

# Gamybinių tvarkaraščių sudarymo uždavinių ir algoritmų analizė

Lina RAJECKAITĖ, Narimantas LISTOPADSKIS

Kauno technologijos universitetas, Fundamentaliųjų mokslų fakultetas  
Studentų g. 50, LT-51368 Kaunas  
el. paštas: lina.rajeckaite@ktu.stud.lt; narimantas.listopadskis@ktu.lt

**Santrauka.** Nuoseklios gamybos tvarkaraščių sudarymo uždavinys yra vienas iš kombinatorinės optimizacijos uždavinių. Tokio tipo problemos išskyla gamyboje, paskirstant išteklius, logistikoje ir pan. Duotos darbų ir mašinų (aptarnaujančių įrenginių) aibės. Kiekvienas darbas susideda iš vienodo skaičiaus operacijų. Mašinos dirba su pertraukomis. Uždavinio tikslas – minimizuoti visų darbų įvykdymo pabaigos laiko momentą. Darbe pateikiamas realizuotas vienas tikslusis algoritmas – šakų ir ribų, bei du meta-euristiniai algoritmai: modeliuojamasis atkaitinimas ir paieška su draudimais. Atlikta šių trijų algoritmų analizė.

**Raktiniai žodžiai:** nuoseklios gamybos tvarkaraščių sudarymo uždavinys, šakų ir ribų algoritmas, modeliuojamojo atkaitinimo algoritmas, paieškos su draudimais algoritmas.

## 1. Įvadas

Tvarkaraščiai naudojami įvairiose srityse (logistikoje, vadyboje, versle, gamyboje ir kt.). Sudarytas tvarkaraštis smarkiai įtakoja produkcijos gamybą, gaunamą pelną bei verslo sėkmę, todėl labai svarbu sudaryti tokį, kuris su mažiausiomis laiko bei išteklių sąnaudomis duotų geriausius rezultatus. Tvarkaraščių sudarymo uždaviniai sprendžiami įvairiais kombinatorinio optimizavimo metodais.

Toliau pateiktas uždavinys yra aprašytas [6] su tam tikrais pakeitimais. Nagrinėjamas nuoseklios gamybos uždavinys, susidedantis iš  $m \geq 2$  cechų (mašinų centrų)  $[Z_1, Z_2, \dots, Z_m]$ , kurių kiekviename yra  $m_j \geq 1$  lygiagrečių mašinų  $\{M_{j1}, M_{j2}, \dots, M_{jm_j}\}$ . Mašinos  $M_{jk}$  spartą nusako  $g_{jk}$ , o vidutinė  $j$ -tojo cecho mašinų sparta  $ag_j = \sum_{k=1}^{m_j} g_{jk}$ . Mašinos dirba su nedarbingumo intervalais. Tariame, kad kiekvienos mašinos nedarbingumo intervalai nepersidengia ir yra išrikiuoti didėjančia tvarka pagal nedarbingumo intervalo pradžios momentus. Mašinos  $M_{jk}$   $l$ -tojo nedarbingumo intervalo pradžios momentą ir ilgį pažymėkime atitinkamai  $s_{jk,l}$  ir  $h_{jk,l}$ . Reikia atlikti  $n$  darbų  $\{J_i | 1 \leq i \leq n\}$ . Kiekvienas darbas  $J_i$  susideda iš  $m$  operacijų, kurių kiekvienai įvykdyti reikalingas laikas  $p_{ij} (1 \leq i \leq n, 1 \leq j \leq m)$ . Visi darbai atliekami ta pačia tvarka, operacija turi būti atlikta viena iš ceche esančių lygiagrečių mašinų. Kiekviena mašina vienu laiko momentu gali atlikti tik vieną operaciją, o kiekviena operacija vienu laiko momentu gali būti atliekama daugiausia viena mašina. Uždavinio esmė – minimizuoti visų darbų pabaigos laiką, t.y. maksimalų laiką, reikalingą visoms operacijoms atlikti.

Kitame skyrelyje pateikiami šiame darbe nagrinėti algoritmai: šakų ir ribų algoritmas, modeliujamojo atkaitinimo algoritmas bei paieškos su draudimais algoritmas. Trečioje dalyje – atlikti tyrimai ir gauti rezultatai.

## 2. Nagrinėti algoritmai

**Šakų ir ribų algoritmas.** Šakų ir ribų algoritmas yra tikslusis algoritmas skirtas įvairiems optimizavimo uždaviniams, ypač diskretaus ir kombinatorinio optimizavimo uždaviniams, spręsti. Pirmą kartą šis metodas buvo pasiūlytas 1960 m. [7]. Šakų ir ribų algoritmas gali būti aprašytas taip:

- konstruoti paieškos medį;
- genėti (mažinti) šakas;
- baigti paiešką, kai tenkinamos tam tikros sąlygos.

Šakų ir ribų algoritmas susideda iš kelių paprastų operacijų, taikomų skirtingo ar vienodo prioriteto viršūnėms, kurios sudaro viršūnių aibę, dažnai vadinamą pirmumo eile:

- *ištrinti\_min* – pasirinkti ir ištrinti aukščiausią prioritetą turintį elementą;
- *įterpti* – įterpti naują elementą, kurio prioritetas žinomas;
- *ištrinti\_didesnį* – ištrinti elementus, kurie netenkina tam tikrų sąlygų.

**Modeliuojamojo atkaitinimo algoritmas.** Modeliuojamasis atkaitinimas yra plataus naudojimo tikimybinis meta-euristinis algoritmas, skirtas rasti gerą globalaus optimumo artinį duotajai funkcijai, kai ieškojimų erdvė yra plati. Jis nepriklausomai buvo pristatytas S. Kirkpatrick, C.D. Gelatt ir M.P. Vecchi 1983 m., bei V. Černý 1985 m. Šis metodas sukurtas kaip Monte Carlo metodo apibendrinimas, skirtas tirti sistemos, sudarytos iš  $n$  kūnų, būsenos lygtis ir sustingusias būsenas [1,7]. Modeliuojamojo atkaitinimo metodas yra naudojamas įvairiems kombinatorinio optimizavimo uždaviniams, pavyzdžiui, grupavimo, tvarkaraščių sudarymo, spręsti. 1 pav. pateiktas modeliujamojo atkaitinimo algoritmo pseudo kodas [3].

**Paieškos su draudimais algoritmas.** Paieškos su draudimais metodas priskiriamas lokališios paieškos metodų klasei [4,7]. Pagrindinis tikslas yra išvengti ciklų susidarymo, uždraudžiant ar skiriant baudos taškus ėjimams, kurie parenka iš sprendinių aibės sprendinius, artimus neseniai tikrintiems. 2 pav. pateiktas paieškos su draudimais algoritmo pseudo kodas [5].

## 3. Atliktas tyrimas ir gauti rezultatai

Naudojant sukurtą programą, tiriamas metodų efektyvumas, gaunamų rezultatų priklausomybė nuo parinktų parametrų. Kad gautus rezultatus būtų lengviau analizuoti ir daryti išvadas, nuspręsta, kad duomenų failuose visos mašinos bus skirtingų tipų, vienodos spartos ir dirbs visą laiką, operacijos negalima nutraukti, o kita operacija galės būti pradėta vykdyti tik šiai pasibaigus. Tyrimui naudoti šiai programai pritaikyti duomenų failai iš J.E. Beasley tinklalapio [2].

Kadangi modeliujamojo atkaitinimo ir paieškos su draudimais metoduose pradinis uždavinio sprendinys generuojamas atsitiktinai, o modeliujamojo atkaitinimo metode dar naudojama ir tikimybė, todėl rastas darbų pabaigos laikas yra atsitiktinis. Dėl šios

```

// k - Bolcmano konstanta;
// maxit - maksimalus iteracijų skaičius
// T0 - pradinė temperatūra

s,          // pradinis sprendinys
s',         // geriausias sprendinys
sn,        // dabartinis sprendinys
sb;        // buvęs sprendinys
it = 0;    // iteracijų skaičius
T = T0;    // temperatūra
sn = s;
sb = s;

kol(T > 0){
  kol(it <= maxit) {
    it = it + 1;
    sn = Kaimynas(); // generuojamas kaimyninis sprendinys
    jei(sn.pabaiga() < sb.pabaiga()) {
      jei(sn.pabaiga() < s'.pabaiga()) {
        s' = sn;
        sb = sn;
      }
      kitu atveju jei(atsSkaicius() < exp(-(sn.pabaiga() - sb.pabaiga())/(kT)))
        sb = sn;
    }
    T = MažintiTemp(); // pagal pasirinktą metodiką mažinama temperatūra
    it = 1;
  }
}

```

1 pav. Modeliuojamojo atkaitinimo algoritmo pseudo kodas.

```

// k - iteracijų skaičius
// maxK - maksimalus iteracijų skaičius

s // pradinis sprendinys
so // geriausias sprendinys
sg // geriausias kaimyninis sprendinys

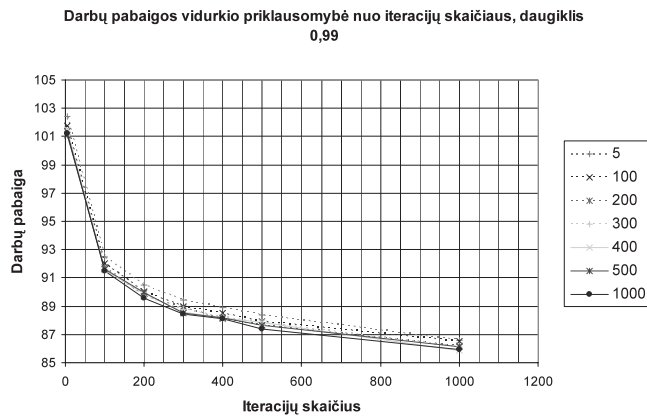
so = sg = s;
kol(k <= maxK YraNedraudžiamųKaimyniniųSprendinių(s)){
  sg = GeriausiasKaimyninis(s); // iš visų kaimyninių sprendinių išrenkamas tas,
  // su kuriuo tikslo funkcija (pabaiga) yra mažiausia
  s = sg;
  jei(s.pabaiga() < so.pabaiga()) {
    so = s;
    k = 0;
  } kitu atveju
    k = k + 1;
  AtnaujintiDraudimųSarašą();
}

```

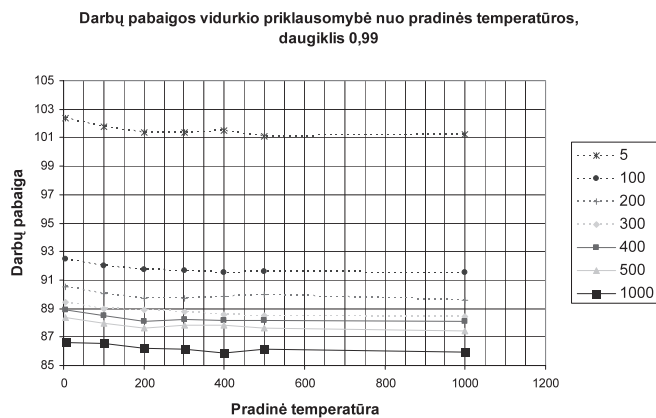
2 pav. Paieškos su draudimais algoritmo pseudo kodas.

priežasties tvarkaraštis sudaromas 100 kartų, išsaugomas geriausias rastas rezultatas (mažiausias darbų pabaigos laiko momentas), apskaičiuojama darbų pabaigos empirinis vidurkis, empirinė dispersija ir vidutiniškai per kiek laiko sudaromas vienas tvarkaraštis. Kiekvienam parametru rinkiniui eksperimentas pakartojamas 5 kartus ir kiekvienam stebimam dydžiui apskaičiuojamas vidurkis. Šakų ir ribų metodas yra tikslus ir nepriklauso nuo jokių parametru, todėl tvarkaraštis sudaromas tik vieną kartą.

Naudojant šakų ir ribų algoritmą įvairaus sudėtingumo uždaviniams spręsti, tvarkaraščiui sudaryti reikalingas laikas sparčiai didėjo sudėtingėjant uždaviniui (nuo kelių ms iki kelių parų). Naudojant modeliuojamąjį atkaitinimą, darbų pabaigos vidurkio priklausomybė nuo iteracijų skaičiaus bei pradinės temperatūros nagrinėjant vieną iš testinių failų pateikiama 3 ir 4 pav. Testuojant vieną iš testinių failų naudojant



3 pav. Darbų pabaigos vidurkio priklausomybė nuo iteracijų skaičiaus.

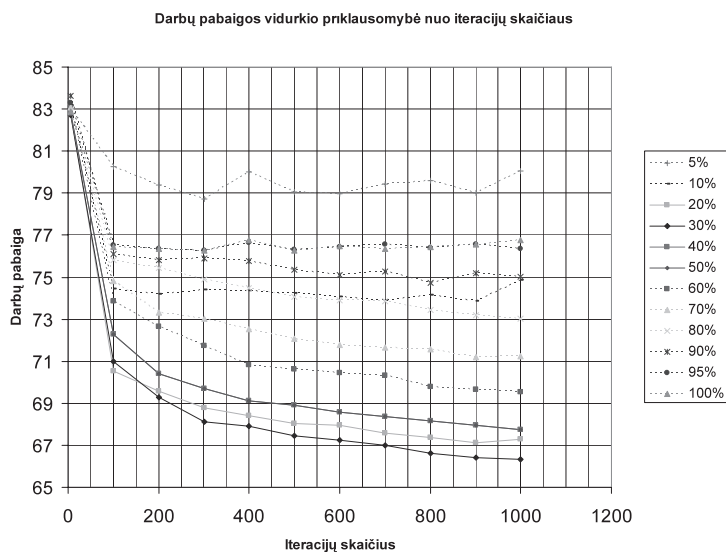


4 pav. Darbų pabaigos vidurkio priklausomybė nuo pradinės temperatūros.

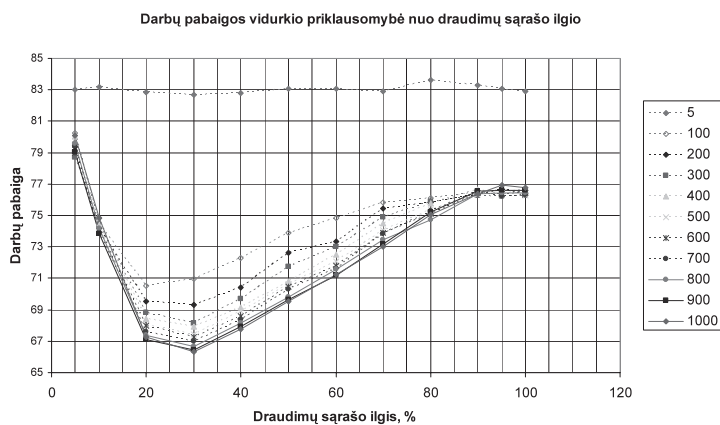
paiešką su draudimais, darbų pabaigos vidurkio priklausomybė nuo iteracijų skaičiaus bei draudimų sąrašo ilgio pateikiama 5 ir 6 pav.

#### 4. Išvados

1. Šakų ir ribų algoritmas labai jautrus uždavinio sudėtingumui laiko, reikalingo optimaliam sprendiniui rasti, atžvilgiu. Šį algoritmą patartina naudoti tik mažos apimties uždaviniams spręsti.
2. Naudojant modeliuojamojo atkaitinimo algoritmą, geriausi rezultatai gaunami, kai iteracijų skaičius yra nuo 100 iki 1000, pradinė temperatūra apie 1000, o temperatūros daugiklis – 0,95–0,99.
3. Naudojant paieškos su draudimais algoritmą, geriausi rezultatai gaunami, kai iteracijų skaičius yra 900–1000, draudimų sąrašo ilgis – 30%–70%.



5 pav. Darbų pabaigos vidurkio priklausomybė nuo iteracijų skaičiaus.



6 pav. Darbų pabaigos vidurkio priklausomybė nuo draudimų sąrašo ilgio.

4. Remiantis gautais rezultatais bei skaičiavimams atlikti reikalingu laiku, galima teigti, kad tikslusis metodas – šakų ir ribų – yra efektyvus tik nesudėtingiems uždaviniams spręsti. Sprendžiant sudėtingesnius uždavinius efektyvesni yra meta-euristiniai metodai.
5. Iš dviejų nagrinėtų meta-euristinių metodų, modeliuojamojo atkaitinimo ir paieškos su draudimais, pastarasis yra efektyvesnis.

### Literatūra

1. GAUL: Genetic Algorithm Utility Library. [žiūrėta 2009-05].  
<http://gaul.sourceforge.net/tutorial/sa.html>
2. Home Page J.E. Beasley. [žiūrėta 2009-05].  
<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/flowshop1.txt>
3. Simulated Annealing. [žiūrėta 2007-11].  
<http://members.aol.com/btluke/simann1.htm>
4. Tabu Search. [žiūrėta 2009-05].  
<http://www.cs.sandia.gov/opt/survey/ts.html>
5. Tabu Search. [žiūrėta 2009-05].  
<http://www.idsia.ch/~monaldo/tabusearch.html>
6. X. Wang, J. Xie. Branch and bound algorithm for flexible flowshop with limited machine availability. *Asian Information-Science-Life*, 1(3):241–248. [žiūrėta 2009-05].  
<http://faculty.math.tsinghua.edu.cn/~jxie/papers/AISF2002.pdf>
7. Wikipedia The Free Encyclopedia. [žiūrėta 2009-05]. <http://en.wikipedia.org/wiki>

### SUMMARY

#### **L. Rajeckaitė, N. Listopadskis. Analysis of flow shop problems and their algorithms**

The combinatorial optimization problem considered in this paper is flow shop scheduling problem arising in logistics, management, business, manufacture and etc. A set of machines and a set of jobs are given. Each job consists of a set of operations. Machines are working with unavailability intervals. The task is to minimize makespan, i.e. the overall length of the schedule. There is overview of combinatorial optimization, scheduling problems and methods used to solve them. There is also presented and realized one exact algorithm – Branch and Bound, and two meta-heuristics: Simulated Annealing and Tabu Search. Analysis of these three algorithms is made.

*Keywords:* flow shop, branch and bound, simulated annealing, tabu search.