Planiranje materialnih potreb z uporabo razširjene formulacije problema pakiranja

Dejan Gradišar, Miha Glavan
Institut Jožef Stefan
Jamova 39, 1000 Ljubljana
dejan.gradisar@ijs.si, miha.glavan@ijs.si

Material resource planning using extended bin packing problem formulation

In this paper we discuss the problem of material resource planning, where time and grouping restrictions has to be taken into account. The problem is motivated by steel processing production process, where raw materials are provided as defined with purchase orders and has to be cut in smaller parts in order to satisfy the needs of work orders. It is necessary to minimize the leftovers and if needed also the time delays. Purchase orders define expected time delivery and the size of delivered material. Work orders reflect the due dates, quantities of needed material and specific restriction, which determine which work orders can be produced from one group of raw materials. Based on possible problems that occurs in the production, we have designed several optimization problems in the form of mixed-integer liner programming. Here we have used the generalized form of bin packing problem (BPP) formulation. Formulation was extended with variable bin sizes, time and batching constrains. The method was tested on data obtained from industrial environment.

Povzetek

V prispevku obravnavamo problem planiranja materialnih potreb, kjer je potrebno upoštevati tako časovne kot tudi šaržne omejitve. Problem je motiviran s strani proizvodnje za razrez pločevine. Vhodni material je določen z nabavnimi nalogi in mora biti razrezan na manjše kose, kot je to določeno z delovnimi nalogi, pri tem pa je potrebno minimizirati odpad in, če je to potrebno, tudi zamude delovnih nalogov. Nabavni nalog je določen s časom dostave in njegovo velikostjo, delovni nalog pa z rokom za dokončanje posameznih nalog, količino materialnih potreb in še s posebno zahtevo, ki določa da morajo biti vse naloge enega delovnega naloga proizvedene iz iste skupine (šarže) vhodnega materiala. Glede na možne probleme, ki lahko nastopajo v proizvodnji, smo zasnovali več formalnih zapisov optimizacijskega problema v obliki celoštevilčnega linearnega programiranja. Pri tem smo uporabili več razširitev oblike formulacije problema pakiranja (ang. *Bin Packing Problem*). Razširitev se je nanašala na spremenljive velikosti košev, upoštevanja časovnih in pa tudi šaržnih omejitev. Metoda je bila preizkušena na podatkih, pridobljenih iz industrijskega okolja.

1 Uvod

Dandanes je vsak proizvodni proces podprt s sistemi za planiranje in vodenje. Ti podpirajo različne aktivnosti v proizvodnji, od planiranja do podrobnega razvrščanja proizvodnih operacij [11]. Vseeno pa se pogost zgodi, da implementirani sistemi ne podpirajo problemov dovolj celovito, zato so potrebne specifične rešitve. Prispevek se ukvarja z problemom določanja optimalnih potreb po materialu (Material Requirements Planning – MRP). Izdelki se proizvajajo glede na MRP plan, za katerega pa morajo biti na voljo ustrezni polizdelki.

Problem lahko predstavimo kot klasični problem pakiranja (ang. *Bin Packing Problem, BPP*), ki se pojavlja v raziskavah že od zgodnjih 70ih [9][5]. Problem se ukvarja s problemom pakiranja *m* predmetov v *n* identičnih košev. Vsak predmet je potrebno dodeliti enemu izmed košev, pri tem pa ne sme biti presežena kapaciteta koša. Hkrati pa je cilj uporabiti čim manj košev. Problem pakiranja lahko predstavimo kot celoštevilčni linearni program. Problem je NP-težak in je običajno rešen s pomočjo hevrističnih algoritmov. Podroben pregled različnih matematičnih predstavitev problemov in algoritmov reševanja je podan v [2].

Da je problem pakiranja uporaben v različnih praktičnih problemih, se pojavlja več razširjenih oblik klasične oblike problema pakiranja. V papirnicah, jeklarski ali tekstilni industriji se pogostokrat uporablja ekvivalentna predstavitev problema v obliki problema optimalnega razreza (ang. *Cutting Stock*) [1]. Ena od pogostih razširjenih oblik obravnava problem z dvema ali več dimenzijami [3][8]. Za predstavitev bolj realnih problemov se pojavljajo formulacije, ki predvidevajo koše različnih velikosti [4][7].

Formulacij BPP se pogosto uporablja tudi za predstavitev problemov, kjer je potrebno upoštevati tudi časovno komponento (npr. razvrščanje). Enostaven primer je, ko modeliramo opravila, katerih čase trajanja predstavimo z velikostjo predmetov oziroma košev. Naprednejši zapisi uporabljajo čase trajanj in roke za izdelavo kot dodatne omejitve v optimizacijskem problemu. Te so lahko realizirane kot striktne ali pa tudi kot mehke omejitve. Primer uporabe takega problema je predstavljen v [12].

Problem, ki ga rešujemo v prispevku, prihaja iz procesa razreza pločevine, z zakonitostmi, ki jih v z obstoječimi rešitvami ni moč opisati. Surovine, ki so naročene z nabavnimi nalogi, so dostavljene v šaržah. Odvisno od časa dostave pa se lahko razlikujejo tudi vizualne lastnosti surovine. Če na to nismo pazljivi, je lahko končni izdelek narejen iz surovin različnih lastnosti, kar vpliva na kvaliteto izdelka, lahko pa tudi na dodatne stroške v primeru zahtevanih popravil. Potrebujemo torej rešitev, ki bo v pomoč proizvodnemu planerju pri naročanju in pripravi polizdelkov za nadaljnjo sestavo. Tak plan mora poleg minimiziranja izmeta upoštevati tudi več omejitev, kot so velikosti polizdelkov in surovin, upoštevanje rokov za pripravo polizdelkov in pa dejstvo, da je končni izdelek lahko narejen le iz iste skupine surovin.

V naslednje poglavju je predstavljena razširjena formulacija problema pakiranja, s katero lahko formalno predstavimo problem planiranja materialnih potreb. V tretjem poglavju je na ilustrativnem primeru prikazana uporaba prestavljene formulacije. V četrtem poglavju so podani zaključki.

2 Formulacija BPP problema za planiranje materialnih potreb

2.1 Predstavitev klasičnega BPP problema

Klasični problem pakiranja obravnava problem, kjer je potrebno *m* predmetov (ang. *Items*) velikosti s_i zložiti v *n* identičnih košev (ang. *Bins*) s kapaciteto *c*. Cilj je uporabiti minimalno število košev, pri čemer vsota posameznih predmetov v ene košu na sme preseči velikosti le-tega (*c*). Množico vseh predmetov označimo z *I*, množico razpoložljivih košev pa z *B*. Problem lahko predstavimo kot celoštevilčni linearni program, glej enačbo (1).

$$z_1: \quad min \sum_{j=1}^{n} y_j \qquad (1)$$

pri pogojih

$$\sum_{i=1}^{m} s_i \cdot x_{ij} \leq c \cdot y_j, j = 1 \dots n$$

$$\sum_{i=1}^{m} x_{ij} = 1, j = 1 \dots n$$

$$y_j \in \{0,1\}$$

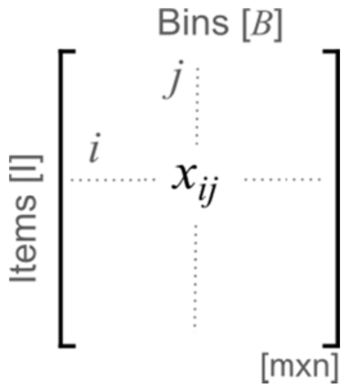
$$x_{ij} \in \{0,1\}$$

Pri tem sta y_j in x_{ij} binarni odločitveni spremenljivki, ki zavzameta vrednost:

$$y_j = \begin{cases} 1 & \text{koš } j \text{ je uporabljen} \\ 0 & \text{drugače,} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{predmet } i \text{ uporabljen v košu } j \\ 0 & \text{drugače.} \end{cases}$$

Odločitveno spremenljivko x_{ij} lahko predstavimo z matriko na sliki 1.



Slika 1: Odločitvena spremenljivka x_{ij}.

Prva omejitev v enačbi *(1)* zagotavlja, da vsota velikosti vseh predmetov v košu B_j ne preseže vrednosti kapacitete koša *c*, medtem ko druga omejitev zagotavlja, da je en predmet dodeljen le enemu košu.

2.2 Spremenljiva velikost košev

V praksi se večinoma soočamo s problemi, kjer so koši različnih velikosti. Velikosti posameznih košev, ki si niso enaki, označimo z c_j. Množica predmetov je lahko dodeljena košu B_j le v primeru, če vsota velikosti vseh teh predmetov ne preseže kapacitete koša c_j. Iz tega razloga je potrebno prvo omejitev iz enačbe (1) prilagoditi, glej (2).

$$\sum_{i=1}^{m} s_i \cdot x_{ij} \leq c_j \cdot y_j, j = 1 \dots n \quad (2)$$

Pri pakiranju je običajno zaželeno, da je vsak uporabljen koš čim bolj zapolnjen. V primeru identičnih košev optimizacijski kriterij iz (1) minimizira tako število uporabljenih košev kot tudi velikost preostankov (ang. *leftover*). To pa v primeru reševanja problema z različno velikimi koši običajno ne drži. V tem primeru je potrebno prilagoditi tudi optimizacijski kriterij (1) in uporabiti (3), ki minimizira vsoto vseh preostankov.

$$z_2: \quad min \sum_{j=1}^{n} (c_j \cdot y_j - \sum_{i=1}^{m} s_i \cdot x_{ij}) \quad (3)$$

2.3 Upoštevanje časovne komponente

Dodatna razširitev formulacije problema je potrebna, če želimo upoštevati tudi časovne omejitve. V proizvodnih okoljih imamo pogosto opravka s problemi, kjer mora biti koš B_j na voljo, preden so na voljo predmeti I_i, ki jih želimo dodeliti v koš. V ta namen definiramo predmet I_i z elementoma (s_i, d_i), kjer je d_i rok, do katerega mora biti predmet I_i na voljo (ang. *due date*). Koš definiramo z elementoma (c_j, a_j), kjer a_j določa, kdaj bo na voljo B_j (ang. *delivery time*). V tem delu predvidevamo dva optimizacijska problema. V prvem upoštevamo trde časovne omejitve, kjer mora biti B_j na voljo pred časom, kot ga določa rok za izdelavo predmeta I_i ($d_j \geq a_j$), v drugem pa upoštevamo mehke omejitve, kjer je zamujanje dovoljeno, vendar je to v kriterijski funkciji dodatno kaznovano.

V primeru obravnave problema, kjer zamude niso dovoljene, moramo v našem optimizacijskem problemu upoštevati dodatno omejitev (4).

$$a_j \cdot x_{ij} \leq d_i \; i = 1 \dots m, j = 1 \dots n \quad (4)$$

Praktično bolj uporaben je primer, ko so zamude dovoljene, vendar kaznovane. V tem primeru časovne omejitve omehčamo. V ta namen najprej vpeljemo dodatni kriterij kasnosti (g_{ij}), ki je določen z enačbo (5).

$$g_{ij} = \begin{cases} a_j - d_i & a_j > d_i \\ 0 & \text{drugače,} \end{cases} \quad (5)$$

V primeru, da je trda časovna omejitev kršena, g_{ij} določa čas zamujanja koša B_j za rokom izdelave predmeta I_i. V primeru, da zamujanja ni, kriterij kasnosti g_{ij} zavzame vrednost 0, v primeru zamujanja pa 1. Z uporabo tega lahko optimizacijski kriterij razširimo tako, da upoštevamo tudi zamudo (ang. *tardiness*).

$$z_2: min \quad w_1 \cdot \sum_{j=1}^{n}(c_j \cdot y_j - \sum_{i=1}^{m} s_i \cdot x_{ij}) + w_2 \cdot \sum_{i=1}^{m} g_{ij} \cdot x_{ij} \tag{6}$$

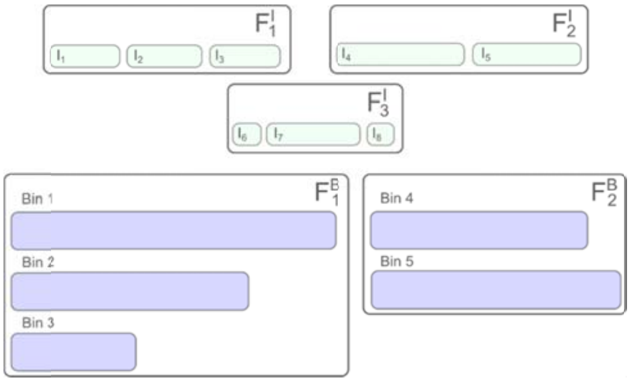
Enačba (6) predstavlja več-kriterijski problem, s katero optimiziramo tako preostanek kot tudi zamude. Kateri od teh kriterijev je za določen problem bolj pomemben, določimo z utežnima faktorjema w_1 in w_2.

2.4 Združevanje košev/predmetov

V nadaljevanju bomo predstavili problem, ki predpisuje, da lahko določeno skupino predmetov dodelimo le eni od skupin košev. To dosežemo z dodatnimi omejitvami v našem optimizacijskem problemu.

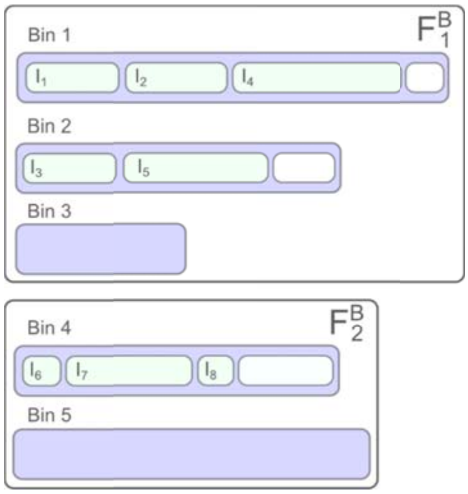
Predpostavimo, da imamo množico F^I, ki jo sestavlja $|F^I|$ skupin predmetov in množico F^B, ki jo sestavlja $|F^B|$ skupin košev. Predmet tako predstavimo z elementi (s_i, d_i, F_k^I), kjer $F_k^I \in F^I$ določa, kateri skupini predmetov predmet I_i pripada. Koš B_j pa je definiran z (c_j, a_j, F_l^B), kjer $F_l^B \in F^B$ označuje skupino košev, h kateri pripada koš B_j. *K*-ta skupina predmetov je sestavljena iz $|F_k^I|$ predmetov in *l*-ta skupina košev je sestavljena iz $|F_l^B|$ košev.

Primer, ko je osem predmetov združenih v tri skupine predmetov in pet košev v dve skupini košev, je prikazan na sliki 2. Predmeti iz ene skupine predmetov morajo biti dodeljeni košem, ki pripadajo enaki skupini košev.



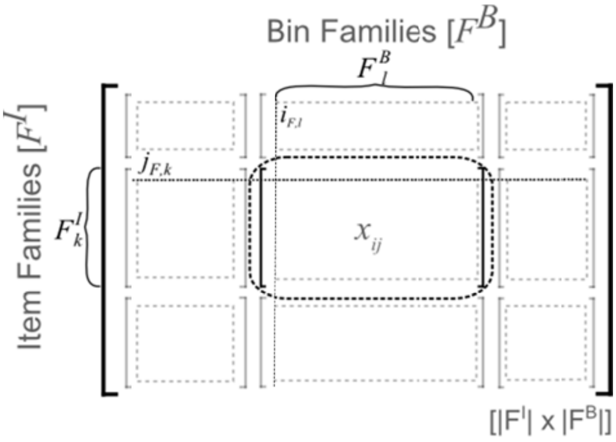
Slika 2: Skupine predmetov in skupine košev.

Rešitev tega problema bi lahko na primer bila rešitev predstavljena na sliki 3. Tu so predmeti iz F_1^I in F_2^I dodeljeni košem iz skupine košev F_1^B in predmeti iz F_3^I h košu iz skupine F_2^B. Beli pravokotniki ilustrirajo preostanke.



Slika 3: Izvedljiva rešitev problema iz slike 2.

Odločitveno spremenljivko x_{ij} lahko preuredimo tako, da so predmeti in koši iz istih skupin v matriki postavljeni drug ob drugem, kot je to predstavljeno na sliki 4. Podmatrike, označene na sliki, označujejo vse možne kombinacije skupin predmetov in skupin košev.



Slika 4: Odločitvena spremenljivka x_{ij} – skupini predmetov in košev

Predmeti iz *k*-te skupine so lahko dodeljeni košem iz *l*-te skupine le v primeru, ko je vsota vseh x_{ij} enaka velikosti skupine predmetov F_k^I. V tem primeru *i* označuje vse predmete iz *k*-te skupine predmetov in *j* vse koše iz *l*-te skupine košev. Formalno lahko to predstavimo z dodatnimi enakostnimi omejitvami (7), ki jih vključimo v naš optimizacijski problem.

$$\sum_{i=i_{F,k}}^{i_{F,k}+|F_k^I|} \sum_{j=j_{F,l}}^{j_{F,l}+|F_l^B|} x_{ij} = |F_k^I| \cdot \zeta_{l,k} \qquad (7)$$

Tu je $k = 1 \dots |F^I|$ in $l = 1 \dots |F^B|$. $i_{F,k}$ določa zaporedno število prvega predmeta v *k*-ti skupini predmetov in je izračunana kot:

$$i_{F,k} = \begin{cases} 1 & k = 1 \\ 1 + \sum_{p=1}^{F_k^I - 1}(|F_p^I|) & k = 2, \dots |F^I|. \end{cases} \qquad (8)$$

$j_{F,l}$ določa zaporedno število prvega koša v *l*-ti skupni košev in je izračunano kot:

$$i_{F,l} = \begin{cases} 1 & l = 1 \\ 1 + \sum_{q=1}^{F_l^B - 1}(|F_q^B|) & l = 2, \dots |F^B|. \end{cases} \qquad (9)$$

Indikator $\zeta_{l,k}$ podaja informacijo o tem, ali je ta kombinacija skupine predmetov in košev uporabljena, t.j. zavzame vrednosti 0 ali 1. Določen je kot vsota vrednosti v podmatriki x_{ij} (10), kjer podmatrika vsebuje vse koše iz *l*-te skupine košev in le prvega izmed predmetov v *k*-ti skupini predmetov.

$$\zeta_{l,k} = \sum_{j=j_{F,l}}^{j_{F,l}+|F_l^B|} x_{i_{F,k}j} \qquad (10)$$

Upoštevanje skupin predmetov in skupin košev pravzaprav zmanjšujejo kompleksnost problema, saj enakostne omejitve reducirajo prostor rešitev.

3 MRP planiranje z uporabo BPP formulacije

Predstavljeno formulacijo celoštevilčnega linearnega programiranja lahko uporabimo za namene planiranja materialnih virov. Ukvarjamo se s problemom, kjer imamo na voljo več paketov surovin, ki jih moramo razdeliti na več manjših kosov, da zadostimo potrebe po vhodnih materialih za izvedbo delovnih naročil. Vhodni materiali morajo biti na voljo v zahtevanih količinah pred zahtevanim rokom, hkrati pa morajo biti vhodni materiali za en končni izdelek pridobljeni iz iste šarže paketov surovin.

Podatki o surovinah so določeni z nabavnimi nalogi (PO), ki jih v našem primeru formulacije problema predstavimo s skupino košev. c_j določa velikost paketa surovin in a_j čas, ko bo posamezna surovina na voljo. Pri čemer a_j ni nujno enak za vse surovine posameznega PO. Vsak PO ima oznako, ki označuje kateri skupini surovin pripada, F_l^B.

Delovni nalogi določajo, koliko katere surovine (polizdelkov) je potrebnih za sestavo končnega izdelka. Vsak predmet je v predlagani formulaciji BPP predstavljen s predmetom, kjer je s s_i določena velikost polizdelka in z d_i rok, ko mora biti polizdelek na voljo. Ker se pogostokrat lahko zgodi, da surovine v nekaterih parametrih odstopajo glede na šaržo, moramo tudi v tem primeru vpeljati dodatne zahteve. Zagotoviti moramo, da je končni izdelek sestavljen iz surovin z istimi lastnostmi. V ta namen so vsi polizdelki (predmeti), potrebni za izdelavo enega izdelka, označeni z F_k^I.

Operater, ki mora uravnavati naročila in izdelavo delovnih nalogov, lahko tako pri svojem delu uporabi optimizacijske algoritme. Odvisno od problema ima na voljo različne optimizacijske funkcije. V primeru, da časovnih zamud ni, lahko uporabi optimizacijski kriterij, določen z enačbo (4), kjer se optimizira le preostanek. V primeru, ko prihaja do zamud pri naročilih, pa je potrebno uporabiti kriterij z mehkejšimi omejitvami, t.j. optimizacijski problem, določen z enačbo (6). V tem primeru se optimizira tako preostanek kot tudi zamude in operater mora določiti tudi uteži večkriterijske funkcije.

3.1 Primer

V tem podpoglavju si bomo pogledali rešitev problema, ki smo ga ilustrativno predstavili že v poglavju 2.4. V kontekstu MRP problema, imamo problem, kjer moramo izdelati tri izdelke iz petih dobavljenih surovin. Surovine so naročene z dvema nabavnima nalogoma. V razširjeni formulaciji BPP to pomeni, da imamo na voljo dve skupini košev:

- $F_1^B = \{B_1, B_2, B_3\}$
- $F_1^B = \{B_4, B_5\}$.

Posamezni izdelki so sestavljeni iz več polizdelkov. V razširjeni formulaciji BPP so izdelki predstavljeni s skupino predmetov, posamezni polizdelki pa kot predmeti:

- $F_1^I = \{I_1, I_2, I_3\}$
- $F_2^I = \{I_4, I_5\}$
- $F_3^I = \{I_6, I_7, I_8\}$

Podrobnosti o posameznih nabavnih (koših) in delovnih nalogih (predmetih) so povzeti v tabelah 1 in 2.

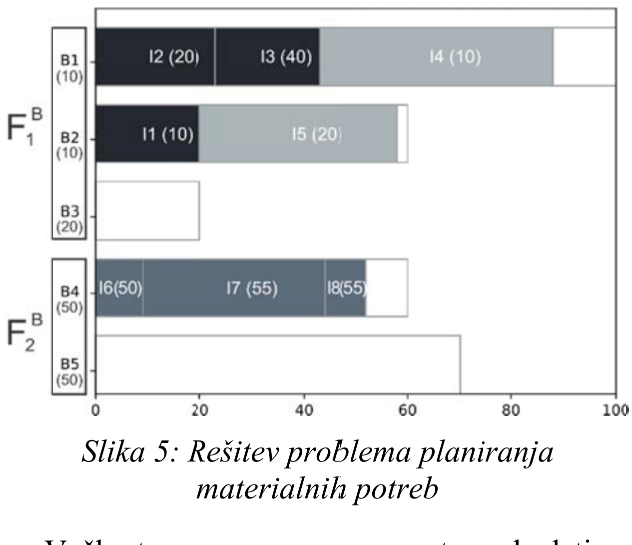
Tabela 1: Nabavni nalogi – Koši

Koš	c_j	a_j	F_l^B
B_1	100	10	1
B_2	60	10	1
B_3	20	20	1
B_4	60	50	2
B_5	70	50	2

Tabela 2: Delovni nalogi – Predmeti

Predmet	s_i	d_i	F_k^I
I_1	20	10	1
I_2	23	20	1
I_3	20	40	1
I_4	45	10	2
I_5	38	20	2
I_6	9	50	3
I_7	35	55	3
I_8	8	55	3

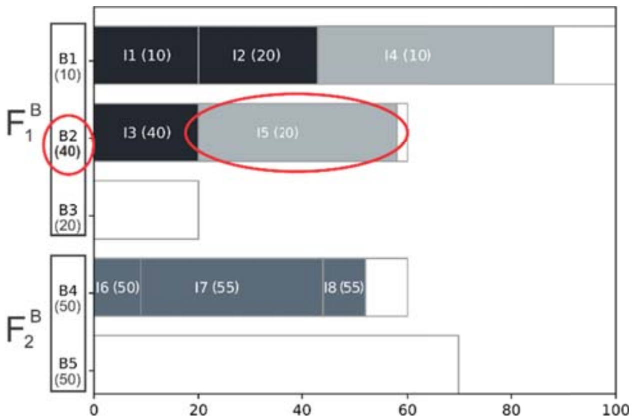
Predstavljen problem je mogoče rešiti brez čakanja, saj obstaja rešitev, ki zagotavlja surovine za vse izdelke pravočasno. Za iskanje optimalne rešitve lahko tako uporabimo optimizacijsko funkcijo s katero minimiziramo le preostanek, t.j. (3). Rešitev je grafično predstavljena na sliki 5, kjer so predmeti (sestavni deli) nanizani znotraj posameznega koša (surovine). Velikost koša oz. predmeta določa količino surovine oz. sestavnega dela. Poleg oznake za koš/sestavni del je v oklepajih podan tudi čas dobave (a_j) oz. rok za pričetek izvedbe (d_i). Skupine košev so na sliki označene z okvirom, predmeti iz ene skupine pa s skupno barvo. Kot vidimo so predmeti iz skupin F_1^I in F_2^I dodeljeni virom iz skupine F_1^B, medtem ko so predmeti iz F_3^I dodeljeni košu B_4 iz skuine F_2^B. Beli pravokotniki predstavljajo preostanek. Tako je od uporabljenih košev B_1, B_2, in B_4 preostalo še 12, 2 in 8 enot neuporabljene surovine. Skupni preostanek je torej $z_2=22$. Koša B_3 in B_5 nista uporabljena v rezultirajočem planu.



Slika 5: Rešitev problema planiranja materialnih potreb

Večkrat pa se mora operater ubadati s problemom, ko katera od pošiljk s surovino ne more biti na voljo pravočasno. Tako situacijo si lahko ilustriramo, ko opazujemo naš primer, pri čemer surovina predstavljena s košem B_2

zamuja za 30 časovnih enot ($a_j = 40$). Proizvodnja in vse ostale okoliščine ostajajo nespremenjene. Ker v tem primeru izvedljiva rešitev z uporabo optimizacijske funkcije (3) ne obstaja, smo prisiljeni uporabiti optimizacijski kriterij, ki dovoljuje časovne zamude, t.j. optimizacijski kriterij predstavljen s (6). Za predstavljen problem smo upoštevali tako kriterij, ki minimizira preostanke kot tudi zamude enakovredno, t.j. uporabili smo $w_1=1$ in $w_2=1$. Rezultirajoči plan je predstavljen na sliki 6. Tudi v tem primeru so uporabljene tri surovine (koši), kjer od vsake pridelamo nekaj preostankov ($B_1 = 12$, $B_2 = 2$ in $B_4 = 8$). Kot vidimo, je na koš B_2 dodeljen tudi predmet I_5, ki bo na voljo z zamudo 20 časovnih enot (glej rdeči označbi na sliki 6). Z optimizacijo dosežemo vrednost kriterijske funkcije $z_3 = 42$. Rezultirajoči plan producira enake preostanke, kot v prejšnjem primeru. Da je skupna zamuda krajša, sta polizdelka I_1 in I_3 v planu dodeljena nasprotnima košema.



Slika 6: Rešitev problema planiranja materialnih potreb z zamudami

Predstavljeni problem ilustrira pristop delovanja na majhnem problemu, ki ga je mogoče nazorno prikazati. Seveda pa je mogoče predstavljen pristop skalirati na nivo realnega problema v podjetju, kjer se srečujejo s problemom reda 50 nabavnih in 20 delovnih nalogov sestavljenih iz okoli 200 polizdelkov.

4 Zaključki

V prispevku smo razvili razširjeno obliko problema pakiranja, ki smo jo uporabili v primeru določanja optimalnih potreb po materialu. Predlagana rešitev razbremeni proizvodnega planerja, ki mu na ta način ni potrebno skrbeti za vse podrobnosti v zvezi z roki za dobavo in za ustrezno kombiniranje vhodnih materialov za kakovosten končni izdelek. Z uporabo predstavljene rešitve lahko proizvodni planer pride hitreje do izvedljivega plana, ki sistematično kompenzira neželene variacije v kvaliteti surovin. Ker je tak plan bolj ustrezen, je manj tudi potreb za njegovo korekcijo.

Problem je bil predstavljen na manjšem praktičnem primeru, kjer smo prikazali pristop k minimizaciji tako preostanka, kot tudi zamude, če pride do zamud v dobavi surovin.

Zahvala

Delo je bilo izvedeno v sklopu programa GOSTOP, ki ga delno financirata Republika Slovenija – Ministrstvo za izobraževanje, znanost in šport ter Evropska Unija – Evropski sklad za regionalni razvoj in v sklopu nacionalnega raziskovalnega programa Sistemi in vodenje, P2-0001.

5 Literatura

[1] de Carvalho, J. M. V., 2002. Lp models for bin packing and cutting stock problems. *European Journal of Operational Research*. 141, 253–273.

[2] Delorme, M., Iori, M., Martello, S., 2016. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*. 255 (1), 1 – 20.

[3] Epstein, L., Levy, M., 2010. Dynamic multi-dimensional bin packing. *Journal of Discrete Algorithms*. 8 (4), 356 – 372.

[4] Friesen, D. K., Langston, M. A., Feb. 1986. Variable sized bin packing. SIAM J. Comput. 15 (1), 222–230.

[5] P.C. Gilmore, R.E. Gomory. 1961. A linear programming approach to the cutting-stock problem. *Operations Research*, 9 , 849–859.

[6] Gurobi, 2018. *Gurobi optimizer*. URL http://www.gurobi.com/

[7] Jansen, K., Kraft, S., Aug 2016. An improved approximation scheme for variable-sized bin packing. *Theory of Computing Systems*. 59 (2), 262–322.

[8] Lodi, A., Martello, S., Monaci, M., 2002. Two-dimensional packing problems: A survey. *European Journal of Operational Research*. 141 (2), 241 – 252.

[9] Martello, S., Toth, P., 1990. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA.

[10] Mitchell, S., Kean, A., Mason, A., O'Sullivan, M., Phillips, A., 2018. *Optimization with PuLP*.

[11] Onwubolu, G., 2002. *Emerging Optimization Techniques in Production Planning and Control*. Imperial College Press.

[12] H. Reinertsen, T.W. Vossen. 2010. Discrete optimization. *European Journal of Operational Research*. 201 (3), 701–711.