ISSN 1648-8776
JAUNŲJŲ MOKSLININKŲ DARBAI. Nr. 50 (1). 2020 DOI: https://doi.org/10.21277/jmd.v50i1.280

NUOSEKLIOS GAMYBOS PLANAVIMO TVARKARAŠČIO SUDARYMAS

Donatas Kavaliauskas[1], Leonidas Sakalauskas[2]

[1] *Vilniaus universitetas, Lietuva*

[2] *Šiaulių universitetas, Lietuva*

El. p.: donatas.worshipper@gmail.com, Leonidas.Sakalauskas@mii.vu.lt

Įvadas

Daugelis diskrečių problemų yra NP tipo sunkiai sprendžiami uždaviniai, kuriuos galima išspręsti tik pasinaudojus eksponentinio ar dar aukštesnio sudėtingumo algoritmais. Tokiems uždaviniams spręsti yra taikomi euristiniai ir metaeuristiniai metodai, kurie dažnai yra kuriami analogijomis, aptinkamomis gamtoje ar biologinėse sistemose, ir leidžia spręsti efektyviai atskiras globalios optimizacijos problemas. Sėkminga metaeuristika turėtų užtikrinti pusiausvyrą tarp paieškos erdvės tyrinėjimo ir išnaudojimo.Gamybos tvarkaraščių sudarymas priklauso prie aktualių NP srities problemų.

Gamybos projektavimo ir vadybos uždaviniuose dažnai susiduriama su įvairiomis planavimo problemomis: išteklių, žaliavų, personalo. Šių uždavinių pradiniai duomenys susideda iš informacijos apie aibę tam tikrų operacijų, susijusių nuoseklumo sąryšiais ir ištekliais, kurie reikalingi užduotims atlikti. Dažnai praktikoje pasitaiko šie pradiniai duomenys:

- Užsakymų sąrašas su nustatytais įvykdymo terminais ir eiliškumu.
- Staklių (mašinų / įrenginių) sąrašas.
- Kiekvienai operacijai priskirtos alternatyvios staklės, kuriomis galima atlikti operaciją (staklės kaip operacijos charakteristika).
- Operacijos atlikimo laikas su tam tikromis staklėmis, kuris apskaičiuojamas įvertinant kiekvienų staklių darbo greitį (charakteristiką) vykdant konkrečią operaciją. Jei staklės gali atlikti kelias operacijas, tai joms priskiriamos kelios greičio charakteristikos.
- Kartu dirbančių staklių skaičius, kuris priklauso nuo darbininkų skaičiaus ir žmonių charakteristikos.
- Kiekvienų staklių darbo laikas, t. y. darbo pradžios ir darbo pabaigos laikas, pertraukų laikas, išdirbtų savaitėsdienų kiekis, iš kurių vėliau skaičiuojama darbo laiko trukmė.
- Staklės, kuriomis gali dirbti darbininkas, ir darbininko charakteristika, kurios gali būti kelios.
- Darbininkų sąrašas.

Paprastai užduoties sąvoka apibrėžiama, kai per tam tikrą laiką priskirti procesoriai (vykdytojai, mašinos) atlieka tam skirtus darbus, kurių atlikimas sunaudojareikalingus išteklius. Ištekliai ir laikas, kuris reikalingas projektui įgyvendinti, dažniausiai yra riboti. Siekiant efektyvumo reikia rasti tokį užduočių atlikimo planą, kuris tenkintų nuoseklumo sąryšius, procesorių pajėgumą bei išteklių ribojimus, siekiant minimizuoti tikslo funkciją. Didžioji dalis tokių planų sudarymo uždavinių priklauso NP sudėtingumo uždavinių klasei, kuriems spręsti dažnai taikomi euristiniai metodai [1, 2, 6].

Optimizuojant tvarkaraščius dažniausiai tenka rinktis vieną ar keletą optimizavimo kriterijų. Dažniausiai taikomi yra tokie kriterijai:

- minimalus viso projekto užbaigimo laikas (angl. *Makespan*),
- minimalus viso projekto vėlavimas nuo nustatyto užbaigimo laiko,
- kompleksiniai kriterijai,
- minimalios sąnaudos, išlaidos,
- maksimalus pelnas,
- optimalus vykdytojų (procesorių) panaudojimas.

Daugelyje planavimo, statybos ir kitų uždavinių vienu iš svarbiausių optimizavimo kriterijų yra viso projekto užbaigimo laikas, kuris paprastai yra minimizuojamas. Pramonėje pasitaiko kitas optimizavimo kriterijus – maksimalus vėlavimas, kuris taip pat minimizuojamas. Iš kelių kriterijų galima sukonstruoti ir kompleksinius svorinius optimizavimo kriterijus. Pavyzdžiui, visas svorinis baigimo laikas (angl. *total weighted completion time*, Σ *wjCj*), bendras svorinis vėlyvumas (angl. *total weighted tardiness*, Σ *wjTj*), svorinis skaičius pavėluotų užduočių (Σ *wjUj*). Priklausomai nuo uždavinių specifikos, taip pat gali būti minimizuojamos sąnaudos (išlaidos). Straipsnio tyrimo objektas – gamybos tvarkaraščių optimalaus planavimo realiu laiku sudarymas. Nagrinėjama šakų ir ribų algoritmo tvarkaraščiams su ribotais ištekliais ypatybės ir būdai pritaikyti algoritmą universaliai šiai

problemai spręsti. Darbo tikslas yra sudaryti nuoseklios gamybos ir gamybos partijomis planavimo algoritmą.

Tvarkaraščių su ribotais ištekliais uždavinio formuluotė

Nuoseklios gamybos planavimo problemą (angl. *flow shop*) sudaro baigtinis darbų rinkinys J = {1, 2, ... n} ir baigtinis mašinų rinkinys M = {1, 2, ... m}. Tiksliau, kiekvieną darbą sudaro baigtinis operacijų rinkinys. Operacija turi būti įvykdomasu iš anksto paskirta mašina, t. y. i-oji užduoties j operacija, žymima $o_{i,j}$, naudojama mašina $\mu_{i,j}$ ε M. Kiekvienos užduoties operacijų eiliškumas yra fiksuotas, todėl reikia atsižvelgti į kiekvienos užduoties technologinę seką. Tikslas yra sudaryti optimalų šių n darbų tvarkaraštį m mašinoms. Kitos sąlygos yra šios:

- Apdorojant i-ąją darbo j operaciją per pi, j > 0 laiko vienetui.
- Kiekviena operacija turi būti atlikta tiksliai vieną kartą.
- Atleidimas neleidžiamas operacijų metu.
- Vykdomos operacijos negali sutapti.
- Nėra priklausomybių nuo mašinų ar jų sekų.

Pagal anksčiau paminėtas priklausomybes matematinė formuluotė uždavinio gali atrodyti taip:

$$min \sum_{j=1}^{n} w_j * t_j, \quad (1)$$

Kai yra ribojimai:

$$s_{ij} + p_{ij} \leq s_{i+1,j} \quad \forall j \, \epsilon J, \forall i \, \epsilon M, \, (2)$$

$$s_{ij} + p_{ij} \leq s_{k,l} V \, s_{kl} + p_{kl} \leq s_{i,j} \, \forall j, l \epsilon J, \forall i, k \, \in M \; kur \; j \neq l \; ir \; \mu_{ij} = \mu_{kl}, \, (3)$$

$$t_j \geq s_{mj} + p_{mj} - d_j \, \forall j \epsilon J, \, (4)$$

$$t_j \geq 0 \, \forall j \epsilon J, \, (5)$$

$$s_{1j} \geq r_j \, \forall j \epsilon J, \, (6).$$

Tikslo funkcija (1) nusako viso užduočių komplekso užbaigimo laiką.Atliekant tikslo funkcijos skaičiavimą svertinė atidumo suma turi būti kuo mažesnė, kur tj = max {0, cj-dj} yra svertinis rezultatas darbo j tvarkaraštyje ir c_j yra atlikimo laikas, o w_j – darbo svoris j darbui. Pirmasis apribojimas (2) užtikrina kiekvieno darbo technologinę seką. Čia sprendimo kintamasis s_{ij} nurodo operacijos pradžios laiką, p_{ij} nurodo i-tojo darbo j-tosios operacijos trukmę. Disjunkcinis,kuris yra antrasis apribojimas (3), užfiksuoja sekos nustatymo problemą kiekviename įrenginyje. Apribojimai (4) ir (5) yra programos dalis, kuria siekiama išmatuoti dėl to atsirandantį kiekvieno darbo lėtumą. Čia d_j yra darbo terminas j darbui. Galiausiai, paskutinis apribojimas (6) užtikrina, kad darbas negali būti pradėtas anksčiau nei išleidžiamas, ir tokiu būdu užfiksuojamas sprendimo kintamųjų s_{ij} neigiamumas. Čia r_j yra darbo išleidimo laikas, o s_{1j} – darbo pradžia. Atliekant tikslo funkcijos skaičiavimą, svertinė atidumo suma turi būti kuo mažesnė, kur t_j = max {0, c_j-d_j} yra svertinis rezultatas darbo j tvarkaraštyje ir cj yra atlikimo laikas [1, 2, 6].

Šis uždavinys yra NP sudėtingumo stipriąja prasme, kadangi vienmatis pakavimo į konteinerius uždavinys yra šio uždavinio atskiras atvejis [3]. Juolab, bet kuriam ε > 0, mažai tikėtina rasti apytikslius polinominius algoritmus su garantuotu tikslumo įverčiu $n^{1-\varepsilon}$. Taigi, todėl euristikų kūrimas yra perspektyviausias kelias tokių uždavinių sprendimui. Viena iš dažnai praktiškai naudojamų gamybos planavimo euristikų yra šakų ir ribų metodas.

Šakų ir ribų algoritmas

Šakų ir ribų algoritmas (angl. *Branch & Bound*, sutr. BaB) yra tikslusis algoritmas, kuris skirtas įvairiems optimizavimo uždaviniams spręsti. Pirmą kartą šis metodas pasiūlytas 1960 metais [4]. Šio algoritmo idėja yra „skaldyk ir valdyk“. Taikant šį metodą optimizavimo uždavinio sprendinių aibė yra skaldoma į poaibius, apskaičiuojant viršutinę ir apatinę, kiekvienam poaibiui priklausančių tikslo funkcijos sprendinių rėžius. Sprendžiant minimizavimo uždavinius, tie poaibiai, kurių apatiniai rėžiai yra didesni už palie-

kamo nagrinėti poaibio viršutinius rėžius, iš tolesnio sprendinių perrinkimo pašalinami. Šis procesas tęsiamas tol, kol randamas optimalus sprendinys.

Paieškos medis pradedamas konstruoti nuo tėvynės šakos, kurioje visi užsakymai yra nulinės vertės. Tada yra planuojama vaikų poaibiai. Kiekvienose vaikų šakose yra pridedama po vieną užsakymą, kuris yra suplanuojamas, o likusiuose užsakymų laikuose paimamas teorinis laikas. Šakojimo operacija skaido aibę X į vis mažesnius poaibius tol, kol gauname tokį uždavinį, kurį galime išspręsti. Paieškos medžio viršūnės reprezentuoja poaibius (dalinius sprendinius), o briaunos – ryšį tarp tėvo-poaibio ir jo vaikų-poaibių, gautų šakojant viršūnę [5].

Šakų ir ribų algoritmas leidžia rasti d dydžio optimalų požymių rinkinį nevykdant pilnos paieškos, tačiau požymių atrinkimo kriterijus turi tenkinti monotoniškumo sąlygą, kuri teigia, jog turint du XD aibės poaibius A ir B tokius, kad A ε B, turi būti tenkinama kita sąlyga: A εB => J(A) <=J(B). Kitaip tariant, požymių poaibiui pridėjus požymį jis niekada neduos prastesnio vertinančios funkcijos J rezultato.

Viena pagrindinių operacijų yra šakų genėjimas. Kiekvienai viršūnei, gautai šakojimo metu, skaičiuojamas apatinis rėžis. Apatinis rėžis yra apskaičiuojamas pagal formulę:

$$LB = (\sum_{i=1}^{n} T_i) + (\sum_{i=k+1}^{n} T_i), \quad (7)$$

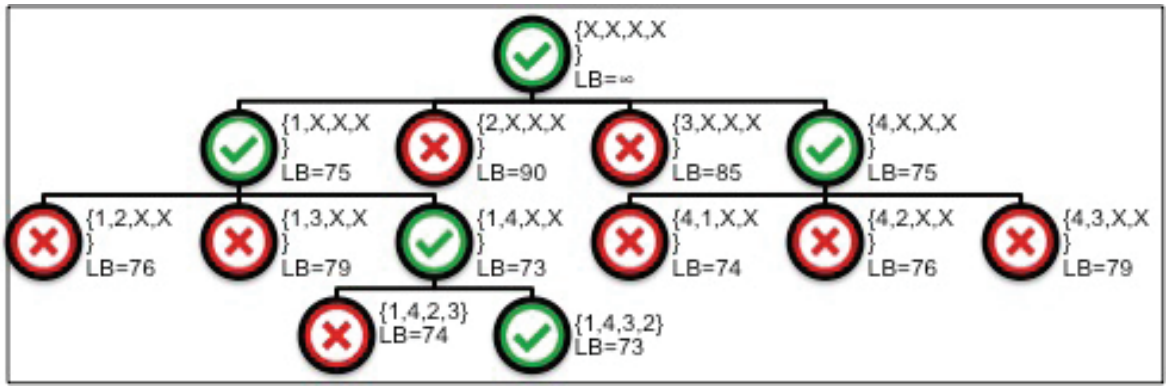
kur k yra suplanuotų užduočių skaičius ir k priklauso n. N yra visų užduočių skaičius, o T yra laikas. Tokiu būdu sumažinama paieška, kadangi tik tos viršūnės, kurių apatinis rėžis patenka į tam tikrą paieškos intervalą, yra tiriamos toliau, o kitos yra pašalinamos [8]. Paieškos baigimo sąlygos nurodo, kad uždavinys yra išspręstas ir rastas optimalus sprendinys. Tai atsitinka, kai visos viršūnės yra išnagrinėtos arba atmestos.Planavimo uždaviniuose paieškos pabaiga yra tada, kai visos užduotys yra sudedamos į seką [7].

Algoritmo veikimas paaiškinamas sprendžiant pavyzdį, kurio duomenys pateikti 1 lentelėje, sprendimo schema pateikta 1 pav. Tarkime, reikia atlikti 4 užsakymus, kurių operacijos turi būti sudėliotos tam tikra tvarka, siekiant juos atlikti per trumpiausią laiką (žr. 1 lentelę). Šis uždavinys yra paremtas iš dalies nuoseklios tvarkaraščio sudarymo problemos (angl. *job shop*) sprendimu, kai kiekviena mašina gali atlikti visas operacijas, tačiau mašinos turi skirtingą atlikimo laiką.

1 lentelė. ***Testinio uždavinio trukmių lentelė***

	1	2	3	4
A	75 min.	90 min.	85 min.	75 min.
B	74 min.	76 min.	79 min.	73 min.
C	74 min.	74 min.	73 min.	74 min.
D	74 min.	73 min.	74 min.	74 min.

Pagal pateiktus duomenis pradėtas formuoti sprendimas, kuris pateikiamas 1 paveikslėlyje.



1 pav. Šakų ir ribų algoritmo pavyzdys

Inicijavimas: tvarkaraščio sugeneravimas be užduočių eiliškumo ir tai nurodome(*,*,*,*). Čia „*“ darbo sekoje rodo, kad nė vienas darbas nebuvo priskirtas į poziciją. 1 žingsnis: norėdami sukonstruoti tvarkaraštį, pradedant nuo pirmos pozicijos, perkeliame iš tėvinės šakos (*,*,*,*) į vieną iš keturių galimų vaikų (1,*,*,*), (2,*,*,*), (3,*,*,*), (4,*,*,*). 2 žingsnis: apskaičiuojame šakų apatinius rėžius. Priskiriame naujas užduotis į tam tikras šakas, kur yra mažiausias apatinis rėžis. Šaka (1,*,*,*) suteikia tris galimybes (1,2,*,*), (1,3,*,*) ir (1,4,*,*) ir t. t. 3 žingsnis: kartojamas 2 žingsnis, kol lieka dvi nepriskirtos užduotys. Likus dviem užduotims yra iškart sudaromos dvi paskutinės šakos. Šaka (1,4,*,*) davė geriausią apatinį rėžį, todėl nereikėjo atlikti papildomo žingsnio. Tai yra (1,4,2,3) ir (1,4,3,2). Šis pavyzdys atspindi [1] įrodytą algoritmo potencialą, tačiau neparodo galimų problemų sprendžiant realios gamybos uždavinius.

Šakų ir ribų algoritmo rezultatai

Pagal pateiktą algoritmą panagrinėkime nuoseklios gamybos planavimo uždavinį, kai procesai vykdomi nuosekliai. Procesai planuojami nuosekliai vienas paskui kitą pagal technologines sekas. Procesas yra gamybos partijomis pobūdžio, jei dėl proceso įrangos fizinio struktūrizavimo ar dėl kitų veiksnių susideda iš identiškų vieno ar kelių etapų sekų, kurios turi būti atliekamos nustatyta tvarka. Pasibaigus šiems etapams, pagaminamas baigtinis gaminių kiekis. Jei reikia sukurti daugiau gaminių, seka turi būti kartojama.

Yra keletas gamybos partijomis procesams būdingų charakteristikų:

- produktų gamyba partijomis;
- nepertraukiamas medžiagų srautas;
- gamybos eiga nustatoma pagal laiko / pabaigos tašką;
- gamyba vykdo operacijų etapus [10].

Tarkime,užsakymasyra sudarytas iš šių komplektų: Ko1, Ko1, Ko4, Ko4, Ko4. Kiekvieną komplektą sudaro 6 operacijų rinkiniai. Komplektų specifikacijos pateiktos 2 lentelėje.

Panagrinėkime užsakymą Ko1(2) Ko4(3). Nesunku apskaičiuoti, kad šiam užsakymui atlikti prireiks 234 operacijų.

2 lentelė. ***Uždavinio komplektų specifikacijos***

Komplektas	Komplekto sudėtis
Ko1	Ko2(2), Ko3(3), Op1(1), Op2(1)
Ko2	Op3(2), Op4(3)
Ko3	Op3(1), Op5(2)
Ko4	Ko5(2), Ko7(3), Op1(1), Op2(1)
Ko5	Ko2(3), Ko6(2), Op4(2), Op5(1)
Ko6	Op3(2), Op5(3)
Ko7	Op6(2)

Užsakymus sudarančių operacijų duomenys pateikti 3 lentelėje.

3 lentelė. ***Operacijų duomenys***

Operacija	Kiekis	Trukmė vienetui	Bendra trukmė
OP1	5	8 min.	40 min.
OP2	5	10 min.	50 min.
OP3	74	11 min.	814 min.
OP4	78	15 min.	1170 min.
OP5	54	5 min.	270 min.
OP6	18	7 min.	126 min.

Kadangi šiame uždavinyje vyrauja operacijos (OP4) darbai, apatinio rėžio formulė pritaikyta atsižvelgiant į tai. Iš 3 lentelės matyti, kad operacija OP4 trunka ilgiausiai, todėl galima teigti, kad ji sudaro „butelio kakliuką“. Panagrinėkime užsakymo komplektus. Kadangi užsakymą sudaro 2 Ko1 komplektai ir 3 Ko4 komplektai, reikia panagrinėti kiekvieną iš šių komplektų atskirai (žr. 4 ir 5 lenteles).

4 lentelė. ***Ko1 komplekto analizės duomenys***

Operacija	Kiekis	Trukmė vienetui	Bendra trukmė	Procentinė dalis
OP1	1	8 min.	8 min.	20%
OP2	1	10 min.	10 min.	20%
OP3	7	11 min.	77 min.	9,46%
OP4	6	15 min.	90 min.	7,75%
OP5	6	5 min.	30 min.	11,1%
OP6	0	7 min.	0 min.	0%

5 lentelė. ***Ko4 komplekto analizės duomenys***

Operacija	Kiekis	Trukmė vienetui	Bendra trukmė	Procentinė dalis
OP1	1	8 min.	8 min.	20%
OP2	1	10 min.	10 min.	20%
OP3	20	11 min.	220 min.	27,9%
OP4	22	15 min.	330 min.	28%
OP5	14	5 min.	70 min.	25,9%
OP6	6	7 min.	36 min.	33%

Taigi, atsižvelgiant į tą dalį, kurią sudaro kiekvienas komplektas atsirandant „butelio kakliukui“, galima taikyti tokią apatinio rėžio formulę:

$$LB = 100 - \frac{100 * \left(\underbrace{\sum_{i=1}^{n} T_i}\right)}{\underset{Nesuplanuotosužduotys}{Btsum}}, \quad (8)$$

čia Btsum yra „butelio kakliuko“ dar nesuplanuotas laikas. Po kiekvienos iteracijos šis laikas yra perskaičiuojamas. Žymeniu n yra nurodomas suplanuotų operacijų skaičius, o T yra jų trukmė. Algoritmas rinkdamasis tarp šakų renkasi mažiausią procentinę dalį turintį apatinį rėžį.

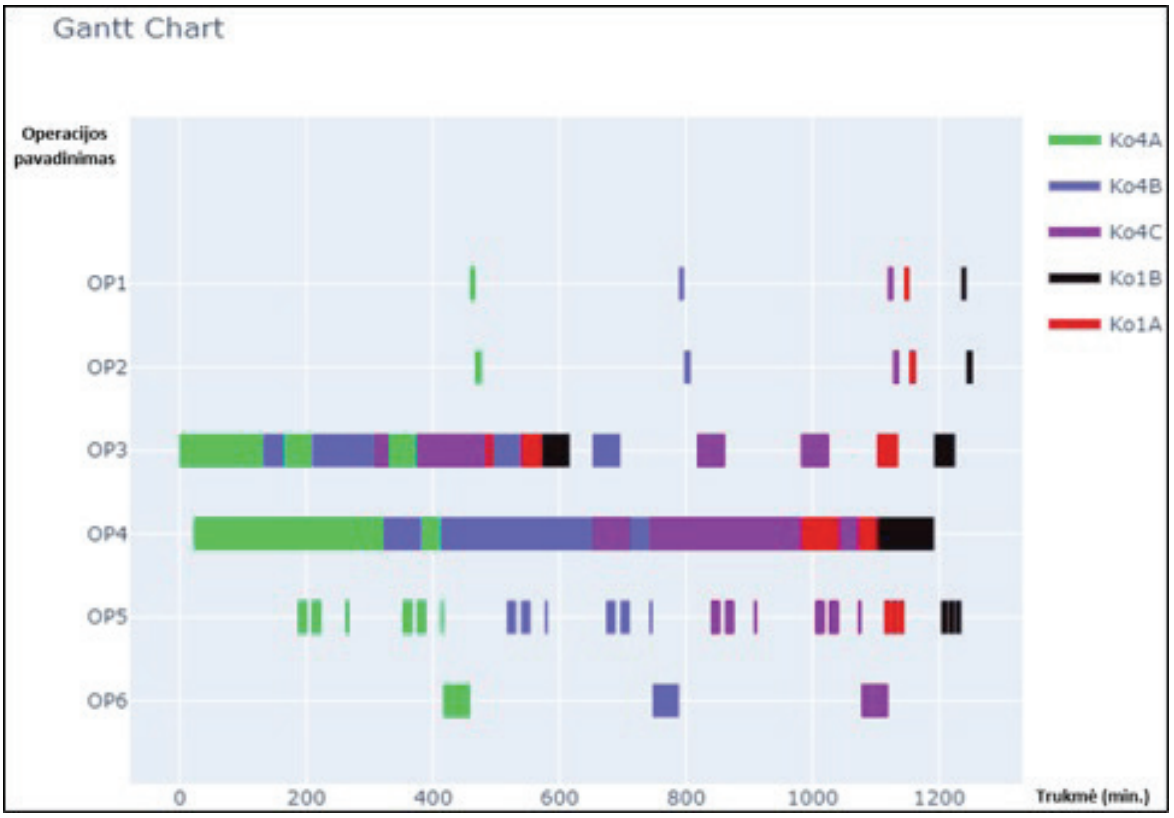
Remiantis L. Rajeckaitės darbu apatinį rėžį galima aprašyti formule:

$$LBC(S\ j(A)) = \max\{LBM\ (S\ j(A)),\ LBJ\ (S\ j(A))\}, \quad (9),$$

kur Sj(a) yra dalinis tvarkaraštis j cechui, LBM yra mašinos apatinis rėžis, o LBK – darbo apatinis rėžis. Tačiau naudojant šią formulę su 10 darbų, jau negalima taikyti šakų ir ribų algoritmo dėl per ilgos skaičiavimo trukmės. Dėl šios priežasties darbe buvo netirti rezultatai su (9) formule [8].

Taigi, pirmame žingsnyje, renkantis tarp K01 ir Ko4 komplektų, algoritmas pasirenka gaminti Ko4, nes jo apatinis rėžis bus 72 %, o Ko1 bus 92,3 %.

Kadangi nėra svarbu kurį iš Ko4 reikia pasirinkti pirmuoju, tai algoritmas pasirenka pirmą, einantį iš eilės. 3 paveiksle yra pateikiama Ko1(2) + Ko4(3) uždavinio sprendimo Ganto diagrama, o žemiau pats detalus sprendinys. Gautas tvarkaraštis yra 1253 minučių trukmės, kai kritinių kelių algoritmų suskaičiuotas tvarkaraštis buvo 1311 min. Šis tvarkaraštis atsilieka 28 min. nuo optimalaus tvarkaraščio, o jo Ganto diagrama yra pateikiama 2 paveikslėlyje.



2 pav. Užsakymo Ganto diagrama, gauta šakų ir ribų metodu

Pagal 2 paveikslėlyje pateiktą Ganto diagramą galima pasakyti, jog sudarytas tvarkaraštis pateikia optimalų tvarkaraštį. Tačiau operacijos OP5 būtų galima dar optimizuoti, tai yra pašalinti trumpas pertraukas tarp darbų. Tokiu atveju sudarytas tvarkaraštis būtų priimtinesnis realiai gamybai.

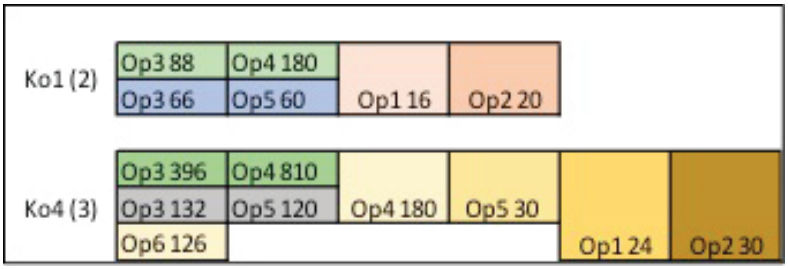
Pakoreguokime uždavinį į gamybą partijomis, kai procesai vykdomi nuosekliai. Procesai planuojami nuosekliai vienas paskui kitą pagal technologinius medžius.Yra keletas gamybos partijomis procesams būdingų charakteristikų:

- produktų gamyba partijomis;
- nepertraukiamas medžiagų srautas;
- gamybos eiga nustatoma pagal laiko / pabaigos tašką;
- gamyba vykdo operacijų etapus [10].

Tvarkaraščių su ribotais ištekliais uždavinio formuluotės skyriuje prie išvardytų ribojimų pridėkime papildomą ribojimą, kad atitiktų gamybos partijomis charakteristiką:

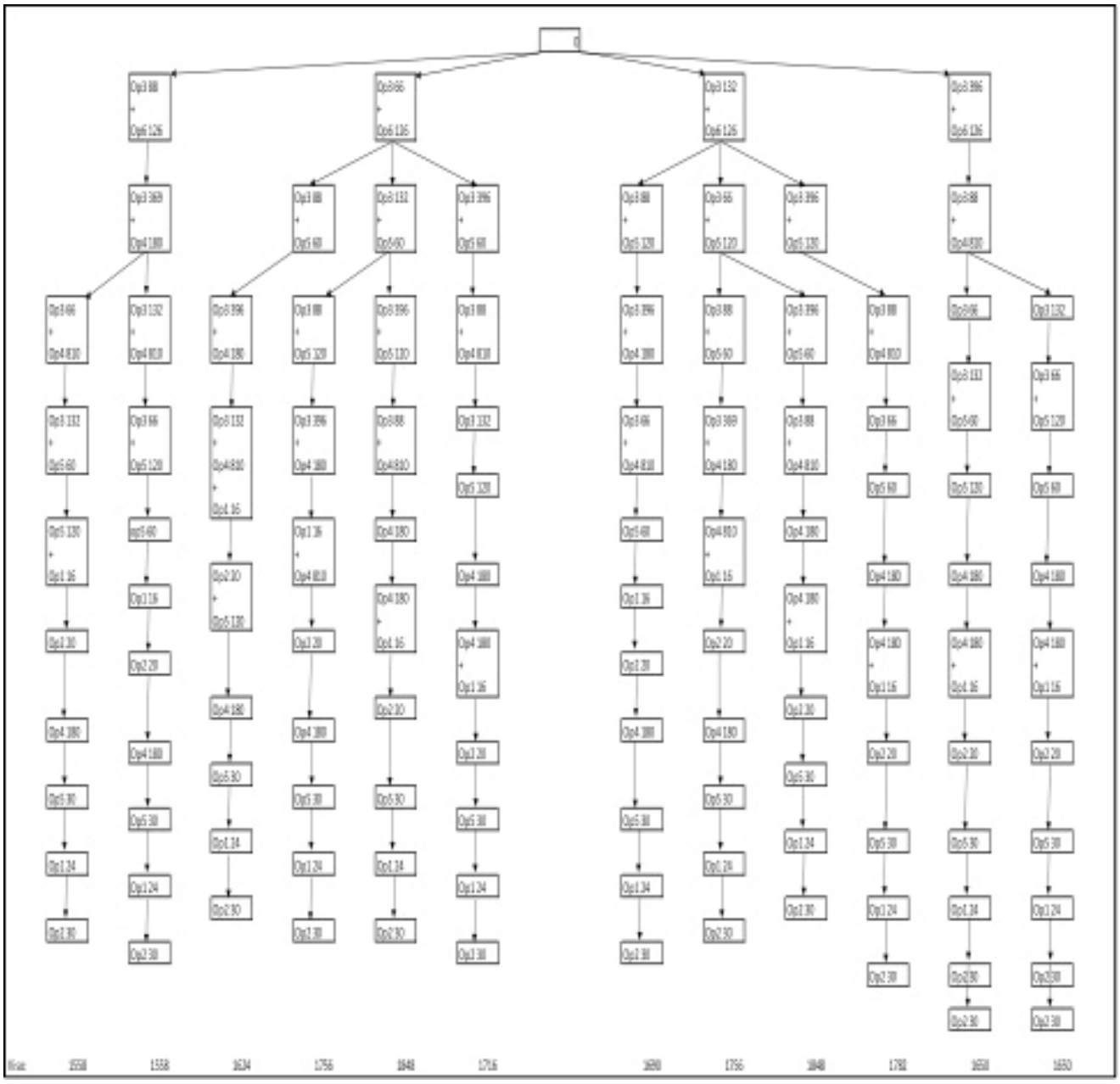
- Jei pradedame gaminti komplektą, tai operaciją užfiksuojame mašinoje, kad visi gaminiai, turintys šį komplektą, turi būti gaminami iš eilės.

Tokiu atveju mūsų anksčiau spręstas uždavinys su Ko4 (3) ir Ko1 (2) turės technologinius medžius, kurie yra pateikiami 3 paveikslėlyje.



3 pav. Užsakymų Ko4 (3) ir Ko1 (2) technologiniai medžiai

Komplektas Ko4 bus sudarytas iš 9 operacijų, kurių trukmės yra padidintos nuo pradinių sąlygos trukmių. Tai atsitinka su komplektu Ko1. Šiuo atveju vienodos operacijos yra sujungiamos į vientisą operaciją su padidinta trukme. 3 paveikslepastebime, kad turime 4 galimas pradžias, kurioms yra reikalinga Op3 ir vienas kelias, pradedant su Op6. Kadangi Op3 ir Op6 yra atliekamos skirtingų mašinų, tai galime pradėti vykdyti operacijas vienu metu. Sudarykime galimų variantų medį šiam uždaviniui, kuris yra pavaizduotas 4 paveikslėlyje.



4 pav. Galimi uždavinio sprendiniai

Pagal pateiktą visų galimų variantų medį pastebime, kad uždavinys turi du kelius, vedančius į optimumą. Anksčiau naudota apatinio rėžio formulė mums šiuo atveju yra netinkama, nes taikydami ją nepasieksime optimalios reikšmės. Reikia adaptuoti apatinio rėžio formulę taip, kad pirmame žingsnyje būtų atliekama ilgiausiai užimamo komplekto ar jų derinių operacija, ir kartu pradėti atlikti trumpiausią galimą darbą, jei yra galimybė, kad kiti darbai pasidengtų atliekant ilgiausią operaciją. Kadangi gamyboje tvarkaraščiai yra dinaminiai, todėl yra sunku pritaikyti šakų ir ribų algoritmą realioje gamyboje.

Autoriai M. I. Takano ir M. S. Nagano šakų ir ribų algoritmą adaptavo mišriam sveikojo skaičiaus tiesiniam programavimui (angl. *Mixed Integer Linear Programming*, sutr. MILP). Straipsnyje jie panaudojo keturias apatinio rėžio formuluotes, tačiau turint tik dvylika užduočių ir keturias mašinas visos apatinio rėžio formuluotės pradėjo daryti paklaidas [9].

Išvados ir rekomendacijos

1. Testo rezultatai parodė, kad šakų ir ribų algoritmas gali pateikti optimalų sprendinį, jeigu algoritmas yra adaptuotas uždavinio sąlygai. Pagrindinė adaptacija atsiranda pasirenkant apatinio rėžio formulę, kuri parenka kitą nagrinėjamą šaką.
2. Šiame darbe yra pateiktos rekomendacijos šakų ir ribų algoritmo derinimui dinaminiam nuoseklios gamybos planavimui. Esminė problematika atsiranda renkantis apatinio rėžio formulę uždaviniui. Nagrinėti šaltiniai rekomenduoja šiai problemai vis skirtingas apatinio rėžio formuluotes, todėl vienašališkos nuomonės, kuri yra geriausia, nėra.
3. Norint pritaikyti algoritmą veikti realioje gamyboje, reikėtų sukurti mašininio mokymosi metodologiją, kuri identifikuotų uždavinį ir pagal turimą informaciją parinktų geriausiai tinkančią apatinio rėžio formulę šakų ir ribų algoritmui. Tam reikėtų sukurti istorinių uždavinių duomenų bazę ir apatinio rėžio formulių biblioteką. Tokiu būdu aplikacija galėtų save išsimokyti parinkti tinkamą apatinio rėžio formulę uždaviniui.
4. Pateikti du tvarkaraščio sudarymo su ribotais ištekliaispavyzdžiai parodo, kad ir sprendžiama ta pati matematinė problema gali pareikalauti skirtingo sprendimo būdo, kai uždavinys yra pakoreguojamas tik vienu papildomu ribojimu. Abiejų uždavinių duomenys tie patys, tačiau tikslo funkcijos minimumas skirtingas.

Literatūra

1. Talapatra, S., Rahman, S. C., Das, C., Kumar Dey, U., 2014, *Application of Branch and Bound algorithm for solving flow shop scheduling problem comparing it with tabu search algorithm,* https://www.researchgate.net/profile/Subrata_Talapatra3/publication/324160684_Application_of_Branch_and_Bound_algorithm_for_solving_flow_shop_scheduling_problem_comparing_it_with_Tabu_search_algorithm/links/5ac2334ea6fdcccda65e9ee5/Application-of-Branch-and-Bound-algorithm-for-solving-flow-shop-scheduling-problem-comparing-it-with-Tabu-search-algorithm.pdf?origin=publication_detail
2. Grzechca, W., 2016, *Manufacturing in Flow Shop and Assembly Line Structure*, http://www.ijmmm.org/vol4/219-SP15315.pdf
3. Kocetov, J. A., Stoliar, A. A., 2003, Application of alternating environments to approximate solving of RCPSP. *Discrete analysis and operation research*. Ser. 2, Vol. 10, No. 2. P. 29–55 (in Russian).
4. Land, A. H., Doig, A. G.,1960, An Automatic Method of Solving Discrete Programming Problems*, Econometrica*. Vol. 28, No. 3. P. 497–520.
5. Ignall, E., Schrage, L., 1964, *Applicationof the branch and bound technique to some flow-shop scheduling problems*, http://www.rspq.org/pubs/sch.pdf
6. Kolisch, R., Hartmann, S., 1999, *Heuristic algorithms for solving the resource-constrained project scheduling problem: classification and computationalanalysis*, Published Springer Science + Business Media. New York, USA.
7. Defraeye, M., Van Nieuwenhuyse, I., 2013, *Abranch-and-bound algorithm for shift scheduling with nonstationary demand*, https://core.ac.uk/download/pdf/34585816.pdf
8. Rajeckaitė, L., 2009, G*amybinių tvarkaraščių sudarymo uždavinių ir algoritmų analizė.* Magistro darbas. Kauno technologijos universitetas, https://core.ac.uk/download/pdf/51707954.pdf
9. Takano, M. I., ir Nagano, M. S., 2017, A branch-and-bound method to minimize the makespan in a permutation flow shop with blocking and setup times. *Cogent Engineering.* Vol. 4, https://www.cogentoa.com/article/10.1080/23311916.2017.1389638
10. Torenli, A., 2009, *Assembly line design and optimization,* https://www.proplanner.com/media/cms/Artun-Torenli_Thesis_MAN_Public_41372EB3DF63E.pdf

Summary

PLANNING FLOW SHOP SCHEDULING

Donatas Kavaliauskas, Leonidas Sakalauskas

Various planning problems often arise in production design and management: planning of resources, raw materials, personnel. These problems fall into the complexity class of NP fullness. It is not enough to use greedy algorithms to get the optimal solution, heuristic algorithms must be used. One of heuristic algorithms is the branch and bound algorithm. This algorithm is a computer-simulated solution to the problem of flow shop. The paper presents several variants of the flow shop planning problem, for which the branch and bound algorithm has been adapted.

Keywords: *optimization, branch and bound algorithm, flow shop.*

Santrauka

NUOSEKLIOS GAMYBOS PLANAVIMO TVARKARAŠČIO SUDARYMAS

Donatas Kavaliauskas, Leonidas Sakalauskas

Gamybos projektavimo ir vadybos uždaviniuose dažnai susiduriama su įvairiomis užduočių planavimo problemomis: išteklių, žaliavų, personalo. Šios problemos yra priskiriamos NP pilnumo sudėtingumo klasei. Norint gauti optimalų sprendinį neužtenka pasinaudoti godžiaisiais algoritmais, tačiau reikia naudoti euristinės klasės algoritmus. Vienas iš euristinių algoritmų yra šakų ir ribų algoritmas. Naudojant šį algoritmą su kompiuteriu yra modeliuojamas nuoseklios gamybos planavimo uždavinio sprendimas. Taip pat straipsnyje yra pateikiami keli nuoseklios gamybos planavimo uždavinio variantai, kuriems yra adaptuojamas šakų ir ribų algoritmas.

Prasminiai žodžiai: *optimizacija, šakų ir ribų algoritmas, nuoseklios gamybos planavimas.*

Įteikta 2020 02 23
Priimta 2020 05 21