

Nusileidimo–pakilimo principu pagrįsto euristinio algoritmo empiriniai rezultatai ir svarba informacijos tvarkyboje

Alfonsas Misevičius

Kauno technologijos universitetas
Multimedijos inžinerijos katedra
alfonsas.misevicius@ktu.lt
<https://orcid.org/0000-0002-3746-8200>
<https://ror.org/01me6gb93>

Gintaras Palubeckis

Kauno technologijos universitetas
Multimedijos inžinerijos katedra
gintaras.palubeckis@ktu.lt
<https://ror.org/01me6gb93>

Dovilė Verenė

Kauno technologijos universitetas
Multimedijos inžinerijos katedra
dovile.verene@ktu.lt
<https://ror.org/01me6gb93>

Santrauka. *Pagrindimas.* Kombinatorinio optimizavimo uždavinių sprendimas naudojant euristinius optimizavimo algoritmus yra labai aktuali skaitmeninės intelektikos, taip pat dirbtinio intelekto sritis. *Metodas.* Šiame straipsnyje pristatomi empiriniai (kompiuteriniai) rezultatai, kurie yra gauti eksperimentuojant su nusileidimo–pakilimo principu pagrįstu euristiniu optimizavimo algoritmu (NPA), skirtu, visų pirma, kombinatorinio optimizavimo uždaviniams spręsti. Savo ruožtu, nusileidimo–pakilimo algoritmas – taip trumpai jį įvardijant – savo ištakas turi vadinamosios lokaliosios paieškos (angl. *local search*) paradigmoje. Ypatybė yra ta, jog sprendžiamo uždavinio tikslo funkcijos minimumo paieška (t. y. nusileidimas) yra kombinuojama su tam tikromis sprendinių perturbacijomis (t. y. pakilimais), siekiant išvengti godaus (ir determinuoto) paieškos pobūdžio ir kartu priešlaikinės stagnacijos nebūtinai aukštos kokybės lokaliuosiuose optimumuose. *Rezultatai.* Atlikti eksperimentai su šiuo algoritmu ir gauti rezultatai liudija algoritmo santykinai aukštą efektyvumo laipsnį sprendžiant gerai žinomą kvadratinio paskirstymo (KP) uždavinį (angl. *quadratic assignment problem*). *Tyrimo praktinė reikšmė.* KP uždavinys yra svarbus žaliosios ekonomikos, naujos kartos industrijos, skaitmeninės transformacijos, atsinaujinančiosios energetikos, tvaryjų logistikos sistemų ir kituose socialinės-techninės sferos bei informatikos tvarkybos kontekstuose.

Pagrindiniai žodžiai: kombinatorinis optimizavimas; euristiniai optimizavimo algoritmai; lokalioji paieška; nusileidimo–pakilimo principu pagrįsta paieška; kvadratinio paskirstymo uždavinys; informacijos tvarkyba.

Received: 2025-07-28. Accepted: 2025-09-17.

Copyright © 2025 Alfonsas Misevičius, Gintaras Palubeckis, Dovilė Verenė. Published by Vilnius University Press. This is an Open Access article distributed under the terms of the [Creative Commons Attribution Licence](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

A Descent-Ascent Principle Based Heuristic Algorithm: Empirical Results and Relevance in Information Management

Abstract. *Rationale.* Solving combinatorial optimization problems by means of heuristic optimization algorithms is a highly important area of computational intelligence as well as artificial intelligence. *Methodology.* This paper presents empirical (computational) results obtained through experiments with a descent-ascent (D-A) principle-based heuristic optimization algorithm, primarily designed for solving combinatorial optimization problems. The descent-ascent algorithm – referred to briefly in this way – takes its origins from the local search paradigm. Its distinctive feature is that the minimization of the objective function of an optimization problem (i.e., descent) is combined with certain perturbations of solutions (i.e., ascents) in order to avoid a greedy/deterministic search behavior and, at the same time, premature convergence to suboptimal local optima. *Results.* The experiments conducted with this algorithm and the results obtained demonstrate a relatively high level of algorithmic efficiency in solving the well-known combinatorial problem – the quadratic assignment problem. *Practical relevance of the research.* The problem is highly relevant in such areas as the green economy, next-generation industry, digital transformation, renewable energy, sustainable logistics systems, and other socio-technical and information management contexts.

Keywords: combinatorial optimization, heuristic optimization algorithms, local search, descent-ascent principle-based search, quadratic assignment problem, information management.

Įvadas

Nuolat intensyvėjanti praktinė-eksperimentinė ir mokslinė-tiriamoji veikla skatina tai, jog nuolat atsiranda vis didesnis skaičius sudėtingų optimizavimo uždavinių, kurių efektyviam sprendimui neegzistuoja greitai veikiančių tikslųjų algoritmų. Šiuo atveju viena iš natūralių, logiškų išeikių yra euristiniai algoritmai (EA) (angl. *heuristic algorithms*), kaip viena iš svarbių dirbtinio intelekto metodų grupių (Edelkamp, Schrödl, 2012; Salhi, 2017; Du ir kt., 2022; Tsai, Chiang, 2023). Nors panaudojant šio tipo algoritmus ir nėra optimalių sprendinių suradimo garantijos, šiuo atveju atsiranda geros sąlygos žymiai sumažinti skaičiavimų apimtį, kas yra labai aktualu taupiau išnaudojant techninius-energetinius išteklius. Kita vertus, gaunami sprendiniai, nors ir nebūdami optimalūs, dažnai yra pakankamai aukštos kokybės ir gali visiškai patenkinti uždavinių sprendinių naudotojus. Pabrėžtina, jog naujų eksperimentinės plėtros rūšių iškeliami uždaviniai dažnai yra komplikotesni negu nuodugnai išnagrinėti klasikiniai matematiniai-teoriniai uždaviniai, taigi dar aktualesni tampa tyrimai, kuriais siekiama sukurti inovatyvius, efektyvesnius euristinius algoritmus (Amodeo ir kt., 2018; Kochenderfer, Wheeler, 2019; Abualigah, 2024; Wang ir kt., 2024). Pabrėžtina, jog efektyvių EA kūrimas yra svarbus ir platesniuose kontekstuose, tarp jų įvairiose socialinėse-techninėse sferose, pvz., žaliwoje ekonomikoje, naujos kartos industrijoje, skaitmeninėje transformacijoje, atsinaujinančiojoje, tausojančioje energetikoje, tvariiosiose logistikos sistemose, pažangiosiose miestų infrastruktūrose ir kitose informatikos tvarkybos srityse (žr. taip pat sk. „Kvadratinio paskirstymo uždavinys, jo svarba“).

Yra platus euristinių algoritmų įvairovės spektras, pradedant vienetinių sprendinių nagrinėjimo („vieno sprendinio“) algoritmais (angl. *single solution-based algorithms*) ir baigiant populiaciniais(-evoliuciniais), kolektyvinio intelekto bei gamtos inspiruotais

algoritmais (angl. *population(-evolutionary)-based, swarm intelligence, nature-inspired algorithms*) (Aarts, Lenstra, 1997; Bonabeau ir kt., 1999; Talbi, 2009; Yang, 2010a; 2010b; Brownlee, 2011; Misevičius ir kt., 2011; Drezner, Misevičius, 2013; Siarry, 2016; Bozorg-Haddad ir kt., 2017; Gendreau, Potvin, 2018; Martí ir kt., 2018; Misevičius, Kuznecovaitė, 2019; Khadem ir kt., 2023).

Šiame straipsnyje susitelkiama į vienetinių sprendinių nagrinėjimo algoritmus (dar žinomos kaip [paieškos] „trajektorijų“ algoritmus), paminint, visų pirma, (klasikinės) lokalsios paieškos (LP) (siaurąja prasme) bei tabu paieškos, taip pat iteratyviosios lokalsios / tabu paieškos algoritmus. Tik priminsime, jog vienas iš labai svarbių pozityvių LP algoritmų faktorių yra tas, jog šiems algoritmams nebūtina sprendinių populiacija, pakanka operavimo (manipuliavimo) tik su vienu sprendiniu. Nors lokalsios paieškos algoritmų istorija apima daugiau kaip 60 metų, ir dabartiniais laikais LP algoritmai kartais vis dar susilaukia akademinės bendruomenės dėmesio ir yra aktuali mokslinių tyrinėjimų tematika (Burke, Bykov, 2017; Amaral ir kt., 2021; Goudet ir kt., 2024).

Yra pasiūlyta naujų originalių LP algoritmų patobulinimų / modifikacijų; čia paminėtini tokie jų variantai: kintamojo gylio (lokalioji) paieška (angl. *variable depth search*) (Lin, Kernighan, 1973), „seno viengungio“ pasirinkimas (angl. *old bachelor acceptance*) (Hu ir kt., 1995), didelių aplinkų paieška (angl. *large neighbourhood search*) (Ahuja ir kt., 2000), valdomoji lokalioji paieška (angl. *guided local search*) (Voudouris, Tsang, 1999), „intensyvioji“ paieška (angl. *intensive search*) (Misevičius, 2000), iteratyvioji lokalioji paieška (angl. *iterated local search*) (Lourenco ir kt., 2002), hierarchinė iteratyvioji lokalioji paieška (angl. *hierachical iterated local search*) (Hussin, Stützle, 2009), lokalioji paieška su pertraukimais (angl. *breakout local search*) (Benlic, Hao, 2013), žingsnių įvertinimo (skaičiavimo) lokalioji paieška (angl. *step counting local search*) (Bykov, Petrovic, 2016), užvėlinto priėmimo (reagavimo) LP euristinis algoritmas (angl. *late acceptance hill climbing heuristic*) (Burke, Bykov, 2017), lokalioji paieška su (gerinimo) vėlinimu (angl. *delayed improvement local search*) (Amaral ir kt., 2021), ekstremalusis optimizavimas (angl. *extremal optimization*) (Boettcher, Percus, 1999).

Nepaisant šių jau pasiūlytų LP variantų, manytina, jog vis dar yra tikslinga ieškoti papildomų LP patobulinimų. Šia prasme viena iš perspektyvių kryptių yra lokalsios paieškos algoritmų tobulinimas tinkamai kombinuojant intensifikavimą (lokaliai optimalių sprendinių paiešką, kitaip tariant, nusileidimą) ir diversifikavimą (sprendinių erdvės „žvalgymą“ (angl. *exploration*), kitaip tariant, pakilimą), kas turėtų leisti atverti naujas potencialias galimybes ir sinergiją tarp nors ir kontrastingų, bet vienas kitą sėkmingai papildančių paieškos aspektų. Tokia perspektyvi kryptis yra nusileidimo–pakilimo (angl. *descent-ascent*) algoritmas, pasiūlytas (Misevičius ir kt., 2025) straipsnyje, kur tas algoritmas yra detalai pateiktas. Dabartiniame straipsnyje tik primename šio algoritmo idėją (pateikdami algoritmo bendrą veikimo principą), o pagrindinę dėmesį skiriame empiriniams NPA algoritmo tyrimams ir, be kita ko, konfigūraciniams parametrams, ieškodami kiek įmanoma efektyvesnio algoritmo realizavimo scenarijaus.

Straipsnio struktūra. Iš pradžių pristatomas NPA algoritmo funkcionavimo principo aprašymas, pagrindiniai parametrai. Po to pateikiami empirinių, t. y. kompiuterinių-eks-

perimentinių tyrimų rezultatai, gauti išbandžius skirtingus nusileidimo–pakilimo principo algoritminius realizavimo variantus, kitaip tariant, įvairias NPA algoritmo valdymo parametrų konfigūracines kombinacijas (variacijas). Straipsnis baigiamas apibendrinamosiomis pastabomis.

Nusileidimo–pakilimo algoritmas

Algoritmo idėjos esmė yra ta, jog norint, kad įvyktų „dideli“ dalykai, paprastai yra reikalingi ir tam tikri nuopuoliai. Algoritmų kontekste tai reiškia, jog sėkmingam algoritmo veikimui gali būti / yra naudinga tai, jog kartu su sprendinių pagerinimu (optimizavimu, t. y. nusileidimu) – minimizavimo uždavinių kontekste – būtų vykdomos ir tam tikros perturbacijos / mutacijos („invazijos“, t. y. pakilimai, kitaip sakant, pabloginimai) – vėlgi, kalbame apie minimizavimo uždavinius.

Dar konkrečiau, nusileidimo–pakilimo algoritmą (NPA) būtent sudaro iteraciniu būdu kartojami lokalsios paieškos (sprendinio pagerinimo) ir sprendinio randomizuotos perturbacijos / pertvarkymo žingsniai. Perturbacijų ypatybė yra ta, jog jos įgalina naujas sprendinių savybes ir leidžia perėjimus į vis kitas, dar nenagrinėtas paieškos erdvės sritis. Perturbacijos vyksta atsitiktiniais momentais – tam naudojami pirminiai skaičiai, kaip labai paprastas, natūralus atsitiktinumų šaltinis. Taigi, jeigu einamojo žingsnio (iteracijos) numeris (indeksas) yra pirminis skaičius, tai ir yra signalas (indikatorius) perturbacijai įvykdyti. Kalbant apie sprendinio pagerinimą, jo vaidmenyje gali būti ne tik įprastinė / klasikinė lokaloji paieška, bet ir tabu paieška, taip pat iteratyvioji lokaloji / tabu paieška, taip pat kelių lygių, t. y. hierarchinė iteratyvioji lokaloji / tabu paieška. Sprendinio pagerinimo lygį nusako lokalsios / tabu paieškos iteracijų (ir iteratyviosios lokalsios / tabu paieškos iteracijų) skaičius.

1 pav. Nusileidimo–pakilimo algoritmo bendrasis aprašas

1 Algoritmas //Nusileidimo–pakilimo principu pagrįsto algoritmo veikimo aukšto lygio aprašas

```
//Algoritmo duomenys: s – pradinis sprendinys (perstatymas)
// (Taip pat yra laikoma, jog yra žinoma (konkreči) uždavinio tikslo funkcija z)
//Algoritmo rezultatas: s* – geriausias surastas sprendinys (perstatymas)
//Algoritmo (valdymo) parametrai: N, T, IT, η, μ, TP, TK,
//      Q1, Q2 – du teigiami (natūriniai) skaičiai (Q1 < Q2 – C,
//      C – teigiama konstanta (rekomenduojama, pvz., C ≥ 10)

1. i = 1;
2. j = Q1;
3. if prime(j) = true then atliekamas dviejų atsitiktinai parinktų elementų
                        sukeitimas / perturbacija sprendinyje (perstatyme) s
   else begin // prime(j) = false
       (a) jeigu prieš tai buvusioje iteracijoje buvo nustatytas pirminis skaičius,
           tai atliekamas toks dviejų elementų sukeitimas sprendinyje
           (perstatyme) s/lokaloji paieška, jog yra pagerinama tikslo funkcija – z
           (jei tai įmanoma); jei tai įmanoma, einama į 4;
           jei tai neįmanoma, einama į 5;
```

-
- (b) priešingu atveju, jeigu prieš tai buvusioje iteracijoje buvo tikslo funkcijos pagerinimas, tai atliekamas toks dviejų elementų sukeitimas perstatyme $s/lokalioji$ paieška, jog yra pagerinama tikslo funkcija (jei tai įmanoma); jei tai įmanoma, einama į 4; jei tai neįmanoma, einama į 5;
- (c) priešingu atveju, t.y. jeigu prieš tai buvusioje iteracijoje nebuvo pagerinta tikslo funkcija, tai su vienoda tikimybe (1/3) pasirenkamas vienas iš variantų ((c1), (c2) ar (c3)):
- (c1) atliekamas dviejų atsitiktinai parinktų elementų sukeitimas / perturbacija perstatyme s , po to atliekamas toks dviejų elementų sukeitimas perstatyme $s/lokalioji$ paieška, jog yra pagerinama tikslo funkcija, su sąlyga, jog negrižtama į prieš tai buvusį perstatymą (jei tai įmanoma);
- (c2) atliekamas dviejų atsitiktinai parinktų elementų sukeitimas / perturbacija perstatyme s (tai atliekama du kartus), po to atliekamas toks dviejų elementų sukeitimas perstatyme $s/lokalioji$ paieška, jog yra pagerinama tikslo funkcija (jei tai įmanoma);
- (c3) atliekamas dviejų atsitiktinai parinktų elementų sukeitimas / perturbacija perstatyme s arba dviejų atsitiktinai parinktų elementų sukeitimas / perturbacija du kartus (su vienoda tikimybe (1/2)), po to atliekamas toks dviejų elementų sukeitimas perstatyme $s/lokalioji$ paieška, jog yra pagerinama tikslo funkcija, su sąlyga, jog negrižtama į prieš tai buvusį perstatymą (jei tai įmanoma)
-

end;

4. išimamas geriausias rastas sprendinys (perstatymas) (perstatymas su mažiausia tikslo funkcijos reikšme), jei toks rastas – s^* ;
 5. **if** $j < Q_2$ **then** $j = j + 1$ **else** $j = Q_1$;
 6. $i = i + 1$;
 7. jeigu $i \leq N$, tai grįžti (kartoti iteracijų) į 3; priešingu atveju eiti į 8;
 8. pabaiga (rezultatų grąžinimas).
-

Pastabos. 1. Pateiktas bazinės (pradinės) algoritmo versijos aprašas. 2. Nusileidimo–pakilimo algoritmas yra labai universalizuotas ir jame, savo ruožtu, galima naudoti įvairius lokalsios paieškos paradigma besiremiančius kitus euristinius algoritmus. Pavyzdžiui, mūsų konkrečiu atveju NPA algoritmas naudoja pradinį sprendinį, kuris suformuojamas panaudojant godžiosios randomizuotos adaptyvios paieškos procedūrą (angl. *greedy randomized adaptive search procedure* (GRASP)) (Li ir kt., 1994). Tuo tarpu lokalsios paieškos vaidmenyje yra naudojamas hierarchinis iteratyviosios tabu paieškos algoritmas (Misevičius ir kt., 2024).

Kalbant apie perturbacijas, tam pakanka tiesiog tam tikro skaičiaus sprendinio elementų atsitiktinių sukeitimų. (Čia turima omenyje perstatymų – kas ir nagrinėjama šiame straipsnyje – elementų sukeitimai.) Perturbavimo / mutavimo proceso stiprumas kontroliuojamas panaudojant parametą μ (teoriškai galioja apribojimas $2 \leq \mu \leq n$, čia n yra uždavinio dydis (elementų skaičius)). (Parametras μ yra svarbus perturbavimo procesą ir apskritai visą NPA algoritmo veikimą reguliuojantis faktorius. Kuo didesnė μ reikšmė,

tuo labiau perturbuotas sprendinys „nutolsta“ nuo esamo sprendinio ir atvirkščiai; taigi, turi būti pasirenkamas kompromisinis variantas tarp dviejų ekstremalių atvejų.)

Viena iš NPA algoritmo siekiamybių yra išsiaiškinti tinkamą paieškos intensifikavimo (nusileidimo) ir diversifikavimo (pakilimo) veiksnių balansą. Taip pat svarbu yra tinkamai diversifikuoti paieškos procesą naudojant skirtingas perturbacijas ir taip bandant išvengti paieškos stagnacijos, kas yra vienas pagrindinių barjerų, jeigu kalbame apie euristinius algoritmus.

NPA algoritmo bazinio varianto detalizuotas aprašas yra pateiktas 1 paveiksle.

Prieš pereidami prie eksperimentinės dalies, pateikiame pagrindinius NPA algoritmo valdančiuosius parametrus:

- bendras NPA algoritmo iteracijų skaičius – N ;
- tabu paieškos (TP) (esančios hierarchinės iteratyviosios tabu paieškos algoritmo viduje) iteracijų skaičius – T ;
- (hierarchinės) iteratyviosios tabu paieškos (ITP) (esančios NPA algoritmo algoritmo viduje) iteracijų skaičius – IT ;
- perturbacijų (esančių NPA algoritmo viduje) stiprumas / lygis – μ ;
- perturbacijų (esančių ITP algoritmo viduje) stiprumas / lygis – η ;
- tabu paieškos (esančios perturbavimo procedūros viduje) iteracijų skaičius – TP ;
- tabu paieškos (esančios perturbavimo procedūros viduje) (pa)kartojimų skaičius – TK .

Kvadratinio paskirstymo uždavinys, jo svarba

Mūsų atliktuose empiriniuose tyrimuose NPA algoritmas buvo išbandytas su gerai žinomu kvadratinio paskirstymo (KP) uždaviniu (Çela, 1998). Kvadratinio paskirstymo uždavinys matematiškai formuluojamas taip: duotos dvi neneigiamų sveikųjų skaičių matricos $A = (a_{ij})_{n \times n}$ ir $B = (b_{kl})_{n \times n}$, taip pat aibė Π_n , kurią sudaro visi galimi natūrinių skaičių nuo 1 iki n perstatymai¹. Tikslas yra surasti perstatymą $p = (p(1), p(2), \dots, p(n)) \in \Pi_n$ tokį, jog būtų minimizuota ši funkcija:

$$z(p) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{p(i)p(j)}. \quad (1)$$

Dėstymo teorijos (angl. *location theory*) kontekste su KP uždaviniu gali būti modeliuojamas n įrengimų išdėstymas į n paskyrimo vietų, turint tikslą sumažinti bendrus, suminius kaštus / kainą, atsižvelgiant į tam tikrus srautus tarp įrengimų bei įrengimų susietumą ir atstumus tarp paskyrimo vietų (Koopmans, Beckmann, 1957). Čia matricos A reikšmės asocijuojamos su srautų kiekiais (svoriais) tarp įrengimų, o matricos B reikšmės nurodo atstumus tarp paskyrimo vietų. Perstatymas $p = (p(1), p(2), \dots, p(n))$ gali būti interpretuojamas kaip įrengimų išdėstymo paskyrimo vietose konfigūracija. Atskiras perstatymo

¹ Priminsime, jog perstatymą p formaliai galima nusakyti rinkiniu rinkiniu $p = (p(1), p(2), \dots, p(n)); p(i) \in \{1, 2, \dots, n\}; i = 1, 2, \dots, n; p(i) \neq p(j), \text{ kai } i \neq j; i, j = 1, 2, \dots, n$.

elementas $p(i)$ šiuo atveju nurodo numerį vietos, į kurią paskirtas i -tasis įrengimas. Tokiu būdu z gali būti suprantama kaip tam tikra kaina, kuomet visi įrengimai yra išdėstyti atitinkamose n vietų.

Atskiras KP uždavinio atvejis yra vadinamasis minimalaus skirtingumo (MS) uždavinys (arba minimalaus skirtingumo klasterio sudarymo uždavinys). Esant šiai situacijai, visi matricos A elementai (išskyrus diagonalę) yra lygūs 1, o tikslo funkcija išreiškiama taip:

$$z(p) = \sum_{i=1}^m \sum_{j=1}^m b_{p(i)p(j)},$$

čia m ($m < n$) yra klasterio dydis.

Šiuo atveju atsiranda naujų, aktualių taikymų – su sąlyga, jog tikslas yra suformuoti minimalaus skirtingumo, t. y. minimaliai konfliktuojančią, frustruojančią aibę, o (są)ryšiai tarp aibės elementų yra išreiškiami kiekybiškai. Šiame kontekste, visų pirma, paminėtinos tokios sociotechninės sritys:

- harmoningų firmų (įskaitant virtualiąsias) kūrimas;
- nekonfliktuojančių darbo grupių formavimas pačiose firmose ar organizacijose;
- gerai suderintų kuopelių sudarymas tam tikroms, specifinėms projektinėms užduotims / tikslams įgyvendinti;
- darnių komandų suformavimas komandinio sporto rungtyse;
- duomenų centrų užduočių paskirstymas;
- telekomunikacinių potinklų projektavimas;
- kitų nekonfliktinių darinių (asociacijų, bendruomenių ir pan.) sudarymas.

Mūsų šiuolaikiniame gyvenime labai svarbūs yra ir komunikavimo, rinkodaros, dirbtinio intelekto įrankių panaudojimo ir kiti informacijos tvarkybos bei valdymo aspektai. Žvelgiant iš šio požiūrio taško, MS uždavinys gali turėti ir dar daugiau pritaikymų. Čia, visų pirma, turima omenyje sudarymas optimizuotų socialinių poaibių, pvz., socialinių tinklų tam tikrų pogrupių, kuriuos sietų bendra ar panaši profesinė veikla, giminingi pomėgiai, interesai, siekiai ir kurie galėtų lemti vienijančius tikslus bei produktyvius rezultatus.

Pagaliau, MS uždavinys gali būti taikomas ir įvairiose rekomendavimo sistemose, pvz., suformuojant tinkamiausių prekių, paslaugų krepšelius, sudarant priderintus, individualizuotus skaitomų straipsnių, klausomos muzikos grojaraščius, rekomenduojamų knygų, filmų, televizijos laidų / kanalų, tinklalaidžių sąrašus. Galima netgi parengti personalizuotus mokymosi, savišvietos ar saviugdros planus, suplanuoti lankytinų vietų (įskaitant atostogų keliones), muziejų, kitų kultūros objektų maršrutus ir t. t. Visa tai gali būti aktualu tiek individualiame, tiek organizaciniame kontekste.

Literatūroje galėtumėte rasti ir dar daugiau bei nuodugnesnių KP uždavinio praktinio taikymo pavyzdžių aprašymų (Loiola ir kt., 2007; Drezner, 2015).

KP uždavinys, kita vertus, yra sudėtingas teorinis-matematinis uždavinys, priklausantis kombinatorinio optimizavimo (angl. *combinatorial optimization*) uždavinių kategorijai. Šiuo atveju tikslas yra surasti optimalų sprendinį, kur sprendiniai yra išreiškiami natūrinių skaičių perstatymais. Optimalumo kriterijus aprašomas (1) formule, o funkcija z yra vadinama tikslo funkcija.

Įrodyta, jog KP uždavinys priklauso NP sunkių optimizavimo uždavinių klasei (Sahni, Gonzalez, 1976). Tai reiškia, kad skaičiavimų laikas, reikalingas optimumui pasiekti, yra susietas su uždavinio apimtimi (n) eksponentine priklausomybe.

Empiriniai kompiuterinių eksperimentų rezultatai

Šiame darbe buvo atlikti empiriniai (praktiniai) kompiuteriniai-eksperimentiniai tyrimai, kurių tikslas – pademonstruoti NPA algoritmo tinkamumą bei pasiekti kiek įmanoma didesnę algoritmo veikimo efektyvumą, išbandant įvairias esminių algoritmo valdančiųjų parametrų (reikšmių) konfigūracijas (kitais tariant, konfigūracines variacijas).

Vykdam eksperimentinius tyrimus, buvo panaudoti atrinkti kvadratinio paskirstymo uždavinio (standartinių / etaloninių) testavimo duomenų pavyzdžiai, pateikti (De Carvalho ir kt., 2006) straipsnyje: b136, b149, b164, b181, b1100, b1121, b1144, c136, c149, c164, c181, c1100, c1121, c1144. Šie testavimo duomenys yra iš konkrečių praktinių taikymų (būtent iš vadinamųjų ribinių ilgių (angl. *border length*) ir indeksų konfliktų (angl. *conflict index*) minimizavimo kontekstų). Jie laikytini vienas iš labai sunkiai sprendžiamų KP uždavinio testinių pavyzdžių; ypač tai pasakytina apie testavimo duomenis b1*. (Šie duomenys yra vieši ir juos gali naudoti pasaulio mokslininkai testuodami ir lygindami savo algoritmus KP uždaviniui.)

Eksperimentuose naudotas 3 GHz taktinio dažnio stacionarus asmeninis kompiuteris. Buvo naudota A. Misevičiaus sudaryto NPA algoritmo programinė realizacija C# programavimo kalba.

Atliekant eksperimentus, NPA algoritmo veikimo efektyvumo įvertinimo ir įvairių jo variantų palyginimo kriterijus yra gaunamų sprendinių kokybės vidutinis santykinis procentinis nuokrypis nuo geriausios žinomos tikslo funkcijos reikšmės (GŽR) (angl. *best known value*). Vidutinis santykinis procentinis nuokrypis ($\bar{\theta}$) apskaičiuojamas pagal formulę: $\bar{\theta} = 100(\bar{z} - z_{G\check{Z}R})/z_{G\check{Z}R}[\%]$, čia $\bar{z}$ yra vidutinė tikslo funkcijos reikšmė, gauta atlikus pakartotinių algoritmo vykdymų, $z_{G\check{Z}R}$ žymi geriausią žinomą tikslo funkcijos reikšmę, atitinkančią galimai (pseudo)optimalų sprendinį.

Empiriniuose eksperimentuose buvo tirti jau minėti NPA algoritmo valdantieji parametrai: 1) NPA algoritmo iteracijų skaičius (N); 2) tabu paieškos iteracijų skaičius (T); 3) (hierarchinės) iteratyviosios tabu paieškos iteracijų skaičius (IT); 4) perturbacijų, esančių NPA algoritmo viduje, stiprumas (μ); 5) perturbacijų, esančių ITP algoritmo viduje, stiprumas (η); 6) tabu paieškos, esančios perturbavimo procedūros viduje, iteracijų skaičius (TP); 7) tabu paieškos, esančios perturbavimo procedūros viduje, (pa)kartojimų skaičius (TK); 8) trijų parametrų integruota visuma (N, T, IT). (Pastaba. Naudotos normalizuotos, t. y. atvaizduotos į intervalą $[0, 1]$, parametrų μ, η reikšmės.)

Empiriniai rezultatai, gauti eksperimentuojant su skirtingomis parametrų reikšmėmis, pateikti 2–9 paveiksluose. Buvo pasirinkta eksperimentuoti su testinių duomenų pavyzdžiu b149, norint pademonstruoti įvairių rezultatų tendencijas. Šiuo atveju buvo eksperimentuota esant trumpalaikiams NPA algoritmo įvykdymams. Paveikslėliuose pateikiami atitinkamų gaunamų sprendinių vidutiniai procentiniai nuokrypiai nuo optimumo

($\bar{\theta}$), kai parametrai kinta tam tikruose diapazonuose. Vaizdumui padidinti parodytos ir rezultatų eigos tendencijų kreivės (angl. *trend lines*). (Pastaba. Turint omenyje multiderinį $\langle N, T, IT \rangle$, buvo tirtos tokios parametų reikšmių kombinacijos: I) $\langle N = 1, T = 1, IT = 1 \rangle$; II) $\langle N = 2, T = 2, IT = 2 \rangle$; III) $\langle N = 3, T = 3, IT = 3 \rangle$; IV) $\langle N = 4, T = 4, IT = 4 \rangle$; V) $\langle N = 5, T = 5, IT = 5 \rangle$; VI) $\langle N = 7, T = 7, IT = 7 \rangle$; VII) $\langle N = 10, T = 10, IT = 10 \rangle$; VIII) $\langle N = 20, T = 20, IT = 20 \rangle$.)

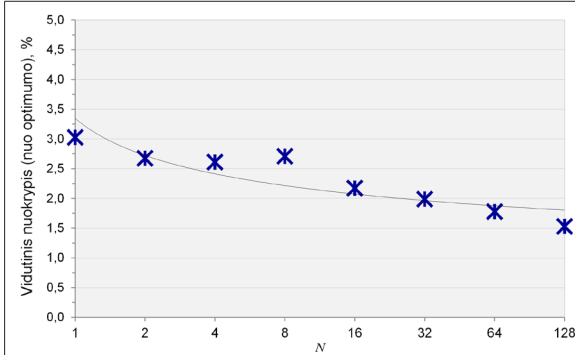
Iš pateiktų grafikų matyti, jog NPA algoritmo rezultatai yra efektyviai pagerinami, tiesiog tik atskirai modifikuojant valdančiųjų parametų reikšmes. Todėl nebereikia kurti naujų (konceptualių) algoritmo patobulinimų ir iš naujo perprogramuoti algoritmo (kas pareikalautų tikrai daug laiko) – pakanka tik algoritmo elgseną lanksčiai valdyti taikant išorinius parametrus pagal vartotojo / tyrėjo poreikius.

2–9 paveiksluose pateiktuose grafikuose galima stebėti nevienodą atskirų parametų poveikį (įtaką) algoritmo rezultatams. Matyti, jog galimai labiausiai rezultatams įtakos turi bendras iteracijų skaičius (N), taip pat iteratyviosios tabu paieškos iteracijų skaičius (IT). Be to, galima stebėti, jog labai gerai algoritmo rezultatus paveikia tam tikros parametų kombinacijos (savotiškos „daugdaros“), kaip antai $\langle N, T, IT \rangle$. Čia galima matyti ne tiek gautą absoliučiai geriausią rezultatą, bet labai išreikštą rezultatų gerėjimo perspektyvinę tendenciją, nors pradinis sprendinys ir nėra pakankamai geras. Tai suponuoja įžvalgą, jog algoritmo našumo didinimas eksperimentuojant su parametų deriniais (multidariniais) yra labai daug žadanti strategija, ir tai galėtų būti toliau tiriama kituose eksperimentuose.

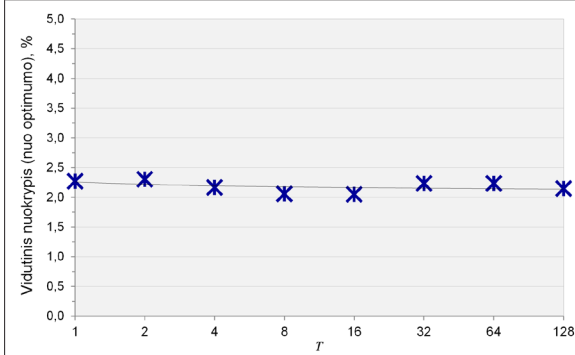
Įdomu, jog perturbavimo laipsnio reikšmių (μ, η) padidinimas nebūtinai pagerina gaunamų sprendinių kokybę. Gali būti, jog šiuo atveju yra subtilesnis dydžių μ, η ir gaunamų sprendinių kokybės priklausomybės pobūdis. Taip pat ir tiriant parametą TP nepastebėta aiškios (determinuotos) rezultatų gerėjimo tendencijos (reikalingi papildomi tyrimai).

Išankstiniai atlikti eksperimentai suteikė galimybes atskleisti „jautriausius“ algoritmo konfigūravimo parametrus, t. y. tuos konkrečius parametrus (parametų grupes), kurių kalibravimas įgalina labiausiai padidinti algoritmo efektyvumą / našumą (t. y. sprendinių kokybę). Būtent, panaudoję tinkamai sukalibruotus parametrus ir / arba jų derinius, mes sugebėjome ilgalaikiuose eksperimentuose stabiliai pasiekti (pseudo)optimalius (galimai optimalius) sprendinius didžiajai daliai tirtų testinių duomenų pavyzdžių.

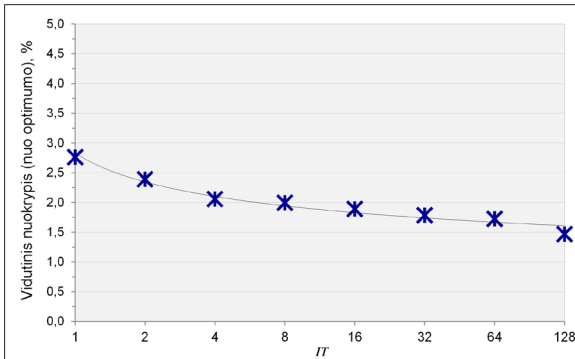
Šių atliktų eksperimentų rezultatai pateikti 1 lentelėje. Lentelėje parodytos gautų sprendinių vidutinių nuokrypių nuo (pseudo)optimalių sprendinių reikšmės ($\bar{\theta}$). Matyti, jog testavimo duomenų pavyzdžiams b136, b149, b164, b181, ci36, ci49, ci64, ci81, ci100, ci121 visuose bandymuose yra pasiekti (pseudo)optimalūs sprendiniai. Kitiems testavimo duomenims (b1100, b1121, b1144, ci144) vidutinis nuokrypis neviršija 0,2 %. Bendras suminis nuokrypis yra lygus 0,029 %. Tai tik patvirtina tikrai labai aukštą realų NPA algoritmo efektyvumo laipsnį, taip pat jame glūdintį potencialą.



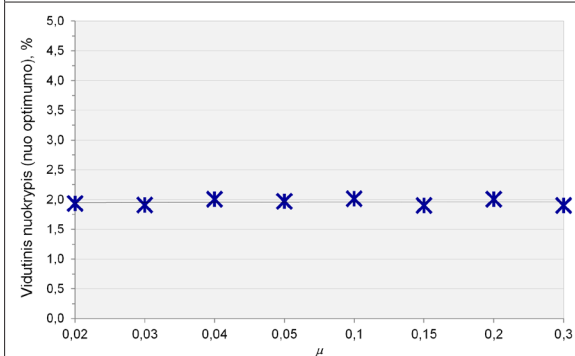
2 pav. Parametro N tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)



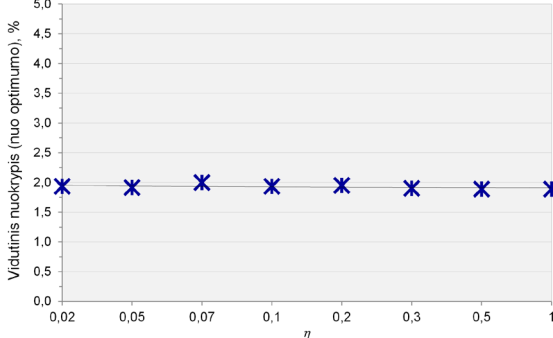
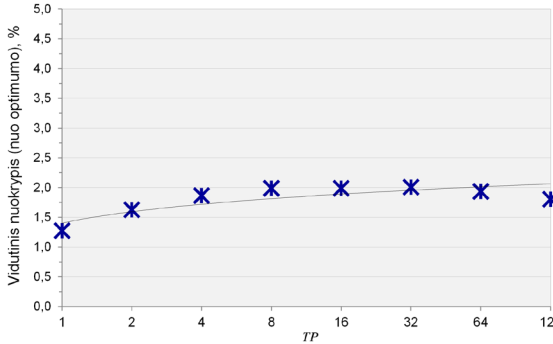
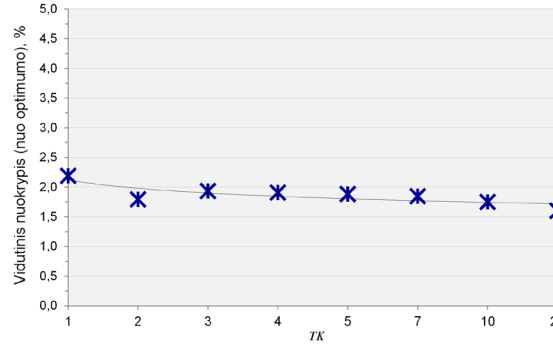
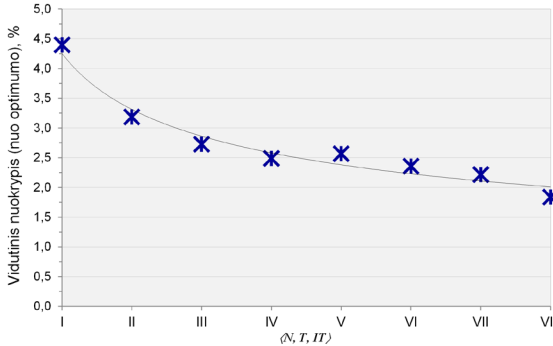
3 pav. Parametro T tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)



4 pav. Parametro IT tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)



5 pav. Parametro μ tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)

	6 pav. Parametro η tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)
	7 pav. Parametro TP tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)
	8 pav. Parametro TK tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)
	9 pav. Parametrų darinio $\langle N, T, IT \rangle$ tyrimo rezultatai (esant kitoms sąlygoms vienodoms ir esant trumpalaikiams algoritmo įvykdymams)

1 lentelė. NPA algoritmo rezultatai testavimo duomenų pavyzdžiams b1*, ci* (esant ekstensyviems, ilgalaikiams algoritmo įvykdymams)

b1*				ci*			
Pavyzdys	GŽR		Vykdymo laikas (sek.)	Pavyzdys	GŽR		Vykdymo laikas (sek.)
b136	3296	0,000	7,550	ci36	168611971	0,000	1,073
b149	4548	0,000	635,900	ci49	236355034	0,000	4,432
b164	5988	0,000	1640,000	ci64	325671035	0,000	45,220
b181	7532	0,000	49460,000	ci81	427447820	0,000	233,700
b1100	9248	0,090	75730,000	ci100	523146366	0,000	4367,000
b1121	11392	0,122	130300,000	ci121	653409588	0,000	116500,000
b1144	13428	0,198	187200,000	ci144	794811636	0,002	199100,000

Pastabos. 1. GŽR testinių duomenų pavyzdžiams b136, b149, b164 yra iš (Drezner, 2008; Drezner, Marcoulides, 2009) straipsnių. 2. GŽR testinių duomenų pavyzdžiui b181 yra iš (Drezner, Misevičius, 2013) straipsnio. 3. GŽR testinių duomenų pavyzdžiams b1100, b1121, b1144 yra iš (Misevičius ir kt., 2024) straipsnio. 4. GŽR testinių duomenų pavyzdžiams ci36, ci49, ci64, ci81, ci100 yra iš (Drezner, Marcoulides, 2009) straipsnio. 5. GŽR testinių duomenų pavyzdžiams ci121, ci144 yra iš (Drezner, Misevičius, 2013) straipsnio. 6. Yra pateikiamas vieno algoritmo įvykdymo vidutinis laikas.

Baigiamosios pastabos

Šiame straipsnyje pateiktas vadinamojo nusileidimo–pakilimo algoritmo principas ir veikimas. Pagrindinis dėmesys skirtas empiriniams NPA algoritmo tyrimams, siekiant identifikuoti efektyviausius algoritmo konfigūracinius variantus / variacijas.

Tyrimams panaudotas gerai žinomas kvadratinio paskirstymo uždavinys, ir buvo tirti skirtingi NPA algoritmo parametrų „konfigūravimo“ scenarijai bei jų įtaka algoritmo efektyvumui.

Apibendrintai tariant, atlikti empiriniai eksperimentiniai tyrimai panaudojant sudėtingus KP uždavinio testavimo pavyzdžius rodo, jog NPA algoritmas (kai jis yra tinkamai sukalibruotas) leidžia gauti labai aukštos kokybės sprendinius. Tą patvirtina rezultatai, gauti intensyviai eksperimentuojant su testinių duomenų pavyzdžiais b1*, ci*. Didžiąją dalį šių duomenų yra stabiliai, nuosekliai gaunami (pseudo)optimalūs sprendiniai.

Taip pat gauti empiriniai rezultatai indikuoja, jog gerai pasiteisino sprendimas nagrinėti ne atskirus parametrus, bet parametrų kombinacijas, t. y. multidarinius, kuomet algoritmo našumą siekiama padidinti išbandant ir kalibruojant įvairišką algoritmo parametrų „daugdarą“.

Būtent parametrų multidarinių tolesnei analizei turėtų būti skirtas pagrindinis dėmesys ateities eksperimentiniuose tyrimuose. Tai galėtų būti formuluojama kaip multidimensio globaliojo optimizavimo uždavinys, kurio sprendimui galima pasitelkti ir kolektyvinio intelekto metodus.

Autorių indėlis

Alfonsas Misevičius: konceptualizacija, metodologija, formalioji analizė, tyrimas, straipsnio rašymas – pirminis parengimas.

Gintaras Palubeckis: konceptualizacija, metodologija, formalioji analizė.

Dovilė Verenė: straipsnio rašymas – peržiūra, redagavimas, vizualizacija.

Literatūra

- Aarts, E., & Lenstra, J. K. (Eds.). (1997). *Local Search in Combinatorial Optimization*. Wiley.
- Abualigah, L. (Ed.). (2024). *Metaheuristic Optimization Algorithms: Optimizers, Analysis, and Applications*. Morgan Kaufmann.
- Ahuja, R. K., Orlin, J. B., & Sharma, D. (2000). Very Large-Scale Neighborhood Search. *International Transactions in Operational Research*, 7(4-5), 301–317. <https://doi.org/10.1111/j.1475-3995.2000.tb00201.x>
- Amaral, H. F., Urrutia, S., & Hvattum, L. M. (2021). Delayed Improvement Local Search. *Journal of Heuristics*, 27, 923–950. <https://doi.org/10.1007/s10732-021-09479-9>
- Amodeo, L., Talbi, E.-G., & Farouk, Y. (2018). *Recent Developments in Metaheuristics (Operations Research/Computer Science Interfaces Series)*. Springer International Publishing AG.
- Benlic, U., & Hao, J.-K. (2013). Breakout Local Search for the Quadratic Assignment Problem. *Applied Mathematics and Computation*, 219(9), 4800–4815. <https://doi.org/10.1016/j.amc.2012.10.106>
- Boettcher, S., & Percus, A. G. (1999). Extremal Optimization: Methods Derived From Co-evolution. In W. Banzhaf, J. M. Daida, A. E. Eiben, M. H. Garzon, & V. Honavar (Eds.), *GECCO'99: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation (Orlando, FL, USA)* (Vol. 1, pp. 825–832). Morgan Kaufmann.
- Bonabeau, E., Theraulaz, G., & Dorigo, M. (1999). *Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies on the Sciences of Complexity)* (1st ed.). Oxford University Press.
- Bozorg-Haddad, O., Solgi, M., & Loáiciga, H. A. (2017). *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization (Wiley Series in Operations Research and Management Science)*. John Wiley & Sons.
- Brownlee, J. (2011). *Clever Algorithms. Nature-Inspired Programming Recipes*. Lulu.com.
- Burke, E. K., & Bykov, Y. (2017). The Late Acceptance Hill-Climbing Heuristic. *European Journal of Operational Research*, 258(1), 70–78. <https://doi.org/10.1016/j.ejor.2016.07.012>
- Bykov, Y., & Petrovic, S. (2016). A step counting hill climbing algorithm applied to university examination timetabling. *Journal of Scheduling*, 19, 479–492. <https://doi.org/10.1007/s10951-016-0469-x>
- Čela, E. (1998). *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer.
- De Carvalho Jr., S. A., & Rahmann, S. (2006). Microarray Layout as a Quadratic Assignment Problem. In D. Huson, O. Kohlbacher, A. Lupas, K. Nieselt, & A. Zell (Eds.), *German Conference on Bioinformatics, GCB 2006, Lecture Notes in Informatics – Proceedings* (Vol. P-83, pp. 11–20). Gesellschaft für Informatik.
- Drezner, Z. (2008). Tabu Search and Hybrid Genetic Algorithms for Quadratic Assignment Problems. In W. Jaziri (Ed.), *Tabu Search* (pp. 89–108). In-Tech. <https://doi.org/10.5772/5595>
- Drezner, Z. (2015). The Quadratic Assignment Problem. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location Science* (pp. 345–363). Springer. https://doi.org/10.1007/978-3-319-13111-5_13

- Drezner, Z., & Marcoulides, G. A. (2009). On the Range of Tabu Tenure in Solving Quadratic Assignment Problems. In P. Petratos & G. A. Marcoulides (Eds.), *Recent Advances in Computing and Management Information Systems* (pp. 157–168). Athens Institute for Education and Research.
- Drezner, Z., & Misevičius, A. (2013). Enhancing the Performance of Hybrid Genetic Algorithms by Differential Improvement. *Computers & Operations Research*, 40(4), 1038–1046. <https://doi.org/10.1016/j.cor.2012.10.014>
- Du, D.-Z., Pardalos, P. M., Hu, X., & Wu, W. (2022). *Introduction to Combinatorial Optimization (Springer Optimization and Its Applications, 196)* (1st ed.). Springer Nature Switzerland AG.
- Edelkamp, S., & Schrödl, S. (2012). *Heuristic Search. Theory and Applications*. Morgan Kaufmann Publishers.
- Gendreau, M., & Potvin, J.-Y. (Eds.). (2018). *Handbook of Metaheuristics (International Series in Operations Research & Management Science 272)* (3rd ed.). Springer.
- Goudet, O., Amri Sakhri, M. S., Goëffon, A., & Saubion, F. (2024). Emergence of New Local Search Algorithms with Neuro-Evolution. In T. Stützle & M. Wagner (Eds.), *Evolutionary Computation in Combinatorial Optimization. EvoCOP 2024. Lecture Notes in Computer Science* (Vol. 14632, pp. 33–48). Springer. https://doi.org/10.1007/978-3-031-57712-3_3
- Hu, T.-C., Kahng, A. B., & Tsao, C.-W. A. (1995). Old Bachelor Acceptance: A New Class of Non-Monotone Threshold Accepting Methods. *ORSA Journal on Computing*, 7(4), 417–425. <https://doi.org/10.1287/ijoc.7.4.417>
- Hussin, M. S., & Stützle, T. (2009). Hierarchical Iterated Local Search for the Quadratic Assignment Problem. In M. J. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, & A. Schaerf (Eds.), *Hybrid Metaheuristics, HM 2009, Lecture Notes in Computer Science* (Vol. 5818, pp. 115–129). Springer.
- Khadem, M., & Eshlaghy, A. T., & Hafshejani, K. F. (2023). Nature-inspired metaheuristic algorithms: literature review and presenting a novel classification. *Journal of Applied Research on Industrial Engineering*, 10(2), 286–339. <https://doi.org/10.22105/jarie.2021.287733.1330>
- Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for Optimization*. The MIT Press.
- Koopmans, T. C., & Beckmann, M. (1957). Assignment Problems and the Location of Economic Activities. *Econometrica*, 25(1), 53–76.
- Li, Y., Pardalos, P. M., & Resende, M. G. C. (1994). A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem. In P. M. Pardalos & H. Wolkowicz (Eds.), *Quadratic Assignment and Related Problems. DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (Vol. 16, pp. 237–261). AMS.
- Lin, S., & Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*, 21(2), 498–516. <https://doi.org/10.1287/opre.21.2.498>
- Loiola, E. M., De Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., & Querido, T. (2007). A Survey for the Quadratic Assignment Problem. *European Journal of Operational Research*, 176(2), 657–690. <https://doi.org/10.1016/j.ejor.2005.09.032>
- Lourenco, H. R., Martin, O. C., & Stützle, T. (2002). Iterated Local Search. In F. Glover & G. Kochenberger (Eds.), *Handbook of Metaheuristics* (pp. 321–353). Kluwer.
- Martí, R., Pardalos, P. M., & Resende, M. G. C. (Eds.). (2018). *Handbook of Heuristics*. Springer.
- Misevičius, A. (2000). An Intensive Search Algorithm for the Quadratic Assignment Problem. *Informatica*, 11(2), 145–162. <https://doi.org/10.3233/INF-2000-11204>
- Misevičius, A., Andrejevas, A., Ostreika, A., Verenė, D., & Žekienė, G. (2024). An Improved Hybrid Genetic-Hierarchical Algorithm for the Quadratic Assignment Problem. *Mathematics*, 12(23), Article 3726. <https://doi.org/10.3390/math12233726>

Misevičius, A., Blonskis, J., & Bukšnaitis, V. (2011). Bičių spiečių imitavimas sprendžiant optimizavimo uždavinius. *Informacijos mokslai*, 56, 163–173. <https://doi.org/10.15388/Im.2011.0.3140>

Misevičius, A., & Kuznecovaitė, D. (2019). Dviejų lygių iteracinis tabu paieškos algoritmas kvadratinio paskirstymo uždaviniui. *Informacijos mokslai*, 85, 115–134. <https://doi.org/10.15388/Im.2019.85.19>

Misevičius, A., Palubeckis, G., Verenė, D., & Žekienė, G. (2025). A Descent-Ascent Principle Based Algorithm for the Quadratic Assignment Problem [Accepted for publication]. *Information and Software Technologies, 31st International Conference, ICIST 2025, Proceedings, Communications in Computer and Information Science (CCIS)*. Springer.

Salhi, S. (2017). *Heuristic Search. The Emerging Science of Problem Solving*. Palgrave Macmillan-Springer.

Sahni, S., & Gonzalez, T. (1976). P-complete Approximation Problems. *Journal of the ACM*, 23(3), 555–565. <https://doi.org/10.1145/321958.321975>

Siarry, P. (Ed.). (2016). *Metaheuristics*. Springer.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation (Wiley Series on Parallel and Distributed Computing)*. Wiley.

Tsai, C.-W., & Chiang, M.-C. (2023). *Handbook of Metaheuristic Algorithms: From Fundamental Theories to Advanced Applications (Uncertainty, Computational Techniques, and Decision Intelligence)*. Academic Press.

Voudouris, C., & Tsang, E. (1999). Guided Local Search and its Application to the Traveling Salesman Problem. *European Journal of Operational Research*, 113(2), 469–499. [https://doi.org/10.1016/S0377-2217\(98\)00099-X](https://doi.org/10.1016/S0377-2217(98)00099-X)

Wang, G.-G., Zhao, X., & Li, K. (2024). *Metaheuristic Algorithms: Theory and Practice* (1st ed.). CRC Press.

Yang, X.-S. (2010a). *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley.

Yang, X.-S. (2010b). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.