. Kot običajno, je analiza metode izvedena s pomočjo numeričnih eksperimentov.

V (Carnieri et al., 1993) je predstavljen heuristični algoritem za razrez stavbnega lesa z napako (en del lesa je neprimeren za uporabo in se ne sme uporabiti v načrtu razreza). Tudi tu je kot glavni problem omenjeno zelo veliko število možnih vzorcev. V (Pisinger, 2002) je več heurističnih algoritmov za reševanje problema pakiranja pravokotnih škatel v zabojnik. Najboljši od teh algoritmov (kakovost algoritmov je bila ugotovljena na podlagi statistične analize) pri večjih problemih doseže do 95% izkoriščenost zabojnika. Podobno Yazgac in Ozdemir (2004) preučujeta problem iz pohištvene industrije, kjer je potrebno uporabiti dvodimenzionalni razrez. Z metodo vzorcev minimizirata izgubo in potrebno število kosov materiala.

Za razliko od teh metod, kjer je glavni cilj minimiziranje izgube, se Bhadury in Chandrasekaran (1996) ukvarjata z minimiziranjem dolžine rezov, ki so potrebni, da izrežemo določen vzorec iz razpoložljivega materiala. S tem namreč dosežemo krajši čas razreza, hkrati pa tudi manjšo obrabo rezila.

¹⁶ Glej denimo (Lin, 2005) za preprosto metodo za optimizacijo razreza v papirni industriji.

Pregled nekaterih novejših metod za druge tipe razreza in pakiranja je tudi v (Wang, Wäscher, 2002), pregled nekaterih metod odkritih pred letom 1990 v (Sweeney, Paternoster, 1992), pregled metod za pakiranje pa v (Coffman et al., 1984). V doktorski disertaciji pa se v nadaljevanju ukvarjam samo z enodimenzionalnim razrezom.

3.3 Enodimenzionalni razrez

3.3.1 Splošni pristopi k reševanju

Ker so problemi dvodimenzionalnega razreza/pakiranja po strukturi praviloma bistveno drugačni od enodimenzionalnega, se v nadaljevanju doktorata osredotočim na enodimenzionalni razrez, ki je v praksi in teoriji tudi daleč najpogostejši. O enodimenzionalnem razrezu govorimo vedno, kadar so preostale dimenzije fiksne, zanemarljive ali pa sploh ne obstajajo (denimo »razrez« časa, denarja...).

Metode za rešitev problema enodimenzionalnega razreza glede na temeljni uporabljeni princip lahko razdelimo na dve veliki skupini (razdelitev povzemamo po Dyckoff, 1990), in sicer:

- metoda vzorcev (angl. pattern oriented): z različnimi metodami določimo vzorce rezanja in frekvenco vsakega vzorca. Večino tovrstnih metod temelji na algoritmu, ki sta ga prva predstavila Gilmore in Gomory (1961; 1963). Tovrstne metode so uporabne le v primeru, če so vse palice v zalogi enakih dolžin (oziroma če imamo nekaj različnih standardnih dolžin),
- posamično obravnavanje (angl. item oriented): pri teh metodah ne določimo vzorcev, ampak za vsako palico v zalogi posamično pripravimo načrt razreza. Metode na tej podlagi so splošno uporabne, saj jih lahko uporabimo tako pri standardnih kot pri nestandardnih dolžinah palic v zalogi. Seveda so ob standardnih dolžinah palic metode vzorcev običajno bolj fleksibilne in primernejše za uporabo pri večjih problemih. Po drugi strani pri nestandardnih dolžinah pridejo v poštev samo metode iz druge skupine.

Pri metodah vzorcev je pri iskanju optimalne rešitve največji problem veliko število možnih vzorcev, ki jih lahko uporabimo za razrez posamezne palice. Za srednje velik problem, kjer je denimo 50 različnih dolžin naročil in posamezno palico razrežemo na 5 delov (iz vsake palice torej pridobimo 5 različnih ali enakih dolžin naročil), lahko število možnih vzorcev za eno palico izračunam po enačbi za kombinacije s ponavljanjem¹⁷:

$$C_{n+r-1}^r = \binom{n+r-1}{r}$$

¹⁷ Formulo za kombinacije s ponavljanjem moramo izbrati zato, ker lahko posamezen element v izbrani podmnožici nastopa večkrat. Povedano drugače - iz vsake palice lahko torej praviloma izrežemo več enakih dolžin naročila

V našem primeru je torej $n = 50$, $r = 5$, saj iz množice s 50 elementi (število različnih dolžin naročil) izbiramo po 5 elementov (število dolžin naročil, ki se uporabijo za razrez posamezne palice). Tako lahko torej za posamezno palico uporabimo 3.162.510 različnih vzorcev. Ker moramo pri razrezu običajno uporabiti več palic, je število vseh možnih kombinacij vzorcev in palic, ki jih bomo uporabili pri razrezu, še bistveno večje. Iz tega je razvidno, da je vsaka metoda, ki bi se poskušala do rešitve prebiti s preizkusom vseh možnih vzorcev, vnaprej obsojena na neuspeh, razen za zelo majhne primere.

Zato metode, ki temeljijo na tem pristopu, za večje probleme ne preizkusijo vseh vzorcev, ampak samo nekatere. Ključno vprašanje, s katerim se soočajo, je, katere od možnih vzorcev preizkusiti. Tako se na različne načine najprej določijo najboljši vzorci in nato frekvenca za vsak posamezen vzorec. Od delovanja metode in vrste problema je odvisno, ali zagotovo privede do optimalne rešitve (kar je redko) ali ne.

Po drugi strani pa pri posamičnem obravnavanju vsako palico, ki jo je potrebno razrezati, obravnavamo ločeno in zanjo posebej določimo načrt razreza. V kolikor so vse palice različnih dolžin, kot je primer tudi v tej doktorski disertaciji, je posamično obravnavanje tudi edini realistično možen pristop k reševanju. Teoretično bi sicer lahko uporabili tudi metodo vzorcev, vendar bi bili v tem primeru edini možni frekvenci 0 in 1, tako da z uporabo metode vzorcev ne bi ničesar pridobili. Uporaba posamičnega obravnavanja je splošna in možna tudi v primeru, če so vse palice različnih dolžin, vendar je v takih primerih za večje primere običajno ustrežnejša uporaba vzorcev.

Tako pri metodah vzorcev kot pri posamičnem obravnavanju pa lahko uporabljamo eksaktne in hevristične metode za reševanje problemov. Eksaktne metode temeljijo na algoritmih, ki zagotovo v končnem številu korakov privedejo do optimalne rešitve problema. Poglavitna prednost teh metod je prav v tej optimalnosti rešitve. Glavna pomanjkljivost je eksponentialno naraščanje potrebnega časa za določitev rešitve ob večanju problema zaradi omenjene NP težkosti večine problemov razreza. Eksaktne metode običajno uporabljajo različne tehnike za pridobitev rešitve, kot so na primer:

- linearno programiranje (obsežen pregled in klasifikacija metod, ki temeljijo na linearnem programiranju, je v (Carvalho, 2002),
- metoda razveji in omeji – to metodo podrobneje predstavljam še v nadaljevanju te doktorske disertacije,
- dinamično programiranje.

Hevristična metoda je procedura, ki privede do dobrih ali skoraj optimalnih rešitev za optimizacijski problem. V splošnem so primernejše za večje probleme. V poštev pridejo predvsem v primerih, kjer algoritem za optimalno rešitev še ni bil najden ali pa je tak algoritem preveč računsko zahteven za večje primere (Eiselt, Sandblom, 2000). Tovrstne metode se štejejo

kot dovolj dobre, če so tako pridobljene rešitve dovolj dobre. To pomeni, da dokazano ali verjetno odstopajo od optimalne rešitve za manj kot neko (po različnih kriterijih določeno) še sprejemljivo vrednost. Dokaz, da rešitev odstopa manj kot za to vrednost, je lahko matematičen ali (kar je pogostejše) statističen na podlagi večjega števila generiranih in rešenih problemov (Hinxman, 1980).

Pod prvim imam v mislih matematični dokaz o največjem možnem odstopanju hevristične od optimalne rešitve. Pod drugim pa podrobno statistično analizo tako pridobljenih rešitev in njihovo primerjavo z optimalnimi (ali predhodno razvitimi metodami) za različne vrste problemov. Takšna analiza (primer le-te predstavlja tudi peto oziroma šesto poglavje te disertacije) lahko tudi pokaže, za kakšno vrsto in naravo problemov je primerna posamezna hevristična ali eksaktna metoda.

Problem hevrističnih metod je tudi ta, da so običajno precej specifične in primerne le za posamezen primer. Hevristična metoda, ki dobro deluje na enem primeru ob določenih ciljih in predpostavkah, namreč zelo verjetno ne bo ustrezna za kakšen drug problem, čeprav sta oba problema medsebojno navidez morda zelo podobna. To velja tudi za v disertaciji predstavljeno metodo CUT, ki ni več primerna za denimo v sedmem poglavju predstavljen problem, kjer število palic, ki jih lahko vračamo v skladišče (pod pogojem, da so dovolj dolge), povečam iz ena na neomejeno.

Hevristične metode lahko glede na način, kako poskušajo pridobiti dobro rešitev problema, razdelimo na več skupin (Hinxman, 1980; Nilsson, 1971):

- preiskava grafa (angl. state-space search): delne rešitve problema vzamemo kot vozle v grafu in poiščemo najboljšo pot v grafu od začetnega (v celoti nerešenega problema) do končnega stanja (končna rešitev problema, ki je optimalna ali vsaj dovolj dobra). Posamezni vozli in povezave med njimi se kreirajo dinamično (Chen, 1998),
- razdelitev problema (angl. problem reduction): osnovni problem razdelimo na več manjših podproblemov in vsak podproblem rešimo ločeno (ker so ti podproblemi manjši lahko uporabimo tudi eksaktne metode). Končna rešitev je unija posameznih rešitev,
- prekinitev iteracije (angl. cut-off): v kolikor eksaktna metoda temelji na pridobitvi optimalne rešitve z iteracijami, jo lahko pretvorimo v hevristično tako, da določimo pogoj, kdaj naj se iteracije prekinejo. Iteracije lahko prekinemo, ko se rešitev razlikuje od optimalne za manj kot neko določeno vrednost ali pa ko bi stroški in čas računanja presegli neko še sprejemljivo mejo. To metodo delno uporabim v tej disertaciji pri reševanju problemov z metodo razveji in omeji. Rešitve, pridobljene v neki vnaprej določeni časovni omejitvi, upoštevam kot končne rešitve, čeprav niso nujno optimalne,

- iskanje sprejemljive rešitve (angl. aspiration level): v kolikor hevristična metoda temelji na pridobivanju različnih možnih rešitev za problem, je možen pristop, da sprejmemo prvo rešitev, ki izpolnjuje nek vnaprej določen kriterij. To je koristno predvsem, kadar ne potrebujemo optimalne, ampak samo neko dovolj dobro rešitev,
- ponavljanje vzorca (angl. repeated exhaustion reduction): tovrstne metode najprej pridobijo nek ustrezen vzorec z majhno izgubo. Ta vzorec se nato ponovi čim večkrat, dokler ne pride do prevelike proizvodnje posamezne dolžine naročila. Nato izločimo zadnjo uporabo vzorca ter določimo nov vzorec, ki ga spet uporabimo čim večkrat. Tako nadaljujemo do popolne izpolnitve vseh dolžin naročila,
- vzorčenje (angl. sampling): gre za podoben pristop kot pri iskanju sprejemljive rešitve: identificiramo več možnih rešitev in za vsako izračunamo vrednost kriterijske funkcije. Nato izberemo točko z najboljšo (najnižjo pri minimizacijskih, najvišjo pri maksimizacijskih problemih) vrednostjo kriterijske funkcije.

3.3.2 Metode za rešitev enodimenzionalnega problema razreza

Prva kvalitetna in najbolj znana metoda za pridobitev dobre hevristične rešitve na podlagi vzorcev je metoda Gilmoreja in Gomoryja (1961; 1963). Različica problema, s katerim se ukvarjata v prvem članku, je naslednja: na zalogi imamo neomejeno število kosov posamezne standardne dolžine palice. Zagotoviti moramo določeno število kosov vnaprej znanih dolžin. Zaradi neomejene količine zaloge bo naročilo možno izpolniti v vsakem primeru, razen v kolikor je najdaljša dolžina naročila daljša od najdaljše dolžine palice. Vsaka palica na zalogi ima določeno ceno, skupni stroški razreza so vsota stroškov vseh v razrezu uporabljenih palic (Gilmore, Gomory, 1961). Cilj je poiskati načrt razreza z minimalnimi stroški.

Osnovni problem smo že omenili – gre za veliko število možnih vzorcev. Ta problem je v tej metodi rešen tako, da se v vsaki fazi simpleks metode, ko potrebujemo nov stolpec oziroma novo aktivnost¹⁸, ne pregleda celotna zbirka možnih stolpcev, ampak se ustrezen stolpec najde z rešitvijo pomožnega problema. V tem primeru gre pri tem problemu za tako imenovani problem nahrbtnika.

Tudi sam problem nahrbtnika je v preteklih desetletjih pritegnil veliko pozornosti teoretikov in praktikov. Rešiti ga je mogoče na različne načine, podrobnejši pregled metod za reševanje bi presegel okvir te doktorske disertacije. Zelo dober pregled starejših eksaktnih metod za rešitev problema nahrbtnika je v (Dudzinski, Walukiewicz, 1987), pregled približnih/hevrističnih metod pa v (Fisher, 1980).

V poznejšem članku (Gilmore, Gomory, 1963) je omenjena metoda prilagojena za uporabo pri razrezu papirja. Tako so narejene nekatere spremembe v formulaciji problema (omejitve pri

¹⁸ Z izrazom aktivnost je mišljen načrt razreza za posamezno dolžino palice v zalogi – v bistvu gre torej za vzorec.

številu nožev za rezanje, možnost približne izpolnitve naročila kupca) in v algoritmu za rešitev, ki predvsem vpliva na hitrejšo pridobitev rešitve (hitrejša metoda za rešitev problema nahrbtnika, način za zmanjšanje velikosti tega problema...). Osnovna ideja metode ostaja nespremenjena – pri pridobivanju rešitve v vsakem koraku uvedemo samo en nov stolpec (vzorec) in ne upoštevamo vseh možnih vzorcev, kar reši problem prevelikega števila možnih vzorcev.

V kasnejših letih se je razvilo veliko število metod, ki rešujejo take ali podobne probleme – nekatere metode gradijo na osnovnem pristopu Gilmorja in Gomoryja ter ga prilagodijo za konkreten primer, druge iščejo drugačne poti do končne rešitve, ki je bodisi optimalna bodisi neka dobra heuristična rešitev. Tudi vse v tej disertaciji razvite metode se zaradi specifičnosti problemov, ki so podrobneje predstavljene v nadaljevanju, odmaknejo od njunega pristopa in iščejo nove poti do optimalne ali vsaj zelo dobre rešitve.

Možno rešitev za enodimenzionalni razrez z eno dolžino palice je predlagana v (Vahrenkamp, 1996), kjer je uporabljen genetski algoritem¹⁹, pri katerem se v vsaki fazi algoritma upošteva samo tako imenovane učinkovite vzorce – to so taki vzorci, kjer je izguba manjša od neke vnaprej določene meje. Tak način sicer ne zagotavlja optimalne rešitve, pač pa neko rešitev, ki je tej blizu, dostikrat pa dobimo celo optimalno rešitev. Uporaba metod na temelju genetskih algoritmov je možna tudi pri minimizaciji izgub pri dvodimenzionalnem razrezu. Predstavitev metode, ki je – sodeč po eksperimentalnih rezultatih – boljša od preostalih, je v (Onwubolu, Mutingi, 2003).

Podobna je metoda predstavljena v (Ragsdale, Zobel, 2004), kjer je osnovna zahteva, da se posamezna naročila narežejo zaporedno. To naj bi bilo uporabno predvsem za hitrejšo in celostno izpolnjevanje naročil ob množičnem prilagajanju v proizvodnji, čeprav se članek s samim konceptom uporabe razreza pri omogočanju množičnega prilagajanja ne ukvarja. Do rešitve metoda pride s kombinacijo genetskega algoritma z različnimi heurističnimi metodami.

Uporaba metod na temelju genetskih algoritmov je možna tudi pri minimizaciji izgub pri enodimenzionalnem razrezu v jeklarstvu (Khalifa et al., 2006) ter pri dvodimenzionalnem razrezu. Genetski algoritmi za reševanje problema razreza se uporabljajo predvsem zaradi nekaterih pomanjkljivosti »klasičnih« pristopov (Shahin, Salem, 2004). Ravno tako se lahko genetski algoritmi uporabijo tudi pri operativnem načrtovanju razreza in obremenjenosti rezil - glej denimo (Sardinas et al., 2006) ali nekoliko širšo obravnavo podobnega problema ob vključitvi stroškov proizvodnje v (Yildiz, Ozturk, 2006). Pomen ustrezne formulacije in

¹⁹ Genetski algoritem (angl. genetic algorithm) je heuristična metoda za iskanje rešitev, ki posnema biološko evolucijo za iskanje vedno boljših rešitev za težke optimizacijske probleme (Holland, 1992), (Koza et al., 1999). Nove rešitve se iščejo s "parjenjem" (menjavanjem komponent med posameznimi rešitvami) in "mutacijami" (naključne zamenjave med rešitvami). Rezultat je "bazen genov", ki sčasoma vodijo do dobrih rešitev, pri čemer se obdržijo predvsem dobri "kromosomi" (Ragsdale, Zobel, 2004). Običajno se s takim pristopom najde dobra rešitev v razumni časovni omejitvi (Berger, Ragsdale, 1999).

morebitne transformacije modela za pripravo načrta razreza pri uporabi genetskih algoritmov je poudarjen v (Yeung, Tang, 2003), kjer je genetski algoritem kombiniran z ustrezno hevristično metodo, ki vodi do zelo dobrih rezultatov ob kratkem času reševanja.

V (Antonio et al., 1999) sta predstavljeni dve metodi na podlagi dinamičnega programiranja, pri katerih v vsaki fazi obdržimo samo tiste rešitve, za katere je najverjetneje, da bodo privedle do dobre rešitve. Manjše število rešitev, ki jih obdržimo v posamezni fazi, pomeni hitrejši čas reševanja, hkrati pa tudi večjo verjetnost, da bo tako dobljena rešitev slaba. Pri eni od predstavljenih metod je tako poudarek na času reševanja (ta se uporablja, kadar je potreben takojšen odgovor na povpraševanje kupcev in približna ocena možnosti za razrez), pri drugi na kvaliteti rešitve (ta se uporablja za pridobitev boljših rešitev v nočnem času).

Standardnega problema razreza se lahko lotimo tudi tako, da poiščemo rešitev LP sprostitve (angl. LP relaxation), nato pa realne vrednosti zaokrožimo na celoštevilčne. Dve hevristični metodi za zaokroževanje, ki vodita do dobrih (dostikrat tudi optimalnih) rešitev sta predstavljeni v (Scheitauer, Terno, 1995).

Gradišar et al. (1999a) predstavljajo podoben problem, kjer je večina zalog v obliki ene ali nekaj standardnih dolžin palic, del zaloge pa predstavljajo nestandardne dolžine palic, ki so večinoma še uporabni ostanki iz prejšnjih razrezov. Predstavljena rešitev je dvofazna: najprej pridobijo vzorce, nato v drugi fazi odstranijo tiste vzorce, ki vsebujejo več naročil, kot jih potrebujemo (s tem preprečijo preveliko proizvodnjo posameznih naročil). Seveda s tem za nekatera naročila ne pridobijo dovolj kosov. Ta del nato rešimo z ustrezno zaporedno hevristično proceduro, končno rešitev pa dobimo tako, da združimo rezultate obeh faz.

S podobnim problemom večjega števila standardnih dolžin se ukvarja Holthaus (2002), ki uporabi tudi precej podoben pristop. Najprej poišče optimalno rešitev za LP sprostitve. Tako pridobljene vrednosti nato zaokroži navzdol, preostanek problema pa reši z optimalnimi ali hevrističnimi metodami. Na koncu se tako kombinirana rešitev še poizkuša izboljšati z uporabo posebnih algoritmov. Podoben problem rešujeta tudi Alves in Carvalho (2008), ki opozorita na primanjkljaj eksaktnih rešitev za probleme z različnimi dolžinami palic²⁰. Predlagata pristop na podlagi razveji-oceni-odreži (angl. branch-and-price-and-cut), za katerega na podlagi numeričnih eksperimentov pokažeta, da je boljši od predhodnih. Del njune rešitve je tudi pristop na temelju razveji in omeji za hitrejšo generacijo zanimivih stolpcev – tak pristop lahko bistveno pohitri delovanje algoritmov, ki ga uporabljajo.

Podobno Lee (2007) prične z rešitvijo celoštevilskega linearnega problema, ki jo izboljša s hevrističnim pristopom. Posebna pozornost je namenjena generiranju novih stolpcev, ki naj bodo taki, da izboljšujejo strukturo celoštevilskega problema. Metoda naj bi bila posebej primerna v

²⁰ Enega od možnih pristopov za eksaktno rešitev problema razreza z različnimi dolžinami palic predstavljam v petem poglavju

primerih, ko je zaradi omejitev proizvodnje možno uporabiti le omejeno število vzorcev. Metoda namreč poišče rešitev z nižjim številom vzorcev tudi na račun nekaj višje izgube materiala. Metoda je numerično testirana na primeru podjetja iz kemične industrije.

Dolžina palic na zalogi pa ni nujno vnaprej določena. Cui et al. (2006) tako predstavljajo metodo, ki najprej določi optimalno dolžino palic na zalogi, nato pa še dinamični algoritem, ki pripravi načrt razreza. Računski rezultati so pokazali, da je algoritem ustrezen tako glede računske zahtevnosti kot tudi izgube materiala.

Eden od problemov pri Gilmorejevi metodi je tudi nelinearnost, ki nastane zaradi medsebojne odvisnosti med dvema spremenljivkama: posameznimi uporabljenimi vzorci in pogostostjo njihove uporabe. Metoda, predstavljena v (Johnston, Sadinlija, 2003), ta problem rešuje z mešano-celoštevilskim programiranjem ob uvajanju novih binarnih spremenljivk.

Pomanjkljivost metode Gilmore-a in Gomory-ja, ki generira le majhen del vzorcev, delno popravljata Degraeve in Schrage (1999), ki kombinirata to metodo z metodo razveji in omeji. S pomočjo te metode se pridobi tiste stolpce, ki jih Gilmorejeva metoda ne upošteva, lahko pa privedejo do optimalne rešitve. Predstavljena metoda vodi do optimalnih rešitev in je nekoliko boljša od prejšnje metode (Wäscher, Gau, 1994), ki je do zagotovo optimalne rešitve privedla v 3984 od 4000 primerov (torej v 99,6 % vseh obravnavanih primerih). Avtor podrobnejšo primerjavo svoje metode z drugimi podobnimi metodami za praktične probleme iz industrije predstavlja v kasnejšem članku (Degraeve, Peeters, 2003).

Prej omenjen problem prevelikega števila možnih vzorcev, ki vodijo do računsko nerešljivih problemov, je možno rešiti tako, da vnaprej generiramo možne vzorce s pomočjo iskalnega drevesa (angl. search tree) in nato pri izdelavi načrta razreza upoštevamo samo te vzorce (Suliman, 2001).

Belov in Scheithauer (2002) predstavljata metodo za rešitev problema razreza s palicami različnih dolžin (na zalogi imamo več primerkov palice posameznega tipa). Pri tej metodi, ki je primernejša za probleme z več različnimi tipi palic kot za tiste z manj, kombinirata tako imenovani rezalni algoritem (angl. cutting plane algorithm) z generacijo stolpcev. Na ta način dosežeta optimalno rešitev pri približno 90% testnih primerov ob časovni omejitvi nekaj minut. Ista avtorja (Belov, Scheitauer, 2006) sta za podoben problem kasneje razvila še metodo, ki najprej pridobi rešitev za LP sprostitev, ki je že zelo dobra meja pri iskanju optimalne rešitve. Ta rešitev se nato še izboljša z uporabo mešano-celoštevilskih rezov. Vendar so praktični eksperimenti pokazali, da metoda za problem enodimenzionalnega razreza le redko vodi do dobrih rešitev. Zato pa pri dvodimenzionalnem problemu razreza večinoma pripelje do boljših rešitev kot primerljive metode.

Kombinacija hevristične in eksaktne metode, ki vodi do skoraj optimalnih rešitev tudi za zelo velike probleme pri splošnem problemu enodimenzionalnega razreza je predstavljena v

(Gradišar, Trkman, 2002a; Gradišar, Trkman, 2005). Pri tej metodi, ki je bila razvita v okviru raziskav pri pripravi te doktorske disertacije in je v nadaljevanju tudi podrobneje predstavljena, najprej z obstoječo hevristično metodo pridobimo dobro (vendar neoptimalno) rešitev, ki jo v drugem koraku izboljšamo tako, da manjši, kritični del problema rešimo še z eksaktno metodo. Metoda vodi do zelo dobrih rešitev ob razmeroma nizki časovni zahtevnosti reševanja.

Metoda uporabi možnost ponovne uporabe delno narezanega materiala, kar je v zadnjih letih razmeroma pogosto uporabljen pristop. Tako denimo metoda, predstavljena v (Kos, Duhovnik, 2002), minimizira izgubo materiala, ob čemer predpostavlja, da se lahko delno narezan material vrne v skladišče in kasneje ponovno uporabi. Njuna verzija problema je primer enodimenzionalnega pakiranja z različno velikostjo kontejnerja. Problem rešita s hibridnim genetskim algoritmom, ki vodi do skoraj optimalnih rešitev. Na študiji primera pokažeta, da lahko tak pristop vodi do izboljšane priprave in izvedbe procesa pri razrezu jekla in kovin. Avtorja poudarita pomen ustreznega uporabniškega vmesnika za praktično uporabo njune metode.

Ponovno uporabnost materiala vključuje tudi metoda predstavljena v (Alfieri et al., 2007), kjer je potrebno večje število naročil razporediti v palice na zalogi tako, da je posamezen ostanek bodisi dovolj kratek (da se lahko zavrže) ali pa dovolj dolg (tako se lahko verjetno uporabi za razrez v prihodnje). Primer je (ob dovolj veliki količini palic na zalogi) razmeroma trivialen, zato ne preseneča, da avtorji dokažejo, da je njihova verzija problema rešljiva v $O(n \log n)$ času – ne gre torej za NP težak problem.

3.3.3 Večkriterijske metode za razrez

Do sedaj predstavljene rešitve so praviloma predpostavljale, da je edini cilj doseči čim manjšo možno izgubo materiala. Dostikrat pri pripravi načrta razreza želimo doseči tudi druge cilje, kot so na primer čim manjše število vzorcev, čim krajši čas razreza in podobno.

Venkateswarlu (2001) tako navaja naslednjih 5 kriterijev za presojo ustreznosti rešitev:

- izguba materiala: kot odstotek celotnega uporabljenega materiala,
- prevelika proizvodnja: kosi naročila, ki presegajo naročilo – kot odstotek vseh narezanih kosov²¹,
- povprečno stanje zalog: izmerjeno stanje zalog po izpolnitvi kupčevega naročila,
- povprečno število različnih dolžin palic v skladišču: povprečje različnih dolžin, ki ostanejo po izpolnitvi naročila kupca,

²¹ Prevelika proizvodnja (angl. overproduction) je problem pri metodah vzorcev, saj se lahko zaradi ponavljanja vzorcev zgodi, da narežemo preveč kosov posameznega naročila. Modeli, ki so predstavljeni v nadaljevanju doktorske disertacije, temeljijo na posamičnem obravnavanju palic, tako da do tega problema ne prihaja.

- povprečno število palic v skladišču: število palic, ki ostanejo v skladišču po izpolnitvi naročila kupca.

V doktoratu se najprej ukvarjam z enokriterijskim problemom razreza (edini cilj je čim manjša izguba materiala, torej prvi kriterij). Od sedmega poglavja dalje pa problem razširim z vključevanjem novih kriterijev, predvsem sta pomembna tretji in peti. Poleg tega pa je omogočena tudi vključitev drugih dejavnikov. Nato preučujem še doseganje kriterijev v zaporednih časovnih obdobjih.

Zgornji kriteriji so sicer prilagojeni za v omenjenem članku predstavljen problem razreza, vendar pa se podobni kriteriji (včasih je dodan še kakšen ali izpuščen kateri od obstoječih) uporabljajo tudi pri drugih metodah, ki so predstavljene v nadaljevanju tega poglavja. Predhodne metode so se osredotočale le na prvo točko, ki je v marsikaterem primeru dejansko najpomembnejša.

V tem članku (Venkateswarlu, 2001) se avtor sicer ukvarja z dvodimenzionalnim problemom pravokotnega razreza v lesni industriji, pri čemer je material različnih oblik in v različnih kakovostih lesa. Predstavlja 14 različnih hevrstičnih metod za razrez, pri čemer osnovni princip nekoliko spominja na v nadaljevanju predstavljeni CUT. Pred reševanjem namreč razvrsti kose v zalogi po velikosti in najprej išče rešitev za daljše, krajše ali ustrezno kombinacijo daljših in krajših kosov (odvisno od delovanje posamezne hevrstične metode).

Med članki, ki vključujejo različne kriterije pri optimizaciji razreza, je eden najzanimivejših (Gramani, Franca, 2006). Avtorja najprej pokažeta, da problem povezave razreza in določanja velikosti serij še ni bil ustrezno rešen. Članek analizira povezavo med problemom razreza in planiranja v različnih obdobjih in poskuša minimizirati vsoto stroškov izgube materiala, skladiščenja in priprave proizvodnje ter predlaga matematični pristop za rešitev tega problema. Prvi rezultati kažejo, da povezava problema razreza in priprave proizvodnje (oz. velikosti proizvodnih serij) lahko vodi do približno 28% boljših rezultatov, kot če bi vsakega od teh problemov reševali ločeno.

Vsebinsko zelo podoben problem rešujeta avtorja v (Nonaas, Thorstenson, 2000; Nonaas, Thorstenson, 2008). V prvem članku najprej razvijeta metodo, ki deluje pri statičnih in determinističnih podatkih. V kasnejšem članku svojo metodo kombinirata s predhodno razvito hevrstično metodo (Hässler, 1971) ter tako razvijeta metodo, ki je primerna tudi za velike probleme. S testiranjem na večjem številu primerov pokažeta, da je njuna metoda primerna tudi za večje probleme ter da vodi do dobrih rezultatov tudi, če upoštevamo samo stroške priprave proizvodnje in menjave vzorca.

Eksaktna rešitev za problem, pri katerem je cilj minimiziranje potrebnega števila vzorcev za izpolnitev naročila (s tem dosežemo tudi minimiziranje potrebnih sprememb nastavitev rezila in tako praviloma krajši čas razreza), je predstavljena v (Vanderbeck, 2000). S to metodo je avtorju v dveh urah uspelo optimalno rešiti 12 od 16 testnih problemov (75% vseh). Podoben problem iz

jeklarske industrije rešujeta avtorja v (Hajizadeh, Lee, 2007) – potrebno je razrezati neskončno dolgo jekleno cev. Do izgube materiala tako seveda sploh ne pride, zato metoda minimizira vsoto časa za razrez in časa za spremembo vzorca za izpolnitev povpraševanja. Čas za spremembo vzorca je pri tem definiran kot linearna funkcija števila uporabljenih nožev v posameznem vzorcu. Njun pristop naj bi vodil do 44% boljših rezultatov od obstoječih.

V (Umetani et al., 2003) je predstavljena hevristična metoda, pri kateri je glavni cilj minimiziranje števila vzorcev, ki jih uporabimo v načrtu razreza. Število vzorcev je glavni cilj zaradi predpostavke, da največjega stroška pri razrezu materiala ne predstavlja morebitna izguba materiala ampak premestitev rezil, ki je potrebna ob vsaki spremembi vzorca. Tudi tu izguba materiala ni nepomembna – postavljena je namreč omejitev, koliko največ je lahko izguba posameznega vzorca. Vzorci, ki to omejitev presegajo, ne pridejo v poštev za razrez.

Zelo podobna tej je preprosta hevristična rešitev, ki je predstavljena v (Yanasse, Limeira, 2005), kjer poskušajo zmanjšati tako izgubo kot predvsem število uporabljenih vzorcev (glavni razlog za slednje so predhodno omenjeni stroški spremembe rezil). Tudi tu pridejo v poštev samo nekateri vzorci – tisti, ki poleg izgube pod določeno mejo izpolnjujejo tudi pogoj, da iz njih narežemo vsaj dve različni naročili (razlog za slednje je predvsem preprečevanje prevelike proizvodnje). Podobne cilje zasleduje metoda v (Förster, Wäscher, 2000).

Kot vidimo, se pri večini metod, ki poskušajo zmanjšati število vzorcev, uporablja pristop večkratne uporabe posameznega dobrega vzorca (angl. RPET: Repeated pattern exhaustion technique). To je seveda logično, saj čimbolj večkratna uporaba posameznega vzorca zagotovo zmanjšuje skupno število potrebnih vzorcev.

Podobna kot nekatere od zgoraj opisanih, je metoda, ki ravno tako kreira vzorce in omogoča vključitev drugih dejavnikov v izdelavo načrta razreza, kot so denimo stroški menjave vzorcev ter prihodek od morebitnega presežka narezane količine (Schilling, Georgiadis, 2002).

Westerlund et al. (1998) predstavljajo dvofazno metodo za minimiziranje stroškov pri razrezu v papirni industriji. Stroške predstavljajo izguba materiala, skladiščenje delno porabljenih navitkov papirja ter spremembe v nastavitvi stroja in nožev. Prvi korak metode je nelinearna procedura za generiranje vseh možnih vzorcev razreza. V drugem koraku se končna rešitev pridobi z rešitvijo mešano celoštevilskega linearnega problema (angl. mixed integer linear programming).

Ravno tako se z izgubo materiala v papirni industriji ukvarjajo Yen et al. (2004). Gre za standardni problem razreza, ki pri reševanju upošteva tudi stroške spreminjanja vzorca in ponovne nastavitve rezil. V članku je posebej poudarjena pomembnost testiranja čim večjega števila dobrih rešitev. To načelo upoštevata tudi metodi CUT in C-CUT, ki sta predstavljeni v nadaljevanju doktorske disertacije.

Metoda, ki se v papirni industriji ukvarja predvsem z osnovnim problemom razporeditve naročil po več identičnih strojih, je razvita v (Menon, Schrage, 2002). Osnovna ideja je hitra pridobitev dobre rešitve, ki se lahko uporabi kot zgornja meja za črtanje (angl. pruning) večjega števila vozlov pri metodi razveji in omeji. Njena metoda je učinkovita predvsem v primerih, ko že kmalu dobimo tesno zgornjo mejo, nakar lahko hitro izločimo velik del možnih rešitev. Podobna ideja je uporabljena tudi pri v nadaljevanju predstavljeni metodi C-CUT, vendar je pristop pri izpeljavi te ideje bistveno drugačen.

Vasko et al. (1999) opisujejo algoritem, ki se uporablja za razrez jekla. Ta algoritem na podlagi hierarhičnega pristopa zasleduje naslednje 3 cilje (cilji so navedeni po pomembnosti v padajočem vrstnem redu):

- čim bolje izpolniti naročilo kupca (torej imeti čim manj nenarezanih naročil) – to je glavni cilj tega algoritma,
- čim bolj zmanjšati razrez iz predobrega materiala (angl. overgrading): dostikrat se zgodi, da naročnik želi material slabše kakovosti. Če podjetje tega naročila ne more izpolniti s to kakovostjo, ga lahko izpolni z boljšim materialom. To pa seveda pomeni ustrezno višje stroške, ki jih mora kriti podjetje, saj stroškov višje kakovosti običajno ni možno prevaliti na kupca. Zato algoritem poskuša tak razrez iz predobrega materiala čim bolj zmanjšati,
- čim manjša izguba materiala – kot vidimo, je ta cilj šele na tretjem mestu, saj je za doseganje poslovnih ciljev podjetja v tem konkretnem primeru pomembnejša izpolnitev drugega in predvsem prvega cilja.

Podoben kot zadnji primer je predstavljen v (Rönnqvist, 1995), kjer imamo pri razrezu lesa v isti deski les različne kvalitete. Poglavitna omejitev pri tem problemu je zelo kratek čas za odločitev, saj se mora načrt razreza izdelati v 2 sekundah, nato gre deska na žago. Zato je izvajanje algoritma omejeno na 10 iteracij, ki običajno najdejo dobro rešitev, ki pa ni nujno tudi optimalna.

Čim boljša izpolnitev naročila kupca ali proizvodnje je tudi cilj metode, predstavljene v (Trkman, Gradišar, 2003). Pri tej metodi ob pomanjkanju materiala poleg izgube pri razrezu upoštevamo tudi, katera naročila so bolj nujna in bi njihova neizpolnitev pomenila večje stroške za podjetje. Ta metoda je bila razvita v okviru raziskav te doktorske disertacije in je podrobneje predstavljena v sedmem poglavju.

Podobno se s prioriteto naročil ukvarjajo tudi v (Ghodsi, Sassani, 2005), kjer pa se večja prioriteta arbitrarno določi tistim naročilom, ki so zaradi določenih pogojev težje izpolnjiva. Ker mora metoda delovati v realnem času, se nadaljuje s preverjanjem le tistih kombinacij, ki imajo majhno izgubo ali pa vsebujejo veliko naročil z višjo prioriteto. Metoda se dopolnjuje s podobno metodo (Ghodsi, Sassani, 2005a), kjer pa je predvsem poudarek na reševanju problema različne

kakovosti lesa pri razrezu v pohištveni industriji. Kakovost materiala namreč ni znana vnaprej, zato mora metoda najti ustrezno rešitev v realnem času.

Podobno stremi k čim boljši izpolnitvi naročila kupca tudi metoda za razrez papirja različne kakovosti. Polizdelki se nato prodajo kupcu ali pa uporabijo v nadaljnji proizvodnji, medtem ko ponovna uporaba rol ni možna (Chauhan et al., 2008). Za binarni nelinearni model avtorji predlagajo dva pristopa za reševanje: razveji in oceni (angl. branch & price) ter hevristično metodo na temelju mejnih stroškov. Metodo so testirali na praktičnem primeru ter tudi na večjem številu naključno generiranih problemov.

Na rezultatih in vhodnih podatkih avtorja doktorata temelji metoda predstavljena v (Cherri, 2006; Cherri et al., 2007; Cherri et al., 2008), ki obravnava predvsem problem, kdaj je ponovna uporaba materiala smiselna. Obstoječe hevristične metode prilagodi tako, da ne pripeljejo do neželenih vzorcev (to so predvsem tiste s srednje dolgimi ostanki, ki niso primerni za ponovno uporabo in hkrati predolgi za odmet). Metodo testirajo na več naključno generiranih problemih. Podobno na mojem pristopu gradijo v (Yang et al., 2006). Tako kot v predhodnih člankih (Gradišar et al., 1997; Gradišar, Trkman, 2005) ostanki palic, ki so daljši od neke predhodno določene meje, ne štejejo kot izguba. Metoda je testirana na primerih razreza v ladjedelništvu.

V (Chu, Antonio, 1999) je predstavljen hevristični algoritem za minimiziranje izgube materiala in časa razreza pri rezanju kovinskih cevi. Kriterijska funkcija je tu ponderirana vsota izgube in časa, potrebnega za izvedbo razreza (čas se lahko skrajša s hkratnim razrezom več palic). Podobno kot v (Antonio et al., 1999) sta predstavljeni dve različici algoritma – ena hitro privede do solidne rešitve, druga v nekaj daljšem času do rešitve, ki je blizu optimalne. Ta algoritem pripelje do rešitev, ki so za skoraj 10% boljše od tistih, ki so jih prej ročno izdelali izkušeni zaposlenci.

Faggioli in Bentivoglio (1998) predstavljata trifazno metodo za problem določanja vrstnega reda razreza. Najprej se s požrešnim algoritmom (angl. greedy algorithm)²² pridobi dobra začetna rešitev. Ta rešitev se nato izboljša z generaliziranim lokalnim iskanjem. Tako dobljena rešitev se nato uporabi kot zgornja meja pri iskanju optimalne rešitve z implicitnim oštevilčevanjem (angl. implicit enumeration). Kot pri vseh metodah za optimalno reševanje je problem hitro naraščanje časa z naraščanjem velikosti problema – za problem s 40 vzorci se tako porabi skoraj cel dan za optimalno rešitev. Seveda lahko v tem primeru obdržimo rešitev druge faze kot končno (hevristično) rešitev problema. Ta rešitev potem seveda ni optimalna, je pa večinoma blizu optimalne in predvsem zahteva bistveno krajši čas reševanja.

Z določanjem vrstnega reda razreza se ukvarjata tudi Yuen in Richardson (1995), ki ravno tako predstavljata dve metodi. Enostavnejša se lahko uporablja le v nekaterih specifičnih primerih,

²² Požrešni algoritem v vsaki fazi reševanja izbere trenutno najboljšo možnost brez upoštevanja morebitnih kasnejših problemov pri reševanju (Biggs, 2002).

medtem ko je druga splošneje uporabna. Pri drugi metodi, ki preišče vsa možna zaporedja razreza, je spet problem čas računanja, ki narašča več kot proporcionalno z naraščanjem števila vzorcev. Določanje vrstnega reda na različnih strojih, na katerih se izvaja razrez, je bistveno tudi pri primeru razreza jekla v gradbeništvu, ki je predstavljen v (Armbuster, 2002).

Degraeve in Vandebroek (1998) sta avtorja mešano celoštevilskega modela za razrez v tekstilni industriji, kjer se minimizira število potrebnih nastavitev ob ohranitvi nizke izgube. Model omogoča upoštevanje različnih omejitev, kot so na primer dolžina nožev, debelina tkanine in dolžina mize, na kateri izvajamo razrez. Podobno Liang et al. (2002) v svoji metodi na podlagi evolucijskega programiranja poleg čim manjše izgube zasledujejo tudi cilj, da bi bilo čim manjše število palic z izgubo. Pri tem želijo minimizirati tako število uporabljenih kot predvsem število delno uporabljenih palic.

Nekoliko drugačen od prej omenjenih problemov enodimenzionalnega razreza je problem večfaznega razreza (Zak, 2002; Zak, 2002a). Tu najprej material na prvem stroju razrežejo v polizdelke, nato pa razrez dokončajo na drugem stroju. Za pridobitev dobre rešitve za ta problem je bila uporabljena posplošitev Gilmorejeve metode generiranja stolpcev. Poleg stolpcev se namreč sedaj sprti generirajo tudi vrstice, ki predstavljajo dolžino polizdelkov.

Več ciljev ima tudi metoda, ki uporablja linearno programiranje za minimiziranje celotnih stroškov, povezanih z razrezom. To so denimo stroški prevelike količine narezanih naročil, stroški materiala, stroški skladiščenja in seveda stroški izgube materiala. Pri tem je možen problem tudi pridobitev natančnih in popolnih podatkov, denimo o stroških zalog (Wäscher, 1990).

Carvalho in Rodrigues (1995) predstavljata dvofazni problem razreza, kjer v prvi fazi najprej dobimo vmesne polizdelke, potem pa te v drugi fazi razrežemo na končne izdelke. Ta metoda poleg čim manjše izgube zasleduje tudi druge cilje, kot so na primer čim manjše število vzorcev, čim manjše število vmesnih dolžin palic in čim manjše število različnih polizdelkov (zaradi lažjega načrtovanja razreza v drugi fazi). Po ugotovitvah avtorja je ta metoda primerna za manjše in srednje velike probleme, za večje pa ne.

Dvofazni je tudi algoritem za razrez v jeklarstvu (Ferreira et al., 1990), ki temelji na modificiranem Gilmorejevem algoritmu. Hevristična procedura generira dobre vzorce za obe fazi in vsak vzorec uporabi čim večkrat. Ta procedura se ponavlja, dokler niso izpolnjene vse potrebe. Najprej poiščemo rešitev za drugo fazo, na podlagi pridobljenih rezultatov še za prvo, pri čemer se najprej poišče rešitev za dolga naročila. Pri iskanju rešitve se upoštevajo stroški materiala in stroški spremembe vzorca.

Wagner (1999) se ukvarja z razrezom stavbnega lesa. Ker ta metoda prvotno dobi neceloštevilске rezultate, je poseben poudarek na zaokrožitvi teh števil – metoda za zaokroževanje lahko namreč pomembno vpliva na končno izgubo. Ugotovljeno je tudi, da iz

razumljivih razlogov metode na podlagi linearnega programiranja običajno najdejo rešitev z več zamenjavami vzorcev kot druge metode. V to metodo je možno vključiti tudi druge cilje, kot so na primer čim manjše število zamenjav vzorcev ali minimalno število porabljenih kosov materiala.

Podobne probleme, kot je razrez, najdemo tudi denimo pri planiranju letalskega prometa. Tak problem, ki je dejansko 3-dimenzionalen problem pakiranja v različno velike kontejnerje, je predstavljen v (Chan et al., 2006). Najprej je z LP modelom določena spodnja meja za globalni optimum (minimalni stroški palet, ki so potrebne). Kot je pogosto pri tovrstnih metodah (podoben pristop uporablja tudi metoda C-CUT), se pridobljena rešitev izboljša s hevristično metodo, ki določi tudi vrstni red in postopek nalaganja palet na letala.

Uporabnost problema razreza na različnih področjih kaže tudi metoda, predstavljena v (Morgan et al., 2006), kjer se ukvarjajo s problemom lansiranja satelitov v vesolje. Ob danih zahtevah (torej potrebah po pokritosti s sateliti) je potrebno zagotoviti takšno kombinacijo izstreljenih satelitov, ki bo minimizirala stroške. Pristop za reševanje je zelo podoben kot pri klasičnih problemih razreza.

Zaradi številnih metod, ki rešujejo podobne probleme, se več člankov ukvarja tudi z medsebojno primerjavo posameznih metod. Tako denimo v (Gradišar et al., 1999b; Gradišar et al., 2002) primerjajo dve hevristični metodi za splošni enodimenzionalni problem razreza. V (Bingul in Oysu, 2005) je primerjava treh hevrističnih metod za različno zahtevne verzije istega problema. Trkman in Gradišar (2002; 2003a) pa s pomočjo odločitvenih dreves primerjata eksaktno in hevristično metodo. Praviloma je primernost posamezne metode odvisna tudi od velikosti in računske zahtevnosti problema.

Poleg matematičnih metod za pripravo vzorcev ali posamičnih načrtov za posamezno palico, pa nekateri avtorji obravnavajo tudi samo izvedbo razreza, pri čemer vključujejo predvsem fizikalne lastnosti materiala ali rezil. Tako se denimo Stosic (2006) ukvarja predvsem z neenakomerno obrabo rezil in problemi, povezanimi s tem; Kopač (2002) pa obravnava sile, ki delujejo na posamezno rezilo pri razrezu. Podrobnejša obravnava tovrstnih metod bi presegla okvire te doktorske disertacije.

Vse v tem poglavju in nadaljevanju disertacije predstavljene metode so uporabne v praksi za doseganje želenih ciljev pri razrezu ali podobnih problemih. Metode operacijskih raziskav se lahko razmeroma enostavno vključijo tudi v sisteme za podporo odločanja, kar predvsem olajša njihovo praktično uporabo za ljudi s slabšim poznavanjem matematičnega ozadja metod (denimo osebam, ki so v podjetju zadolžene za operativno pripravo proizvodnje). Praktičen primer integracije metod v take sisteme je na primeru razreza v pohištveni industriji predstavljen v (Čižman, Černetič, 2004), medtem ko je v (Ladanyi et al., 2005) predstavljena pobuda, ki (ob uporabi odprtokodnih principov) stremi k hitrejšemu razvoju in deljenju znanja med strokovnjaki na področju operacijskih raziskav.

Po tej kratki predstavitvi metod za druge primere se v nadaljevanju ukvarjam le še z različicami splošnega enodimenzionalnega problema razreza in njegovimi razširitvami. Omenjeno različico problema podrobneje predstavljam v naslednjem poglavju disertacije.

4 Definicija problema razreza

Problem, s katerim se ukvarjam v tej doktorski disertaciji, je ena od številnih različic enodimenzionalnega problema razreza. Kot že omenjeno, se ta problem od preostalih do sedaj v doktorski disertaciji predstavljenih problemov razlikuje toliko, da ni mogoče uporabiti zgoraj opisanih hevrističnih metod. Poglavitni razliki sta, da so vse palice različnih dolžin in da največ ena delno narezana palica ne šteje kot ostanek.

Do različnih dolžin palic v skladišču lahko pride iz različnih razlogov, denimo zaradi prihrankov pri naročanju, različnih dobaviteljev, različnih naravnih danosti²³ in podobnih razlogov. Do tega lahko pride tudi zaradi zavestne odločitve podjetja, da bo skladiščilo različne dolžine zaradi večjega števila kombinacij za razrez ali boljšega izpolnjevanja naročil kupcem oziroma prilagajanja le-tem (denimo pri uporabi množičnega prilagajanja). Dodatna možnost za nastanek tega problema je tudi skladiščenje in ponovna uporaba posameznih palic.

Dokazano je, da večje število različnih dolžin palic v zalogi vodi do boljših rešitev predvsem v primerih, ko je manj različnih naročil in manjše število kosov posameznega naročila (Holthaus, 2003). Omenjena analiza ne vključuje vračanja in ponovne uporabe materiala, ki je ena od posebnosti metod, razvitih v tej disertaciji.

Obravnavani problem je definiran tako: za izpolnitev naročila (vsako naročilo predstavlja točno določeno število kosov posamezne dolžine) imamo na razpolago določeno število palic, dolžina palic je v splošnem medsebojno različna. Predpostavljamo, da so vsi podatki celoštevilčni. V kolikor niso celoštevilčni, jih je možno pomnožiti z določenim faktorjem in tako dobiti celoštevilčne podatke.

Cilj je pridobiti tak načrt razreza (natančen načrt, katere palice bomo uporabili pri razrezu ter katere dolžine naročil bomo izrezali iz vsake posamezne palice), ki bo prinesel čim manjšo izgubo materiala. Gre torej za enokriterijski problem, ki ga v nadaljevanju doktorske disertacije razširimo z dodatnimi kriteriji (različnim pomenom izpolnitve posameznih naročil) ter vključitvijo kontinuiranega zagotavljanja materiala.

V splošnem ločimo primere, kjer je dovolj materiala, ter primere, kjer materiala primanjkuje. V obeh primerih je cilj minimizirati izgubo materiala, pri čemer v primerih, kjer materiala primanjkuje, del naročila ne bo izpolnjen.

²³ Praktičen primer so denimo drevesna debela različnih dolžin.

Tako palice kot naročila obravnavamo kot enodimenzionalne – to pomeni, da sta dve dimenziji fiksni, nas pa zanima samo načrt razreza za tretjo dimenzijo – gre torej za enodimenzionalni razrez. V skladu z Dyckhoffovo klasifikacijo (1990) lahko naš problem razreza uvrstimo v kategorijo 1/V/D/M za primere z dovolj materiala oziroma v kategorijo 1/B/D/M za primere s premalo materiala

V primerih z dovolj materiala nam ostane ena palica, ki ni do konca narezana. Ta ostanek lahko vrnemo v skladišče in je uporaben tudi v kasnejših načrtih razreza. Zato ne šteje kot izguba, vendar le v kolikor je daljši od neke vnaprej določene zgornje meje. Z eno od omejitev v modelu (omejitve 5) določimo, da lahko v skladišče vrnemo največ eno preostalo palico. Tako namreč preprečimo take načrte razreza, ki bi vodili do hitrega naraščanja števila palic v skladišču. Te palice bi bile namreč čedalje krajše, kar bi vodilo do večjih izgub v kasnejših razrezih.

V modelih uporabljamo naslednje spremenljivke:

$s_i =$ dolžina naročil; $i = 1, \dots, n$

$b_i =$ potrebno število kosov dolžine s_i

$d_j =$ dolžine palic v zalogi; $j = 1, \dots, m$

$x_{ij} =$ število kosov dolžine s_i , ki smo jih narezali iz palice d_j

UB – zgornja meja za izgubo – palica daljša od te meje se lahko uporabi še kasneje in zato ni nujno, da šteje kot izguba

Model se seveda razlikuje za primere z dovolj in primere s premalo materiala ter ga zapišemo takole.

Primer 1: Naročilo lahko izpolnimo, saj imamo na zalogi dovolj materiala (izpolnjen je torej

pogoj $\sum_{j=1}^m d_j \geq (\sum_{i=1}^n s_i * b_i)$ - celotni razpoložljivi material presega celotne potrebe)

(1) $\min \sum_{j=1}^m t_j$ (minimiziraj vsoto ostankov, ki štejejo kot izguba)

s pogojem, da

(2) $\sum_{i=1}^n (s_i * x_{ij}) + \delta_j = d_j (1 - y_j) \quad \forall j$ (nahrbtnik omejitev)

(3) $\sum_{j=1}^m x_{ij} = b_i \quad \forall i$ (omejitve povpraševanja – število potrebnih kosov je fiksno)

(4) $\delta_j - UB * u_j \geq 0 \quad \forall j$ (u_j kaže, ali je preostanek palice d_j daljši od UB)

(5) $\sum_{j=1}^m u_j \leq 1$ (največje število preostalih palic, ki ne štejejo kot izguba, saj so daljše od UB)

(6) $\delta_j - t_j - (u_j + y_j) \cdot (\max d_j) \leq 0 \quad \forall j$ (t_j kaže, kakšna je izguba pri palici d_j in je enak δ_j , razen pri neuporabljenih palicah in pri eni palici, ki ne šteje kot izguba)

(7) $UB \leq \max s_i$ ali $UB \leq \min s_i$

$x_{ij} \geq 0$, celoštevilčni $\forall i, j$

$t_j \geq 0 \quad \forall j$

$\delta_j \geq 0 \quad \forall j$

$u_j \in \{0,1\} \quad \forall j$

$y_j \in \{0,1\} \quad \forall j$

Primer 2: Naročilo zaradi pomanjkanja materiala ne more biti v celoti izpolnjeno:

(1) $\min \sum_{i=1}^n \delta_j$ (minimiziraj vsoto izgub)

s pogojem, da

(2) $\sum_{i=1}^n (s_i \cdot x_{ij}) + \delta_j = d_j \quad \forall j$ (nahrbtnik omejitev)

(3) $\sum_{j=1}^m x_{ij} \leq b_i \quad \forall i$ (omejitve povpraševanja – posamezne dolžine naročila ne smemo

narezati več, kot je povpraševanje po njej)

(4) $x_{ij} \geq 0$, celoštevilčni $i=1, \dots, n; j=1, \dots, m$
 $\delta_j \geq 0 \quad j=1, \dots, m$

V prvem primeru minimiziram vsoto ostankov, ki štejejo kot izguba. Kot že omenjeno, največ ena palica ne šteje kot izguba, v kolikor je daljša od zgornje meje. Druga omejitev določa, da vsota dolžin narezanih naročil iz ene palice ne more biti daljša od dolžine te palice, v kolikor je ta palica vključena v načrt razreza ($y_j=0$), oziroma mora biti enaka 0, v kolikor te palice ni v načrtu razreza ($y_j=1$). Razlika med dolžino palice in vsoto dolžin narezanih naročil je enaka ostanku pri tej palici.

Tretja omejitev določa, da mora biti število narezanih kosov natančno enako potrebam – morebitno rezanje večjega števila kosov (torej prevelika proizvodnja), kot je zahtevano, ni dopustno. V četrti omejitvi preverjam, ali ta palica šteje kot izguba ali ne. u_j je lahko enak 1, le v kolikor je ostanek pri tej palici daljši od določene UB , za vse ostale palice je enak 0.

S peto omejitvijo navzgor omejim število palic, ki so vključene v načrt razreza, njihov ostanek pa ne šteje kot izguba. Število takih palic je lahko največ 1. Brez te omejitve bi namreč prišlo do takih načrtov razreza, kjer bi iz vsake palice izrezali le nekaj naročil, ves preostanek pa vrnili v skladišče. S tem bi sicer načeloma zelo enostavno prišli do načrtov razreza z izgubo 0, po drugi strani bi to seveda povzročilo povečevanje števila čedalje krajših palic v skladišču, posledično pa višanje stroškov skladiščenja (zaradi krajših dolžin palic ob vedno večjem številu kosov) in

predvsem večanje izgub pri kasnejših načrtih razreza. Zato je ta omejitev ob preučevanju razreza v enem obdobju nujna.

S šesto omejitvijo določim razmerje med δ_j (ostankom posamezne palice) in t_j (izgubo pri posamezni palici). Izguba je enaka ostanku pri vseh palicah razen pri tistih, ki sploh niso bile uporabljene v načrtu razreza (torej so ostale nenarezane in imajo $y_j=1$) in pri največ eni palici, ki ne šteje kot izguba (torej ima $u_j=1$).

V sedmi omejitvi določim še, kakšna naj bo UB (več o tem še v nadaljevanju) ter zahtevam celoštevilskost nekaterih spremenljivk. Pogoj, da sta celoštevilčni tudi spremenljivki δ_j in t_j , ni potreben, saj to zagotavljata že omejitvi 2 oziroma 6 – ker so namreč celoštevilske vse preostale spremenljivke v tej omejitvi, bosta tako posledično celoštevilski tudi ti dve.

Pri 6. omejitvi je v neenačbi vključen tudi $\max d_j$. Namesto tega bi lahko v neenačbo vključil tudi katero koli število N , ki bi za vsak j zagotovilo, da je vrednost $\delta_j - N \leq 0$. S tem namreč zagotovimo, da je, v kolikor je, ali u_j ali y_j enako 1, vrednost 6. omejitve negativna, tudi v kolikor je t_j enako 0. Glede na to da je δ_j lahko največ enaka vrednosti $\max d_j$ (to je največja dolžina palice na zalogi), sem to vrednost vključili v neenačbo. Brez spremembe vsebine, načina reševanja ali rešitve modela pa bi lahko vključil tudi poljubno večjo vrednost.

V kolikor imata u_j ali y_j vrednost 1, torej že ta del neenačbe zagotavlja, da bo vrednost manjša od 0. Zato lahko t_j zavzame poljubno vrednost. Ker je v tem primeru s stališča kriterijske funkcije optimalna možnost $t_j=0$, bo ta spremenljivka torej vedno zavzela to vrednost. Podobno lahko razložimo tudi četrto omejitev.

Pri izbiri, kateri model bomo uporabili za reševanje posameznega problema, je manjša težava, da je pogoj $\sum_{j=1}^m d_j \geq (\sum_{i=1}^n s_i * b_i)$ potreben, ne pa tudi zadostni pogoj za izpolnitev celotnega naročila.

V mejnih primerih, kjer celotni material le minimalno presega celotne potrebe, je namreč možno, da naročila ne bo možno izpolniti, saj lahko izguba pri optimalni rešitvi presega razliko med razpoložljivim materialom in potrebami.

To je težko predvideti brez pridobitve optimalne rešitve, saj je tudi problem, ali je celotni material dovolj za izpolnitev celotnih potreb, verjetno NP težek in odgovora na to ne moremo najti z algoritmom s polinomske časovno zahtevnostjo. Zato moramo v takih primerih najprej poizkusiti rešiti model 1. V kolikor je ta model nerešljiv, rešimo še model 2. To je zelo redek primer, saj so izgube (kot je razvidno iz naslednjih poglavij) majhne in le redko presegajo 1% porabljenega materiala. V praksi pri več tisoč rešenih problemih pri pripravi doktorske disertacije ni prišlo do tega problema niti enkrat, vendar je potrebno na to vseeno opozoriti.

Zelo pomembno vprašanje je, kako ustrezno postaviti zgornjo mejo. Seveda se to vprašanje pojavlja le v primeru 1 (torej ko imamo dovolj materiala), medtem ko v primeru 2 (problemi s

premalo materiala), zgornja meja sploh ni postavljena, saj moramo porabiti celoten material. Običajno mora biti ta meja postavljena nekje med $\min s_i$ in $\max s_i$. Z višanjem te meje se nekoliko otežuje tudi problem razreza in pri optimalni rešitvi bo tako verjetno prišlo do nekaj višje izgube. V praksi se običajno uporablja omejitev $UB = \min s_i$ (Gradišar et al., 1997).

Nadaljnje pomembno vprašanje pa je, kakšno je pravzaprav število palic, ki presegajo zgornjo mejo. V (Gradišar et al., 1999) je postavljen pogoj, da lahko dolžino zgornje meje presega največ en ostanek, vsi ostali ostanki pa morajo biti krajši.

Model v tej doktorski disertaciji se v tem delu nekoliko razlikuje, saj z omejitvama 5 in 6 namreč ne zahteva, da je število palic, kjer je ostanek daljši od zgornje meje, večje ali kvečjemu enako 1. Model omejuje le to, da je število takih ostankov daljših od zgornje meje, ki so vrnjeni v skladišče in ne štejejo kot izguba, največ 1. Število ostankov daljših od UB je torej lahko tudi večje kot 1, vendar v tem primeru le eden od njih (to je seveda najdaljši zaradi minimizacije celotne izgube) ne šteje kot izguba. Eno palico torej vrnemo v skladišče, preostale pa zavržemo.

Ta razlika je majhna in vpliva le v izjemnih primerih, ko je spodnja meja postavljena na manj kot $\max s_i$. V takih primerih pa je opredelitev v tej disertaciji bistveno ustrežnejša. V kolikor je zgornja meja namreč postavljena na $\max s_i$, lahko enostavno dokažemo, da je pri optimalni rešitvi največje število narezanih palic, kjer je ostanek daljši od zgornje meje, enako 1. Predpostavimo, da je A rešitev problema, pri kateri sta dva ostanka pri narezanih palicah (pri palicah z indeksom k in p) daljša od zgornje meje, eden ne šteje kot izguba (to je daljši ostanek), drugi pa je vključen v izgubo. V tem primeru velja:

$$y_k = 0, u_k = 0, t_k = \delta_k$$

$$y_p = 0, u_p = 1, t_p = 0$$

$$\delta_k > \max s_i$$

$$\delta_p > \max s_i$$

$$\delta_p > \delta_k$$

$$UB < \max s_i$$

$$\text{skupna izguba} = \delta_k$$

Sedaj zagotovo obstaja tak x_{ip} ($i = 1, \dots, n$); $x_{ip} > 0$, da pri rešitvi B velja:

$$x_{ipB} = x_{ip} - 1$$

$$x_{ikB} = x_{ik} + 1$$

$$\delta_{pB} = \delta_p + s_i$$

$$\delta_{kB} = \delta_k - s_i; \quad \delta_{kB} > 0$$

$$\text{skupna izguba B} = \text{skupna izguba A} - s_i$$

Povedano drugače – v tem primeru lahko v načrtu razreza en kos naročila zagotovo prenesemo iz palice z daljšim ostankom na palico s krajšim (to je gotovo, saj vidimo, da je tudi ta ostanek daljši od najdaljše dolžine naročila). Tako dosežemo, da je palica, ki se vrne v skladišče daljša; palica, ki šteje kot izguba, pa za toliko krajša. Izguba je tako ustrezno manjša.

Iz tega dokaza jasno sledi, da nobena rešitev, pri kateri sta dva ostanka narezanih palic daljša od UB , ne more biti optimalna in je slabša od optimalne za vsaj $\min s_i$. V tem primeru sta torej pri optimalni rešitvi pogoja, da je lahko največ ena palica daljša od UB , oziroma da največ ena palica ne šteje kot izguba, ekvivalentna.

Podobno lahko dokažemo tudi to, da je nemogoče, da bi bila dva ostanka narezanih palic daljša od $\max s_i$ ne glede na to, kje v intervalu $(0, \max s_i)$ je postavljena zgornja meja.

Če je zgornja meja postavljena denimo na $\min s_i$, števila ostankov daljših od zgornje meje ne moremo vnaprej predvideti in se lahko giblje med 0 in številom narezanih palic. V kolikor bi v takem primeru pogoj zaostri tako, da bi zahtevali, da je največ ena narezana palica daljša od zgornje meje, bi se namreč lahko zgodilo, da pri določenih problemih model ne bi našel rešitve, čeprav je sam model povsem enostavno rešljiv.

To bi se denimo zgodilo pri primeru z naslednjimi podatki: dolžine palic: 1100, 1200, 1300, 1400, 1500, 1600; potrebujemo po 2 kosa naročil dolžine 810, 820, 830 in 10^{24} . Spodnja meja je torej postavljena na $\min s_i$, kar je v tem primeru 10. Pri optimalni rešitvi je izguba 2370, prav vseh šest ostankov palic pa je daljših od spodnje meje (očitno je, da za te podatke tak načrt razreza, kjer bi bil vsaj en ostanek krajši od UB , niti ni možen), le najdaljši med njimi pa ne šteje kot izguba. Če bi pogoj zaostri in zahtevali, da je le en ostanek daljši od UB , bi torej po nepotrebnem prišli do nerešljivega problema.

Iz tega sledi, da je v primerih $UB = \min s_i$ opredelitev tega pogoja v našem modelu ustrežnejša kot v (Gradišar et al., 1997), medtem ko v primeru $UB = \max s_i$ med obema opredelitvama ni razlik. V prejšnjem odstavku navedeni problem namreč po opredelitvi v (Gradišar et al., 1997) predstavlja nerešljiv problem, čeprav je problem v praksi dejansko popolnoma rešljiv. Zato v nadaljevanju disertacije uporabljam ustrežnejšo opredelitev števila ostankov nad zgornjo mejo.

Pri problemu, ki ga z različnih vidikov obravnavam v tej doktorski disertaciji, zaradi dejstva, da so vse palice na zalogi različnih dolžin, uporaba metod, ki temeljijo na vzorcih, ni možna. Zato v nadaljevanju razvijam metode, ki vsebinsko spadajo v drugo skupino. Obstoječe metode se namreč ukvarjajo predvsem s standardnim problemom razreza (angl. Standard One-Dimensional Cutting Stock Problem – SID-CSP) in ne rešujejo problemov, kjer so vse palice na zalogi lahko različnih dolžin.

²⁴ Podatki za primer so namerno prirejeni tako, da lažje pokažem zgoraj obrazloženi problem.

Primeri, kjer so vse palice lahko različnih dolžin (angl. General One-Dimensional Cutting Stock Problem – G1D-CSP), se lahko rešujejo eksaktno (na primer Trkman, Gradišar, 2002a; Gradišar et al., 2001) ali hevristično (na primer Gradišar et al., 1997; 1999). Problem eksaktne metode je, da je zaradi NP težkosti tega problema neprimerna za večje probleme. Problem hevrističnih metod je, da lahko privedejo do večjih izgub, saj tako pridobljena rešitev ni optimalna. Zato v tej disertaciji predstavim metodo, ki kombinira oba pristopa na tak način, da dobimo boljšo rešitev kot s hevrističnimi metodami in hkrati ohranimo nizko časovno kompleksnost pri reševanju.

Pred razvojem kombinirane metode pa predstavim in testiram eksaktni pristop ter predstavimo tudi obstoječe hevristične rešitve.

5 Eksaktna rešitev²⁵

5.1 Pristop razveji in omeji

Zgoraj predstavljeni model predstavlja mešani celoštevilčni problem (angl. mixed integer programming), ki se običajno rešuje s tehniko imenovano razveji in omeji, ki je podrobneje predstavljena v (Winston, 1993; Papadimitriou, Steiglitz, 1982 ter tudi v Kozak et al., 1986, od koder povzemam predvsem prevode izrazov v slovenščino).

Glede na to da je število možnih rešitev (angl. feasible point)²⁶ pri večjih problemih zelo veliko in je nemogoče preveriti vse, ta metoda temelji na pametnem izboru točk, ki jih bomo preverili. Princip tega pristopa k reševanju celoštevilskega linearnega programa je bil prvič predstavljen v (Land, Doig, 1960), kasneje pa se je še precej dodelal. Osnovna ideja je, da najprej rešimo linearno (LP) sprostitev problema²⁷ in pridobimo rešitev, pri kateri vrednosti spremenljivk običajno niso celoštevilčne. V kolikor so vrednosti celoštevilčne, je to zagotovo optimalna rešitev problema, saj velja, da je optimalna rešitev LP sprostitev, pri kateri so vse vrednosti celoštevilčne, hkrati tudi optimalna rešitev celoštevilčnega problema (Winston, 1993).

V kolikor vrednosti niso celoštevilčne, nam rešitev linearne sprostitev problema predstavlja spodnjo mejo pri minimizacijskem problemu (oziroma zgornjo mejo, če gre za maksimizacijski problem) za rešitev – zagotovo vemo, da je končna rešitev slabša²⁸ (ali v skrajnem primeru kvečjemu enaka) od te vrednosti.

²⁵ Metoda in primeri rešitev za reševanje z eksaktno metodo so povzeti in razširjeni po (Trkman, Gradišar 2002a), (Gradišar et al., 2001), (Trkman, 2002). Širši teoretični pregled delovanja pristopa razveji & omeji ter generatorjev naključnih števil za kreiranje ustreznega števila testnih problemov je dopolnitev v tej doktorski disertaciji.

²⁶ Koliko je lahko možnih vzorcev samo za eno palico pri razrezu, sem pokazal v tretjem poglavju.

²⁷ Reševanje takega problema brez celoštevilčne omejitve je seveda bistveno lažje in hitrejše; lahko se reši s katero koli od dobro znanih metod za linearno programiranje, denimo simplex (pregled različnih verzij metode je v (Morgan, 1997)), push & pull (Arsham et al., 2003) in podobno.

²⁸ Če gre za minimizacijski problem, je vrednost končne rešitve večja, pri maksimizacijskih problemih pa manjša od tako pridobljene vrednosti.

Naslednji korak je, da problem razdelimo na dva dela, tako da ga razvejimo (angl. branching) z dodajanjem nove omejitve. Izberemo eno od spremenljivk²⁹, ki v prvotni rešitvi nima celoštevilske vrednosti. Denimo, da je to spremenljivka y_I in da ima v rešitvi LP sprostitve vrednost 7,5674. V enem podproblemu dodamo omejitev $y_I \geq 8$, v drugem podproblemu pa $y_I \leq 7$. S tem smo dobili dva podproblema, ki očitno nimata nobene skupne točke (saj sta dodani omejitvi medsebojno izključujoči). Skupaj pa podproblema vključujeta vse možne celoštevilске rešitve problema, ne pa tudi optimalne rešitve LP sprostitve, ki ni celoštevilska.

Prikaz vseh kreiranih podproblemov imenujemo drevo (angl. tree), vsak posamezen rešen podproblem pa vozle (angl. node).

Sedaj izberemo enega od podproblemov in poiščemo rešitev njegove sprostitve. V splošnem imamo pri rešitvi LP sprostitve podproblema tri možnosti:

1. ta problem je nerešljiv: to pomeni, da ne obstaja nobena rešitev, ki izpolnjuje vse pogoje, vključno z dodanimi omejitvami. Očitno je, da nadaljnje razvejevanje takega vozla nima smisla, saj ne more privedi do nobene rešitve in tako seveda tudi ne do optimalne. Pravimo, da je tak vozle prečrtan (angl. fathomed, killed).
2. V rešitvi podproblema so vse vrednosti celoštevilске: dobili smo možno rešitev (angl. candidate solution) problema. V tem trenutku še ne moremo trditi, da je ta rešitev optimalna, nam pa ta vrednost predstavlja zgornjo mejo za rešitev – zagotovo vemo, da je vrednost kriterijske funkcije pri optimalni rešitvi manjša ali kvečjemu enaka, kot je ta zgornja meja (pri maksimizacijskih problemih ravno obratno). Ko najdemo tak vozle, seveda ne nadaljujemo z njegovim razvejevanjem.

Seveda velja, da če je tako pridobljena možna rešitev slabša od celoštevilске rešitve, ki smo jo že dobili prej, lahko tudi ta vozle prečrtamo, saj imamo že boljšega kandidata za končno rešitev.

3. Rešitev problema vsebuje neceloštevilске vrednosti: v tem primeru nadaljujemo z razvejevanjem tega vozla (lahko takoj ali pa prej razvejimo še katerega od preostalih nerazvejenih vozlov). V kolikor pa optimalna vrednost rešitve LP sprostitve ne presega vrednosti do sedaj dosežene možne rešitve, lahko tudi ta vozle prečrtamo, saj bodo vse rešitve, ki bi jih lahko pridobili iz tega vozla z dodajanjem novih omejitev, zagotovo slabše od trenutne rešitve tega vozla, s tem pa tudi od že pridobljene možne rešitve.

Z navedenim postopkom nadaljujemo toliko časa, dokler niso vsi vozli prečrtani. Rešitev, ki ostane takrat, je zagotovo optimalna. Osnovna prednost te ideje je, da lahko s prečrtanjem enega vozla hkrati izločimo večje število možnih točk, ki ne morejo privedi do optimalne rešitve.

²⁹ Več o tem, kako izberemo, po kateri spremenljivki bomo razvejili problem, v nadaljevanju.

Za velike probleme tudi uporaba metode razveji in omeji ne omogoča vedno, da najdemo optimalno rešitev. V tem primeru lahko obdržimo do sedaj pridobljeno rešitev (to je torej tisto, ki nam trenutno predstavlja zgornjo mejo pri minimizacijskih problemih) kot najboljšo najdeno rešitev problema. To sem dostikrat uporabili tudi pri izdelavi načrta razreza, ko ta metoda za večje probleme ni našla optimalne rešitve v dani časovni omejitvi. Tudi to, da že med potekom reševanja dobimo potencialne rešitve, je ena od prednosti te metode reševanja. Dostikrat je lahko tudi ta sicer ne optimalna rešitev dovolj dobra za praktično uporabo.

Tako metoda deluje, če so vse spremenljivke celoštevilčne. Pri mojem primeru nekatere lahko zavzamejo tudi realne vrednosti. Edina razlika je v tem, da sedaj problem razvejujemo le po tistih spremenljivkah, ki morajo nujno biti celoštevilčne, po preostalih pa seveda ne.

Pomembno vprašanje je, po katerih spremenljivkah naj razmejujemo in katere vozle naj razmejimo najprej. Jasnega teoretičnega odgovora na to vprašanje ni, v praksi pa se uporablja več različnih strategij. Pri izbiri vozlov, ki jih razmejujemo najprej, se dostikrat ravnamo po LIFO (angl. last in first out) načelu, kar pomeni, da najprej rešimo tisti podproblem, ki smo ga ustvarili nazadnje. To pomeni, da najprej obdelamo en del drevesa, pridobimo možno rešitev, nato pa nadaljujemo z reševanjem preostalega drevesa (angl. backtracking).

Druga možnost je, da vedno rešimo tisti podproblem, ki ima najboljšo vrednost kriterijske funkcije. To pomeni, da dostikrat skačemo z enega na drug dela drevesa (angl. jumptracking) – na ta način poskušamo čim prej priti do dobre rešitve, ki jo nato poskušamo še izboljšati.

V svojem primeru sem uporabljal naslednje nastavitve: za izbor spremenljivke, po kateri razmejujemo, sem programu dovolil določitev pravila za izbor spremenljivke na podlagi problema in napredka pri reševanju problema. Ravno tako je bil programu prepuščen izbor, katero od obeh vej, ki izhajajo iz posameznega vozla, bo obdelal najprej. Za razmejevanje je bil uporabljen tisti vozle, ki je imel najboljšo vrednost kriterijske funkcije pri rešitvi LP sprostitve.

Očitno je torej, da z opisano metodo lahko pridemo do optimalnih rešitev posameznega problema. Zaradi opisane NP težkosti tega problema in s tem povezanega hitrega naraščanja časa, potrebnega za pridobitev rešitve pri večanju obsega problema, je pomembno vprašanje, koliko je metoda uporabna v praksi in kakšne je približna velikost problema razreza, pri kateri jo lahko še uporabimo. Zato sem metodo obsežno eksperimentalno testiral. Ko govorim o rešitvah, ki sem jih pridobil z »eksaktno metodo«, imamo torej v mislih rešitve, najdene v dani časovni omejitvi. Še enkrat naj poudarim, da to niso nujno tudi optimalne rešitve.

Ta rešitev je lahko optimalna (metoda razveji in omeji je torej našla optimalno rešitev v krajšem času, kot je bila postavljena časovna omejitev) ali pa neka do tedaj najboljša najdena rešitev, ki ni nujno optimalna. V nadaljevanju disertacije je predstavljeno tudi obsežno eksperimentalno testiranje eksaktne metode za različne velikosti problema ob različnih časovnih omejitvah.

5.2 Rezultati z eksaktno metodo

Z eksaktno metodo sem najprej rešil problem že naveden v literaturi (Gradišar et al., 1999), kjer je tudi naveden natančen načrt razreza. Ker celotni razpoložljivi material ne zadošča za izpolnitev celotnih potreb, rešujemo model za premalo materiala (model 2), zgornja meja pa tako ni določena.

Predhodno navedena rešitev ima izgubo 2 centimetra, medtem ko ima rešitev predstavljena na sliki 6 za polovico manjšo izgubo (1 cm). Rešitev je bila najdena v 10,1 sekundah, potem ko je razveji in omeji metoda preiskala 59.467 vozlov. Proces reševanja se je izvedel na osebнем računalniku (AMD, 1300 MHz) z reševalcem MPL/CPLEX. Isti računalnik in reševalec sem uporabili tudi pri reševanju nadaljnjih problemov razreza.

Slika 6: Izboljšana rešitev problema iz (Gradišar et al., 1999)

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov
1	304	12
2	319	38
3	397	14
4	415	34
5	366	27

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
1	13535
2	11301
3	9408
4	9341

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI NAROČIL

zap. št. naročila	zap. št. palice	št. kosov
1	1	1
1	2	1
1	3	0
1	4	10
2	1	29
2	2	4
2	3	1
2	4	2
3	1	4
3	2	1
3	3	9

3	4	0
4	1	4
4	2	11
4	3	8
4	4	11
5	1	2
5	2	13
5	3	6
5	4	3

IZGUBA

zap. št. palice	izguba
1	0
2	1
3	0
4	0

REALIZACIJA

zap. št. naročila	načrt	realizacija	razlika
1	12	12	0
2	38	36	2
3	13	13	0
4	34	34	0
5	25	22	3

skupna izguba= 1 cm

Vir: Lastni izračun

Pri problemu, predstavljenem na sliki 6, gre seveda za razmeroma majhen primer s štirimi dolžinami palic in petimi dolžinami naročil, kar pomeni, da je zelo primeren za reševanje z eksaktno metodo. To se pokaže tudi v dejstvu, da smo po desetih sekundah reševanja za 50% izboljšali do sedaj poznano rešitev. Sedaj torej preostane nekaj manj skupne dolžine nenarezanih naročil.

Ob povečevanju števila spremenljivk zaradi NP težkosti problema kompleksnost problema hitro narašča, s tem pa tudi čas, potreben za rešitev. Dostikrat zato v normalnem času (npr. nekaj minutah), ni možno najti optimalne rešitve. K sreči pa razveji in omeji, kot sem navedel pri opisu te metode, deluje tako, da se postopno bliža optimumu, medtem pa že privede do rešitve, ki je lahko blizu optimalni, v vsakem primeru pa je izvedljiva (angl. feasible).

Če želim določiti približno velikost problemov, ki jih lahko rešujem z eksaktno metodo ob dani tehnologiji, potrebujem obširno eksperimentalno testiranje ob različnem številu omejitev in spremenljivk. Uporabili bom eksperimentalni pristop s statistično analizo pridobljenih rezultatov, saj bi analitični pristop zahteval natančne podatke o strukturi problemov, načinu reševanja pa

tudi o načinu in hitrosti izvajanja operacij z danim računalniškim procesorjem. Zato se skoraj vedno za testiranje primernosti metod uporablja eksperimentalni pristop.

Preden pa se lotim testiranja metode, potrebujem zadostno število podatkov, ki jih lahko uporabim kot vhodne podatke za metodo. Eden od problemov, ki se pojavljajo v literaturi pri medsebojni primerjavi različnih algoritmov in metod, je prav pomanjkanje ustreznega števila testnih problemov, saj so tako rezultati predstavljeni le na nekaj posebej izbranih primerih (primer tega lahko vidimo denimo v (Pierce, 1964), (Johnston, 1986), (Stadtler, 1990)), kar ne omogoča temeljite presoje ustreznosti metode. Zato je tudi za učinkovito testiranje metod v tej doktorski disertaciji nujno uporabiti ustrezen generator naključnih problemov, ki na podlagi podanih parametrov ustvari željeno število problemov.

V ta namen sem uporabil generator PGEN (Gradišar et al., 1999b; Gradišar et al., 2002), ki je razširjena in izboljšana različica generatorja CUTGEN1 (Gau, Wäscher, 1995). Najprej predstavim generator CUTGEN1, ki je primeren za generiranje standardnih dolžin palic in naročil. Nastavimo lahko vrednosti naslednjih parametrov, na podlagi katerih se nato tvorijo naključni podatki:

- m – velikost problema (število različnih naročil),
- L – standardna dolžina palice,
- v_1 – spodnja meja za relativno dolžino naročil v primerjavi s standardno dolžino palice $L : l_i \geq v_1 * L \ (i=1, \dots, m)$,
- v_2 – zgornja meja za relativno dolžino naročil v primerjavi s standardno dolžino palice $L : l_i \leq v_2 * L \ (i=1, \dots, m)$,
- d – povprečno naročilo posamezne dolžine naročila.

Za vsak primer s parametri (m, L, v_1, v_2, d) moram (ob dani oziroma nepomembni standardni dolžini palice, torej L) sedaj naključno generirati $2 * m$ spremenljivk (m spremenljivk za dolžine naročil in m spremenljivk za število kosov posameznega naročila).

Dolžina naročil se v tem generatorju dobi tako, da najprej izračunamo vrednost y_i , nato pa to vrednost zaokrožimo navzdol na največje celo število. Za generiranje naključnih števil na intervalu med 0 in 1 ($\text{rand}_i(0,1)$) se uporablja Lehmerjev generator naključnih števil (predstavljen v (Hutchinson, 1966) in (Park, Miller, 1988)). Torej:

$$y_i = (v_1 + (v_2 - v_1) * \text{rand}_i(0,1)) * L + \text{rand}_i(0,1)$$

Vrednost y_i nato zaokrožimo navzdol na najbližje celo število:

$$l_i = \lfloor y_i \rfloor$$

Ob takem generiranju vrednosti je sicer možno, da je več dolžin naročil medsebojno enakih in da torej dejansko nismo generirali m različnih naročil, ampak kakšno manj, kar pa ni bistveno.

Za generiranje potreb po posameznem naročilu se celotno naročilo D ($D = m * d$) razdeli na posamezna naročila po naslednjem postopku:

$$d_i = \max \{1, |\bar{d}_i + 0,5|\}$$

$$\bar{d}_i = \frac{rand_i(0,1)}{\sum_{k=1}^m rand_k(0,1)} * D \quad i=1, \dots, m-1$$

$$d_m = \max \{1, D - \sum_{k=1}^{m-1} d_k \}$$

Povpraševanja po dolžini d_i ($i=1, \dots, m-1$) so popravljena s prištevanjem 0,5, saj bi bila sicer ta povpraševanja sistematično podcenjena, s tem pa bi bila sistematično precenjena povpraševanja po d_m . Načeloma je možno, da je celotno povpraševanje $\sum_{i=1}^m d_i$ nekoliko večje od želenega D .

Vendar to ni večji problem, saj ima ta razlika zanemarljivo majhen vpliv na povprečno povpraševanje d , ki smo ga določili v nastavitvah generatorja.

Opisani generator je sicer primeren za generiranje testnih podatkov za standardni problem razreza, kjer so vse palice v zalogi enakih dolžin. Za moj primer pa v taki obliki ni uporaben, saj ne generira različnih dolžin palic v zalogi. Dejstvo, da so vse palice v zalogi različnih dolžin, pa je ena od glavnih značilnosti splošnega enodimenzionalnega problema razreza.

Zato za generiranje podatkov v nadaljevanju uporaba tega generatorja ni možna. Zato sem uporabil razširjen in izboljššan generator PGEN (Gradišar et al., 1999b; Gradišar et al., 2002). PGEN uvede 5 novih parametrov, tako da ima naslednje parametre:

- n - število različnih dolžin naročil,
- v_1, v_2 - zgornja in spodnja meja za dolžino naročil – vsa naročila so torej dolga med v_1 in v_2 ,
- d - povprečno število kosov posamezne dolžine naročila,
- p - število različnih standardnih dolžin palic v zalogi,
- s_1, s_2 - zgornja in spodnja meja za dolžino standardnih palic v zalogi, torej $s_1 \leq L_k \leq s_2$ ($k = 1, \dots, p$),
- m - število nestandardnih dolžin palic v zalogi,
- u_1, u_2 - zgornja in spodnja meja za dolžino nestandardnih palic v zalogi, torej $u_1 \leq U_j \leq u_2$ ($j = 1, \dots, m$),

- r - število generiranih primerov problema (parameter r torej pomeni, koliko problemov z istimi vrednostmi zgornjih parametrov želimo generirati – tako generirani problemi seveda niso medsebojno enaki).

Zaradi uvedbe teh parametrov je PGEN primeren tudi za probleme z različnimi dolžinami palic v skladišču, kakršen je tudi primer v tej doktorski disertaciji. Seveda pa se lahko ta generator uporablja tudi za generiranje podatkov za testiranje metod za reševanje standardnega problema razreza.

Z generatorjem PGEN je mogoče (ob uporabi istega semena) ob istih vhodnih parametrih ponovno dobiti identične podatke, te podatke nato uporabiti za reševanje in na koncu primerjati rešitve, pridobljene z različnimi metodami. Vhodni podatki se izračunajo na podlagi parametrov kot naključni vzorec enega ali več testnih problemov. Uporaba tega generatorja za medsebojno primerjavo dveh hevrističnih metod je prikazana v (Gradišar et al., 1999b; Gradišar et al., 2002). V tem poglavju disertacije ga uporabim za pridobitev zadostnega števila podatkov za ustrezno testiranje eksaktne metode, kasneje pa tudi za pridobitev vhodnih podatkov za učno in testno množico pri uporabi ene od tehnik rudarjenja po podatkih.

Uporabil sem naslednje vhodne podatke :

- določanje števila različnih dolžin naročila in povprečne potrebe po posamezni dolžini ter zgornjih in spodnjih mej za naročila:
 - število različnih dolžin naročila n zavzame vrednosti 5, 10, 15,
 - v_1 in v_2 zavzemajo naslednje vrednosti ($v_1 = 100$ in $v_2 = 200$; $v_1 = 200$ in $v_2 = 400$; $v_1 = 300$ in $v_2 = 600$),
 - povprečno potrebno število kosov posamezne dolžine zavzema vrednosti ($d = 5, 10, 15$).

S kombinacijo teh treh možnosti sem pridobil 27 problemov z različnimi parametri,

- določitev standardnih dolžin palic:
generator PGEN omogoča tudi, da določimo nekaj standardnih dolžin palic. Ker pa se v tej doktorski disertaciji ukvarjam izključno s splošnim problemom enodimenzionalnega razreza, kjer ni standardnih dolžin palic, je ta parameter postavljen na 0,
- določitev nestandardnih dolžin palic:
Število nestandardnih palic (m) se spreminja od 5 do 15 v koraku po 5, zgornja meja za dolžino palic se spreminja od 2000 do 6000, spodnja meja pa od 1000 do 3000.

Podroben prikaz generiranja testnih problemov in način določanja semena sta razvidna iz sheme, ki je prikazana na sliki 7.

Slika 7: Prikaz sheme procedure PROGEN, uporabljene za generiranje testnih problemov

Procedura PROGEN:

od $i = 1$ *do* 3

od $j = 1$ *do* 3

od $k = 1$ *do* 3

$n \leftarrow i * 5$

$v_1 \leftarrow j * 100$

$v_2 \leftarrow j * 200$

$d \leftarrow k * 5$

$m \leftarrow k * 5$

$u_1 \leftarrow k * 1000$

$u_2 \leftarrow k * 2000$

$c \leftarrow \text{int}(\frac{m}{10}) * 9 + 1$

$\text{seme} \leftarrow m + 10 * c * d + 10 * c^2 * v_2 + 1000 * c^2 * v_1 + 1000000 * n$

$r \leftarrow 10$

Ključni PGGEN ($n, v_1, v_2, d, m, u_1, u_2, \text{seme}, r$)

naslednji k

naslednji j

naslednji i

Zgoraj navedeno PROGEN proceduro sem pripravil s programskim jezikom Visual Basic for Applications (VBA). Parametri, ki sem jih dobil s programom v VBA, so prikazani v tabeli 3.

Tabela 3: Parametri za 27 testnih primerov

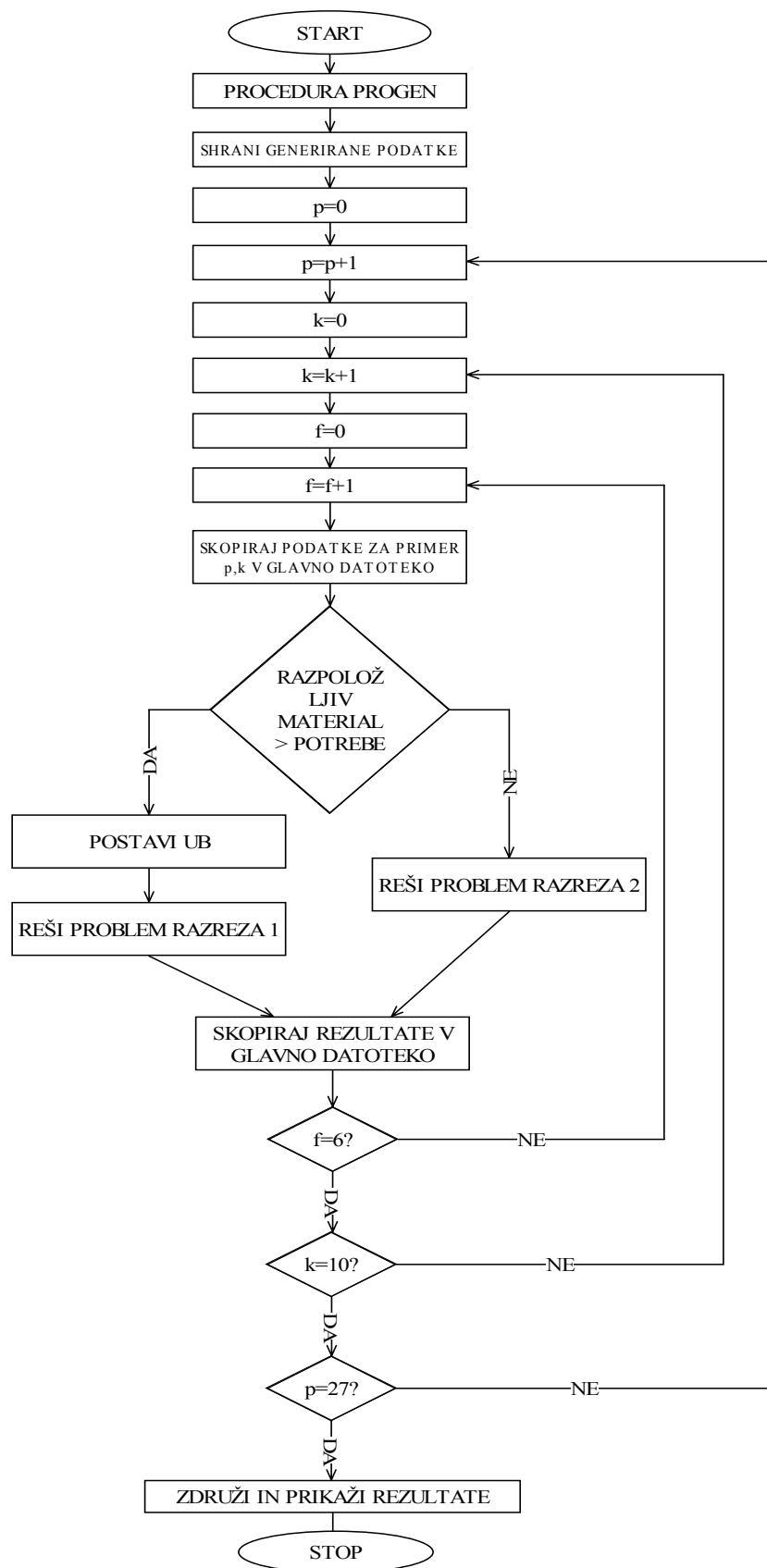
zap. št.	n	v_1	v_2	d	m	u_1	u_2	seme
1	5	100	200	5	5	1000	2000	5102055
2	5	100	200	10	10	2000	4000	510201010
3	5	100	200	15	15	3000	6000	510201515
4	5	200	400	5	5	1000	2000	5204055
5	5	200	400	10	10	2000	4000	520401010
6	5	200	400	15	15	3000	6000	520401515
7	5	300	600	5	5	1000	2000	5306055
8	5	300	600	10	10	2000	4000	530601010
9	5	300	600	15	15	3000	6000	530601515
10	10	100	200	5	5	1000	2000	10102055
11	10	100	200	10	10	2000	4000	1010201010
12	10	100	200	15	15	3000	6000	1010201515
13	10	200	400	5	5	1000	2000	10204055
14	10	200	400	10	10	2000	4000	1020401010
15	10	200	400	15	15	3000	6000	1020401515
16	10	300	600	5	5	1000	2000	10306055
17	10	300	600	10	10	2000	4000	1030601010
18	10	300	600	15	15	3000	6000	1030601515
19	15	100	200	5	5	1000	2000	15102055
20	15	100	200	10	10	2000	4000	1510201010
21	15	100	200	15	15	3000	6000	1510201515
22	15	200	400	5	5	1000	2000	15204055
23	15	200	400	10	10	2000	4000	1520401010
24	15	200	400	15	15	3000	6000	1520401515
25	15	300	600	5	5	1000	2000	15306055
26	15	300	600	10	10	2000	4000	1530601010
27	15	300	600	15	15	3000	6000	1530601515

Vir: Lasten izračun

Za vsak testni primer sem s proceduro PROGEN generiral 10 testnih primerov ($r=10$). Skupaj je torej za testiranje na razpolago 270 primerov, od tega jih je 150 takih, kjer je dovolj materiala, 120 pa takih, kjer materiala primanjkuje. Še enkrat naj poudarim, da prikaz parametrov v tabeli 2 (vključno z uporabljenim semenom) omogoča, da vsakdo z uporabo procedure PROGEN pride do istih podatkov. Pridobljene rešitve z morebitnimi novimi metodami lahko enostavno primerja z rezultati, prikazanimi v tej doktorski disertaciji.

Poleg samega testiranja metode in njene primerjave s hevrističnimi rešitvami je bil eden od ciljev tudi ugotoviti, kako spreminjanje časovne omejitve vpliva na izboljševanje pridobljene rešitve. Zato se časovni limit povečuje od 2 preko 10 do 60 sekund.

Slika 8: Diagram poteka za eksperiment



Način testiranja je prikazan z diagramom poteka na sliki 8. V diagramu je uporabljena naslednja notacija:

p = število testnih primerov ($p = 1, \dots, 27$),

k = število podprimerov, generiranih za vsak testni primer ($k = 1, \dots, 10$),

f = število različnih časovnih omejitev v eksperimentu ($f = 1, \dots, 6$).

Skupni rezultati za vseh 27 primerov so predstavljeni v tabeli 4 (ob upoštevanju $UB = \min s_i$) in tabeli 5 (ob upoštevanju $UB = \max s_i$). V tabeli je v vsaki vrstici združenih 10 primerov, ki so bili pripravljeni z istimi parametri. Za vsako časovno omejitev je navedena celotna izguba (vsota izgub v vseh desetih problemih), izguba v odstotkih in število optimalno rešenih primerov.

Pri postavitvi UB na $\min s_i$ je bila v 2 sekundah najdena optimalna rešitev za 57 primerov, v 60 sekundah pa za 110 primerov. Izguba po 60 sekundah niha od 0 do 2,4%, pri čemer je potrebno poudariti, da je izguba 2,4% v 7. primeru optimalna rešitev (torej tudi z uporabo kake druge metode ne bi mogli najti boljše rešitve). V 7. primeru je namreč število palic, število naročil in povprečno naročilo majhno (vse te tri spremenljivke imajo vrednost 5), kar pomeni, da imamo razmeroma majhno število možnih kombinacij. Posledica tega je, da je optimalno rešitev sicer razmeroma lahko najti, izguba pri optimalni rešitvi pa je posledično precej velika. Podobno velja tudi za primer 4, ki ima večino podproblemov rešenih optimalno, izguba pa tudi tu presega 1%.

V vseh ostalih primerih je izguba 1% ali manj, kar pomeni, da je eksaktna metoda za ta velikostni razred problema razmeroma primerna. Pri povečevanju časovnega limita za reševanje lahko vidimo, da največje zmanjšanje izgube prinese povečanje časa iz 2 na 10 sekund (zmanjšanje izgube za 35%), v nekoliko manjši meri pa tudi povečanje iz 10 na 20 sekund (izguba se zmanjša za slabih 20%). Nadaljnja povečanja razpoložljivega časa sicer privedejo do delnega zmanjšanja izgube, vendar razlike niso več tako očitne (povečanje časovnega limita za trikrat iz 20 na 60 sekund tako izgubo dodatno zmanjša le še za manj kot 15%). To je pričakovani rezultat, saj zaradi eksponentialne časovne zahtevnosti NP polnih problemov povečevanje časa za reševanje (ali denimo hitrosti procesorja in podobno), ne vodi več do bistveno boljših (ali celo optimalnih) rezultatov.

V tabeli 5, kjer je $UB = \max s_i$, so rezultati prikazani samo za primere, kjer je dovolj materiala, saj v primerih s premalo materiala UB sploh ni vključena v model in bi bili tako rezultati za te primere v obeh tabelah identični.

Zaradi strožje omejitve glede zgornje meje palice, ki ne šteje kot izguba, so izgube razumljivo nekoliko večje, sicer pa veljajo podobne ugotovitve kot za primere z nižje postavljeno zgornjo mejo. Rezultati kažejo, da višina zgornje meje ne vpliva bistveno na ustreznost opisanega pristopa.

Tabela 4: Rezultati za $UB = \min s_i$ ob različnih časovnih limitih

zap. št.	2 sekundi			10 sekund			20 sekund			30 sekund			45 sekund			60 sekund		
	izguba		opt	izguba		opt	izguba		opt	izguba		opt	izguba		opt	izguba		opt
	cm	%		cm	%		cm	%		cm	%		cm	%		cm	%	
1	1	0,0027%	9	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10
2	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10
3	3	0,0026%	8	1	0,0009%	9	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10
4	1301	1,6761%	5	1028	1,3445%	9	1028	1,3445%	9	1028	1,3445%	9	1028	1,3445%	9	1028	1,3445%	9
5	831	0,4978%	0	351	0,2119%	1	295	0,1759%	2	255	0,1495%	2	228	0,1327%	2	212	0,1229%	2
6	852	0,3552%	0	295	0,1240%	0	83	0,0364%	1	69	0,0301%	1	62	0,0269%	2	62	0,0269%	2
7	1702	2,3912%	9	1702	2,3912%	10	1702	2,3912%	10	1702	2,3912%	10	1702	2,3912%	10	1702	2,3912%	10
8	2884	1,3326%	0	2325	1,0791%	0	1755	0,8112%	0	1541	0,7116%	0	1519	0,6996%	0	1457	0,6706%	0
9	4103	1,1964%	0	3045	0,8837%	0	2555	0,7473%	0	2201	0,6467%	0	1911	0,5595%	0	1780	0,5196%	0
10	157	0,2295%	1	53	0,0730%	2	27	0,0383%	3	27	0,0383%	3	22	0,0305%	4	18	0,0246%	5
11	96	0,0671%	2	25	0,0164%	4	10	0,0065%	6	10	0,0065%	6	8	0,0052%	6	8	0,0052%	6
12	269	0,1189%	0	83	0,0372%	7	64	0,0285%	7	39	0,0175%	8	27	0,0122%	8	23	0,0103%	8
13	99	0,1245%	2	41	0,0515%	4	33	0,0419%	4	17	0,0219%	5	9	0,0113%	6	8	0,0100%	6
14	3022	1,0156%	0	2048	0,6839%	0	1703	0,5687%	0	1666	0,5572%	0	1666	0,5572%	0	1613	0,5393%	0
15	6652	1,4417%	0	2753	0,5913%	0	2565	0,5483%	0	2502	0,5334%	0	2324	0,4923%	0	1833	0,3866%	0
16	266	0,3686%	1	114	0,1569%	5	49	0,0681%	8	49	0,0681%	9	49	0,0681%	9	49	0,0681%	9
17	4011	1,3846%	0	3013	1,0466%	0	2630	0,9136%	0	2580	0,8953%	0	2272	0,7893%	0	2074	0,7236%	0
18	9599	1,6617%	0	7936	1,2251%	0	7041	1,0860%	0	6533	1,0099%	0	6533	1,0099%	0	6533	1,0099%	0
19	29	0,0402%	5	10	0,0142%	9	5	0,0071%	9	3	0,0043%	9	0	0,0000%	10	0	0,0000%	10
20	1179	0,5150%	0	756	0,3203%	0	638	0,2707%	0	557	0,2361%	0	434	0,1827%	1	429	0,1803%	1
21	7753	2,1027%	0	1143	0,3401%	0	600	0,1691%	0	568	0,1599%	0	444	0,1235%	1	438	0,1217%	1
22	141	0,1831%	4	37	0,0477%	5	21	0,0275%	7	4	0,0051%	8	1	0,0013%	9	1	0,0013%	9
23	2580	0,8612%	0	1941	0,6456%	0	1551	0,5126%	0	1357	0,4517%	0	1071	0,3559%	0	1043	0,3468%	0
24	16250	3,0632%	0	13977	2,3567%	0	7639	1,1572%	0	7101	1,0736%	0	6856	1,0387%	0	5615	0,8531%	0
25	427	0,5728%	1	113	0,1457%	2	93	0,1197%	2	93	0,1197%	2	48	0,0636%	2	41	0,0549%	3
26	2037	0,6872%	0	1319	0,4446%	0	1054	0,3547%	0	1054	0,3547%	0	993	0,3325%	0	984	0,3294%	0
27	11150	1,6180%	0	7518	1,0954%	0	5984	0,8730%	0	5984	0,8730%	0	5859	0,8553%	0	5316	0,7742%	0
SKUPAJ	77394	0,8707%	57	51628	0,5678%	87	39126	0,456%	98	36941	0,4334%	102	35067	0,4106%	109	32268	0,3895%	110

Tabela 5: Rezultati za $UB = \max s_i$ ob različnih časovnih limitih

	2 sekundi			10 sekund			20 sekund			30 sekund			45 sekund			60 sekund		
Zap. št.	izguba		opt	izguba		opt	izguba		opt	izguba		opt	izguba		opt	izguba		opt
	cm	%		cm	%		cm	%		cm	%		cm	%		cm	%	
1	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10	1	0,0027%	10
2	2	0,0026%	9	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10
3	1	0,0008%	9	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10	0	0,0000%	10
4	1842	2,4094%	6	1799	2,3524%	8	1799	2,3524%	8	1799	2,3524%	8	1799	2,3524%	9	1799	2,3524%	9
5	1299	0,7972%	0	580	0,3257%	0	531	0,2971%	1	508	0,2833%	1	442	0,2466%	1	313	0,1754%	1
6	720	0,3041%	0	306	0,1248%	0	305	0,1242%	0	294	0,1198%	1	197	0,0808%	1	162	0,0663%	1
8	3594	1,6466%	0	2112	0,9747%	0	1757	0,8068%	0	1501	0,7016%	0	1427	0,6638%	0	1369	0,6364%	0
9	5916	1,6859%	0	4596	1,3164%	0	3629	1,0453%	0	3601	1,0354%	0	3353	0,9612%	0	2601	0,7537%	0
10	106	0,1522%	2	53	0,0748%	2	24	0,0355%	5	24	0,0355%	5	19	0,0277%	6	17	0,0248%	6
11	173	0,1208%	1	65	0,0461%	4	47	0,0333%	5	47	0,0333%	5	18	0,0127%	6	17	0,0120%	6
12	253	0,1095%	2	72	0,0306%	5	68	0,0288%	6	51	0,0213%	7	30	0,0124%	8	0	0,0000%	10
14	2905	0,9699%	0	2175	0,7247%	0	1929	0,6437%	0	1892	0,6321%	0	1878	0,6276%	0	1722	0,5763%	0
15	13879	2,9446%	0	6611	1,4494%	0	5709	1,2615%	0	4543	1,0074%	0	4528	1,0043%	0	2869	0,6296%	0
18	11526	2,0006%	0	7308	1,2592%	0	6963	1,2005%	0	6561	1,1339%	0	6484	1,1194%	0	6444	1,1129%	0
20	7933	2,8766%	0	7282	2,5875%	0	7205	2,5529%	0	642	0,2803%	0	361	0,1551%	1	337	0,1454%	1
21	5344	1,5468%	0	4708	1,3642%	0	4496	1,3068%	1	4469	1,2991%	1	889	0,2529%	1	867	0,2457%	2
24	9685	2,3776%	0	4210	0,9170%	0	4200	0,9149%	0	4101	0,8926%	0	9801	1,8811%	0	9619	1,8464%	0
SKUPAJ	74007	0,83%	60	48736	0,5360%	82	44654	0,5204%	94	35746	0,4194%	99	36331	0,4254%	107	33014	0,3985%	110

Vir: (Trkman, Gradišar, 2002)

Izguba pri manjših primerih (to so denimo primeri z zaporedno številko 1, 2, 3) je tudi pri višji zgornji meji zanemarljivo majhna, za vse te primere pa najkasneje v 10 sekundah najdemo optimalno rešitev. Pri večjih problemih izguba hitro narašča in tudi povečevanje časovne omejitve (denimo v primeru 24) dostikrat ne privede do bistvene izboljšave rešitev.

To je v skladu z ugotovitvami o NP polnih problemih, navedenimi v tretjem poglavju doktorske disertacije. Čeprav so izgube tudi za večje probleme, gledano v odstotkih celotnega materiala, razmeroma majhne, je v tem primeru smiselno poskusiti najti še boljšo rešitev s kakšno drugo metodo. Metoda za reševanje večjih problemov, ki kombinira hevristično in eksaktno metodo, je predstavljena v nadaljevanju disertacije.

6 Hevristične rešitve

6.1 Obstoječe hevristične rešitve

Poleg predstavljene eksaktne rešitve obstajata tudi dve hevristični metodi za rešitev istega problema in sicer COLA (Gradišar, Jesenko, 1996; Gradišar et al., 1997), ki je bila v osnovi razvita za optimizacijo razreza v tekstilni industriji, vendar se lahko uporablja tudi za razrez v drugih panogah, ter CUT (Gradišar et al., 1999). Glede na to da je CUT razširjena in izboljšana različica COLA-e in da je v (Gradišar et al., 2002) pokazano, da je ta metoda boljša³⁰, to metodo predstavljamo nekoliko obširneje. COLA deluje po zelo podobnem principu, vendar je nekoliko enostavnejša, zato tudi nekoliko hitrejša, vendar običajno privede do načrta razreza z večjo izgubo.

Pri CUT-u gre za tako imenovano zaporedno hevristično proceduro (angl. Sequential Heuristic Procedure – SHP), ki deluje tako, da minimizira vpliv končnih pogojev (angl. ending conditions). Ti so namreč pri hevrističnih procedurah dostikrat problem – prav zadnje faze priprave načrta razreza lahko namreč prispevajo večji del k celotni izgubi, če v predhodnih fazah procedure ni zagotovljeno, da bo tudi za zadnje faze preostalo dovolj različnih, potencialno dobrih kombinacij. Brez teh »popravkov« bi se namreč vsaka procedura obnašala požrešno in minimizirala predvsem izgubo v prvih fazah, kar pa bi vodilo do nesorazmerno večjih izgub v naslednjih fazah.

Sama definicija problema je enaka kot pri eksaktni metodi z izjemo treh manjših razlik. Prva je ta, da je pri formulaciji problema za hevristično metodo (kadar je dovolj materiala) postavljen pogoj, da je največ ena palica daljša od zgornje meje, pri eksaktni pa, da največ ena palica ne šteje kot izguba. V doktorski disertaciji sem že dokazal, da sta formulaciji za

³⁰ CUT preizkusi vse različice končnih rešitev, ki jih je že preizkusila COLA, poleg tega pa tudi precejšnje število dodatnih »zanimivih« kombinacij, ki pogosto privedejo do še boljše rešitve. Časovna kompleksnost se ob tem ne poveča bistveno, natančnejša analiza pa je v že omenjenem (Gradišar et al., 2002)

obe metodi večinoma enakovredni, razen v nekaterih skrajnih primerih, ko je formulacija za eksaktno metodo ustreznejša, saj drugačna formulacija ne privede do rešitve.

Druga razlika je vidna le v primeru, da imamo premalo materiala. Obe hevristični metodi v skladu z modelom minimizirata $\sum_{i=1}^n \Delta_i * s_i$ (torej vsoto nenarezanih naročil), medtem ko eksaktna rešitev, predstavljena v tej doktorski disertaciji, minimizira $\sum_{j=1}^m \delta_j$ (vsoto ostankov po posameznih palicah). Poleg tega CUT dejansko minimizira $\sum_{i=1}^n \Delta_i * s_i$ preko minimiziranja $\sum_{j=1}^m t_j$, zato moramo v teh primerih UB nujno postaviti na $\max s_i$, saj le tako dosežemo pravilno delovanje algoritma (Gradišar et al., 1999). Čeprav sta kriterijski funkciji formulirani različno, sta dejansko enakovredni, saj velja:

$$\sum_{i=1}^n (\Delta_i * s_i) + \sum_{i=1}^n \sum_{j=1}^m x_{ij} * s_j = \sum_{i=1}^n s_i * b_i \quad (1)$$

$$\sum_{j=1}^m \delta_j + \sum_{i=1}^n \sum_{j=1}^m x_{ij} * s_j = \sum_{j=1}^m d_j \quad (2)$$

Prva enačba pomeni, da je vsota dolžin nenarezanih in narezanih kosov enaka vsoti vseh kosov (to je očitno, saj je vsak kos lahko, ali narezan ali nenarezan). Druga enačba pomeni, da je vsota dolžin vseh ostankov in narezanih kosov enaka vsoti dolžin vseh palic (tudi ta enačba v primerih s premalo materiala velja, saj smo v načrtu razreza res uporabili vse palice).

Če sedaj enačbo (1) odštejemo od enačbe (2), dobimo:

$$\sum_{j=1}^m \delta_j - \sum_{i=1}^n \Delta_i * s_i = \sum_{j=1}^m d_j - \sum_{i=1}^n s_i * b_i$$

Ker je desna stran te enačbe konstanta, smo torej dokazali, da je minimiziranje $\sum_{j=1}^m \delta_j$ (kar minimizira kriterijska funkcija pri eksaktni metodi), ekvivalentno minimiziranju $\Delta_i * s_i$ (to minimizira kriterijska funkcija pri hevristični metodi).

Tretja razlika je, da hevristični metodi omogočata še vključitev parametra Y , ki določa največje število različnih dolžin naročil, ki jih lahko izrežemo iz ene palice. V modelu za eksaktno rešitev problema tega parametra nismo vključili, po potrebi pa bi seveda bila možna vključitev tudi tega ali drugih dodatnih pogojev.

Hevristična metoda temelji na postopnem izgrajevanju načrta razreza in na zaporednem obravnavanju palic. Na začetku vse dolžine palic pripadajo skupini nenarezanih palic, skupina narezanih palic je prazna. Nato se v vsakem koraku skupina nenarezanih palic zmanjša za 1, skupina narezanih palic pa poveča za 1 (v vsakem koraku torej poiščemo načrt razreza za eno palico). Ustrezno popravimo tudi število potrebnih naročil posameznih dolžin. Algoritem je sestavljen iz treh korakov:

1. izberi dolžine naročil,
2. izberi palico in jo razreži na izbrane dolžine naročil,
3. ponavlaj ta dva koraka, dokler niso narezane vse palice ali izpolnjena vsa naročila.

Pri izbiri naročil in palic se ravnamo po predpostavkah (te predpostavke so se v praksi pokazale za pravilne, matematično pa niso dokazane), da je lažje najti dobro rešitev, če:

1. lahko izbiramo iz čim večjega števila možnih rešitev. Število možnih rešitev je večje, če je:
 - a. na razpolago čim večje število različnih dolžin naročil, ki še niso bila izpolnjena v celoti,
 - b. razmerje med povprečno dolžino palice in povprečno dolžino naročila je čim večje,
 - c. število nenarezanih dolžin naročil je čim večje;
2. razlike med posameznimi možnimi rešitvami so čim večje. To velja, če:
 - a. je razmerje med najdaljšo in najkrajšo dolžino naročila čim večje,
 - b. je razmerje med najdaljšo in najkrajšo dolžino palice na zalogi čim večje.

Omenjene predpostavke so statistično dokazane (Gradišar et al., 1999), vendar ni nujno, da veljajo v vsakem primeru. Z upoštevanjem predpostavk dosežemo, da lahko tudi v zadnjih korakih algoritma še vedno izbiramo med razmeroma velikim številom možnih rešitev, kar privede do manjše izgube materiala. To je predvsem pomembno v primerih s premalo materiala, saj imamo v primerih, kjer je materiala dovolj v zadnjih korakih nekaj več svobode, saj nam tudi na koncu še ostane nekaj nenarezanih palic. Na ta način torej zmanjšamo vpliv končnih pogojev.

Najpomembnejše vprašanje je način izbora dolžin naročil. Načeloma lahko uporabimo 2 proceduri:

1. v vsakem koraku najprej izberemo tista naročila, ki imajo največje število še nenarezanih kosov naročil – tako izberemo prvih Y naročil, urejenih po padajočem številu nenarezanih kosov (Δ_i). Na ta način bo na koncu ostalo več različnih naročil, vendar bodo ostala na koncu tudi najdaljša naročila, kar je v nasprotju s predpostavko 1.b. in 1.c.,

2. v vsakem koraku najprej izberemo najdaljša naročila – izberemo torej prvih Y naročil, urejenih po padajoči dolžini (s_i).

Prvi pristop je primeren predvsem pri majhnih razlikah med najdaljšo in najkrajšo dolžino naročila, drugi pa pri velikih razlikah.

V hevristični proceduri zato kombiniramo oba pristopa in del naročil izberemo po prvem, del naročil pa po drugem načinu. Uporabljamo naslednje kombinacije:

- f naročil izberemo po proceduri 1, $Y-f$ po proceduri 2 ($f = 0, \dots, Y$),
- f naročil izberemo po proceduri 2, $Y-f$ po proceduri 1 ($f = 1, \dots, Y-1$),
- naročila lahko izbiramo tudi po padajoči vrednosti produkta $s_i * \Delta_i$.

Vsega skupaj torej imamo $2*Y+1$ možnih kombinacij za izbor³¹. Hevristična procedura CUT preizkusi vse kombinacije in izbere tisto, ki privede do najboljše končne rešitve.

Pri izboru palice, ki bo razrezana v vsakem koraku, hevristična procedura deluje tako, da izbere tisto, ki ima od preostalih palic pridobljen načrt razreza z najmanjšo izgubo. V primeru, da ima več palic enako izgubo, procedura (v skladu s predpostavko 1.b.) izbere najkrajšo palico s tako izgubo. To doseže tako, da najprej poskuša pripraviti načrt razreza za kratke palice (palice torej uredi naraščajoče po dolžini). Ker je verjetno, da bodo boljše rešitve dosežene pri daljših palicah, bi to privedlo do tega, da bi za kasnejše faze preostajale predvsem krajše palice. Za zmanjšanje tega vpliva se lahko določi parameter r – to je tista izguba, ki se pri posamezni palici lahko zanemari. Vse rešitve za posamezno palico z izgubo manjšo od r se torej štejejo kot enakovredne. Večji kot je r , bolj bodo kratke palice prišle v poštev na začetku algoritma. Posledično bo izguba večja na začetku, zato pa manjša na koncu. Omenjeni algoritem je napisan v programskem jeziku FORTRAN in vsebuje 2500 vrstic kode.

V (Gradišar et al., 1999) je tudi statistična analiza dejavnikov, ki vplivajo na kvaliteto rešitve. Ugotovljeno je, da je rešitev tem boljša, čim večje so razlike med posameznimi dolžinami naročil in čim večje je povprečno število kosov, izrezanih iz ene palice na zalogi. Važno je torej predvsem čim večje razmerje med povprečno dolžino palic in povprečno dolžino naročil. Omenjena spoznanja koristno uporabim kasneje za izbor velikosti in sestave podproblema pri kombinirani metodi.

6.2 Primerjava eksaktne in hevristične rešitve

Ker eksaktna in hevristične metode rešujejo isti problem, je seveda ključno vprašanje, katera metoda je boljša za posamezno velikost problema. Boljša je seveda tista metoda, ki

³¹ To je tudi ena od pglavitnih prednosti algoritma CUT pred COLA. COLA namreč preizkusi samo 3 kombinacije za sortiranje in izbor naročil, in sicer sortiranje po preostalih potrebah, sortiranje po dolžini ter sortiranje po produktu dolžine in preostalih potreb. Ne preizkusi pa tudi vmesnih kombinacij (torej izbor dela naročil po preostalih potrebah, dela naročil pa po dolžini).

vodi do rešitev, ki je bližje optimalni. Praviloma so hevristične rešitve primernejše za večje probleme, eksaktne za manjše. Zanima pa me, kje je meja med »velikimi« in »majhnimi« problemi.

Pri tem je potrebno posebej poudariti različno razumevanje pojma »dobra rešitev« med omenjenimi predhodnimi članki (Gradišar et al., 1997, Gradišar et al., 1999, Gradišar et al. 2002) in pričujočo doktorsko disertacijo. V predhodnih člankih se kot dobra rešitev šteje tista, ki ima majhno odstotno izgubo materiala³², v tej doktorski disertaciji pa kot dobre rešitve problema vzamemo tiste, ki so blizu optimalni rešitvi. Tako je denimo po tem kriteriju primer 4 iz tabele 4 oziroma 5 dobro rešen, čeprav ima daleč največjo izgubo od vseh primerov, ker je kar devet od desetih podproblemov tega problema rešeno optimalno.

Primerjava med rezultati, pridobljenimi z eksaktno metodo (ob upoštevanju $UB = \min s_i$ in časovne omejitve ena minuta), in metodo CUT je predstavljena v tabeli 6.

Tabela 6: Primerjava rezultatov doseženih z eksaktno metodo in rezultatov CUT

zap. št.	dovolj materiala?	izguba CUT		izguba eksaktna metoda	
		cm	%	cm	%
1	D	8	0,0213%	1	0,0027%
2	D	0	0,0000%	0	0,0000%
3	D	0	0,0000%	0	0,0000%
4	D/N	1182	1,5460%	1028	1,3445%
5	D	28	0,0162%	212	0,1229%
6	D	9	0,0039%	62	0,0269%
7	N	1940	2,7256%	1702	2,3912%
8	D	213	0,0980%	1457	0,6706%
9	D	285	0,0832%	1780	0,5196%
10	D/N	59	0,0807%	18	0,0246%
11	D	0	0,0000%	8	0,0052%
12	D	2	0,0009%	23	0,0103%
13	N	88	0,1103%	8	0,0100%
14	D/N	172	0,0575%	1613	0,5393%
15	D	22	0,0046%	1833	0,3866%
16	N	227	0,3155%	49	0,0681%
17	N	272	0,0949%	2074	0,7236%
18	D/N	541	0,0836%	6533	1,0099%
19	N	7	0,0095%	0	0,0000%
20	D	10	0,0042%	429	0,1803%
21	D	0	0,0000%	438	0,1217%
22	N	47	0,0618%	1	0,0013%

³² Glede na to da procedura CUT ne daje nobene indikacije, ali je dobljena rešitev optimalna ali ne, v takem primeru drugačna opredelitev pojma »dobra rešitev« niti ni možna. Možnost dokaza, da je rešitev optimalna, je tudi ena od prednosti tukaj predstavljene eksaktne metode.

23	N	36	0,0120%	1043	0,3468%
24	D	159	0,0242%	5615	0,8531%
25	N	81	0,1085%	41	0,0549%
26	N	93	0,0311%	984	0,3294%
27	N	112	0,0163%	5316	0,7742%
SKUPAJ		5593	0,0675%	32268	0,3895%

Vir: Trkman, Gradišar, 2003a.

Tako kot v prejšnjih tabelah je v eni vrstici združenih 10 primerov z istimi parametri. V drugi vrstici je označeno, ali gre za primere z dovolj materiala (model 1, oznaka D) ali za primere s premalo materiala (model 2, oznaka N). Oznaka D/N pomeni, da je v nekaterih podprimerih dovolj materiala, v drugih pa ne – do tega pride v primerih, ko je majhna razlika med povprečnim skupnim materialom in povprečnimi skupnimi potrebami.

Skupno je eksaktna metoda našla boljšo rešitev v 64 primerih, hevristična metoda CUT pa v 139, medtem ko sta obe metodi v 67 primerih našli enakovredno rešitev. To pomeni rešitev z enako izgubo materiala, ne pa nujno tudi identičnega načrta razreza. V skladu s pričakovanji daje eksaktna metoda boljše rezultate za manjše primere (v primerih, kjer je $d \leq 5$ in $m \leq 5$, eksaktna metoda najde boljšo rešitev tudi ob zelo kratkih časovnih omejitvah), CUT pa v večjih primerih ($d \geq 10$ in $m \geq 10$).

Skupna izguba v vseh 270 primerih je precej manjša pri uporabi CUT, saj znaša 5.593 cm, medtem ko pri eksaktni metodi znaša 32.268 cm. Vendar pa ima eksaktna metoda (poleg že omenjene primernosti za manjše probleme) nekatere prednosti. Ena od njih je na primer ta, da CUT vedno privede do enakega rezultata, medtem ko bi lahko rešitve, pridobljene z eksaktno metodo, izboljšali s povečanjem časovnega limita ali procesorske moči računalnika. Pri tem je potrebno poudariti, da zaradi NP težkosti problema pri večjih problemih to povečanje ne bi privedlo do bistvene izboljšave rezultatov ali celo do optimalne rešitve.

Druga prednost eksaktne metode je, da daje približen odgovor na to, kako daleč je pridobljena rešitev od optimalne, medtem ko CUT na to vprašanje ne odgovarja. Tako imata denimo primera 4 in 18 podobno odstotno izgubo materiala po 10 sekundah reševanja z eksaktno metodo. Glede na to da je v primeru 4 optimalno zagotovo rešeno že 9 od 10 primerov, je jasno, da kakršnokoli povečanje časovnega limita ali morebitna izboljšava metode ne moreta privedi do bistveno boljših rezultatov, saj se lahko izboljša rešitev le v enem podprimeru.

V primeru 18 v 10 sekundah ni optimalno rešen noben primer. Torej bi lahko s povečanjem časa ali morebiti z drugo metodo prišli do boljših rezultatov. Dejansko se v primeru 4 ob povečanju časa iz 10 na 60 sekund skupna izguba ne spremeni (ostane pri 1028 cm), medtem ko se v primeru 18 zmanjša iz 7936 cm na 6533 cm. Pri hevristični metodi pa ne moremo ugotoviti razlike med pridobljeno in dejansko optimalno rešitvijo ali števila

optimalno rešenih podproblemov, saj lahko z gotovostjo o optimalni rešitvi govorimo le, kadar je skupna izguba enaka 0.

6.2.1 Izbor metode³³

Ker obstaja več metod za rešitev istega problema, se seveda pojavi vprašanje, katero metodo izbrati za posamezen problem. Glede na primerjavo, predstavljeno v tabeli 6, je sicer očitno, da je eksaktna metoda primernejša za manjše probleme (kjer je denimo število palic na zalogi in naročil okoli 5 ali manj), CUT pa za večje probleme (kjer je število različnih palic in dolžin naročil večje od 10).

Vendar pa dosedanje analize niso dale odgovora na dve pomembni vprašanji:

- katera metoda naj se uporabi za primere, ki ležijo med zgoraj navedenima mejama,
- katere spremenljivke v modelu najbolj vplivajo na kompleksnost problema in s tem na verjetnost, da bo možno dobiti optimalno rešitev v vnaprej določeni časovni omejitvi.

Kot odgovor na prvo vprašanje, bi seveda lahko vsak problem rešili z obema metodama in obdržali boljši rezultat. Vendar bi za to potrebovali dodaten čas, kar je lahko problem predvsem, če želimo optimizacijo v realnem času.

Zato bi bilo dobro imeti način, kako vnaprej predvideti, katera metoda bo prinesla boljši rezultat. Za to sem v tej doktorski disertaciji uporabil eno od možnosti za rudarjenje po podatkih, in sicer metodo odločitvenih dreves (angl. decision tree). Rudarjenje po podatkih je proces iskanja zakonitosti, vzorcev ali gibanj v veliki količini podatkov, zbirki ali skladišču podatkov z uporabo namenskih orodij (ISlovar, 2005), (Thuraisingham, 1998).

S pomočjo odločitvenih dreves sem poskušal za vsak primer vnaprej (na podlagi parametrov tega primera) napovedati, ali bo eksaktna metoda v dani časovni omejitvi, na danem računalniku in ob uporabi istega programa ter metode za reševanje celoštevilskih linearnih problemov privedla do optimalnega rezultata ali ne.

6.2.2 Predstavitev odločitvenih dreves

Odločitveno drevo je grafična predstavitev procedure za klasifikacijo primerov v posamezne razrede. Ena od glavnih prednosti odločitvenih dreves je, da kompleksen odločitveni proces razbijejo na več zaporednih preprostejših odločitev (Safavian, Landgrebe, 1991). Lahko jih uporabljamo brez računalnika, razlogi, ki so privedli do odločitve, pa so natančno vidni (Lenič et al., 2002). Poleg tega jih je razmeroma lahko

³³ To podpoglavje je delno povzeto in razširjeno po (Trkman, Gradišar, 2002; Trkman, Gradišar, 2003a; Trkman 2002)

zgraditi in razumeti, vključimo lahko diskretne in zvezne spremenljivke, omogočajo pa tudi identifikacijo spremenljivk z največjim vplivom (Sherrod, 2005). Odločitvena drevesa so neparametrška, pri kreiranju ne zahtevajo vmešavanja uporabnika in so kot taka zelo primerna za preiskovalno odkrivanje znanja (angl. exploratory knowledge discovery) (Androulakis, 2005).

Zaradi omenjenih prednosti so odločitvena drevesa pogosto uporabljena v statistiki, prepoznavanju vzorcev, strojnem učenju, odločanju in pri gradnji nevronske mreže – pregled najrazličnejših možnosti uporabe je v (Murthy, 1999). Seveda odločitvena drevesa niso primerna tehnika za vse vrste problemov – pogoje, ki morajo biti izpolnjeni za njihovo uspešno uporabo, navajam v nadaljevanju.

Pri odločitvenem drevesu so učni primeri predstavljeni kot par (lastnosti atributov, odločitev). Lastnosti so zbirka več atributov, ki opisujejo posamezen primer (v mojem primeru so to denimo število palic, število naročil...). Izbiro atributov, ki jih uporabimo pri gradnji odločitvenega drevesa, moramo narediti vnaprej. Odločitev je znana pri učni množici, ne pa tudi pri objektih, o katerih bomo kasneje sprejemali odločitve (Kokol et al., 2001). V našem primeru torej za objekte v učni množici vemo, ali je eksaktna metoda pripeljala do optimalne rešitve ali ne.

Ti podatki o lastnostih in odločitvah pri posameznih učnih objektih (angl. training objects) v učni množici (angl. training set) torej predstavljajo osnovo za izgradnjo odločitvenega drevesa. Namen odločitvenega drevesa je poiskati pravila, ki določajo, v kateri razred naj bi spadal posamezen primer. Jasno je, da pri realnih primerih nobena klasifikacija na podlagi nekaj atributov ne more biti povsem natančna, lahko pa poskušamo napovedati, v kateri razred spada posamezen nov primer.

Osnovna ideja in algoritem, po katerem deluje program See5/C5³⁴, ki sem ga uporabil v tej doktorski disertaciji, je naslednja (Hunt et al., 1966; Quinlan, 1993): imamo T testnih primerov, ki pripadajo različnim razredom (razrede označimo $\{C_1, C_2, \dots, C_k\}$). Imamo tri možnosti:

- T vsebuje več primerov, ki vsi pripadajo razredu C_j . Odločitveno drevo za T je list (angl. leaf), ki kaže, da primeri v tem listu pripadajo razredu C_j ,
- T ne vsebuje nobenega primera. V tem primeru to spet predstavlja list. Razred, ki je povezan s tem listom, mora biti sedaj določen na nek drug način. Program See5 povezan razred določi na podlagi najpogostejšega razreda na predhodnem vozlu (angl. parent node),
- T vsebuje več primerov, ki pripadajo različnim razredom. V tem primeru T razdelimo na več podskupin. V tem primeru izberemo tak test A , ki na podlagi

³⁴ C5 je oznaka za različico programa, ki deluje na Unix-u, See5 pa za različico, ki deluje v operacijskem sistemu Windows. V doktorski disertaciji sem uporabil različico za Windows. Gre za enega najbolj uveljavljenih programov za gradnjo odločitvenih dreves.

preverjanja vrednosti enega atributa, množico T razdeli na več medsebojno izključujočih delov. Tako je sedaj odločitveno drevo za T sestavljeno iz enega vozlišča (angl. node), ki predstavlja test A in po ene veje za vsako od podmnožic. To proceduro ponavljamo za vsako od podmnožic, dokler ne dobimo končnega odločitvenega drevesa.

Odločilno vprašanje je seveda, kako v vsaki fazi izbrati test A , ki bo razdelil učno podmnožico na dva dela. Cilj je kreiranje takega drevesa, ki bo čim boljše opisalo učno množico in bo torej čim manjše. Tako drevo ima namreč tudi največjo moč za napovedovanje rezultatov novih primerov.

Načeloma bi lahko izdelali vsa možna drevesa in izbrali najenostavnejšega. Vendar je dokazano, da je izdelava najboljšega odločitvenega drevesa NP poln problem (Hyafil, Rivest, 1976). Celu ugotovitev prvega vozla, po katerem bi morali deliti za optimalno rešitev, je NP težka (Cox et al., 1989), zato optimalne rešitve za srednje in večje probleme ne moremo pričakovati. Zato se v praksi uporabljajo hevristični algoritmi, običajno na podlagi požrešnih metod. Pri vsaki delitvi torej izmed vseh možnih testov izberemo tistega, ki privede do največje pridobitve v informacijski količini (angl. information gain). To je tudi ena od glavnih pomanjkljivosti odločitvenih dreves, saj taka metoda ne privede nujno do izdelave najboljšega možnega drevesa. Poleg tega se v procesu deljenja zvezne spremenljivke implicitno spremenjene v diskretne, pri čemer se izgubi nekaj informacije (Dreiseitl, Ohno-Machado, 2002).

Informacijsko količino, potrebno za določitev i -tega stanja sistema, ki se v tem stanju nahaja z verjetnostjo p_i , lahko izračunamo po formuli (Gradišar, 2003; Shannon, 1948):

$$I(X) = -\log p_i$$

oziroma $I(X) = -\sum_{i=1}^n p_i \log p_i$ kot vsoto informacijskih količin (entropije) n možnih stanj sistema.

Če torej izberemo en primer iz skupine S primerov in ugotovimo, da ta primer pripada razredu C_j , je verjetnost za to enaka:

$$p_j = \frac{\text{freq}(C_j, S)}{|S|}$$

Informacijska količina za ta primer pa znaša:

$$I(X) = -\log_2\left(\frac{\text{freq}(C_j, S)}{|S|}\right) \text{ bitov.}$$

Če seštejemo informacijsko količino za vse primere iz skupine S , dobimo naslednjo formulo (Quinlan, 1993):

$$info(S) = - \sum_{j=1}^k \frac{freq(C_j, S)}{|S|} * \log_2 \left(\frac{freq(C_j, S)}{|S|} \right)$$

$info(S)$ torej meri povprečno količino informacije (oziroma entropijo), potrebno za določitev razreda, v katerega spada posamezen element.

Po delitvi množice na več podmnožic lahko sedaj povprečno količino informacije izračunamo kot ponderirano vsoto posameznih povprečnih količin informacij po formuli:

$$info_x(S) = \sum_{i=1}^n \frac{|S_i|}{|S|} * info(S_i)$$

Želimo seveda izbrati tako delitev, ki bo privedla do največje pridobitve v informacijski količini oziroma delitev, ki je najbolj informativna (Mladenić et al., 2003). Torej izberemo tisti test, ki ima maksimalno vrednost $info(S) - info_x(S)$. Ekvivalenta je opredelitev, da izberemo tisti test, ki kar najbolj zmanjša entropijo³⁵.

Problem tega kriterija je v tem, da preferira tiste teste, ki imajo večje število možnih izidov (Quinlan, 1993). Denimo, da bi želeli klasificirati študente Ekonomske fakultete po številu let študija do diplome in bi kot atribut med drugim vključili tudi vpisno številko študenta. Ta kriterij bi nato učno množico razdelil ravno po vpisni številki na toliko podmnožic, kot je študentov. V vsaki podmnožici bi bil tako samo en študent, entropija vsake podmnožice pa bi bila 0. Očitno pa je, da vsebinsko gledano vpisna številka kot atribut nima nobene napovedne vrednosti.

To pomanjkljivost odpravimo tako, da ustrezno popravimo pridobitev informacijske količine pri testih z več možnimi izidi. Tako namesto absolutne pridobitve v količini informacije izračunamo količnik. Najprej izračunamo potencialno pridobitev v količini informacije, če razdelimo množico T v n podmnožic. To izračunamo po naslednji formuli:

$$split\ info(X) = - \sum_{i=1}^n \frac{|T_i|}{|T|} * \log_2 \left(\frac{|T_i|}{|T|} \right)$$

Vrednost $split\ info(X)$ se praviloma povečuje s povečevanjem različnih izidov pri posameznem testu. Nato količnik izračunamo tako, da delimo pridobitev v informaciji s faktorjem:

³⁵ Entropija oziroma informacijska količina oziroma kompleksnost je definirana kot število bitov, ki so potrebni za popolno rekonstrukcijo posameznega niza (Chaitin, 1987).

$$\text{količnik} = (\text{info}(S) - \text{info}_x(S)) / \text{split info}(X)$$

Ta količnik torej meri, koliko je koristne (netrivialne) pridobitve v količini informacije. Tako popravljen količnik pridobitve količine informacije praviloma daje bistveno boljše rezultate kot sam kriterij pridobitve informacije (Quinlan, 1988), zato ga uporablja tudi program See5.

Dodaten problem je upoštevanje zveznih spremenljivk pri izdelavi odločitvenega drevesa, saj je načeloma težko določiti mejo, ki naj se uporabi za delitev množice na dva dela. Kot prag se lahko uporabi katerakoli vrednost med v_i in v_{i+1} ³⁶ (če je denimo $v_6 = 5$ in $v_7 = 10$, katero koli število med 5 in 10 povzroči enako delitev učne množice). Ena od možnosti je preslikava teh atributov v diskretni prostor tako, da vrednosti razdelimo na intervale in nato na vsak interval gledamo kot na diskretno vrednost. Intervale lahko razdelimo na ekvidistančne dele, z določitvijo praga ali z dinamično delitvijo (Kokol et al., 2001).

Program See5 uporablja eno od različic delitve intervala z določitvijo praga, zato to metodo predstavljamo nekoliko obširneje (Kokol et al., 2001; Quinlan, 1993). Imamo nek zvezni atribut A , ki v učni množici zavzema N različnih vrednosti. Če sedaj razporedimo te vrednosti po velikosti, dobimo množico $\{v_1, v_2, \dots, v_n\}$. Sedaj torej dobimo $N-1$ možnih pragov (angl. threshold) za delitev učne množice na dva dela. Vse vrednosti med v_i in v_{i+1} so namreč medsebojno enakovredne, saj učno množico razdelijo na enak način³⁷. Pri delitvi množice torej upoštevamo vse možne pragove in izberemo tistega, ki prinese največji pridobitek pri informaciji, v kolikor seveda je ta pridobitek večji od tistega pri delitvi po vrednosti kakšnega drugega (diskretnega ali zveznega) atributa.

Zaradi več možnih delitev se sicer nekoliko podaljša čas izdelave drevesa, vendar za običajne primere še vedno lahko pričakujemo rezultat v nekaj sekundah. Dodatna prednost je, da pravzaprav ni potrebno preiskati vseh $N-1$ možnih pragov. V kolikor vrednosti v_i in v_{i+1} pripadata istemu razredu, nobeden od pragov med tema vrednostma ne more biti optimalen (Fayyad, Irani, 1992).

To metodo je uporabljala prejšnja različica programa C4.5., pri čemer je kot prag bila izbrana manjša od obeh vrednosti³⁸. Osnovni problem je v tem, da imajo tako zvezni atributi prednost pri izbiri, po katerem atributu bomo delili učno množico³⁹. Do tega pride, ker pri zveznih atributih lahko izberemo enega od $N-1$ možnih pragov, pri diskretnih pa je

³⁶ Predpostavljamo seveda, da so vrednosti v množici urejene po velikosti, bodisi padajoče bodisi naraščajoče.

³⁷ Imamo samo $N-1$ in ne $N+1$ možnih pragov, kajti vsak prag, ki je manjši od v_1 ali večji od v_n ne pride v poštev, saj ne razdeli množice na dva dela, ampak vsi primeri ostanejo v eni množici.

³⁸ Načeloma lahko kot prag izberemo katerokoli vrednost med v_i in v_{i+1} . V praksi se dostikrat uporablja kar sredina tega intervala, torej $\frac{v_i + v_{i+1}}{2}$, medtem ko C4.5. uporablja kar vrednost v_i .

³⁹ Ravno tako imajo prednost zvezni atributi z večjim številom različnih vrednosti pred tistimi z manjšim (Quinlan, 1996).

običajno na razpolago precej manjše število možnih delitev. Zato je program dostikrat izbral delitev po zveznem atributu, čeprav bi delitev po diskretnem atributu morda bolje opisala dejansko stanje (Quinlan, 1996; Auer et al., 1995; Dougherty et al., 1995).

Možne rešitve tega problema temeljijo na upoštevanju Ginijevega koeficienta, χ^2 kontingenčne statistične tabele in podobnega (Han, Kamber, 2001). Vsaka od predlaganih rešitev ima svoje prednosti in slabosti (White, Liu, 1994).

See5 ta problem odpravi z dvema popravkoma pri izbiri ustreznega testa za delitev učne množice (Quinlan, 1996):

- prvi popravek temelji na principu najmanjše potrebne opisne dolžine (angl. Minimum Description Length Principle) (Rissanen, 1983). Po tem principu se izbere tista delitev, ki potrebuje minimalno število bitov za opis teorije in izjem. Test po zvezni spremenljivki tako zahteva dodatnih $\log_2(N-I)$ bitov za opis teorije, pri čemer $N-I$ predstavlja število pragov in s tem število možnih delitev. Za to vrednost torej znižamo po prej navedeni formuli izračunano pridobitev informacije za delitev po zvezni spremenljivki,
- drugi popravek temelji na tem, da pri izračunu količnika pridobitve v informacijski količini, to pridobitev delimo s potencialno pridobitev v količini informacije, če razdelimo množico T v n podmnožic (split info). Slednja je funkcija praga in je maksimalna, kadar je število primerov pod in nad pragom enako. Zato najprej izberemo prag t , ki prinese maksimalno korist. Nato je končni izbor atributa, ki bo uporabljen za razdelitev množice, narejen na podlagi prilagojene pridobitve v informaciji.

Predstavljen način kreiranja odločitvenega drevesa pomeni, da na koncu dobimo zelo razvejano drevo z mnogo listi, saj se drevo deli toliko časa, dokler niso v vsaki podskupini samo primeri, ki pripadajo istemu razredu ali pa nadaljnji testi ne bi prinesli novega izboljšanja. To bi pomenilo preveliko prilagajanje (angl. overfitting) razpoložljivim podatkom, kar bi privedlo celo do zmanjšanja natančnosti predvidevanja (Quinlan, 1993). To pomanjkljivost lahko odpravimo s poenostavljanjem (angl. pruning) odločitvenega drevesa in sicer z uporabo ene od dveh možnosti:

- predhodno klestenje (angl. prepruning ali stopping): običajno to izvedemo tako, da za vsako podskupino določimo najboljši način za delitev v dve podskupini z vidika statistične značilnosti, zmanjšanja števila napak, pridobitve informacije ali kakega drugega kriterija. Če taka delitev ne doseže vnaprej določene meje, potem te delitve ne izvedemo, ampak pustimo celo podskupino v enem listu.

Prednost tega pristopa je predvsem ta, da je nekoliko hitrejši, saj ne izgubljammo časa z določanjem vej v drevesu, ki jih nato niti ne bomo uporabili. Osnovni problem tega pristopa je prav v določitvi višine tega kriterija – previsok kriterij povzroči premalo razvejano drevo, prenizek pa drevesa ne poenostavi dovolj (Breiman et al., 1984),

- naknadno klestenje (angl. postpruning): pri tem pristopu najprej zgradimo celotno odločitveno drevo, ki ga nato postopoma poenostavimo z odstranjevanjem posameznih vej. To naredimo tako, da del drevesa zamenjamo z enim samim listom ali eno vejo. To lahko naredimo na več načinov, denimo z uporabo algoritma na podlagi stroškovne kompleksnosti (angl. cost complexity). Po tem algoritmu za vsak vozel izračunamo pričakovano stopnjo napake, če bi ta vozel oklestili. Nato izračunamo še pričakovano stopnjo napake za neoklesteno drevo. Če je slednja večja, potem oklestimo ta del drevesa. Ta postopek ponavljamo toliko časa, dokler ne pridemo do drevesa z najmanjšo pričakovano stopnjo napake (Han, Kamber, 2001).

Tak algoritem izdelave drevesa omogoča pridobitev rešitve v kratkem času tudi za večje probleme (Androulakis, 2005). Pregled ostalih možnosti za poenostavljanje drevesa je v (Breslow, Aha, 1996).

Odločitvena drevesa sem izbral kot ustrezno metodo za problem izbora metode pri standardnem problemu enodimenzionalnega razreza, saj moj problem izpolnjuje pogoje za uspešno uporabo odločitvenih dreves, ki jih navaja Quinlan (1993):

- opis problema z atributi: vsi podatki o posameznem primeru morajo biti izraženi s fiksnim številom lastnosti oziroma atributov. Vsak atribut ima lahko diskretne ali številske vrednosti, bistveno je, da imajo vsi primeri iste attribute.

Ta pogoj je izpolnjen, saj lahko dejansko vsak primer opišem z vnaprej določenimi atributi, kot so število palic v zalogi, število naročil, povprečno povpraševanje po posameznem naročilu in različna razmerja (denimo razmerje med celotnim materialom in celotnimi potrebami). Več o izbiri atributov, ki sem jih vključili v model, podajam v nadaljevanju,

- predhodno definirani razredi: kategorije, v katere bom razporejali primere, morajo biti določene vnaprej. Gre za tako imenovano nadzorovano učenje.

Vnaprej sem določil dve kategoriji, in sicer:

- o problem je rešen optimalno v vnaprej določeni časovni omejitvi (razred 1),
- o optimalna rešitev z eksaktno metodo v časovni omejitvi ni bila najdena (razred 0),
- diskretni razredi: razredi morajo biti medsebojno ostro ločeni – posamezen primer bodisi spada v en razred ali pa ne. V kateri razred spada določeni primer, mora biti jasno vnaprej. To dejstvo je v določenih primerih lahko pomembna pomanjkljivost (Kang et al., 1998). Celotno število razredov mora biti bistveno manjše od števila primerov, ki so na razpolago za izgradnjo odločitvenega drevesa.

Moj primer izpolnjuje oba pogoja, saj je ločnica med obema razredoma jasna, število testnih primerov pa je (zaradi uporabe generatorja naključnih števil) lahko skoraj poljubno veliko in tako seveda znatno presega število razredov,

- zadostno število podatkov: metoda odločitvenih dreves temelji na različnih statističnih testih, ki ugotavljajo vzorce v podatkih. Zato da ločimo dejanske od naključnih vzorcev, je potrebno zadostno število primerov, ki jih lahko uporabimo za kreiranje vzorcev. Koliko je zadostno število, je odvisno od kompleksnosti modela in števila atributov ter razredov.

Glede na to da lahko z generatorjem naključnih števil pridobimo poljubno število primerov in jih rešimo z avtomatizirano proceduro, je ta pogoj v našem primeru vsekakor izpolnjen,

- logični modeli za klasifikacijo: pravila za odločanje se morajo dati izraziti kot skupek logičnih pravil o vrednosti posameznih atributov. Ta metoda torej ni primerna, če želimo posamezne attribute ponderirati in na tej podlagi priti do rezultata.

6.2.3 Uporaba odločitvenih dreves

Pri obravnavanem problem se lahko ta metoda uporabi v različnih primerih in med drugim prinese odgovore na naslednja vprašanja:

- da ugotovimo, ali je bolje, da posamezen problem rešimo s hevristično ali eksaktno metodo. Primer te uporabe prikazujem v tem podpoglavju,
- da ugotovimo, kateri dejavniki imajo največji vpliv na časovno kompleksnost pri reševanju problema s predlagano metodo,
- da ugotovimo ustrezno velikost pri določanju podproblema. Marsikateri problem je možno razdeliti na več manjših problemov in rešiti vsakega od teh problemov optimalno ali pa rešiti večji del problema hevristično in le manjši del optimalno. Optimalno rešeni del problema je običajno najpomembnejši del, torej del, ki prispeva največ k skupni izgubi. Kako velik naj bo posamezen podproblem, je zelo pomembno vprašanje za dobro delovanje metode.

Analiza, prikazana v nadaljevanju tega poglavja, mi omogoča, da natančneje določim ustrezno velikost podproblema in dejavnike, ki jih moram pri določitvi podproblema upoštevati. Ugotovitve iz tega poglavja sem uporabil pri izdelavi kombinirane metode za reševanje, ki jo podrobneje predstavljam v naslednjem poglavju.

Najprej sem želeli torej ugotoviti, pri katerih primerih lahko pričakujemo optimalno rešitev z uporabo eksaktne metode v časovnem intervalu ene minute, ki sem ga arbitrarno določil

kot najdaljši čas, ki ga še lahko porabimo za reševanje posameznega problema v praksi⁴⁰. Kot je razvidno iz prejšnjih poglavij, je večinoma jasno, da pri problemih, kjer je palic v zalogi manj kot 5, lahko z veliko verjetnostjo pričakujemo optimalno rešitev v tej časovni omejitvi (oziroma v praksi večinoma še precej prej, običajno v nekaj sekundah). Po drugi strani pri primerih z več kot 10 različnimi dolžinami palicam v zalogi, eksaktna metoda zelo verjetno ne bo dala optimalne rešitve v tem času, oziroma bo rešitev celo daleč od optimalne. Zato je v takih primerih priporočljiva uporaba hevristične metode.

Za vmesne primere pa odgovor ni jasen, zato je potrebna podrobnejša analiza. Ravno tako še nisem ugotovili, kateri dejavniki (poleg števila palic) najbolj vplivajo na verjetnost za pridobitev optimalne rešitve.

Zato sem z uporabo odločitvenih dreves preveril primere, pri katerih se število palic v zalogi giblje med 5 in 10. Testne primere sem, podobno kot prej, pridobil s pomočjo generatorja naključnih števil PGEN, določanje parametrov (njihov pomen je isti kot na sliki 7) pa je prikazan na sliki 9.

Na ta način pridobimo 1215 testnih primerov (po 5 primerov za vsako od 243 uporabljenih kombinacij parametrov, saj je vrednost parametra r enaka 5), kar zadostuje za kreiranje in testiranje odločitvenega drevesa.

Slika 9: Shema procedure PROGEN, uporabljene za generiranje testnih problemov za izdelavo odločitvenega drevesa

Procedura PROGEN :

```

od g=1 do 3
  od h=1 do 3
    od i=1 do 3
      od j=1 do 3
        od k=1 do 3
           $u_1 \leftarrow 1000 * k$ 
           $u_2 \leftarrow 2000 * k$ 
           $m \leftarrow (j * 2) + 3$ 
           $d \leftarrow (i * 2) + 3$ 
           $v_1 \leftarrow h * 100$ 
           $v_2 \leftarrow h * 200$ 
           $n \leftarrow (g * 2) + 3$ 
           $seme \leftarrow 1000000 * n + 1000 * v_1 + 10 * v_2 + 10 * d + m$ 
           $r \leftarrow 5$ 

```

⁴⁰ Seveda bi lahko povsem isti pristop uporabili tudi ob kakšni drugi metodi ali drugačni časovni omejitvi.

Ključni PGEN ($n, v_1, v_2, d, m, u_1, u_2, seme, r$)

naslednji k

naslednji j

naslednji i

naslednji h

naslednji g

Vsak od navedenih primerov je bil nato rešen z eksaktno metodo na isti način in na istem računalniku kot že primeri v petem poglavju disertacije. Za vsak primer so bili zabeleženi celoten čas reševanja, izguba (optimalna rešitev ali najboljša najdena rešitev v 1 minuti, v kolikor primer ni bil rešen optimalno) ter dejstvo, ali je bil primer zagotovo rešen optimalno ali ne.

Celoten eksperiment (torej kreiranje podatkov, reševanje vseh problemov ter zbiranje rezultatov) je vsega skupaj trajal nekaj več kot 10 ur in je v celoti avtomatiziran, tako da je možna enostavna ponovitev ob drugačnih parametrih. Procedura za izvajanje eksperimenta je napisana v Visual Basic for Applications, MS Excel pa je bil uporabljen za zbiranje in prikaz pridobljenih rezultatov, ki so bili pridobljeni z istim orodjem za reševanje celoštevilskih linearnih programov (MPL/CPLEX) na istem računalniku (AMD, 1300 MHz).

Prvih 30 kombinacij parametrov in rešitve teh problemov je prikazanih v tabeli 7 (zaradi velikega števila podatkov ne prikazujem vseh pridobljenih podatkov)⁴¹. Pomen oznak v tabeli je enak kot v prejšnjih tabelah. V vsaki vrstici je združenih pet podprimerov, ki so bili pridobljeni z uporabo istih parametrov, v zadnjem stolpcu je prikazano število optimalno rešenih primerov za to kombinacijo. Pri izdelavi odločitvenega drevesa je seveda vsak primer vključen ločeno.

⁴¹ Celotni primeri in njihove rešitve so na voljo pri avtorju disertacije.

Tabela 7: Prvih 150 generiranih kombinacij parametrov in njihove rešitve

<i>št</i>	<i>n</i>	<i>v₁</i>	<i>v₂</i>	<i>d</i>	<i>m</i>	<i>u₁</i>	<i>u₂</i>	<i>seme</i>	<i>skupna izguba</i>	<i>skupna izguba (v %)</i>	<i>optim</i>
1	5	100	200	5	5	1000	2000	5102055	1	0,0053%	5
2	5	100	200	5	5	2000	4000	5102055	0	0,0000%	5
3	5	100	200	5	5	3000	6000	5102055	0	0,0000%	5
4	5	100	200	5	7	1000	2000	5102057	0	0,0000%	5
5	5	100	200	5	7	2000	4000	5102057	0	0,0000%	5
6	5	100	200	5	7	3000	6000	5102057	0	0,0000%	5
7	5	100	200	5	9	1000	2000	5102059	0	0,0000%	5
8	5	100	200	5	9	2000	4000	5102059	0	0,0000%	5
9	5	100	200	5	9	3000	6000	5102059	0	0,0000%	5
10	5	100	200	7	5	1000	2000	5102075	9	0,0301%	4
11	5	100	200	7	5	2000	4000	5102075	0	0,0000%	5
12	5	100	200	7	5	3000	6000	5102075	0	0,0000%	5
13	5	100	200	7	7	1000	2000	5102077	5	0,0181%	3
14	5	100	200	7	7	2000	4000	5102077	0	0,0000%	5
15	5	100	200	7	7	3000	6000	5102077	0	0,0000%	5
16	5	100	200	7	9	1000	2000	5102079	4	0,0138%	3
17	5	100	200	7	9	2000	4000	5102079	0	0,0000%	5
18	5	100	200	7	9	3000	6000	5102079	0	0,0000%	5
19	5	100	200	9	5	1000	2000	5102095	98	0,2976%	2
20	5	100	200	9	5	2000	4000	5102095	0	0,0000%	5
21	5	100	200	9	5	3000	6000	5102095	0	0,0000%	5
22	5	100	200	9	7	1000	2000	5102097	13	0,0350%	3
23	5	100	200	9	7	2000	4000	5102097	0	0,0000%	5
24	5	100	200	9	7	3000	6000	5102097	0	0,0000%	5
25	5	100	200	9	9	1000	2000	5102099	4	0,0116%	4
26	5	100	200	9	9	2000	4000	5102099	0	0,0000%	5
27	5	100	200	9	9	3000	6000	5102099	0	0,0000%	5
28	5	200	400	5	5	1000	2000	5204055	801	2,0828%	4
29	5	200	400	5	5	2000	4000	5204055	6	0,0145%	5
30	5	200	400	5	5	3000	6000	5204055	1	0,0024%	5

Preden z uporabo programa See5 kreiram odločitveno drevo, je potrebno odgovoriti še na eno pomembno vprašanje, in sicer katere spremenljivke uporabiti kot attribute za izdelavo drevesa. Očitno moram vključiti tiste spremenljivke, za katere na podlagi dosedanjih rezultatov pričakujem, da vplivajo na časovno kompleksnost reševanja problemov z eksaktno metodo.

Tako sem vključil naslednje spremenljivke:

- število palic v zalogi, število naročil, povprečno povpraševanje po posameznem naročilu: očitno te spremenljivke vplivajo na velikost modela, saj število palic v

zalogi in število naročil vplivata na število celoštevilskih spremenljivk in omejitev v modelu, medtem ko povpraševanje po posamezni dolžini naročila vpliva na število možnih kombinacij.

Vse te spremenljivke so vključene kot atributi pri izdelavi odločitvenega drevesa.

- povprečnih dolžin palic in naročil seveda ne bi bilo smiselno vključiti kot absolutne vrednosti, saj njihovo absolutno spreminjanje ne vpliva na model (če bi na primer vse dolžine pomnožili s faktorjem 10, s tem modela ne bi čisto nič spremenili). Zato so te spremenljivke vključene kot del dveh razmerij:

- r – razmerje med povprečno dolžino palice in povprečno dolžino naročila:

$$\frac{u_1 + u_2}{v_1 + v_2}. \text{ V (Gradišar et al., 1999) je bilo statistično ugotovljeno, da}$$

povečanje tega razmerja večinoma pomeni tudi boljše rešitve pri uporabi hevristične metode; pri tem je mišljena manjša povprečna odstotna izguba materiala⁴². Želel sem ugotoviti, kakšen vpliv ima to razmerje pri uporabi eksaktne metode. Pri tem naj poudarim, da me tokrat ne zanima, kakšna je absolutna ali relativna izguba, ampak kakšna je verjetnost, da bo posamezen primer rešen optimalno,

- q – razmerje med celotnim razpoložljivim materialom in celotnimi

potrebami:
$$\frac{\sum_{j=1}^m d_j}{\sum_{i=1}^n s_i * b_i}$$
 - večje razmerje med celotnim materialom in

celotnimi potrebami naj bi v skladu s pričakovanji pomenilo večjo verjetnost, da bom pridobil optimalno rešitev. To trditev sem želel empirično preveriti.

Omenjene podatke sem nato uporabil za generiranje odločitvenega drevesa. Pri tem sem uporabil 70% podatkov za izdelavo odločitvenega drevesa (»učenje« drevesa), 30% pa kot testne podatke za preverjanje natančnosti odločitvenega drevesa. Razporeditev podatkov v obe skupini je bila naključna in jo je izvedel program See5.

Da bi preprečil pretirano prileganje podatkov, je bil postavljen dodaten pogoj, da mora vsaka veja v drevesu vsebovati vsaj 10 primerov. Pri deljenju drevesa torej ne upoštevamo tistih možnih delitev, ki bi učno množico razdelile tako, da bi ena od podmnožic imela manj kot 10 učnih objektov.

⁴² Razliko med opredelitvijo pojma »dobra rešitev« v predhodnih delih in tej disertaciji sem podal že na strani 72.

Tako pridobljeno odločitveno drevo je prikazano na sliki 10. Številke v oklepaju na sliki pomenijo, koliko primerov, uporabljenih za trening, pripada posameznemu listu.

Oznaka pred oklepajem pomeni, ali je verjetno, da bo primer rešen optimalno z eksaktno metodo v 60 sekundah (številka 1) ali ne (številka 0). Prva številka v oklepaju pomeni število pravilno, druga število napačno klasificiranih primerov v tem razredu.

Slika 10: Odločitveno drevo za izbiro ustrezne metode za rešitev problema razreza glede na vrednosti parametrov

```

q > 2.18239:
...m <= 7: 1 (126/2)
: m > 7:
: ...n > 5: 1 (76/2)
:   n <= 5:
:     ...q <= 2.86825 : 0 (16/4)
:     q > 2.86825 : 1 (44/1)
q <= 2.18239:
...r > 15: 1 (30/2)
  r <= 15
  ...m > 5
    .....q > 1.69128
      : ...n <= 5 : (28/2)
      :   : n > 5 : (50/20)
      :   q <= 1.69128
      :   ...m > 7 : 0 (124/6)
      :   m <= 7
      :   :
      :   ...n >= 5 : 0 (115/20)
      :   n <= 5
      :   .....r <= 5 : 1 (22/4)
      :   r > 5 : 0 (16/2)
      m <= 5
      ...r <= 5 : 1 (65/7)
      r > 5
        .....q > 1.6137 : 1 (19)
        q <= 1.6137
        ...n <= 5 : 1 (27/9)
        n > 5
        ....r <= 6.667: 0(20/7)
        r > 6.667
          ..n <= 7: 0(38/17)
          n > 7
          ...q <= 0.78137: 1(14/3)
          q > 0.78137 : 0 (20/4)

```

S slike 10 je denimo razvidno, da bi problem z 10 dolžinami palic, 5 različnimi dolžinami naročil in razmerjem med celotnim materialom in celotnimi potrebami 3, reševali z eksaktno metodo. Od 78 takih problemov v učni množici podatkov je namreč takšna metoda primerna za kar 76 problemov.

Gledano v celoti je odločitveno drevo pravilno predvidelo, ali bo problem rešen optimalno v eni minuti v 86,1% podatkov za trening in 84,1% testnih podatkov, kar predstavlja precej visok odstotek in je vsekakor zadovoljivo za moj namen. To še dodatno velja ob upoštevanju dejstva, da sem testiral velikostni razred problema, ki je po velikosti ravno nekje na meji med primernostjo obeh metod. Pri drugih velikostnih razredih bi lahko zato pričakovali še boljše rezultate.

Ker so, kot navedeno zgoraj, odločitvena drevesa po svojih lastnostih zelo primerna za moj problem, in ker smo z njihovo uporabo tudi dejansko dobili zelo dobre rezultate, se nisem odločil še za uporabo kake druge metode podatkovnega rudarjenja. To odločitveno drevo lahko brez težav implementiramo kot podprogram v katerem koli programskem jeziku in ga nato uporabimo za avtomatično izbiro ustrezne metode za vsak individualni primer.

S slike 10 je tudi razvidno, da ima za to vrsto in približno velikost problema na kompleksnost problema največji vpliv razmerje med celotnim materialom in potrebami, precejšen vpliv pa ima tudi število palic. Ti dve spremenljivki se namreč nahajata precej visoko v odločitvenem drevesu. Vpliv števila naročil in povprečnega povpraševanja po posameznem naročilu imata presenetljivo majhen vpliv, saj se nahajata nizko v odločitvenem drevesu. Te ugotovitve sem uporabil v naslednjem poglavju doktorske disertacije pri načinu izbire velikosti podproblema pri kombinirani metodi.

Očitno je, da samo visok delež pravilno klasificiranih testnih podatkov in podatkov za trening ne more biti zadostno zagotovilo, da je odločitveno drevo res ustrezno za izbiro pravilne metode pri problemih take velikosti.

Zato sem odločitveno drevo dodatno testiral na podatkih, prikazanih v tabeli 6. Ker so bili tudi rezultati v tabeli 6 pridobljeni na istem računalniku in z istim reševalcem, so primerni za testiranje odločitvenega drevesa. Ob tem je potrebno posebej poudariti, da podatki iz tabele 7 na noben način niso bili uporabljeni ne pri kreiranju (učna množica) ne pri prvotnem testiranju (testna množica) odločitvenega drevesa. Na tak način sem popolnoma izločil potencialni škodljivi vpliv pretiranega prilagajanja (angl. overfitting).

V tabeli 8 tako predstavljam 27 primerov (vsak sestavljen iz 10 podprimerov) ter njihovo izgubo z obema metodama. Oznaka D pomeni reševanje modela za dovolj materiala, oznaka N modela ob primanjkljaju materiala. Oznaka D/N pomeni, da je bilo v nekaterih podprimerih dovolj materiala, v drugih pa ne.

Bistven prispevek odločitvenih dreves je napotek, katero izmed metod naj izberemo. To je prikazano v predzadnjem stolpcu (oznaka E pomeni, naj rešimo tak problem z eksaktno metodo; oznaka H pa, da naj ga rešujemo s hevristično metodo CUT).

Tabela 8: Izbira ustrezne metode na podlagi odločitvenih dreves

zap. št.	Dovolj materiala?	izguba CUT		izguba eksaktna metoda		izbrana metoda	storjena napaka
		cm	%	cm	%		
1	D	8	0,0213%	1	0,0027%	E	0
2	D	0	0,0000%	0	0,0000%	H	0
3	D	0	0,0000%	0	0,0000%	H	0
4	D/N	1182	1,5460%	1028	1,3445%	E	0
5	D	28	0,0162%	212	0,1229%	H	0
6	D	9	0,0039%	62	0,0269%	H	0
7	N	1940	2,7256%	1702	2,3912%	E	0
8	D	213	0,0980%	1457	0,6706%	H	0
9	D	285	0,0832%	1780	0,5196%	H	0
10	D/N	59	0,0807%	18	0,0246%	H	41
11	D	0	0,0000%	8	0,0052%	E	8
12	D	2	0,0009%	23	0,0103%	E	21
13	N	88	0,1103%	8	0,0100%	E	0
14	D/N	172	0,0575%	1613	0,5393%	H	0
15	D	22	0,0046%	1833	0,3866%	H	0
16	N	227	0,3155%	49	0,0681%	E	0
17	N	272	0,0949%	2074	0,7236%	H	0
18	D/N	541	0,0836%	6533	1,0099%	H	0
19	N	7	0,0095%	0	0,0000%	E	0
20	D	10	0,0042%	429	0,1803%	H	0
21	D	0	0,0000%	438	0,1217%	E	438
22	N	47	0,0618%	1	0,0013%	E	0
23	N	36	0,0120%	1043	0,3468%	H	0
24	D	159	0,0242%	5615	0,8531%	H	0
25	N	81	0,1085%	41	0,0549%	H	40
26	N	93	0,0311%	984	0,3294%	H	0
27	N	112	0,0163%	5316	0,7742%	H	0

Vir: (Trkman, 2002b; Trkman, Gradišar, 2003a).

Zadnji stolpec predstavlja napako, ki bi jo dejansko zagrešili z izborom take metode, oziroma razliko med izgubo pri izbrani metodi in tisto, ki je dejansko boljša za take primere. Seveda lahko to, katera metoda je boljša, z gotovostjo trdimo šele, ko rešimo problem z obema metodama. Zgrajeno odločitveno drevo pa nam omogoča, da vsak problem rešujemo samo enkrat.

V skladu z drevesom bi morali z eksaktno metodo rešiti primere z naslednjimi zaporednimi številkami iz tabele 5: 1, 4, 7, 11, 12, 13, 16, 19, 21 in 22, medtem ko naj bi ostale rešili s hevristično metodo.

Od teh 27 primerov dajeta v dveh primerih (primera 2 in 3) obe metodi enak rezultat, ki je hkrati tudi optimalen. Zato v teh dveh primerih seveda ne moremo ugotavljati, ali je odločitveno drevo izbralo pravo metodo. Od preostalih 25 problemov je drevo pravo metodo izbralo v 20 primerih (80,0%), v 5 primerih pa je naredilo napako. Od teh napak so štiri imele za posledico le manjše povečanje izgube (gre za primer 11, kjer je 8 cm namesto 0 cm; primer 12, kjer je izguba 23 cm namesto 2 cm; primer 25, kjer je izguba 81 cm namesto 41 cm ter primer 10, kjer je izguba 59 cm namesto 18 cm). Le v enem primeru je bila storjena večja napaka (primer 21, kjer je izguba 438 cm, namesto 0 cm, kot bi bila, če bi drevo za reševanje teh problemov izbralo hevristično metodo). Tri napake so bile posledica izbora eksaktne namesto hevristične metode, ena napaka pa izbora hevristične namesto eksaktne metode.

Najpomembnejši rezultat je, da bi bila celotna izguba 5.593 cm, če bi vse probleme rešili s hevristično metodo, 32.268 cm ob reševanju z eksaktno metodo in 5.310 cm, če vsak problem rešimo s tisto metodo, ki jo je izbralo odločitveno drevo. To kaže na to, da predlagani pristop dejansko vodi k izboljšanim rezultatom in zmanjšani izgubi materiala.

Očitno bi lahko sicer vsak primer rešili z obema metodama in obdržali le boljši rezultat, vendar bi to zahtevalo dodaten napor in predvsem čas. Vnaprejšnja izbira ustrezne metode pa zagotavlja zelo podobne rezultate kot ob reševanju z obema metodama. Ko je odločitveno drevo pripravljeno, pa sam postopek izbire traja zanemarljivo malo časa. Še enkrat naj poudarim, da sem pri kreiranju odločitvenega drevesa uporabili povsem druge primere, tako da sem se izognil problemu pretirane prilagajanja drevesa.

Pri odgovoru na drugo in tretje vprašanje na začetku tega podpoglavja (torej vpliv posameznih dejavnikov na rešljivost ter primerno sestavo podproblema pri kombiniranih metodah) sem si pomagal tudi z dodatno statistično analizo. Izvedel sem analizo povezanosti med odvisno spremenljivko (dejstvom, ali smo v časovni omejitvi našli optimalno rešitev) in posameznimi zgoraj navedenimi spremenljivkami. Pri tem je potrebno poudariti, da je ena od izbranih »neodvisnih« spremenljivk delno odvisna od preostalih (kolienarnost), saj število palic, naročil in kosov posameznega naročila delno vpliva na razmerje q (vendar na to razmerje bolj vpliva dolžina naročil in palic, ki pa iz zgoraj opisanih razlogov nista vključeni med neodvisne spremenljivke), medtem ko razmerje r ni odvisno od teh treh spremenljivk.

Izvedel sem korelacijsko analizo teh dejavnikov, s katero sem posamično testiral povezanost vsake spremenljivke in optimalnosti rešitev. Rezultati, prikazani v tabeli 9, kažejo, da je ob vzorcu 1215 primerov verjetnost optimalne rešitve statistično značilno pozitivno povezana z razmerjem med materialom in potrebami ter razmerjem med

celotnim materialom in celotnimi potrebami. Statistično značilno negativno korelacijo pa smo ugotovili s številom palic, številom naročil ter povprečnim številom kosov posameznega naročila.

Tabela 9: Korelacija med vhodnimi spremenljivkami in optimalnostjo rešitve

spremenljivka	kazalec	optimalnost rešitve
<i>m</i>	Korelacijski koeficient	-,214
	Značilnost (dvostranski preizkus)	,000
<i>d</i>	Korelacijski koeficient	-,161
	Značilnost (dvostranski preizkus)	,000
<i>n</i>	Korelacijski koeficient	-,228
	Značilnost (dvostranski preizkus)	,000
<i>q</i>	Korelacijski koeficient	,426
	Značilnost (dvostranski preizkus)	,000
<i>r</i>	Korelacijski koeficient	,441
	Značilnost (dvostranski preizkus)	,000

Rezultati so pričakovano v skladu z rezultati, pridobljenimi z odločitvenimi drevesi. Visoki razmerji namreč olajšujeta pridobitev optimalne rešitve, medtem ko povečevanje ostalih spremenljivk povečuje obseg problema in s tem zmanjšuje možnost pridobitve optimalne rešitve. Tudi to so pomembne informacije za nadaljnje delo pri izboru primerne velikosti podproblema ob kombinaciji eksaktne in hevristične metode. Pri tem pa je potrebno rezultate iz tabele 9 vendarle jemati z zadržkom zaradi razmeroma preproste statistične tehnike, uporabljene za analizo.

Pri tem je vendarle potrebno upoštevati, da sklepi v tem poglavju disertacije veljajo za tovrsten problem in za približni velikostni razred, iz katerega prihajajo obravnavani problemi. Za drugačen problem ali bistveno drugačno velikost istega problema (oziroma ob reševanju na bistveno hitrejšem ali počasnejšem računalniku, ob drugi časovni omejitvi, z drugačno metodo...) ne morem trditi, da na čas reševanja z eksaktno metodo vplivajo iste spremenljivke in to na isti način kot pri primerih tega velikostnega razreda.

Zato odločitveno drevo ter tudi zgoraj predstavljena analiza povezanosti nista univerzalno uporabna za vsako velikost problema. Za drugačne probleme bi morali tovrstno analizo ponoviti, pri čemer obstoječe procedure za izvajanje eksperimenta in generatorji naključnih števil omogočajo, da to naredimo razmeroma enostavno, hitro in brez pretiranega napora. Nato lahko zgrajeno drevo uporabljamo za izbiro metode za vse nadaljnje probleme.

6.3 Kombinacija hevristične in eksaktne metode (C-CUT) ⁴³

Do sedaj sem v doktorski disertaciji že predstavil hevristično in eksaktno metodo za rešitev problema razreza ter način, kako lahko izberemo ustrezno metodo glede na karakteristike posameznega problema.

Obe metodi imata klasične pomanjkljivosti navedene v literaturi – eksaktna metoda zaradi NP težkosti obravnavanega problema ni primerna za večje primere, hevristična pa tudi pri manjših primerih dostikrat ne privede do optimalne rešitve. Tudi če je rešitev dejansko optimalna, tega metoda ne ugotovi (razen pri izgubi 0). Problem pri hevristični metodi so predvsem že omenjeni končni pogoji, zaradi katerih je običajno slabše rešen predvsem manjši del problema.

Ker obe metodi rešujeta isti problem, se hitro pojavi ideja, da bi metodi povezali. V tem delu tako predstavljam način, kako lahko s kombiniranjem hevristične metode za rešitev večine problema in eksaktne metode za rešitev manjšega dela dosežemo precej boljše rezultate, kot s katero koli od treh prej predstavljenih metod.

Osnovna ideja tovrstne kombinacije je združitev prednosti hevristične metode (predvsem dejstva, da zelo hitro pridemo do dobre rešitve tudi pri večjih problemih) ter eksaktne metode (možnost, da pridemo do optimalne rešitve za manjše probleme). Hkrati želim čim bolj zmanjšati slabosti obeh metod (torej dejstva, da hevristična metoda praviloma ne daje optimalnih rešitev, ter dejstva, da eksaktna metoda ni primerna za večje probleme zaradi hitro rastoče časovne zahtevnosti).

Osnovna ideja te kombinacije je, da celoten problem najprej rešim s CUT, s čimer že pridobim eno od možnih rešitev problema, ki je uporabna in običajno precej dobra, ni pa nujno optimalna. Kot že omenjeno, je namreč pokazano, da je CUT boljši od obeh hevrističnih metod (Gradišar et al., 2002), zato sem ga izbral za uporabo v novi metodi. CUT je namreč razširjena in izboljšana različica prejšnje hevristične metode COLA (Gradišar et al., 1996; Gradišar et al., 1997).

Pri reševanju problema s CUT je običajno, da večino izgube povzroči nekaj slabše narezanih palic – zaradi vpliva končnih pogojev so to skoraj vedno tiste palice, za katere je algoritem poiskal načrt razreza nazadnje (ni pa to nujno, niti ni pomembno za delovanje novo razvite metode). Rešitev za najslabše rešeni del problem torej želim izboljšati, saj bom s tem pomembno izboljšal tudi rešitev celotnega problema.

Tako v doktorski disertaciji predstavljam nov dvofazni pristop k optimizaciji razreza za zmanjševanje izgub. V prvi fazi torej pridobim že zelo dobro rešitev s pomočjo hevristične procedure CUT, ki (kot je bilo pokazano v predhodnih člankih) že prinese majhno izgubo

⁴³ Predstavitev metode je povzeta in razširjena po članku (Gradišar, Trkman, 2005); (Trkman, 2002b).

materiala. Vendar pa želim to rešitev še izboljšati z boljšo rešitvijo kritičnega dela problema.

Zato palice, ki so največ prispevale k celotni izgubi, v drugi fazi ponovno »narežem« z uporabo eksaktne metode. Ta je v tem primeru uporabna, saj je obseg kritičnega podproblema razmeroma majhen. S to metodo namreč poiščem načrt razreza samo za manjše število palic, za večji del problema pa obdržim rešitev, pridobljeno v prvi fazi s hevristično metodo.

Končni načrt razreza je tako kombinacija rezultata prve (za večino palic) in druge (za tiste palice, ki smo jih narezali ponovno) faze. Metodo sem poimenoval C-CUT (angl. combined cutting) in je predstavljena tudi v (Gradišar, Trkman, 2002a; Gradišar, Trkman, 2005; Trkman, 2002b). Ker je osnovni problem povsem enak kot v prejšnjih poglavjih, tokrat modelov za probleme s preveč in premalo materiala ne navajam ponovno. Drugačen je le pristop k reševanju.

Ključno vprašanje je, kako izbrati tako obseg kot strukturo podproblema, da bom čim boljše izkoristili omenjene prednosti obeh uporabljenih metod ob minimizaciji njunih šibkih strani. Osnovna ideja (torej izbor najslabše narezanih palic na tak način, da obseg problema ne presega velikostne meje, ki je še primerna za reševanje z eksaktno metodo) je bila že opisana.

Kot je bilo pokazano v prejšnjem poglavju, lahko pri eksaktni metodi pričakujemo boljše rezultate kot pri CUT-u ob sprejemljivi časovni omejitvi (denimo 1 minuta), v kolikor je podproblem sestavljen iz manj kot 10 različnih dolžin palic in dolžin naročil. Kot so pokazale predhodne analize, je ključno predvsem število različnih palic ter razmerji med celotnim materialom in celotnimi potrebami ter med dolžino palic in dolžina naročila.

Ob taki kombinaciji lahko pričakujemo znatno boljše rezultate ob nekoliko večji časovni kompleksnosti. Ker je velikost podproblema enaka ne glede na velikost osnovnega problema in ker je CUT primeren tudi za reševanje tudi zelo velikih problemov, je torej tu razvita metoda primerna za vse velikosti osnovnega problema.

Natančen algoritem izbora palic v vsakem posameznem primeru je predstavljen v nadaljevanju in tudi na sliki 11. V opisu metode uporabljam naslednje nove spremenljivke:

p – število palic, ki jih je v načrtu razreza uporabila metoda CUT,

f – število palic, uporabljenih v 1. fazi, ki jih bom vključil v podproblem za 2. fazo,

g – število neuporabljenih palic iz 1. faze, ki jih bom vključil v podproblem za 2. fazo,

b'_i – število kosov dolžine s_i , ki jih bomo vključil v podproblem za 2. fazo.

Pomen preostalih spremenljivk je enak kot v zgornjih primerih.

Algoritem za rešitev ima naslednje korake in se nekoliko razlikuje za primere s preveč in premalo materiala.

1. korak: reši problem razreza z metodo CUT in shrani tako pridobljen rezultat;
2. korak: če je rešitev zanesljivo optimalna, potem prenehaj in prikaži rešitev 1. faze kot končno rešitev.

Rešitev je zagotovo optimalna, če je izpolnjen eden od naslednjih pogojev:

$$(1) \sum_{j=1}^m t_j = 0 \text{ (za primere z dovolj materiala)}$$

$$(2) \sum_{j=1}^m \delta_j = 0 \text{ (za primere s pomanjkanjem materiala)}$$

$$(3) (0 < (\sum_{j=1}^m d_j - \sum_{i=1}^n (s_i * b_i)) < UB) \wedge (\sum_{j=1}^m t_j = \sum_{j=1}^m d_j - \sum_{i=1}^n (s_i * b_i))$$

Prva dva pogoja sta jasna – rešitev, pri kateri je izguba 0, je očitno zagotovo optimalna. Tretji pogoj morda zahteva nekoliko razlage – da je rešitev optimalna, lahko trdimo tudi, kadar je materiala dovolj, vendar razpoložljiv material presega zahtevane potrebe za manj kot UB . V kolikor je v takem primeru izguba enaka razliki med razpoložljivim materialom in potrebami, je taka rešitev zanesljivo optimalna. Tudi če bi vse ostanke po posameznih palicah združili v eni palici, bi bil tak ostanek namreč še vedno krajši od zgornje meje. Kot vemo, vsi ostanki krajši od te meje, vedno štejejo kot izguba. Od 270 testnih primerov je sicer le en tak, ki izpolnjuje tretji pogoj, in lahko tako trdimo, da je rešitev optimalna, čeprav izguba ni 0.

Ob tem je potrebno poudariti, da je izpolnitev enega od zgoraj navedenih pogojev zadostni, ne pa tudi potreben pogoj za optimalnost rešitve. Kot že omenjeno, ne CUT ne COLA ne pokažeta, ali je pridobljena rešitev optimalna ali ne. Uporabljeni pristop pri hevrističnih metodah tudi ne omogoča kakšnega dodatnega testa optimalnosti, razen seveda reševanja z eksaktno metodo za manjše probleme⁴⁴.

Zato se kot optimalne rešitve štejejo le tiste, ki izpolnjujejo enega od navedenih pogojev, vse ostale rešitve (med njimi po vsej verjetnosti tudi nekaj optimalnih, česar pa ne moremo ne dokazati ne ovreči) vstopijo v naslednjo fazo. Glede na opisani način delovanja metode C-CUT s tem ne izgubimo ničesar, saj se na koncu tako ali tako obdrži najboljša rešitev in torej ne »izgubimo« tako rešenih problemov (tistih, ki so sicer bili rešeni optimalno,

⁴⁴ V (Gradišar, 1999) se kot zadosten pogoj za optimalnost rešitve sicer navaja pogoj, da je skupna izguba manjša od najkrajše dolžine naročila, vendar to dejansko ni niti zadosten niti potreben pogoj za optimalnost rešitve.

vendar tega nismo mogli zagotoviti). Nekoliko se podaljša edino skupni čas reševanja, vendar ne za več kot eno minuto.

3. korak: v tem koraku določimo, katere palice in naročila bomo vključili v podproblem, ki ga bomo nato rešili z eksaktno metodo. Kot že omenjeno, izberemo predvsem palice, ki so imele precejšnjo izgubo v prvi fazi, dodatno pa še palice, ki zaradi svoje dolžine vodijo do izboljševanja strukture problema (predvsem razmerja med celotnim razpoložljivim materialom in celotnimi potrebami v podproblemu) in s tem možnosti za pridobitev boljše rešitve v drugi fazi.

Omenjene palice in naročila izberemo na podlagi rezultatov iz prve faze, pri čemer se postopek razlikuje za primere z dovolj in za primere s premalo materiala.

Za primere z dovolj materiala je postopek naslednji:

- vse uporabljene palice ($y_j=0$) razvrsti po padajočem δ_j ($j=1, \dots, p$)

Glede na to da so vse dolžine palic lahko različne in da vrstni red, v katerem režemo palice, ni pomemben⁴⁵, lahko to razvrščanje izvedemo, ne da bi kakorkoli vplivali na rešitev ali na manjšo splošno uporabnost metode.

Palice obvezno razvrstimo po padajoči izgubi zato, da zagotovimo, da bomo v naslednjem koraku, ko bomo izbrali prvih nekaj palic iz te skupine, v podproblem vključili tiste palice, ki največ prispevajo h končni izgubi. Zelo pomembno je, da palice razvrstimo po δ_j (torej ostanku palice) in ne po t_j (celotna izguba palice).

Le pri takem razvrščanju namreč zagotovimo, da bomo v podproblem vključili tudi palico (v kolikor taka palica pri prvotni rešitvi seveda obstaja), kjer je $u_j = 1$, $\delta_j > 0$ in $t_j=0$ – gre torej za palico, ki ni bila narezana do konca, pri kateri je ostanek večji od UB in ki ne šteje kot izguba. Če te palice v podproblem ne bi vključili, bi bilo namreč možno, da bi pri končni rešitvi (ko bi združili rešitev hevristične metode in rešitev podproblema) dobili dve palici, ki bi imeli vrednost $u_j = 1$, kar bi bilo v direktnem nasprotju s peto omejitvijo⁴⁶.

- vse neuporabljene palice ($y_j=1$) razvrsti po padajočem d_j ($j=p+1, \dots, m$)

V podproblem vključimo tudi nekaj neuporabljenih palic, torej takih, ki niso bile vključene v prvotni načrt razreza. Osnovna ideja je, da s tem povečamo število možnih kombinacij in posledično tudi možnost, da bo rešitev podproblema boljša od rešitve, ki jo je pridobil CUT.

⁴⁵ Vrstni red bi bil pomemben, v kolikor bi bilo več palic enake dolžine in bi v model vključil tudi stroške za spreminjanje nastavitve rezila za različne vzorce ali ob podobnih razširitvah modela.

⁴⁶ Ena taka palica bi izhajala iz prve faze reševanja s hevristično metodo, druga pa iz druge faze. Z vključitvijo palice iz prve faze tudi v drugo fazo sem tak možen zaplet vnaprej preprečil.

Neuporabljene palice sortiramo od največje do najmanjše zato, ker s tem povečamo razmerje med celotnim materialom in celotnimi potrebami ter tudi razmerje med povprečno dolžino palice in povprečno dolžino naročila v vzorcu. Statistične analize, prikazane v tej doktorski disertaciji in tudi v (Gradišar et al., 2002), so pokazale, da povečanje teh razmerij poveča tudi možnost, da bo končna pridobljena rešitev imela manjšo skupno izgubo materiala ter da bomo našli optimalno rešitev podproblema z eksaktno metodo ob dani časovni omejitvi.

- izberi podmnožico palic in sicer prvih f uporabljenih in prvih g neuporabljenih palic iz predhodno razvrščenih seznamov

Statistična analiza je pokazala, da ima število palic precej večji vpliv na verjetnost, da bomo v določenem času našli optimalno rešitev kot pa število naročil ali povprečno povpraševanja po posameznem naročilu. Zato je število palic v pod problemu določeno vnaprej, medtem ko se število naročil in potreba po posameznem naročilu izračuna v naslednjem koraku. Vrednost f in g določimo glede na čas, ki je na razpolago za rešitev podproblema, kakovost uporabljenega reševalca in procesorsko moč računalnika, na katerem poteka izdelava načrta razreza.

- ugotovi podmnožico naročil in izračunaj potrebno število kosov posameznega naročila

Potrebno število kosov posameznega naročila enostavno izračunamo tako, da seštejemo vse kose, ki so bili izrezani iz tistih palic, ki smo jih vključili v podproblem. To torej izračunamo po naslednji formuli:

$$b'_i = \sum_{j=1}^f x_{ij} \quad \forall i$$

Možno je, da za kako od naročil velja, da je $b'_i = 0$ – da torej ta dolžina naročila ni nastopala v nobeni od palic, ki smo jih vključili v podproblem. V tem primeru takšnega naročila seveda ne vključimo v podproblem, saj so že izpolnjene celotne potrebe po tej dolžini naročila.

Za primere s premalo materiala je postopek naslednji:

- vse uporabljene palice ($y_j=0$) razvrsti po padajočem δ_j ($j=1 \dots p$)

Seveda so v primerih s premalo materiala vse palice uporabljene v načrtu razreza. Ravno tako v skladišče ne vračamo nobene delno uporabljene palice, saj za te primere zgornja meja sploh ni določena.

- izberi prvih f palic s tako razvrščenega seznama

Motiv za izbiro teh palic je podoben kot pri prvem primeru – izberemo tiste palice, ki imajo največjo izgubo in so torej največ prispevale tudi k celotni izgubi pri hevristični

rešitvi. Ponovna rešitev za te palice, pri katerih je večja izguba najverjetneje posledica končnih pogojev in ne same neprimernosti palice za razrez, najverjetneje lahko vodi do boljših rešitev.

- *ugotovi dolžine naročil in potrebno število kosov posamezne dolžine*

Potrebno število kosov posamezne dolžine izračunamo po naslednji enačbi:

$$b'_i = \sum_{j=1}^f x_{ij} + (b_i - \sum_{j=1}^m x_{ij})$$

Iz enačbi je razvidno, da moramo poleg tistih kosov, ki so bili izrezani iz palic, ki jih smo jih vključili v podproblem, upoštevati tudi tiste kose, ki so pri hevrstični rešitvi ostali nenarezani. Možno je, da bo eksaktna metoda v načrt razreza poleg drugih palic vključila tudi drugačno število posamezne dolžine naročila.

Podobno kot pri prvem primeru dolžine naročil, kjer je $b'_i = 0$, ne vključimo v model.

4. korak: Poišči rešitev podproblema z eksaktno metodo.

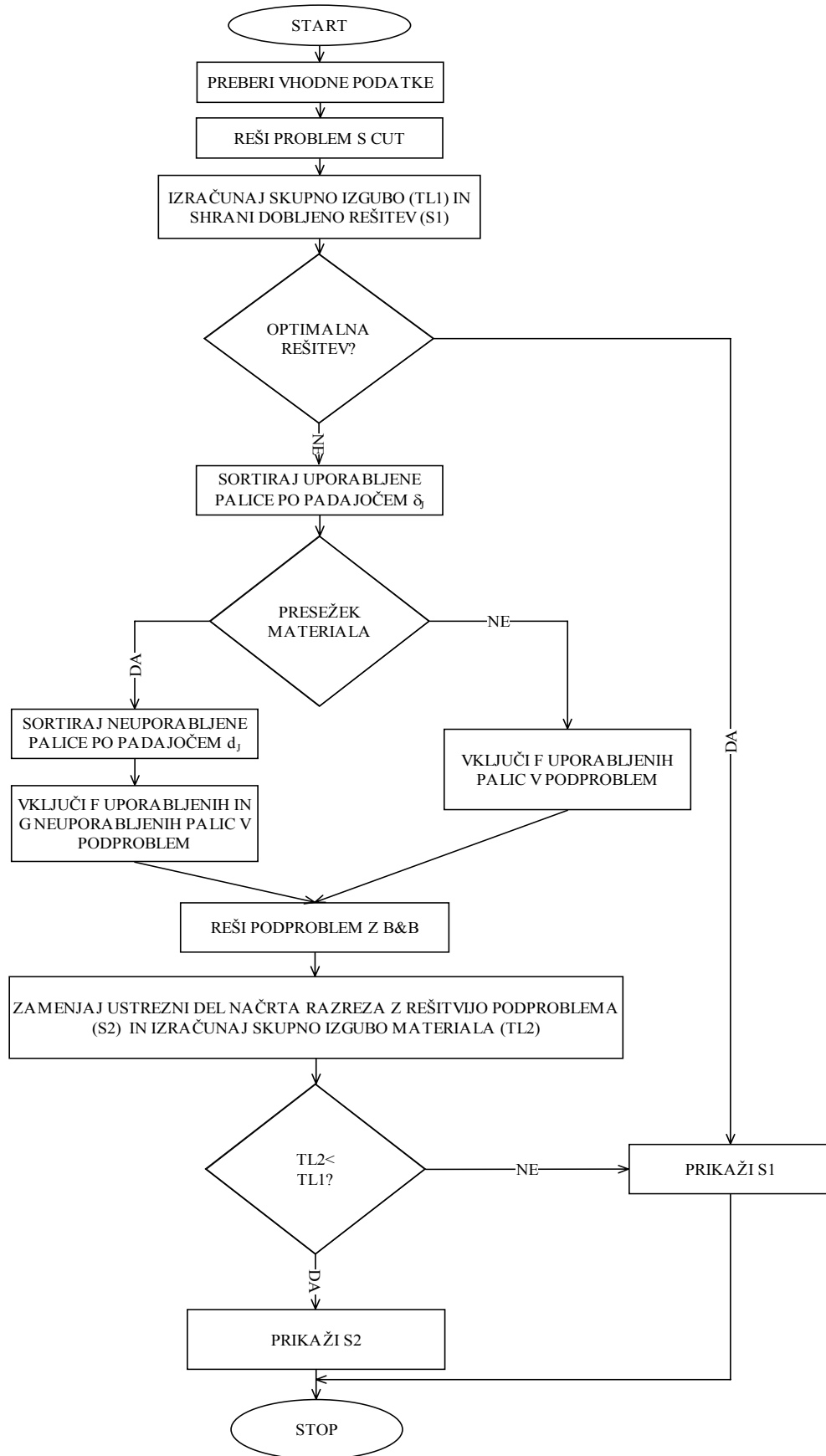
Način reševanja se ne razlikuje za primere s premalo in preveč materiala, moramo pa seveda uporabiti ustrezen model, ki je bil predstavljen že v petem poglavju. Za reševanje lahko uporabimo katerikoli razpoložljiv reševalec za celoštevilčno linearno programiranje.

Parametra f in g običajno določimo tako, da lahko pričakujemo optimalno rešitev ob danem časovnem limitu, kakovosti reševalca in procesorski moči računalnika. Kljub temu je možno, da v določenem času ne najdemo optimalne rešitve, ampak le neko začasno rešitev podproblema. V kolikor je ta začasna rešitev vseeno boljša od rešitve, ki jo je za podproblem našel CUT, obdržimo to rešitev, sicer pa kar hevrstično rešitev.

5. korak: zamenjaj ustrezen del načrta razreza z rešitvijo podproblema iz 4. koraka – seveda le v primeru, da je ta rešitev boljša od rešitve, ki jo je našel CUT. V nasprotnem primeru pa obdržimo rešitev iz prve faze.

Ta natančno opisan algoritem je prikazan tudi na sliki 11.

Slika 11: Prikaz algoritma metode C-CUT



Zaradi ustreznega algoritma, rešitev, ki jo pridobi C-CUT, zagotovo ustreza vsem omejitvam v modelu. V kolikor je bila najdena optimalna rešitev podproblema, je nova rešitev seveda vsaj tako dobra kot prejšnja. V nobenem primeru pa končna rešitev ne more biti slabša od rešitve prve faze. Na koncu še prikažemo boljšo od obeh rešitev

Če je podproblem v 4. koraku rešen optimalno, je nova rešitev (S_2) vsaj enakovredna obstoječi, običajno pa je kar precej boljša, kot je razvidno tudi iz nadaljevanja te doktorske disertacije. V nasprotnem primeru pa obdržimo staro rešitev (S_1).

Rešitev podproblema je v prvi vrsti (ob predpostavki, da so časovna omejitve, kvaliteta reševalca in procesorska moč uporabljenega računalnika dani in konstantni) odvisna od določitve parametrov f in g , torej števila palic, ki jih vključimo v podproblem. Večanje teh parametrov veča tudi kompleksnost problema in s tem verjetnost, da optimalne rešitve ne bomo našli v določeni časovni omejitvi.

Dodaten problem je, da bi bila ob zelo velikem obsegu podproblema (taki so denimo primeri 18, 24 in 27 iz tabele 4) tudi delna rešitev, pridobljena v določeni časovni omejitvi, bistveno slabša od hevristične in tako ne bi prispevala k izboljševanju rešitve kot celote. Zato moramo paziti, da obseg podproblema ne preseže tistih mej, ki še omogočajo rešitev podproblema, ki bo optimalna ali vsaj blizu temu – le take rešitve so namreč praviloma boljše od hevristične.

Zato morata biti f in g postavljena tako, da lahko optimalno rešitev pričakujemo s precejšnjo verjetnostjo⁴⁷. Povečanje parametra f po drugi strani tudi zmanjša razmerje med

$$\sum_{j=1}^m d_j \text{ in } \sum_{i=1}^n s_i * b_i, \text{ ki tudi vpliva na verjetnost pridobitve optimalne rešitve.}$$

Po drugi strani bi nizke vrednosti f in g pomenile, da rešitev podproblema ne bi mogla biti bistveno boljša od rešitve, ki jo je pridobil CUT, saj ne bi preostalo več dosti možnih kombinacij za izdelavo načrta razreza. Rešitev majhnega podproblema bo torej sicer hitra, vendar bo le malo ali pa sploh ne prispevala k izboljšanju rešitve.

Očitno je, da lahko postavimo $f = g = 0$ in v tem primeru se celoten problem reši hevristično z metodo CUT. Po drugi strani lahko postavimo tudi $f = p$ in $g = m - p$, kar pomeni, da v podproblem vključimo vse palice v zalogi in posledično tudi vse kose naročil, ki jih potrebujemo. Tako se celoten problem reši z eksaktno metodo. To kaže na to, da je C-CUT dejansko razširitev obstoječih metod za rešitev tega problema in ob ustrezni postavitvi parametrov omogoča uporabo samo ene od kombiniranih metod.

⁴⁷ Pri tej določitvi f in g seveda lahko koristno uporabimo spoznanja iz prejšnjega poglavja. Predhodno predstavljene analize so tako predpogoj za uspešno uporabo tovrstne kombinacije metod.

Kot že nekajkrat omenjeno, je izbor parametrov f in g odvisen od:

- značilnosti problema:
 - o razmerja med $\sum_{j=1}^m d_j$ in $\sum_{i=1}^n s_i * b_i$ – torej med celotnim materialom in celotnimi potrebami,
 - o razmerja med povprečno d_j in povprečno s_j , torej med povprečno dolžino palice in povprečno dolžino naročila,
- procesorske moči računalnika,
- časovne omejitve za rešitev v drugi fazi,
- kvalitete reševalca v drugi fazi.

Kot so dokazali eksperimenti, prikazani v prejšnjih poglavjih (vsi eksperimenti so se izvajali na istem računalniku in z uporabo istega reševalca), je ustrezna vrednost za vsoto parametrov f in g nekje med 5 in 10.

Pri večjem številu palic, vključenih v podproblem (na primer 15 vključenih palic ali več), bo rešitev, pridobljena z eksaktno metodo v eni minuti, veliko predaleč od optimalne in kot taka bistveno slabša od hevristične. Pri manjšem številu rešitev ne bo mogla biti bistveno boljša od rešitve pridobljene s CUT.

6.3.1 Rezultati C-CUT-a

Jasno je, da lahko trdim, da je predlagana metoda znatno boljša od obstoječih hevrističnih metod le ob obsežnem testiranju na različnih primerih. Zato sem metodo testiral na predhodno objavljenih problemih (Gradišar et al., 2002). V tem članku je bilo generiranih 270 problemov z generatorjem problemov PGEN. Problemi so bili rešeni tako s CUT kot s COLA algoritmom. Kot že omenjeno, nam uporaba generatorja PGEN ob poznavanju uporabljenega semena omogoča, da pripravimo enake probleme in jih nato rešimo. To sem uporabil tudi v spodaj prikazanih eksperimentih.

Podrobnosti o parametrih problemov in načinu za določanje semena so torej enake kot v (Gradišar et al., 2002) in so razvidne iz dinamične programske sheme procedure PROGEN, ki je prikazana na sliki 12.

Slika 12: Shema procedure PROGEN, uporabljena za pridobitev 270 podprimerov

Procedura PROGEN:

```

od i= 1 do 3
    od j= 1 do 3
        od k=1 do 3
             $n \leftarrow i * 5$ 
             $v_1 \leftarrow j * 100$ 
             $v_2 \leftarrow j * 300$ 
             $d \leftarrow k * 10$ 
             $m \leftarrow i * j * 10$ 
             $u_1 \leftarrow j * 500$ 
             $u_2 \leftarrow j * 1500$ 
             $seme \leftarrow n * 100000000$ 
             $seme \leftarrow seme + v_1 * 100000$ 
             $seme \leftarrow seme + v_2 * 1000$ 
             $seme \leftarrow seme + d * 100$ 
             $seme \leftarrow seme + m$ 
        r ← 1
        Ključ PGEN ( $n, v_1, v_2, d, m, u_1, u_2, seme, r$ )
        naslednji k
    naslednji j
naslednji i

```

Kot sem že ugotovil v tej doktorski disertaciji, lahko optimalno rešitev z eksaktno metodo pričakujemo, v kolikor je število palic manj kot 10. Zato sem v primerih, kjer je preveč materiala, vrednost spremenljivke f postavil na 6, g pa na 2. Dokazal sem že, da bi, v kolikor bi število palic v podproblemu preseglo 10, bila pridobljena rešitev z eksaktno metodo v dani časovni omejitvi zelo verjetno slabša od prvotne rešitve CUT-a. Zato nadaljnje povečevanje števila palic v podproblemu ni smiselno. Iz zgoraj navedenih razlogov ni smiselno niti zmanjševanje števila palic, saj prikazani eksperimenti kažejo na ustreznost izbranih nastavitvev.

V kolikor je število palic, kjer je $\delta_j > 0$, manjše kot 6, sem f zmanjšal na število palic z izgubo ostankom večjim od 0, hkrati pa ustrezno povečal g , tako da je bil enak $\min(8-f, \text{število neuporabljenih palic v prvotnem načrtu razreza})$. To se sicer običajno zgodi le v manjših primerih, kjer je lahko manj kot pet uporabljenih palic z ostankom. V primerih, kjer je materiala premalo, mora g očitno biti enak 0, tako da je f enak 8.

Praktični eksperiment, katerega rezultati so predstavljeni v naslednjih slikah in tabelah, je potrdil, da so bile take nastavitve za vrednosti f in g ustrezne in da opisana metoda dejansko vodi k manjši izgubi materiala. Časovna omejitev za drugo fazo algoritma je bila

60 sekund, pri čemer je bila večina primerov rešenih že v krajšem času. Celotni čas reševanja problemov se lahko v primerjavi s CUT poveča za največ 1 minuto, saj je prva faza algoritma identična CUT-u. Čas, porabljen za preostale faze algoritma (denimo določanje palic in naročil, ki bodo vključene v podproblem), je zanemarljiv. Povprečni časi za rešitev posameznega podproblema so prikazani tudi v nadaljevanju.

CUT algoritem je napisan v programskem jeziku FORTRAN. Program za branje rezultatov prve faze, obdelavo podatkov in pripravo podatkov za drugo fazo je napisan v programskem jeziku VBA. Omenjen program po potrebi omogoča odgovornemu za razrez tudi ročno izbiro palic, ki jih želi vključiti v podproblem za drugo fazo⁴⁸. V 4. koraku algoritma podproblem rešim z MPL/CPLEX programom, tako kot sem reševal že probleme v prejšnjem poglavju. Vsi koraki algoritma so bili izvedeni na istem računalniku kot vsi preostali testi v tej doktorski disertaciji (AMD, 1300 MHz).

V naslednjih slikah je prikazana rešitev drugega generiranega podproblema. Prvi podproblem je bil namreč optimalno rešen že v prvi fazi, torej samo s CUT algoritmom. Zato tu prikazujem rešitev drugega podproblema, ki je šel skozi obe fazi algoritma.

Na sliki 13 je prikazana rešitev 1. faze (torej rešitev pridobljena s CUT-om), na sliki 14 pa rešitev podproblema, pridobljena z eksaktno metodo.

Slika 13: Začetna rešitev, pridobljena s CUT algoritmom

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov
1	144	2
2	194	3
3	249	18
4	157	15
5	129	12

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
1	2663
2	1862
3	1805
4	1963
5	2461
6	1963
7	2196
8	1532

⁴⁸ Na pripravljenem obrazcu označi palice, za katere želi pripraviti nov načrt razreza.

9	1954
10	1158

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI NAROČIL

zap. št. naročila	zap. št. palice	št. kosov
1	9	1
1	1	1
2	7	2
2	1	1
3	7	6
3	8	3
3	9	6
3	1	3
4	2	2
4	7	2
4	8	5
4	9	2
4	1	4
5	2	12

PODATKI O OSTANKIH

indeks palice	ostanek (δ_j)
1	950
2	0
7	0
8	0
9	2

IZGUBA

zap. št. palice	izguba (t_j)
1	0
2	0
7	0
8	0
9	2

skupna izguba= 2 cm

V skladu z algoritmom in rezultati na sliki 13 so v 2. fazi v podproblem vključene palice z naslednjimi indeksi: 1, 3, 4, 5, 6, 9. Iz tega primera je tudi razvidna pomembnost sortiranja palic po padajočem δ_j in ne po t_j . Če bi palice sortiral po padajoči izgubi, potem palica 1 ne bi bila vključena v model, kar bi pomenilo, da bi celotna rešitev, ki bi jo ponudil algoritem

C-CUT, kršila omejitev 5. S takim izborom palic za drugo fazo sem zagotovil, da bo končna rešitev, tudi ko bom kombiniral rezultate 1. faze in optimalne rešitve podproblema, ustrezala vsem pogojem v modelu.

Slika 14: Rešitev podproblema v drugi fazi algoritma

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov
1	144	2
2	194	1
3	249	9
4	157	6

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
1	2663
3	1805
4	1963
5	2461
6	1963
9	1954
10	1158

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI NAROČIL

zap. št. naročila	zap. št. palice	št. kosov
1	3	1
1	5	1
2	3	1
3	3	4
3	5	5
4	3	3
4	5	3

OSTANKI / IZGUBA

zap. št. palice	ostanek (δ_j)	izguba (t_j)
3	0	0
5	601	0

Eksaktna rešitev podproblema v 2. fazi je bila najdena v 0,17 sekunde in je prikazana na sliki 14. S slike 14 so razvidne tudi skupne potrebe po posameznih dolžinah naročil v 2. fazi (b_i').

Dodam naj, da je pri eksaktni rešitvi podproblema ostanek pri palici 5 enak 601 centimetrov in torej ne šteje kot izguba, saj je daljši od prej določene zgornje meje ($UB = 249$ cm). Skupna izguba je torej enaka 0 cm, kar pomeni, da smo zagotovo našli optimalno rešitev. Boljša rešitev od tiste, ki jo je ponudil algoritem CUT, je bila najdena tako, da se je v načrtu razreza namesto palic z indeksom 1 in 9 uporabilo palice z indeksom 3 in 5, kar je omogočilo zmanjšanje izgube za 2 centimetra (oziroma za 100%).

Navedeni primer je namenjen predvsem za ilustracijo delovanja predlagane metode. Na podlagi enega samega primera seveda ne moremo sklepati o učinkovitosti ali uporabnosti metode.

Zato so v tabeli 10 predstavljeni združeni podatki za vseh 270 primerov. V vsaki vrstici tabele je skupna izguba ugotovljena kot vsota izgub pri 10 problemih, ki jih je generirala procedura PROGEN z istimi parametri. Povprečen odstotek izgube je izračunan kot povprečje desetih odstotnih izgub⁴⁹. Odstotne izgube so izračunane kot razmerje med izgubo materiala in skupno dolžino palic, uporabljenih v načrtu razreza.

⁴⁹ Posamične rešitve vseh 270 obravnavanih primerov so na razpolago pri avtorju doktorata.

Tabela 10: Primerjava rezultatov med metodo CUT in metodo C-CUT

zap. št.	n	v ₁	v ₂	d	m	u ₁	u ₂	seme	izguba CUT		izguba C-CUT	
									(cm)	(%)	(cm)	(%)
1	5	100	300	10	10	1000	3000	510301010	12	0,0161	9	0,0121
2	5	100	300	20	10	1000	3000	510302010	248	0,1424	129	0,0740
3	5	100	300	30	10	1000	3000	510303010	86	0,0427	35	0,0174
4	5	200	600	10	20	2000	6000	520601020	20	0,0126	14	0,0088
5	5	200	600	20	20	2000	6000	520602020	105	0,0287	84	0,0230
6	5	200	600	30	20	2000	6000	520603020	571	0,1120	523	0,1026
7	5	300	900	10	30	3000	9000	530901030	16	0,0070	7	0,0031
8	5	300	900	20	30	3000	9000	530902030	166	0,0313	118	0,0223
9	5	300	900	30	30	3000	9000	530903030	365	0,0482	332	0,0438
10	10	100	300	10	20	1000	3000	1010301020	8	0,0044	0	0,0000
11	10	100	300	20	20	1000	3000	1010302020	303	0,0795	212	0,0556
12	10	100	300	30	20	1000	3000	1010303020	52	0,0125	0	0,0000
13	10	200	600	10	40	2000	6000	1020601040	10	0,0029	1	0,0003
14	10	200	600	20	40	2000	6000	1020602040	18	0,0024	6	0,0008
15	10	200	600	30	40	2000	6000	1020603040	350	0,0313	292	0,0261
16	10	300	900	10	60	3000	9000	1030901060	5	0,0009	0	0,0000
17	10	300	900	20	60	3000	9000	1030902060	18	0,0015	10	0,0009
18	10	300	900	30	60	3000	9000	1030903060	324	0,0185	238	0,0136
19	15	100	300	10	30	1000	3000	1510301030	19	0,0070	0	0,0000
20	15	100	300	20	30	1000	3000	1510302030	276	0,0487	96	0,0169
21	15	100	300	30	30	1000	3000	1510303030	47	0,0077	12	0,0020
22	15	200	600	10	60	2000	6000	1520601060	13	0,0024	0	0,0000
23	15	200	600	20	60	2000	6000	1520602060	45	0,0039	11	0,0009
24	15	200	600	30	60	2000	6000	1520603060	126	0,0071	83	0,0047
25	15	300	900	10	90	3000	9000	1530901090	9	0,0011	2	0,0003
26	15	300	900	20	90	3000	9000	1530902090	11	0,0006	0	0,0000
27	15	300	900	30	90	3000	9000	1530903090	37	0,0014	22	0,0008
SKUPAJ									3260	0,0250	2236	0,0159

Skupno je CUT zagotovo našel optimalno rešitev v 70 primerih. Tako je v drugo fazo vstopilo 200 primerov. V teh 200 primerih je bila boljša rešitev najdena v 155 primerih, od tega je bila rešitev v 91 primerih zagotovo optimalna⁵⁰. Skupaj je bilo torej optimalno rešenih vsaj 161 od 270 primerov (oziroma 59,6% vseh). Boljša rešitev je bila torej najdena v več kot 75% prvotno neoptimalno rešenih primerih, skupno število optimalno rešenih problemov pa se je več kot podvojilo.

Za preostalih 109 primerov ne moremo z gotovostjo trditi, ali so rešeni optimalno. Že CUT ne daje indikacije o optimalnosti pridobljene rešitve. Zato tudi v kolikor je podproblem rešen optimalno, ne moremo trditi, da je optimalno zagotovo rešen celotni problem.

⁵⁰ Optimalno je bil rešen tako podproblem kot tudi celoten problem, vključno z delom rešitve, ki ni bil vključen v podproblem. Celotna izguba je torej 0.

Najpomembnejši kriterij pa je skupna izguba materiala. Kot je razvidno iz tabele, se je le-ta zmanjšala za 31,4% (iz 3260 centimetrov na 2236 cm). Iz vsega napisanega je jasno, da C-CUT privede do bistveno boljših rezultatov kot CUT, ob tem pa ohrani nizko časovno kompleksnost, ki je ena od pglavitnih prednosti CUT-a v primerjavi z eksaktno metodo. Čas, potreben za rešitev, se namreč v primerjavi s CUT poveča največ za 1 minuto, saj je le-ta postavljena kot limit za reševanje podproblema. Druge potrebne operacije v algoritmu (kot je na primer izbira palic, ki tvorijo podproblem) porabijo zanemarljivo malo časa. Zato lahko trdim, da je C-CUT očitno boljši od metode CUT ter da posledično vodi do nižjih stroškov preskrbe z materialom v oskrbovalni verigi. Ker je C-CUT boljši od CUT-a, je seveda tudi boljši od COLA-e.

Končna izguba materiala je nizka, saj v nobenem od primerov ne presega 0,11% uporabljenega materiala. Izguba materiala je višja v primerih, kjer je razmerje med razpoložljivim materialom in celotnimi potrebami blizu 1. Tri skupine primerov z najvišjo izgubo so tako primeri 6, 2 in 11, kjer je to razmerje enako 1,3; 1,0 ter 1,0. To je posledica dejstva, da ima v takih primerih CUT in posledično C-CUT manjši možni nabor potencialnih rešitev, saj se mora uporabiti oziroma narezati večina palic in dolžin naročila. Natančnejša statistična analiza dejavnikov, ki vplivajo na kvaliteto rešitve CUT, je sicer v (Gradišar et al., 1999).

Zaradi nizke časovne kompleksnosti je C-CUT tudi boljši od eksaktne metode za reševanje tega tipa problema, saj je primeren tudi za večje probleme s petnajst ali več palicami. Edino za reševanje zelo majhnih primerov je primernejša eksaktna metoda. Za to metodo lahko na podlagi rezultatov v tej disertaciji trdim, da je primerna predvsem za primere z manj kot približno osmimi palicami. Rešitve eksaktne metode za večje primere so že bistveno slabše od CUT-a.

Dokazal sem torej, da je C-CUT boljši od preostalih metod za splošen problem enodimenzionalnega razreza. Poleg primerjave z že omenjenimi metodami C-CUT-a žal ne morem neposredno primerjati z drugimi metodami za podobne primere. Čeprav so nekatere metode namenjene za podobne probleme (denimo Schilling, Georgiadis, 2002; Belov, Scheithauer, 2002), pa se definicije teh problemov od tu omenjenih razlikujejo preveč za neposredno primerjavo. Schilling in Georgiadis (2002) se tako ukvarjata s palicami enakih dolžin ali v skrajnem primeru z dvema skupinama standardnih dolžin, medtem ko v doktoratu obravnavam problem, kjer so lahko vse palice različne dolžine. Podobno Belov in Scheithauer (2002) preučujeta problem z več skupinami standardnih dolžin – uporabita metodo vzorcev, pridobljene rezultate pa zaokrožita z ustrezno hevristično metodo.

Zaradi kakovosti metode, predstavljene v tem poglavju, je le-ta v preteklih letih pritegnila tudi precej pozornosti v mednarodnih znanstvenih krogih, saj je med drugim kot prispevek pri optimizaciji enodimenzionalnega razreza materiala omenjena v (Aktin, Ozdemir, 2008), (Cherri, 2006), (Cherri et al., 2007), (Cherri et al., 2008), (Weng, Sung, 2008), (Tsai,

Huang, 2005), (Tsai, Hsieh, 2005a), (Sung et al., 2004), (Ghodsi, Sassani, 2005), (Shen et al., 2007), (Rodriguez, Vecchietti, 2008), (Saad et al., 2007) ter (Yang et al., 2006).

V tem poglavju sem torej predstavil novo metodo za reševanje splošnega problema enodimenzionalnega razreza, ki vodi do skoraj optimalnih rešitev ob ohranjanju nizke časovne kompleksnosti. Predstavljena metoda se lahko uporabi tako za primere s presežkom kot primanjkljajem materiala. S kombinacijo hevristične in eksaktne metode, kjer sem eksaktno metodo uporabil za rešitev manjšega podproblema, sem zmanjšal vpliv končnih pogojev. Na podlagi opisane metode sem razvil algoritem C-CUT, ki vodi do bistveno boljših rezultatov od obstoječih metod.

7 Model od upoštevanju stroškov neizpolnitve⁵¹

V prejšnjem poglavju razvita metoda vodi za v doktoratu obravnavane primere do tako majhnih izgub, da nadaljnje izboljšave verjetno niso smiselne. Zato se v tem poglavju podrobneje ukvarjam s problemom, ko razpoložljivi material v skladišču ne zadošča za izpolnitev vseh trenutnih naročil. Omenjeni problem sem sicer že obravnaval pri vseh do sedaj razvitih metodah, vendar nisem posebej opredelil stroškov neizpolnjevanja naročil in predvsem tega, katera naročila ostanejo neizpolnjena. Iskal sem pač rešitev s čim manjšimi izgubami pri razrezu.

V nadaljevanju poglavja se torej ukvarjam s primerom, ko material na zalogi ne zadošča za izpolnitev vseh naročil. Do takega primera lahko pride zaradi motenj v preskrbi z materialom v oskrbovalni verigi (nekateri potencialne razloge sem predstavil v drugem poglavju) ali pa zato, ker namenoma vzdržujemo zalogo konkurenčnih naročil, da bi bolje izkoristili material, ki sicer dolgoročno priteka v zadostnih količinah. Možno je tudi, da dovolj materiala določene vrste ni na mestu razreza in da ga je potrebno šele pripeljati iz oddaljenega skladišča. Ravno tako je možno, da zaradi različnih razlogov pride do nepričakovanega povečanja povpraševanja. Na sliki 4 (str. 20) je primer s premalo materiala vključen pri aktivnostih 21 in 23 v modelu procesa razreza.

V vsakem od teh primerov bomo torej potrebe po narezanih kosih izpolnili le delno. Pomembno vprašanje, ki smo ga do sedaj zanemarili, je, ali bomo nekaterim naročilom dali prednost.

Model, ki je predstavljen v nadaljevanju, omogoča, da pri odločitvi, katerim naročilom bomo dali prednost, upoštevamo tudi druge kriterije. To so predvsem prioriteta posameznih naročil, izguba materiala pri razrezu in dosednji čas čakanja na izpolnitev naročila. Po potrebi pa predstavljeni model omogoča tudi vključitev poljubnih novih dejavnikov.

⁵¹ Omenjeno poglavje je povzeto in razširjeno po (Trkman, Gradišar, 2003) ter (Trkman, 2002b).

V praksi lahko do vsaj kratkoročnega pomanjkanja materiala lahko pride pogosto. Podjetje se mora namreč običajno odločiti, kako visoko raven zalog v skladišču želi imeti. Visoka raven zalog povečuje stroške, povezane z zalogami, kot so na primer stroški skladiščenja, stroški vezanega kapitala v zalogah, pakiranje zalog, možne okvare, kraje, stroški zavarovanja in podobno (Thomas, 1980; Plossl, Wight, 1967) ter zmanjšuje fleksibilnost v oskrbovalni verigi.

Po drugi strani pa višja raven zalog zmanjšuje verjetnost, da v določenem obdobju nekaterih naročil ne bomo mogli sproti izpolnjevati in bo podjetje nosilo s tem povezane stroške. Tudi v primerih, ko je materiala dovolj, višja raven zalog poveča verjetnost, da bo možno najti načrt razreza z majhno izgubo, saj imamo tako na razpolago večje število kombinacij pri načrtu razreza. Držanje tako visoke ravni zalog, da bo v vsakem primeru zadostila potrebam po razrezu v večini primerov ni smiselno, saj je težko natančno vnaprej predvideti nihanja v povpraševanju kupcev. Številne možne pristope, ki jih za boljše upravljanje z zalogami nudi učinkovit management oskrbovalne verige, sem predstavil že v drugem poglavju. Problem določanja optimalne zaloge bi presegel okvire te disertacije – glej denimo (Tarim, Kingsman, 2004) za matematično analizo optimalne politike za osveževanje zaloge ob želeni ravni storitve za kupca ali (Trkman et al., 2007) za simulacijski pristop za določanje optimalne ravni zaloge v naftni industriji.

V primerih s premalo materiala je jasno, da bodo nekatera naročila ostala neizpolnjena. Do sedaj sem predpostavljal, da je edini cilj, da čim bolj zmanjšamo izgubo materiala, medtem ko je vseeno, katera naročila ostanejo neizpolnjena. V tem poglavju pa problem razširim s predpostavko, da so nekatera naročila pomembnejša od preostalih in bi njihova morebitna neizpolnitev za podjetje pomenila večji strošek kot neizpolnitev preostalih naročil. Omenjena predpostavka je dovolj logična in pogosta tudi v praksi, saj vsa naročila pri razrezu praviloma niso enako nujna in pomembna.

Poleg tega upoštevamo še čas čakanja posameznega naročila na izvedbo. S tem preprečimo, da bi v primeru konstantnega pomanjkanja materiala naročila z nizko prioriteto čakala na razrez neskončno dolgo. Menim, da so te predpostavke realne v marsikaterem praktičnem primeru.

Hkrati ne smemo pozabiti na osnovni cilj pri kreiranju vsakega načrta razreza – to je zmanjšanje izgube materiala. Za razliko od vseh dosedanjih modelov, razvitih v tej doktorski disertaciji, se torej sedaj srečujemo s trikriterijskim problemom, kjer imamo tri cilje:

- izpolniti predvsem tista naročila, ki so pomembnejša (z višjo prioriteto),
- doseči čim manjšo izgubo materiala,
- razrezati predvsem naročila, ki že dolgo čakajo na izvedbo.

Pri obravnavi tega problema uporabim podoben pristop, kot je predstavljen v (Wäscher, 1990), ki sem ga ustrezno popravil, nadgradil in prilagodil za obravnavani primer. V tem članku se minimizirajo celotni stroški razreza, v mojem primeru pa minimiziram celotne stroške neizpolnitve. Za vsako dolžino naročila ocenim, kakšne stroške povzroči neizpolnitev posameznega kosa tega naročila. To so lahko denimo pogodbeni kazni, stroški zastoja v nadaljnji proizvodnji, cena nakupa na trgu ali tudi nematerialni stroški (denimo izguba dobrega imena podjetja). Upoštevamo lahko tudi oportunitetne stroške neizpolnitve⁵², to so denimo izgubljeni prihodki.

Nato vrednost stroškov neizpolnitve pomnožimo s številom neizpolnjenih kosov posameznega naročila. Skupni stroški neizpolnitve so vsota stroškov neizpolnjenih naročil posameznih dolžin.

7.1 Model za reševanje

Ponovno predpostavljam, da so v skladišču palice različnih dolžin. To je lahko tudi posledica odločitve, da palice, ki jih ne razrežemo do konca, vračamo v skladišče, kot sem to prikazal v prejšnjem poglavju. Zaradi različnih dolžin palic ni možen pristop na podlagi vzorcev, zato tudi ta problem rešujem z uporabo celoštevilskega linearnega programiranja oziroma z metodo razveji in omeji.

Zaradi NP težkosti problema razreza ta metoda sicer ni primerna za večje probleme. Ker pa v primeru v tem poglavju rešujem problem razreza sproti s prihajanjem naročil, se celoten problem razdeli na več manjših delov, kar je tipičen primer pri pristopih, kot so dobava ob pravem času (angl. just-in-time – JIT) in neprestano načrtovanje materiala (CRP)⁵³. Dejansko nekatere študije kažejo, da se velikost naročil zmanjšuje (Wall, 2002). Vsakega od teh manjših delov pa lahko rešim optimalno z uporabo omenjenega pristopa, saj predpostavljam, da velikost posameznega problema ne presega meje, pri kateri je tak pristop še učinkovit.

⁵² Oportunitetne stroške lahko definiramo kot koristi, ki smo jih izpustili zaradi uporabe investicijskih virov v nekem projektu namesto v alternativni uporabi (Vitale, Giglierano, 2002). V tem primeru gre torej predvsem za izgubljeni prihodek zaradi prenizke investicije v zaloge in posledično prenizke ravni zalog.

⁵³ To je koncept kontinuirane dobave materiala, ki temelji na avtomatizirani izmenjavi informacij o povpraševanju, zalogah in upravljanju z zalogami v okviru dogovorjene nabavne politike. Cilj je doseči natančen in odziven pretok izdelkov z minimalnimi stroški zalog in transakcij (Applause Consulting, 2005). Temelji na medorganizacijskem informacijskem sistemu, ki poskuša izenačiti tok izdelkov s povpraševanjem kupcev ob hkratnih izboljšavah v logistiki in upravljanju z zalogami (Kopanaki, Smithson, 2002).

Ker sedaj ni vseeno, katera naročila ostanejo nenarezana, za kriterijsko funkcijo ne morem več uporabiti $\sum_{i=1}^n \delta_j$, kot sem lahko prej. Zato uporabim funkcijo $\sum_{i=1}^n \Delta_i * s_i$, pri čemer pa dolžino posameznega naročila zamenjam s stroški neizpolnitve tega naročila. Tako dobim nov izraz za kriterijsko funkcijo:

$$\sum_{i=1}^n \Delta_i * op_i$$

Stroške neizpolnitve lahko denimo izračunam po formuli⁵⁴:

$$op_i = b_i * (1 + m * \sqrt{\text{čas}_i}) * (1 + n * \text{prioriteta}_i)$$

S to formulo v bistvu popravimo vrednost b_i (dolžino posameznega naročila) za nek faktor, ki ga določim na podlagi časa čakanja na razrez in prioritete posameznega naročila. m in n sta faktorja, ki določata relativno prioriteto pri razrezu in s katerima lahko določamo, kaj nam je pri načrtu razreza pomembnejše (čim manjša izguba, čim boljša izpolnitev naročil z dolgim čakalnim časom ali čim boljša izpolnitev naročil z visoko prioriteto). Višja vrednost teh uteži pomeni večjo relativno pomembnost tega faktorja.

Čas čakanja se v vsakem naslednjem obdobju povečuje za 1, stroški neizpolnitve pa se povečujejo s korenom povečevanjem čakanja časa na izvedbo. Prioriteto lahko določimo arbitrarno na podlagi tega, kakšno škodo bi podjetju povzročila neizpolnitev posameznega naročila. Prioriteta je vključena iz jasnih razlogov – pomembna naročila želimo izpolniti čim prej, tudi če bo to imelo za posledico nekoliko večjo izgubo materiala.

Seveda je v enačbi za izračun stroškov neizpolnitve vključena tudi dolžina naročila – s tem zagotovim, da model upošteva tudi izguba materiala kot enega od faktorjev pri optimizaciji.

V tako opredeljeno enačbo lahko po potrebi dodajamo tudi druge faktorje, ki vplivajo na to, da je izpolnitev posameznega naročila pomembnejša od preostalih (kot na primer, če bi želeli prej izpolniti daljša naročila zaradi zmanjšanja izgub v kasnejših razrezih).

Jasno je, da vsako povečanje obeh ponderjev ali vključitev novih kriterijev v model, povečuje izgubo materiala, saj le-ta sedaj ni več edini faktor, ki vpliva na izdelavo načrta razreza. V primerjavi z modelom, predstavljenim v tretjem poglavju, je sedaj izguba materiala večja (ali v najboljšem primeru enaka), kot je bila v tistem modelu.

⁵⁴ To formulo lahko načeloma določimo arbitrarno, odvisno od tega kakšno pomembnost pripisujemo posameznim faktorjem, v obravnavanem primeru času in prioriteti posameznega naročila. Po potrebi bi lahko v formulo vključil tudi druge relevantne faktorje.

Seveda lahko določimo tudi $m = n = 0$. V tem primeru je vrednost kriterijske funkcije enaka kot v prejšnjih primerih in spet minimiziramo samo izgubo materiala.

Konkreten zapis zgornje formule bi bil lahko tudi poljubno drugačen, vključili bi lahko tudi druge faktorje. Cilj je, da tako izračunani stroški čim natančneje odražajo dejanske stroške neizpolnitve posameznega naročila v podjetju. Nadaljnji postopek reševanja je enak ne glede na uporabljeno formulo za izračun stroškov neizpolnitve.

Seveda moram ustrezno popraviti tudi omejitve 3, tako da dobim naslednji model:

(1) $\min \sum_{i=1}^n \Delta_i * op_i$ (minimiziraj število nenarezanih kosov, ponderirano s stroški neizpolnitve)

s pogojem, da

(2) $\sum_{i=1}^n (s_i * x_{ij}) < d_j \quad \forall j$ (nahrbtnik omejitev)

(3) $\sum_{j=1}^m x_{ij} + \Delta_i = b_i \quad \forall i$ (omejitve povpraševanja – posamezne dolžine naročila ne

smemo narezati več, kot je povpraševanje po njej)

(4) $x_{ij} \geq 0$, celoštevilčni $\forall i, j$
 $\Delta_i \geq 0 \quad \forall i$.

Na ta način sem torej spet prišel do običajnega enokriterijskega problema, ki ga lahko rešim s poljubnim reševalcem za celoštevilsko linearno programiranje.

7.2 Rezultati

Uporabo razširjenega modela razreza prikazujem na konkretnem primeru. Tako kot prejšnji primeri je bil tudi ta rešen z uporabo programa MPL/CPLEX na osebнем računalniku (AMD, 1300 MHz).

Vrednost ponderjev m in n je bila postavljena na 0,3, predstavljam pa razrez v dveh zaporednih obdobjih, kjer je bilo obakrat premalo materiala. Vsi podatki o dolžinah palic, dolžinah, potrebnem številu, prioriteti in čakalnem času posameznih naročil ter številu narezanih kosov so predstavljeni na sliki 15 (za 1. obdobje) ter sliki 16 (za 2. obdobje). Tudi podatke za ta primer sem pridobil z generatorjem naključnih problemov PROGEN (Gradišar et al., 2002).

Slika 15: Rešitev problema enodimenzionalnega razreza ob upoštevanju stroškov neizpolnitve v 1. obdobju

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov	čakalni čas	prioriteta	stroški neizpolnitve
1	144	22	2	1	266,6222
2	194	11	2	1	359,1994
3	249	29	0	1	323,7000
4	157	37	0	3	298,3000

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
1	2663
2	1805
3	2461
4	1963

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI PALICE

zap. št. palice	zap. št. naročila	št. kosov
1	1	1
1	4	16
2	1	9
2	2	1
2	4	2
3	1	4
3	4	12
4	1	6
4	4	7

IZGUBA

zap. št. palice	izguba
1	7
2	1
3	1
4	0

REALIZACIJA

zap. št. naročila	načrt	realizacija	razlika
1	22	20	2
2	11	1	10
3	29	0	29
4	37	37	0

V drugem obdobju se čas čakanja pri naročilih, ki so preostala iz prvega obdobja, poveča za 1, v proizvodnjo pa pride nov delovni nalog za izpolnitev naročila (13 kosov dolžine 188 s prioriteto 2 in časom čakanja 0), poleg tega v skladišče pridejo še nove palice (vsi podatki so razvidni tudi s slike 16).

Slika 16: Rešitev problema enodimenzionalnega razreza ob upoštevanju stroškov neizpolnitve v 2. obdobju

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov	čakalni čas	prioriteta	stroški neizpolnitve
1	144	2	3	1	284,4720
2	194	10	3	1	383,2470
3	249	29	1	1	420,8100
5	188	13	0	2	300,8000

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
5	2518
6	1638
7	2019
8	1791

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI PALICE

zap. št. palice	zap. št. naročila	št. kosov
5	1	1
5	2	1
5	3	8
5	5	1
6	1	1
6	3	6
7	2	3
7	3	5
7	5	1
8	2	6
8	3	1
8	5	2

IZGUBA

zap. št. palice	izguba
5	0
6	0
7	4
8	2

REALIZACIJA

zap. št. naročila	načrt	realizacija	razlika
1	2	2	0
2	10	10	0
3	29	20	9
5	13	4	9

Skupna izguba materiala je torej 15 cm (9 cm v prvem ter 6 cm v drugem obdobju). Če bi problem reševal brez upoštevanja stroškov neizpolnitve bi prišli do izgube 0 cm, pri čemer bi za razliko od predstavljenih rezultatov na slikah 15 in 16 narezali nekaj več kosov naročila 3 in 5. Zato pa bi slabše izpolnili naročila 1 in predvsem 4, ki ima med vsemi največjo prioriteto. Vidimo, da se je izguba materiala povečala le za nekaj centimetrov, saj 15 cm predstavlja manj kot 0,09 % celotnega materiala. To verjetno v marsikaterem praktičnem primeru ni previsoka cena za boljšo izpolnitev preostalih dveh ciljev. S povečanjem ponderjev m in n bi dosegli še boljše izpolnjevanje teh ciljev, seveda ob še nekoliko višji izgubi.

Možno pa je ta model razširiti tudi tako, da se izguba materiala ne poveča, hkrati pa vendarle upoštevamo tudi druge faktorje, ki so podrejeni glavnemu cilju. Dostikrat je namreč možno, da obstaja več različnih rešitev istega problema z isto izgubo. Brez dodatnega kriterija se naključno izbere ena od ekvivalentnih rešitev.

Ko govorim o naključnem izboru, je seveda potrebno poudariti, da je algoritem, po katerem deluje razveji in omeji, determinističen in (ob istih podatkih in istih nastavitvah za izbor vozlov, ki jih bomo razmejili, ter drugih nastavitvah v reševalcu) vedno privede do iste rešitve. Z naključnim izborom mislim na to, da nobena od teh rešitev z isto izgubo materiala ni favorizirana in da je bila do sedaj odločitev o izboru med ekvivalentnimi rešitvami prepuščena programu. Sedaj pa z ustrezno spremembo modela zagotovimo, da bo v primeru obstoja več rešitev z enako izgubo, vedno izbrana tista rešitev, ki bolje zadovoljuje preostale cilje.

Če želim zagotoviti, da se izguba ne bo povečala niti za centimeter, mora biti razlika med stroški neizpolnitve in dejansko dolžino posameznega naročila tako majhna, da bodo preostali faktorji upoštevani le pri izbiri med rešitvami z isto izgubo materiala. Ta razlika mora biti torej tako majhna, da nobena rešitev A , ki ima večjo izgubo kot rešitev B , ne bo imela boljše vrednosti nove kriterijske funkcije od rešitve B , dodatni kriteriji pa se bodo uporabili le, če bosta imeli obe rešitvi isto izgubo. Povedano drugače: veljati mora naslednje:

$$\left(\sum_{j=1}^m \delta_{jA} > \sum_{j=1}^m \delta_{jB} \right) \Leftrightarrow \left(\sum_{i=1}^n \Delta_{iA} * op_i > \sum_{i=1}^n \Delta_{iB} * op_i \right)$$

V zgornji enačbi se indeks A nanaša na vrednosti spremenljivk pri rešitvi A , indeks B pa na vrednosti pri rešitvi B (stroški neizpolnitve so pri obeh rešitvah praviloma izračunani na enak način).

Izpolnitev navedenega pogoja najlažje dosežemo tako, da faktorja m in n nastavimo tako nizko, da bo veljalo⁵⁵:

$$\sum_{i=1}^n |(b_i - op_i) * s_i| < 1$$

Z izpolnitvijo tega pogoja smo dosegli, da bo doprinos razlike med stroški neizpolnitve in dolžino posameznega naročila skupno manjši od 1. Tako nobena rešitev, ki je po modelu za primere s premalo materiala (brez upoštevanja stroškov) slabša od druge, od te rešitve ne more biti boljša po vključitvi stroškov. Če je ena rešitev slabša od druge, to namreč pomeni, da je slabša vsaj za 1, saj so vrednosti kriterijske funkcije pri modelu brez upoštevanja stroškov celoštevilčne.

S tem smo dosegli želeno: med ekvivalentnimi rešitvami po izgubi materiala ne izbiramo več naključne ampak tisto, ki bolje izpolnjuje dodatne pogoje (kar smo omenili že prej, velja tudi tu – ti dodatni pogoji so lahko tudi kaj drugega kot čas čakanja na razrez in prioriteta posameznega naročila). Nekaj praktičnih preizkusov kaže na to, da tak pristop dejansko privede do drugačne rešitve ob isti izgubi materiala in nekoliko daljšem času reševanja.

Eden od preprostih primerov je predstavljen na sliki 17 (podatki za ta primer), sliki 18 (rešitev brez upoštevanja stroškov neizpolnitve) in sliki 19 (rešitev ob upoštevanju stroškov neizpolnitve). Oba ponderja m in n sta bila postavljena na 0,0000002. Vrednost zgornjega pogoja je sedaj 0,0144, torej manj kot 1.

Kot je razvidno iz obeh slik, je izguba v obeh primerih 0 cm, tako kot seveda tudi izguba vsake posamezne palice. Vendar je pri drugi rešitvi narezano bistveno več (21 namesto 15) kosov dolžine številka 4, ki ima najvišjo prioriteto (ter posledično manj kosov dolžine naročila številka 1, ki ima precej nižjo prioriteto). To je posledica različnega razreza tretje palice, medtem ko je bil pri obeh rešitvah za prvi dve palici najden isti načrt razreza.

⁵⁵ Spodnji pogoj je zadosten, ne pa tudi potreben, za izpolnitev zgornje zahteve.

Slika 17: Podatki za primer

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov	čakalni čas	prioriteta	stroški neizpolnitve
1	144	22	1	1	144,0001
2	194	11	1	1	194,0001
3	249	29	0	1	249,0000
4	157	37	0	8	157,0003

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
1	2663
2	1805
3	2461

Slika 18: Rešitev brez upoštevanja stroškov neizpolnitve (model za premalo materiala)

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI PALICE

zap. št. palice	zap. št. naročila	št. kosov
1	1	4
1	2	3
1	3	1
1	4	8
2	1	1
2	2	1
2	3	4
2	4	3
3	1	11
3	2	0
3	3	1
3	4	4

REALIZACIJA

zap. št. naročila	načrt	realizacija	razlika
1	22	16	6
2	11	4	7
3	29	6	23
4	37	15	22

Slika 19: Rešitev ob upoštevanju stroškov neizpolnitve

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI PALICE

zap. št. palice	zap. št. naročila	št. kosov
1	1	4
1	2	3
1	3	1
1	4	8
2	1	1
2	2	1
2	3	4
2	4	3
3	1	1
3	2	0
3	3	3
3	4	10

REALIZACIJA

zap. št. naročila	načrt	realizacija	razlika
1	22	6	16
2	11	4	7
3	29	8	21
4	37	21	16

V tem poglavju sem torej prikazal rešitev splošnega problema enodimenzionalnega razreza, kjer so palice različnih dolžin; zaradi prenizke ravni zalog v skladišču, težav v oskrbovalni verigi ali hitrega povečanja povpraševanja kupcev pa količina razpoložljivega materiala ne zadošča za izpolnitev vseh naročil. Zato je potrebno sprejeti odločitev, katera naročila bodo ostala neizpolnjena. Problem bi lahko rešili s povečanjem zalog, kar pa bi povečalo druge stroške; to povečanje bi v marsikaterem primeru preseglo prihranek zaradi boljšega izpolnjevanja naročil.

Zato je bila v tem poglavju predstavljena metoda, ki omogoča vključevanje različnih dejavnikov pri sprejemanju odločitve, katera naročila bodo ostala neizpolnjena. Vpeljava stroškov neizpolnitve omogoča preoblikovanje več kriterijev v enega, ki ga nato uporabimo za rešitev problema razreza z običajnimi metodami.

Pomembna omejitev tega poglavja je, da sem novi pristop prikazal le na nekaj primerih. To pomeni, da tak pristop ni nujno uporaben za različne vhodne parametre ali želene cilje. Omejitev sem delno zmanjšal z matematičnim dokazom, da tak pristop (ob ustreznih nastavitvi parametrov) ne more privedi do višje izgube materiala kot v predhodnih poglavjih obsežno testirane metode, lahko pa privede do boljše izpolnitve preostalih ciljev.

8 Optimizacija kontinuiranega razreza⁵⁶

V vseh dosedanjih poglavjih sem reševal problem, kjer je bilo potrebno v danem trenutku čim boljše zadovoljiti trenutne potrebe iz razpoložljivega materiala. Taka situacija je večinoma le približna slika realnosti. Kot že omenjeno v drugem poglavju, je v večini praktičnih primerov proces razreza nenehen proces, ki poteka v podjetju oziroma oskrbovalni verigi. Tako obravnavanje razreza pa nudi nove smeri raziskovanja predvsem pri spremembi iz obravnavanja razreza kot enkratne aktivnosti (kar je bil primer v večini predhodnih raziskav) v zavedanje, da je razrez poslovni proces, ki se praviloma ponavlja kontinuirano in je tesno povezan s preostalimi procesi v podjetju (kot sem nakazal že na sliki 1).

Po eni strani namreč v skladišče prispevajo nove zaloge, po drugi strani pa tudi nova naročila, ki jih je potrebno sproti izpolnjevati. Podjetje ima običajno tudi možnost vplivati na naročene palice in njihovo dolžino. Prednosti tesnejšega povezovanja z dobavitelji in kupci ter posledično večjo možnost vplivanja na vhodne in izhodne podatke za podjetje sem predstavil v drugem poglavju. V tem poglavju se ukvarjam predvsem z razrezom kot neprestanim procesom v podjetju. Tak proces razreza sem poimenovali kontinuiran razrez⁵⁷ (angl. consecutive cutting).

Omenjeno različico kontinuiranega razreza lahko torej definiramo kot stalni, periodični poslovni proces razreza, kjer mora podjetje v vsakem obdobju zadovoljiti povpraševanje po narezanem materialu iz razpoložljivega materiala. Za vsako naslednje obdobje je raven zalog obnovljena z določenim številom novih palic. Natančni podatki o prihodnjem povpraševanju (torej potrebne dolžine in število kosov posamezne dolžine) so neznani ali je kvečjemu znana le njihova verjetnostna porazdelitev.

Podjetje torej išče rešitev problema razreza, ki ne bo izboljšal/optimiziral proizvodnje ter zmanjšal izgube le v enem časovnem trenutku, ampak bo vplival na vsoto vseh rezultatov v celotnem relevantnem obdobju. Neuporabljen material iz enega razreza je shranjen za naslednji razrez, če je daljši od predhodno določene zgornje meje (*UB*). Krajši ostanki štejejo kot izguba. Nastavitev zgornje meje, tako kot že v predhodnih modelih, preprečuje nabiranje večjega števila zelo kratkih palic, ki ne bi bile uporabne za bodoče razreze. Odvisno od ciljev metode lahko optimiziramo samo izgubo materiala ali pa tudi druge cilje (denimo stroške ponovne uporabe in priprave na razrez, stroške skladiščenja, prioriteto posameznih naročil...).

⁵⁶ Poglavje je povzeto in razširjeno po (Trkman, Gradišar, 2005; 2007a).

⁵⁷ Pri čemer kontinuiranega razreza ne gre mešati z zveznimi spremenljivkami (angl. continuous). Razrez se izvaja v zaporednih časovnih obdobjih, denimo dnevno, tedensko. Ker želim poudariti stalnost tega procesa, pa sem to vrsto razreza poimenoval z besedo kontinuiran.

Odvisno od praktičnega primera lahko razrez poteka dnevno ali celo pogosteje (denimo nekajkrat na dan), tedensko ali mesečno. Daljše časovno obdobje je lahko mesec, četrletje, leto... Praktična uporaba se nekoliko razlikuje od primera do primera, osnovni koncepti pa ostajajo nespremenjeni.

Kot že predhodno opisani modeli, je tudi model v tem poglavju še posebej koristen, kadar so stroški materiala relativno visoki v primerjavi z drugimi stroški. Takrat prihranki zaradi ponovne uporabe delno narezanega materiala presežejo morebitne dodatne stroške zaradi povečanega časa za razrez in dodatnega dela in stroškov s shranjevanjem daljših ostankov.

V marsikateri praktični situaciji namreč lahko velja, da so stroški zalog in stroški priprave na razrez (denimo vračanje delno uporabljenih palic v skladišče) večji od stroškov izgube materiala. V takem primeru vse delno uporabljene palice preprosto zavržemo. To še posebej velja, kadar je nezaželeno skladiščenje palic različnih dolžin, ker bi to bistveno povečalo stroške priprave razreza.

Zato je predstavljena metoda ustrezna za primere, kjer so stroški materiala in izgube relativno visoki v primerjavi z drugimi stroški. Prihranki zaradi ponovne uporabe delno narezanega materiala v takem primeru presežejo dodatne stroške, ki pri tem nastanejo. Na primernost predstavljene metode lahko poleg stroškov vračanja vplivajo tudi stroški skladiščenja. Vsaj v primeru, da je čas med posameznimi razrezi kratek (denimo ob dnevnem razrezu), pa lahko te stroške zanemarimo. Uporaba IT ter avtomatizacije proizvodnje lahko pomembno zmanjša stroške priprave razreza, zato lahko v bodoče pričakujemo rast pomena metod, ki bodo omogočale shranjevanje in ponovno uporabo delno narezanega materiala. Metodologija, ki sem jo predstavil v poglavju 2.5., pa omogoča tako merjenje kot zmanjševanje stroškov posamezne izvedbe procesa (za primer uporabe predstavljene metodologije glej Erjavec et al., 2008).

Kontinuiran razrez v taki obliki predhodno še ni bil definiran. Ravno tako v svetovni literaturi nisem zasledil podobnega problema in pristopa k reševanju, kot je predstavljen v nadaljevanju. Nekateri avtorji so obravnavali nekoliko podobne probleme, ki predstavljajo delni prehod od trenutne h kontinuirani optimizaciji. Nekateri predhodne metode tako že vsebujejo nekaj podobnih značilnosti, delno sem jih predstavil tudi v predhodnih poglavjih.

V (Arbib et al., 2002; Arbib, Marinelli, 2005) se avtorji ukvarjajo s problemom izdelave trakov za varnostne pasove v avtomobilski industriji, pri čemer so glavni cilji zmanjšanje izgube materiala, visoka kvaliteta, enakomerna obremenitev proizvodnih strojev ter zmanjšanje stroškov priprave na razrez. Osnovni pristop je prehod iz dnevnega v tedensko načrtovanje ter delna integracija procesov nabave in proizvodnje. Pripravi se tedenski načrt razreza, ki se potem razdeli po dnevih. Pri tem so razpoložljiv material in potrebe za cel teden znane vnaprej. Upoštevajo tudi ponovno uporabo delno narezanega materiala, pri čemer pa je možno lepljenje takih manjših delcev. Stroške, povezane z razrezom, so s

takim pristopom uspeli zmanjšati za 43% v primerjavi s kratkoročnim, dnevnim načrtovanjem.

Možnost ponovne uporabe delno narezanega materiala obravnavajo tudi v (Johnson et al., 1997), kjer pa se ukvarjajo predvsem z možnostjo lepljenja materiala, tako da se z optimizacijo v zaporednih časovnih obdobjih ne ukvarjajo.

Praktično vse preostale metode, predstavljene tudi v tretjem poglavju disertacije, se ukvarjajo s trenutno rešitvijo problema. Nekatere metode, ki predstavljajo delen prehod k obravnavanju razreza kot poslovnega procesa, sem predstavil tudi v 2. poglavju.

V prihodnje lahko pričakujemo povečevanje pomena kontinuiranega razreza iz več razlogov. Čedalje bolj se uveljavljajo različni pristopi neprekinjenega dopolnjevanja zalog (denimo CRP). Zaradi tega posamezna naročila postajajo manjša (Wall, 2005), se pa periodično ponavljajo. Glavno težišče proizvodnje tako niso več materiali, ampak čas (Pagel, 1999). Manjšanje posameznih naročil je zato pritegnilo veliko pozornost pri optimizaciji nabave (denimo pri naročanju ob pravem času), saj izboljšuje delovanje sistema z zmanjšanjem zalog in odpadka, manjšimi stroški nadzora, hitrejšim zaznavanjem napak in podobnim (Kim, Ha, 2003), vendar pri problemu razreza ta vidik praviloma ni bil upoštevan.

Tesnejše sodelovanje med dobavitelji in kupci torej pomeni, da mora dobavitelj večkrat zaporedno izpolnjevati naročila. Ravno tako lahko velikost posameznega naročila zmanjšuje uporaba množičnega prilagajanja. Uporaba dobave ob pravem času še dodatno zmanjšuje možnost čakanja in združevanja (angl. batch processing) in skupnega izpolnjevanja naročil po narezanem materialu ter zahteva metodo, primerno za izvajanje kontinuiranega razreza.

Ravno tako se, kot sem obrazložil že v drugem poglavju, v zadnjih letih povečuje pomen medorganizacijskih procesov in integracije oskrbovalne verige, kjer je proizvodnja (in posledično razrez) eden od kontinuiranih procesov, ki jih je potrebno integrirati. Metoda, predstavljena v nadaljevanju poglavja, je zato načeloma primerna za uporabo ob tovrstni integraciji.

Zaradi pričakovanega naraščanja pomena teh metod bi bila koristna klasifikacija, ki bi omogočala razlikovanje med metodami, ki vodijo do trenutnega optimuma, ter metodami, ki upoštevajo daljše časovno obdobje. Obstoječi klasifikaciji (Dyckhoff, 1990; Wäscher et al., 2007) v ta namen nista primerni. Zato predlagam razširitev Dyckhoffove klasifikacije z novim razredom.

Razred: Obdobje optimizacije (angl. time-span of optimization⁵⁸)

- IN – trenutna (angl. instantaneous) optimizacija – pomeni reševanje problema razreza ali pakiranja ob danih podatkih za eno časovno obdobje,
- CD – kontinuirana deterministična (angl. consecutive deterministic) – kaže optimizacijo razreza v več zaporednih časovnih obdobjih, pri čemer pa so potrebe in razpoložljiv material znane vnaprej. Neuporabljen material iz predhodnih obdobjih se lahko ponovno uporabi. Poseben primer tega tipa je, kadar lahko nekatere narezane dolžine iz predhodnega obdobja shranimo, dokler ni taka dolžina zahtevana v enem od naslednjih obdobjih,
- CS – kontinuirana stohastična (angl. consecutive stochastic) – bodoči material in potrebe niso znane vnaprej ali pa je znana le njihova verjetnostna porazdelitev. Neuporabljen material iz predhodnih obdobjih se lahko ponovno uporabi.

Za prvi in drugi tip problema se lahko uporabijo eksaktne ali hevristične metode. Za tip CS je za celo obdobje možna edino hevristična rešitev, saj bodoča povpraševanja niso znana. Zato je potrebno reševanje problema zastaviti tako, da eksaktne ali hevristične rešitve problemov v posameznem trenutku vodijo tudi do dobrih rešitev v celotnem obdobju.

Problem kontinuiranega razreza, ki ga predstavljam v nadaljevanju, torej pripada razredu 1/V/D/M/CS po tej razširjeni tipologiji. Vsi predhodno razviti modeli v tej disertaciji spadajo v razred 1/V/D/M/IN. Zgoraj opisani problem iz avtomobilske industrije (Arbib et al., 2002; Arbib, Marinelli, 2005) pa je tipičen primer tipa CD.

Podobno bi morali razširiti tudi klasifikacijo, objavljeno v (Wäscher et al., 2007). Pri tipu problema bi tako poleg minimizacije inputa in maksimizacije outputa morali kot tretji tip problema dodati tudi maksimizacijo poslovne vrednosti izvajanja procesa razreza za posamezno podjetje. V ta sklop bi nato uvrstili različne verzije problema kontinuiranega razreza ter probleme, povezane s prenovo procesa razreza.

V vseh do sedaj predstavljenih modelih v predhodnih poglavjih sem predpostavljal, da lahko pri vsakem razrezu v skladišče vrnemo največ eno palico, ki presega neko vnaprej določeno zgornjo mejo. Taka palica je uporabna pri kasnejših razrezih in zato ne šteje kot izguba. V prejšnjih poglavjih je bila ta omejitev potrebna, saj metod nisem testiral v zaporednih časovnih obdobjih in se je tako predvsem preprečevalo preveč požrešno vedenje algoritma.

Sedaj pa ta omejitev ni več potrebna, saj bom izvedli testiranje v več zaporednih časovnih obdobjih. Ker je lahko dostikrat koristno v skladišče vrniti več kot eno palico, sem to omejitev opustili, tako da lahko hkrati vrnemo več palic. Posebej torej želim preučiti

⁵⁸ Prevodi v angleščino so povzeti po (Trkman, Gradišar, 2007a).

možnost ponovne uporabe delno narezanega materiala. Lepljenje materiala ali drugi načini združevanja delno uporabljenega materiala pa ne pridejo v poštev.

Osnovni razliki v primerjavi z do sedaj predstavljenimi modeli v doktorski disertaciji sta torej:

- model testiram v zaporednih časovnih obdobjih; zanima me skupni rezultat (seštevek) vseh obdobj,
- v skladišče lahko vrnemo poljubno število delno uporabljenih palic – seveda pod pogojem, da so dovolj dolge (daljše od določene zgornje meje). Do sedaj smo lahko v skladišče vračali samo eno palico.

V tem delu sicer uporabljam prilagojeno metodo, predstavljeno v prejšnjih poglavjih. Lahko bi uporabil tudi druge metode, denimo (Schilling, Georgiadis, 2002), vendar bi jih bilo potrebno predhodno prilagoditi tako, da bi omogočale rešitev splošnega problema razreza ter vračilo ene ali več delno narezanih palic. Zato sem uporabil lastne metode, ki so vsaj implicitno že predvidevale ponovno uporabo delno narezanih palic. Le-to je bilo v teh metodah že omogočeno, vendar pa posledice vračanja palic za razreze v kasnejših obdobjih niso bile preučevane.

8.1 Kontinuiran razrez – model 1

Najprej predstavim in testiram model, ki je zelo podoben tistemu iz četrtega oziroma petega poglavja. Poglavitna razlika je, da sedaj lahko vrnem več palic. Zato tudi niso potrebne večje spremembe v zapisu modela.

Pri kontinuiranem razrezu v nadaljevanju tega poglavja uporabljam naslednja dva izraza:

- posamezno obdobje: trenutek, ko moramo z razpoložljivim materialom kar najbolje izpolniti dano naročilo,
- celotno relevantno obdobje: to je celotno obdobje optimizacije in je sestavljeno iz več zaporednih posameznih obdobj.

V model uvedem naslednjo novo spremenljivko⁵⁹:

- g_j – kaže, ali je palica j večja od zgornje meje (UB)

Omenjena spremenljivka nadomešča predhodno uporabljano u_j . Namen nove oznake je predvsem poudarek osnovne razlike – sedaj je lahko od UB daljših več palic, ki jih vse vrnemo v skladišče.

⁵⁹ Poleg tega v modelu uporabljam tudi večino spremenljivk, ki sem jih opredelil že pri osnovnem modelu; nekatere tam opredeljene spremenljivke pa niso več potrebne. Spremenljivke so definirane na str. 68.

Celotni model za posamezno obdobje lahko sedaj zapišem kot:

$$(1) \quad \min \sum_{j=1}^m t_j \quad (\text{minimiziraj vsoto ostankov, ki so pod } UB; \text{ torej štejejo kot izguba})$$

s pogojem, da

$$(2) \quad \sum_{i=1}^n (s_i * x_{ij}) + \delta_j = d_j + (1 - y_j) \quad \forall j \quad (\text{omejitev nahrbtnika})$$

$$(3) \quad \sum_{j=1}^m x_{ij} = b_i \quad \forall i \quad (\text{omejitve povpraševanja – število potrebnih kosov je fiksno})$$

$$(4) \quad UB - \delta_j + UB(g_j - 1) \leq 0 \quad \forall j \quad (g_j \text{ kaže, ali je preostanek palice } d_j \text{ daljši od } UB \text{ in je enak 1 samo, če je preostanek daljši od te meje})$$

$$(5) \quad \delta_j - t_j - (g_j + y_j) * (\max d_j) \leq 0 \quad \forall j \quad (t_j \text{ kaže, kakšna je izguba pri palici } j, \text{ in je enak } \delta_j \text{ samo pri palicah, ki so krajše od } UB; \text{ pri palicah daljših od } UB, \text{ pa je enak 0})$$

$$(6) \quad \min d_j \geq UB \geq \min s_i$$

$$x_{ij} \geq 0, \text{ celoštevilčni} \quad \forall i, j$$

$$t_j \geq 0 \quad \forall j$$

$$\delta_j \geq 0 \quad \forall j$$

$$g_j \in \{0,1\} \quad \forall j$$

Seveda se spremeni samo model za razrez, kadar je dovolj materiala. V primeru, da materiala ni dovolj, postavljanje zgornje meje ni smiselno, saj moramo za vsaj delno izpolnitev naročila v vsakem primeru razrezati ves razpoložljivi material.

CUT in posledično C-CUT sta kot večina hevrističnih metod izrazito specializirana in prilagojena za v prejšnjih poglavjih opisan problem, ko lahko vračamo v skladišče samo eno palico. Zato v taki obliki nista primerna za tako spremenjen problem, ko lahko vračamo več palic. Zato bom probleme v tem poglavju reševal samo z eksaktno metodo.

Ob sprotnem razrezu lahko pričakujemo, da bo posamezno naročilo sedaj ustrezno manjše, saj pripravljamo in izvajamo načrte razreza sproti. Zato lahko v večini praktičnih primerov pričakujemo, da bomo tudi z eksaktnim pristopom prišli do optimalne rešitve za posamezno obdobje v dani časovni omejitvi. V kolikor pa bi bila naročila še vedno prevelika za ta način, bi morali ustrezno prilagoditi metodo CUT ali C-CUT ter ju uporabiti za reševanje problema v posameznem obdobju. Približne meje ustreznosti eksaktne metode sem že ugotovil v poglavjih 6. 1. in 6. 2.

8.2 Rezultati – Model 1

Najprej predstavljam rešitev problema, ki sem ga že rešil s C-CUT-om. Rešitev je predstavljena na sliki 13. Zgornja meja je bila postavljena na 400. Čas reševanja problema, prikazanega na spodnji sliki (slika 20), je 0,16 sekunde.

Slika 20: Rešitev problema razreza z Modelom 1

PODATKI O NAROČILIH

zap. št. naročila	dolžina	št. kosov
1	144	2
2	194	3
3	249	18
4	157	15
5	129	12

PODATKI O PALICAH V ZALOGI

zap. št. palice	dolžina
1	2663
2	1862
3	1805
4	1963
5	2461
6	1963
7	2196
8	1532
9	1954
10	1158

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI PALICE

zap. št. palice	zap. št. naročila	št. kosov
1	4	12
1	5	2
2	4	3
3	5	10
5	1	2
5	2	3
7	3	5
8	3	4
9	3	6
10	3	3

nenarezani sta palici 4 in 6.

PODATKI O OSTANKIH PALIC

zap. št.	δ_j
palice	
1	521
2	1391
3	515
4	1963
5	1591
6	1963
7	951
8	536
9	460
10	411

Izguba pri vseh palicah (t_j) je enaka 0, saj so vse daljše od zgornje meje (400).

Kot je razvidno s slike 13, je končna izguba enaka kot pri C-CUT-u (in manjša kot pri uporabi samo CUT-a), rešitev pa najdem bistveno hitreje, kot če bi problem reševali samo z eksaktno metodo ter tudi nekoliko hitreje kot z dvofaznim C-CUT-om.

Problem je v bistveno večjem številu palic, ki so delno narezane. Pri tej metodi je bilo v načrtu uporabljenih kar 8 palic, ki so vse ostale delno nenarezane (in s tem ustrezno krajše kot prej), medtem ko je C-CUT uporabil le 5 palic, od tega se je le ena delno nenarezana vrnila v skladišče, vse preostale pa sem uporabil v celoti.

Razlika se še izraziteje pokaže, če zgornjo mejo postavim na 1000. V tem primeru namreč model najde optimalno rešitev v 0,71 sekunde; izguba materiala pri tej rešitvi je ravno tako 0. Vendar pa so dolžine preostalih palic (natančnega načrta razreza tokrat ne navajam) zaporedoma 1040, 1077, 1033, 1021, 1027, 1022, 1006, 1034, 1012 in 1029. Vse palice so torej sedaj narezane skoraj natančno do zgornje meje. To bo verjetno zelo otežilo razrez v naslednjem obdobju, saj bo možno te palice uporabiti le v celoti ali pa sploh ne.

Tako krajšanje palic pa povzroča vse naštetе probleme (predvsem večje stroške skladiščenja in večje težave ter izgube kasneje), povezane z večjim številom palic⁶⁰ in njihovo krajšo dolžino.

8.2.1 Rezultati model 1 – zaporedna časovna obdobja

Seveda pa se prednosti in slabosti manj strogih omejitev pri vračanju v skladišče povsem pokažejo šele pri eksperimentih, ki vključujejo celotno relevantno obdobje razreza, saj sem že poudaril, da lokalni optimumi sami po sebi ne vodijo nujno do dobre rešitve v celem obdobju.

⁶⁰ Kot sem pokazal tudi v enem od prejšnjih poglavij, ima število palic pomemben vpliv na možnost za pridobitev optimalne rešitve problema v dani časovni omejitvi.

Intuitivno lahko glede na zgoraj opisane probleme pričakujemo sicer dobre rešitve v prvih obdobjih, potem pa vedno večje težave ter večje izgube v naslednjih. Ravno tako lahko pričakujemo kopičenje večjega števila delno narezanih palic v skladišču.

Seveda pa moram omenjeno sklepanje empirično preveriti. Zato sem v tem poglavju predstavljeno metodo testiral ob naslednjih predpostavkah:

- metodo testiram za razrez v zaporednih obdobjih,
- v vsakem posameznem obdobju rešujem ustrezen model (torej bodisi model za premalo materiala s stroški neizpolnitve⁶¹, bodisi v tem poglavju predstavljen model),
- vsi ostanki, daljši od zgornje meje, se vrnejo v skladišče in so na razpolago za nadaljnja obdobja,
- v vsakem obdobju poleg obstoječe zaloge prispe v skladišče še nekaj novih dolžin palic,
- v vsakem obdobju prispe novo naročilo, ki vsebuje določeno število kosov določenih dolžin,
- končni rezultat je skupna izguba v celotnem relevantnem obdobju; zanima pa me tudi čas reševanja problema v posameznem obdobju in skupno število palic, ki so v skladišču na koncu in začetku posameznega obdobja. Čas reševanja je pomemben, ker mora metoda voditi do rešitve dovolj hitro, da bo primerna za uporabo v realnem času. Skupno število palic je pomembno, ker večje število krajših ostankov oteži iskanje optimalne ali vsaj dobre rešitve v določeni časovni omejitvi.

Vsebinsko zelo podobne pristope bo potrebno uporabiti tudi za testiranje morebitnih novih metod za to ali druge različice kontinuiranega razreza.

Za generiranje podatkov sem uporabil generator naključnih števil PGEN (Gradišar et al., 2002), ločeno za vsako obdobje. Generirani podatki za vsako obdobje so torej medsebojno neodvisni, problemi v različnih obdobjih pa so med sabo povezani preko prenosa preostalega materiala ali neizpolnjenih naročil v naslednje obdobje.

Za reševanje modela sem uporabili isti reševalec kot v prejšnjih problemih, torej MPL/CPLEX, ter isti računalnik (AMD, 1300 MHz). Celotna programska koda za eksperiment je napisana v programskem jeziku VBA, ki skrbi za pripravo podatkov za model za vsako od obdobj, izbiro pravilnega modela (preveč / premalo materiala) ter zabeleženje ključnih podatkov. Pripravljena procedura je tako primerna tudi za morebitno ponovitev eksperimenta ob drugačnih predpostavkah ali določenih spremembah v modelu.

⁶¹ Model je bil predstavljen v 7. poglavju.

Spremljam predvsem čas reševanja modela, izgubo po obdobjih in skupno izgubo ter dolžine preostalih palic, ki se vrnejo v skladišče in so vključene v problem naslednjega obdobja.

Dodatno me je zanimal tudi vpliv določitve zgornje meje na reševanje problema. Zato sem eksperiment ponovil z različnimi zgornjimi mejami. Očitno je najnižja smiselna določitev spodnje meje enaka najkrajši dolžini naročila, razen v kolikor v prihodnje pričakujemo tudi bistveno krajše dolžine naročil. Najvišja smiselna določitev pa je v približni dolžini najkrajše nove palice, saj bi lahko sicer kot neuporabne opredelili celo predhodno povsem neuporabljene palice. Zato sem pri testiranju spreminjal zgornjo mejo znotraj teh omejitev.

Vhodni podatki, na katerih sem testiral zgornji model, so naslednji:

- v vsakem od devetih obdobjih prispe novo naročilo. Novo naročilo je delovni nalog, ki je sestavljen iz določenega števila kosov štirih različnih dolžin,
- v vsakem obdobju so za razrez na razpolago štiri nove palice ter vse preostale palice iz preteklih razrezov, ki so bile daljše od zgornje meje,
- dolžina posamezne palice je med 3000 in 6000. Dolžina narezanega materiala je med 180 in 380, potrebujemo med 5 in 20 kosov posamezne dolžine,
- v vsakem od omenjenih obdobjih z modelom poiščem rešitev ob časovni omejitvi 60 sekund. V kolikor v 60 sekundah optimalna rešitev ni najdena, kot rešitev tega obdobja obdržim do tedaj pridobljeno rešitev,
- vsakega od opisanih eksperimentov ponovimo ob večanju zgornje meje v intervalih po 200 od vključno 400 do vključno 2000.

Poleg zgoraj navedenih podatkov sem seveda upošteval tudi vse zgoraj navedene splošne predpostavke. Podatki o dolžinah palic, njihovih zaporednih številkah in razporeditvi po obdobjih so razvidni iz tabele 11.

Tabela 11: Podatki o dolžinah palic

posamezno obdobje	j	d_j	posamezno obdobje	j	d_j	posamezno obdobje	j	d_j
1	1	5972	4	13	3668	7	25	3798
1	2	5822	4	14	3412	7	26	4238
1	3	5811	4	15	3233	7	27	4326
1	4	5625	4	16	5049	7	28	4160
2	5	5508	5	17	4563	8	29	3700
2	6	5483	5	18	4505	8	30	3598
2	7	5114	5	19	4473	8	31	3774
2	8	5049	5	20	3669	8	32	3858
3	9	4563	6	21	3668	9	33	4103
3	10	4505	6	22	3412	9	34	3516
3	11	4473	6	23	3233	9	35	5816
3	12	3669	6	24	3412	9	36	3798

Podatki o dolžinah in številu naročil, njihovih zaporednih številkah ter razporeditvi po obdobjih so razvidni iz tabele 12.

Tabela 12: Podatki o dolžinah in številu naročil

posamezno obdobje	i	b_i	s_i	posamezno obdobje	i	b_i	s_i	posamezno obdobje	i	b_i	s_i
1	1	18	240	4	13	7	256	7	25	18	240
1	2	14	238	4	14	5	254	7	26	14	238
1	3	39	221	4	15	20	246	7	27	39	221
1	4	11	203	4	16	25	245	7	28	11	203
2	5	8	202	5	17	4	243	8	29	8	202
2	6	8	370	5	18	20	234	8	30	8	370
2	7	4	357	5	19	11	203	8	31	4	357
2	8	5	356	5	20	1	189	8	32	5	356
3	9	26	336	6	21	4	309	9	33	26	336
3	10	4	309	6	22	8	290	9	34	4	309
3	11	8	290	6	23	3	261	9	35	8	290
3	12	3	261	6	24	20	234	9	36	3	261

Nastavitve generatorja naključnih števil so bile v obravnavanem primeru takšne, da je razpoložljivi material v vseh obdobjih in ob vseh nastavitvah zgornje meje zadoščal za izpolnitev zahtevanih naročil. Zato sem v vseh primerih uporabil model za reševanje za dovolj materiala.

Rezultati po posameznih obdobjih reševanja in ob različnih zgornjih mejah so prikazani v tabeli 13. Podatki o različnih zgornjih mejah so prikazani po vrsticah. Prva številka pomeni izgubo v posamezni fazi, druga število palic, ki ostanejo (delno) nenarezane v skladišču na koncu, tretja pa čas reševanja v sekundah. V času reševanja ni vključen čas za pripravo problema ter zagon modela, ki je v vseh primerih zanemarljiv (manj kot 2 sekundi). V zadnjem stolpcu so seštetimi podatki za celotno relevantno obdobje ob določeni zgornji meji.

Tabela 13: Izgube materiala, število palic in čas reševanja po posameznih obdobjih ob različnih nastavitvah zgornje meje

obd. UB	obd1	obd2	obd3	obd4	obd5	obd6	obd7	obd8	obd9	skupaj
400	0/4/0	0/7/0	0/11/0	0/15/0	0/19/0	0/23/0	0/27/25	0/30/0	0/34/0	0/34/26
600	0/4/0	0/7/0	0/11/0	36/15/60	0/18/0	0/22/0	0/26/0	8/30/60	0/33/0	44/33/121
800	0/4/0	0/7/0	0/11/0	27/15/60	0/18/0	0/22/0	0/26/0	0/30/9	1/33/60	28/33/129
1000	0/4/0	0/7/0	0/11/0	//	//	//	//	//	//	//
1200	0/4/0	0/7/0	0/11/0	44/15/60	0/18/0	0/22/0	0/26/0	0/30/0	2/34/60	46/34/120
1400	0/4/0	0/7/0	0/7/11	//	//	//	//	//	//	//
1600	0/4/0	0/6/0	0/10/0	1/14/60	0/17/0	0/21/0	12/25/60	0/25/0	8/29/60	21/29/180
1800	0/4/0	0/6/45	0/10/0	//	//	//	//	//	//	//
2000	0/4/0	3/6/60	7/9/60	//	//	//	//	//	//	//

Znak // pomeni, da izvedljiva (angl. feasible) rešitev za ta problem ne obstaja oziroma v časovni omejitvi ni bila najdena. Zato seveda problema ne morem reševati niti za naslednja obdobja, saj nimam vhodnih podatkov iz prejšnjega obdobja, ki so predpogoj za reševanje.

Čas reševanja 60 sekund pomeni, da je bila izvedljiva rešitev sicer najdena prej kot v 60 sekundah, vendar pa ta rešitev ni nujno optimalna. Krajši čas reševanja pomeni, da je bila pridobljena rešitev tudi zagotovo optimalna za tisto časovno obdobje. Dejstvo, da je bila večina rešitev pridobljenih v bistveno krajšem času kot 60 sekund, kaže na to, da je eksaktna metoda primerna za tako velikost problemov.

8.2.2 Analiza rezultatov

Kot je razvidno iz zgornje tabele, število palic v skladišču hitro narašča predvsem ob nizko postavljeni zgornji meji. Ob zgornji meji 400 sta tako od 36-ih palic, ki so prišle v skladišče tekom eksperimenta, do konca narezani le dve, 34 pa jih je preostalo v skladišču.

Po drugi strani višanje zgornje meje sicer nekoliko zmanjša to povečevanje števila palic (pri meji 1600 je tako na koncu 29 palic v skladišču), vendar hkrati povečuje izgube materiala pri razrezu. Predvsem pa se povečuje kompleksnost problema zaradi visoke zgornje meje in povečevanja števila palic. Za slednjo sem že dokazal, da pomembno vpliva na težavnost reševanja s celoštevilskim linearnim programiranjem. Posledično se povečuje čas, potreben za rešitev, tako da marsikdaj v naslednjih obdobjih sploh ne morem več pridobiti izvedljive rešitve znotraj časovne omejitve.

To je med drugim tudi posledica zgoraj omenjenega dejstva, da ta metoda večino palic delno nareže tako, da so le malo daljše od določene zgornje meje. Dostikrat tako velja:

$$\delta_j - UB < \min s_i$$

Takšne palice pa so nato v naslednjih obdobjih razmeroma neuporabne pri razrezu. Uporabne so samo, v kolikor jih uspemo narezati povsem do konca, ne moremo pa jih več vračati v skladišče.

Številni numerični eksperimenti in podrobna analiza predstavljene metode v okviru priprave doktorske disertacije (nekateri od rezultatov so bili predstavljeni tudi v tem poglavju) potrjujejo ugotovitve, do katerih bi lahko prišli tudi intuitivno. Tovrstna arbitrarna določitev zgornje meje, pri čemer vsi ostanki daljši od zgornje meje ne štejejo kot izguba, pričakovano vodi do požrešnega obnašanja algoritma za razrez, ki večino palic nareže do te zgornje meje (le zelo manjši del palic pa razreže do konca). To zanesljivo vodi do stalnega povečevanja števila palic v zalogi (in zmanjševanja njihove dolžine), kar vodi do nesorazmernega povečevanja tako težavnosti problema kot izgube pri optimalni ali hevristični rešitvi v prihodnje. Eksperiment je tako denimo pokazal tudi neustreznost pristopa, predstavljenega v (Alfieri et al., 2007); le-ta je namreč vsebinsko gledano zelo

podoben pristopu, predstavljenem v tem podpoglavju, vendar pa obravnava le razrez v enem obdobju.

Dodatno moram upoštevati dejstvo, da večje število palic, ki so uporabljene v načrtu razreza in nato vrnjene v skladišče, verjetno pomeni dodatne stroške za podjetje (stroške transporta, delo, administracija...). Te stroške sem do sedaj zanemarjal.

Čeprav so torej problemi v začetnih obdobjih lažje in hitreje rešljivi, je cena za to bistveno višja izguba ali celo nerešljivi primeri v prihodnjih obdobjih. S tem sem ostal daleč od zastavljenega cilja, ki je bil najti dobro rešitev kot vsoto vseh obravnavanih obdobj.

Problem delovanja modela lahko dodatno pojasnim z naslednjim primerom. Z vidika tega modela sta v posameznem obdobju enakovredni rešitvi, kjer:

- več palic narežemo do konca (izguba 0), vse preostale palice v celotni dolžini pa pustimo nedotaknjene za nadaljnje razreze,
- od vsake palice odrežemo le manjše število kosov naročil, preostanek palic pa uskladiščimo.

Očitno pa je, da prva rešitev vodi do bistveno manjših stroškov skladiščenja in predvsem od bistveno večje verjetnosti za manjše izgube pri prihodnjih razrezih. Vendar pa ji model ne bo dal prednosti pri izbiri niti, če je pri obeh izguba materiala enaka.

Zaradi vsega navedenega je torej omenjeni pristop neuporaben za minimizacijo stroškov pri kontinuiranem razrezu materiala. Ne le da ne privede do dobrih rešitev glede stroškov, ampak lahko vodi celo do umetnega povzročanja nerešljivosti problemov. Pristop namreč izrazito teži k iskanju lokalnih optimumov brez oziranja na približevanje globalnemu optimumu.

Omenjena ugotovitev je zanimiva tudi z vidika številnih metod, ki so bile pregledno predstavljene v tretjem poglavju. Vse zgoraj predstavljene metode namreč vodijo do dobrih/optimalnih rešitev v posameznem obdobju. Brez natančne analize pa lahko le ugibamo, ali ti lokalni optimumi tudi na srednji ali daljši rok pripomorejo k dobrim rešitvam oziroma izpolnjevanju poslovnih ciljev posameznega podjetja. Zato bi bilo večino teh metod potrebno testirati in ustrezno prilagoditi.

Osnovne značilnosti vseh štirih do sedaj predstavljenih pristopov k reševanju so na kratko povzete v tabeli 14.

Tabela 14: Osnovne prednosti in slabosti posamezne metode

Metoda	Prednosti	Slabosti
CUT	Hitro reševanje, primeren tudi za večje probleme	Rešitev ni (nujno) optimalna
C-CUT	Razmeroma hitro reševanje, primeren tudi za večje probleme	Rešitev ni (nujno) optimalna, je pa običajno boljša od CUT
Eksaktna	Vodi do optimalnih rešitev za manjše probleme	Ni primerna za večje probleme
Kontinuiran razrez – model 1	Srednje hitro reševanje, nižja izguba pri optimalni rešitvi zaradi blažjih omejitev	Vodi do velikega števila delno narezanih palic in posledično višjih stroškov skladiščenja in večjih težav pri kasnejših razrezih. V taki obliki ni primerna za kontinuiran razrez.

Glavni namen eksperimentov v tem podpoglavju je bil pokazati, da je za kontinuirano optimizacijo z neznanimi podatki o bodočem povpraševanju ter zalogah ključno, da obdržim čim nižjo težavnost tudi za naslednja obdobja. Samo v tem primeru je verjetno, da lahko dobro rešitev pričakujem čez celotno obdobje optimizacije.

Še vedno sicer iščem lokalne optimume, torej optimume za določeno časovno obdobje. Drugačen pristop zaradi nepoznavanja podatkov o prihodnjem povpraševanju/potrebah niti ni možen. Vendar pa moram upoštevati dejstvo, da so rešitev posameznega obdobja in vhodni podatki v naslednje tesno povezani. Zato mora biti model strukturiran tako, da optimalne rešitve problemov v posameznem obdobju ne bodo preveč oteževale problemov v naslednjih obdobjih.

Če hočem to doseči, moram seveda čim natančneje vedeti, kateri dejavniki vplivajo na težavnost problema, približno kakšne rešitve lahko pričakujem in kako lahko vplivam na težavnost problema tako, da bom prišel do dobrih rešitev. Z drugimi besedami: obravnavanje kontinuiranega razreza na tak način ni možno, v kolikor zelo dobro ne poznamo delovanja metod za trenutno optimizacijo pri obravnavani različici problema razreza. Vse predhodne analize v tej disertaciji pa omogočajo ravno zelo dobro poznavanje uporabljene metode.

Enega od možnih pristopov za doseganje tega cilja predstavljam v naslednjem podpoglavju. Poglavitno slabost modela namreč v pretežni meri odpravim z vključitvijo stroškov vračanja palice v skladišče.

8.3 Model ob stroških vračanja palic v skladišče

Vračanje delno uporabljenih palic v skladišče je nedvomno koristen pristop, saj lahko na ta način zmanjšamo izgubo materiala, delno narezane palice pa koristno uporabimo v naslednjih obdobjih.

Po drugi strani tak pristop povzroča dve glavni težavi:

- preveliko število vrnjenih palic, ki so delno narezane (in zato krajše), vodi do večjih stroškov skladiščenja in zelo verjetno tudi do večjih izgub pri kasnejših razrezih⁶²,
- vračanje palic v skladišče: vračanje palic v skladišče pomeni določene stroške za podjetje (transport, zlaganje, evidenca, administrativni postopki in podobno) – tudi te stroške je dostikrat smiselno upoštevati, v predhodnih poglavjih pa sem jih zanemarjal.

Želim torej predvsem zmanjšati število vrnjenih palic. Zato v tem podpoglavju model razširim z vključitvijo stroškov za vračanje palice. Sedaj torej ne morem več minimizirati samo izgube materiala (denimo centimetre odpadnega lesa), ampak je v ta namen potrebno stroškovno ovrednotiti tudi izgubo materiala pri razrezu. Ne zanima me več količina izgubljenega materiala, temveč stroški, ki so povezani s celotno pripravo in izvedbo razreza. Če razrez obravnavam kot enega od procesov v oskrbovalni verigi, kot je bilo prikazano v drugem poglavju, je tak pristop tudi bistveno bolj smiseln. Za konkurenčnost posameznega podjetja ali oskrbovalne verige so stroški v nekem obdobju boljši kazalec kot pa izguba materiala sama po sebi. Pri tem naj poudarim, da nam v drugem poglavju prikazani metodološki pristop omogoča, da s pomočjo simulacij ocenimo stroške, ki jih povzroča vračanje palic (ali pa denimo priprava palic na razrez).

Za zapis modela tako uvedem dva nova parametra, in sicer:

- c – stroški izgube na enoto materiala
- p_v – stroški vračanja palice v skladišče

Spremenljivka g_j ima nekoliko drugačen pomen kot prej in označuje palice, ki se vrnejo v skladišče. Tudi palica, ki je daljša od UB , se lahko zavrže in šteje kot izguba. Omejitev 4 to namreč omogoča. Do tega bi prišlo v primeru, da je $(t_j * c) < p_r$ ter $(t_j > UB)$, kar bi pomenilo, da so stroški vračanja za to palico večji od stroškov izgubljenega materiala.

Vrednost kriterijske funkcije, ki jo minimiziram, je torej sedaj enaka vsoti:

- stroškov, ki jih povzroča izguba materiala, kar je vsota produktov med stroški izgube na enoto (c) in izgube pri razrezu posamezne palice (t_j),
- stroškov vračanja palic v skladišče, ki je zmnožek med številom palic, ki jih vrnem, ter stroškom za vračanje ene palice.

Ponovno pa potrebujem tudi spremenljivko y_j , ki kaže, ali je bila palica y_j vključena v načrt razreza ali ne ($y_j=1$, v kolikor palica ni bila vključena v načrt razreza). Sedaj obravnavam torej 3 »vrste« palic:

⁶² Dejstvo, da krajše palice (oziroma manjše razmerje med povprečno dolžino palic in povprečno dolžino naročil) vodijo do večjih izgub pri razrezu, je tudi statistično dokazano (Gradišar et al., 1999).

- palice, ki niso bile vključene v razrez ($y_j = 1$) – te palice ne povzročajo nobenih stroškov,
- palice, ki so bile vključene v razrez, a jih vrnemo v skladišče ($g_j=1$) – preostanek teh palic torej ne šteje kot izguba, vendar pa vsaka taka palica povzroči strošek p_r ,
- palice, ki so bile vključene v razrez, vendar jih zavržemo – preostanek take palice šteje kot izguba, ki jo pomnožimo s stroškom izgube na enoto materiala (c). Stroškov z vračanjem takih palic seveda ni.

Spremenjen optimizacijski model lahko sedaj zapišem kot:

$$(1A) \quad \min \sum_{j=1}^m (t_j * c + g_j * p_r) \quad (\text{minimiziraj vsoto stroškov izgube in stroškov vračanja})$$

s pogojem, da

(2, 3) enako kot na str. 87

(4A) $UB - \delta_j + UB(g_j - 1) \leq 0 \quad \forall j$ (g_j kaže, ali je preostanek palice d_j daljši od UB in je lahko enak 1 samo, če je preostanek daljši od te meje)

(5,6) enako kot na str. 87

8.3.1 Rezultati modela s stroški vračanja

Ustreznost predstavljenega modela moram preveriti z reševanjem večjega števila primerov. Z novim modelom sem najprej rešil predhodno navedeni problem (rešen v prejšnjem podpoglavju ter tudi s C-CUT-om), pri čemer je bila zgornja meja ponovno nastavljena na 400. Strošek za izgubljen enoto materiala sem nastavili na 2, strošek za vračanje palice v skladišče pa na 10. Podatki o palicah in naročilih so razvidni s slike 13. Oceno stroškov vračanja sem v tem primeru sicer predpostavil, vendar simulacije poslovnih procesov nudijo tudi možnost za boljšo oceno stroškov izvajanja posameznega podprocesa (glej denimo Trkman et al., 2007 ali Trkman, McCormack, 2008).

Spodnja rešitev (prikazana na sliki 21) je bila najdena v 3 minutah in ni nujno optimalna. Zagotovo optimalne rešitve namreč program MPL/CPLEX v časovni omejitvi desetih minut ni našel. To je predvsem posledica dejstva, da je model za reševanje sedaj bolj kompleksen. Model v predhodnem podpoglavju je predvsem ob nizko določeni zgornji meji namreč zelo lahek za rešitev, vendar je vodil do zgoraj analiziranih problemov.

Kot je razvidno s spodnje slike, se torej izguba sicer poveča iz 0 na 2 centimetra, kar predstavlja povečanje od 0 na 0,022% celotnega naročila. Vendar se po drugi strani (v primerjavi s sliko 13) število palic, uporabljenih v načrtu razreza, zmanjša z osem na pet. Poleg tega se povprečna dolžina preostalih palic v skladišču poveča iz 1013 na 1717 (kar

za 60%). Začetna povprečna dolžina palic (pred razrezom) je bila sicer 1955, tako da je v drugem modelu prišlo le do neznatnega zmanjšanja povprečne dolžine.

Slika 21: Rešitev problema ob upoštevanju stroškov vračanja

REZULTATI, RAZPOREJENI PO ZAPOREDNI ŠTEVILKI PALICE

zap. št. palice	zap. št. naročila	št. kosov
1	1	1
1	2	3
1	3	1
1	4	5
1	5	7
3	1	1
3	3	5
3	4	1
3	5	2
5	3	3
5	4	2
7	3	6
7	4	2
7	5	3
8	3	3
8	4	5

nenarezane so palice 2, 4, 6, 9 in 10.

PODATKI O OSTANKIH PALIC

zap. št. palice	δ_j	
1	0	(izguba 0)
2	1862	(nenarezana)
3	1	(izguba 1)
4	1963	(nenarezana)
5	1400	(vrnjena v skladišče – stroški 10)
6	1963	(nenarezana)
7	1	(izguba 1)
8	0	(izguba 0)
9	1954	(nenarezana)
10	1158	(nenarezana)

izguba = 2

št. vrnjenih palic = 1

skupni stroški = $2 * 2 + 1 * 10 = 24$

Vidimo lahko, da sem z uvedbo teh stroškov dosegel oba želena cilja: po eni strani se sedaj v načrtu razreza uporabi manj palic, kar pomeni tudi manj transportnih stroškov. Kot posledica večjega števila do konca narezanih in manjšega števila delno narezanih palic, pa

se pomembno poveča tudi povprečna dolžina preostalih palic v skladišču, kar nudi boljše možnosti za nadaljnje razreze. Vračanje ene ali nekaj palic v skladišče pa še vedno omogoča pridobitev načrta razreza z nizko izgubo materiala.

Z vsem tem sem prišel do zelenega cilja: težavnost problema se iz obdobja v obdobje predvidoma ne bo več povečevala. Zato lahko pričakujem dobre rezultate čez celotno relevantno obdobje.

Vpliv povečanja števila do konca narezanih palic in povprečne dolžine preostalih pa se še izraziteje pokaže pri ponovitvi drugega eksperimenta. Vse nastavitve tega eksperimenta so povsem enake, kot so bile opisane v prejšnjem podpoglavju. Ravno tako sem uporabil iste podatke o dolžini palic in dolžini ter številu naročil. Edina razlika je torej, da uporabljam drug model za reševanje.

Vsi rezultati so prikazani v tabeli 15, kjer je pomen številčk podoben kot v tabeli 13. Edina razlika je, da druga številka sedaj pomeni skupne stroške izgube materiala in vračanja palic v skladišče. Ta razlika je seveda posledica spremenjene kriterijske funkcije. Prva številka pa spet predstavlja, tako kot v zgornji tabeli, skupno izgubo materiala v posameznem obdobju.

Tabela 15: Izgube materiala, skupni stroški, število palic in čas reševanja po posameznih obdobjih ob različnih nastavitvah zgornje meje

obdobje UB	obd1	obd2	obd3	obd4	obd5	obd6	obd7	obd8	obd9	skupaj
400	0/20/4/3	1/22/5/0	0/20/8/2	7/54/10/60	0/20/12/1	0/20/14/12	5/50/16/60	0/0/16/1	4/28/18/60	17/234/18/200
600	0/20/4/2	1/22/5/0	0/20/8/5	0/40/10/60	0/20/12/2	6/32/14/60	6/120/16/60	9/18/19/35	12/164/20/60	34/456/20/285
800	0/20/4/0	0/20/5/0	0/20/8/16	13/66/10/60	0/0/12/0	0/10/14/2	13/226/16/60	0/20/16/5	2/24/18/60	28/406/18/203
1000	0/20/4/0	0/20/5/0	0/20/8/8	4/88/10/60	0/0/13/1	0/20/14/10	7/74/16/60	0/20/17/25	9/38/19/60	20/300/19/224
1200	0/20/4/0	0/20/5/0	0/20/8/8	7/134/10/60	0/20/13/4	8/16/15/6	4/88/16/60	0/20/15/15	8/76/18/60	27/414/18/214
1400	0/20/4/0	0/20/5/0	3/26/8/20	6/92/9/60	0/20/12/5	0/20/14/15	0/20/16/60	0/20/16/38	8/56/18/60	17/294/18/258
1600	0/20/4/0	0/20/5/0	3/26/8/60	0/40/9/60	0/20/11/1	0/20/13/7	0/20/15/60	0/20/14/7	0/20/16/60	3/240/16/255
1800	0/20/4/0	0/20/5/0	3/26/8/60	0/80/10/60	0/20/13/2	0/20/15/11	3/66/17/60	0/20/16/28	0/40/18/60	6/312/18/282
2000	0/20/4/0	0/20/5/0	0/20/8/6	0/60/9/60	0/0/11/0	0/20/13/5	1/22/15/60	0/20/15/8	0/120/17/60	1/302/17/198

Prva ugotovitev je, da so sedaj vsi primeri v vseh obdobjih reševanja rešljivi že ob časovni omejitvi ene minute. Čas reševanja se je pri rešljivih problemih sicer nekoliko povečal v primerjavi s tabelo 13, kjer pa do rešitve za kasnejša obdobja ob višji nastavitvi zgornje meje sploh nisem prišel.

Zanimiva ugotovitev je, da ni moč opaziti vpliva spreminjanja zgornje meje, ne na skupne čase reševanja, ne na končne skupne stroške razreza v celotnem relevantnem obdobju. Čas reševanja se tako giba od 198 do 285 sekund, najmanjši pa je prav pri najmanjši in največji nastavljeni zgornji meji. Skupna izguba materiala je med 0,001% in 0,031% skupnega povpraševanja v celotnem relevantnem obdobju.

Skupni stroški razreza so od 234 do 456 denarnih enot, ravno tako pa ni opazno, da bi se večali ob povečevanju zgornje meje. Tudi statistična analiza pridobljenih rezultatov ni

pokazala statistično značilne povezave med zgornjo mejo ter skupnimi stroški razreza, saj je Pearsonov korelacijski koeficient med tema spremenljivkama enak 0,29 s stopnjo značilnosti 0,45. Tako ne morem trditi, da ima spreminjanje *UB* na intervalu od 400 do 2000 učinek na skupne stroške razreza v celotnem relevantnem obdobju.

Ravno tako ni opaziti bistvenega povečevanja izgube v kasnejših obdobjih razreza, kar verjetno kaže na to, da je tako zastavljen model ustrezen, saj ne privede do dobrih rešitev v posameznem obdobju na račun preostalih obdobj.

Predstavljeni model torej ne glede na spreminjanje nastavitve zgornje meje in povečevanje števila zaporednih obdobj vodi do približno enakih rešitev ob približno enaki časovni kompleksnosti. Slednje je pomembna prednost pred v prejšnjem podpoglavju predstavljenim modelom brez stroškov vračanja palic. Večanje zgornje meje ima verjetno predvsem dva učinka na optimalno rešitev:

- negativen (za dano obdobje): problemi v danem trenutku so nekoliko težji, saj mora biti posamezni ostanek palice sedaj ustrezno daljši, da ga lahko vrnemo v skladišče in ne šteje kot izguba,
- pozitiven (za naslednja obdobja): dolgoročno gledano so sedaj vrnjene palice v skladišče daljše, tako da je lažje reševati probleme v prihodnjih obdobjih.

Iz predstavljenih rezultatov je očitno, da se ob obravnavanih lastnostih problema in opisanih nastavitvah, omenjena učinka medsebojno približno izničita in večanje zgornje meje nima posebno opaznega vpliva na skupne stroške oziroma, da nastavitev zgornje meje kjerkoli od 400 do 2000 vodi do dobrih rezultatov.

Vsebinsko gledano so taki rezultati razveseljivi, saj so indikator ustrezne zasnove modela. Zgornja meja je namreč v podjetju običajno arbitrarno določena, da se na ta način preprečuje zbiranje prevelikega števila kratkih palic. Eksperiment potrjuje, da vse nastavitve od 400 do 2000 (kar je približno 50% povprečne dolžine nove palice) vodijo do ustreznih rezultatov.

Zgornja meja pa mora tudi ob vključitvi stroškov vračanja ostati v modelu in mora biti določena vsaj v višini minimalne dolžine naročila. V nasprotnem primeru bi lahko prišlo do situacije, ko bi v skladišče vračali palice, ki bi bile krajše od najkrajše dolžin naročil in tako povsem neuporabne za prihodnje razreze ter bi ostale v skladišču za vselej (oziroma do prvega problema razreza s premalo materiala, kjer zgornja meja ni določena in vsi preostanki štejejo kot izguba).

Pri stroških po obdobjih nisem upošteval časovne vrednosti denarja, saj predvidevam, da gre praviloma za zelo kratka posamezna časovna obdobja (denimo dnevni razrez) in celotno relevantno obdobje predstavlja samo nekaj dni ali kvečjemu tednov. Kot ilustracijo pa v tabeli 16 navajam tudi rezultate, diskontirane na prvo časovno obdobje. V tabeli

prikazujem samo diskontirane stroške, saj so časi reševanja in število palic enaki kot v tabeli 15.

Uporabil sem naslednje predpostavke:

- posamezno obdobje predstavlja en mesec. Gre torej za 9-mesečno optimizacijo,
- letna diskontna stopnja znaša 10%, kar predstavlja ob konformnem preračunu 0,797% mesečno diskontno stopnjo.

Tabela 16: Stroški razreza, diskontirani na 1. obdobje (mesec)

obd. UB	obd1	obd2	obd3	obd4	obd5	obd6	obd7	obd8	obd9	skupaj
400	20	22	22	53	19	19	48	0	26	227
600	20	22	22	39	19	31	114	17	154	436
800	20	20	20	64	0	10	215	19	23	391
1000	20	20	20	86	0	19	71	19	36	290
1200	20	20	20	131	19	15	84	19	71	399
1400	20	20	26	90	19	19	19	19	53	284
1600	20	20	26	39	19	19	19	19	19	200
1800	20	20	26	78	19	19	63	19	38	302
2000	20	20	20	59	0	19	21	19	113	290

Očitno je, da so tudi ob upoštevanju razmeroma dolgega obdobja en mesec rezultati še vedno zelo podobni. V praksi se običajno srečujemo s krajšo dolžino posameznega obdobja, zato v večini praktičnih primerov upoštevanje časovne vrednosti denarja najverjetneje ne bi vplivalo na uporabnost pridobljenih rezultatov.

Dokazal sem torej, da sem z omenjeno razširitvijo modela dosegel, da lokalni optimumi v enem obdobju sedaj vodijo tudi do dobrih rešitev za celotno relevantno obdobje. Predlagani pristop lahko torej pomaga podjetju k sprejemanju boljših odločitev, ki bodo vodile do nižjih stroškov in večjih dobičkov.

Čeprav prikazana metoda ni bila razvita za konkreten problem določene organizacije, pa je možna njena enostavna integracija v sistem za podporo odločanja – praktičen primer tega je prikazan v (Čižman, Černetič, 2004).

Poleg tega procedura za eksperiment, napisana v VBA, uporablja Microsoft Excel kot uporabniški vmesnik. To orodje je praviloma dobro poznano večini odločevalcev na različnih nivojih odločanja. Tovrstna uporaba Microsoft Excela kot uporabniškega vmesnika pri uporabi metod za podporo odločanja je dejansko čedalje pogostejša v poslovni praksi (Shim et al., 2002; Buehlmann et al., 2000). Preprosta uporaba in relativno kratki časi reševanja pa omogočajo tudi kaj-če (angl. what-if) analize. Tako lahko iščemo odgovore na vprašanja, kako bi na celotne stroške procesa vplivala sprememba v povprečni ravni zalog, obsegu povpraševanja, pogostosti razreza, razmerju med stroški materiala in stroški vračanja. Možna je tudi nadgradnja omenjene procedure za povečanje prijaznosti

uporabe za odločevalca pri razrezu ali njena vključitev v integrirani informacijski sistem organizacije; načeloma pa celo v medorganizacijski informacijski sistem.

Kaj-če analiza torej odgovarja tudi na vprašanja, kaj bi se zgodilo z optimalno rešitvijo ob različnih predpostavkah. Taka vprašanja pogosteje postavljajo managerji, ki sprejmejo končno odločitev, kot pa strokovnjaki za operacijske raziskave (Hillier, Lieberman, 2004). Zato je prijaznost uporabniškega vmesnika in hitrost reševanja še toliko pomembnejša.

Vse naštetu omogoča tudi analizo potencialnih sprememb v oskrbovalni verigi (prehod na drugačen način naročanja, pogostejša naročila, povečevanje fleksibilnosti posameznega člana in celotne verige); nekatere od tovrstnih sprememb sem predstavil v drugem poglavju (teoretično in na študiji primera).

8.3.2 Nadaljnje razširitve modela

Transformacija kriterijske funkcije v predstavljenem modelu omogoča, da po potrebi brez težav dodajamo tudi nove predpostavke in stroške. Večino preostalih elementov pri optimizaciji razreza je namreč možno razmeroma enostavno prevesti na stroškovno osnovo.

Tako lahko denimo domnevam, da tudi prenos palice iz skladišča na mesto razreza povzroča določene stroške. Le-ti so lahko enaki ali pa tudi različni od stroškov vračanja, odvisno od dejanske organizacije razreza v podjetju.

Če sedaj označim stroške prvotnega prenosa s p_t , potem lahko zgornjo kriterijsko funkcijo preprosto zapišem kot:

$$\min \sum_{j=1}^m (t_j^* c + g_j^* p_r) + (m - \sum_{j=1}^m y_j) * p_t$$

Pri tem v izrazu $m - \sum_{j=1}^m y_j$ izračunamo število palic, ki jih moramo pripraviti za razrez, kot razliko med številom vseh palic in številom palic, ki niso bile uporabljene v načrtu razreza (in imajo $y_j=1$). Vse omejitve so enake kot zgoraj, saj razširitev modela v tem primeru spremeni samo zapis kriterijske funkcije.

Zgoraj predlagana dopolnitev pomeni, da bo model ceteris paribus iskal rešitve, ki bodo zahtevale prenos manjšega števila palic iz skladišča do rezalnega stroja. Podobno lahko v model dodamo druge stroške, omejitve ali cilje, ki jih dodamo k kriterijski funkciji ali pa med omejitve modela. Cilj takega razširjanja je kar najbolje opisati probleme, s katerimi se sooča podjetje pri doseganju svojih poslovnih ciljev skozi daljše časovno obdobje.

V poglavju sem torej predlagal nov pristop k reševanju problema razreza, kjer iščem dobro rešitev v zaporednih časovnih obdobjih. Predlagal sem posebno razširitev obstoječe

klasifikacije za ustrezno organizacijo novega področja pri raziskovanju enodimenzionalnega razreza – torej optimizacije problemov skozi daljše časovno obdobje kot nadgradnja pristopov za trenutno optimizacijo.

Predstavil sem dva modela, ki sem jih tudi eksperimentalno testiral. Rezultati so razkrili številne pomanjkljivosti prvega modela, predvsem povečanje izgube materiala in celo nerešljive probleme v kasnejših časovnih obdobjih zaradi večanja težavnosti problema. Te pomanjkljivosti sem odpravil z uvedbo stroškov za vračanje palic v skladišče. Drugi model je sedaj rešljiv v vseh časovnih obdobjih in vodi do nizkih skupnih stroškov za celotno relevantno obdobje, kar kaže na pomen ohranjanja nespremenjene težavnosti problema čez celotno obdobje in preprečevanje požrešnega obnašanja algoritma v prvih fazah.

Rezultati drugega modela so bistveno boljši od prvega. Tudi absolutna višina stroškov je razmeroma nizka. Za dokončno potrditev ustreznosti pristopa bi bila potrebna primerjava rezultatov z drugimi objavljenimi metodami⁶³. Ker pa je to prvi objavljeni pristop za tako opredeljen tip problema, taka primerjava trenutno žal ni možna. Upam in pričakujem, da se bodo v prihodnje razvile nove metode za ta ali podobne tipe problemov, kar bo omogočilo neposredne primerjave.

Pričakujemo lahko, da se bo pomen kontinuiranega razreza v prihodnje še povečeval, predvsem zaradi dejavnikov, povezanih z večanjem pomena managementa oskrbovalne verige, ki sem jih nakazal v drugem poglavju. Marsikatero od obstoječih metod za trenutno optimizacijo bo potrebno testirati in tudi prilagoditi, da se bo izboljšala njena ustreznost za uporabo pri optimizaciji kontinuiranega razreza.

9 Sklep

9.1 Ključna spoznanja

V doktorski disertaciji sem po eni strani problem razreza obravnaval v širšem kontekstu od preostalih raziskovalcev v svetu, po drugi strani pa sem tudi za predhodno opisane probleme z novimi pristopi uspel najti boljše rešitve. Prispevek pričujoče disertacije je torej tako pri izboljševanju pristopov za reševanje za predhodno opazovane, opisane in analizirane probleme kot tudi pri nakazovanju novih smeri razvoja ter odkrivanju pionirskih metod na teh področjih (denimo utemeljitev kontinuiranega razreza, uporaba metod strojnega učenja za izbor metode, obravnavanje razreza kot procesa v oskrbovalni verigi).

Spoznanja disertacije so koristne za vsa podjetja, ki se srečujejo z razrezom ali s problemi s podobno logično strukturo. Še posebej so ključna za tista podjetja, kjer je razrez eden

⁶³ Tako kot sem za C-CUT pokazal, da je boljši od predhodnih metod za splošni problem enodimenzionalnega razreza.

pomembnejših procesov, celotni stroški razreza pa predstavljajo relativno visok delež skupnih stroškov.

Ključne prispevke disertacije lahko razvrstimo na naslednja področja:

- nove metode za razrez, ki zmanjšujejo izgube,
- razvoj metod, ki omogočajo upoštevanje dodatnih kriterijev,
- pristop za izbor ustrezne metode,
- obravnavanje razreza kot kontinuiranega procesa.

9.1.1 Metode za razrez

V preteklih letih je bilo razvitih že veliko metod za pripravo načrta razreza, nekaj od njih tudi posebej prilagojenih za splošni problem enodimenzionalnega razreza, kjer je omogočeno vračanje ene delno uporabljene palice v skladišče. Metode so praviloma vodile do majhnih izgub tudi za najzahtevnejše probleme razreza. V tej disertaciji sem najprej predstavil pristop pridobivanja optimalnih rešitev, za katerega sem dokazal, da za manjše probleme vodi do boljših rezultatov (v obravnavanem primeru je to manjša izguba materiala) od obstoječih metod.

Ker zaradi NP težkosti problema razreza predstavljena metoda ni primerna za večje probleme, sem najprej temeljito analiziral meje problema in dejavnike, ki najbolj vplivajo na njeno ustreznost za posamezne probleme. Ta spoznanja sem, skupaj s predhodnimi analizami obstoječih heurističnih metod, uporabil za razvoj nove metode C-CUT, ki združuje večino prednosti eksaktnega in heurističnega pristopa ob hkratni minimizaciji njunih pomanjkljivosti. Z rešitvijo velikega števila predhodno objavljenih problemov sem dokazal, da je metoda dejansko bistveno boljša od predhodno razvitih metod za enake probleme (CUT, COLA).

Razvite metode torej vodijo do manjših izgub materiala in posledično do zmanjševanja stroškov, povezanih z razrezom materiala.

9.1.2 Metode z dodatnimi kriteriji

Čeprav večina razvitih metod kot ključni ali celo edini kriterij poudarja minimizacijo izgube materiala, se v zadnjih letih povečuje pomen metod, ki razrez obravnavajo širše in v optimizacijske modele vključujejo tudi druge relevantne kriterije. Tako zastavljene metode lahko namreč vodijo do načrtov razreza, ki bolje izpolnjujejo cilje podjetja.

Obstoječe metode za splošni problem enodimenzionalnega razreza do sedaj dodatnih kriterijev praviloma niso vključevale, razen zahteve, da ni prevelike proizvodnje (Gradišar et al., 1999a). Dodatni kriteriji so lahko ključni predvsem v primerih, ko zaradi različnih razlogov razpoložljiv material ne zadošča za izpolnitev celotnih potreb. V sedmem poglavju sem predstavil pristop, ki omogoča vključevanje poljubnih dodatnih kriterijev pri

sprejemanju odločitev, katera naročila bodo ostala nenarezana. Ustrezno prilagojeni pristop v določenih primerih omogoča celo možnost boljšega izpolnjevanja pomembnejših naročil brez povečanja izgube materiala. Tako podjetje ali oskrbovalna veriga lahko bolje sledi ostalim ciljem brez povečevanja stroškov materiala.

V osmem poglavju pa sem v okviru obravnavanja kontinuiranega razreza omogočil tudi vključitev dodatnih dejavnikov v stroškovno ovrednoteno kriterijsko funkcijo (denimo stroške priprave razreza ali stroške prenosa palic iz skladišča in njihove vrnitve v skladišče).

Vpliv vključevanja dodatnih dejavnikov na pripravo načrta razreza sem prikazal in preučil tudi na nizu računskih primerov, ki so ilustrirali uporabo razvitih pristopov.

9.1.3 Izbor metode

V preteklih letih je bilo razvito veliko število metod, ki so primerne za različne probleme v teoriji in praksi. Velik del razvitih metod in različne pristope k reševanju sem povzel v tretjem poglavju. Predvsem metode, ki temeljijo na hevrističnem pristopu, so običajno zelo specifične in primerne samo za posamezno različico problema. Kljub temu smo pogosto soočeni s situacijo, ko za isti problem obstajajo različni pristopi za reševanje. Odločevalec v podjetju mora izbrati ustrezno metodo za posamezen problem, saj zaradi časovnih ali stroškovnih omejitev običajno ni možno ali smiselno uporabiti vseh obstoječih metod.

Kljub hitremu razvoju metod za podporo pri odločanju na podlagi strojnega učenja, tak pristop za izbor metode za reševanje problema razreza še ni bil uporabljen. Zato sem v poglavju 6. 2. prikazal pristop na podlagi odločitvenih dreves. Na predhodno objavljenih problemih sem dokazal, da reševanje posameznega problema z metodo, izbrano s pomočjo odločitvenih dreves, vodi do nižje izgube materiala kot reševanje samo z eno od obeh metod.

9.1.4 Razrez kot kontinuiran poslovni proces

Eden od pomembnejših prispevkov te disertacije je obravnava razreza kot kontinuiranega procesa v podjetju ali oskrbovalni verigi. V drugem poglavju sem prikazal pomen managementa poslovnega procesa preskrbe z materialom in znotraj tega tudi vlogo postopka razreza. Pokazal sem, kako lahko ustrezna integracija procesov prinese prednosti tako za dobavitelja materiala kot za podjetje, ki nabavlja in reže material (oziroma za podjetje, ki kupuje narezan material) ter vključenost procesa razreza v povezane (mejne) procese v podjetju

Praktično vse obstoječe metode zasledujejo le cilje pri optimizaciji razreza v enem trenutku. V osmem poglavju sem ugotovil potencialne pomembne pomanjkljivosti takega pristopa, ki ni nujno primeren za uporabo pri kontinuiranem razrezu. Optimizacija v danem trenutku namreč lahko vodi do dobrih trenutnih rezultatov (lokalnega optimuma) na račun

bistveno slabših rezultatov v celotnem obdobju (globalnega optimuma). V disertaciji sem zato predstavil nov pristop za testiranje takih metod, ki vodi do ustreznih rezultatov ter nizke izgube materiala in celotnih stroškov razreza čez celotno obdobje. Pristop omogoča relativno enostavno vključevanje novih relevantnih kriterijev.

Ob utemeljitvi večanja pomena kontinuiranega razreza sem ugotovil pomembno pomanjkljivost obstoječe klasifikacije problemov razreza in pakiranja ter predlagal njeno razširitev z novim razredom, ki omogoča ustrezno klasifikacijo metod glede na obdobje in vrsto problemov, ki jih zajemajo v optimizacijskem modelu.

9.2 Nadaljnje delo

Ocenjujem, da nadaljnje izboljševanje metode za trenutni splošni problem enodimenzionalnega razreza ni več smiselno, saj v okviru te disertacije razvita metoda C-CUT že vodi do zelo majhnih, praktično zanemarljivih izgub materiala. Predstavljeni rezultati namreč kažejo, da izguba nikjer ne presega 0,1% celotnega uporabljenega materiala. Kot omenjeno že v omejitvah disertacije, je sicer možno, da ob drugačnih vhodnih kriterijih ali ciljnih tudi ta metoda odpove. V takih primerih je smiselno razvijati izboljšane metode za specifične probleme.

Vendarle pa več izzivov za nadaljnjo raziskovalno delo na tem področju vidim predvsem v širšem obravnavanju tematike ter povezovanju problema razreza s skladiščenjem, naročanjem ter pripravo proizvodnje. Gre torej za širše obravnavanje razreza v kontekstu upravljanja z oskrbovalno verigo – možne pristope in nadaljnje smeri raziskovanja sem nakazal v drugem poglavju.

Nadaljnja raziskovalna vprašanja na področjih, ki so povezana s temo disertacije, so tako predvsem:

- vključevanje stroškov skladiščenja: kot pokazano, v disertaciji razviti modeli poleg stroškov materiala (izgube pri razrezu) omogočajo tudi vključitev drugih stroškov, denimo stroškov priprave razreza ali vračanja palic v skladišče. Dodatna smer raziskovanja bi bila tudi eksplicitno vključevanje stroškov skladiščenja oziroma iskanje tiste ravni zalog, ki bi verjetno vodila do najmanjših skupnih stroškov ob upoštevanju stroškov naročanja, zalog, razreza ter stroškov morebitne neizpolnitve posameznih naročil (glej denimo Erjavec et al., 2008a za primer takega pristopa). Razviti simulacijski model bi ob danih predpostavkah omogočal podjetju oceniti ustrezno velikost skladišča ter obsega in raznovrstnosti materiala na zalogi,
- v povezavi s stroški skladiščenja je pomembno vprašanje tudi optimalna količina naročila. Medtem ko se je z optimalno količino in pogostnostjo naročila ukvarjalo že veliko raziskovalcev, pa ta problem v povezavi s problemom razreza še ni bil preučevan. Pomembno vprašanje je, kako velikost posameznega naročila vpliva na možnost pridobivanja dobrih rešitev pri razrezu in kakšna je

optimalna nabavna politika podjetja za doseganje svojih ciljev ali ciljev celotne oskrbovalne verige,

- v disertaciji sem delno že prikazal potencialno vlogo množičnega prilagajanja pri optimizaciji razreza. Nadaljnje raziskovalno vprašanje je predvsem, kakšna mera množičnega prilagajanja pri razrezu je sploh smiselna, glede na to da razrez nastopa bolj na začetku oskrbovalne verige – običajno je namreč smiselno prilagajanje izvesti čim bolj proti koncu verige. Potrebna je natančna študija dodatnih stroškov ter potencialnih dodatnih koristi zaradi take uporabe množičnega prilagajanja na praktičnem primeru.

Opreديلil sem torej nekatera nadaljnja raziskovalna vprašanja. Nadaljnji razvoj teorije in prakse na tem področju pa bo pokazal, katere smeri razvoja so najbolj aktualne in vredne povečanega raziskovalnega napora.

10 Literatura

1. Abdel-Malek Layek, Das Sanchoy, Wolf C: Design and implementation of flexible manufacturing solutions in agile enterprises. *International Journal of Agile Management Systems* 2 (3), 2000, str. 187–195.
2. Ahlstrom Par, Westbrook Roy: Implications of mass customization for operations management. *International Journal of Operations and Production Management*, 19 (3/4), 1999, str. 262-275.
3. Aktin Tülin, Özdemir Rifat: An integrated approach to the one-dimensional cutting stock problem in coronary stent manufacturing, *European Journal of Operational Research* (2008). V tisku.
4. Alfieri Arianna, van de Velde Steef, Woeginger Gerhard: Roll cutting in the curtain industry, or: A well-solvable allocation problem. *European Journal of Operational Research*, 183 (3), 2007, str. 1397-1404.
5. Alves Claudio, Carvalho Valerio: A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Computers & Operations Research*, 35 (4), 2008, str. 1315-1328.
6. Androulakis Ioannis: Selecting maximally informative genes. *Computers & Chemical Engineering*, 29 (3), 2005, str. 535-546.
7. Antonio Julien, Chauvet Fabrice, Chu Chengbin, Proth Jean-Marie: The cutting stock problem with mixed objectives: Two heuristics based on dynamic programming. *European Journal of Operational Research*, 114 (2), 1999, str. 395-402.
8. Arbib Claudio, Marinelli Fabrizio, Rossi Fabrizio, Di Iorio Francesco: Cutting and reuse: the application from automobile component manufacturing. *Operations Research*, 50 (6), 2002, str. 923-934.

9. Arbib Claudio, Marinelli Fabrizio: 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), 2005, str. 617–630.
10. Armbuster Michael: A solution procedure for a pattern sequencing problem as part of one-dimensional cutting stock problem in the steel industry. *European Journal of Operational Research*, 141 (2), 2002, str. 328-340.
11. Arsham Hossein, Damij Talib, Grad Janez: An algorithm for simplex tableau reduction: the push-to-pull solution strategy. *Applied mathematics and computation*, 137 (2-3), 2003, str. 525-547.
12. Auer Peter, Holte Robert, Maass Wolfgang: Theory and application of agnostic pac-learning with small decision trees. *Proceedings of Twelfth International Conference on Machine Learning*, San Francisco: Morgan Kaufmann, 1995, str. 21-29.
13. Belov Gleb, Scheithauer Guntram: A cutting plane algorithm for one-dimensional cutting stock problem with multiple stock lengths. *European Journal of Operational Research*, 141 (2), 2002, str. 274-294.
14. Belov Gleb, Scheithauer Guntram: 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), 2006, str. 85-106.
15. Bergey PK, Ragsdale Cliff: Evolver 4.0.:The genetic algorithm super solver. *OR/MS Today* 6 (4), 1999, str. 44-47.
16. Bertrand Will, Fransoo Jan: Operations Management Research Methodologies using quantitative modelling. *International Journal of Operations & Production Management*. 22 (2), 2002, str. 241-254.
17. Bhadury Joy, Chandrasekaran Ramaswamy: Stock cutting to minimize cutting length. *European Journal of Operational Research*, 88 (1), 1996, str. 69-87
18. Biggs Norman: *Discrete Mathematics*. Oxford: Oxford University Press, 2002, 440 str.
19. Bingul Z, Oysu C: Comparison of stochastic and approximation algorithms for one-dimensional cutting problems. *Advances in intelligent computing; Lecture Notes in Computer Science*, 3644, 2005, str. 976-985.
20. Bosilj-Vuksić Vesna, Indihar Štemberger Mojca, Jaklič Jurij, Kovačič Andrej: Assessment of E-Business Transformation Using Simulation Modeling. *Simulation*, 78 (12), 2002, str. 731-744.
21. Boulianne Emilio: The impact of procurement card usage on cost reduction, management control, and the managerial audit function. *Managerial Auditing Journal*, 20(6), 2005, str. 592-604.
22. Breiman Leo, Friedman Jerome, Olshen Richard, Stone Charles: *Classification and Regression Trees*. Wadsworth: Belmont, 1984.

23. Brown Steve, Bessant John: The manufacturing strategy – capabilities links in mass customization and agile manufacturing – an exploratory study. *International Journal of Operations & Production Management*, 23 (7/8), 2003, str. 707-730.
24. Buehlmann Urs, Ragsdale Cliff, Gfeller B: A spreadsheet-based decision support system for wood panel manufacturing, 29 (3), 2000, str. 207-227.
25. Burgess Kevin, Singh Prakash, Koroglu Rana: Supply chain management: a structured literature review and implications for future research. *International Journal of Operations & Production Management* 26 (7), 2006, str. 703-729.
26. Buxey Geoff: Reconstructing inventory management theory. *International Journal of Operations & Production Management*, 26 (9), 2006, str. 996-1012
27. Carnieri Celso, Mendoza Guillermo, Luppold William: Optimal cutting of dimension parts from lumber with a defect: A heuristic solution procedure. *Forrest Products Journal*, 43 (9), 1993, str. 66-75.
28. Carvalho Valerio, Rodrigues Guimaraes: An LP-based approach to a two-stage cutting stock problem. *European Journal of Operational Research*, 84 (3), 1995, str. 580-589.
29. Carvalho Valerio: LP models for bin packing and cutting stock problems. *European Journal of Operational Research*, 141 (2), 2002, str. 253-273.
30. Carver Liz, Lockwood Freyja, Parsons Tom: Guidelines For Collaboration And The Role Of Collaborative Knowledge Sharing Platforms. *Vivace Project Report*, 2007. 50 str.
31. Chaitin Gregory: *Information, Randomness and Incompleteness: Papers on Algorithmic Information Theory*. Singapore : World Scientific Press, 1987.
32. Chan Felix, Bhagwat Rajat, Kumar N, Tiwari MK, Lam Philip: Development of a decision support system for air-cargo pallets loading problem: A case study. *Expert Systems with Applications*, 31 (2006), str. 472–485.
33. Chauhan, S. S., Martel, Alain. and DAmour, Sophie: Roll assortment optimization in a paper mill: An integer programming approach. *Computers & Operations Research*, 35 (2), 2008, str. 614-627
34. Chen Minder: *BPR Methodologies: Methods and Tools*. Business Process Engineering. Massachusetts: Kluwer Academic Publishers, 1999, str. 187-212.
35. Chen Xhengxin: *Computational Intelligence for Decision Support*. CRC Press, 1998, 380 str.
36. Cherri Adriana: O problema de corte de estoque com reaproveitamento das sobras de material. USP-Sao Carlos. Februar 2006.
37. Cherri Adriana, Arenales Marcos, Yanasse Horacio: Resolution of the unidimensional cutting stock problem with usable leftover. *ICORD-VI - International Conference on*

Operational Research for Development, Fortaleza, Brazil, 29. – 31. avgust, 2007, str. 474-486.

38. Cherri Adriana, Arenales, Marcos, Yanasse, Horacio: The one dimensional cutting stock problem with usable leftover – a heuristic approach, *European Journal of Operational Research* (2008). V tisku.
39. Chu Chengbin, Antonio Julien: Approximation algorithms to solve real-life multicriteria cutting stock problems. *Operations Research*, 47 (4), 1999, str. 495-508.
40. Cochran J, Ramanujan B: Carrier-mode logistics optimization of inbound supply chains for electronics manufacturing, *International Journal of Production Economics*, 103 (2006), str. 826–840.
41. Coffman Edward, Garey Richard, Johnson David: Approximation algorithms for bin packing – An updated survey. *Algorithm Design for Computer Systems Design*, New York: Springer-Verlag, 1984, str. 49-106.
42. Cox Andrew, Chicksand Lorna, Ireland Paul: *The E-business Report*. Boston: Earlsgate Press, 2001.
43. Cox Louis, Qiu Yuping, Kuehner Warren: Heuristic least-cost computation of discrete classification functions with uncertain argument values. *Annals of Operations Research*, 21 (1), 1989, str. 1-30.
44. Cui Yaodong, Chen Yuanyan, Wu JL: Selecting the best sheet length for the steel stock used in circular blank production. *IIE Transactions*, 38 (10), 2006, str. 829-836.
45. Cui Yaodong: A cutting stock problem and its solution in the manufacturing industry of large electric generators. *Computers & Operations Research*, 32 (7), 2005, str. 1709-1721.
46. Cui Yaodong: Generating optimal T-shape cutting patterns for circular blanks. *Computers & Operations Research*, 32 (1), 2005a, str. 143-152.
47. Čižman Anton, Černetič Janko: Improving competitiveness in veneers production by a simple-to-use DSS. *European Journal of Operational Research*, 156 (1), 2004, str. 241-260.
48. Čižman Anton, Urh Marko: A PC-based decision support system for optimal cutting of logs in veneers production. *Informatica*, 30 (2), 2006, str. 213-220.
49. Das Sanchoy, Abdel-Malek Layek: Modeling the flexibility of order quantities and lead-times in supply chains. *International Journal of Production Economics*, 85 (2), 2003, str. 171–181.
50. Daugherty Patricia, Myers Matthew, Authry Chad: Automatic replenishment programs: an empirical examination. *Journal of business logistics*, 20 (2), 1999, str. 63-82.

51. Davenport, T., J. Short: The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review* 31(4), 1990, str. 11-27.
52. Davies Islay, Green Peter, Rosemann Michael, Indulska Marta, Gallo Stan: How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3), 2006, str. 358-380.
53. Davila Antonio, Gupta Mahendra, Palmer Richard: Moving Procurement Systems to the Internet: the Adoption and Use of E-Procurement Technology Models. *European Management Journal*, 21(1), 2003, str. 11-23.
54. Degraeve Zeger, Peeters Marc: Optimal integer solutions to industrial cutting-stockproblems: Part 2, benchmark results. *INFORMS Journal on Computing*, 15 (1) 2003, str. 58-81.
55. Degraeve Zeger, Schrage Linus: Optimal Integer Solutions to Industrial Cutting Stock Problems. *Inform Journal on Computing*, 11 (4), 1999, str. 406-419.
56. Degraeve Zeger, Vandebroek Martina: A mixed integer programming model for solving a layout problem in the fashion industry. *Management Science*, 44 (3), 1998, str. 301-310.
57. Denton Brian, Diwakar Gupta, Keith Kawahir: Managing increasing product variety at integrated steel mills. *Interfaces*, 33 (2), 2003, str. 41-53.
58. Desel Jörg, Ervin Thomas: Modeling, Simulation and Analysis of Business Processes. *Business Process Management*. Berlin: Springer Verlag, 2000, str. 129-141.
59. DiToro Lou: BPM Software Report: BPM Simplicity with iGrafx. *BPMEnterprise Insights*, 2(5), 2007.
60. Dougherty James, Kohavi Ron, Sahami Mehran: Supervised and unsupervised discretization of continuous features. *Proceedings of Twelfth International Conference on Machine Learning*, San Francisco: Morgan Kaufmann, 1995, str. 194-202.
61. Dowsland KA, Gilbert M, Kendall G: A local search approach to a circle cutting problem arising in the motor cycle industry. *Journal of the Operational Research Society*, 58 (4), 2007, str. 429-438.
62. Dreiseitl Stephan, Ohno-Machado Lucila: Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, 35 (5-6), 2002, str. 352-359.
63. Dudzinski Krzysztof, Walukiewicz Stanislaw: Exact methods for the knapsack problem and its generalizations. *European Journal of Operational Research*, 28 (1), 1987, str. 3-21.
64. Dutta Amitava, Roy Rahul: A Process-Oriented Framework for Justifying Information Technology Projects in e-Business Environments. *International Journal of Electronic Commerce*, 9 (1), 2004, str. 49-68.

65. Dyckhoff Harald: A typology of cutting and packing problems. *European Journal of Operational Research*, 44 (2), 1990, str. 145-159.
66. Eatock Julie, Giaglis George, Paul Ray, Serrano, Alan: The Implications of Information Technology Infrastructure Capabilities for Business Process Change Success. *Systems Engineering for Business Process Change*. London: Springer-Verlag, 2000, str. 127-137.
67. Eiselt Horst, Sandblom C: *Integer Programming and Network Models*. Springer, 2000, 524 str.
68. Erjavec Jure, Gradišar Miro, Trkman Peter: Renovation of the Cutting Stock Process. *International Journal of Production Research*. 2008. V tisku.
69. Erjavec Jure, Gradišar Miro, Trkman Peter: Assessment of the optimal warehouse size for cutting stock. *International Journal of Production Economics*, 2008a. V recenziji.
70. Faggioli Enrico, Bentivoglio Carlo: Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research*, 110 (3), 1998, str. 564-575.
71. Fayyad Usama, Irani Keki: Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann, 1992, str. 1022-1027
72. Feller Andrew: E-business strategy and the integrated supply chain. *Transportation and Distribution*, 41 (5), 2000, str. 127-130.
73. Ferreira Soeiro, Neves Antonio, Fonseca Castro: A two-phase roll cutting problem. *European Journal of Operational Research*, 44 (2), 1990, str. 185-196.
74. Fisher Marshall: Worst case analysis of heuristic algorithm. *Management Science*, 26 (1), 1980, str. 1-17.
75. Förster Hildegard, Wäscher Gerhard: Pattern reduction in one-dimensional cutting stockproblem. *International Journal of Production Research*, 38 (7), 2000, str. 1657–1676.
76. Fowler Alan: Operations management and systemic modelling as frameworks for BPR. *International Journal of Operations & Production Management*, 18 (9/10), 1998, str. 1028-1056.
77. Fredriksson Peter, Gade Lars-Erik: Flexibility and rigidity in customization and build-to-order production. *Industrial Marketing Management*, 34 (7), 2005, str. 695-705.
78. Gansler Jacques, Lucyshyn William, Ross Kimberly: Digitally Integrating the Government Supply Chain: E-Procurement, E-Finance, and E-Logistics: IBM Endowment for the Business of Government, 2003.

79. Garey Michael, Johnson David: Computers and Intractability: A guide to the Theory of NP-Completeness. New York: W. H. Freeman, 1979. 340 str.
80. Gass Saul: Linear Programming, Methods and Applications. New York: McGraw-Hill, 1985. 532 str.
81. Gau Thomas, Wäscher Gerhard: CUTGEN1: A problem generator for the Standard One-Dimensional Cutting Stock Problem. European Journal of Operational Research, 84 (3), 1995, str. 572-579.
82. Ghodsi Reza, Sassani Farrokh: Online cutting stock optimization with prioritized orders. Assembly Automation, 25 (1), 2005, str. 66-72.
83. Ghodsi Reza, Sassani Farrokh: Real-time optimum sequencing of wood cutting process. International journal of production research, 43 (6), 2005a, str. 1127-1141.
84. Giaglis George: A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques. The International Journal of Flexible Manufacturing Systems, 13 (2001), str. 209-228.
85. Gilmore P. C., Gomory R. E.: A linear programming approach to the cutting stock problem. Operations Research, 9, 1961, str. 849-859.
86. Gilmore P. C., Gomory R. E.: A linear programming approach to the cutting stock problem, Part II. Baltimore: Operations Research, 11, 1963, str. 863-888.
87. Gradišar Miro, Jesenko Jože: Optimization of One Dimensional Cutting in Clothing Industry. Informatica, 20 (2), 1996, str. 211-221.
88. Gradišar Miro, Jesenko Jože, Resinovič Gortan: Optimization of roll cutting in clothing industry. Computer & Operations Research, 24 (10), 1997, str. 945-953.
89. Gradišar Miro, Resinovič Gortan, Jesenko Jože, Kljajić Miroljub: A sequential heuristic procedure for one-dimensional cutting. European Journal of Operational Research, 114 (3), 1999, str. 557-568.
90. Gradišar Miro, Kljajić Miroljub, Resinovič Gortan: A hybrid approach for optimization of one-dimensional cutting. European Journal of Operational Research, 119 (3), 1999a, str. 165-174.
91. Gradišar Miro, Resinovič Gortan, Kljajić Miroljub: Evaluation of algorithms for one-dimensional cutting. Working paper No. 90, Ljubljana: Ekonomska fakulteta, 1999b. 14 str.
92. Gradišar Miro, Trkman Peter, Indihar Štemberger Mojca: Exact solution of general one-dimensional cutting stock problem. Working paper No. 123. Ljubljana: Ekonomska fakulteta, 2001. 17 str.
93. Gradišar Miro, Resinovič Gortan, Kljajić Miroljub: Evaluation of algorithms for one-dimensional cutting. Computers & Operations Research, 29 (9), 2002, str. 1207-1220.

94. Gradišar Miro, Trkman Peter: A combined approach for solution of general onedimensional cutting stock problem. Working paper No. 124, Ljubljana: Ekonomska fakulteta, 2002a. 15 str.
95. Gradišar Miro: Uvod v informatiko. Ljubljana : Ekonomska fakulteta, 2003. 516 str.
96. Gradišar Miro, Trkman Peter: A combined approach to the solution to the general one-dimensional cutting stock problem. *Computers & Operations Research*, 32 (7), 2005, str. 1793-1807.
97. Gramani, M. and Franca, P.: The combined cutting stock and lot-sizing problem in industrial processes. *European Journal of Operational Research*, 174 (1), 2006, str. 509-521.
98. Hajizadeh Iman, Lee Chi-Guhn: Alternative configurations for cutting machines in a tube cutting mill. *European Journal of Operational Research*, 183 (3), 2007, str. 1385-1396.
99. Hammer Michael, Champy, James: *Reengineering the corporation*. New York: Harper Collins Books, 1993.
100. Han Jiwai, Kamber Micheline: *Data mining: concepts and techniques*. San Francisco: Morgan Kaufmann, 2001. 550 str.
101. Handfield Robert, Bechtel Christian: The role of trust and relationship structure in improving supply chain responsiveness. *Industrial Marketing Management*, 31 (4), 2002, str. 367-382.
102. Hässler RW: A heuristic programming solution to a nonlinear cutting stock problem. *Management Science*, 17(12), 1971, str. 793-802.
103. Hifi Mhand: Exact algorithms for the guillotine strip cutting/packing problem. *Computers & Operations Research*, 25 (11), 1998, str. 925-940.
104. Hifi Mhand, Hallah Rym: Strip generation algorithms for constrained two-dimensional two-staged cutting problems. *European Journal of Operational Research*, 172 (2), 2006, str. 515-527.
105. Hillier Frederick, Lieberman Gerald: *Introduction to Operations Research*. New York: McGraw-Hill, 2004.
106. Hinxman Anthony: The trim-loss and assortment problems: a survey. *European Journal of Operational Research*, 5 (1), 1980, str. 175-184.
107. Hoffman Karla: Combinatorial optimization: Current successes and directions for the future. *Journal of Computational and Applied Mathematics*, 124 (2000), str. 341-360.
108. Holland John: Genetic algorithms. *Scientific American*, 267 (1), 1992, str. 66-72.

109. Holthaus Oliver: Decomposition approaches for solving the integer one-dimensional cutting stock problem with different types of standard lengths. *European Journal of Operational Research*, 141 (2), 2002, str. 295-312.
110. Holthaus Oliver: On the best number of different standard lengths to stock for one-dimensional assortment problems. *International Journal of Production Economics*, 83 (3), 2003, str. 233-246.
111. Holweg Matthias, Bicheno John: Supply chain simulation - a tool for education, enhancement and endeavour. *International Journal of Production Economics*, 78 (2), 2002, str. 163-175.
112. Hommes Bart-Jan, Van Reijswoud Victor: Assessing the Quality of Business Process Modeling Techniques. 33rd Hawaii International Conference on System Sciences, Vol. 1, 2000, str. 1007.
113. Hong Ilyoo: A new framework for interorganizational systems based on the linkage of participants' roles. *Information & Management*, 39 (4), 2002, str. 261-270
114. Hsu Li-Ling: SCM system effects on performance for interaction between suppliers and buyers. *Industrial Management & Data Systems*, 105(7), 2005, str. 857-875.
115. Hunt Earl, Marin Janet, Stone Philip: *Experiments in Induction*. New York: Academic Press, 1966.
116. Hutchinson David: A new uniform pseudorandom number generator. *Communications of the ACM*, 9 (6), 1966, str. 432-433.
117. Hyafil Laurent, Rivest Ronald: Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5 (1), 1976, str. 15-17.
118. Indihar Štemberger Mojca, Jaklič Jurij, Popovič Aleš: Suitability of process maps for business process simulation in business process renovation projects. *Proceedings of the 2004 European Simulation Symposium*, Budapest, October 17-20, 2004.
119. Indihar Štemberger Mojca, Jaklič Jurij, Trkman Peter, Groznik Aleš: The role of business process management in a lean supply chain: two case studies = Vloga managementa poslovnih procesov v vitki oskrbovalni verigi: dve študiji primera. *Delovni zvezek št. 186*. Ljubljana: Ekonomska fakulteta, 2006. 24 str.
120. Jaklič Jurij, Groznik Aleš, Kovačič Andrej: Towards E-government - The role of simulation modeling. *Simulations in industry*, October 26-29, 2003, Delft: SCS, str. 257-262.
121. Jaklič Jurij, Trkman Peter, Groznik Aleš, Indihar Štemberger Mojca: Enhancing lean supply chain maturity with business process management. *Journal of Information and Organizational Sciences*, 30 (2), 2006, str. 205-223.
122. Johnson MP, Rennick C, Zak E: Skiving addition to the cutting stock problem in the paper industry. *Siam Review*, 39 (3), 1997, str. 472-483.

123. Johnston Robert: Rounding algorithms for cutting stock problems. *Asia-Pacific journal of operational research*, 3, 1986, str. 166-171.
124. Johnston Robert, Sadinlija Enes: A new model for complete solutions to one-dimensional cutting stock problems. *European Journal of Operational Research*, 153 (1), 2003, str. 176-183.
125. Kang Boo-Sik, Lee Jang-Hee, Shin Chung-Kwan, Yu Song-Jin, Park Sang-Chan: Hybrid machine learning system for integrated yield management in semiconductor manufacturing. *Expert Systems with Applications*, 15 (2), 1998, str. 123-132.
126. Keskinocak, P., Wu, F., Goodwin, R., Sesh, M., Akkiraju, R., Kumaran, S. , Derebail, A.: Scheduling solutions for the paper industry. *Operations Research*, 50 (2), 2002, str. 249–259.
127. Ketchen David, Hult Tomas: Toward greater integration of insights from organization theory and supply chain management. *Journal of Operations Management*, 25(2), 2007, str. 455-458.
128. Kettinger William, Teng James, Guha Subashish: Business process change: a study of methodologies, techniques, and tools. *MIS Quarterly*, 21 (1), 1997, str. 55-80.
129. Khalifa Y, Salem O, Shahin A: Cutting stock waste reduction using genetic algorithms. *Proceedings of the 8th annual conference on Genetic and evolutionary computation table of contents*. Seattle, Washington, ZDA, 2006, str. 1675-1680
130. Kim Seung-Lae, Ha Daesung: A JIT lot-splitting model for supply chain management: Enhancing buyer–supplier linkage. *International Journal of Production Economics*, 86 (1), 2003, str. 1-10.
131. Kirikova Marite, Makna Janis: Renaissance of business process modelling. *Information Systems Development: Advances in Theory, Practice and Education*. Springer, 2005, str. 403-414.
132. Klamroth K, Wiecek M: Dynamic Programming Approaches to the Multiple Criteria Knapsack Problem. *Naval Research Logistics*, (47) 2000, str. 57-76.
133. Kobayashi Takashi, Tamaki Masato, Komoda Norihisa: Business process integration as a solution to the implementation of supply chain management systems, *Information & Management* 40 (8), 2003, str. 769–780.
134. Kokol Peter, Hleb Babič Špela, Podgorelec Vili, Zorman Milan: *Inteligentni sistemi*. Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko, 2001. 211 str.
135. Kopač Janez: Cutting forces and Their Influence on the Economics of Machining. *Strojniški vestnik - Journal of Mechanical Engineering* , 48(3), 2002, str. 121-132.
136. Kopanaki Evangelia, Smithson Steve: The Impact of a Continuous Replenishment Program on Organsiatonal Flexibility. V: Monteiro Joao (ur.) *Towards the*

- Knowledge Society: Ecommerce, Ebusiness, and Egovernment. Kluwer Academic Publishers, 2002.
137. Kos Leon, Duhovnik Jože: Cutting optimization with variable-sized stock and inventory status data. *International Journal of Production Research*, 40 (10), 2002, str. 2289-2301.
 138. Kovačič Andrej, Jaklič Jure, Indihar Štemberger Mojca, Groznik Aleš: Prenova in informatizacija poslovanja. Ljubljana: Ekonomska fakulteta, 2004. 345 str.
 139. Koza John, Bennett Forrest, Andre David, Keane Martin: Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann, 1999. 1154 str.
 140. Kozak Jernej: Podatkovne strukture in algoritmi. Ljubljana: Društvo matematikov, fizikov in astronomov SR Slovenije, 1986. 388 str.
 141. Krajewski Lee, Wei Jerry, and Tang Ling-Lang: Responding to schedule changes in build-to-order supply chains. *Journal of Operations Management*, 23 (5), 2005, str. 452-469.
 142. Ladanyi L, Lee J, Lougee-Heimer R: Rapid prototyping of optimization algorithms using COIN-OR: A case study involving the cutting-stock problem. *Annals of Operations Research*, 139 (1), 2005, str. 243-265.
 143. Lambert Douglas, Cooper Martha: Issues in supply chain management. *Industrial Marketing Management*, 29 (1), 2000, str. 65-83.
 144. Land AH, Doig AG: An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 28 (3), 1960, str. 496-520.
 145. Lee Jon: In situ column generation for a cutting-stock problem. *Computers & Operations Research*, 34 (8), 2007, str. 2345-2358.
 146. Lee RG, Dale BG: Business process management: a review and evaluation. *Business Process Re-engineering & Management Journal*, 4 (3), 1998, str. 214-225.
 147. Lengnick-Hall Cynthia: Customer contributions to quality a different view of the customer-oriented firm. *Academy of Management Review*, 21 (3), 1996, str. 791-824.
 148. Lenič Mitja, Kokol Peter, Yamamoto Ryuichi: Internet storitev za podporo pri odločanju v medicini. *Dnevi slovenske informatike 2002*, Ljubljana: Slovensko društvo Informatika, 2002, str. 501-505.
 149. Li Suhong, Rao Subba, Ragu-Nathan TS, Ragu-Nathan Bhanu: Development and validation of a measurement instrument for studying supply chain management practices. *Journal of Operations Management*, 23 (6), 2005, str. 618-641.
 150. Liang Ko-Hsin, Yao Xin, Newton Charles, Hoffman David: A new evolutionary approach to cutting stock problems with and without contiguity. *Computers & Operations Research*, 29 (12), 2002, str. 1641-1659.

151. Lin Pei-Chun: Application of column generation approach in minimizing waste and predicting future inventory of one dimensional cutting stock problem. *Journal of Statistics & Management Systems*, 8 (3), 2005, str. 569-585.
152. Lockamy Archie, McCormack Kevin: The development of a supply chain management process maturity model using the concepts of business process orientation. *Supply Chain Management: An International Journal*, 9 (4), 2004, str. 272-278.
153. Lodi Andrea, Martello Silvano, Vigo Daniele: Heuristic and meta heuristic approaches for a class of two-dimensional bin packing problems. *Inform Journal on Computing*, 11, 1999, str. 345-357.
154. Lodi Andrea, Martello Silvano, Vigo Daniele: Recent advances on twodimensional bin packing problems. *Discrete Applied Mathematics*, 123 (1-3), 2002, str. 379-396.
155. Marques Fabiano, Arenales Marcos: The constrained compartmentalised knapsack problem. *Computers & Operations Research*, 34 (7) 2007, str. 2109-2129.
156. McCormack Kevin, Johnson, William: *Business Process Orientation: Gaining the E-Business Competitive Advantage*. Delray Beach : St Lucie Press, 2000.
157. Meixell Mary, Gargeya Vidhyaranya: Global supply chain design: A literature review and critique. *Transportation Research Part E: Logistics and Transportation Review*, 41 (6), 2005, str. 531-550.
158. Melao Nuno, Pidd, Michael: A conceptual framework for understanding business processes and business process modelling. *Information Systems Journal*, 10 (2), 2000, str. 105–129.
159. Menon Syam, Schrage Linus: Order allocation for stock cutting in the paper industry. *Operations Research*, 50 (2002) 2, str. 324-332.
160. Mladenović Dunja, Lavrač Nada, Bohanec Marko, Moyle Steve: *Data Mining and Decision Support: Integration and Collaboration*. Kluwer Academic Publishers, 2003. 304 str.
161. Morgan Leslie, Morton Alysse, Daniels Richard: Simultaneously determining the mix of space launch vehicles and the assignment of satellites to rockets. *European Journal of Operational Research*, 172 (3), 2006, str. 747–760.
162. Motwani Jaideep, Larson Lars, Ahuja Suraja: Managing a global supply chain partnership. *Logistics Information Management*, 11 (6), 1998, str. 349-354.
163. Murthy Sreerama: Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, 2 (4), 1997, str. 345-389.

164. Nilsson Anders: Information systems development (ISD): past, present, future trends. *Information Systems Development: Advances in Theory, Practice and Education*. Springer, 2005, str. 29-40.
165. Nilsson Nils: *Problem Solving Methods in Artificial Intelligence*. New York: McGraw-Hill, 1971.
166. Nonaas Sigrid and Thorstenson Anders: A combined cutting stock and lot sizing problem. *European Journal of Operational Research*, 120(2), 2000, str. 327-342.
167. Nonaas Sigrid and Thorstenson Anders: Solving a combined cutting-stock and lot-sizing problem with a column generating procedure. *Computers & Operations Research*, 35 (10), 2008, str. 3371-3392.
168. Onwubolu Godfrey, Mutingi Michael: A genetic algorithm approach for the cutting stock problem. *Journal of Intelligent Manufacturing*, 14 (2), 2003, str. 209-218.
169. Ould Martyn: *Business Processes: Modelling and Analysis for Re-engineering and Improvement*, New York: John Wiley & Sons, 1995.
170. Pagel Darrick: Managing for optimal performance through effective coordination of the supply chain. *Production and Inventory Management Journal*, 40 (1), 1999, str. 66-70.
171. Papadimitriou Christos, Steiglitz Kenneth: *Combinatorial optimization*. New Jersey: Englewood Cliffs, 1982. 496 str.
172. Park Stephen, Miller Keith: Random number generators: Good ones are hard to find. *Communications of the ACM*, 31 (10), 1988, str. 1192-1201.
173. Paull Allan: *Linear Programming: A Key to Optimum Newsprint Production*. Pulp and Paper Magazine of Canada, 57 (1), 1956, str. 85-90.
174. Pernici Barbara, Weske Mathias: Business process management. *Data & Knowledge Engineering*, 56 (1), 2006, str. 1-3
175. Persson Fredrik, Olhager Jan: Performance simulation of supply chain designs. *International Journal of Production Economics*, 77 (3), 2002, str. 231-245.
176. Pierce John: *Some Large-Scale Production Scheduling Problems in the Paper Industry*. Englewood Cliffs: Prentice-Hall, 1964. 255 str.
177. Pine Joseph: *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press, 1992.
178. Pisinger David: Heuristics for the container loading problem. *European Journal of Operational Research*, 141 (2), 2002, str. 382-392.
179. Plossl George, Wight Oliver: *Production and Inventory Control*. Englewood Cliffs: Prentice Hall, 1967. 432 str.

180. Popovič Aleš: Uporabnost simulacijskega modeliranja pri projektih prenove poslovanja. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2005. 94 str.
181. Popovič Aleš, Indihar Štemberger Mojca, Jaklič Jurij: Applicability of process maps for simulation modeling in business process change projects. *Interdisciplinary journal of information, knowledge, and management*, 2006, 1, str. 119-133
182. Prasad Biren: Hybrid re-engineering strategies for process improvement. *Business Process Management Journal*, 5 (2), 1999, str. 178-197.
183. Presutti William: Supply management and e-procurement: creating value added in the supply chain. *Industrial Marketing Management*, 32(3), 2003, str. 219-226.
184. Puschmann Thomas, Alt Rainer: Successful use of e-procurement in supply chains. *Supply Chain Management - An International Journal*, 10(2), 2005, str. 122-133.
185. Quinlan Ross: Decision trees and multi-valued attributes. *Machine Intelligence*, 11, 1988, str. 305-318.
186. Quinlan Ross: C 4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann Publishers, 1993. 302 str.
187. Quinlan Ross: Improved Use of Continuous Attributes in C4.5. *Journal of Artificial Intelligence Research*, 4, 1996, str. 77-90.
188. Ragsdale Cliff, Zobel Christopher: The Ordered Cutting Stock Problem. *Decision Sciences*, 35 (1), 2004, str. 83-100.
189. Rajagopal, Rajagopal, A: Buyer–supplier relationship and operational dynamics. *Journal of the Operational Research Society*, 2008. V tisku
190. Rissanen Jorma: A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11 (2), 1983, str. 416-431.
191. Rodriguez Maria, Vecchietti, Aldo: Enterprise optimization for solving an assignment and trim-loss non-convex problem. *Computers and Chemical Engineering*, 2008. V tisku.
192. Rönnqvist Mikael: A method for the cutting stock problem with different qualities. *European Journal of Operational Research*, 83 (1), 1995, str. 57-68.
193. Saad Omar, El-shafei Mervat, Ezzat Lamia: On Treating Multiobjective Cutting Stock Problem in the Aluminum Industry under Fuzzy Environment. *Journal of Applied Sciences Research*, 3(10), 2007, str. 938-945.
194. Safavian Rasoul, Landgrebe David: A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21 (3), 1991, str. 660-674.

195. Sardinias RQ, Santana MR, Brindis EA: Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering applications of artificial intelligence*, 19 (2), 2006, str. 127-133.
196. Scheitauer Guntram, Terno Johannes: The modified integer round-up property for one-dimensional cutting stock problem. *European Journal of Operational Research*, 84 (3), 1995, str. 562-571.
197. Schilling Gordian, Georgiadis Michael: An algorithm for the determination of optimal cutting patterns. *Computers & Operations Research*, 29 (8), 2002, str. 1041-1058.
198. Scholz-Reiter, B. and Stickel, E., (eds.), *Business Process Modelling*. Springer-Verlag, Berlin, 1996.
199. Schrijver Alexander: *Theory of linear and integer programming*. Chichester: John Wiley & Sons, 1986. 471 str.
200. Shahin, A., Salem, O.: Using Genetic Algorithms in Solving the One-Dimensional Cutting Stock Problem in the Construction Industry. *National Research Council of Canada, Canadian Journal of Civil Engineering*, 31 (2), 2004, str. 321-332.
201. Shannon Claude: A Mathematical Theory of Communication. *The Bell System Technical Journal*. 27, 1948, str. 379–423, str. 623–656.
202. Shen Xianjun, Li Yuanxiang, Yang Jincai, Yu Li: A Heuristic Particle Swarm Optimization for Cutting Stock Problem Based on Cutting Pattern. *Lecture Notes in Computer Science*, 4490 (2007), str. 1175–1178.
203. Shim JP, Warkentin Merrill, Courtney, James, Power Daniel, Shard Ramesh, Carlsson Christer: Past, present, and future of decision support technology, 33 (2), 2002, str. 111-126.
204. Smith Howard, Fingar Peter: *Business Process Management: The Third Wave*. Tampa: Meghan-Kiffer Press, 2003. 311 str.
205. Stadtler Hartmut: A one-dimensional cutting stock problem in the aluminium industry and its solution. *European Journal of Operational Research*, 44 (2), 1990, str. 209-223.
206. Stosic N: A geometric approach to calculating tool wear in screw rotor machining, *International Journal of Machine Tools and Manufacture*, 46 (15), 2006, str. 1961-1965.
207. Suliman Saad: Pattern generating procedure for the cutting stock problem. *International Journal of Production Economics*, 74 (1-3), 2001, str. 293-301.
208. Sung Tso-Chung, Weng Wei-Chu, Yang Chien-Tung: A Two-Stage Optimization of Piece Arrangement for the Cutting Problem in Shipbuilding. *Journal of Marine Science and Technology*, 12 (3), 2004, str. 175-182.

209. Sweeney Paul, Paternoster Elizabeth: Cutting and packing problems: A Categorised, Application-Orientated Research Bibliography. *Journal of the Operational Research Society*, 43 (7), 1992, str. 691-706.
210. Škrinjar Rok, Hernaus Tomislav, Indihar Štemberger Mojca: SCM maturity in Central Europe. V: McCormack Kevin: *Business process maturity : theory and application*. Seattle: BookSurge, 2007, str. 219-236.
211. Srivardhana T, Pawlowski Suzanne: ERP systems as an enabler of sustained business process innovation: A knowledge-based view. *Journal of Strategic Information Systems* 16 (1), 2007, str. 51-69
212. Tarim Armagan, Kingsman Brian: The stochastic dynamic production/inventory lot-sizing problem with service-level constraints *International Journal of Production Economics*, 88 (1), 2004, str. 105-119.
213. Thomas Adin: *Stock Control in Manufacturing Industries*. Hampshire: Gower Press, 1980. 221 str.
214. Thuraisingham Bhavani: *Data Mining*. CRC Press, 1998, 288 str.
215. Trkman Peter, Gradišar Miro: Choice of method for General One-Dimensional Cutting Stock Problem. V: Mastokaris Nikos, Kluev Vitalij, Koruga Djuro. *Advances in simulation, systems theory and systems engineering, (Electrical and computer engineering series)*. WSEAS Press, 2002, str. 17-21.
216. Trkman Peter, Gradišar Miro: Eksaktna rešitev problema enodimenzionalnega razreza. *Dnevi slovenske informatike 2002*. Ljubljana: Slovensko društvo Informatika, 2002a, str. 355-359.
217. Trkman Peter, Gradišar Miro: Optimizacija postopka razreza materiala. *Optimization of the cutting-stock process. Journal of Mechanical Engineering/Strojniški vestnik*, 49 (9), 2003, str. 469-475.
218. Trkman Peter, Gradišar Miro: A comparison between exact and approximate method for solution of general one-dimensional cutting stock problem. *Informatica*, 27 (4) 2003a, str. 495-501.
219. Trkman Peter, Gradišar Miro: Ljubljana: Minimizacija izgub pri kontinuiranem razrezu materiala = Trim loss minimization in continuous cutting stock problem. *DSI - Dnevi slovenske informatike 2005*, Portorož, Slovenija, 13.-15. april. Ljubljana: Slovensko društvo Informatika, 2005, str. 448-453.
220. Trkman Peter, Jaklič Jure, Indihar Štemberger, Groznik Aleš: Vloga učinkovite izmenjave informacij pri integraciji procesov znotraj oskrbovalne verige = The role of efficient distribution of information for integration of supply chain. *DSI - Dnevi slovenske informatike 2005*. Ljubljana: Slovensko društvo Informatika, 2005a, str. 21-26.

221. Trkman Peter, Indihar Štemberger Mojca, Jaklič Jure: Information Transfer in Supply Chain Management. Issues in Informing Science and Information Technology Education, 2, 2005b, str. 559-573.
222. Trkman Peter, Gradišar Miro: The Cutting Stock Problem. V: Selected Decision Support Models for Production and Public Policy Problems. Ljubljana: Slovenian Society Informatika, 2005c, str.135-164.
223. Trkman Peter, Gradišar Miro: Mass customization in cutting stock process. Wseas Transactions on Business and Economics, 2005d, 2 (4), str. 219-224.
224. Trkman Peter, Štemberger Mojca, Jaklič Jurij, Groznik, Aleš: Process approach to supply chain integration. Supply Chain Management - An International Journal, 12(2), 2007, str.116-128.
225. Trkman Peter, Gradišar Miro: One-dimensional cutting stock optimization in consecutive time periods. European Journal of Operational Research, 179(2), 2007, str. 291-301.
226. Trkman Peter, McCormack Kevin: Simulation of procure-to-pay-process. INFORMS, Washington, september 2008. Sprejeto v objavo.
227. Tsai Jung-Fa, Huang Yao-Huei: A cutting and manufacturing optimization support system. 35th International Conference on Computers and Industrial Engineering, 2005, str. 1959-1964
228. Umetani Shunji, Yagiura Mutsunori, Ibaraki Toshihide: One-dimensional cutting stock problem to minimize the number of different patterns. European Journal of Operational Research, 146 (2), 2003, str. 388-402.
229. Vahrenkamp Richard: Random search in the one-dimensional cutting stock problem. European Journal of Operational Research, 95 (1), 1996, str. 191-200.
230. Vanderbeck Francois: Exact algorithm for minimizing the number of setups in the one-dimensional cutting stock problem. Operations Research, 48 (6), 2000, str. 915-926.
231. Vasko Francis, Newhart Dennis, Stott Kenneth: A hierarchical approach for onedimensional cutting stock problems in the steel industry that maximizes yield and minimizes overgrading. European Journal of Operational Research, 114 (1), 1999, str. 72-82.
232. Venkateswarlu Pulakanam: The Trim-Loss Problem in a Wooden Container Manufacturing Company. Journal of Manufacturing Systems, 20 (3), 2001, str. 166-176.
233. Vitale Rob, Giglierano Joe: Business to Business Marketing Analysis and Practice in a Dynamic Environment. Mason: South-Western/Thomas Learning. 534 str.

234. Wagner Brett: A genetic algorithm for one-dimensional bundled stock cutting. *European Journal of Operational Research*, 117 (2), 1999, str. 368-381.
235. Wall Brian: At the crossroads to a greener future. *Works Management*, 58 (12), 2005, str. 30-32.
236. Wang Pearl, Wäscher Gerhard: Cutting and packing, editorial. *European Journal of Operational Research*, 141 (2), 2002, str. 239-240.
237. Wang Minhong, Wang Huaiqing: From process logic to business logic: a cognitive approach to business process management. *Information & Management* 43 (2), 2006, str. 179–193.
238. Wäscher Gerhard: An LP-based approach to cutting stock problems with multiple objectives. *European Journal of Operational Research*, 44 (2), 1990, str. 175-184.
239. Wäscher Gerhard, Gau Thomas: Generating Almost Optimal Solutions for the Integer One-dimensional Cutting Stock Problem. Working Paper No. 94/06, Institut für Wirtschaftswissenschaften, Technische Universität Braunschweig, 1994.
240. Wäscher Gerhard, Haußner Heike, Schumann Holger: An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183 (3), 2007, str. 1109–1130.
241. Weng Wei-Chu, Sung Tso-Chung: The optimization of line cutting procedure for ship hull construction by an effective tabu search. *International Journal of Production Research*, 2008. V tisku.
242. Westerlund Tapio, Harjunkoski Iiro, Isaksson Johnny: Solving a Production Optimization Problem in a Paper-Converting Mill with MILP. *Computers & Chemical Engineering*, 22 (4-5), 1998, str. 563-570.
243. White Allan, Liu Wei Zhang. Technical note: Bias in information-based measures in decision tree induction. *Machine Learning*, 15 (3), 1994, str. 321-329.
244. White Stephen: Introduction to BPMN. *BPTrends*, julij 2004, str. 1-11.
245. Williamson Elizabeth, Harrison David, Jordan Mike: Information systems development within supply chain management. *International Journal of Information Management*, 24 (5), 2004, str. 375–385.
246. Winston Wayne: *Operations research: applications and algorithms*. Belmont: Duxbury Press, 1993. 1318 str.
247. Yanasse Horacio, Limeira Marcelo: A cutting stock problem and its solution in the manufacturing industry of large electric generators. *Computers & Operations Research*, 32 (7), 2005, str. 1709-1721.
248. Yang Chien-Tung, Sung Tso-Chung, Weng Wei-Chu: An improved tabu search approach with mixed objective function for one-dimensional cutting stock problems. *Advances in Engineering Software*, 37 (8), 2006, str. 502–513.

249. Yazgac Tulin, Ozdemir Rifat: A cutting sequencing approach to modular manufacturing. *Journal of Manufacturing Technology Management*, 15 (1), 2004, str. 20-28.
250. Yen Chia, Wond David, Jang Shi: Solution of trim-loss problem by an integrated simulated annealing and ordinal optimization approach. *Journal of Intelligent Manufacturing*, 15 (5), 2004, str. 701-709.
251. Yeung L, Tang W: A hybrid genetic approach for garment cutting in the clothing industry. *IEEE Transactions on Industrial Electronics*, 50 (3), 2003, str. 449-455.
252. Yildiz AR, Ozturk F: 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), 2006, str. 2041-2053.
253. Yuen Boon, Richardson Ken: Establishing the optimality of sequencing heuristics for cutting stock problems. *European Journal of Operational Research*, 84 (3), 1995, str. 590-598.
254. Zak Eugene: Modeling multistage cutting stock problems. *European Journal of Operational Research*, 141 (2), 2002, str. 313-327.
255. Zak Eugene: Row and column generation technique for a multistage cutting stock problem. *Computers & Operations Research*, 29 (9), 2002a, str. 1143-1156.

Viri

1. Applause Consulting: ECR Compass. [URL: http://www.applause.hu/terms_e.htm], 12. 3. 2005.
2. BPR Glossary of Terms [<http://www.gao.gov/special.pubs/bprag/bprgloss.htm>]
3. Breslow Leonard, Aha David. Simplifying decision trees: A survey. Technical Report. Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, 1996 [URL: www.aic.nrl.navy.mil/papers/1996/AIC-96-014.ps].
4. Bridgefield Group ERP/Supply Chain Glossary. [URL: <http://www.bridgefield-group.com/glos8.htm>], 1. 11. 2005.
5. Cook Stephen: The P versus NP problem. Clay Mathematics Institute, maj 2000 [URL: http://www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf].
6. Hommes BJ: Overview of Business Process Modelling Tools, 2001 [URL: <http://is.twi.tudelft.nl/~hommes/scr3tool.html>].
7. Igrafx, 2007 [URL: <http://www.igrafx.com/>]

8. ISlovar: Slovar informatike. Slovensko društvo informatika [URL: <http://www.islovar.org/>], 21. 10. 2005.
9. Morgan Steven: A Comparison of Simplex Method Algorithms. University of Florida, 1997 [URL: <http://www.cise.ufl.edu/~davis/Morgan/>].
10. NP-complete. [URL: <http://c2.com/cgi/wiki?NpComplete>], 4. 9. 2002.
11. SCOR Version 7.0. Overview. Supply-Chain Operations Reference Model. Supply-chain Council, 2005 [URL: www.supply-chain.org].
12. Sherrod Phil: DTreg: Classification and Regression Trees And Support Vector Machines (SVM) For Predictive Modeling and Forecasting. [URL: <http://www.dtreg.com/>], 7. 5. 2005.
13. Wäscher, Gerhard, Haußner, H. and Schumann, H., 2006. Categorised literature [URL: <http://www-f.uni-magdeburg.de/mansci/themes/ManSci/daten/daten/forsch/typoa03.pdf>]
14. Web of Science, 2008. Thompson Scientific [URL: <http://isiknowledge.com>]