

Genetinio ir tabu paieškos algoritmų naudojimo gamybinių tvarkaraščių sudarymui analizė

Edgaras ŠAKUROVAS, Narimantas LISTOPADSKIS (KTU)

el. paštas: edgaras.sakurovas@stud.ktu.lt, narlis@ktu.lt

Reziumė. Šiame darbe sprendžiama gamybinių (darbų, srautinio, atvirojo fabriko) tvarkaraščių sudarymo problema, kuri priklauso kombinatorinio optimizavimo uždavinių klasei. Naudojami metaeuristiniai metodai: genetinio algoritmo modifikacijos valdymo parametrų atžvilgiu ir tabu paieškos modifikacijos sprendinių aplinkos atžvilgiu. Remiantis atliktu įgyvendintų algoritmų efektyvumo tyrimu analizuojama modifikacijų įtaka algoritmų kokybei.

Raktiniai žodžiai: genetinis algoritmas, tabu paieška, algoritmo strategija, algoritmo modifikacija.

1. Įvadas

Tvarkaraščių sudarymo problema šiame industriniame amžiuje yra išties aktuali: traukinių maršrutų tvarkaraštis, darbo valandų tvarkaraštis, materialių resursų tvarkaraščiai bei pan.

Šiame darbe nagrinėjome gamybinius tvarkaraščius: darbų fabriko (kiekvienai užduočiai atskira darbų tvarka), srautinio fabriko (visose darbų atlikimo mašinose tvarka vienoda) ir atvirojo fabriko (nėra atlikimo tvarkos apribojimų) problemas.

Dauguma tvarkaraščio sudarymo problemų negali būti išspręstos tiesioginiais arba taip vadinamais jėgos metodais (pavyzdžiui, visų sprendinių išnagrinėjimu), visų pirma dėl per didelių laiko ir skaičiavimo technikos resursų sąnaudų. Čia atrodo labiau tinkamesni tie metodai, kurie gali užkoduoti tvarkaraštį-sprendinį į labiau prieinamą arba lengviau išsprendžiamą formą, tuomet toje erdvėje atlikti atitinkamus tyrimus ir sugražinti (atkoduoti) optimalų sprendinį, neprarandant abipusio vienareikšmiškumo. Tokiu būdu ši problema siejama su kombinatorinio optimizavimo uždavinių klase.

Minėtų metodų savybėmis pasižymi metaeuristiniai metodai (plačiau žr. [1,5,9]). Pastarųjų klasei priklauso genetiniai (plačiau žr. [3,7,10]) ir tabu paieškos (plačiau žr. [2,4,6]) algoritmai.

Genetiniai algoritmai paprastai įgyvendinami kaip tam tikrų individų (sprendinių) populiacijos evoliucijos kompiuterinė imitacija: individai tarpusavyje dirbtinai kryžminami ir/arba mutuoja, tikintis gerų savybių išlikimo ateinančiose kartose.

Tabu paieškos metodas paremtas lokaliaus paieškos principais, tačiau turi papildomų komponentų (priešingai nuo lokaliaus paieškos) neleidžiančių sustoti lokaliame minimume.

Tam, kad paieškos metodas būtų pavadintas tabu paieška, reikalingos trys charakteristikos [8]: a) tabu: laikini apribojimai, išskiriant tam tikrus pasirinkimus iš paieškos;

b) trumpalaikė atmintis: tabu sąrašas tam, kad tam tikrą laiko periodą apsaugotų nuo grįžimo į ankstesnius žingsnius; c) aspiracijos lygis: kriterijai, skirti apeiti tabu.

2. Algoritmų taikymai

Problemos ir sprendimo vientisumui užtikrinti visiems algoritmams buvo taikytas permutacinis sprendinio kodavimas ($c = n \cdot m$, kur c – sprendinio reprezentacijos simbolių kiekis, n – tvarkaraščio sudarymo problemos darbų skaičius, m – problemos mašinių skaičius) bei sprendinio įverčio funkcija, įtraukianti visų trijų tipų tvarkaraščių apribojimus (plačiau žr. [11]).

Genetiniai algoritmai. Bazinis genetinis algoritmas pritaikytas taip pat kaip ir minėtoje [11] publikacijoje, tačiau atlikus tabu paieškos algoritmo taikymo tyrimus, padaryti tam tikri patobulinimai. Juos reziumuoja 1 lentelė. Jei pirmos dvi strategijos pakankamai akivaizdžios ir natūralios, tai trečios strategijos esmė apibūdinama taip: iš pradžių algoritmas yra su kiek padidinta intensifikacija, iteracijoms didėjant intensifikacija mažėja, didėja diversifikacija. Jei geriausias sprendinys nepagerinamas, intensifikacija sumažėja iki minimumo, o diversifikacija padidėja iki maksimumo. Kaskart (kas kelias ar keliasdešimt iteracijų) sprendiniui gerėjant, intensifikacija atitinkamai didinama, diversifikacija mažinama. Jei algoritmą išstiko stagnacija, sugeneruojami atsitiktiniai sprendiniai (žr. 1 pav.). Šios strategijos parametrai:

$$K = 0,8 + \frac{kitMin}{iterSk} \cdot 0,15 + \frac{einIt}{iterSk} \cdot 0,05,$$

$$M = 0,05 + \frac{kitMin}{iterSk} \cdot 0,8 + \frac{einIt}{iterSk} \cdot 0,05,$$

$$Mlygis = 1 + f_{floor} \left(\frac{kitMin \cdot c}{2 \cdot iterSk} - 1 \right),$$

čia $kitMin$ – geriausio surasto sprendinio stabilios būsenos iteracijose skaičius, $einIt$ – einamoji algoritmo iteracija, $iterSk$ – maksimalus iteracijų skaičius (jei algoritmas nebus nutrauktas), f_{floor} – funkcija, kuri gražina tik sveikąją dalį, $Mlygis$ – mutacijos lygis, t.y. kiek kartų sprendiniui bus pritaikyta dviejų elementų sukeitimo mutacija su tikimybe M . Algoritmas sugeneruoja naujus sprendinius, jei $kitMin$ daugiau nei ketvirtadalis visų iteracijų.

Tabu paieška. Bazinis tabu paieškos algoritmas remiasi kanoniniu tabu paieškos metodo apibrėžimu (atliekami keli atsitiktiniai startai, tabu sąrašo ilgis “magiška” 7 reikšmė [4], išrenkamas geriausias sprendinys tenkinantis tabu apribojimus ar geresnio naujo sprendinio aspiracijos kriterijų, algoritmo sustojimo sąlyga – iteracijų skaičius). Sprendinio aplinkos sudarymo funkcija f paremta Hemingo atstumo (šiuo atveju lygaus 2) tarp dviejų simbolių eilučių idėja, norint neišeiti iš lokalsios aplinkos apibrėžimo rėmų. Tačiau dėl baigtinio tabu sąrašo ilgio algoritmas turi ciklą bei dėl artimos Hemingo prasme aplinkos algoritmo konvergavimas lėtas.

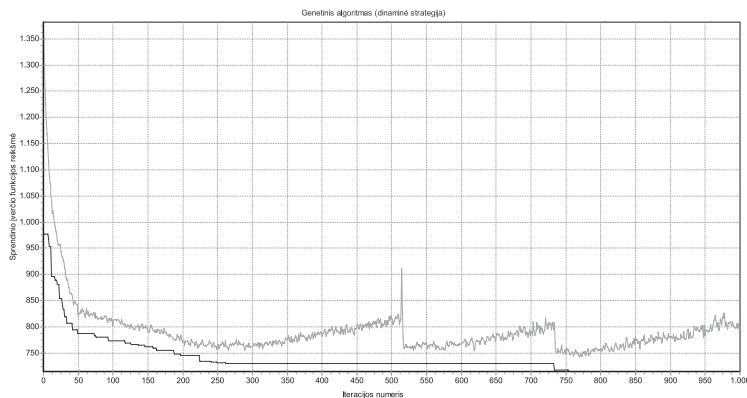
Siekiant išvengti minėtų trūkumų buvo atlikta keletas modifikacijų: 1) diferencinė atranka – modifikuota geriausio sprendinio-kandidato atranka. Jos esmė yra ta, kad kandidato įverčio funkcijai suteikiama tam tikra bauda, kuo daugiau kandidatas skiriasi nuo globalaus minimumo, tuo mažesnis jo įvertis ir atvirkščiai (žr. 2 pav.).

2) Padidinta diversifikacija – modifikuota aplinkos sudarymo funkcija. Visiems sprendinio aplinkos nariams kartu taikomas dviejų ir daugiau elementų sukeitimas, t.y. $Klygis = f_{ceil}(\frac{kitMin-c}{iterSk})$, kur $Klygis$ – aplinkos narių elementų sukeitimų skaičius, f_{ceil} – apvalinimo funkcija, kuri gražina sveikąją dalį + 1. Tabu paieškos su padidinta diversifikacija pavyzdys pateiktas 3 pav.

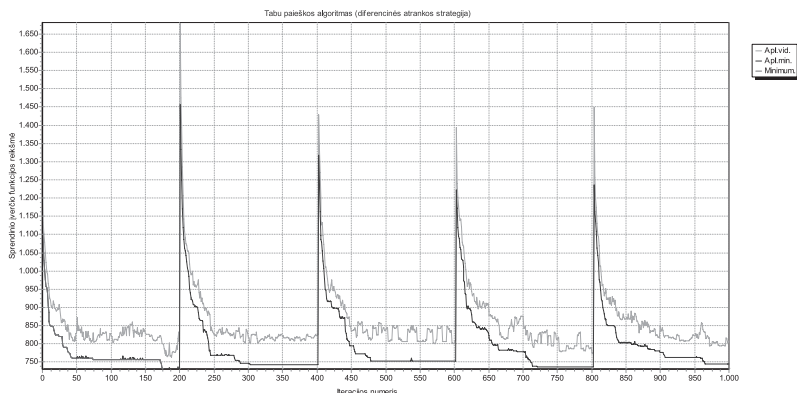
Kiti algoritmai. Norėdami pavaizduoti nagrinėjamos problemos sudėtingumą, pritaikėme du įgyvendinimo prasme paprastus algoritmus: tikslios pilnutinės (šiuo atveju dalinės) paieškos ir atsitiktinio klaidžiojimo.

1 lentelė. Siūlomos genetinio algoritmo strategijos (algoritmo parametrų atžvilgiu)

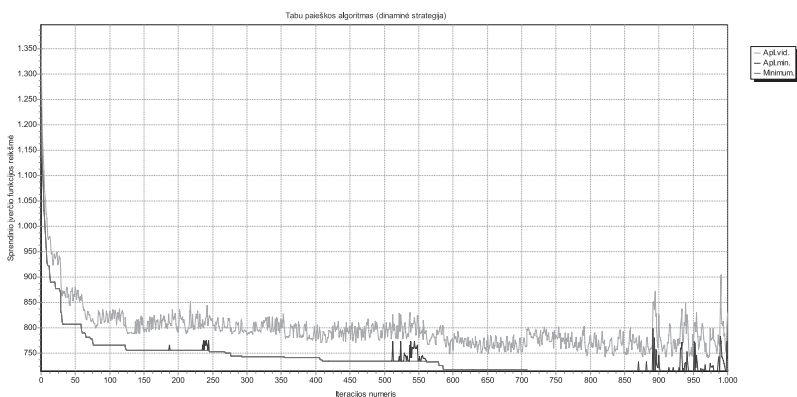
| Parametras | Stabilioji | Intensyvioji | Dinaminė |
|----------------------|--|---|--|
| Iteracijų skaičius | priklausomas nuo problemos (pvz., $10 \cdot m \cdot n$) | priklausomas nuo problemos (pvz., $5 \cdot m \cdot n$) | skaičius nustatomas algoritmo eigoje (pagrindas $10 \cdot m \cdot n$) |
| Populiacijos dydis | $2 \cdot m \cdot n$ | $m \cdot n$ | $2 \cdot m \cdot n$ |
| Kryžminimo tikimybė | $\sim 0,8$ | $\sim 0,9$ | tikimybė nustatoma algoritmo eigoje (pagrindas 0,8) |
| Mutacijos tikimybė | $\sim 0,1$ | $\sim 0,3$ | tikimybė nustatoma algoritmo eigoje (pagrindas 0,1) |
| Elitarizmo procentas | ~ 20 | ~ 5 | ~ 50 (tam, kad nepablogintų daugumos narių) |



1 pav. Genetinio algoritmo dinaminės strategijos sprendinių įverčių grafiko pavyzdys 10×10 formato Open Shop problemai. Viršutinė kreivė – einamosios sprendinių populiacijos vidurkis, apatinė – einamasis minimumas.



2 pav. Tabu paieškos su diferencine atranka algoritmo sprendinių įverčių grafiko pavyzdys 10×10 formato Open Shop problemai. Viršutinė kreivė – einamosios sprendinio aplinkos vidurkis, vidurinė – einamasis minimumas, apatinė – geriausias surastas minimumas.



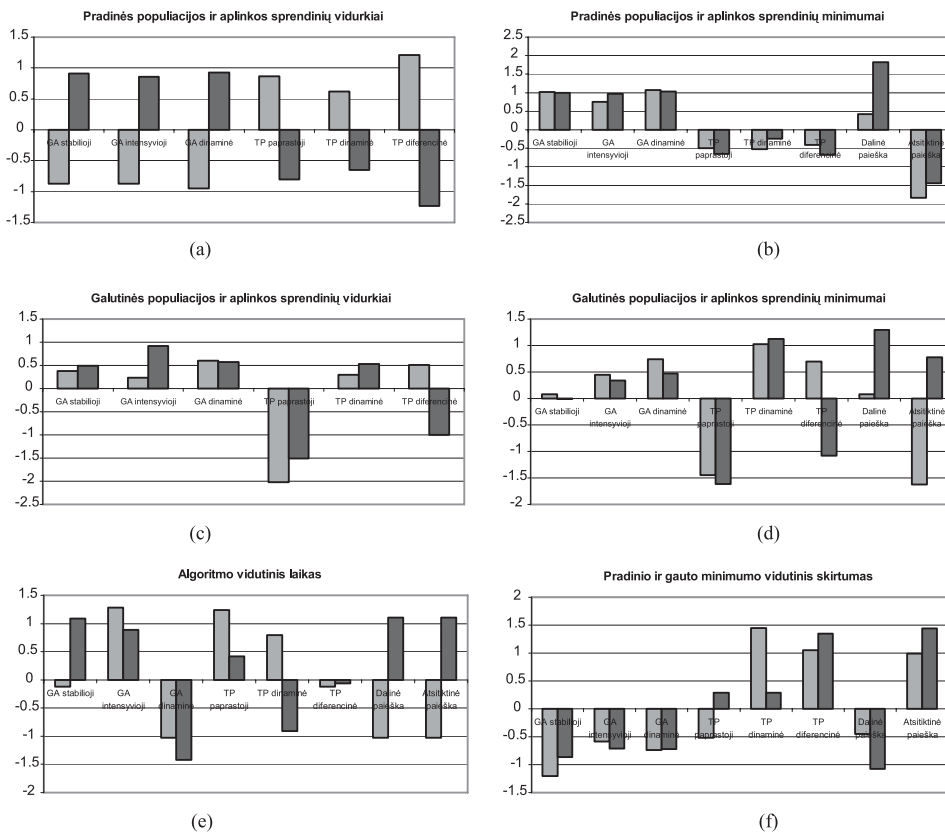
3 pav. Tabu paieškos su padidinta diversifikacija algoritmo sprendinių įverčių grafiko pavyzdys 10×10 formato Open Shop problemai. Viršutinė kreivė – einamosios sprendinio aplinkos vidurkis, vidurinė – einamasis minimumas, apatinė – geriausias surastas minimumas.

Kaip pilnosios paieškos pavyzdį galime pateikti vieną iš atvirojo fabriko problemų (3×3 formato), kuriai pabandėme rasti tikslų sprendinį. Programa jį apskaičiavo per 1 s 92 ms. Atlikę paprastus algebrinius skaičiavimus, radome, jog 3×4 problemai skaičiavimo laikas ≈ 42 min 14 s 40 ms, o 4×4 problemai ≈ 3 metai 6 mėn. 3 paros 6 h 43 min 12 s.

3. Tyrimai ir rezultatai

Šiame darbe nagrinėjamos tvarkaraščių sudarymo problemoms spręsti bei algoritmams realizuoti “Borland C++ Builder” terpėje buvo sukurta programinė įranga “GATS”. Duomenys eksperimentams sugeneruoti atsitiktinai naudojant tolygųjį intervalė [0, 100] skirstinį.

Algoritmų efektyvumui kiekvienai tvarkaraščių sudarymo klasei sugeneravome 100 atsitiktinių 10 × 10 formato problemų ir 100 kartų taikėme algoritmus. Visi algoritmai buvo leidžiami atitinkamą iteracijų skaičių, tik dalinės bei atsitiktinės paieškos algoritmai buvo vykdomi tiek, kiek ilgiausiai laiko užtruko prieš tai leistas algoritmas. Gauti rezultatai pateikti standartizuotoje (kokybinėje) formoje 4 pav.



4 pav. Aštuonių algoritmų efektyvumo tyrimo kokybiniai grafikai. GA – genetinis algoritmas, TP – tabu paieška. Kairysis stulpelis – algoritmo sprendinių standartizuotas vidurkis, dešinysis – standartizuotas nuokrypis.

4. Išvados

Akivaizdu, jog tikslioji ar atsitiktinė paieška yra netinkami metodai nagrinėjamai problemai spręsti.

Jei kalbame apie bazinius algoritmus, kanoniniai (stabiliosios ar intensyvios strategijos) genetiniai algoritmai pateikia aukštesnės kokybės sprendinius nei paprastoji tabu paieška. Tai lemia įvairesnė paieškos sritis, kuri geresnė kaip ir gauti sprendiniai intensyviosios strategijos atveju.

Genetinis algoritmas su dinamine strategija (padidintos diversifikacijos modifikacija) pateikia aukštos kokybės sprendinius (nusileidžia tik tabu paieškai su padidinta diversifikacija), tačiau lieka atviras algoritmo trukmės (laiko resursų) klausimas. Tabu paieškos algoritmo modifikacijomis sukuriama tinkama paieškos diversifikacija. Jei diferensinė atranka yra elementari modifikacija (tačiau gerokai pagerinusi algoritmo rezultatus), tai tabu paieškos algoritmas su padidintos diversifikacijos modifikacija pateikia geriausios kokybės sprendinius ir reikalauja tolimesnių studijų.

Literatūra

1. C. Blum, A. Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Computing Surveys*, **35**, 268–308 (2003).
2. F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Norwell (2001).
3. R.L. Haupt, S.E. Haupt, *Practical Genetic Algorithms*, Wiley-Interscience, New Jersey (2004).
4. A. Hertz, E. Taillard, D. de Werra, *A Tutorial on Tabu Search*, <http://www.cs.colostate.edu/~whitley/CS640/hertz92tutorial.pdf>
5. A. Misevičius, Intelektualieji optimizavimo metodai, *Informacijos mokslai*, **26**, 160–166 (2003).
6. A. Misevičius, J. Blonskis, Experiments with tabu search for random quadratic assignment problems, *Information Technology and Control*, **34**, 237–244 (2005).
7. M. Mitchell, *An Introduction to Genetic Algorithms*, A Bradford Book The MIT Press, Cambridge (1999).
8. S. Morrison, *Tabu Search for Process Scheduling*, <http://futurebatch.com/ScheduleOptimization/TabuSearch.pdf>
9. D.T. Pham, D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, Springer, London (2000).
10. C. Reeves, Genetic algorithms, in: F. Glover, G.A. Kochenber (Eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, New York (2003), pp. 55–82.
11. E. Šakurovas, N. Listopadskis, Genetinio algoritmo taikymas ir parametrų nustatymo problemos gamybinių tvarkaraščių sudarymui, *Liet. mat. rink.*, **47** (spec. nr.), 479–483 (2007).

SUMMARY

E. Šakurovas, N. Listopadskis. Analysis of usage of genetic and tabu search algorithms in shop scheduling

A wide area of scheduling problem is industrial so-called shop scheduling (Job Shop, Flow Shop and Open Shop) which has important applications in real world industrial problems. Metaheuristic algorithms (Genetic and Tabu search algorithms in this case) seem to be one of the best candidates for finding nearby-optima in proper time. In this work we implemented several genetic algorithms (separated by values of their parameters) and several Tabu search algorithms (separated by neighborhood of solution). Finally, implemented eight algorithms are examined for random shop scheduling problems in terms of various criteria.

Keywords: genetic algorithm, tabu search, algorithm strategy, algorithm modification.