Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Keliaujančio prekeivio uždavinių generavimas ir jų sprendimo algoritmų tyrimas

Baigiamasis magistro projektas

Jokūbas Šakys

Projekto autorius

Asist. dr. Kęstas Rimkus

Vadovas

Kaunas, 2026

Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Keliaujančio prekeivio uždavinių generavimas ir jų sprendimo algoritmų tyrimas

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

Jokūbas Šakys

Projekto autorius

Asist. dr. Kęstas Rimkus

Vadovas

Asist. dr. Andrius Knyš

Recenzentas

Kaunas, 2026

Kauno technologijos universitetas

Elektros ir elektronikos fakultetas

Jokūbas Šakys

Keliaujančio prekeivio uždavinių generavimas ir jų sprendimo algoritmų tyrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Jokūbas Šakys

Patvirtinta elektroniniu būdu

Šakys, Jokūbas. Keliaujančio prekeivio uždavinių generavimas ir jų sprendimo algoritmų tyrimas. Magistro baigiamasis projektas / vadovas asist. dr. Kęstas Rimkus; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): elektronikos inžinerija, inžinerijos mokslai.

Reikšminiai žodžiai: optimizavimas, maršrutų optimizavimas, sintetinių duomenų generavimas, keliaujančio prekeivio problema, genetinis algoritmas, skruzdžių kolonijos optimizavimas, Lin-Kernighan-Helsgaun, Google OR-Tools.

Kaunas, 2026. 48 p.

Santrauka

Šio darbo metu buvo sukurtas keliaujančio prekeivio problemos sintetinių duomenų rinkinių su žinomu optimaliu sprendimu generavimo įrankis bei sprendimo algoritmų testavimo sistema. Naudojantis šiais įrankiais buvo tiriami keturi keliaujančio prekeivio problemos sprendimo algoritmai: genetinis algoritmas, skruzdžių kolonijos optimizavimo algoritmas, *Lin-Kernighan-Helsgaun* algoritmas, *Google OR-Tools* algoritmas. Visų pirma atlikta literatūros analizė, kuri buvo skirta surinkti informaciją apie keliaujančio prekeivio problemos sprendimo algoritmus, jų veikimo principus, privalumus ir trūkumus. Toliau buvo pasirinkti 4 metodai minimi pastarųjų metų literatūroje kaip tinkamiausi dirbant su dideliais duomenų rinkiniais. Eksperimentinė dalis buvo pradėta pasinaudojant duomenų generavimo įrankiu ir sukuriant 120 duomenų rinkinių su skirtingu taškų skaičiumi (100, 1000, 10000) bei masteliu (nuo miesto iki žemyno). Tuomet buvo atlikti 480 eksperimentų (bendras laikas – 9 h 53 min 14 s) su pradiniais parametrais, kurie rasti literatūroje – *OR-Tools* rado daugiausiai optimalių sprendimų (92,50 %), tačiau buvo lėčiausias (vidutiniškai 118,8 s vienam eksperimentui), o LKH metodas pasižymėjo geriausiu kokybės (77,50 %) bei greičio (45,9 s) santykiu. ACO metodas tikslumo prasme buvo artimas LKH (75,80 %), tačiau šiek tiek lėtesnis (68,7 s), o genetinis algoritmas su 1000 ir 10000 taškų duomenų rinkiniais pasirodė itin blogai dėl laiko limito (46,7 % optimalių sprendimų ir vidutiniškai 63,3 s vienam eksperimentui). Vėliau buvo atliktas metodų stabilumo tyrimas atliekant 480 eksperimentų (12 duomenų rinkinių po 10 pakartojimų visiems 4 metodams, bendras laikas – 7 h 34 min 21 s) su tais pačiais parametrais, kuris parodė, jog atsitiktinio skaičiaus pradinė vertė neturi reikšmingos įtakos sprendimo paieškos kokybei bei laikui.

Šakys Jokūbas Generation of Traveling Salesman Problem and Research of Algorithms for Solving It. Master's Final Degree Project / supervisor asst. dr. Kęstas Rimkus; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): electronics engineering, engineering science.

Keywords: optimisation, route optimisation, synthetic data generation, Travelling Salesman Problem, Genetic Algorithm, Ant Colony Optimisation, Lin-Kernighan-Helsgaun, Google OR-Tools.

Kaunas, 2026. 48 p.

Summary

During this study, a synthetic dataset generation tool for the *Travelling Salesman Problem* with known optimal solutions, as well as a benchmarking system for evaluating solution algorithms, was developed. Using these tools, four algorithms for solving the *Travelling Salesman Problem* were investigated: the *Genetic Algorithm*, *Ant Colony Optimization* algorithm, *Lin–Kernighan–Helsgaun* algorithm, and *Google OR-Tools* algorithm. First, a literature review was conducted to gather information about *Travelling Salesman Problem* solution algorithms, including their operating principles, advantages, and disadvantages. Subsequently, four methods frequently identified in recent literature as the most suitable for handling large-scale datasets were selected for further analysis. The experimental phase began with the use of the dataset generation tool to create 120 datasets with varying numbers of nodes (100, 1000, and 10000) and different spatial scales (from city-level to continent-level). Afterwards, 480 experiments were carried out (total execution time – 9 h 53 min 14 s) using the initial parameter configurations identified in the literature. The *OR-Tools* method found the highest number of optimal solutions (92.50 %) but was also the slowest, requiring an average of 118.8 s per experiment. Meanwhile, the LKH method demonstrated the best balance between solution quality (77.50 %) and execution speed (45.9 s). The ACO method achieved accuracy results close to LKH (75.80 %) but was slightly slower (68.7 s). The Genetic Algorithm performed particularly poorly on datasets containing 1000 and 10000 nodes due to the imposed time limit, achieving only 46.70 % optimal solutions with an average execution time of 63.3 s per experiment. Finally, a stability analysis of the methods was performed by conducting 480 additional experiments (12 datasets with 10 repetitions for all 4 methods, total execution time – 7 h 34 min 21 s) using the same parameter settings. The results demonstrated that the initial random seed had no significant impact on either the solution quality or the execution time.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Santrumpos:

TSP – keliaujančio prekeivio problema (angl. *Travelling Salesman Problem*);

GA – genetinis algoritmas (angl. *Genetic Algorithm*);

ACO – skruzdžių kolonijos optimizavimas (angl. *Ant Colony Optimization*);

LKH – Lin-Kernighan-Helsgaun algoritmas;

CPU – centrinis procesorius (angl. *central processing unit*);

GPU – grafinis procesorius (angl. *graphics processing unit*);

Terminai:

Atsitiktinis skaičius – pradinė reikšmė, naudojama skaičių generatoriui inicializuoti.

Hamiltono ciklas – uždaras maršrutas, kurio kiekviena viršūnė aplankoma tik vieną kartą ir grįžtama į pradinį tašką.

Heuristika – metodas, skirtas greitai rasti pakankamai gerą problemos sprendimą, tačiau negarantuojantis optimalaus rezultato.

Metaheuristika – aukštesnio lygio optimizavimo strategija, naudojama sudėtingų problemų sprendinių paieškai didelėje sprendimų erdvėje.

Optimizavimas – procesas, kurio metu ieškoma geriausio galimo sprendimo pagal pasirinktus kriterijus.

Lokalus minimumas – sprendinys, kuris yra geresnis už artimus sprendinius, tačiau nėra geriausias visoje sprendimų erdvėje.

Konvergencija – algoritmo artėjimas prie stabilaus arba optimalaus sprendimo.

Atstumų matrica – duomenų struktūra, kurioje saugomi atstumai tarp visų taškų porų.

Iteracija – vienas algoritmo vykdymo ciklas, kurio metu atliekami sprendinio gerinimo veiksmai.

Sintetiniai duomenys – dirbtinai sugeneruoti duomenys, naudojami algoritmų testavimui.

Įvadas

Technologijų pažanga skatina procesų spartėjimą ir efektyvumo augimą. Nepaisant įvairių tvarumo iniciatyvų, vartojimas nuolat didėja. Tai gamintojams, paslaugų tiekėjams atneša ne tik pelną, tačiau ir iššūkius, susijusius su logistika: sandėliavimu, transportavimu. Norint išlikti rinkoje turi pagaminti bei pristatyti arba suteikti paslaugą greičiau, tiksliau ir kokybiškiau. Tam tenka optimizuoti savo veiklą. Vienas iš pavyzdžių – keliaujančio prekeivio problemos (angl. *Travelling Salesman Problem*, TSP) sprendimo paieška. Tai plačiai žinoma kombinatorinės optimizacijos problema. Jos tikslas yra rasti minimalų maršrutą, kurio metu prekeivis aplanko kiekvieną tašką po vieną kartą ir grįžta į pradinį tašką [1, 2]. Ši problema gali būti plačiai pritaikoma mikroschemų projektavime, duomenų tinklais perdavime, tiekimo grandinių planavime – maršrutų planavime [3, 4]. Nors panaudojimo galimybių yra kur kas daugiau, šiame darbe bus išsamiau panagrinėta pastaroji – maršrutų planavimas. Norint paslaugas ar prekes pristatyti, suteikti kuo greičiau tenka optimizuoti pristatymo maršrutą. Nepaisant, atrodytų, paprasto aprašymo, TSP yra sudėtinga problema [5], todėl priskiriama NP-sunkių (angl. *NP-hard*) problemų klasei, kas reiškia, jog šiuo metu nėra žinomas joks algoritmas, galintis polinominio sudėtingumo ribose surasti optimalų sprendimą, kuris tiktų visiems atvejams. Nors mažam taškų skaičiui problema gali būti išspręsta išbandant visas įmanomas kombinacijas, didesniems duomenų masyvams toks sprendimo būdas tampa neefektyvus. Praktikoje dažniausiai taikomi kiti metodai, kurių tyrimas ir bus atliekamas šiame darbe.

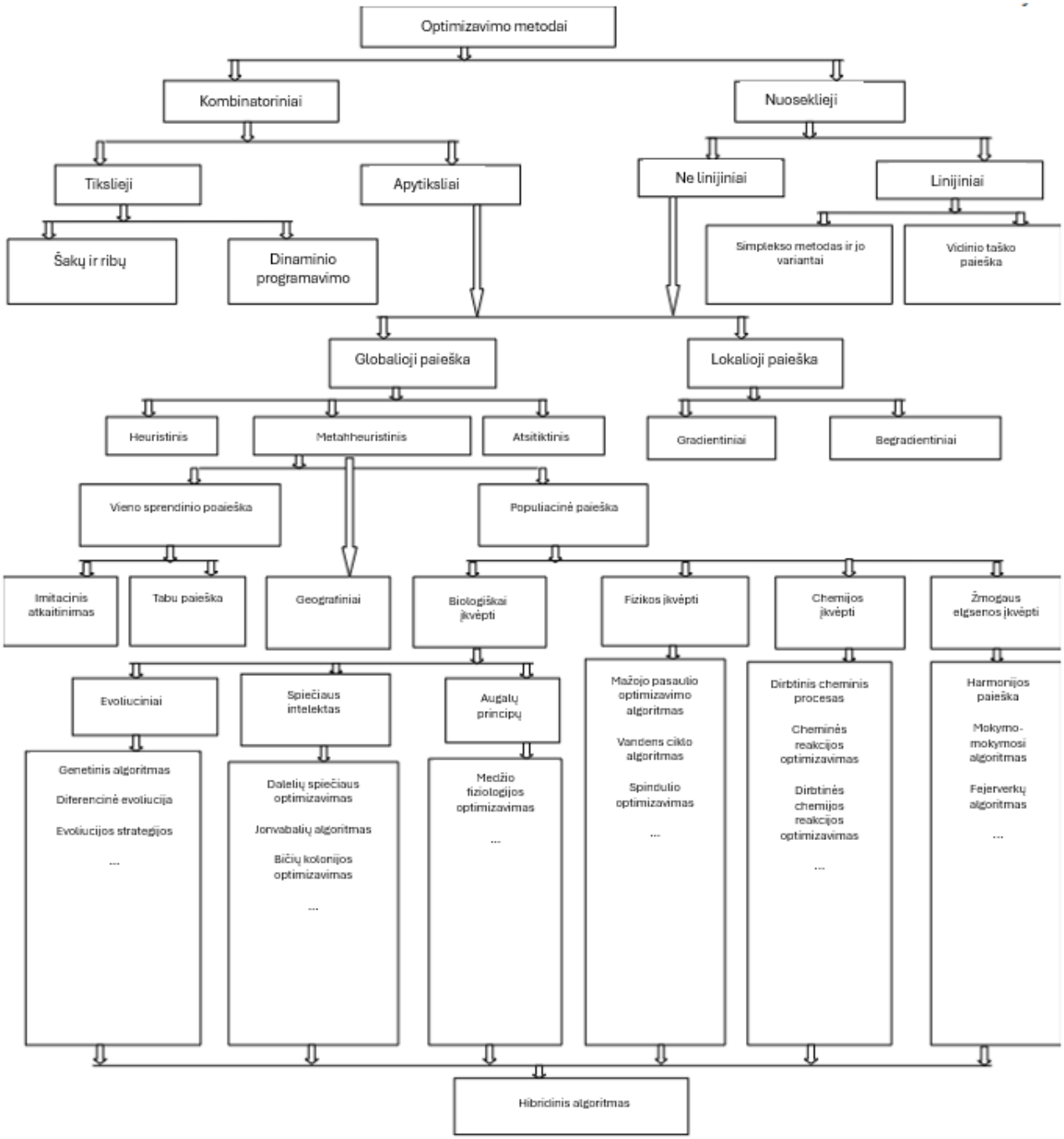
Darbo tikslas – keliaujančio prekeivio uždavinių generavimas su žinomu optimaliu sprendimu ir jų sprendimo algoritmų tyrimas.

Išsikelti uždaviniai:

1. sukurti keliaujančio prekeivio uždavinių generavimo įrankį, leidžiantį generuoti duomenų rinkinius su žinomu optimaliu sprendimu bei sprendimo algoritmų testavimo sistemą ir sugeneruoti daugiau nei 100 sintetinių duomenų rinkinių su skirtingais dydžiais ir masteliais;
2. atlikti bent 3 keliaujančio prekeivio problemos sprendimo algoritmų tyrimą su literatūroje rastais pradiniais tų sprendimo algoritmų parametrais ir palyginti juos tarpusavyje;
3. atlikti bent 3 keliaujančio prekeivio problemos sprendimo algoritmų stabilumo tyrimą ir palyginti juos tarpusavyje.

1. Literatūros apžvalga

Šiame skyriuje bus apžvelgti kai kurie metodai naudojami spręsti optimizavimo uždavinius. Toliau galima matyti bendrą algoritmų ir metodų klasifikavimą (1 pav.).



1 pav. Optimizavimo metodų klasifikacija [6]

Egzistuoja ir daugiau metodų, kurie šiame paveiksle (1 pav.) paminėti nėra, tačiau jų visų apžvelgti nėra įmanoma. Remiantis naujausiais šaltiniais ir surinkta informacija, bus apžvelgti metodai, kurie yra arba naudojami kaip atskaitos taškas (angl. *ground truth*) arba yra dažniausiai naudojami keliaujančio prekeivio užduočiai spręsti.

Pasirinkti buvo šie metodai:

I. Tikslieji (angl. *exact methods*):
a. Grubiosios jėgos algoritmas (angl. *Brute Force*);
b. Šakų ir ribų algoritmas (angl. *Branch & Bound*);

II. Heuristiniai (angl. *heuristic*):
a. Artimiausio kaimyno algoritmas (angl. *Nearest Neighbour*);
b. Pigiausio įterpimo algoritmas (angl. *Cheapest Insertion*);

III. Metaheuristiniai (angl. *metaheuristic*):
a. Gamtos įkvėpti (angl. *Bio-inspired*):
1. Genetinis algoritmas (angl. *Genetic algorithm*);
2. Skruzdžių kolonijos optimizavimo algoritmas (angl. *Ant Colony Optimization*);
b. Neuroniniais tinklais grįstų (angl. *Neural Network Based*):
1. Grafų neuroninis tinklas (angl. *Graph Neural Network*);
2. Rodyklių tinklai (angl. *Pointer Networks*).

1.1. Tikslieji metodai (angl. *exact methods*)

Tai optimizavimo metodai, kurie garantuoja optimalaus sprendimo suradimą, jeigu turime pakankamai laiko ir skaičiavimo resursų. Šie metodai gali būti naudojami su nedideliu taškų skaičiumi ir puikiai tinka patikrinti, ar apytiksliai metodai grąžina panašius rezultatus su mažu taškų skaičiumi, prieš generuojant didelių taškų skaičiaus sprendimus.

1.1.1. Grubiosios jėgos algoritmas (angl. *Brute Force*, BF)

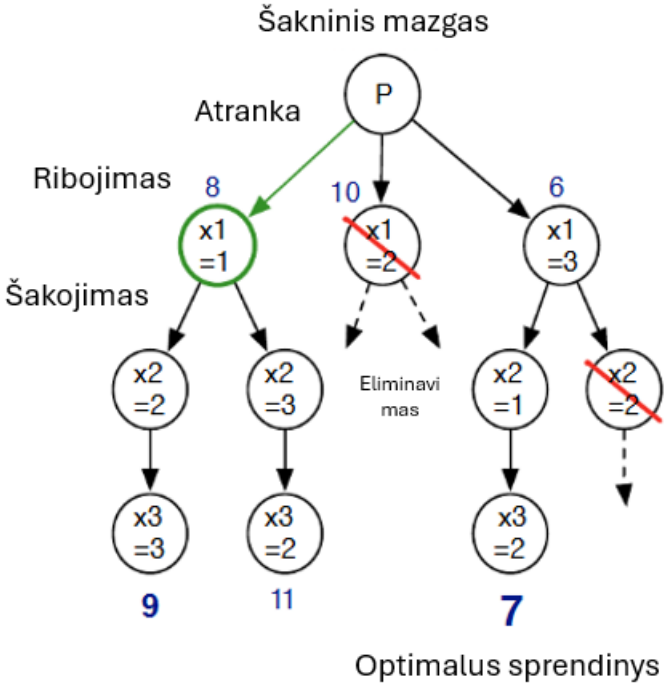
Grubiosios jėgos algoritmas yra vienas iš pagrindinių metodų, naudojamų optimizavimo problemoms spręsti [7], ne išimtis ir keliaujančio prekeivio problema. Šio metodo veikimo principas – kiekvieno galimo sprendimo tikrinimas [1]. Tai reiškia, jog algoritmas apskaičiuoja atstumą kiekvienai galimai taškų kombinacijai. Devynių taškų aplankymo atveju grubiosios jėgos metodas turėtų apskaičiuoti 40 320 galimų maršrutų bendrą atstumą. Todėl, dėl faktorialu didėjančio kombinacijų skaičiaus augant taškų skaičiui šis sprendimo metodas nors ir tikslus, tačiau labai lėtas – tinkamas tik mažai taškų apimančių problemų sprendimų paieškai.

Atliekant tyrimą su devyniais prekybos centrais Bandar Lampung mieste [1] grubiosios jėgos algoritmas gavo optimalų sprendimą 32,80 km. Kiti du lyginti metodai: artimiausio kaimyno bei pigiausio įterpimo rado tik artimus optimaliam sprendimus (atitinkamai 33,20 km ir 33,35 km). Grubiosios jėgos algoritmas dar geba veikti su devyniais taškais, tačiau padidinus bent vienu tašku šią aibę, skaičiavimo resursų reikėtų beveik 10 kartų daugiau – 362 880 galimos kombinacijos. Taigi, jei tokia sąlyginai nedidelė paklaida (atitinkamai 1,22 % ir 1,68 %) nuo optimalaus sprendimo tenkina, tuomet tikrai verčiau būtų rinktis kurį nors iš anksčiau paminėtų heuristinių metodų.

1.1.2. Šakų ir ribų algoritmas (angl. *Branch & Bound*, BB)

Šakų ir ribų algoritmas yra dar vienas tikslusis metodas, naudojamas optimizavimo problemoms spręsti [7]. Tai jau labiau sistemiškas sprendimo ieškojimas nei prieš tai aprašyto grubiosios jėgos, kur tikrinamos visos įmanomos kombinacijos. Šiuo atveju, geriausias sprendimas yra ieškomas

kartojant tris veiksmus: išsišakojimas (angl. *branching*), ribojimas (angl. *delimiting*), genėjimas (angl. *pruning*) [8]. Išsišakojimas – visų galimų sprendimų padalijimas į mažesnius poaibius. Ribojimas – apatinės ribos (angl. *lower-bound*, LB) bei su kai kuriomis metodo atmainomis viršutinės ribos (angl. *upper-bound*, UB) skaičiavimas kiekvienam naujam taškui, po to kai jie yra surandami. Genėjimas – šakos atmetimas (angl. *elimination*), jei prisijungus tašką ir paskaičiavus apatinę ribą vertė yra didesnė nei jau esantis tarpinis optimalus sprendimas. Tokiu būdu, siaurėja sprendimo paieškos laukas ir mažėja įmanomų, tikrinamų kombinacijų skaičius. Nors šis metodas ir negarantuoja, jog visuomet bus rastas optimalus sprendimas, tačiau vis tiek laikomas tiesiniu metodu, kadangi jį nesunkiai galima tokiu paversti: išbandyti visus pradinius taškus ar nustatyti, jog genėjant būtų palikta ne viena, bet, pavyzdžiui, dvi geriausios šakos ir tolimesnius skaičiavimus atlikti su jomis (2 pav.).



2 pav. Nuosekliojo šakų ir ribų algoritmo principinė schema [6]

Šie trys veiksmai yra kartojami tol, kol nebelieka nepasirinktų taškų. Tuomet iš paskutinio taško yra grįžtama į pradinį tašką taip uždarant pilną Hamiltono ciklą. Gautas bendras atstumas ir yra maršruto ilgis. Sprendžiant optimizavimo uždavinį šiuo metodu egzistuoja galimybė skaičiavimus atlikti paraleliai, skirtinguose centrinio procesoriaus branduoliuose (angl. *multi-core CPUs*) [8] taip jį pagreitinant ar išplečiant ir priartinant prie grubiosios jėgos metodo.

Viename iš cituojamų šaltinių [8], šakų ir ribų algoritmas buvo lyginimas su kitais trejais: gobšus algoritmas, dinaminio programavimo algoritmas, genetinis algoritmas. Kadangi vėlesniuose skyriuose bus aprašomas tik genetinis algoritmas, tad ir čia lyginama tik su juo. Remiantis rezultatais, padavus tris skirtingus rinkinius po 10 taškų (miestų), visuose trijuose rinkiniuose abu algoritmai sugebėjo surasti optimalų sprendimą, tačiau skyrėsi sprendimo paieškos laikas. Genetinis algoritmas visus kartus sugebėjo sprendimą rasti greičiau. Ribų ir šakų metodas užtruko: 2,77 s, 13,16 s, 1,52 s, kur genetinis algoritmas atitinkamai: 0,025 s, 0,021 s, 0,032 s. Galima daryti išvadą, jog, nors šakų ir ribų metodas su užduotimi susitvarkė taip pat gerai (100 % optimalaus sprendimo radimas), tačiau jis buvo imlesnis laikui – keliasdešimt kartų ilgiau truko nei genetinis algoritmas.

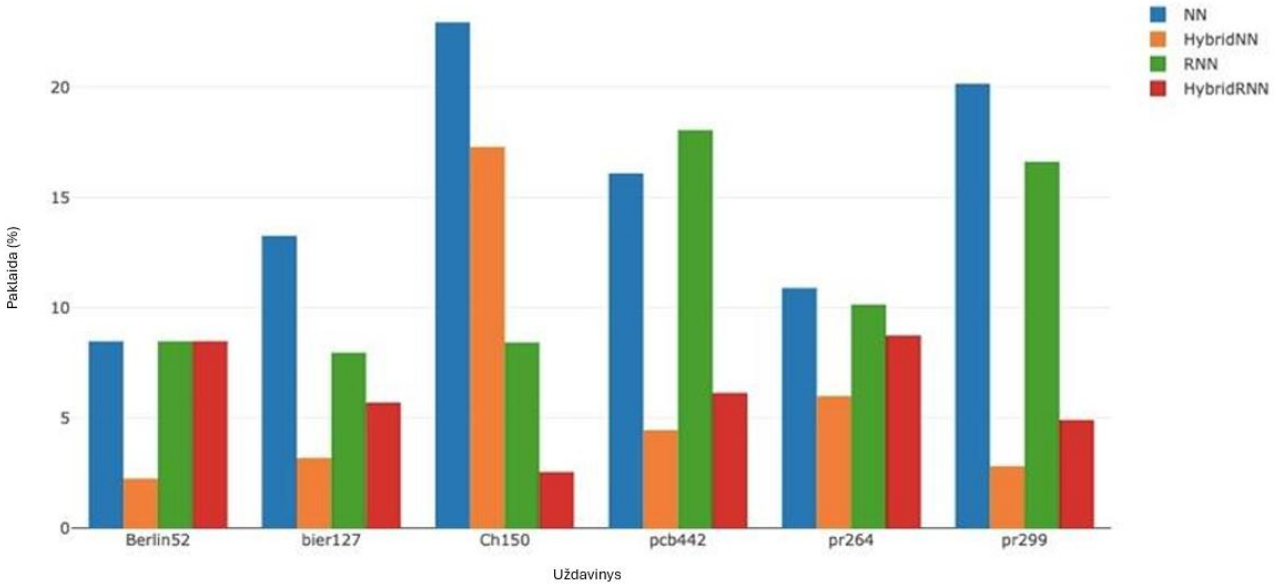
1.2. Heuristiniai metodai (angl. *heuristic methods*)

Tai optimizavimo metodai, kurie negarantuoja optimalaus sprendimo, bet randa artimą jam per pakankamai mažą arba nustatytą laiką. Šie metodai jau geba ieškoti sprendimo esant dideliam kiekiui taškų, todėl labai tinkami keliaujančio prekeivio problemos sprendimo paieškoms.

1.2.1. Artimiausio kaimyno algoritmas (angl. *Nearest Neighbour*, NN)

Artimiausio kaimyno algoritmas yra vienas iš heuristinių metodų, naudojamas optimizavimo uždavinių, tokių kaip keliaujančio prekeivio, sprendimui. Šio metodo veikimo principas – pradedant kuriame nors taške, kiekvienoje iteracijoje rinktis artimiausią neaplankytą tašką, tol, kol bus pasirinkti visi taškai [3]. Didžiausias šio metodo trūkumas, jog jo rezultatas labai priklausomas nuo pradinio taško pasirinkimo, tad norint gauti tikslesnius rezultatus, reikėtų metodą panaudoti kelis kartus.

Tam buvo sukurtas kartotinio artimiausio kaimyno metodas (angl. *Repetitive Nearest Neighbour,* RNN). Jo veikimo principas labai panašus į jau aprašyto artimiausio kaimyno, tačiau skiriasi tuo, jog yra sudaromas rinkinys galimų maršrutų – tai padidina tikimybę gauti geresnį sprendimą. Šis metodas buvo aprašytas ir palygintas su paprastu artimiausio kaimyno metodu ir kitame tyrime [10]. Testavimas buvo atliktas su 6 skirtingais duomenų rinkiniais iš TSPLIB (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/), kurie yra laisvai prieinami optimizavimo uždaviniams skirti skirtingo dydžio duomenų rinkiniai su pateiktu žinomu optimaliu sprendimu. Palyginus artimiausio kaimyno ir kartotinio artimiausio kaimyno metodus galima teigti, jog RNN metodas su keliaujančio prekeivio problema susitvarko geriau: iš 6 testų, keturiuose pasirodė geriau, viename prasčiau ir viename taip pat. Testuose, kuriuose pasirodė geriau, rasdavo vidutiniškai 3 % geresnį sprendimą, tai yra artimesnį žinomam optimaliam (3 pav.).



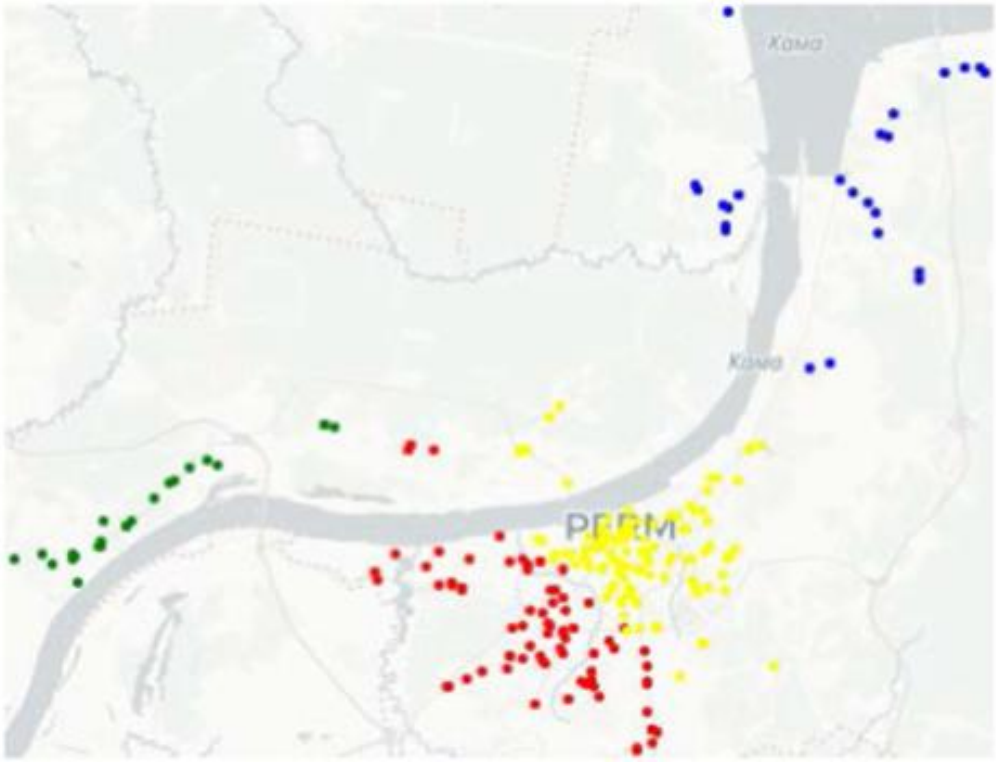
3 pav. Heuristinių metodų palyginimo tyrimas [10]

Taigi, kadangi paprasto NN metodo patobulinimas iki kartotinio NN metodo nėra sudėtingas ir reikalaujantis sudėtingų architektūros pokyčių, būtų verta įtraukti jį į savo sprendimą. Kaip jau anksčiau aptarta, šis metodas – tai keletą kartų leidžiamas paprastas artimiausio kaimyno metodas, tad reikia turėti omenyje, jog ir skaičiavimo laikas išauga. Aptartame tyrime, į laiko resursą atsižvelgta nebuvo.

1.2.2. Pigiausio įterpimo algoritmas (angl. *Cheapest Insertion*, CI)

Pigiausio įterpimo algoritmas arba pigiausio įterpimo heuristika, tai dar vienas heuristinių metodų pavyzdys. Jo veikimo principas: sudaromas pradinis maršrutas (angl. *sub-tour*) bent iš dviejų taškų (artimiausio kaimyno metode pradinis maršrutas buvo vienas taškas, todėl kaip atskiras punktas išskirtas nebuvo), ieškomas taškas, kurį įterpus tarp dviejų bet kurių taškų maršrute (pirmoje iteracijoje maršrutą sudaro tie du pasirinkti pradiniai taškai), bendras atstumas arba kitais žodžiais „kaina“ (angl. „*cost*“) padidėja mažiausiai. Tuomet tas taškas yra įterpiamas į tą atkarpą ir veiksmas kartojamas su visais likusiais nepanaudotais taškais. Pasibaigus iteracijoms gaunamas bendras maršruto ilgis [1].

Kaip jau aprašyta anksčiau, šis metodas buvo palygintas su grubiosios jėgos metodu ir rezultatai parodė, jog pastarasis susitvarkė šiek tiek geriau – rado 1,68 % ilgesnį nei optimalus maršrutą. Kitame tyrime [11], TSP uždavinys buvo sprendžiamas įvairiais metodais klasterizavimo būdu (4 pav.). Klasterizavimas leidžia didelio taškų skaičiaus optimizavimo užduotį paversti į kelis smulkesnius, mažiau skaičiavimo resursų reikalaujančius uždavinius. Tokios metodikos trūkumas toks, jog tokiu atveju sumažėja tikimybė gauti kuo artimesnį optimaliam maršrutą, kadangi skirtingų klasterių taškai vienas apie kitą neturi informacijos. Be to, po atskirų klasterių optimalių kelių radimo reikia dar ir klasterių eilės tvarkos optimizavimo. Tai dar vienas papildomas uždavinys.



4 pav. Keturi optimalūs klasteriai naudoti tyrime [11]

Tačiau šiame darbe klasterizavimas, dėl prieš tai išvardintų priežasčių atlikinėjamas nebus. Iš 13 rezultatų lentelėje esančių algoritmų [11], buvo palyginti keli aptariami šiame darbe: NN, NNR (RNN) ir CI.

Pirmame klasteryje mažiausią vidutinį bendrą atstumą rado kartotinis artimiausio kaimyno metodas (51,01 km). Artimiausio kaimyno bei pigiausio įterpimo atitinkamai 56,21 km ir 52,75 km. Kalbant

apie rezultatų pastovumą, šių trijų metodų 95 % suskaičiuotų ilgių tilpo į NN – 0,16 km, NNR – 0 km, CI – 0,05 km rėžius atitinkamai. Taigi stabiliausiai veikė NNR.

Kituose klasteriuose geriausius maršrutus rado kiti algoritmai, tačiau bendra tendencija išliko panaši: C2 (antrasis klasteris) – CI trumpiausias vidurkis (23,94 km, NN – 24,93 km, NNR – 24,27 km), 95 % verčių tokiuose rėžiuose: NN – 0,06 km, NNR – 0 km, CI – 0 km; C3 – NNR trumpiausias vidurkis (37,06 km, NN – 41,07 km, CI – 43,37 km), 95 % verčių tokiuose rėžiuose: NN – 0,19 km, NNR – 0 km, CI – 0,08 km; C4 – CI trumpiausias vidurkis (58,25 km, NN – 63,96 km, NNR – 59,57 km), 95 % verčių tokiuose rėžiuose: NN – 0,14 km, NNR – 0 km, CI – 0,06 km.

Taip pat vertėtų pažymėti, jog tyrimas buvo atliktas ir be klasterių. Rezultatai parodė, jog trumpiausią vidurkį pavyko pasiekti su NNR algoritmu (168,02 km, NN – 185,60 km CI – 185,96 km). Kadangi buvo matuojamas bendras, o ne klasterizuotas atstumas, tai ir 95 % verčių talpinantys rėžiai prasiplėtė (NN – 0,57 km, NNR – 0,00 km, CI – 0,10 km).

Taigi iš tokių rezultatų galima teigti, jog NNR arba kitaip RNN – kartotinis artimiausio kaimyno algoritmas, rodo geresnius rezultatus nei paprastas artimiausio kaimyno algoritmas (NN). Tai tik įrodo anksčiau aprašytą teiginį, jog norint gauti tikslesnius rezultatus, NN algoritmą reikėtų panaudoti kelis kartus, pradiniu tašku pasirenkant vis kitą tašką. O lyginant NNR ir CI algoritmus, didelio skirtumo nėra – abu dviejuose klasteriuose rado geriausius maršrutus, o didžiosios dalies verčių rėžiai buvo lygus 0,00 km arba neviršijo 0,10 km.

1.3. Metaheuristiniai metodai (angl. *metaheuristic methods*)

Tai aukštesnio lygio, nuo problemos nepriklausomi metodai, kurie sprendimų paieškai naudoja visą duomenų (keliaujančio prekeivio problemos atveju – taškų) imtį, o ne artimiausią aplinką, kaip kad prieš tai skyriuje aprašyti heuristiniai metodai [8]. Tačiau šie metodai taip pat tinka didelių taškų skaičių problemoms spręsti.

Toliau bus nagrinėjama tik metaheuristinių metodų šaka – populiacija grįstos paieškos algoritmai (angl. *population based search*).

1.3.1. Genetinis algoritmas (angl. *Genetic Algorithm*, GA)

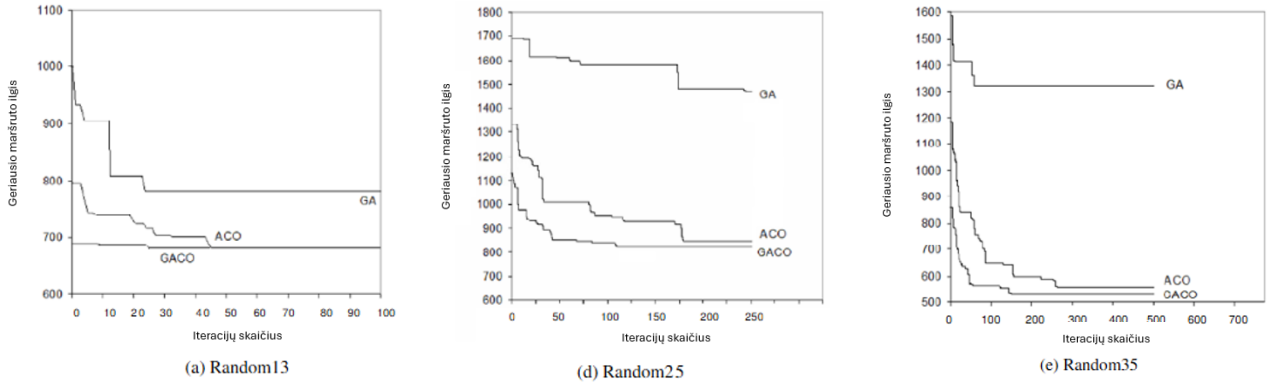
Genetinis algoritmas – natūralios atrankos įkvėptas metaheuristinis optimizavimo mechanizmas. Šis algoritmas geba rasti optimalius arba pakankamai gerus sprendimus per sąlyginai trumpą laiką, netgi ir dideliems duomenų kiekiams [12].

Jo veikimo principas gali būti paaiškintas šešiais etapais. Algoritmas prasideda nuo pradinės galimų sprendimų populiacijos (inicializavimas, angl. *initialization*), kurios vienetai gamtoje vadinami individais. Paprastai pradinės populiacijos yra sudaromos atsitiktiniu būdu. Toliau, tinkamumo funkcijos pagalba (angl. *individual evaluation*) yra įvertinamas sprendimas, TSP atveju – atvirkštinė maršruto ilgio vertė. Tuomet, trečiame etape atrenkami (angl. *selection*) geriausi individai (taškai sekoje), kurie simbolizuoja tėvus naujai individų (taškų) kartai. Pasirinktų tėvų informacija (gamtoje genai) yra keičiama kryžmiškai (angl. *crossover*), siekiant sukurti naują kartą (TSP atveju – naują sprendimą). Penktajame etape, mutacijos (angl. *mutation*) operatorius atsitiktinai pakeičia dalį naujos kartos genų (taškų sekos eiliškumą), siekiant įvesti variacijų arba kitais žodžiais tariant – mutacijų tam, kad būtų išvengta užstrigimo lokaliame minimume [13,15] Šie penki etapai kartojami tol, kol

pasiekiama nutraukimo sąlyga (angl. *termination condition*), pavyzdžiui, maksimalus iteracijų skaičius ar pakankamai geras sprendimas.

Genetinio algoritmo pritaikymo keliaujančio prekeivio problemai tyrimas buvo atliktas lyginant jį su skruzdžių kolonijos metodu (apie jį bus kalbama tolimesniame skyriuje) [16]. Atlikus tris bandymus su 13, 25 ir 35 atsitiktine tvarka išsidėsčiusiais miestais, padaryta išvada, jog laiko atžvilgiu abu metodai veikia labai panašiai: 13 miestų sprendimą abu rado per 1 laiko vienetą, 25 miestų per 3 laiko vienetus, o 35 per 8 (GA) ir 16 (ACO) laiko vienetus. Tačiau kalbant apie rastą bendrą maršruto atstumą, tai šis skyrėsi itin žymiai. Genetinio algoritmo rasti atstumai buvo kur kas prastesni (ilgesni) nei skruzdžių kolonijos: 13 miestų maršrute GA rado 15 % ilgesnį atstumą, 25 miestų maršrute tas pats genetinis algoritmas rado 83 % prastesnį sprendimą, o 35 miestų – jau net 156 % ilgesnį maršrutą kaip geriausią (1 278,64 prieš 499,98 ilgio vienetus). Galima daryti išvadą – skruzdžių kolonijos algoritmas pasirodė kur kas geriau sprendžiant šią optimizavimo užduotį.

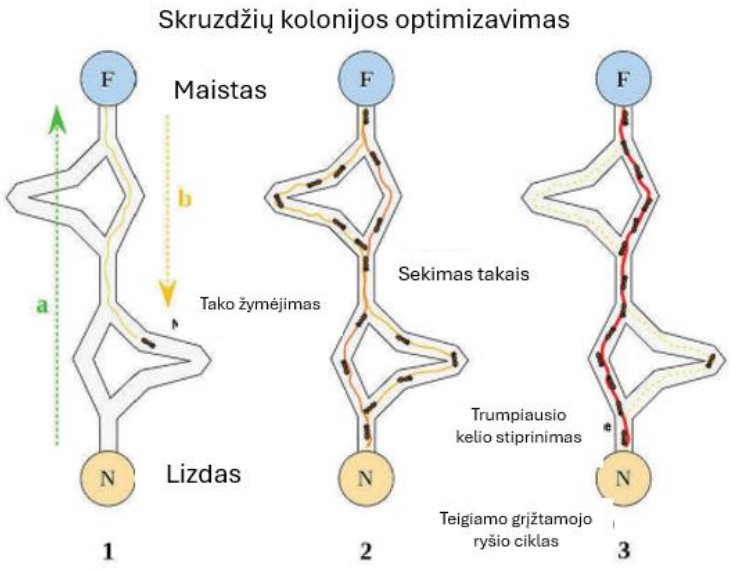
Taip pat verta pažymėti ir tai, jog kaip galima matyti tyrimo grafike (5 pav.), genetinio algoritmo sprendimo kokybė didėjant iteracijų skaičiui didėja lėčiau nei skruzdžių kolonijos algoritmo ar čia neaptarto, bet tyrime tirto šių dviejų metodų hibrido (GACO).



5 pav. Geriausių tyrimo sprendimų palyginimas su skirtingomis duomenų imtimis [16]

1.3.2. Skruzdžių kolonijos optimizavimo algoritmas (angl. *Ant Colony Optimization*, ACO)

Skruzdžių kolonijos optimizavimas – tai dar vienas metaheuristinis algoritmas ir taip pat atrastas stebint gamtą, tačiau šįkart – konkrečiai skruzdes. Skruzdės geba rasti trumpiausius kelius tarp lizdo ir maisto (6 pav.). Tai daro palikdamos chemines medžiagas, vadinamas feromonais. Kitos skruzdės pagal feromonus gali nustatyti maisto šaltinio kokybę ir atstumą. Kuo didesnė feromonų koncentracija kelyje, tuo didesnė tikimybė, jog skruzdė pasirinks būtent tą kelią [17]. Žinoma, feromonai per laiką išgaruoja, todėl mažiau naudojami keliai yra eliminuojami ir lieka tik pagrindiniai [18].



6 pav. Principinė skruzdžių elgesio ACO algoritme schema [20]

Nuo to laiko kai buvo sukurtas skruzdžių sistemos (angl. *Ant System,* AS), algoritmas 1996 m., tyrėjai sukūrė ne vieną atmainą ar patobulintą šio algoritmo modifikaciją. Toliau pateikiami keletas jų:

1. Skruzdžių kolonijos sistema (angl. *Ant Colony System*, ACS) – tai algoritmas, kuriame feromonų lygis yra pastoviai sumažinus prie tam tikrų sąlygų tam, kad būtų išvengiama įstrigimo lokaliuose minimumuose [19];
2. MAX-MIN skruzdžių sistema (angl. *MIN-MAX Ant System*, MMAS) – tai ACO algoritmas, kuriame yra nustatomos ieškomų takų ribos su tikslu skatinti platesnę paiešką ir neapsistoti ties pirmaisiais sprendimais [20]. Šis metodas yra pagrindas FACO algoritmui [20].
3. Sufokusuotas ACO algoritmas (angl. *Focused ACO*, FACO) – dar viena ACO atmaina, kurioje yra kontroliuojamas skirtumas tarp sprendimų taip įgalinant algoritmą rasti geresnius sprendimus mažose viso sprendimo dalyse [20].

Kaip buvo galima suprasti iš sukurtų skruzdžių kolonijos optimizavimo algoritmo modifikacijų aprašymų, vienas iš šio metodo trūkumų – rizika patekti į vietinio optimumo scenarijų [21].

Skruzdžių kolonijos optimizavimo metodas buvo palygintas su kitas penkiais metodais: genetiniu algoritmu (GA), imitacinio atšildymo algoritmu (SA), dirbtinių bičių kolonijos algoritmu (angl. *Artificial Bee Colony*, ABC), salpių spiečiaus algoritmu (angl. *Salp Swarm Algorithm*, SSA), pilkųjų vilkų optimizacijos algoritmu (angl. *Grey Wolf Optimization Algorithm*, GWO) [22].

Tyrime buvo atlikti T-testai – testai padedantys išsiaiškinti, ar yra didelių skirtumų tarp analizuojamų algoritmų. Testai buvo atlikti su mažo (*burma14*, *berlin52 kroA100*) dydžio duomenų rinkiniais, vidutinio (*ts225*, *att532*) dydžio duomenų rinkiniais ir didelio (*rat783*, *dsj1000*) dydžio duomenų rinkiniais. Šios duomenų imtys ir vėl buvo paimtos iš viešai prieinamos TSPLIB bibliotekos, o imties pavadinime esantis skaičius reiškia taškų skaičių.

Remiantis rezultatais, P-vertė mažesnė nei 0,05 (kas reiškia, jog skirtumas tarp statistinių reikšmių nebuvo atsitiktinis) buvo tik prie vidutinio bei didelio dydžio duomenų rinkinių. Labiausiai dominančios poros, žinoma, buvo tos, kuriose bent vienas iš lyginamų metodų yra aprašytas šiame darbe:

1. Genetinis algoritmas (GA) vidutinio ir didelio dydžio atvejais pasiekė panašius rezultatus kaip ir imitacinio atkaitinimo (SA) ar dirbtinių bičių kolonijų algoritmai (ABC);
2. Skruzdžių kolonijos algoritmas (ACO) savo rezultatais aplenkė daugelį kitų algoritmų, kurie buvo tikrinti. Nusileido tik vieninteliam hibridiniam IGSAA algoritmui (angl. *Improved Genetic Simulated Annealing Algorithm*), kuris tinka tik mažo skaičiaus taškų TSP uždaviniams spręsti.

Palyginus genetinį algoritmą su skruzdžių kolonijos algoritmu paaiškėjo, jog ACO yra pranašesnis už GA pagal nuoseklumą, tačiau nusileidžia greičiu.

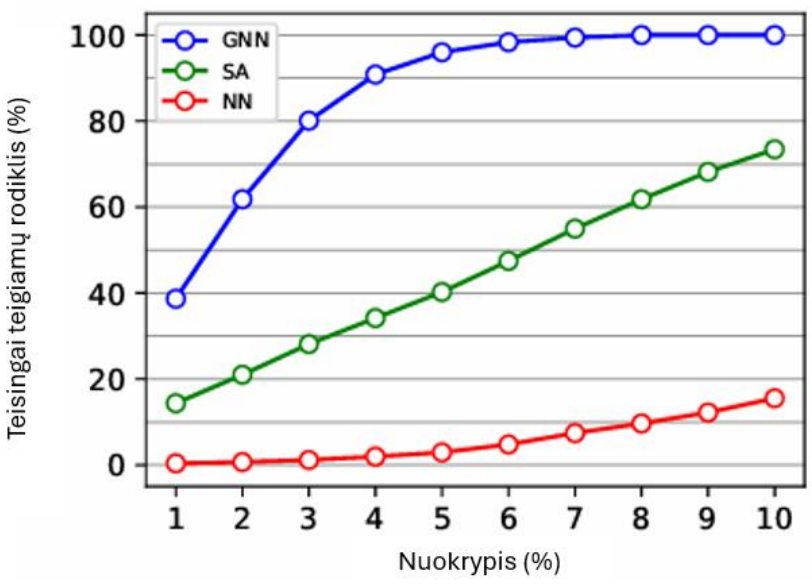
1.4. Neuroninių tinklų metodai (angl. *Neural Network methods*)

Neuroninių tinklų metodai tai dar viena metaheuristinių metodų atšaka, kuri vis dažniau naudojama TSP uždaviniui spręsti [8]. Jų pagrindinis išskirtinumas nuo visų likusių metodų yra tas, jog jie keliaujančio prekeivio uždavinį iš paieškos uždavinio paverčia į mokymosi užduotį ir vietoj to, jog iš anksto nustatytomis strategijomis ar metodais ieškotų geriausio sprendimo, tų strategijų ieško patys.

1.4.1. Grafų neuroninis tinklas (angl. *Graph Neural Network*, GNN)

Grafų neuroninis tinklas yra skirtas apdoroti duomenis, kurie turi grafų struktūrą [22] ir iš jų mokytis problemos struktūrinių savybių. Nors GNN algoritmas susitvarko su dideliais kiekiais duomenų, tačiau jis nėra tinkamas labai tankiems grafams, kokie dažnai būna keliaujančio prekeivio uždavinio grafai, kadangi juose visi taškai yra susijungę tarpusavyje. Tam buvo sukurti išankstinio apdorojimo (angl. *preprocessing*) metodai, kurie išretina tuos grafus (angl. *sparse*). Retinimas vyksta pašalinant mažiausiai perspektyvius kelius.

Grafų neuroninis tinklas buvo tirtas lyginant jį su imitacinio atšildymo (SA) ir artimiausio kaimyno (NN) algoritmais [23]. Buvo sudarytas grafikas (7 pav.), kuriame matomas optimalių maršrutų sugeneravimo dažnis su leidžiamu nuokrypiu kiekvienam iš metodų. Pateikti duomenys leidžia matyti, jog GNN metodas jau su 5 % leidžiamu nuokrypiu pasiekė beveik 100 % teisingų maršrutų generavimo dažnį, kur SA ir NN metodai net ir su 10 % leidžiamu nuokrypiu pasiekė tik 70 % ir 15 % atitinkamai.



7 pav. NN, SA ir GNN metodų palyginimo tyrimo rezultatas [23]

1.4.2. Rodykliniai tinklai (angl. *Pointer Networks*, PtrNets)

Rodykliniai tinklai buvo sukurti spręsti problemas, kuriose išvestis yra tiesiogiai susijusi su įvestimi. Keliaujančio prekeivio užduoties atveju, tai reiškia, jog modelis kiekvienoje iteracijoje nurodo (angl. *points to*) kitą tašką, kurį reikia aplankyti. *PtrNets* naudoja dėmesio (angl. *attention*) mechanizmą, leidžiantį modeliui dinamiškai nuspręsti, kuriems taškams skirti daugiau dėmesio ir pagal tai nuspręsti kuris taškas turi būti kitas maršrute.

Keliaujančio prekeivio uždavinio atveju *PtrNets* algoritmas yra apmokomas prognozuoti trumpiausio maršruto seką. Dažniausiai tai atliekama naudojant prižiūrimą mokymą (angl. *supervised learning,* SL) arba pastiprinamąjį mokymą (angl. *reinforcement learning*, RL). Pastarasis yra naudojamas dažniau, kadangi SL reikalauja didžiulių sukauptų duomenų kiekio, yra brangus ir neefektyvus, bei tinka tik problemoms su jau žinomais rezultatais ir optimaliais sprendimais. Atvirkščiai yra su RL, kuris leidžia mokytis be iš anksto apskaičiuotų optimalių sprendimų.

Buvo atliktas tyrimas [24], kuriame *PtrNets* algoritmas palygintas su keliais kitais metodais. Tyrime buvo ieškomas trumpiausias kelias arbatų lapų nurinkimui po robotu. Kaip galima matyti iš tyrimo rezultatų lentelės (**8** pav.), pasiūlytas metodas (*PtrNets* su savaiminio dėmesio, angl. *self-attention,* moduliu) nenusileido ir net aplenkė kai kuriuos anksčiau aprašytus metodus. Pavyzdžiui, genetinis algoritmas pasirodė prasčiau tiek atstumu, tiek laiko prasme prieš tyrime pasiūlytą metodą. Imitacinis atšildymas taip pat stipriai nusileido tyrime pasiūlytam metodui, ypač greitaveikos klausimu – užtrukdavo po kelias ar keliolika sekundžių (priklausomai nuo taškų skaičiaus), kur pasiūlytas metodas *PtrNets* + *self-attention* modulis daugiausiai (su 100 taškų užduotimi) užtruko tik 62,58 ms.

Metodas	$n = 20$		$n = 50$		$n = 100$	
	Ilgis	Laikas	Ilgis	Laikas	Ilgis	Laikas
Ours (greedy)	3.96	0.42 (ms)	5.99	2.34(ms)	8.40	9.26(ms)
Ours (BS10)	3.89	2.72 (ms)	5.91	15.77 (ms)	8.29	62.58 (ms)
GA	3.91	1.24(s)	6.49	2.19(s)	17.09	3.81(s)
SA	6.57	1.53(s)	19.47	5.31(s)	42.14	16.51(s)
IA	3.93	25.59(s)	6.32	53.35(s)	10.36	100.81(s)
ACA	3.86	8.38(s)	6.00	24.03(s)	8.47	54.43(s)

8 pav. Vidutinis maršruto ilgis ir skaičiavimo laikas tyrime tirtų metodų [24]

1.5. *Lin–Kernighan–Helsgaun* heuristika (angl. *Lin-Kernighan-Helsgaun heuristic,* LKH)

Lin-Kernighan-Helsgaun yra viena populiariausių pasiūlyto *Lin-Kernighan* algoritmo atmainų [25]. Ši heuristika taip pat naudojama keliaujančio prekeivio problemos sprendimų paieškai. Ji remiasi *k-opt* vietine paieška (angl. *k-opt local search*), kurios veikimo pagrindas yra iteracinis maršruto gerinimas. Pradžioje yra sudaromas pradinis maršrutas. Tuomet yra taikoma *k-opt* paieška – pašalinamos tam tikros maršruto jungtys ir pakeičiamos kitomis. Taip keičiant siekiama gauti mažesnį bendrą maršruto atstumą. Kitaip nei paprastesni *2-opt* ar *3-opt* metodai, ši heuristika dinamiškai keičia

k reikšmę paieškos metu. Tokiu būdu yra efektyviau tyrinėjama sprendimų erdvė ir atrenkami tik perspektyviausi kandidatai tolimesnėms iteracijoms.

Lin-Kernighan-Helsgaun heuristika pastaraisiais metais mokslinėje literatūroje minima kaip vienas efektyviausių metodų spręsti keliaujančio prekeivio problemą dėl gebėjimo generuoti itin aukštos kokybės sprendimus per palyginti trumpą laiką. Taip pat verta pažymėti, jog literatūroje akcentuojamas ir dar vienas šio metodo privalumas – gebėjimas spręsti ypač dideles keliaujančio prekeivio problemas, kuriose būna keli ar net keliasdešimt tūkstančių taškų [25].

Vienas iš naujausių darbų [26] panaudojo *Lin-Kernighan-Helsgaun* algoritmą kaip pagrindinį heuristinį sprendimo komponentą ir bandė pagerinti jo veikimą, jį kombinuojant su pastiprintojo mokymosi metodais (angl. *reinforcement learning*) (Q-Learning, Sarsa, Monte Carlo). Tyrimas buvo atliktas su 111 skirtingų TSPLIB bibliotekos pavyzdžių, tarp kurių buvo ir 85 900 taškų turintis rinkinys. Tai parodė, jog šis metodas tinkamas labai didelių duomenų rinkinių keliaujančio prekeivio problemos sprendimo paieškai.

Lin-Kernighan-Helsgaun metodas buvo palygintas ir su prieš tai minėtu genetinio algoritmo metodu [27], tiksliau jo atmaina – briaunų surinkimo kryžminimo genetinis algoritmas (angl. *Edge Assembly Crossover Genetic Algorithm*, EAX-GA). Tyrimas parodė, jog genetinio algoritmo atmainos metodas pasižymėjo labai gerais rezultatais didžiajai daliai keliaujančio prekeivio problemos duomenų rinkinių, tačiau *Lin-Kernighan-Helsgaun* metodas pasirodė efektyvesnis kai kuriais specifiniais atvejais, o likusiuose neparodė itin prastesnių rezultatų bei laiko prasme žymiai ilgesnių skaičiavimų, tad galima teigti, jog metodai gali būti naudojami papildant vieną kitu.

1.6. Metrikos

Siekiant kokybiškai įvertinti anksčiau išvardintų optimizavimo metodų sprendimus, bus naudojamos šios metrikos:

1. Maršruto ilgis (kuo mažesnis) – visų maršruto briaunų suma;
2. Santykinė paklaida (kuo mažesnė) – lyginamas algoritmo gautas rezultatas su žinomu optimaliu sprendimu;
3. Laiko sąnaudos (kuo mažesnės) – bendras sprendimo paieškos laikas. Ši metrika bus naudojama tik iki tam tikros vertės, kadangi ilgiau nei nustatyta trunkančio sprendimo paieškos bus nutrauktos;
4. Išplečiamumas (kuo didesnis) – bus vertinama, ar metodas veikia su didesniu taškų skaičiumi ir ar nuo to per daug nenukenčia sprendimo kokybė bei laikas;
5. Stabilumas (kuo didesnis) – bus vertinama, kiek svyruoja rezultatai leidžiant tuos pačius algoritmus po kelis kartus.

2. Metodologinė dalis

Šiame skyriuje bus plačiau apžvelgtas keliaujančio prekeivio optimizavimo uždavinio metodų tyrimo tikslas, duomenys, metodai, kriterijai, eiga ir apribojimai.

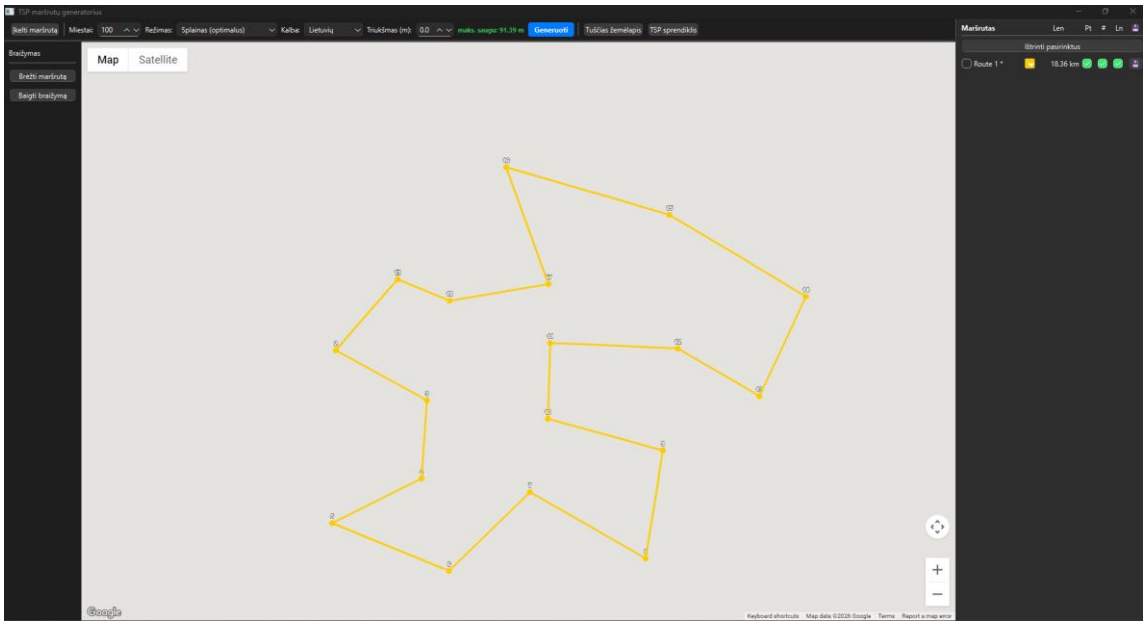
2.1. Tyrimo tikslas

Keliaujančio prekeivio optimizavimo uždavinio metodų tyrimo tikslas yra nustatyti, kurie metodai iš pasirinktų randa artimiausią optimaliam keliui, palyginti juos pasinaudojant metrikomis ir padaryti išvadas.

2.2. Tyrimo duomenys

Šiame tyrime naudojami sintetiniai duomenys buvo generuojami specialiai šiam darbui sukurta programa. Tai interaktyvus maršrutų generavimo įrankis, leidžiantis vartotojui braižyti bazinį maršrutą žemėlapyje (sudėti atraminius taškus) arba įkelti anksčiau išsaugotą maršruto failą. Pagal bazinį maršrutą, vartotojas gali pasirinkti norimą miestų skaičių ir vieną iš kelių sintetinio maršruto generavimo režimų: splaino metodą (angl. *spline*), vienodų atstumų metodą (angl. *equal-distance*), vienodų atstumų su triukšmais metodą (angl. *equal-distance + jitter*), segmentų tankinimo metodą (angl. *densify segments*). Visų šių metodų tikslas yra generuoti taškų aibę taip, kad optimali lankymo seka būtų žinoma iš anksto ir galėtų būti naudojama optimizavimo metodų tikslumui vertinti.

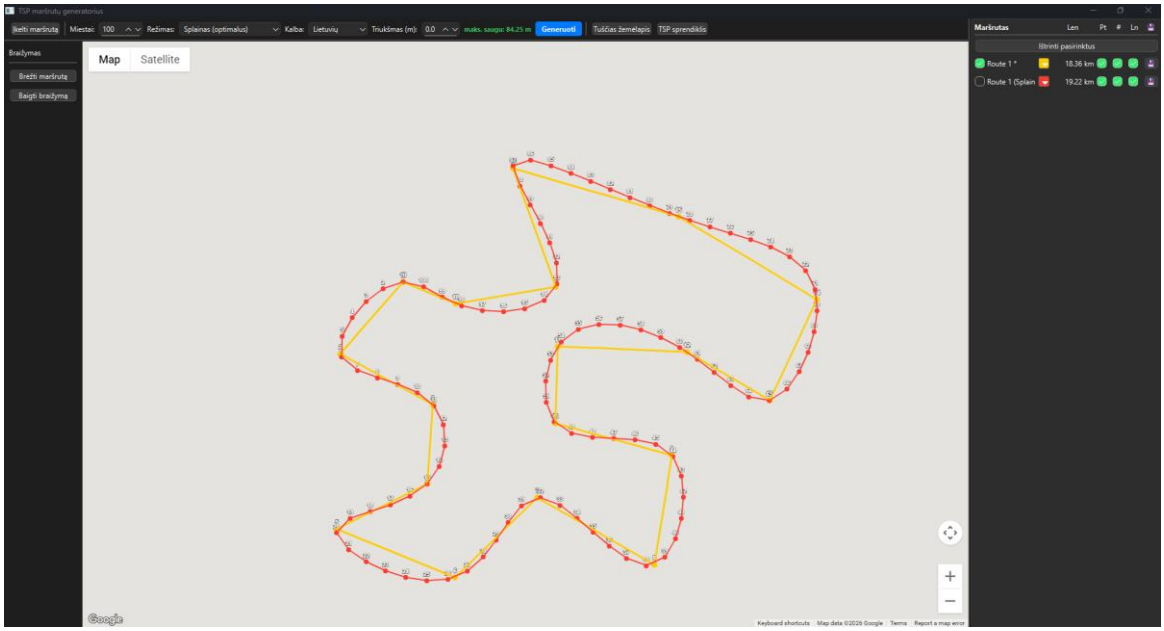
Praktinis programos veikimo pavyzdys: pirmiausia buvo nubraižytas bazinis maršrutas, tai yra sudėti atraminiai taškai, kurie apibrėžia generuojamo maršruto formą (9 pav.). Toliau buvo pasirinkti generavimo parametrai: metodas (splaino funkcija), taškų (miestų) skaičius (100, 1000 ir 10000). Paspaudus „Generuoti“ mygtuką buvo sugeneruoti atitinkamų dydžių maršrutai.



9 pav. Bazinio maršruto pavyzdys

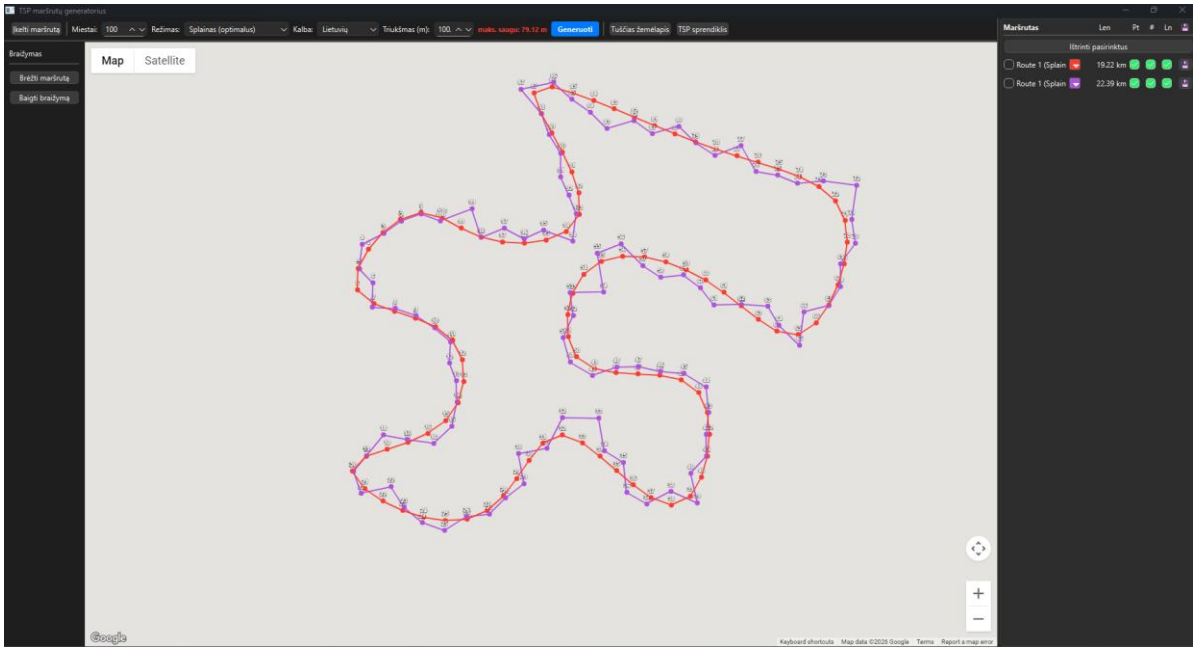
Toliau galite matyti pavyzdį (10 pav.), kuriame sugeneruotas 100 taškų maršrutas (geltona spalva – bazinis maršrutas, raudona spalva – su splaino funkcija sugeneruotas maršrutas). Kaip galima matyti pavyzdyje, įrankio dešinėje pusėje prie maršruto yra parašomas optimalaus maršruto ilgis. Nors

vartotojo nubrėžto bazinio maršruto ilgis buvo 18,36 km, tačiau po sintetinio maršruto sukūrimo šis ilgis padidėjo iki 19,22 km, kadangi atsirado daugiau nuklydimų, nuo prieš tai buvusių tiesių linijų.



10 pav. Bazinis maršrutas ir 100 taškų pavyzdinis maršrutas sugeneruotas su splaino funkcija

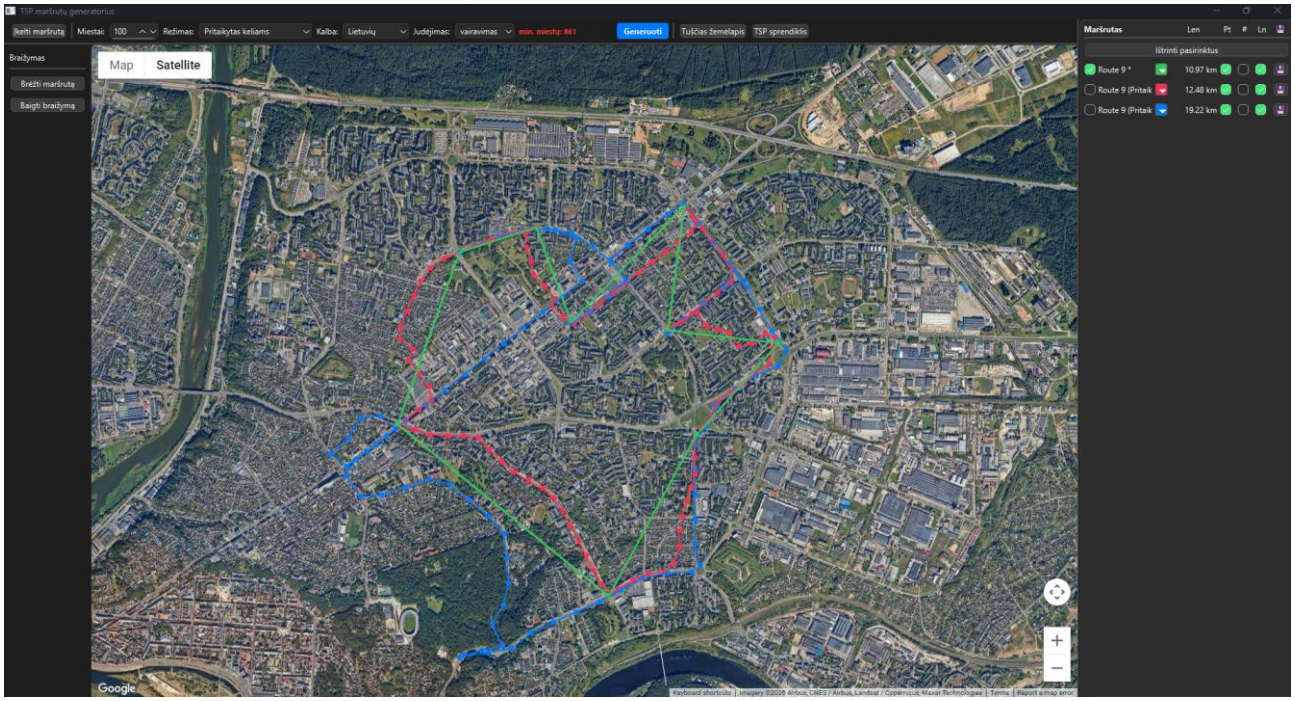
Tarp šiame pavyzdyje nenaudotų, tačiau galimų maršruto generavimo parametrų yra ir triukšmo įvertis, kuris kontroliuoja generuojamų taškų nuokrypį nuo idealios splaino kreivės. Kai triukšmo įvertis yra labai mažas arba artimas nuliui, generuojamas maršrutas lieka artimas tikrai splaino funkcijai, kuri nesant susikirtimų yra optimalus maršrutas. Jei triukšmo įvertis yra padidinamas per daug (11 pav., čia raudona spalva – 100 taškų maršrutas aprašytas prieš tai, violetinė spalva – sugeneruotas maršrutas su 100 metrų galimos maksimalios paklaidos triukšmo įverčio), yra sukuriami natūralesni išsibarstymo duomenų rinkiniai, tačiau atsiranda rizika, jog gautas maršrutas nebėra optimalus, dėl ko sukurti duomenų rinkiniai prarastų prasmę.



11 pav. Triukšmo įverčio pavyzdys generuojant maršrutus

Galiausiai spustelint išsaugojimo ikoną prie kiekvieno iš sugeneruotų maršrutų, yra išsaugomi sugeneruoti maršrutai JSON formatu. Išsaugotuose failuose yra pateikiamos maršruto taškų koordinatės, duomenų rinkinio taškų skaičius, generavimo metodas bei optimalaus maršruto ilgis. Tokiu būdu, pasirinkus duomenų rinkinio failą eksperimentų fazėje, galima palyginti, kokia buvo padaryta paklaida ieškant optimalaus maršruto.

Svarbu pažymėti, jog atliekant bandymus su šiuo įrankiu, buvo įtrauktas ir dar vienas maršrutų generavimo metodas – pritaikymas prie kelių. Vėliau pateiktame pavyzdyje (12 pav.) galite matyti tokio metodo bandymą (žalia spalva – bazinis maršrutas, raudona spalva – pritaikymas pėstiesiems, mėlyna spalva – pritaikymas automobiliams) su įjungtu palydovinės nuotraukos vaizdu. Toks metodas eksperimentų duomenų rinkiniui rinkti naudojamas nebuvo, kadangi kaip galima matyti iš pavyzdžio, kreipiant dėmesį į kelių eismo taisykles ir įmanomus bei neįmanomus kelius ar praėjimus atsiranda labai daug susikirtimų, persidengiančių kelių. Tai neleidžia užtikrinti maršruto optimalaus kelio apskaičiavimo prieš išsaugant tokį duomenų rinkinį eksperimentams.



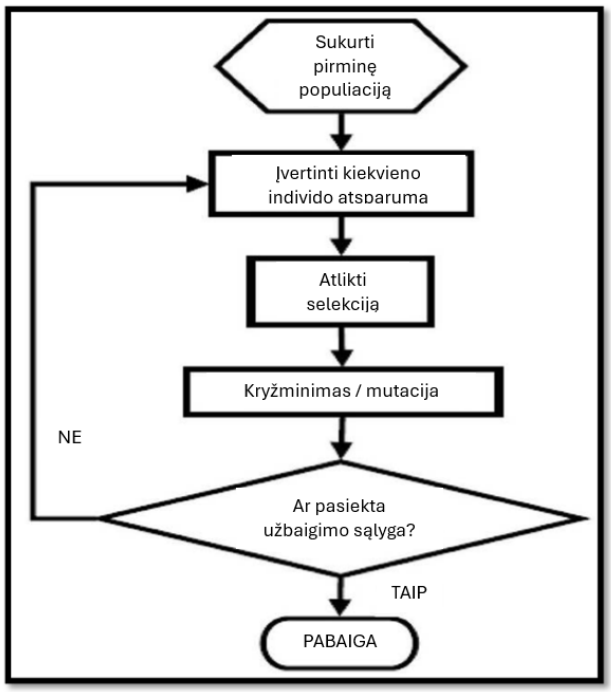
12 pav. Maršruto generavimo pasinaudojant pritaikymo prie kelių pavyzdys

2.3. Tyrimo metodai

Kaip jau buvo apžvelgta literatūros apžvalgos skyriuje, egzistuoja tikrai ne vienas metodas, kuris gali būti taikomas optimizavimo uždaviniams spręsti. Dėl šio darbo laiko apribojimų, buvo nuspręsta pasirinkti tris pastaraisiais metais labiausiai tiriamus metodus, skirtus didelių taškų skaičiaus keliaujančio prekeivio užduotims bei palyginti juos tarpusavyje ir su „Google" kompanijos vystomu *OR-Tools* metodika.

2.3.1. Genetinis algoritmas

Genetinis algoritmas (angl. *Genetic Algorithm*, GA) – tai evoliucinė metaheuristika, kuri remiasi biologinės evoliucijos principais: natūralia atranka, paveldimumu, mutacijomis. Toliau (13 pav.) galima matyti genetinių algoritmų veikimo principinę schemą.



13 pav. Genetinių algoritmų veikimo principinė schema [28]

Kaip matoma paveiksle, šio algoritmo veikimas pradedamas nuo pradinės, atsitiktinai sugeneruotos sprendinių populiacijos, kurioje kiekvienas individas (šiame darbe – maršrutas) yra galimas problemos sprendimas. Tuomet, paeiliui, kol atitiks nustatytą kriterijų, yra kuriamos evoliucinės kartos, kurios vertinamos pagal tinkamumo funkciją. Jos yra atrenkamos pagal pasirinktą metodiką, kryžminamos tarpusavyje ir yra paveikiamos mutacijų, siekiant palaipsniui gerinti sprendimo kokybę. Tokia metodika leidžia algoritmui spręsti dideles ir sudėtingas keliaujančio prekeivio problemos užduotis, su kuriomis tikslieji metodai yra netinkami, o kiti užstringa lokaliuose minimumuose [28].

Vienas iš pagrindinių genetinio algoritmo pranašumų lyginant su kitais optimizavimo metodais, yra gebėjimas palaikyti sprendinių įvairovę visos paieškos metu (jau minėtas neužstrigimas lokaliuose minimumuose). Be to, genetinius algoritmus paprasta konfigūruoti. Be kitų parametrų juose taip pat naudojami skirtingi operatoriai [29]:

1. Šifravimo (angl. *encoding*): dvejatainis (angl. *binary*), aštuntainis (angl. *octal*), šešioliktainis (angl. *hexadecimal*), kombinatorinis (angl. *permutation*), reikšminis (angl. *value*), šakų (angl. *tree*);
2. Atrankos (angl. *selection*): ruletės (angl. *roulette wheel*), rango (angl. *rank*), turnyro (angl. *tournament*), Bolcmano (angl. *Boltzmann*), stochastinė universalios atrankos (angl. *stochastic universal sampling*);
3. Kryžminimo (angl. *crossover*): vieno taško (angl. *single point crossover*), k taškų (angl. *k-point crossover*), tolygusis (angl. *uniform*), dalinio atvaizdavimo (angl. *partially mapped crossover*, PMX), tvarkos (angl. *order*, OX), pirmenybės išsaugojimo (angl. *precedence preserving crossover*, PPX), maišymo (angl. *shuffle crossover*), sumažintas pakaitinis (angl. *reduced surrogate crossover*), ciklinis (angl. *cycle crossover*);
4. Mutacijos (angl. *mutation*): perstūmimo (angl. *displacement mutation*), inversinė (angl. *inversion mutation*), atiktinio maišymo (angl. *scramble mutation*), bito pakeitimo (angl. *bit flipping mutation*), apvertimo (angl. *reversing mutation*).

Eksperimentiniai tyrimai rodo, jog genetinio algoritmo efektyvumas labai priklauso nuo prieš tai išvardintų operatorių parinkimo pagal uždavinį. Viename tyrime [30] buvo atlikta genetinio algoritmo, skirto keliaujančio prekeivio problemai, parametrų analizė. Tyrime buvo keičiamas populiacijos dydis, kryžminimo tikimybė (konkretus šio operatoriaus variantas šaltinyje nenurodytas), mutacijos tikimybė (konkretus šio operatoriaus variantas šaltinyje nenurodytas) ir iteracijų skaičius. Eksperimentų rezultatai parodė, jog šie parametrai ir jų tarpusavio sąveika turi didelę įtaką sprendimo kokybei. Pasak tyrimo autorių, per maža populiacija arba netinkamai parinktos kryžminimo ir mutacijos tikimybės lemia didesnes sprendimų variacijas.

Jau minėtame šaltinyje [28] taip pat buvo atlikta kryžminimo ir mutacijos tikimybių įtaka algoritmo efektyvumui. Šio tyrimo [28] autoriai patvirtina prieš tai minėto šaltinio [30] tyrėjų teiginius, kad esant per didelėms mutacijų ar kryžminimo tikimybėms, genetinį algoritmą gali paversti chaotišku, atsitiktiniu sprendimo paieškos atžvilgiu. Tačiau per mažos šių operatorių tikimybės gali sumažinti gerų sprendimų informacijos perdavimą naujoms sprendimų kartoms.

Taip pat verta paminėti, jog atrankos mechanizmo ir anksčiau išvardintų genetinio algoritmo operatorių pasirinkimas turi tiesioginę įtaką algoritmo artėjimo prie optimalaus sprendinio greičiui ir pastovumui [21]. Autoriai nurodo, jog stipresnę atranką sukuriantys mechanizmai, pavyzdžiui turnyrinė atranka su didesniu turnyro dydžiu, paspartina algoritmą, tačiau kartu kelia riziką per anksti įstrigti lokaliame minimume. Tyrimo išvadose taip pat teigiama, jog tam tikri kryžminimo operatoriai (priklausomai nuo užduoties) efektyviau išsaugo gerų sprendinių struktūras (keliaujančio prekeivio problemos atveju – gretimų miestų sekas), todėl šios yra lengviau perduodamos naujoms kartoms. Netinkamai parinkti kryžminimo operatorių tipai kaip tik ardo efektyvias struktūras ir blogina optimalaus sprendimo paieškas. Dar autoriai užsimena apie mutacijos operatorių vaidmenį kontroliuojant populiacijos įvairovę. Per silpna mutacija lemia sprendimų stagnaciją, o per stipri – paieškos atsitiktinumą ir sprendimų tarpusavio stabilumo sumažėjimą.

2.3.2. Skruzdžių kolonijos optimizacija

Skruzdžių kolonijos optimizacija (angl. *Ant Colony Optimization*, ACO) – tai biologijos įkvėpta metaheuristika, kuri imituoja skruzdžių kolektyvinio elgesio modelį. Ši metaheuristika sprendimus kuria iteracijomis, naudojant skruzdžių feromonų pėdsakų imitaciją. Feromonai naudojami kaip geresnių sprendinių (geresnių maršrutų) žymės (skruzdžių patirtis) ir taip leidžia sustiprinti potencialiai perspektyvesnių maršruto dalių svorį lyginant su kitomis. Tokia metodika leidžia išlaikyti balansą tarp eksploracijos (naujų maršrutų paieška) ir eksploatacijos (esamų maršrutų gerinimas), kas yra itin aktualu sprendžiant didelio masto optimizavimo uždavinius, tokius kaip keliaujančio prekeivio problema. Vis dėlto, naujausi tyrimai šį metodą papildo įvairiais mechanizmais, kurie padeda mažinti stagnaciją ir didelį skaičiavimų kiekį (pavyzdžiui klasterizavimas, adaptyvūs parametrai, vietinė paieška) [21].

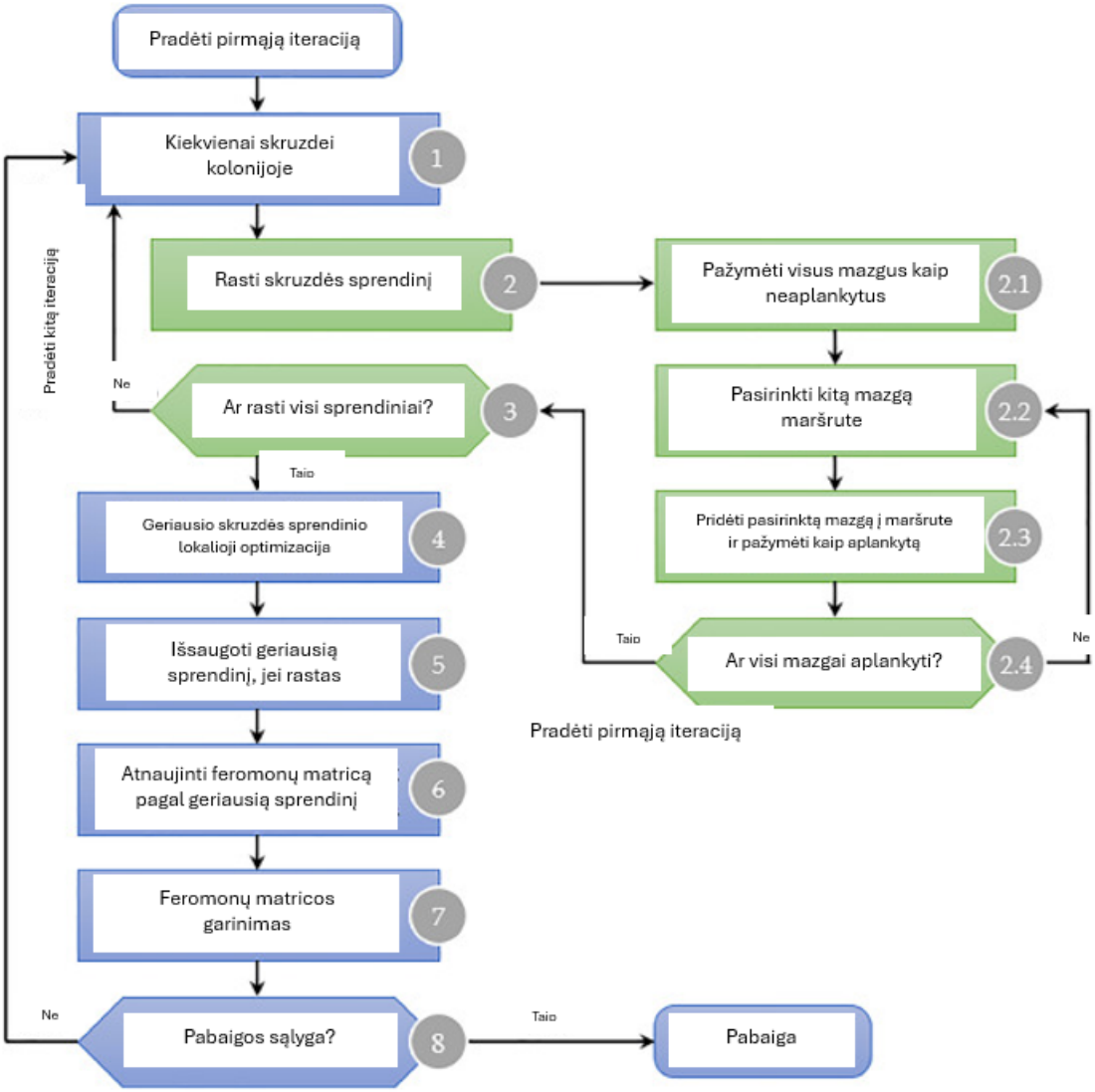
Egzistuoja skruzdžių kolonijos optimizacijos algoritmo modifikacijos, turinčios savo pavadinimus ir išskirtinius bruožus. Toliau pateikiamos pagrindinės literatūroje randamos rūšys:

1. AS (angl. *Ant System*) – bazinis variantas, kur visi agentai (skruzdės) prisideda prie feromonų atnaujinimo. Lėtesnis, labiau linkęs į stagnaciją [32];
2. ACS (angl. *Ant Colony System*) – modifikuota sprendinio konstravimo ir atnaujinimo taisyklė. Iš esmės tai – patobulintas AS variantas [32];

3. MMAS (angl. *Max-Min Ant System*) – feromonai ribojami tam tikrame intervale, kas leidžia apriboti paiešką ir mažina degeneraciją (per didelį feromono poveikį sprendimams) [32];
4. ASRank (angl. *Rank-based Ant System*) – taip pat AS varianto atmaina, kuri įtraukia ir rangus. Tai leidžia feromonus naudoti tik geriausiai pasirodančioms skruzdėms – agentams [32];
5. P-ACO (angl. *Population-based Ant Colony Optimization*) – naudojama populiacinė atmintis išsaugant geriausių sprendinių informaciją [33];
6. Adaptyvūs ACO (angl. *Adaptive Ant Colony Optimization*) – taisyklės keičiamos iteracijų metu [34];
7. Hibridiniai ACO (angl. *Hybrid Ant Colony Optimization*) – ACO padeda sugeneruoti bazinį sprendimą, o jo kokybė gerinama lokalios paieškos metodais (2-opt, 3-opt) ar imitacinio atkaitinimo (angl. *Simulated Annealing*, SA) mechanizmu [35].

Naujausiuose tyrimuose dažniausiai minimi tie patys parametrai, kurių kombinacija ir daro įtaką skruzdžių kolonijos sprendimams:
1. α – feromono įtaka. Per didelė šio parametro vertė skatiną greitą stagnaciją (susikoncentravimą į vieną trajektoriją), o per maža paverčia algoritmą atsitiktinių maršrutų generatoriumi [36];
2. β – atstumo įtaka. Didesnis šio parametro įvertis padeda greitai pasiekti gana neblogą sprendimą, tačiau dėl tos priežasties pasidaro nepajėgus išeiti iš lokalaus minimumo [36];
3. ρ – garavimo greitis. Šis parametras yra atsakingas už feromono nykimo greitį. Cituojamame šaltinyje užsimenama ir apie galimybę šį parametrą padaryti adaptyviu [36];
4. Agentų (skruzdžių) skaičius – šis parametras lemia, kiek agentų bus naudojama maršruto paieškai. Didesnis skaičius paprastai reiškia geresnio sprendimo suradimą, tačiau tam reikalingi didesni skaičiavimo resursai ir laikas [37];
5. Iteracijų / laiko limitas – kaip ir kituose metoduose, galima valdyti ir apriboti skaičiavimo laiką [35];
6. Feromonų atnaujinimo politika – iš esmės šis parametras nurodo anksčiau išvardintų skruzdžių kolonijos optimizavimo metodo atmainų tipą ir kaip jis atrenka feromonus [32].

Viename naujausių tyrimų atliktų 2022 m. [37], buvo analizuojamas skruzdžių kolonijos optimizavimo algoritmo pritaikymas keliaujančio prekeivio problemos sprendimo paieškai taikant taškų klasterizavimą ir adaptyvius algoritmo elementus (šiuo atveju konkrečiai feromonų garavimo greitis). Toks sprendimo būdas buvo pasiūlytas, nes, pasak tyrimo autoriaus, klasikinis ACO yra labai priklausomas nuo kontrolinių parametrų ir yra linkęs įstrigti lokaliuose minimumuose. Straipsnyje taip pat teigiama, jog nuo skruzdžių (agentų) skaičiaus, iteracijų skaičiaus bei feromono garavimo greičio parametro tiesiogiai priklauso skaičiavimo laikas bei sprendimo kokybė. Kaip ir galima spėti, didesnis skaičiavimo resursų kiekis dažniausiai padeda pasiekti geresnių sprendimų, tačiau tas pagerėjimas ne visada prasmingas – priklauso nuo sprendžiamo uždavinio tikslumo reikalavimų.



14 pav. Skruzdžių kolonijos optimizavimo algoritmo veikimo principinė schema [37]

Skruzdžių kolonijos optimizacijos algoritmo parametrų analizė buvo atlikta ir kitame tyrime [36]. Šiame tyrime labiau buvo dėmesys skiriamas ρ parametro įtakos algoritmo elgsenai išsiaiškinti. Anot autoriaus, tarp ρ įverčio ir konvergencijai (priartėjimo prie optimalaus maršruto) reikalingo iteracijų skaičiaus egzistuoja funkcinė priklausomybė. Darbe pristatomi rezultatai parodė, jog netinkamai parinktas ρ parametras gali sąlygoti arba per greitą feromonų augimą ir dėl to stagnaciją, arba per lėtą informacijos kaupimą – neefektyvią paiešką. Siekiant išspręsti šią parametro problemą, buvo išbandytas dinaminis tokio parametro reguliavimas, kuris pagerino konvergenciją ir sumažino priklausomybę nuo fiksuotų parametrų, kurie turi būti labai gerai suderinti kiekvienam uždaviniui atskirai, norint jog algoritmas veiktų sklandžiai ir efektyviai.

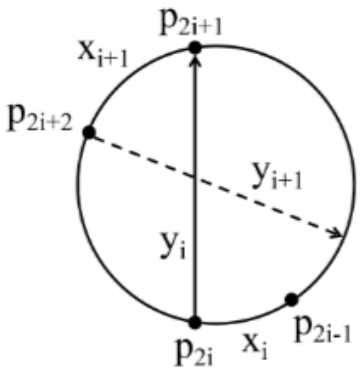
Hibridinį skruzdžių kolonijos optimizavimo metodą išbandė tyrėjai dviejuose darbuose [21, 35]. Tik vienu atveju pridėdami lokalią paiešką *2-opt*, o kitu atveju lokalią paiešką *3-opt*. Pažymėtina, pirmasis iš šių dviejų mini ir kitus hibridinius mechanizmus (klasterizavimą, imitacinį atkaitinimą), tačiau abiejų darbų autoriai sutaria, jog tokie patobulinimai padeda pašalinti pagrindinį klasikinių ACO algoritmų trūkumą – polinkį įstrigti lokaliuose minimumuose, kartu padidinant konvergencijos greitį ir sprendimo kokybę.

2.3.3. Lin-Kernighan-Helsgaun

Lin-Kernighan-Helsgaun (toliau – LKH) heuristika – tai vietinės paieškos (angl. *Local Search*) metodika, skirta keliaujančio prekeivio problemos sprendiniams optimizuoti. Šis metodas yra klasikinio *Lin-Kernighan* algoritmo patobulinimas, pasižymintis efektyvesniu sprendimų erdvės tyrinėjimu bei gebėjimu rasti labai aukštos kokybės sprendinius net ir labai didelio masto uždaviniuose [26]. Skirtingai nei, pavyzdžiui, genetiniai algoritmai, kurių veikimo principas paremtas vis naujų populiacijų sudarymu, LKH metodas veikia su vienu sprendiniu, kuris iteraciniu principu gerinamas taikant sudėtingas vietinės paieškos transformacijas.

Algoritmo veikimas pradedamas nuo pradinio maršruto sudarymo. Pradinis sprendinys gali būti sugeneruojamas atsitiktinai arba naudojant kitus heuristinius metodus (pavyzdžiui, artimiausio kaimyno metodą, paminėtą 1.2.1 skyriuje). Sudarius pradinį sprendinį, toliau vykdoma iteracinė vietinė paieška, kuri paremta *k-opt* operacijomis. *K-opt* operacijų principas – *k* (tam tikro skaičiaus) maršruto jungčių pašalinimas ir jų pakeitimas kitomis taip, jog būtų gaunamas trumpesnis bendras maršrutas.

Kaip matyti principinėje schemoje (15 pav.), nagrinėjama situacija, kai maršrute egzistuoja dvi kraštinės, jungiančios taškų poras (p_{2i}, p_{2i+1}) ir (p_{2i-1}, p_{2i+2}). Atliekant *k-opt* operaciją, šios kraštinės pašalinamos ir pakeičiamos jungtimis, taip pakeičiant dalies maršruto seką. Toks pakeitimas suteikia galimybę rasti mažesnį bendrą ilgį nei buvo prieš operaciją.



15 pav. *k-opt* tipo vietinės paieškos *Lin-Kernighan-Helsgaun* algoritme principinė schema [25]

Skirtingai nei paprasti *2-opt* ar *3-opt* (pakeičiamos atitinkamai 2 arba 3 kraštinės), kuriuose *k* reikšmė yra fiksuota, *Lin-Kernighan-Helsgaun* heuristika leidžia *k* reikšmę keisti dinamiškai vykdant paiešką. Tai leidžia algoritmui adaptuotis prie sprendinio struktūros ir pritaikyti tiek mažus, tiek didelius maršruto pakeitimus.

Siekiant sumažinti sprendimų erdvę ir tokiu būdu pagreitinti skaičiavimus, pritaikant LKH heuristiką naudojami jungčių rinkiniai [26]. Kitaip tariant, vietoj to, jog būtų nagrinėjamos visos galimos jungčių kombinacijos (kaip jau minėtame grubiosios jėgos algoritme), LKH algoritmas apsiriboja tik tomis jungtimis, kurios laikomos perspektyviomis. Būtent toks sprendinių erdvės apribojimas literatūroje įvardijamas, kaip vienas iš didžiausių *Lin-Kernighan-Helsgaun* algoritmo privalumų, leidžiančių išspręsti net ir labai dideles keliaujančio prekeivio problemas.

Toks teiginys lengvai pagrindžiamas iš eksperimentinių duomenų, kuriuose LKH metodas naudojamas kaip bazinis kuriant įvairias šio metodo atmainas. Pavyzdžiui, jau cituotame tyrime [26]

pateikiamas klasikinio LKH ir pastiprinto mokymosi LKH (VSR-LKH) palyginimas su įvairaus dydžio TSPLIB pavyzdžiais. Nagrinėjamame šaltinyje, antroje lentelėje, galima matyti, jog klasikinis LKH metodas kai kuriuose rinkiniuose nepasiekė žinomo optimalaus sprendinio, kai tuo metu sustiprintas variantas jį pasiekė stabiliai beveik su visais duomenų rinkinių pavyzdžiais. Kalbant apie skaičiavimo laiką, LKH metodo modifikacija skaičiavo greičiau nuo beveik 2 iki 48 kartų (vidutiniškai 16).

Naujesniame tyrime [38], buvo pristatytas *NeuroLKH* metodas, kuriame gilaus mokymosi modelis naudojamas LKH kandidatinių jungčių rinkinių generavimui, paliekant pačią *k-opt* vietinės paieškos schemą be pakeitimų. Tyrimo, kurio metu buvo naudojami TSPLIB dideli duomenų rinkiniai, rezultatai parodė, jog sprendinio kokybės atžvilgiu *NeuroLKH* pasiekė labai panašius arba identiškus rezultatus kaip klasikinė *Lin-Kernighan-Helsgaun* heuristika, tačiau laiko atžvilgiu skirtumai tapo ryškesni esant dideliems duomenų rinkiniams. Pavyzdžiui, 5 000 taškų keliaujančio prekeivio uždavinyje, klasikinio LKH algoritme kandidatinių jungčių paruošimas vidutiniškai užtruko apie 38 970 sekundes, kai tuo tarpu *NeuroLKH* tam pačiam tikslui pasiekti prireikė tik apie 208 sekundžių.

2.3.4. *Google OR-Tools*

Google OR-Tools - tai atvirojo kodo kombinatorinio optimizavimo programinė biblioteka, sukurta „Google“. Viena iš bibliotekos dalių, maršrutų planavimo (angl. *routing library*), naudojama keliaujančio prekeivio problemos uždaviniui spręsti [40]. Šioje bibliotekoje uždavinys formuojamas kaip vienos transporto priemonės maršrutas, su sutampančiu pradžios ir pabaigos tašku. Sprendimo procesas susideda iš dviejų pagrindinių etapų. Pirmiausia yra sugeneruojamas pradinis sprendimas naudojant vieną iš konstravimo heuristikų: automatinė (angl. *automatic*), pigiausio segmento (angl. *path cheapest arc*), labiausiai apriboto segmento (angl. *path most constrained arc*), taupymo (angl. *savings*), Christofides algoritmo (angl. *Christofides*), lygiagretaus pigiausio įterpimo (angl. *parallel cheapest insertion*), nuoseklaus pigiausio įterpimo (angl. *local cheapest insertion*), globalaus pigiausio įterpimo (angl. *global cheapest insertion*), vietinio pigiausio įterpimo (angl. *local cheapest arc*), mažiausios nepriskirtos reikšmės (angl. *first unbound min value*).

Toliau pradinis sprendimas yra tobulinamas taikant vietinę paiešką metaheuristinio metodo pagalba. Šioje vietoje taip pat yra galimybė pasirinkti iš kelių skirtingų metodų: automatinė (angl. *automatic*), grynoji vietinė paieška (angl. *greedy descent*), prižiūrima vietinė paieška (angl. *guided local search,* GLS), imitacinis atkaitinimas (angl. *simulated annealing*), tabu paieška (angl. *tabu search*).

OR-Tools pasižymi ir tuo, jog jau minėta maršrutų biblioteka skaičiavimuose naudoja sveikuosius skaičius, todėl norint naudoti atstumus, kurie turi skaitmenų po kablelio (koordinatės būtent tokios ir yra), turi būti perskaičiuojami juos padauginant iš mastelio (angl. *scale*). Per mažas mastelio skaičius gali lemti didžiules apvalinimo paklaidas, tačiau per didelis skaičius išaugina reikalingus skaičiavimo resursus. Šiame darbe buvo pasirinktas didesnio tikslumo variantas – 10 000 mastelio koeficientas. Kita šios metodikos ypatybė yra ta, jog nenustačius sprendimo paieškos laiko limito, būtų vykdoma nesibaigianti paieška ir nebūtų grąžinamas rezultatas.

Nors *OR-Tools* nėra tradicinis metodas pagal apibrėžimą, kadangi tai yra metodų kombinacija, tačiau jis buvo pasirinktas kaip papildomas palyginimo įrankis šiame tyrime, kadangi jis puikiai tinka didelio skaičiaus keliaujančio prekeivio problemos uždaviniams spręsti.

2.3.5. Pasirinkti metodai

Tyrimas bus atliekamas su šiais metodais:

- Genetinis algoritmas (angl. *Genetic Algorithm*, GA);
- Skruzdžių kolonijos optimizacija (angl. *Ant Colony Optimization*, ACO);
- *Lin–Kernighan–Helsgaun* heuristika (angl. *Lin–Kernighan–Helsgaun*, LKH);
- *Google OR-Tools.*

2.4. Vertinimo kriterijai

Kaip jau buvo paminėta 1.5 skyriuje, metodų vertinimas bus atliktas pasitelkiant metrikomis. Žinoma, metrikų svoris priklauso nuo užduoties į kurią orientuojamasi kuriant sprendimą, tačiau šiame tyrime didžiausias svoris suteikiamas maršruto ilgiui, kadangi bus žinomas optimalus maršrutas ir jo ilgis; bei laiko sąnaudos, kadangi dėl šio darbo laiko apribojimų svarbu gauti greitus ir gerus sprendimus.

2.5. Eksperimento metodų parametrai

Kadangi bus atliekami keturių algoritmų tyrimai (GA, ACO, LKH, OR-Tools), reikalingas eksperimentų vykdymo planas. Pirmiausia, pasinaudojant literatūra [19, 38, 39, 41] buvo išrinktos pradinės eksperimentų parametrų konfigūracijos kaip atskaitos taškai.

2.5.1. Genetinio algoritmo pradinė parametrų konfigūracija

Genetiniam algoritmui pradinė parametrų konfigūracija buvo nustatyta pasirėmus viename iš šaltinių [38] atliktu tyrimu, kuriame buvo naudojama genetinio algoritmo atmaina, skirta spręsti ypač didelėms keliaujančio prekeivio problemos užduotims. Joje minimi parametrai išvardinti toliau esančioje lentelėje (1 lentelė):

1 lentelė. Genetinio algoritmo pradinė parametrų konfigūracija

Parametras	**Reikšmė**	**Pastabos**
Koduotė (angl. *encoding*)	Permutacinė koduotė (angl. *permutation encoding*)	
Selekcijos metodas (angl. *selection method*)	Ruletės (angl. *roulette wheel*)	
Kryžminimo operatorius (angl. *crossover operator*)	Nuoseklus kryžminimas (angl. *order crossover*, OX)	Suteikia stabilumo dideliems duomenų rinkiniams
Kryžminimo tikimybė (angl. *crossover probability*)	0.75 (šaltinyje: 0.7 – 0.8)	Dydis, kuris užtikrina balansą tarp eksploracijos ir eksploatacijos
Mutacijos operatorius (angl. *mutation operator*)	Sukeitimo (angl. *swap*) ir inversijos (angl. *inversion*)	
Mutacijos tikimybė (angl. *mutation probability*)	0.1 (šaltinyje: 0.05 – 0.1)	Dydis, kuris valdo, kiek greitai yra leidžiama atrasti lokalų minimumą
Iteracijų skaičius (angl. *number of generations*)	500 (šaltinyje: 500 – 1 000)	

2.5.2. Skruzdžių kolonijos optimizavimo algoritmo pradinė parametrų konfigūracija

Skruzdžių kolonijos optimizavimo algoritmo tyrimas pradedamas su parametrais pateiktais kitame šaltinyje [19]. Jie išvardyti toliau esančioje lentelėje (2 lentelė).

2 lentelė. Skruzdžių kolonijos optimizavimo algoritmo pradinė parametrų konfigūracija

Parametras	**Reikšmė**	**Pastabos**
Skruzdžių skaičius (angl. *number of ants*)	Lygus miestų skaičiui	Dažniausiai naudojama keliaujančio prekeivio problemos sprendimui
Feromono svarbos koeficientas (angl. *pheromone influence*, α)	1.0	Apibrėžia feromono įtaką sprendimo konstravimui
Heuristikos svarbos koeficientas (angl. *heuristic influence,* β)	2.0	Sustiprina trumpų kraštinių pasirinkimą
Feromono garavimo koeficientas (angl. *evaporation rate,* ρ)	0.1	Padeda išvengti per greitos konvergencijos
Pradinė feromono reikšmė (angl. *initial pheromone value,* τ_0)	$\frac{1}{n \cdot I_{nn}}$	Inicializacija pagal artimiausio kaimyno heuristiką
Feromono atnaujinimo strategija (angl. *pheromone update rule*)	Tik globaliai geriausiam sprendimui	Mažina triukšmą ir stabilizuoja paiešką

2.5.3. *Lin-Kernighan-Helsgaun* heuristikos pradinė parametrų konfigūracija

3 lentelėje galite matyti *Lin-Kernighan-Helsgaun* heuristikos tyrimui parinktų parametrų konfigūraciją, kuri buvo sudaryta remiantis 39 šaltiniu, kuris yra parašytas paties metodo kūrėjo Keld Helsgaun.

3 lentelė. *Lin-Kernighan-Helsgaun* heuristikos pradinė parametrų konfigūracija

Parametras	**Reikšmė**	**Pastabos**
Pradinio sprendinio generavimo algoritmas (angl. *initial tour algorithm*)	Artimiausio kaimyno algoritmas (angl. *Nearest Neighbour*)	Numatytoji LKH inicializacija, naudojama pradiniam maršrutui generuoti
Kandidatinių kraštinių skaičius (angl. *maximum number of candidate edges*)	15 – 20	Ribojamas galimų kraštinių rinkinys, siekiant sumažinti paieškos erdvę
Kandidatinio rinkinio tipas (angl. *candidate set type*)	Delaunay	Dažniausiai naudojamas kandidatinio grafo tipas Euklidinėms keliaujančio prekeivio užduotims
Paieškos judesio tipas (angl. *move type*)	*5-opt*	Apibrėžia maksimalų *k-opt* paieškos gylį
Grįžimo lygis (angl. *backtracking*)	5	Leidžia grįžti prie ankstesnių būsenų, jei rezultatas nepagerėja
Bandymų skaičius (angl. *number of runs*)	10	Nepriklausomų paieškos paleidimų skaičius

2.5.4. *Google OR-Tools* metodo pradinė parametrų konfigūracija

Toliau esančioje lentelėje (4 lentelė), galima matyti, kokie parametrai buvo parinkti pradiniams eksperimentams. Ji buvo sudaryta pasinaudojant *Google OR-Tools* dokumentacija [41].

4 lentelė. *Google OR-Tools* metodo pradinė parametrų konfigūracija

Parametras	**Reikšmė**	**Pastabos**
Pradinio sprendimo formavimas (angl. *first solution strategy*)	PATH_CHEAPEST_ARC	Maršrutą plečia su kiekviena iteracija pridėdamas po pigiausią lanką
Vietinės paieškos metaheuristika (angl. *local search metaheuristic*)	GUIDED_LOCAL_SEARCH	*Google OR-Tools* dokumentacijoje tai efektyviausia metaheuristika maršrutų uždaviniams
Mastelio koeficientas (angl. *scale*)	10 000	

2.6. Eksperimentų vykdymo eiga

Pirmiausia bus atliekami skirtingo dydžio duomenų rinkinių eksperimentai su ankstesniame skyriuje paminėtais pradiniais parametrais. Taip pat, nekeičiant parametrų ir taškų skaičiaus, bus atliktas pakartotinis bandymas po kelis kartus, keičiant tik taškų išsidėstymą. Kitaip tariant, bus panaudoti skirtingi sintetiniai žemėlapiai tam, kad būtų patikrintas metodų veikimo stabilumas, išplečiamumas bei atkartojamumas.

2.7. Tyrimo aplinka

Tyrimai bus atliekami naudojant asmeninį kompiuterį, kurio pagrindiniai parametrai pateikiami 5 lentelėje:

5 lentelė. Tyrimo aplinka

Eil. nr.	**Parametras**	**Reikšmė**
1.	Operacinė sistema	Windows 11 Home (25H2)
2.	Centrinis procesorius	12th Gen Intel® Core™ i5-12450H (2.00GHz)
3.	RAM	16GB
4.	Sistema	64-bit
5.	Vaizdo plokštė	Intel® UHD Graphics ir NVIDIA GeForce RTX 4050 Laptop GPU
6.	Programavimo aplinka	Visual Studio Code (1.108.1)
7.	Programavimo kalba	Python 3.8.3

3. Eksperimentinė dalis

Šiame skyriuje bus pristatomi atlikti eksperimentai, apžvelgiami jų rezultatai ir pateikiami įvairių metrikų rodikliai, leidžiantys daryti išvadas apie parametrų bei duomenų rinkinių įtaką skirtingiems keliaujančio prekeivio problemos sprendimo algoritmams.

3.1. Sintetinių duomenų generavimas

Prieš atliekant bet kokius eksperimentus buvo sugeneruoti sintetiniai duomenų rinkiniai su jau metodologiniame skyriuje minėtu įrankiu. Jame ranka buvo nubraižoma atskaitinė kreivė (sudedami atraminiai taškai), kurios pagrindu buvo sugeneruojamas norimas taškų skaičius su žinomu optimaliu maršrutu tarp jų.

Generuojami buvo trijų skirtingų dydžių duomenų rinkiniai:
1. 100 taškų;
2. 1000 taškų;
3. 10000 taškų.

Taip pat svarbu pažymėti, jog kiekvienam iš skirtingų dydžių duomenys buvo generuojami skirtingais maršruto ilgiais (masteliais):
1. Iki 100 kilometrų (miesto ribos) – S;
2. 100-1000 kilometrų (rajono ribos) – M;
3. 1000-10000 kilometrų (kelių valstybių ribos) – L;
4. 10000-100000 kilometrų (žemyno ribos) – XL.

Iš viso buvo sugeneruota 120 duomenų rinkinių (3 skirtingi dydžiai, po 10 kiekvienam maršruto ilgio rėžiui).

3.2. Eksperimentai su pradiniais parametrų rinkiniais

Kaip ir buvo minėta 2.5.4 skyriuje, pirmiausia sukūrus sprendimo algoritmų testavimo įrankį buvo atlikti eksperimentai su įvairaus dydžio ir mastelio duomenų rinkiniais parametrus parenkant pagal literatūroje rastus tyrimus, kurie paminėti 2.5 skyriuje. Šie pradiniai eksperimentai buvo atlikti su 120 duomenų rinkiniu, į testavimo planą kiekvieną metodą įtraukiant tik po vieną kartą. Kitaip tariant šioje vietoje metodų stabilumo analizė atlikta nebuvo. Apie tai bus rašoma kitame skyriuje.

Iš viso buvo atlikta 480 eksperimentų (4 metodai, su visais 120 duomenų rinkinių), kurių bendras laikas buvo 9 h 53 min 14 s. Atlikus šiuos pradinius eksperimentus gauta rezultatų lentelė, kurią galima matyti toliau (6 lentelė).

6 lentelė. Tyrimo su pradiniais parametrais rezultatai

Metodo pavadinimas	Bendras užtruktas laikas	Vidutiniškai užtrukta vienam eksperimentui, s	Optimalių sprendimų (iš 120)	Vidutinė paklaida nesėkmingais atvejais
GA	2 h 6 min 34s	63,30	56 (46,70 %)	31 321,20 %
ACO	2 h 17 min 19 s	68,70	91 (75,80 %)	1,37 %
LKH	1 h 31 min 42 s	45,90	93 (77,50 %)	1,41 %
OR-Tools	3 h 57 min 37 s	118,80	111 (92,50 %)	2,97 %

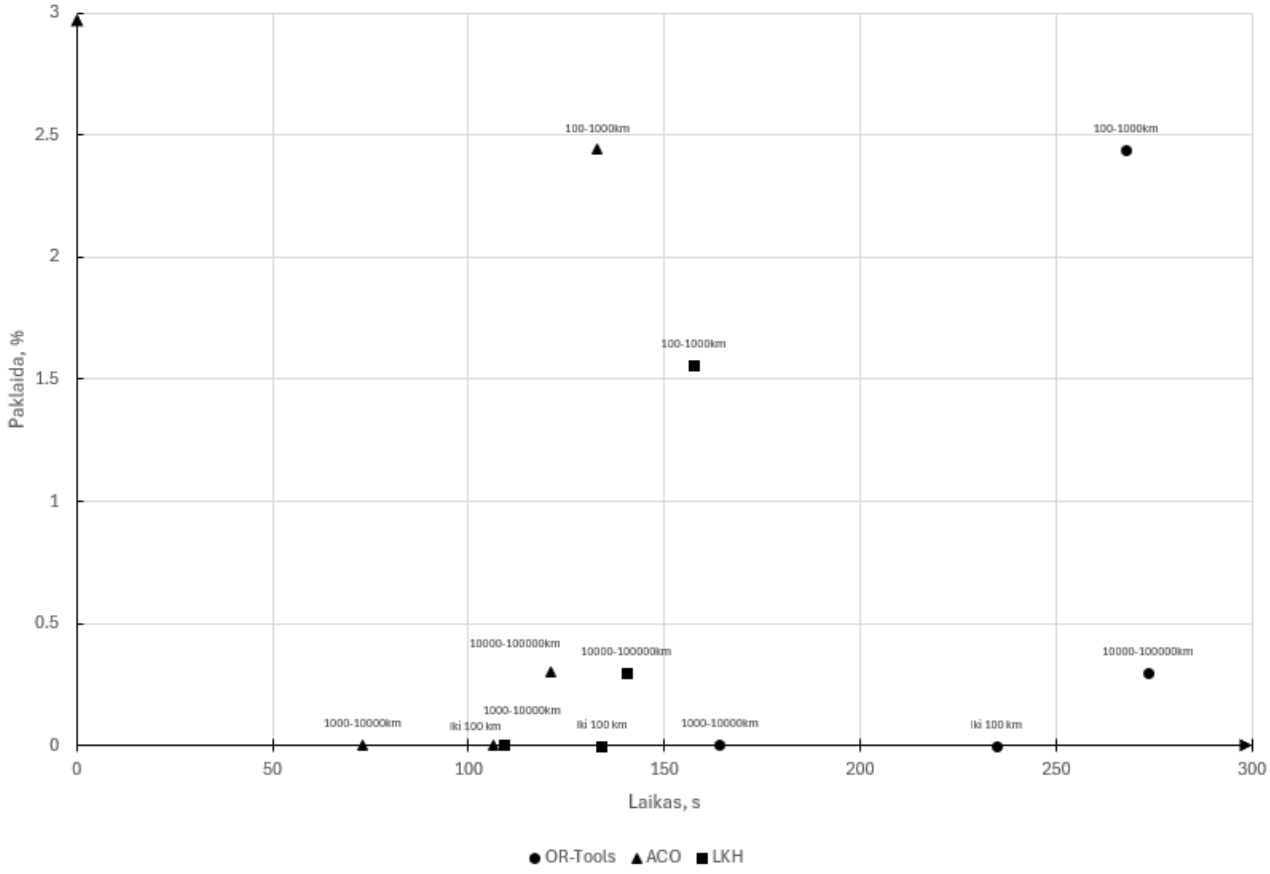
3.2.1. *OR-Tools* eksperimentai su pradiniais parametrais

Šiems eksperimentams buvo nustatytas maksimalių 60 sekundžių limitas, kadangi jo nenustačius kaip jau minėta *OR-Tools* išvis negrąžintų rezultato ir nuolatos stengtųsi patobulinti turimą. Kaip galima matyti iš rezultatų (6 lentelė), daugiausiai skaičiavimo laiko pareikalavo būtent šis metodas – iš viso 3 h 57 min 37 s. Svarbu pažymėti, jog *OR-Tools* pateikiama implementacija yra tokia, jog 60 sekundžių limitas yra skirtas tik maršruto paieškai. Kitaip tariant, laikas skirtas sudaryti atstumų matricai neįskaičiuojamas. Pažvelgus konkrečiau (**7** lentelė), tai esant 100 ir 1000 taškų duomenų rinkiniams matrica buvo sudaryta ganėtinai greitai (0,19 ir 1,60 sekundės atitinkamai). Tačiau atstumų matricos sudarymo laikas esant 10000 taškų išauga itin ryškiai – vidutinis viso sprendimo laikas padidėja iki 235 sekundžių (su maksimalia verte net 322 sekundė) iš kurių tik 60 sekundžių skiriama sprendinio paieškai, o likusios vidutiniškai 175 sekundės sunaudojamos sudarant atstumų matricą.

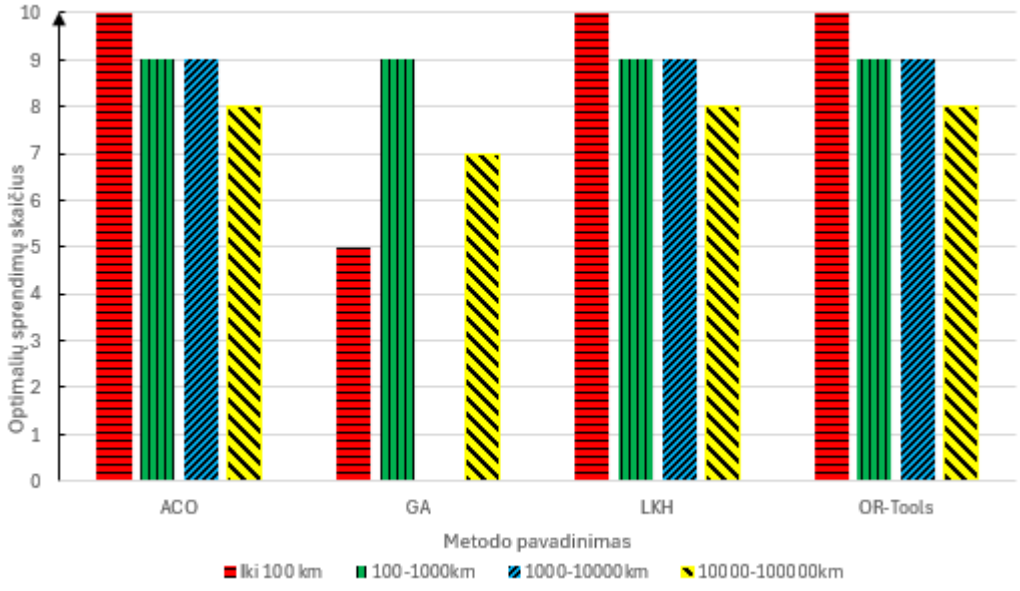
7 lentelė. *Google OR-Tools* algoritmo tyrimo su pradiniais parametrais rezultatai

Duomenų rinkinys	**Bendras užtruktas laikas**	**Vidutinis laikas, s**	**Optimalių sprendimų (10)**	**Vidutinė paklaida nesėkmingais atvejais**
100 taškų (S)	10 min 1 s	60,19	10 (100 %)	-
100 taškų (M)	10 min 1 s	60,15	10 (100 %)	-
100 taškų (L)	10 min 1 s	60,14	8 (80 %)	-0,27 %
100 taškų (XL)	10 min 1 s	60,16	7 (70 %)	-0,07 %
1000 taškų (S)	10 min 11 s	61,17	10 (100 %)	-
1000 taškų (M)	10 min 13 s	61,38	10 (100 %)	-
1000 taškų (L)	10 min 15 s	61,60	10 (100 %)	-
1000 taškų (XL)	10 min 12 s	61,25	10 (100 %)	-
10000 taškų (S)	39 min 6 s	234,62	10 (100 %)	-
10000 taškų (M)	44 min 37 s	267,74	9 (90 %)	24,40 %
10000 taškų (L)	27 min 20 s	164,04	9 (90 %)	0,03 %
10000 taškų (XL)	45 min 32 s	273,27	8 (80 %)	1,50 %

Taip pat buvo gautas grafikas (16 pav.), kuriame dėl didelio vidutinės paklaidos procento nebuvo įtrauktas genetinis algoritmas – šio metodo rezultatai esant didesniems nei iki 100 km (M, L, XL) duomenų rinkiniams buvo tūkstančiais kartų didesni nei kitų metodų. Likę metodai šiame grafike rodo, jog didžiausias paklaidos buvo gautos su 100-1000 km mastelio (M) duomenų rinkiniais (2,44 % LKH ir *OR-Tools* metodai). Jei paanalizavus kitą gautą grafiką (17 pav.), kuriame vaizduojamas optimalių sprendimų skaičius kiekvienam metodui ir esant skirtingam masteliui, tai matome, jog nors paklaidos buvo didžiausios su 100-1000 km (M) duomenų rinkiniais, tačiau optimalių sprendimų visgi pavyko rasti daugiau (36 optimalūs sprendiniai iš 40) nei 1000-10000 km (L) (27 optimalūs sprendiniai iš 40) ar 10000-100000 km (XL) (31 optimalus sprendimas iš 40). Iš to galima daryti išvadą, jog net ir esant didesnei variacijai, gebama rasti daugiau optimalių sprendimų su mažesnio mastelio duomenų rinkiniais, kai tuo tarpu didesnio mastelio duomenų rinkiniai koreliuoja su mažesniu optimalių sprendimų skaičiumi.



16 pav. Pradinių eksperimentų rezultatai su 10000 taškų duomenų rinkiniais



17 pav. Optimalių sprendimų skaičius pagal metodą ir mastelį su 10000 taškų duomenų rinkiniais

3.2.2. Skruzdžių kolonijos optimizavimo algoritmo eksperimentai su pradiniais parametrais

Skruzdžių kolonijos optimizavimo (angl. *Ant Colony Optimization*, ACO) algoritmo bendras suvartotas laikas buvo lygus 2 h 17 min 19 s (vidutiniškai 68,7 sekundės vienam eksperimentui) – tai antras ilgiausiai užtrukęs metodas. Šis metodas išsiskyrė tuo, jog buvo vienintelis, kuriame buvo

galima pritaikyti grafinio procesoriaus (angl. *graphics processing unit*, GPU) spartinimą. GPU esant 10000 taškų duomenų rinkiniams vidutiniškai suvartojo 2769 MB vaizdo atminties. Su 100 taškų duomenų rinkiniais visi eksperimentai išskyrus vieną sugebėjo sprendimą rasti greičiau nei per nustatytą laiką – 60 sekundžių (vidutiniškai 35,7 sekundės). Reikia pažymėti, jog su šiais duomenų rinkiniais į laiką įtilpo net praeidami visas 200 iteracijų. Tačiau su 1000 ir 10000 taškais laiko atžvilgiu situacija buvo prastesnė – su šiais duomenų rinkiniais laikas buvo išnaudotas maksimaliai, o su didžiausiais duomenų rinkiniais net ir viršytas, kadangi, kaip jau minėta 60 sekundžių limitas buvo naudojamas tik sprendinio paieškai, tačiau prieš tai turėjo būti sudaryti atstumų matrica. Toliau galima matyti detalesnę šio metodo eksperimentų su pradiniais parametrais išklotinę (8 lentelė).

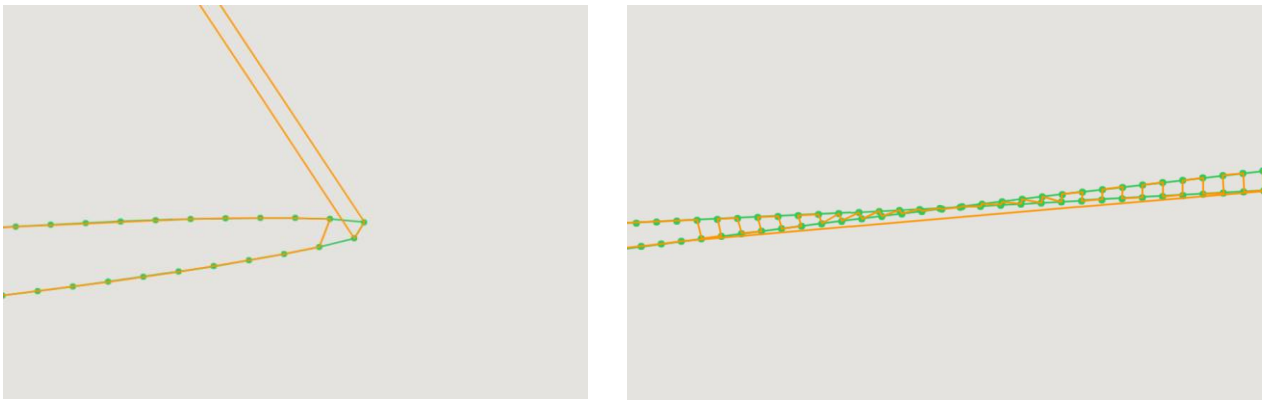
8 lentelė. Skruzdžių kolonijos optimizavimo algoritmo tyrimo su pradiniais parametrais rezultatai

Duomenų rinkinys	**Bendras užtruktas laikas**	**Vidutinis laikas, s**	**Optimalių sprendimų (10)**	**Vidutinė paklaida nesėkmingais atvejais**
100 taškų (S)	6 min 40 s	40,00	10 (100 %)	-
100 taškų (M)	5 min 11 s	31,17	5 (50 %)	0,53 %
100 taškų (L)	6 min 4 s	36,42	5 (50 %)	0,51 %
100 taškų (XL)	5 min 50 s	35,09	4 (40 %)	0,69 %
1000 taškų (S)	10 min 11 s	61,12	10 (100 %)	-
1000 taškų (M)	10 min 21 s	62,11	8 (80 %)	0,31 %
1000 taškų (L)	10 min 36 s	63,62	8 (80 %)	0,63 %
1000 taškų (XL)	10 min 12 s	61,26	5 (50 %)	0,23 %
10000 taškų (S)	17 min 45 s	106,50	10 (100 %)	-
10000 taškų (M)	22 min 8 s	132,85	9 (90 %)	24,40 %
10000 taškų (L)	12 min 9 s	72,91	9 (90 %)	0,04 %
10000 taškų (XL)	20 min 8 s	120,88	8 (80 %)	1,51 %

Ši detalesnė išklotinė rodo kelis dalykus. Pirmiausia, aiškiai matomas bendro užtrukto laiko didėjimas didėjant taškų duomenų rinkinyje skaičiui. Taip yra todėl, jog gaištamas laikas atstumų matricos sudarymui ir tik vėliau ieškomas optimalus sprendimas. Kalbant apie optimalaus sprendimo radimą, tai išanalizavus lentelės duomenis galima pastebėti įdomią koreliaciją – kuo didesnis taškų skaičius maršrute, tuo daugiau optimalių sprendimų buvo rasta, nepriklausomai nuo maršruto mastelio. Tiesa, 10000 taškų duomenų rinkinių eksperimentai nustatytą laiko limitą kai kuriais atvejais viršijo netgi 2 kartus, tačiau kaip ir minėta, skaičiavimui buvo skirta vis tiek tos pačios 60 sekundžių. Optimalaus sprendimo radimas labiau priklausė nuo to, jog atstumai tarp gretimų taškų buvo tokie maži, jog metodams nekildavo klausimų, kokius taškus jungti į seką ieškant optimalaus kelio. Analizės požiūriu įdomus scenarijus buvo pastebėtas 10000 taškų vidutinio mastelio (M) viename iš eksperimentų, kuriame vieninteliame optimalus sprendimas rastas nebuvo, o paklaida siekė netgi 24,40 %. Kaip galima matyti iš toliau esančio paveikslo (18 pav.), skruzdžių kolonijos optimizavimo algoritmas padarė akivaizdų kelio prailginimą. Taip galėjo atsitikti dėl rasto lokalaus minimumo.

18 pav. Skruzdžių kolonijos optimizavimo algoritmo rastas sprendimas su 10000 taškų (M) duomenų rinkiniu. Žalia spalva – originalus maršrutas, oranžinė – pasiūlytas ACO

Toliau galima matyti priartintus ir išdidintus skirtumus tarp originalaus maršruto ir ACO metodo pasiūlyto sprendimo (19 pav.). Svarbu tai, jog atsitiktiniu būdu dėliojant taškus duomenų generavimo etape bei vėliau pasinaudojant minėta splaino funkcija gautas maršrutas nebūtinai yra tikrai optimalus, kadangi dėl per siaurų ruožų, generuojant optimalų maršrutą įvyko sankirta. Toks taškų išsidėstymas nebegarantuoja optimalumo. Be to, sankirtų susidarymas leidžia algoritmams pasirinkti įvairias, gerai skaičiavimuose atrodančias taškų sekas ir tokiu būdu lengvai patekti į lokalius minimumus.



19 pav. Skirtumai tarp originalaus maršruto ir ACO pasiūlyto. Žalia spalva – originalus maršrutas, oranžinė – pasiūlytas ACO

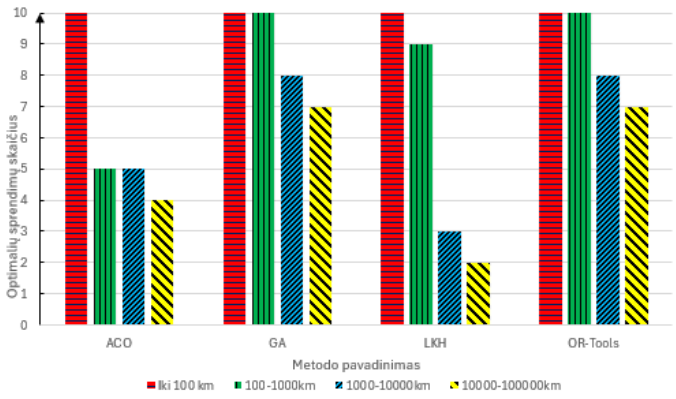
3.2.3. Genetinio algoritmo eksperimentai su pradiniais parametrais

Genetinio algoritmo eksperimentų bendras laikas buvo 2 h 6 min 34 s (vidutiniškai 63,3 sekundės vienam eksperimentui). Nors didžiojoje dalyje eksperimentų algoritmas laiko limito neviršijo, tačiau rezultatai buvo ryškiai prastesni nei likusių trijų tirtų metodų (9 lentelė).

9 lentelė. Genetinio algoritmo tyrimo su pradiniais parametrais rezultatai

Duomenų rinkinys	Bendras užtruktas laikas	Vidutinis laikas, s	Optimalių sprendimų (10)	Vidutinė paklaida nesėkmingais atvejais
100 taškų (S)	10 min 2 s	60,22	10 (100 %)	-
100 taškų (M)	8 min 19 s	50,00	10 (100 %)	-
100 taškų (L)	7 min 40 s	46,05	8 (80 %)	-0,27 %
100 taškų (XL)	7 min 42 s	46,27	7 (70 %)	-0,07 %
1000 taškų (S)	10 min 2 s	60,20	0 (0 %)	3 679,69 %
1000 taškų (M)	10 min 3 s	60,32	0 (0 %)	3 812,82 %
1000 taškų (L)	10 min 3 s	60,36	0 (0 %)	3 588,46 %
1000 taškų (XL)	10 min 2 s	60,28	0 (0 %)	2 693,61 %
10000 taškų (S)	12 min 58 s	77,89	5 (50 %)	99 960,62 %
10000 taškų (M)	15 min 21 s	92,12	9 (90 %)	96 497,70 %
10000 taškų (L)	10 min 14 s	61,48	0 (0 %)	96 869,47 %
10000 taškų (XL)	14 min 2 s	84,26	7 (70 %)	100 605,31 %

Paanalizavus gautą rezultatų lentelę detaliau, galima matyti, jog su 100 taškų algoritmas veikė puikiai – 35 iš 40 testuotų duomenų rinkinių buvo rasti optimalūs sprendimai. Matomas nedidelis procentas su neigiamu paklaidos ženklu reiškia, jog buvo rastas vos geresnis sprendimas nei pradinis nubraižytas maršrutas (kas galėjo atsitikti dėl sankirtų sugeneruotame maršrute) arba tai skaičiavimų apvalinimo klaidos, atsirandančios dėl atstumų matricos sudarymo tikslumo.



20 pav. Optimalių sprendimų skaičius pagal metodą ir mastelį su 100 taškų duomenų rinkiniais

Su 1000 ir 10000 taškų duomenų rinkiniais jau nepriklausomai nuo mastelio buvo kur kas problematiškiau – 1000 taškų duomenų rinkinių eksperimentuose genetinis algoritmas nesugebėjo nei karto rasti optimalaus sprendimo, o 10000 taškų duomenų rinkinių eksperimentuose buvo rasta tik 21 iš 40 optimalių sprendimų (20 pav.). Be to, minėtų eksperimentų vidutinės paklaidos buvo nuo

trijų tūkstančių iki šimto tūkstančių procentų, kas rodo netinkamą algoritmo elgesį bei visiškai menką patikimumą. Įdomu pažymėti tai, jog 10000 taškų duomenų rinkinių eksperimentų metu buvo rastas 21 optimalus sprendimas, kai tuo tarpu mažesni duomenų rinkiniai (su 1000 taškų) nerado nei viename iš 40 vykusių eksperimentų. Kaip jau buvo galima įžvelgti visi metodai susitvarkė šiek tiek greičiau su didelio mastelio duomenų rinkiniais nei su kitais. Genetinio algoritmo eksperimentų atveju vidutinis laikas buvo vos didesnis nei nustatytas (61,48 sekundės), tačiau atvirkščiai nei kiti metodai, genetinis algoritmas visiškai nesusitvarkė su šiais duomenų rinkiniais ir nerado nei vieno optimalaus sprendimo.

3.2.4. *Lin-Kernighan-Helsgaun* algoritmo eksperimentai su pradiniais parametrais

Lin-Kernighan-Helsgaun algoritmo eksperimentų bendras užtruktas laikas buvo 1 h 31 min 42 s. Vidutinis laikas – 45,9 sekundės. Tai greičiausias tirtas metodas. Iš 120 atliktų eksperimentų 93 atvejus (77,5 %) buvo rastas optimalus sprendimas. Galima teigti, jog LKH algoritmas pasižymi aukštesniu stabilumu bei efektyvumu randant optimalius sprendimus įvairių dydžių ir mastelių duomenų rinkiniuose. Nesėkmingais atvejais, kuomet nebūdavo randamas optimalus sprendimas, vidutinė paklaida siekė tik 1,41 %, kas buvo tik vos prasčiau už ACO metodą (1,37 %). Tai rodo, jog net ir radus neoptimalų sprendimą, jis vis tiek yra artimas optimaliam.

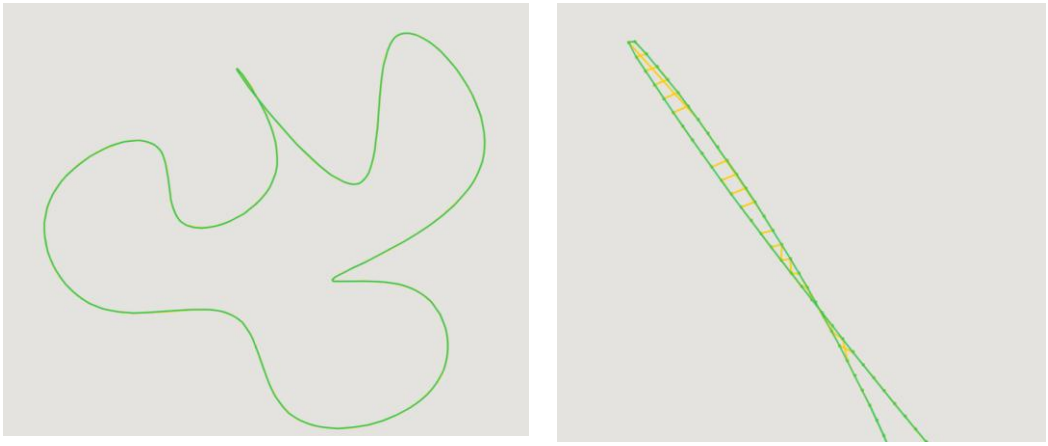
10 lentelė. *Lin-Kernighan-Helsgaun* algoritmo tyrimo su pradiniais parametrais rezultatai

Duomenų rinkinys	Bendras užtruktas laikas	Vidutinis laikas, s	Optimalių sprendimų (10)	Vidutinė paklaida nesėkmingais atvejais
100 taškų (S)	0 min 2 s	0,21	10 (100 %)	-
100 taškų (M)	0 min 2 s	0,21	9 (90 %)	0,82 %
100 taškų (L)	0 min 2 s	0,21	3 (30 %)	0,95 %
100 taškų (XL)	0 min 2 s	0,21	2 (20 %)	1,20 %
1000 taškų (S)	0 min 19 s	1,94	10 (100 %)	-
1000 taškų (M)	0 min 22 s	2,28	8 (80 %)	0,17 %
1000 taškų (L)	0 min 26 s	2,63	9 (90 %)	1,06 %
1000 taškų (XL)	0 min 19 s	1,93	6 (60 %)	0,21 %
10000 taškų (S)	22 min 19 s	133,95	10 (100 %)	-
10000 taškų (M)	26 min 13 s	157,37	9 (90 %)	15,60 %
10000 taškų (L)	18 min 11 s	109,16	9 (90 %)	0,02 %
10000 taškų (XL)	23 min 21 s	140,17	8 (80 %)	1,50 %

Paanalizavus LKH metodo eksperimentų su pradiniais parametrais lentelę (10 lentelė) pastebima, jog esant skirtingiems duomenų rinkinių dydžiams, bet mažam masteliui (bendras maršruto ilgis iki 100 kilometrų – S), visais atvejais buvo rastas optimalus maršrutas. Tiesa, sprendimo laikas skyrėsi itin žymiai – su 100 ir 1000 taškų duomenų rinkiniais buvo įtilpta į kelias sekundes (atitinkamai 0,21 s ir 1,94 s), tačiau su 10000 taškų buvo užtrukta vidutiniškai 133,95 s. Remiantis anksčiau išsakyta mintim, laiko sąnaudos išauga daugiausia dėl atstumų matricos sudarymo. Skaičiuojama yra tik pusė matricos, kadangi ji yra simetrinė, tačiau didinant taškų skaičių ji vis tiek didėja kvadratiškai. Todėl, esant 100 taškų, turime apskaičiuoti 4 950 atstumus, esant 1000 – 499 500, o 10000 taškų rinkiniui net 49 995 000 atstumus. Be to, darbe yra naudojama ir *Haversine* formulė, kuri įvertina ir žemės

paviršiaus kreivumą, skaičiuoja lanką, todėl vietoj paprastų euklidinio atstumo skaičiavimų atsiranda resursams reiklesni trigonometriniai skaičiavimai.

Išskirtinis atvejis ir vėl su 10000 taškų vidutinio mastelio (M) eksperimentais, kur nerastas tik vienas optimalus maršrutas su 15,6 % paklaida. Tačiau šis atvejis yra analogiškas analizuotam skruzdžių kolonijos optimizavimo algoritmo pradinių eksperimentų skyriuje (3.2.2 skyrius). Jame taip pat buvo klaidingai išspręsta maršruto susikirtimo problema tik pasirinkta kita maršruto kirtimo skersai taškų pora. Todėl verčiau buvo pasirinktas paanalizuoti kitas prasčiausias rezultatas – vienas nerastas optimalus sprendimas su 1,06 % paklaida (originaliai – 1 870,19 km, LKH pasiūlytas – 1 890,09 km). Toliau galite matyti tokio eksperimento rezultatą (21 pav., kuriame žalia spalva – originalus maršrutas, geltona – pasiūlytas LKH metodo). Šis atvejis taip pat įrodo, jog ir LKH metodas negeba susitvarkyti su duomenų rinkiniais, kuriuose maršrutas turi sankirtų.



21 pav. LKH metodo 1000 taškų (L) eksperimento rezultatas. Kairėje visas maršrutas, dešinėje – priartintas skirtumo regionas

3.3. Metodų stabilumo analizė

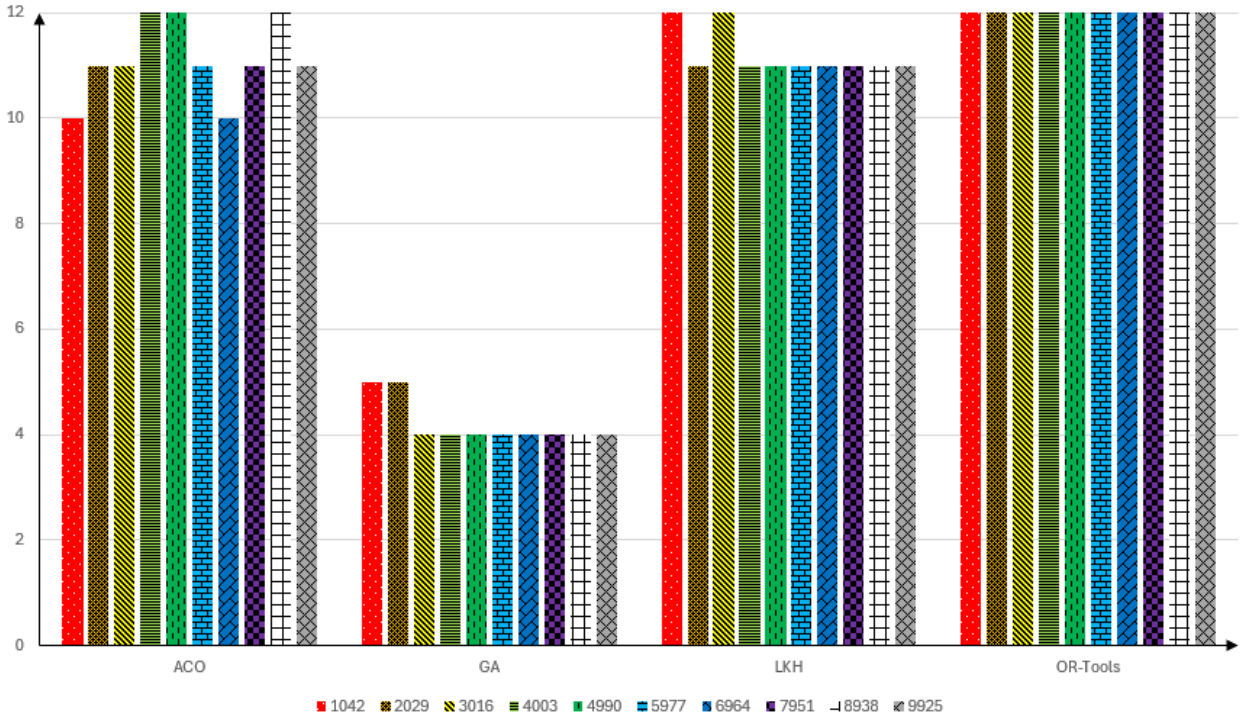
Tam, jog būtų patikrintas praeitame skyriuje (3.2 skyrius) gautų rezultatų stabilumas, buvo atlikti pakartotiniai tyrimai, tačiau su mažesniu kiekiu duomenų rinkinių, kadangi šįkart dėmesys buvo skiriamas atkartojamumui – keičiant atsitiktinio skaičiaus generatoriaus pradinę reikšmę (angl. *seed*) buvo atlikti pakartotiniai eksperimentai išrinkus po 1 skirtingo dydžio, kiekvieno maršruto ilgio rėžių duomenų rinkinius ir su visais metodais atlikus po 10 pakartotinių bandymų. Bendras eksperimentų šioje dalyje skaičius buvo 480 (12 duomenų rinkinių, visi 4 metodai po 10 pakartojimų). Bendras skaičiavimo laikas buvo 7 h 34 min 21 s. Atlikus šiuos eksperimentus gauta rezultatų lentelė, kurią galite matyti pateiktą toliau (11 lentelė).

11 lentelė. Metodų stabilumo analizės rezultatai

Metodo pavadinimas	**Identiškai rastų rinkinių rezultatų visus pakartojimus (iš 12)**	**Optimalių sprendimų (iš 120)**	**Vid. paklaidos diapazonas, %**	**Vid. laiko standartinis nuokrypis, s**
GA	4 (33,3 %)	42 (35,0 %)	111 817,17	3,77
ACO	10 (83,3 %)	111 (92,5 %)	0,7	0,69
LKH	11 (91,7 %)	112 (93,3 %)	1,65	0,81
OR-Tools	12 (100 %)	120 (100 %)	0,00	0,30

Lentelėje pateikiami keturi stabilumo rodikliai: pirmas stulpelis rodo, keliuose iš 12 eksperimentams naudotų duomenų rinkinių visus 10 pakartojimų grąžino tiksliai tą patį maršrutą (tokį patį bendrą maršruto ilgį). Antrame stulpelyje pavadinimu „Optimalių sprendimų (iš 120)“ matoma, kiek eksperimentų iš 120 (12 duomenų rinkinių po 10 pakartojimų) buvo rastas optimalus sprendimas. Kitame stulpelyje kiekvienam duomenų rinkiniui buvo apskaičiuotas skirtumas tarp blogiausio ir geriausio rasto sprendimo paklaidos ir išvestas vidurkis iš 12 duomenų rinkinių. Ši metrika aiškiai parodo, koks buvo didžiausias skirtumas tarp skirtingų paleidimų. Galiausiai, paskutiniame stulpelyje vaizduojamas kiekvieno duomenų rinkinio 10 pakartojimų vykdymo laikų standartinio nuokrypio vidurkis. Pastarasis rodiklis yra ganėtinai žemas (iki 4 sekundžių), tačiau lyginant metodus tarpusavyje paaiškėja, jog to priežastys yra skirtingos. *OR-Tools*, ACO ir GA (išskyrus su 100 taškų duomenų rinkiniais) metodai tokį stabilumą sugebėjo išlaikyti tik dėl laiko limito įvedimo. Kitaip tariant, šie metodai išnaudojo visą įmanoma skaičiavimo laiką (60 s) ir tuomet nutraukė sprendinio paiešką. Nedidelis standartinis laiko nuokrypis šiuo atveju simbolizuoja tik skirtumus tarp likusio duomenų apdorojimo greičio. Vienintelis LKH metodas nepasiekia skaičiavimo laiko limito, nutraukia paiešką anksčiau ir išlaiko laiko rodiklį panašų.

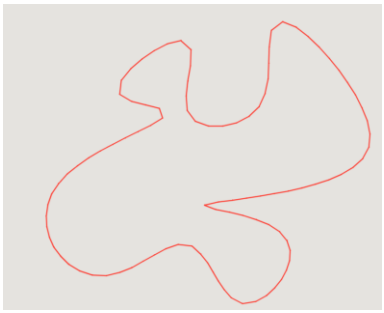
Taip pat analizės požiūriu įdomus ir gautas grafikas (22 pav.), kuriame vaizduojama stabilumo eksperimentų rezultatų (identiškų sprendimų skaičius) priklausomybė nuo metodų bei naudoto atsitiktinio skaičiaus.



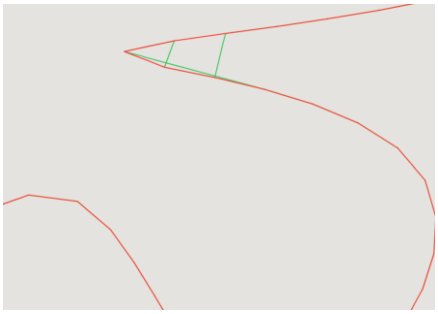
22 pav. Stabilumo eksperimentų rezultatų priklausomybė nuo metodų ir atsitiktinio skaičiaus

Iš to galima daryti išvadą, jog atsitiktinio skaičiaus (angl. *seed*) reikšmė neturėjo beveik jokios įtakos tiek kalbant apie sprendimų kokybę, tiek kalbant apie laiko sąnaudas. *OR-Tools* su visais duomenų rinkiniais (12) ir su visomis atsitiktinio skaičiaus vertėmis (10), iš viso 120 eksperimentų, grąžino identišką optimaliam rezultatą. *Lin-Kernighan-Helsgaun* metodas buvo nestabilus tik su vienu duomenų rinkiniu (100 taškų, labai didelio mastelio). Toliau pateikiami paveikslai (23 pav., 24 pav., ir 25 pav.) yra būtent to vieno maršruto eksperimentų rezultatas. Iš tyrime gautų duomenų yra aišku, jog optimalų sprendimą LKH metodas rado tik 2 kartus iš 10 (atsitiktinis skaičius buvo lygus 1042 ir

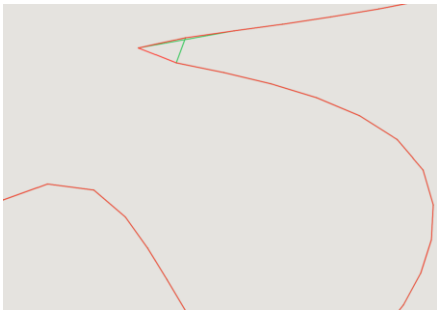
3016). Penkis kartus buvo padaryta 1,65 % paklaida (atsitiktinis skaičius buvo lygus 4003, 4990, 6964, 8938 ir 9925), kurios skirtumą nuo optimalaus maršruto galima pastebėti viduriniame paveiksle (24 pav., čia raudona spalva – originalus maršrutas, žalia spalva – LKH pasiūlytas). Du kartus (atsitiktinis skaičius buvo lygus 2029 ir 5977) buvo padaryta perpus mažesnė paklaida – 0,8 %, o vieną kartą (atsitiktinis skaičius – 7951) buvo padaryta vos 0,45 % paklaida. 25 paveiksle galima matyti priartintą 0,8 % paklaidos skirtumą nuo optimalaus maršruto (raudona spalva – originalus maršrutas, žalia spalva – LKH pasiūlytas). Remiantis rezultatais, LKH metodas su skirtingomis atsitiktinio skaičiaus pradinėmis reikšmėmis yra linkęs nueiti į skirtingus lokalius minimumus, kai duomenų rinkinys turi mažai taškų ir didelį mastelį.



23 pav. LKH metodo stabilumo tyrimo 100 taškų (XL) optimalus maršrutas



24 pav. LKH metodo stabilumo tyrimo su 100 taškų (XL) maršrutu rezultatas su priartinta 1,65 % paklaidos vieta



25 pav. LKH metodo stabilumo tyrimo su 100 taškų (XL) maršrutu rezultatas su priartinta 0,8 % paklaidos vieta

Apie skruzdžių kolonijos optimizavimo algoritmo identiškų sprendimų radimą taip pat nėra prasmės daug kalbėti, kadangi su 1000 ir 10000 taškų šis metodas buvo visiškai stabilus – visus 80 eksperimentų buvo rasti optimalūs maršrutai. Tiesa, su 100 taškų (M) ir 100 taškų (XL) mastelio duomenų rinkiniais buvo problematiškiau, atitinkamai 6 iš 10 ir 5 iš 10 optimalūs sprendimai.

Visiškai kitokį vaizdą matome jau aprašytame grafike (22 pav.) genetinio algoritmo atžvilgiu. Kitaip nei ACO ir LKH metodai, genetinis algoritmas 100 taškų duomenų rinkinių eksperimentuose rado visus optimalius sprendimus (40 iš 40). Su 1000 taškų duomenų rinkiniais šis metodas buvo visiškai nesėkmingas (0 iš 40), didžiausias paklaidos diapazonas buvo daugiau nei 3 000 % (didelio mastelio) – nuo 1 140 % iki 4 086 %. Su 10000 taškų duomenų rinkiniais taip pat beveik visi eksperimentai buvo nesėkmingi ieškant optimalaus sprendimo (2 iš 40). Nors šiuo atveju paklaidos buvo didžiulės (104 875 % – 121 262 %), tačiau jų diapazonas buvo tik 1 – 3 % bendros paklaidos vertės, tad santykinai mažesnis nei 1000 taškų eksperimentų metu. Tai galima paaiškinti tuo, jog esant nustatytam 60 sekundžių limitui, 10000 taškų duomenų rinkinių eksperimentuose genetinis algoritmas nespėja net inicializuoti pradinės populiacijos, todėl grąžinamas praktiškai atsitiktinis maršrutas, kurio kokybė mažai priklauso nuo atsitiktinio skaičiaus reikšmės. Taip pat, skyriuje, kuriame buvo aprašyti GA eksperimentai su pradiniais parametrais, buvo minima, jog 10000 taškų duomenų rinkinių eksperimentuose 21 iš 40 atvejų buvo rastas optimalus sprendimas, nors stabilumo tyrimo rezultatuose matome 2 sėkmingus atvejus (tik su vienu rinkiniu iš 4). Tai paaiškinama tuo, jog stabilumo tyrime buvo naudojami pirmieji kiekvieno mastelio maršrutai, kurie pradinių eksperimentų metu su atsitiktiniu skaičiumi lygiu 42 taip pat buvo nesėkmingi. Kitaip tariant, GA sėkmė arba nesėkmė šių tyrimų metu labiau remiasi ne atsitiktinio skaičiaus reikšme, tačiau konkrečia duomenų rinkinio geometrija.

Išvados ir rezultatai

1. Sukurtas sintetinių duomenų generavimo įrankis, leidžiantis kurti duomenų rinkinius su garantuotu ir žinomu optimaliu sprendimu bei keliaujančio prekeivio problemos sprendimo algoritmų testavimo sistema bei sugeneruota 120 sintetinių duomenų rinkinių.
2. Iš viso atlikta 480 pradinių parametrų eksperimentų. Geriausią greičio ir kokybės santykį išlaiko LKH (77,50 % optimalių sprendimų, vidutinis sprendimo laikas – 45,9 s), ACO pasirodė panašiai (75,80 %, 68,7 s), *OR-Tools* rado daugiausiai optimalių (92,50 %), tačiau buvo lėčiausias (118,8 s), genetinis algoritmas parodė labai blogus rezultatus (46,70 %, 63,3 s). Maršruto mastelis turėjo įtakos sprendimo tikslumui, tačiau ne laikui ar skaičiavimo resursams;
3. Iš viso atlikta 480 stabilumo eksperimentų. Jų metu nustatyta, jog atsitiktinio skaičiaus pradinė reikšmė, neturi pastebimos įtakos sprendimo kokybei bei laikui, nepriklausomai nuo metodo, duomenų rinkinio dydžio bei mastelio.

Literatūros sąrašas

1. AMELIA, M., SHOLEHA, Ilma I., REVANGGA, Yanda R. ir WAMILIANA, W. The comparison of brute force, cheapest-insertion, and nearest-neighbor heuristics for determining the shortest tour for visiting malls in Bandar Lampung. *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi* [interaktyvus]. 2024, 14 (1), 51–54;
2. GOUTHAM, M., MENON, M., GARROW, S., STOCKAR, S. A convex hull cheapest insertion heuristic for the non-euclidean TSP. *arXiv* [interaktyvus]. 2024, arXiv:2302.06582v3;
3. RAHMAN, Azizur M., PARVEZ, H. Repetitive nearest neighbor based simulated annealing search optimization algorithm for traveling salesman problem. *Open Access Library Journal* [interaktyvus]. 2021, 8 (6), 1-17;
4. MOHI EL DIN, H. Comparative analysis of ant colony optimization and genetic algorithm in solving the traveling salesman problem. *Blekinge Institute of Technology* [interaktyvus]. 2021;
5. EGGERT, J. Solving the traveling salesman problem with the quantum approximate optimization algorithm. *Leibniz Universität Hannover, Institute of Theoretical Physics* [interaktyvus]. 2020;
6. RANI, R., JAIN, S., GARG, H. Study of real-world optimization problems using advanced Nature Inspired Algorithms (NIA) discovered from 2019 to 2022. *Research Square* [interaktyvus]. 2024, rs.3.rs-2769987/v1;
7. VIOLINA, S. Analysis of brute force and branch & bound algorithms to solve the travelling salesperson problem (TSP). *Turkish Journal of Computer and Mathematics Education* [interaktyvus]. 2021, 12 (8), 1226–1229;
8. ZHANG, J. Comparison of various algorithms based on TSP solving. *Journal of Physics: Conference Series* [interaktyvus]. 2021, 2083, 032007;
9. ALKHALIFA, R., ALKHOMAYES, F. ALMAZROUA, B., ALHAIDAN, D., ALOTHMAN, M., ALMUHAIDIB, J. Comparative Review of Parallel Exact, Heuristic, Metaheuristic, and Hybrid Optimization Techniques for the Traveling Salesman Problem. *arXiv* [interaktyvus]. 2025, arXiv:2505.18278;
10. ZUHANDA, Muhammad K., ISMAIL, N., CARAKA, Rezzy E., SYAH, R., GIO, Prana U. Hybrid Local Search Algorithm for Optimization Route of Travelling Salesman Problem. *International Journal of Advanced Computer Science and Applications (IJACSA)* [interaktyvus]. 2023, 14 (9), 325–332;
11. SINGH, B., OBERFICHTNER, L., IVLIEV, S. Heuristics for a cash-collection routing problem with a cluster-first route-second approach. *Annals of Operations Research* [interaktyvus]. 2022, 322, 413-440;
12. KRALEV, V., KRALEVA, R. Combining Genetic Algorithm with Local Search Method in Solving Optimization Problems. *Electronics* [interaktyvus]. 2024, 13 (20), 4126;
13. HATEM, E., Dim, M. Comparative Analysis of Ant Colony Optimization and Genetic Algorithm in Solving the Traveling Salesman Problem. *Digitala Vetenskapliga Arkivet* [interaktyvus]. 2021;
14. KARAKOSTAS, P., SIFALERAS, A. A double-adaptive general variable neighborhood search algorithm for the solution of the traveling salesman problem. *Applied Soft Computing* [interaktyvus]. 2022, 108746;
15. ADIB, Muhammed, Yaseen., M., RAZIA, J., RAHMAN, Md., T. Experimental Comparison between Genetic Algorithm and Ant Colony Optimization on Traveling Salesman Problem.

International Journal of Scientific Research in Science, Engineering and Technology [interaktyvus]. 2021, 8 (1), 155–162;

16. ZUKHRI, Z., PAPUTUNGAN, Vitra, I. A Hybrid Optimization Algorithm based on Genetic Algorithm and Ant Colony Optimization. *International Journal of Artificial Intelligence & Applications (IJAIA)* [interaktyvus]. 2013, 4 (5), 63–75;
17. LIANG, S., JIAO, T., DU, W., QU, S. An improved ant colony optimization algorithm based on context for tourism route planning. *PLoS ONE* [interaktyvus]. 2021, 16 (9), e0257317;
18. KIRAN, Mustafa, S., BESKIRLI, M. A New Approach Based on Collective Intelligence to Solve Traveling Salesman Problems. *Biomimetics* [interaktyvus]. 2024, 9 (2), 118;
19. ABDELMOATY, Ahmed, M., IBRAHIM, Ibrahim, I. Comparative Analysis of Four Prominent Ant Colony Optimization Variants: Ant System, Rank-Based Ant System, Max-Min Ant System, and Ant Colony System. *arXiv* [interaktyvus]. 2024, arXiv:2405.15397;
20. SKINDEROWICZ, R. Improving Ant Colony Optimization Efficiency for Solving Large TSP Instances. *arXiv* [interaktyvus]. 2022, arXiv: 2203.02228;
21. HAO, T., YINGNIAN, W., JIAXING, Z., JING, Z. Study on a hybrid algorithm combining enhanced ant colony optimization and double improved simulated annealing via clustering in the Traveling Salesman Problem (TSP). *PeerJ Computer Science* [interaktyvus]. 2023;
22. LISCHKA, A., WU, J., BASSO, R., CHEHREGHANI, Morteza, H., KULCSAR, B. Travelling salesman problem goes sparse with graph neural networks. *ICLR 2024 Conference* [interaktyvus]. 2024;
23. PRATES, M., AVELAR, P., LEMOS, H., LAMB, L., VARDI, M. Learning to Solve NP-Complete Problems: A Graph Neural Network for Decision TSP. *Proceedings of the AAAI Conference on Artificial Intelligence* [interaktyvus]. 2019, 33 (1), 4731–4738;
24. ALANZI, E., MENAI, Mohamed El, B. Solving the traveling salesman problem with machine learning: a review of recent advances and challenges. *Artificial Intelligence Review* [interaktyvus]. 2025, 58 (267);
25. ZHENG, J., HE, K., ZHOU, J., JIN, Y., LI, C. Reinforced Lin-Kernighan-Helsgaun algorithms for the traveling salesman problems. *arXiv* [interaktyvus]. 2023, 260, 110144;
26. ZHENG, J., HE, K., ZHOU, J., JIN, Y., LI, C. Combining Reinforcement Learning with Lin-Kernighan-Heslgaun Algorithm for the Traveling Salesman Problem. *Proceedings of the AAAI Conference on Artificial Intelligence* [interaktyvus]. 2021, 35 (14), 12445–12452;
27. ZHENG, J., ZHONG, J., CHEN, M., HE, K. Reinforced Hybrid Genetic Algorithm for the Traveling Salesman Problem. *arXiv* [interaktyvus]. 2022, arXiv:2107.06870;
28. HASSANAT, A., ALMOHAMMADI, K., ALKAFAWEEN, E., ABUNAWAS, E., HAMMOURI, A., PRASATH, V. B. S. Choosing Mutation and Crossover Ratios for Genetic Algorithms – A Review with a New Dynamic Approach. *Information* [interaktyvus]. 2019, 10 (12), 390;
29. KATOCH, S., CHAUHAN, Singh, S., KUMAR, V. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications* [interaktyvus]. 2021, 80, 8091–8126;
30. NISRINA, N., KEMAL, Muhammad, I., AKBAR, Ilham, A., WIDIANTI, T. The Effect of Genetic Algorithm Parameters Tuning for Route Optimization in Travelling Salesman Problem through General Full Factorial Design Analysis. *EVERGREEN* [interaktyvus]. 2022, 9 (1), 163–203;

31. CONTRERAS-BOLTON, C., PARADA, V. Automatic Combination of Operators in a Genetic Algorithm to Solve The Traveling Salesman Problem. *PLoS ONE* [interaktyvus]. 2015, 10 (9), e0137724;
32. ABDELMOATY, Ahmed, M. Comparative Analysis of Four Prominent Ant Colony Optimization Variants: Ant System, Rank-Based Ant System, Max-Min Ant System, and Ant Colony System. *arXiv* [interaktyvus]. 2024, arXiv:2405.15397v1;
33. MAVROVOUNIOTIS, M., ANASTASIADOU, M., HADJIMITSIS, D. Measuring the Performance of Ant Colony Optimization Algorithms for the Dynamic Traveling Salesman Problem. *Algorithms* [interaktyvus]. 2023, 16 (12), 545;
34. TANG, K., WEI, Fei, X., JIANG, Hao, Y., CHEN, Wei, Z., YANG, L. An Adaptive Ant Colony Optimization for Solving Large-Scale Traveling Salesman Problem. *Mathematics* [interaktyvus]. 2023, 11 (21), 4439;
35. FEI, T., WU, X., ZHANG, L., ZHANG, Y., CHEN, L. Research on improved ant colony optimization for traveling salesman problem. *Mathematical Biosciences and Engineering* [interaktyvus]. 2022, 19 (8), 8152–8186;
36. BOSCH, J. An Analysis of the Ant Colony Optimization Parameter Space. *University of Edinburgh* [interaktyvus]. 2023;
37. STODOLA, P., OTRISAL, P., HASILOVA, K. Adaptive Ant Colony Optimization with node clustering applied to the Travelling Salesman Problem. *Swarm and Evolutionary Computation* [interaktyvus]. 2022, 70, 101056;
38. XU, H., LAN, H. An Adaptive Layered Clustering Framework with Improved Genetic Algorithm for Solving Large-Scale Traveling Salesman Problems. *Electronics* [interaktyvus]. 2023, 12 (7), 1681;
39. HELSGAUN, K. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research* [interaktyvus]. 2000, 126 (1), 106–130;
40. FURNOV, V., PERRON, L. OR-Tools Routing Library v9.12 *Google Developers* [interaktyvus]. 2025;
41. GOOGLE. Travelling Salesperson Problem. *OR-Tools Documentation* [interaktyvus]. 2024.

Priedai

1 priedas. Eksperimentų su pradiniais parametrais rezultatai

dataset_name	vertices	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_10000_001	10000	Ant Colony Optimizatic	17387.07416	75.284222	72.46875	2896.191406	TRUE	2768.62793	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_001	10000	Genetic Algorithm	17387.07416	85.9864279	83.328125	2584.390625	TRUE	0	17387.07416	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_001	10000	LKH	17387.07416	155.3070465	149.78125	2697.476563	TRUE	0	17387.07416	-1.26E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_001	10000	OR-Tools (Routing TS	17387.0783	291.2302643	282.90625	2806.324219	TRUE	0	17387.07416	2.38E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_002	10000	Ant Colony Optimizatic	24631.50989	132.6273018	129.171875	2753.25	TRUE	2768.62793	24631.50989	-1.62E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_002	10000	Genetic Algorithm	27457990.21	61.1148567	60.96875	2397.824219	TRUE	0	24631.50989	111375.0591	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_002	10000	LKH	24631.50989	109.7797623	109.953125	2522.503906	TRUE	0	24631.50989	-5.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_002	10000	OR-Tools (Routing TS	24631.51526	161.9708467	162.40625	2646.480469	TRUE	0	24631.50989	1.10E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_003	10000	Ant Colony Optimizatic	15338.16475	69.9353906	69.84375	2915.148438	TRUE	2768.62793	15338.16475	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_003	10000	Genetic Algorithm	17119233.78	61.4671905	61.453125	2554.800781	TRUE	0	15338.16475	111512.0087	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_003	10000	LKH	15338.16475	106.906041	106.953125	2531.132813	TRUE	0	15338.16475	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_003	10000	OR-Tools (Routing TS	15338.195	157.9896854	156.890625	2646.023438	TRUE	0	15338.16475	0.000197216	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_004	10000	Ant Colony Optimizatic	25850.28459	71.5692884	71.171875	2927	TRUE	2768.62793	25850.28459	-2.96E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_004	10000	Genetic Algorithm	25219165.79	61.1497361	60.421875	2565.578125	TRUE	0	25850.28459	97458.56151	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_004	10000	LKH	25850.28459	113.9541982	113.484375	2544.867188	TRUE	0	25850.28459	-4.50E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_004	10000	OR-Tools (Routing TS	25850.2786	161.040255	160.59375	2648.980469	TRUE	0	25850.28459	-2.32E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_005	10000	Ant Colony Optimizatic	28059.38016	73.804751	73.53125	2846.753906	TRUE	2768.62793	28059.38016	-4.80E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_005	10000	Genetic Algorithm	24366393.95	61.7855612	61.6875	2489.8125	TRUE	0	28059.38016	86738.67504	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_005	10000	LKH	28059.38016	106.0831638	106.359375	2541.46875	TRUE	0	28059.38016	-4.80E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_005	10000	OR-Tools (Routing TS	28059.3787	157.8093929	157.59375	2643.898438	TRUE	0	28059.38016	-5.19E-06	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_006	10000	Ant Colony Optimizatic	26586.51379	71.8858322	71.171875	2930.199219	TRUE	2768.62793	26586.51379	-2.60E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_006	10000	Genetic Algorithm	24677279.78	61.6050433	61.203125	2566.46875	TRUE	0	26586.51379	92718.78766	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_006	10000	LKH	26586.51379	109.9902943	109.75	2547.007813	TRUE	0	26586.51379	-2.87E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_006	10000	OR-Tools (Routing TS	26586.5216	243.8389258	238	2648.296875	TRUE	0	26586.51379	2.94E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_007	10000	Ant Colony Optimizatic	19329.44619	145.7340265	140.4375	2775.539063	TRUE	2768.62793	19329.44619	-5.65E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_007	10000	Genetic Algorithm	19329.44619	96.1339818	93.09375	2376.316406	TRUE	0	19329.44619	-5.65E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_007	10000	LKH	19329.44619	156.3655002	152.421875	2522.773438	TRUE	0	19329.44619	-4.89E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_007	10000	OR-Tools (Routing TS	19329.4479	311.3847814	302.5625	2644.308594	TRUE	0	19329.44619	8.87E-06	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_008	10000	Ant Colony Optimizatic	21936.50837	143.1493126	138.40625	2780.359375	TRUE	2768.62793	21936.50837	-3.32E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_008	10000	Genetic Algorithm	21936.50837	96.0583918	93.171875	2380.425781	TRUE	0	21936.50837	-4.98E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_008	10000	LKH	21936.50837	161.6594217	157.03125	2548.992188	TRUE	0	21936.50837	-4.81E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_008	10000	OR-Tools (Routing TS	21936.5019	250.0111844	246.796875	2654.660156	TRUE	0	21936.50837	-2.95E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_009	10000	Ant Colony Optimizatic	21817.6206	138.4787623	132.65625	2863.707031	TRUE	2768.62793	21817.6206	-3.33E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_009	10000	Genetic Algorithm	21817.6206	98.2635982	94.53125	2389.003906	TRUE	0	21817.6206	-1.17E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_009	10000	LKH	21817.6206	159.3558352	154.265625	2514.757813	TRUE	0	21817.6206	-3.17E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_009	10000	OR-Tools (Routing TS	21817.6249	306.6127153	298.390625	2634.402344	TRUE	0	21817.6206	1.97E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_010	10000	Ant Colony Optimizatic	23557.77963	142.4889658	137.59375	2785.433594	TRUE	2768.62793	23557.77963	-2.78E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_010	10000	Genetic Algorithm	23557.77963	95.341918	92.578125	2382.375	TRUE	0	23557.77963	1.85E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_010	10000	LKH	23557.77963	160.1127254	155.6875	2554.589844	TRUE	0	23557.77963	-3.86E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_010	10000	OR-Tools (Routing TS	23557.7765	304.3029878	298.21875	2655.71875	TRUE	0	23557.77963	-1.33E-05	{"first_solution_strateg": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_011	10000	Ant Colony Optimizatic	134113.2129	140.7683258	137.53125	2927.140625	TRUE	2768.62793	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}

dataset_name	vertex_count	method	cost	elapsed_s	cpu_time_s	peak_mem_org_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_10000_011	10000	Genetic Algorithm	134113.2129	94.7845246	92.265625	2525.304688	TRUE	0	134113.2129	3.04E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_011	10000	LKH	134113.2129	157.4032919	154.765625	2558.578125	TRUE	0	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_011	10000	OR-Tools (Routing TS	134113.2152	305.5622614	300.140625	2646.835938	TRUE	0	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_012	10000	Ant Colony Optimizatic	171655.7568	136.656301	133.1875	2889.296875	TRUE	2768.62793	171655.7568	-3.39E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_012	10000	Genetic Algorithm	171655.7568	95.2594243	93.078125	2485.335938	TRUE	0	171655.7568	-3.39E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_012	10000	LKH	171655.7568	156.8239514	152.9375	2561.410156	TRUE	0	171655.7568	-3.39E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_012	10000	OR-Tools (Routing TS	171655.7601	296.2121707	290.375	2659.898438	TRUE	0	171655.7568	1.93E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_013	10000	Ant Colony Optimizatic	193557.1886	144.5632485	140.234375	2845.53125	TRUE	2768.62793	155590.1318	24.40196965	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_013	10000	Genetic Algorithm	155590.1318	95.4831724	94.015625	2445.308594	TRUE	0	155590.1318	1.87E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_013	10000	LKH	179863.2058	164.1498729	159.59375	2536.992188	TRUE	0	155590.1318	15.60065136	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_013	10000	OR-Tools (Routing TS	193557.1889	239.745762	236.71875	2658.171875	TRUE	0	155590.1318	24.40196986	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_014	10000	Ant Colony Optimizatic	198912.2392	69.5030599	70.171875	2951.789063	TRUE	2768.62793	198912.2392	4.10E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_014	10000	Genetic Algorithm	192144639.7	61.7681585	62.140625	2550.808594	TRUE	0	198912.2392	96497.69581	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_014	10000	LKH	198912.2392	151.6894093	147.890625	2572.527344	TRUE	0	198912.2392	-8.78E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_014	10000	OR-Tools (Routing TS	198912.2412	301.095571	293.359375	2668.527344	TRUE	0	198912.2392	1.01E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_015	10000	Ant Colony Optimizatic	176356.2318	142.3939861	139.390625	2952.777344	TRUE	2768.62793	176356.2318	-3.47E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_015	10000	Genetic Algorithm	176356.2318	95.1725824	93.15625	2554.285156	TRUE	0	176356.2318	4.79E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_015	10000	LKH	176356.2318	157.9121655	154.328125	2577.570313	TRUE	0	176356.2318	-6.11E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_015	10000	OR-Tools (Routing TS	176356.2332	312.4956096	306.75	2669.066406	TRUE	0	176356.2318	8.04E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_016	10000	Ant Colony Optimizatic	180099.8702	137.2581433	134.828125	2959.269531	TRUE	2768.62793	180099.8702	1.45E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_016	10000	Genetic Algorithm	180099.8702	95.0942644	93.046875	2554.503906	TRUE	0	180099.8702	1.45E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_016	10000	LKH	180099.8702	158.2184584	155.40625	2577.109375	TRUE	0	180099.8702	-4.85E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_016	10000	OR-Tools (Routing TS	180099.8674	297.1115673	292.28125	2666.0625	TRUE	0	180099.8702	-1.57E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_017	10000	Ant Colony Optimizatic	239519.8406	136.9672284	134.46875	2958.875	TRUE	2768.62793	239519.8406	1.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_017	10000	Genetic Algorithm	239519.8406	94.8300412	93.34375	2556.339844	TRUE	0	239519.8406	1.09E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_017	10000	LKH	239519.8406	156.5883165	153.1875	2576.957031	TRUE	0	239519.8406	1.09E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_017	10000	OR-Tools (Routing TS	239519.8332	287.3962219	280.296875	2670.925781	TRUE	0	239519.8406	-3.08E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_018	10000	Ant Colony Optimizatic	189997.4821	143.3475914	137.796875	2949.988281	TRUE	2768.62793	189997.4821	-4.60E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_018	10000	Genetic Algorithm	189997.4821	96.4529158	92.875	2560.320313	TRUE	0	189997.4821	6.74E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_018	10000	LKH	189997.4821	156.1507425	149.046875	2573.324219	TRUE	0	189997.4821	-2.45E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_018	10000	OR-Tools (Routing TS	189997.4815	313.0472821	301.5	2612.648438	TRUE	0	189997.4821	-3.24E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_019	10000	Ant Colony Optimizatic	198279.7988	145.229549	140.03125	2803.871094	TRUE	2768.62793	198279.7988	2.94E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_019	10000	Genetic Algorithm	198279.7988	96.1358535	92.359375	2537.390625	TRUE	0	198279.7988	3.82E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_019	10000	LKH	198279.7988	154.7574573	148.703125	2564.917969	TRUE	0	198279.7988	1.47E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_019	10000	OR-Tools (Routing TS	198279.804	163.0820921	163.140625	2634.203125	TRUE	0	198279.7988	2.65E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_020	10000	Ant Colony Optimizatic	243962.9657	131.8228973	128.6875	2816.351563	TRUE	2768.62793	243962.9657	3.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_020	10000	Genetic Algorithm	243962.9657	96.2258275	94.375	2561.457031	TRUE	0	243962.9657	-1.43E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_020	10000	LKH	243962.9657	159.9919052	157.4375	2578.835938	TRUE	0	243962.9657	-9.54E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_020	10000	OR-Tools (Routing TS	243962.9693	161.6716953	160.8125	2671.675781	TRUE	0	243962.9657	1.48E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_021	10000	Ant Colony Optimizatic	1580592.847	73.8369825	73.875	2965.632813	TRUE	2768.62793	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_021	10000	Genetic Algorithm	1689770490	61.7493532	61.546875	2608.058594	TRUE	0	1580592.847	106807.3856	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}

dataset_name	vertex_count	method	cost	elapsed_s	cpu_time_s	peak_mem_org_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_10000_021	10000	LKH	1580592.847	108.1482359	108.328125	2587.640625	TRUE	0	1580592.847	-5.45E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_021	10000	OR-Tools (Routing TS	1580592.841	166.341739	166.640625	2678.699219	TRUE	0	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_022	10000	Ant Colony Optimizatic	2021372.51	73.3821499	73.25	2840.054688	TRUE	2768.62793	2021372.51	4.61E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_022	10000	Genetic Algorithm	1813425192	61.2747261	61.140625	2605.964844	TRUE	0	2021372.51	89612.56821	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_022	10000	LKH	2021372.51	110.46437	110.828125	2586.066406	TRUE	0	2021372.51	-2.30E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_022	10000	OR-Tools (Routing TS	2021372.514	165.1446769	165.140625	2677.011719	TRUE	0	2021372.51	1.69E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_023	10000	Ant Colony Optimizatic	1872334.521	73.3256145	72.265625	2955.65625	TRUE	2768.62793	1871678.581	0.035045529	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_023	10000	Genetic Algorithm	1943441377	61.2624491	61.34375	2606.527344	TRUE	0	1871678.581	103734.141	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_023	10000	LKH	1872122.03	112.6076155	112.421875	2588.609375	TRUE	0	1871678.581	0.023692576	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_023	10000	OR-Tools (Routing TS	1872280.865	163.572835	163.265625	2677.214844	TRUE	0	1871678.581	0.032178813	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_024	10000	Ant Colony Optimizatic	2223060.569	70.3779912	69.9375	2973.765625	TRUE	2768.62793	2223060.569	-1.05E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_024	10000	Genetic Algorithm	2237992331	61.2729398	61.046875	2610.65625	TRUE	0	2223060.569	1005716.759	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_024	10000	LKH	2223060.569	110.5892896	110.8125	2591.59375	TRUE	0	2223060.569	-3.35E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_024	10000	OR-Tools (Routing TS	2223060.57	158.3633385	158.828125	2678.980469	TRUE	0	2223060.569	4.07E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_025	10000	Ant Colony Optimizatic	3171704.33	70.2520314	70.34375	2963.292969	TRUE	2768.62793	3171704.33	-3.96E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_025	10000	Genetic Algorithm	3131418920	61.3862917	61.40625	2611.183594	TRUE	0	3171704.33	98629.84977	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_025	10000	LKH	3171704.33	107.381013	106.421875	2594.519531	TRUE	0	3171704.33	-3.96E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_025	10000	OR-Tools (Routing TS	3171704.338	159.3053744	158.765625	2682.472656	TRUE	0	3171704.33	2.64E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_026	10000	Ant Colony Optimizatic	2812870.74	74.4168957	74.34375	2826.933594	TRUE	2768.62793	2812870.74	2.48E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_026	10000	Genetic Algorithm	2911349767	61.2701509	61.203125	2613.371094	TRUE	0	2812870.74	103401.015	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_026	10000	LKH	2812870.74	109.1281913	108.6875	2593.371094	TRUE	0	2812870.74	-1.99E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_026	10000	OR-Tools (Routing TS	2812870.739	165.753372	166.09375	2684.609375	TRUE	0	2812870.74	-4.62E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_027	10000	Ant Colony Optimizatic	2790981.251	71.5881215	71.34375	2979.230469	TRUE	2768.62793	2790981.251	-2.84E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_027	10000	Genetic Algorithm	3017552363	61.4936777	61.328125	2614.835938	TRUE	0	2790981.251	108017.9733	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_027	10000	LKH	2790981.251	103.0328326	102.453125	2596.925781	TRUE	0	2790981.251	-5.17E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_027	10000	OR-Tools (Routing TS	2790981.253	161.2087395	161.515625	2683.867188	TRUE	0	2790981.251	4.38E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_028	10000	Ant Colony Optimizatic	2786179.349	74.0813366	74.03125	2932.40625	TRUE	2768.62793	2786179.349	-1.17E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_028	10000	Genetic Algorithm	3021697285	62.4243839	62.25	2619.714844	TRUE	0	2786179.349	108353.0788	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_028	10000	LKH	2786179.349	109.3693661	109.46875	2597.996094	TRUE	0	2786179.349	-5.52E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_028	10000	OR-Tools (Routing TS	2786179.354	167.4269147	167.65625	2687.332031	TRUE	0	2786179.349	1.67E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_029	10000	Ant Colony Optimizatic	3003213.927	73.8665052	73.6875	2983.847656	TRUE	2768.62793	3003213.927	-1.55E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_029	10000	Genetic Algorithm	2265197404	61.2418744	60.78125	2620.535156	TRUE	0	3003213.927	75325.77581	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_029	10000	LKH	3003213.927	109.9943697	110.078125	2601.3125	TRUE	0	3003213.927	-5.43E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_029	10000	OR-Tools (Routing TS	3003213.929	165.9831289	166.0625	2690.667969	TRUE	0	3003213.927	5.35E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_030	10000	Ant Colony Optimizatic	3690494.73	74.0118451	73.015625	2986.277344	TRUE	2768.62793	3690494.73	2.65E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_030	10000	Genetic Algorithm	2743560739	61.462664	61.140625	2623.207031	TRUE	0	3690494.73	74241.27238	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_030	10000	LKH	3690494.73	110.8482213	110.859375	2603.777344	TRUE	0	3690494.73	-1.51E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_030	10000	OR-Tools (Routing TS	3690494.728	167.3149112	167.46875	2692.988281	TRUE	0	3690494.73	-6.72E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_031	10000	Ant Colony Optimizatic	13785028.63	73.914451	74.03125	2989.535156	TRUE	2768.62793	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_031	10000	Genetic Algorithm	15214836542	61.3944318	60.953125	2625.136719	TRUE	0	13785028.63	110272.1795	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_031	10000	LKH	13785028.63	109.3911097	109.46875	2606.804688	TRUE	0	13785028.63	-2.03E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}

dataset_name	vertices_count	method	cost	elapsed_s	cpu_time_s	peak_mem_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_10000_031	10000	OR-Tools (Routing TS	13785028.64	166.3126899	165.859375	2695.054688	TRUE	0	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_032	10000	Ant Colony Optimizatic	22212577.06	71.059699	70.921875	2991.054688	TRUE	2768.62793	22204951.63	0.034341156	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_032	10000	Genetic Algorithm	18107899040	61.6394992	61.609375	2628.734375	TRUE	0	22204951.63	81448.92361	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_032	10000	LKH	22209048.93	109.591964	109.0625	2606.699219	TRUE	0	22204951.63	0.018452211	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_032	10000	OR-Tools (Routing TS	22210040.08	158.4346554	158.609375	2690.503906	TRUE	0	22204951.63	0.022915843	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_033	10000	Ant Colony Optimizatic	22890586.24	74.1332098	73.84375	2989.289063	TRUE	2768.62793	22890586.24	1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_033	10000	Genetic Algorithm	25224243248	61.4592599	61.421875	2624.878906	TRUE	0	22890586.24	110094.8329	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_033	10000	LKH	22890586.24	109.8883135	109.671875	2606.078125	TRUE	0	22890586.24	1.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_033	10000	OR-Tools (Routing TS	22890586.24	278.0411178	270.875	2694.3125	TRUE	0	22890586.24	-1.23E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_034	10000	Ant Colony Optimizatic	11201841.75	147.6945523	144.09375	2821.539063	TRUE	2768.62793	11201841.75	-1.83E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_034	10000	Genetic Algorithm	11201841.75	94.9947789	92.875	2427.914063	TRUE	0	11201841.75	1.83E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_034	10000	LKH	11201841.75	160.5657039	155.765625	2574.011719	TRUE	0	11201841.75	-2.33E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_034	10000	OR-Tools (Routing TS	11201841.75	321.9535618	312.5	2656.777344	TRUE	0	11201841.75	1.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_035	10000	Ant Colony Optimizatic	17738469.8	149.6472526	145.265625	2839.542969	TRUE	2768.62793	17738469.8	-2.10E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_035	10000	Genetic Algorithm	17738469.8	96.4445723	94.40625	2436.148438	TRUE	0	17738469.8	5.04E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_035	10000	LKH	17738469.8	157.8203373	153.78125	2608.742188	TRUE	0	17738469.8	-2.31E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_035	10000	OR-Tools (Routing TS	17738469.79	318.1257669	311.234375	2693.972656	TRUE	0	17738469.8	-3.62E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_036	10000	Ant Colony Optimizatic	18085756.55	146.7850297	143.59375	2973.058594	TRUE	2768.62793	17562452.27	2.979676607	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_036	10000	Genetic Algorithm	17562452.27	94.429807	92.765625	2570.503906	TRUE	0	17562452.27	-2.97E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_036	10000	LKH	18085756.55	156.9616725	154.234375	2611.5	TRUE	0	17562452.27	2.979676607	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_036	10000	OR-Tools (Routing TS	18085756.55	241.1316555	236.875	2700.539063	TRUE	0	17562452.27	2.979676624	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_037	10000	Ant Colony Optimizatic	22274779.94	147.1008854	144.28125	2996.710938	TRUE	2768.62793	22274779.94	-2.51E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_037	10000	Genetic Algorithm	22274779.94	93.3992518	92.375	2592.304688	TRUE	0	22274779.94	3.34E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_037	10000	LKH	22274779.94	159.5289928	156.6875	2613.90625	TRUE	0	22274779.94	-2.51E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_037	10000	OR-Tools (Routing TS	22274779.94	316.156705	311.109375	2698.222656	TRUE	0	22274779.94	-7.80E-11	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_038	10000	Ant Colony Optimizatic	30983118.84	141.2832071	139.015625	2907.460938	TRUE	2768.62793	30983118.84	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_038	10000	Genetic Algorithm	30983118.84	91.9336834	90.0625	2491.203125	TRUE	0	30983118.84	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_038	10000	LKH	30983118.84	123.976781	120.921875	2600.070313	TRUE	0	30983118.84	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_038	10000	OR-Tools (Routing TS	30983118.84	302.8586118	290.8125	2691.734375	TRUE	0	30983118.84	1.29E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_039	10000	Ant Colony Optimizatic	28722066.51	112.8692178	110.875	2850.050781	TRUE	2768.62793	28722066.51	-1.30E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_039	10000	Genetic Algorithm	28722066.51	94.369334	91.171875	2417.78125	TRUE	0	28722066.51	3.76E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_039	10000	LKH	28722066.51	156.7961521	152.25	2585.53125	TRUE	0	28722066.51	-1.30E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_039	10000	OR-Tools (Routing TS	28722066.52	309.1601918	302.03125	2703.46875	TRUE	0	28722066.51	6.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_10000_040	10000	Ant Colony Optimizatic	34207221.19	144.2633668	140.8125	2987.027344	TRUE	2768.62793	34207221.19	-2.18E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_10000_040	10000	Genetic Algorithm	34207221.19	91.9205712	89.828125	2583.386719	TRUE	0	34207221.19	2.40E-13	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_10000_040	10000	LKH	34207221.19	157.2127723	153.25	2622.386719	TRUE	0	34207221.19	-2.18E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_10000_040	10000	OR-Tools (Routing TS	34207221.19	320.5237442	308.953125	2677	TRUE	0	34207221.19	1.01E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_001.js	1000	Ant Colony Optimizatic	17385.63477	61.8243089	59.046875	1645.761719	TRUE	27.65478516	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_001.js	1000	Genetic Algorithm	921896.3177	60.1531687	51.59375	1659.871094	TRUE	0	17385.63642	5202.631985	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_001.js	1000	LKH	17385.63642	1.8117372	1.75	1572.265625	TRUE	0	17385.63642	-1.67E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_001.js	1000	OR-Tools (Routing TS	17385.6374	61.2365608	58.6875	1578.667969	TRUE	0	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}

dataset_name	vertices_cost	method	cost	elapsed_s	cpu_time_s	peak_mem_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_1000_002.j	1000	Ant Colony Optimizatic	24628.4668	60.90766	58.65625	1582.800781	TRUE	27.65478516	24628.46877	-8.01E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_002.j	1000	Genetic Algorithm	854417.4	60.1178461	57.46875	822.171875	TRUE	0	24628.46877	3369.226642	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_002.j	1000	LKH	24628.46877	1.9711612	1.734375	822.2734375	TRUE	0	24628.46877	-1.62E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_002.j	1000	OR-Tools (Routing TS	24628.4682	61.2096975	58.765625	822.4570313	TRUE	0	24628.46877	-2.31E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_003.j	1000	Ant Colony Optimizatic	15337.64941	61.888507	61.859375	827.6210938	TRUE	27.65478516	15337.65011	-4.53E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_003.j	1000	Genetic Algorithm	627591.978	60.3375144	57.890625	827.7578125	TRUE	0	15337.65011	3991.83919	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_003.j	1000	LKH	15337.65011	2.0286583	1.921875	220.1992188	TRUE	0	15337.65011	-1.54E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_003.j	1000	OR-Tools (Routing TS	15337.6492	61.1759405	59.296875	220.3203125	TRUE	0	15337.65011	-5.93E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_004.j	1000	Ant Colony Optimizatic	25849.66016	61.4032697	58.9375	220.5625	TRUE	27.65478516	25849.66207	-7.41E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_004.j	1000	Genetic Algorithm	994818.0553	60.0596524	58.15625	220.6445313	TRUE	0	25849.66207	3748.476056	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_004.j	1000	LKH	25849.66207	1.9815979	1.859375	220.7070313	TRUE	0	25849.66207	-1.41E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_004.j	1000	OR-Tools (Routing TS	25849.6618	61.0514329	59.453125	220.8007813	TRUE	0	25849.66207	-1.05E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_005	1000	Ant Colony Optimizatic	28051.75391	61.8922194	60.0625	224.8046875	TRUE	27.65478516	28051.75524	-4.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_005	1000	Genetic Algorithm	869413.5102	60.1629773	58.203125	219.0273438	TRUE	0	28051.75524	2999.319464	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_005	1000	LKH	28051.75524	1.7787535	1.6875	219.1484375	TRUE	0	28051.75524	-6.48E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_005	1000	OR-Tools (Routing TS	28051.7538	61.1583792	58.78125	219.2265625	TRUE	0	28051.75524	-5.15E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_006	1000	Ant Colony Optimizatic	26580.25391	60.7918184	58.90625	219.5117188	TRUE	27.65478516	26580.25646	-9.61E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_006	1000	Genetic Algorithm	707187.8852	60.270041	58.34375	219.5351563	TRUE	0	26580.25646	2560.575654	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_006	1000	LKH	26580.25646	1.9515319	1.796875	212.0273438	TRUE	0	26580.25646	-9.58E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_006	1000	OR-Tools (Routing TS	26580.2574	61.0509918	58.859375	212.1015625	TRUE	0	26580.25646	3.53E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_007	1000	Ant Colony Optimizatic	19326.05078	60.6031101	58.375	216.078125	TRUE	27.65478516	19326.05334	-1.33E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_007	1000	Genetic Algorithm	818913.2453	60.2998128	58.546875	212.2929688	TRUE	0	19326.05334	4137.353746	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_007	1000	LKH	19326.05334	1.7850666	1.6875	212.359375	TRUE	0	19326.05334	-1.88E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_007	1000	OR-Tools (Routing TS	19326.055	61.1836674	58.890625	354.2070313	TRUE	0	19326.05334	8.56E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_008	1000	Ant Colony Optimizatic	21935.49805	60.888127	59	354.3632813	TRUE	27.65478516	21935.49941	-6.20E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_008	1000	Genetic Algorithm	760417.5526	60.2741696	58.34375	346.8164063	TRUE	0	21935.49941	3366.606976	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_008	1000	LKH	21935.49941	2.1810747	2.015625	260.75	TRUE	0	21935.49941	-2.49E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_008	1000	OR-Tools (Routing TS	21935.4992	61.2973984	59.03125	267.046875	TRUE	0	21935.49941	-9.46E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_009	1000	Ant Colony Optimizatic	21815.02148	60.3044741	58.78125	230.1679688	TRUE	27.65478516	21815.02347	-9.09E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_009	1000	Genetic Algorithm	790973.7363	60.1655328	57.796875	197.3125	TRUE	0	21815.02347	3525.821157	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_009	1000	LKH	21815.02347	1.9582147	1.796875	197.4609375	TRUE	0	21815.02347	-2.50E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_009	1000	OR-Tools (Routing TS	21815.0228	61.2107157	58.59375	197.5351563	TRUE	0	21815.02347	-3.06E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_010	1000	Ant Colony Optimizatic	23554.63477	60.6796531	58.734375	201.8164063	TRUE	27.65478516	23554.63619	-6.04E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_010	1000	Genetic Algorithm	941027.0761	60.2024192	58.1875	198.3046875	TRUE	0	23554.63619	3895.082193	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_010	1000	LKH	23554.63619	2.0010754	1.8125	197.4257813	TRUE	0	23554.63619	-6.18E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_010	1000	OR-Tools (Routing TS	23554.6383	61.1354615	59.015625	197.5390625	TRUE	0	23554.63619	8.96E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_011	1000	Ant Colony Optimizatic	134082.3125	61.8271294	60.859375	202.8164063	TRUE	27.65478516	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_011	1000	Genetic Algorithm	6026016.889	60.0732903	58.421875	196.0625	TRUE	0	134082.3155	4394.266725	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_011	1000	LKH	134082.3155	1.994065	1.828125	196.3242188	TRUE	0	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_011	1000	OR-Tools (Routing TS	134082.3163	61.1477527	59.015625	196.7890625	TRUE	0	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_012	1000	Ant Colony Optimizatic	171629.3281	60.2712957	58.703125	205.6445313	TRUE	27.65478516	171629.3454	-1.01E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}

dataset_name	vertices	method	cost	elapsed_s	cpu_time_s	peak_mem_org_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_1000_012	1000	Genetic Algorithm	8470467.526	60.2237143	57.734375	201.609375	TRUE	0	171629.3454	4835.325895	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_012	1000	LKH	171629.3454	1.8659082	1.765625	201.7851563	TRUE	0	171629.3454	-6.78E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_012	1000	OR-Tools (Routing TS	171629.3452	61.0728385	58.9375	201.8671875	TRUE	0	171629.3454	-1.12E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_013	1000	Ant Colony Optimizatic	156214.5313	60.7400303	58.6875	203.1328125	TRUE	27.65478516	155544.7096	0.430629647	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_013	1000	Genetic Algorithm	6322253.657	60.2240264	58.328125	203.15625	TRUE	0	155544.7096	3964.589321	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_013	1000	LKH	156047.9792	2.1995325	2	202.8554688	TRUE	0	155544.7096	0.323552981	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_013	1000	OR-Tools (Routing TS	155544.5723	61.1650079	60.609375	203.6484375	TRUE	0	155544.7096	-8.83E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_014	1000	Ant Colony Optimizatic	198861.3438	61.1658805	59.5625	207.8476563	TRUE	27.65478516	198861.363	-9.67E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_014	1000	Genetic Algorithm	7254643.132	60.2997675	58.09375	203.0117188	TRUE	0	198861.363	3548.090822	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_014	1000	LKH	198861.363	2.049901	1.84375	203.0625	TRUE	0	198861.363	-1.02E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_014	1000	OR-Tools (Routing TS	198861.3633	61.1160435	58.859375	203.3203125	TRUE	0	198861.363	1.64E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_015	1000	Ant Colony Optimizatic	176556.0781	60.4290145	58.765625	207.1640625	TRUE	27.65478516	176238.5431	0.180173435	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_015	1000	Genetic Algorithm	4198687.843	60.489214	58.09375	203.5664063	TRUE	0	176238.5431	2282.389101	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_015	1000	LKH	176279.1031	2.2067064	2.015625	353.3960938	TRUE	0	176238.5431	0.02301428	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_015	1000	OR-Tools (Routing TS	176238.5433	61.3832424	59.625	352.9296875	TRUE	0	176238.5431	1.20E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_016	1000	Ant Colony Optimizatic	180075.125	60.7012629	58.90625	353.0546875	TRUE	27.65478516	180075.1302	-2.91E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_016	1000	Genetic Algorithm	6598762.495	60.3091221	58.171875	353.078125	TRUE	0	180075.1302	3564.449658	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_016	1000	LKH	180075.1302	1.9466703	1.734375	353.0859375	TRUE	0	180075.1302	-1.29E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_016	1000	OR-Tools (Routing TS	180075.1302	61.1081683	59.59375	353.1601563	TRUE	0	180075.1302	-1.98E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_017	1000	Ant Colony Optimizatic	239475.125	60.3833018	56.203125	357.2773438	TRUE	27.65478516	239475.129	-1.67E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_017	1000	Genetic Algorithm	10502709.82	60.5426463	53.71875	200.3046875	TRUE	0	239475.129	4285.7205	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_017	1000	LKH	239475.129	2.853504	2.671875	166.90625	TRUE	0	239475.129	-1.22E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_017	1000	OR-Tools (Routing TS	239475.13	61.9163763	58.453125	180.2734375	TRUE	0	239475.129	4.19E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_018	1000	Ant Colony Optimizatic	189956.0938	65.2663648	63.5	200.3710938	TRUE	27.65478516	189956.1025	-4.63E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_018	1000	Genetic Algorithm	7866892.548	60.2931147	55.859375	284.4179688	TRUE	0	189956.1025	4041.426594	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_018	1000	LKH	189956.1025	2.6043019	2.4375	254.6601563	TRUE	0	189956.1025	-2.30E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_018	1000	OR-Tools (Routing TS	189956.1028	61.5705949	57.40625	268.3242188	TRUE	0	189956.1025	1.34E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_019	1000	Ant Colony Optimizatic	198265.25	61.5375238	59.65625	282.1757813	TRUE	27.65478516	198265.2687	-9.44E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_019	1000	Genetic Algorithm	7285057.173	60.4899446	57.765625	195.9375	TRUE	0	198265.2687	3574.399062	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_019	1000	LKH	198265.2687	2.2439937	2.140625	237.1835938	TRUE	0	198265.2687	-1.76E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_019	1000	OR-Tools (Routing TS	198265.2692	61.8143169	59.25	248.8671875	TRUE	0	198265.2687	2.47E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_020	1000	Ant Colony Optimizatic	243930.6719	68.7388324	63.578125	254.1640625	TRUE	27.65478516	243930.6841	-5.01E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_020	1000	Genetic Algorithm	9117080.852	60.245979	56.96875	266.5546875	TRUE	0	243930.6841	3637.570321	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_020	1000	LKH	243930.6841	2.8155562	2.515625	249.3320313	TRUE	0	243930.6841	-9.54E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_020	1000	OR-Tools (Routing TS	243930.6843	61.5090683	57.5625	260.9804688	TRUE	0	243930.6841	8.84E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_021	1000	Ant Colony Optimizatic	1580438.248	64.8235935	63.4375	204.046875	TRUE	27.65478516	1580438.248	-8.84E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_021	1000	Genetic Algorithm	62024275.91	60.3051827	56.65625	197.1171875	TRUE	0	1580438.248	3824.498537	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_021	1000	LKH	1580438.248	2.7540406	2.65625	169.5078125	TRUE	0	1580438.248	-1.03E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_021	1000	OR-Tools (Routing TS	1580438.248	61.923313	59.140625	180.8203125	TRUE	0	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_022	1000	Ant Colony Optimizatic	2021144.75	66.750856	64.9375	345.890625	TRUE	27.65478516	2021144.849	-4.91E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_022	1000	Genetic Algorithm	89801863.19	60.198152	56.125	342.4609375	TRUE	0	2021144.849	4343.118623	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}

dataset_name	vertices_cost	method	cost	elapsed	cpu_time_s	peak_mem_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_1000_022	1000	LKH	2021144.849	2.6540433	2.4375	232.0976563	TRUE	0	2021144.849	-9.22E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_022	1000	OR-Tools (Routing TS	2021144.851	61.3326319	57.671875	206.2382813	TRUE	0	2021144.849	9.95E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_023	1000	Ant Colony Optimizatic	1892602.25	61.2660225	59.984375	248.9335938	TRUE	27.65478516	1870191.257	1.198326284	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_023	1000	Genetic Algorithm	60226345.37	60.6112495	56.59375	257.1523438	TRUE	0	1870191.257	3120.330817	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_023	1000	LKH	1890092.598	3.1088794	2.796875	188.8125	TRUE	0	1870191.257	1.064134008	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_023	1000	OR-Tools (Routing TS	1870182.428	61.8080029	57.078125	200.625	TRUE	0	1870191.257	-0.000472048	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_024	1000	Ant Colony Optimizatic	2222312.5	65.7898676	62.1875	207.171875	TRUE	27.65478516	2222312.635	-6.08E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_024	1000	Genetic Algorithm	84463733.5	60.3789625	56.59375	207.9648438	TRUE	0	2222312.635	3700.713372	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_024	1000	LKH	2222312.635	2.5157115	2.390625	205.1679688	TRUE	0	2222312.635	-4.19E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_024	1000	OR-Tools (Routing TS	2222312.635	61.6193347	58.15625	210.9140625	TRUE	0	2222312.635	-1.99E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_025	1000	Ant Colony Optimizatic	3172379.25	62.239458	59.4375	213.2265625	TRUE	27.65478516	3170362.756	0.063604518	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_025	1000	Genetic Algorithm	1295397714	60.3360817	53.078125	209.6757813	TRUE	0	3170362.756	3985.960546	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_025	1000	LKH	3170362.756	3.0651852	2.953125	189.671875	TRUE	0	3170362.756	-1.03E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_025	1000	OR-Tools (Routing TS	3170362.754	61.5294903	59.171875	196.4921875	TRUE	0	3170362.756	-6.11E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_026	1000	Ant Colony Optimizatic	2811460.25	64.1613842	60.75	197.28125	TRUE	27.65478516	2811460.487	-8.44E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_026	1000	Genetic Algorithm	129317225.6	60.1812878	56.640625	197.5976563	TRUE	0	2811460.487	4499.645848	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_026	1000	LKH	2811460.487	2.811649	2.59375	197.8789063	TRUE	0	2811460.487	-1.16E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_026	1000	OR-Tools (Routing TS	2811460.488	61.4859013	57.875	198.1875	TRUE	0	2811460.487	4.08E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_027	1000	Ant Colony Optimizatic	2790660	61.7501694	58.6875	201.2578125	TRUE	27.65478516	2790660.312	-1.12E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_027	1000	Genetic Algorithm	138949812.9	60.6250788	57.5625	198.7890625	TRUE	0	2790660.312	4879.101625	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_027	1000	LKH	2790660.312	2.6449143	2.546875	197.6601563	TRUE	0	2790660.312	-6.67E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_027	1000	OR-Tools (Routing TS	2790660.313	61.4816863	56.796875	199.8867188	TRUE	0	2790660.312	1.44E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_028	1000	Ant Colony Optimizatic	2785628	66.3814728	63.171875	203.2148438	TRUE	27.65478516	2785628.043	-1.54E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_028	1000	Genetic Algorithm	83439229.7	60.3866385	55.921875	202.7265625	TRUE	0	2785628.043	2895.347132	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_028	1000	LKH	2785628.043	2.7410267	2.34375	203.2578125	TRUE	0	2785628.043	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_028	1000	OR-Tools (Routing TS	2785628.044	61.4177948	59.203125	354.734375	TRUE	0	2785628.043	3.09E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_029	1000	Ant Colony Optimizatic	3002346.75	61.1818119	59.75	356.234375	TRUE	27.65478516	3002346.933	-6.09E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_029	1000	Genetic Algorithm	84865340.35	60.2474853	58.703125	355.0195313	TRUE	0	3002346.933	2726.633372	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_029	1000	LKH	3002346.933	2.0372161	1.921875	355.296875	TRUE	0	3002346.933	-2.17E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_029	1000	OR-Tools (Routing TS	3002346.933	61.300816	59.453125	355.5	TRUE	0	3002346.933	1.32E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_030	1000	Ant Colony Optimizatic	3689436.5	61.8892924	61.15625	355.8789063	TRUE	27.65478516	3689436.687	-5.08E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_030	1000	Genetic Algorithm	74130337.21	60.3422345	58.859375	355.1953125	TRUE	0	3689436.687	1909.25896	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_030	1000	LKH	3689436.687	1.9749407	1.890625	355.5273438	TRUE	0	3689436.687	-1.39E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_030	1000	OR-Tools (Routing TS	3689436.687	61.4622198	59.65625	356.1289063	TRUE	0	3689436.687	-1.78E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_031	1000	Ant Colony Optimizatic	13782703	61.2825468	60.5	356.5	TRUE	27.65478516	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_031	1000	Genetic Algorithm	420656866	60.1559098	59.390625	356.6757813	TRUE	0	13782703.86	2952.063443	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_031	1000	LKH	13782703.86	1.7960077	1.703125	357.0859375	TRUE	0	13782703.86	-1.22E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_031	1000	OR-Tools (Routing TS	13782703.87	61.2568778	59.8125	357.3515625	TRUE	0	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_032	1000	Ant Colony Optimizatic	22347946	61.0817046	60.203125	361.3085938	TRUE	27.65478516	22191796.81	0.703634743	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_032	1000	Genetic Algorithm	540505012.7	61.0219934	59.546875	357.6015625	TRUE	0	22191796.81	2335.607253	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_032	1000	LKH	22304940.5	1.9635954	1.734375	357.6757813	TRUE	0	22191796.81	0.509844669	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}

dataset_name	vertex_cost	method	cost	elapsed_s	cpu_time_s	peak_mem_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_1000_032	1000	OR-Tools (Routing TS	22191655.36	61.3799356	60.234375	357.8632813	TRUE	0	22191796.81	-0.000637381	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_033	1000	Ant Colony Optimizatic	22888084	60.8062846	60.203125	361.7773438	TRUE	27.65478516	22888085.03	-4.49E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_033	1000	Genetic Algorithm	890260069.8	60.3127488	59.1875	358.0742188	TRUE	0	22888085.03	3789.622346	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_033	1000	LKH	22888085.03	1.8459133	1.765625	358.234375	TRUE	0	22888085.03	-9.77E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_033	1000	OR-Tools (Routing TS	22888085.03	61.3068578	60.078125	358.2890625	TRUE	0	22888085.03	2.63E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_034	1000	Ant Colony Optimizatic	11198574	61.7393666	60.75	362.3046875	TRUE	27.65478516	11198575.24	-1.10E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_034	1000	Genetic Algorithm	341757382.2	60.2072307	59.15625	358.5976563	TRUE	0	11198575.24	2951.79342	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_034	1000	LKH	11198575.24	1.8694403	1.875	358.6992188	TRUE	0	11198575.24	-9.98E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_034	1000	OR-Tools (Routing TS	11198575.24	61.2594709	59.9375	358.7070313	TRUE	0	11198575.24	-5.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_035	1000	Ant Colony Optimizatic	17736308	60.4297235	59.65625	361.9570313	TRUE	27.65478516	17736308.8	-4.48E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_035	1000	Genetic Algorithm	414956799.9	60.0902907	58.78125	358.6757813	TRUE	0	17736308.8	2239.589397	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_035	1000	LKH	17736308.8	1.8084223	1.78125	358.78125	TRUE	0	17736308.8	-8.40E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_035	1000	OR-Tools (Routing TS	17736308.8	61.265108	60.484375	359.328125	TRUE	0	17736308.8	8.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_036	1000	Ant Colony Optimizatic	17569256	62.085577	61.203125	358.3203125	TRUE	27.65478516	17550165.7	0.108775637	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_036	1000	Genetic Algorithm	388224470.9	60.1757171	58.015625	357.1835938	TRUE	0	17550165.7	2112.084362	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_036	1000	LKH	17583661.6	2.048035	1.71875	304.046875	TRUE	0	17550165.7	0.190858032	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_036	1000	OR-Tools (Routing TS	17550163.65	61.1398959	59.703125	304.5976563	TRUE	0	17550165.7	-1.16E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_037	1000	Ant Colony Optimizatic	22283204	61.9012124	60.421875	308.7539063	TRUE	27.65478516	22265828.15	0.078038171	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_037	1000	Genetic Algorithm	567399468.7	60.2776947	58.046875	255.6289063	TRUE	0	22265828.15	2448.29717	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_037	1000	LKH	22283205.28	1.7514717	1.71875	230.8632813	TRUE	0	22265828.15	0.078043935	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_037	1000	OR-Tools (Routing TS	22265828.15	61.2591592	59.515625	231.4375	TRUE	0	22265828.15	-2.67E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_038	1000	Ant Colony Optimizatic	30981012	61.4653463	59.46875	219.3789063	TRUE	27.65478516	30931782.09	0.159156404	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_038	1000	Genetic Algorithm	917120142.2	60.1852954	58.375	207.2695313	TRUE	0	30931782.09	2864.976734	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_038	1000	LKH	30949643.7	2.0524388	1.859375	207.4570313	TRUE	0	30931782.09	0.057745173	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_038	1000	OR-Tools (Routing TS	30931782.09	61.2624511	59.5625	207.59375	TRUE	0	30931782.09	-4.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_039	1000	Ant Colony Optimizatic	28726620	60.3590757	59.109375	211.6445313	TRUE	27.65478516	28704395.5	0.077425415	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_039	1000	Genetic Algorithm	870100230	60.241591	58.265625	207.9414063	TRUE	0	28704395.5	2931.243873	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_039	1000	LKH	28704395.5	1.871462	1.828125	207.40625	TRUE	0	28704395.5	-5.19E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_039	1000	OR-Tools (Routing TS	28704395.5	61.2785573	60.25	207.71875	TRUE	0	28704395.5	1.45E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_1000_040	1000	Ant Colony Optimizatic	34193184	61.4818893	60.125	207.96875	TRUE	27.65478516	34193186.91	-8.50E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_1000_040	1000	Genetic Algorithm	824339830.8	60.0906237	58.453125	207.4804688	TRUE	0	34193186.91	2310.830652	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_1000_040	1000	LKH	34193186.91	2.2720439	2.15625	207.5664063	TRUE	0	34193186.91	-1.31E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_1000_040	1000	OR-Tools (Routing TS	34193186.91	61.1278758	59.328125	207.9140625	TRUE	0	34193186.91	-9.43E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_001	100	Ant Colony Optimizatic	17271.76758	40.3724106	39.359375	208.078125	TRUE	0.4921875	17271.7692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_001	100	Genetic Algorithm	17271.7692	60.2640241	58.453125	207.890625	TRUE	0	17271.7692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_001	100	LKH	17271.7692	0.2056765	0.03125	208.046875	TRUE	0	17271.7692	-4.21E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_001	100	OR-Tools (Routing TS	17271.7687	60.1392275	58.703125	358.8828125	TRUE	0	17271.7692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_002	100	Ant Colony Optimizatic	24343	40.8070658	39.828125	358.9570313	TRUE	0.4921875	24343.00055	-2.25E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_002	100	Genetic Algorithm	24343.00055	60.2804432	58.484375	358.9609375	TRUE	0	24343.00055	-1.49E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_002	100	LKH	24343.00055	0.2021843	0.0625	358.6289063	TRUE	0	24343.00055	1.49E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_002	100	OR-Tools (Routing TS	24343.0004	60.1943153	58.109375	358.6953125	TRUE	0	24343.00055	-6.06E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}

dataset_name	vertices_count	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_100_003.js	100	Ant Colony Optimizatic	15287.15428	41.9775444	41.40625	358.4570313	TRUE	0.422851563	15287.15428	-1.19E-14	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_003.js	100	Genetic Algorithm	15287.15428	60.2108249	58.046875	356.9023438	TRUE	0	15287.15428	-4.76E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_003.js	100	LKH	15287.15428	0.2161665	0.0625	357.0742188	TRUE	0	15287.15428	-2.38E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_003.js	100	OR-Tools (Routing TS	15287.1543	60.2362693	59.0625	357.2773438	TRUE	0	15287.15428	1.45E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_004.js	100	Ant Colony Optimizatic	25789.39453	39.5910063	38	358.890625	TRUE	0.493164063	25789.39624	-6.61E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_004.js	100	Genetic Algorithm	25789.39624	60.1687179	59	358.9609375	TRUE	0	25789.39624	-5.64E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_004.js	100	LKH	25789.39624	0.2183482	0.015625	263.8203125	TRUE	0	25789.39624	-4.23E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_004.js	100	OR-Tools (Routing TS	25789.3962	60.2297272	58.921875	264.046875	TRUE	0	25789.39624	-1.44E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_005	100	Ant Colony Optimizatic	27573.63672	39.6311411	39.25	264.1054688	TRUE	0.493164063	27573.63883	-7.67E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_005	100	Genetic Algorithm	27573.63883	60.0653558	58.75	264.3710938	TRUE	0	27573.63883	-3.96E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_005	100	LKH	27573.63883	0.2101169	0.03125	264.6640625	TRUE	0	27573.63883	-2.64E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_005	100	OR-Tools (Routing TS	27573.6385	60.2152201	58.1875	264.9453125	TRUE	0	27573.63883	-1.21E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_006	100	Ant Colony Optimizatic	26139.40039	41.7300855	40.5625	265.0429688	TRUE	0.495117188	26139.40146	-4.09E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_006	100	Genetic Algorithm	26139.40146	60.2045469	57.140625	191.2304688	TRUE	0	26139.40146	-5.57E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_006	100	LKH	26139.40146	0.210194	0.03125	191.28125	TRUE	0	26139.40146	-5.57E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_006	100	OR-Tools (Routing TS	26139.4018	60.1698676	58.25	191.3476563	TRUE	0	26139.40146	1.30E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_007	100	Ant Colony Optimizatic	19165.10547	43.5365455	42.890625	191.5585938	TRUE	0.494140625	19165.10833	-1.49E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_007	100	Genetic Algorithm	19165.10833	60.2857771	55.859375	191.1796875	TRUE	0	19165.10833	-5.69E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_007	100	LKH	19165.10833	0.2134853	0.0625	190.7617188	TRUE	0	19165.10833	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_007	100	OR-Tools (Routing TS	19165.1082	60.0665672	58.515625	190.9804688	TRUE	0	19165.10833	-6.71E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_008	100	Ant Colony Optimizatic	21850.4707	38.6845249	37.625	189.265625	TRUE	0.492675781	21850.47194	-5.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_008	100	Genetic Algorithm	21850.47194	60.180729	57.671875	189.5507813	TRUE	0	21850.47194	-6.66E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_008	100	LKH	21850.47194	0.2134999	0.015625	189.71875	TRUE	0	21850.47194	-3.33E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_008	100	OR-Tools (Routing TS	21850.4718	60.220979	58.3125	190.1210938	TRUE	0	21850.47194	-6.27E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_009	100	Ant Colony Optimizatic	21583.62109	37.3725007	36.859375	190.2773438	TRUE	0.4921875	21583.62332	-1.03E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_009	100	Genetic Algorithm	21583.62332	60.1711484	58.125	190.46875	TRUE	0	21583.62332	-5.06E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_009	100	LKH	21583.62332	0.2022801	0.015625	190.7109375	TRUE	0	21583.62332	-3.37E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_009	100	OR-Tools (Routing TS	21583.6235	60.2371771	58.0625	191.1367188	TRUE	0	21583.62332	8.29E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_010	100	Ant Colony Optimizatic	23300.55859	37.3385007	36.53125	191.3632813	TRUE	0.4921875	23300.55942	-3.53E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_010	100	Genetic Algorithm	23300.55942	60.3328217	58.328125	191.4570313	TRUE	0	23300.55942	-4.68E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_010	100	LKH	23300.55942	0.2077118	0.046875	191.6484375	TRUE	0	23300.55942	-3.12E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_010	100	OR-Tools (Routing TS	23300.5594	60.1637934	58.140625	191.9609375	TRUE	0	23300.55942	-7.07E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_011	100	Ant Colony Optimizatic	132540.1563	37.0968069	36.46875	193.1679688	TRUE	0.497558594	132177.1632	0.27462617	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_011	100	Genetic Algorithm	132177.1632	60.1836096	58.109375	343.6679688	TRUE	0	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_011	100	LKH	132177.1632	0.211171	0.03125	343.6757813	TRUE	0	132177.1632	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_011	100	OR-Tools (Routing TS	132177.1633	60.1483744	58.609375	343.7421875	TRUE	0	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_012	100	Ant Colony Optimizatic	171428.4375	37.5968573	36.421875	343.7695313	TRUE	0.4921875	170608.5212	0.48058345	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_012	100	Genetic Algorithm	170608.5212	60.1961365	57.578125	343.8242188	TRUE	0	170608.5212	-3.41E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_012	100	LKH	170608.5212	0.211836	0.03125	343.4375	TRUE	0	170608.5212	-1.71E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_012	100	OR-Tools (Routing TS	170608.5212	60.1560116	57.765625	343.6757813	TRUE	0	170608.5212	1.01E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_013	100	Ant Colony Optimizatic	154204.4688	37.4857798	36.578125	345.1328125	TRUE	0.496582031	153571.0197	0.412479572	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}

dataset_name	vertex_count	method	cost	elapsed_s	cpu_time_s	peak_mem_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_100_013	100	Genetic Algorithm	153569.688	60.2588186	58.078125	345.2226563	TRUE	0	153571.0197	-0.0008671151	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_013	100	LKH	153571.6894	0.2078304	0.03125	345.40625	TRUE	0	153571.0197	0.000436133	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_013	100	OR-Tools (Routing TS	153569.6877	60.0744048	58.609375	345.5507813	TRUE	0	153571.0197	-0.000867329	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_014	100	Ant Colony Optimizatic	196738.4531	37.7927321	37.078125	345.6796875	TRUE	0.493164063	195239.8058	0.767593133	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_014	100	Genetic Algorithm	195239.8058	60.1689139	58.546875	345.7617188	TRUE	0	195239.8058	-2.98E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_014	100	LKH	195239.8058	0.2209798	0.046875	345.9296875	TRUE	0	195239.8058	-2.98E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_014	100	OR-Tools (Routing TS	195239.806	60.0968263	58.1875	346.0117188	TRUE	0	195239.8058	1.11E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_015	100	Ant Colony Optimizatic	174617.125	37.3365188	37	346.1132813	TRUE	0.494628906	173346.912	0.732757801	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_015	100	Genetic Algorithm	173346.912	60.1987683	58.609375	346.1132813	TRUE	0	173346.912	-5.04E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_015	100	LKH	174769.4692	0.206833	0.03125	346.2773438	TRUE	0	173346.912	0.820641773	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_015	100	OR-Tools (Routing TS	173346.9121	60.2188229	58.5625	346.5585938	TRUE	0	173346.912	6.95E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_016	100	Ant Colony Optimizatic	178090.6719	37.3297795	36.75	346.5976563	TRUE	0.4921875	178090.684	-6.82E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_016	100	Genetic Algorithm	178090.684	56.0499113	54.75	347.6796875	TRUE	0	178090.684	-8.17E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_016	100	LKH	178090.684	0.2108603	0.03125	347.8671875	TRUE	0	178090.684	-6.54E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_016	100	OR-Tools (Routing TS	178090.6835	60.1538536	60.234375	348.2890625	TRUE	0	178090.684	-2.91E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_017	100	Ant Colony Optimizatic	236659.5938	19.4407787	18.640625	348.3046875	TRUE	0.4921875	236659.6076	-5.83E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_017	100	Genetic Algorithm	236659.6076	34.2600044	34.28125	348.3125	TRUE	0	236659.6076	-1.23E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_017	100	LKH	236659.6076	0.2050168	0.015625	348.265625	TRUE	0	236659.6076	-1.23E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_017	100	OR-Tools (Routing TS	236659.6074	60.1803305	60.5	348.59375	TRUE	0	236659.6076	-6.35E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_018	100	Ant Colony Optimizatic	186711.3125	19.3849642	18.59375	348.6015625	TRUE	0.495605469	186711.3214	-4.78E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_018	100	Genetic Algorithm	186711.3214	34.8824875	35.046875	348.6054688	TRUE	0	186711.3214	-3.12E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_018	100	LKH	186711.3214	0.2031806	0.015625	348.625	TRUE	0	186711.3214	-1.56E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_018	100	OR-Tools (Routing TS	186711.3215	60.1698412	60.484375	348.6484375	TRUE	0	186711.3214	4.41E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_019	100	Ant Colony Optimizatic	197091.6719	19.0071135	18.515625	347.8476563	TRUE	0.4921875	197091.6815	-4.87E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_019	100	Genetic Algorithm	197091.6815	34.0595036	34.125	347.9453125	TRUE	0	197091.6815	-2.95E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_019	100	LKH	197091.6815	0.2089522	0.015625	348.2070313	TRUE	0	197091.6815	-1.48E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_019	100	OR-Tools (Routing TS	197091.6812	60.1848223	60.609375	348.4609375	TRUE	0	197091.6815	-1.42E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_020	100	Ant Colony Optimizatic	241667.2188	29.2474002	28.296875	348.5039063	TRUE	0.496582031	241667.2366	-7.40E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_020	100	Genetic Algorithm	241667.2366	39.7181069	39.3125	348.5546875	TRUE	0	241667.2366	-3.61E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_020	100	LKH	241667.2366	0.2094638	0.03125	342.4726563	TRUE	0	241667.2366	-2.41E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_020	100	OR-Tools (Routing TS	241667.2365	60.1289061	59.109375	342.5625	TRUE	0	241667.2366	-5.19E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_021	100	Ant Colony Optimizatic	1569747.75	33.6680718	33.390625	339.6210938	TRUE	0.4921875	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_021	100	Genetic Algorithm	1569747.854	60.2665086	58.390625	339.9648438	TRUE	0	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_021	100	LKH	1569747.854	0.2072495	0.046875	339.7070313	TRUE	0	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_021	100	OR-Tools (Routing TS	1569747.854	60.1328818	52.109375	340.4960938	TRUE	0	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_022	100	Ant Colony Optimizatic	2008644.75	59.2838528	56.59375	180.5585938	TRUE	0.497558594	2008644.896	-7.28E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_022	100	Genetic Algorithm	2008644.896	44.1870991	43.359375	182.03125	TRUE	0	2008644.896	-4.64E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_022	100	LKH	2024560.673	0.2088492	0.015625	182.2734375	TRUE	0	2008644.896	0.792363889	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_022	100	OR-Tools (Routing TS	2008644.896	60.1595807	58.65625	188.8164063	TRUE	0	2008644.896	3.06E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_023	100	Ant Colony Optimizatic	1861924.625	33.9259232	31.921875	190.6484375	TRUE	0.495117188	1858425.324	0.188293865	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_023	100	Genetic Algorithm	1858273.839	44.7469858	43.140625	190.8671875	TRUE	0	1858425.324	-0.008151256	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}

dataset_name	vertex_count	method	cost	elapsed_s	cpu_time_s	peak_mem_org_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_100_023	100	LKH	1868302.561	0.208789	0.03125	191.1367188	TRUE	0	1858425.324	0.531484204	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_023	100	OR-Tools (Routing TS	1858273.839	60.0652053	58.171875	191.6132813	TRUE	0	1858425.324	-0.008151258	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_024	100	Ant Colony Optimizatic	2230216	33.8205468	31.953125	192.4726563	TRUE	0.495605469	2212942.385	0.780572288	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_024	100	Genetic Algorithm	2212942.385	43.3273346	41.828125	192.609375	TRUE	0	2212942.385	-4.21E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_024	100	LKH	2241565.59	0.2176303	0.015625	192.6640625	TRUE	0	2212942.385	1.293445554	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_024	100	OR-Tools (Routing TS	2212942.385	60.1159525	57.859375	192.796875	TRUE	0	2212942.385	-1.32E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_025	100	Ant Colony Optimizatic	3192221	33.9554692	31.703125	192.4140625	TRUE	0.497558594	3152611.492	1.256403094	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_025	100	Genetic Algorithm	3152611.492	44.2866452	42.953125	192.5507813	TRUE	0	3152611.492	-1.48E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_025	100	LKH	3175958.36	0.2149719	0.015625	192.7890625	TRUE	0	3152611.492	0.740556481	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_025	100	OR-Tools (Routing TS	3152611.492	60.0764323	58.328125	193.0195313	TRUE	0	3152611.492	3.58E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_026	100	Ant Colony Optimizatic	2780963	32.9441259	30.84375	193.0742188	TRUE	0.493164063	2771965.659	0.324583428	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_026	100	Genetic Algorithm	2771965.659	44.155314	42.421875	193.1679688	TRUE	0	2771965.659	-6.72E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_026	100	LKH	2807986.006	0.2121543	0.015625	193.390625	TRUE	0	2771965.659	1.299451403	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_026	100	OR-Tools (Routing TS	2771965.659	60.2088687	58.65625	193.6875	TRUE	0	2771965.659	-5.24E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_027	100	Ant Colony Optimizatic	2771230	33.6986706	31.71875	193.8828125	TRUE	0.493652344	2771230.212	-7.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_027	100	Genetic Algorithm	2771230.212	43.2993618	41.40625	194.0039063	TRUE	0	2771230.212	-5.04E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_027	100	LKH	2771230.212	0.2058632	0.03125	194.1953125	TRUE	0	2771230.212	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_027	100	OR-Tools (Routing TS	2771230.212	60.1333358	57.65625	194.3515625	TRUE	0	2771230.212	-2.02E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_028	100	Ant Colony Optimizatic	2756474.25	34.5789701	32.734375	194.6367188	TRUE	0.503417969	2756474.389	-5.03E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_028	100	Genetic Algorithm	2756474.389	44.1222505	43	195.015625	TRUE	0	2756474.389	-5.07E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_028	100	LKH	2770048.558	0.2097295	0.015625	195.1210938	TRUE	0	2756474.389	0.492446759	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_028	100	OR-Tools (Routing TS	2756474.389	60.1963205	58.5	195.421875	TRUE	0	2756474.389	1.28E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_029	100	Ant Colony Optimizatic	2935457	34.1702784	32.234375	345.625	TRUE	0.492675781	2935457.334	-1.14E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_029	100	Genetic Algorithm	2935457.334	45.873551	44.1875	345.6796875	TRUE	0	2935457.334	-1.59E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_029	100	LKH	2935457.334	0.2172332	0.015625	345.6757813	TRUE	0	2935457.334	1.59E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_029	100	OR-Tools (Routing TS	2935457.334	60.0908989	58.375	345.6914063	TRUE	0	2935457.334	-2.33E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_030	100	Ant Colony Optimizatic	3622610	34.1979013	32.328125	345.796875	TRUE	0.497070313	3623360.093	-0.020701573	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_030	100	Genetic Algorithm	3604010.095	46.2333514	44.953125	345.90625	TRUE	0	3623360.093	-0.534034619	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_030	100	LKH	3677934.626	0.2114784	0.03125	346.1054688	TRUE	0	3623360.093	1.506185754	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_030	100	OR-Tools (Routing TS	3604010.095	60.2204916	58.09375	345.5898438	TRUE	0	3623360.093	-0.534034619	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_031	100	Ant Colony Optimizatic	13666452	34.5658731	32.859375	345.6015625	TRUE	0.498535156	13666452.32	-2.37E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_031	100	Genetic Algorithm	13666452.32	49.9238912	48.78125	345.6992188	TRUE	0	13666452.32	1.36E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_031	100	LKH	13891804.01	0.2064557	0.046875	345.984375	TRUE	0	13666452.32	1.648940633	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_031	100	OR-Tools (Routing TS	13666452.32	60.2140415	58.265625	346.3046875	TRUE	0	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_032	100	Ant Colony Optimizatic	21882874	35.45481	33.40625	346.3085938	TRUE	0.49609375	21865597.52	0.079012148	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_032	100	Genetic Algorithm	21855318.03	46.1106799	44.8125	346.3085938	TRUE	0	21865597.52	-0.047012177	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_032	100	LKH	22084272.88	0.2112813	0.015625	346.4453125	TRUE	0	21865597.52	1.000088653	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_032	100	OR-Tools (Routing TS	21855318.03	60.1212261	58.0625	346.5507813	TRUE	0	21865597.52	-0.047012175	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_033	100	Ant Colony Optimizatic	22682548	33.5980805	32.109375	346.6640625	TRUE	0.492675781	22682549.13	-5.00E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_033	100	Genetic Algorithm	22682549.13	45.9980336	44.796875	346.7539063	TRUE	0	22682549.13	-3.28E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_033	100	LKH	22682549.13	0.2154956	0.015625	343.6445313	TRUE	0	22682549.13	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}

dataset_name	vertex_count	method	cost	elapsed_s	cpu_time_s	peak_mem_org_mb	used_gpu	peak_gpu_memory_mb	optimal_cost	relative_error_percent	params
spline_100_033	100	OR-Tools (Routing TS	22682549.13	60.2282467	58.140625	344.1367188	TRUE	0	22682549.13	-2.94E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_034	100	Ant Colony Optimizatic	11085362	33.6522704	31.5625	344.1914063	TRUE	0.492675781	11006726.13	0.714434663	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_034	100	Genetic Algorithm	11006726.13	44.3942231	43.015625	344.3046875	TRUE	0	11006726.13	-6.77E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_034	100	LKH	11091167.96	0.2056193	0.03125	344.6835938	TRUE	0	11006726.13	0.767183882	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_034	100	OR-Tools (Routing TS	11006726.13	60.1354113	58.015625	344.7070313	TRUE	0	11006726.13	-3.47E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_035	100	Ant Colony Optimizatic	17570856	35.3721187	33.9375	344.7109375	TRUE	0.4921875	17570856.69	-3.95E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_035	100	Genetic Algorithm	17570856.69	45.3423604	43.765625	344.7226563	TRUE	0	17570856.69	-2.12E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_035	100	LKH	17570856.69	0.2079264	0.015625	344.890625	TRUE	0	17570856.69	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_035	100	OR-Tools (Routing TS	17570856.69	60.1889457	58.4375	345.1835938	TRUE	0	17570856.69	-7.71E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_036	100	Ant Colony Optimizatic	17210396	34.3199816	32.34375	345.2773438	TRUE	0.49609375	17133730.9	0.447451287	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_036	100	Genetic Algorithm	17125000.26	46.1975785	44.8125	345.5117188	TRUE	0	17133730.9	-0.050955877	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_036	100	LKH	17125000.26	0.2127193	0.015625	344.859375	TRUE	0	17133730.9	-0.050955877	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_036	100	OR-Tools (Routing TS	17125000.26	60.0567064	58.1875	345.1328125	TRUE	0	17133730.9	-0.050955875	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_037	100	Ant Colony Optimizatic	21674832	33.5829132	32.15625	345.15625	TRUE	0.497558594	21674834.49	-1.15E-05	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_037	100	Genetic Algorithm	21674834.49	45.5852272	44.1875	345.296875	TRUE	0	21674834.49	-3.44E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_037	100	LKH	22013751.99	0.2115826	0.046875	345.3515625	TRUE	0	21674834.49	1.563645181	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_037	100	OR-Tools (Routing TS	21674834.49	60.2192979	57.859375	345.7773438	TRUE	0	21674834.49	-8.40E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_038	100	Ant Colony Optimizatic	30087336	34.5445211	32.390625	345.8789063	TRUE	0.497558594	29453205.34	2.153010702	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_038	100	Genetic Algorithm	29420336.77	46.069723	44.5625	346.0195313	TRUE	0	29453205.34	-0.111595886	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_038	100	LKH	30342499.37	0.2167926	0.015625	346.2304688	TRUE	0	29453205.34	3.019345501	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_038	100	OR-Tools (Routing TS	29420336.77	60.0417467	58.25	346.3867188	TRUE	0	29453205.34	-0.111595885	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_039	100	Ant Colony Optimizatic	27972592	36.1196501	34.609375	346.3789063	TRUE	0.49609375	27896277.3	0.273565894	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_039	100	Genetic Algorithm	27896277.3	47.9109905	46.625	346.4296875	TRUE	0	27896277.3	-4.01E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_039	100	LKH	28207975.83	0.2089089	0.015625	346.6210938	TRUE	0	27896277.3	1.117348118	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_039	100	OR-Tools (Routing TS	27896277.3	60.1484386	59.5625	346.9453125	TRUE	0	27896277.3	8.12E-12	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}
spline_100_040	100	Ant Colony Optimizatic	33584232	39.664699	37.71875	347.1992188	TRUE	0.493652344	33418306.85	0.496509755	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 42, "time_limit_sec": 60}
spline_100_040	100	Genetic Algorithm	33418306.85	45.1281887	43.796875	347.2109375	TRUE	0	33418306.85	-4.46E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 42, "time_limit_sec": 60}
spline_100_040	100	LKH	33594623.7	0.2128398	0.015625	347.2382813	TRUE	0	33418306.85	0.527605585	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 42, "time_limit_sec": 60}
spline_100_040	100	OR-Tools (Routing TS	33418306.85	60.2090011	57.890625	347.4882813	TRUE	0	33418306.85	1.40E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 42, "time_limit_sec": 60}

2 priedas. Stabilumo eksperimentų rezultatai

dataset_nam	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_co	relative_error_percent	params
spline_100_001	Genetic Algorithm	172717692	49.273	35.266	449.605	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	33.268	32.797	450.008	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	32.910	32.625	450.016	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	33.605	33.328	449.727	0.000	172717692	-4.21E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	33.559	33.063	449.773	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	33.188	32.750	449.750	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	33.581	33.250	449.891	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	34.812	34.641	440.262	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	35.019	34.859	440.801	0.000	172717692	-6.32E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_100_001	Genetic Algorithm	172717692	34.924	34.844	441.199	0.000	172717692	-4.21E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.334	18.172	671.910	0.493	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.574	19.734	673.074	0.492	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	19.510	19.094	673.281	0.493	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.005	19.172	673.324	0.492	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.927	19.328	673.328	0.493	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.375	19.125	673.363	0.494	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.918	19.547	673.664	0.492	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	21.155	19.578	673.719	0.492	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	19.912	18.891	673.738	0.493	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_100_001	Ant Colony Optimization	172717.76758	20.562	19.094	673.754	0.493	172717692	-9.38E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.209	0.031	674.051	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.203	0.016	674.090	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.215	0.016	674.117	0.000	172717692	-4.21E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.215	0.000	674.133	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.215	0.031	674.164	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.216	0.016	674.176	0.000	172717692	-4.21E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.201	0.016	674.191	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.216	0.016	674.488	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.214	0.016	674.504	0.000	172717692	-4.21E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_100_001	LKH	172717692	0.211	0.031	674.539	0.000	172717692	-6.32E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.221	59.672	679.492	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.210	59.906	680.262	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.018	59.719	679.246	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.222	59.344	679.379	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.053	59.719	679.375	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.127	59.500	679.422	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.064	59.734	679.469	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.099	59.313	679.563	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.083	59.672	679.559	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_100_001	OR-Tools (Routing TSP)	172717687	60.017	60.016	679.563	0.000	172717692	-2.88E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	35.609	35.141	672.402	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	35.868	35.609	672.579	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	35.751	35.375	672.688	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	35.713	35.516	673.383	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	36.104	35.906	673.461	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	36.337	36.141	673.777	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	36.241	35.953	673.816	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	35.221	35.141	673.984	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	36.135	35.734	674.020	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_100_011	Genetic Algorithm	132177.1632	35.981	35.750	674.035	0.000	132177.1632	-2.20E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132540.1563	22.389	20.688	674.094	0.502	132177.1632	0.27462617	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132177.1563	19.950	19.078	674.203	0.495	132177.1632	-5.23E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132177.1563	19.823	19.000	674.266	0.500	132177.1632	-5.23E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132177.1563	19.371	18.813	674.309	0.494	132177.1632	-5.23E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132177.1563	19.778	19.078	674.555	0.500	132177.1632	-5.23E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132177.1563	19.524	18.750	674.605	0.497	132177.1632	-5.23E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132540.1563	19.932	18.859	674.629	0.500	132177.1632	0.27462617	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}

dataset_nam	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_cost	relative_error_percent	params
spline_100_011	Ant Colony Optimization	132540.1563	19.382	18.813	674.695	0.494	132177.1632	0.27462617	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132177.1563	19.554	18.625	674.711	0.495	132177.1632	-5.23E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_100_011	Ant Colony Optimization	132540.1563	19.544	19.063	674.711	0.495	132177.1632	0.27462617	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9325, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.204	0.016	674.723	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.206	0.016	674.805	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.216	0.016	674.809	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.215	0.031	674.809	0.000	132177.1632	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.215	0.031	674.809	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.213	0.016	674.805	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.214	0.016	674.809	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.206	0.047	674.820	0.000	132177.1632	0	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.210	0.016	674.832	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_100_011	LKH	132177.1632	0.210	0.047	674.932	0.000	132177.1632	-2.20E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.014	59.469	674.832	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.164	59.625	674.973	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.209	59.625	675.043	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.089	59.453	675.031	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.178	59.578	675.031	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.093	59.563	675.086	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.173	59.625	674.516	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.214	59.734	675.207	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.046	59.391	675.367	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_100_011	OR-Tools (Routing TSP)	132177.1633	60.202	59.594	675.445	0.000	132177.1632	9.90E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.413	33.109	675.461	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	34.014	33.750	675.523	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.260	32.969	675.523	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.856	33.563	675.563	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.391	33.141	675.563	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.566	33.438	675.563	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.619	33.328	675.547	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.338	33.172	675.551	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.545	33.219	675.570	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_100_021	Genetic Algorithm	1569747.854	33.712	33.328	675.570	0.000	1569747.854	-4.45E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.778	18.922	675.578	0.493	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.782	18.969	675.637	0.492	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.952	18.844	675.641	0.492	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.771	18.766	675.660	0.493	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.784	18.906	675.660	0.493	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.581	18.688	675.660	0.493	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.525	18.922	675.664	0.492	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.576	19.125	675.660	0.492	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.232	18.688	675.719	0.495	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_100_021	Ant Colony Optimization	1569747.75	19.938	18.719	675.961	0.494	1569747.854	-6.65E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.208	0.016	675.965	0.000	1569747.854	-1.48E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.204	0.031	675.965	0.000	1569747.854	-1.48E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.207	0.016	675.965	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.203	0.031	675.969	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.215	0.000	675.973	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.217	0.000	675.977	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.216	0.016	676.035	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.210	0.000	676.039	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.212	0.000	676.039	0.000	1569747.854	-2.97E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_100_021	LKH	1569747.854	0.209	0.000	676.039	0.000	1569747.854	-1.48E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.085	59.484	676.305	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.069	59.750	676.617	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.079	59.672	676.523	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.021	59.969	676.824	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC","local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}

dataset_nam	method	cost	elapsed_s	cpu_time_s	peak_mem_morq_mb	memory_mb	optimal_co	relative_error_percent	params
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.050	59.797	676.820	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.017	60.156	676.824	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.183	59.953	676.852	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.131	59.766	676.977	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.152	59.734	677.023	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_100_021	OR-Tools (Routing TSP)	1569747.854	60.164	59.944	677.398	0.000	1569747.854	-1.91E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.046	32.922	677.387	0.000	13666452.32	-1.36E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.140	33.047	677.438	0.000	13666452.32	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.573	33.359	677.461	0.000	13666452.32	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.429	33.344	677.793	0.000	13666452.32	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.094	33.016	678.133	0.000	13666452.32	1.36E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	32.968	32.766	678.133	0.000	13666452.32	-1.36E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.577	33.281	678.137	0.000	13666452.32	-1.36E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.048	32.766	678.133	0.000	13666452.32	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.651	33.516	678.121	0.000	13666452.32	0	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_100_031	Genetic Algorithm	13666452.32	33.321	33.078	678.469	0.000	13666452.32	1.36E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13761997	18.719	17.750	678.742	0.496	13666452.32	0.699118353	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13728417	18.212	17.984	678.879	0.496	13666452.32	0.453407182	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13731826	18.353	17.750	678.910	0.496	13666452.32	0.478351475	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13666452	18.180	17.656	678.535	0.495	13666452.32	-2.37E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13666452	18.042	17.266	678.543	0.497	13666452.32	-2.37E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13728417	18.343	17.609	678.590	0.497	13666452.32	0.453407182	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13728417	18.325	17.891	678.754	0.495	13666452.32	0.453407182	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13666452	18.094	17.438	678.979	0.499	13666452.32	-2.37E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13666452	18.311	17.672	678.906	0.497	13666452.32	-2.37E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_100_031	Ant Colony Optimization	13666452	18.349	17.828	678.957	0.498	13666452.32	-2.37E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_100_031	LKH	13666452.32	0.201	0.031	678.969	0.000	13666452.32	5.45E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_100_031	LKH	13776224.83	0.207	0.016	678.969	0.000	13666452.32	0.803226053	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_100_031	LKH	13666452.32	0.216	0.016	679.352	0.000	13666452.32	5.45E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_100_031	LKH	13891804.01	0.213	0.016	679.352	0.000	13666452.32	1.648940633	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_100_031	LKH	13891804.01	0.215	0.016	679.375	0.000	13666452.32	1.648940633	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_100_031	LKH	13776224.83	0.214	0.016	679.398	0.000	13666452.32	0.803226053	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_100_031	LKH	13891804.01	0.212	0.016	679.391	0.000	13666452.32	1.648940633	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_100_031	LKH	13728417.73	0.207	0.016	679.395	0.000	13666452.32	0.453412502	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_100_031	LKH	13891804.01	0.204	0.016	679.395	0.000	13666452.32	1.648940633	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_100_031	LKH	13891804.01	0.209	0.000	679.793	0.000	13666452.32	1.648940633	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.197	59.797	679.891	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.026	59.750	680.031	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.099	59.766	680.461	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.100	59.656	680.613	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.024	59.672	680.633	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.210	59.922	680.668	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.099	59.859	680.684	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.172	59.750	680.770	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.066	59.813	680.855	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_100_031	OR-Tools (Routing TSP)	13666452.32	60.153	59.578	680.867	0.000	13666452.32	2.53E-10	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	311858.482	60.461	60.313	703.766	0.000	17385.63642	1693.770872	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	460123.7594	54.070	53.656	703.188	0.000	17385.63642	2546.574151	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	455876.5631	56.910	56.422	704.031	0.000	17385.63642	2522.14481	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	379325.1481	55.422	55.203	703.863	0.000	17385.63642	2081.830673	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	444032.6844	40.561	40.453	704.109	0.000	17385.63642	2454.02019	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	411741.4679	40.254	39.953	704.348	0.000	17385.63642	2268.285278	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	373902.3423	43.851	43.688	704.488	0.000	17385.63642	2050.633374	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	361615.5885	48.994	48.672	704.652	0.000	17385.63642	1979.967507	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	378653.5407	60.075	59.859	704.723	0.000	17385.63642	2077.967671	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_001	sor Genetic Algorithm	429148.9418	46.044	45.703	705.176	0.000	17385.63642	2368.410884	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_001	sor Ant Colony Optimization	17385.63477	60.346	58.938	705.766	29.348	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}

dataset_name	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_cost	relative_error_percent	params
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.555	59.438	710.035	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.996	59.234	706.219	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.754	59.516	706.215	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.717	59.422	706.215	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.741	59.609	706.215	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.234	58.959	710.031	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.589	58.625	706.195	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.464	58.656	706.215	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_1000_001.jsor	Ant Colony Optimization	17385.63477	60.143	58.766	705.953	28.600	17385.63642	-9.52E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.045	0.906	705.953	0.000	17385.63642	-1.05E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.043	0.859	705.953	0.000	17385.63642	-1.67E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.050	0.906	705.961	0.000	17385.63642	-1.67E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.039	0.953	705.965	0.000	17385.63642	-1.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	0.928	0.891	705.973	0.000	17385.63642	-1.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.108	0.953	705.969	0.000	17385.63642	-1.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.042	1.000	705.973	0.000	17385.63642	-1.05E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.034	0.891	705.973	0.000	17385.63642	-1.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.039	0.906	706.113	0.000	17385.63642	-1.46E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_1000_001.jsor	LKH	17385.63642	1.059	0.922	706.137	0.000	17385.63642	-1.26E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.752	60.344	711.848	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.630	60.422	712.430	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.638	60.375	712.469	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.670	60.531	712.531	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.636	60.250	712.566	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.598	60.453	712.617	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.709	60.281	713.023	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.684	60.453	713.012	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.632	60.328	713.012	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_001.jsor	OR-Tools (Routing TSP)	17385.6374	60.669	60.344	713.020	0.000	17385.63642	5.63E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	2996447.289	60.258	59.984	713.020	0.000	134082.3155	2134.781879	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	5097324.649	36.098	35.891	713.406	0.000	134082.3155	3702.085779	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	5612909.959	40.074	39.797	713.934	0.000	134082.3155	4086.167235	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	2672915.769	52.108	51.719	714.211	0.000	134082.3155	1893.489671	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	5549886.292	36.315	36.188	714.891	0.000	134082.3155	4039.238104	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	4354839.106	38.263	38.000	714.531	0.000	134082.3155	3147.884772	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	3831623.291	44.955	45.031	715.242	0.000	134082.3155	2757.664918	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	3137736.594	43.991	44.031	716.715	0.000	134082.3155	2240.156927	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	3377578.749	45.667	45.453	717.281	0.000	134082.3155	2419.033727	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_011	Genetic Algorithm	3413708.475	44.064	43.859	717.742	0.000	134082.3155	2445.979656	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.362	58.531	718.867	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	61.072	59.172	718.875	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.231	58.578	719.191	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	61.041	59.500	719.711	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.554	59.563	719.703	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.941	59.563	719.699	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	61.040	58.406	720.273	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.743	58.969	720.313	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.609	58.969	720.324	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_1000_011	Ant Colony Optimization	134082.3125	60.641	59.031	720.313	28.600	134082.3155	-2.22E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.044	0.938	720.336	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.039	0.922	720.340	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.045	0.859	720.352	0.000	134082.3155	-1.52E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.066	0.922	720.348	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.054	0.891	720.566	0.000	134082.3155	-1.30E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.054	0.938	720.602	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.075	0.969	720.695	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	0.907	0.906	720.898	0.000	134082.3155	-1.30E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}

dataset_name	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_cost	relative_error_percent	params
spline_1000_011	LKH	134082.3155	1.056	1.000	720.961	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_1000_011	LKH	134082.3155	1.042	0.938	721.215	0.000	134082.3155	-8.68E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.766	60.563	721.672	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.703	60.516	721.695	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.631	60.250	722.035	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.622	60.422	723.227	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.713	60.266	723.219	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.723	60.453	723.484	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.733	60.375	723.504	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.593	60.547	723.512	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.692	60.594	723.766	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_011	OR-Tools (Routing TSP)	134082.3163	60.754	60.313	723.750	0.000	134082.3155	6.16E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	519747777.41	42.330	42.109	723.746	0.000	1580438.248	3188.6307	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	350502009.15	42.217	42.000	724.000	0.000	1580438.248	2117.752525	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	53147895.5	40.846	40.641	724.262	0.000	1580438.248	3262.858092	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	33206543.48	60.200	59.766	724.816	0.000	1580438.248	2001.097182	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	58131061.7	38.630	38.547	725.484	0.000	1580438.248	3578.160902	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	41022647.33	44.893	44.703	725.801	0.000	1580438.248	2495.650124	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	1959860.24	60.307	59.906	726.707	0.000	1580438.248	1140.073839	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	6582163O.95	40.878	40.703	727.801	0.000	1580438.248	4064.770819	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	3685979I.73	42.211	42.047	729.555	0.000	1580438.248	2232.25131	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_021	Genetic Algorithm	53639264.39	35.651	35.547	729.887	0.000	1580438.248	3677.386714	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.921	59.016	731.320	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.178	58.531	731.340	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.849	59.469	731.598	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.618	58.547	731.625	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.961	59.594	735.453	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.649	59.250	731.617	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.946	59.656	735.449	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.372	59.063	735.473	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.176	59.188	731.809	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_1000_021	Ant Colony Optimization	1580438.125	60.523	59.297	736.191	28.600	1580438.248	-7.77E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.054	1.047	732.441	0.000	1580438.248	-8.84E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.041	0.938	730.781	0.000	1580438.248	-1.33E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.085	0.953	732.125	0.000	1580438.248	-8.84E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.031	0.969	732.379	0.000	1580438.248	-8.84E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.075	1.000	732.625	0.000	1580438.248	-8.84E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.220	1.016	732.684	0.000	1580438.248	-1.47E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.031	0.969	732.941	0.000	1580438.248	-8.84E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.028	0.984	733.105	0.000	1580438.248	-1.47E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.034	0.938	733.168	0.000	1580438.248	-8.84E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_1000_021	LKH	1580438.248	1.027	0.969	733.160	0.000	1580438.248	-1.33E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.655	60.484	733.441	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.610	60.266	733.699	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.736	60.516	733.707	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.738	60.641	732.379	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.745	60.484	733.309	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.628	60.484	733.375	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.711	60.313	733.371	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.725	60.453	732.971	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.722	60.563	733.117	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_021	OR-Tools (Routing TSP)	1580438.248	60.714	60.547	734.164	0.000	1580438.248	-3.91E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	416033078.3	44.960	44.844	733.543	0.000	13782703.86	2918.515688	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	509218551.7	42.450	42.281	733.684	0.000	13782703.86	3594.620132	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	505331454.6	34.357	34.156	733.129	0.000	13782703.86	3566.417414	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	3060978O1.4	52.147	51.719	733.141	0.000	13782703.86	2120.883539	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	563411121	32.780	32.500	736.691	0.000	13782703.86	3985.853738	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}

dataset_nam	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_co	relative_error_percent	params
spline_1000_031	Genetic Algorithm	277537206	45.207	45.063	737.672	0.000	13782703.86	1913.662984	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	270077401.9	61.177	60.984	736.195	0.000	13782703.86	1859.538597	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	352740131.1	42.170	41.938	738.305	0.000	13782703.86	2459.295582	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	409743989.8	45.828	45.672	739.707	0.000	13782703.86	2872.885392	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_031	Genetic Algorithm	327561526.3	60.064	59.813	739.664	0.000	13782703.86	2276.612961	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8925, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.405	58.875	741.414	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.144	59.109	743.219	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.961	59.406	740.516	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.332	59.188	740.332	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.399	58.672	740.156	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.954	58.906	744.461	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.494	58.531	743.996	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.402	58.578	741.723	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.588	58.422	745.198	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_1000_031	Ant Colony Optimization	13782703	60.962	59.609	742.988	28.600	13782703.86	-6.27E-06	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8925, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.026	0.906	742.816	0.000	13782703.86	-1.49E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.028	0.969	742.688	0.000	13782703.86	-9.46E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.064	0.953	740.914	0.000	13782703.86	-1.49E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.044	0.984	741.703	0.000	13782703.86	-1.22E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.225	1.078	741.988	0.000	13782703.86	-1.49E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.069	0.938	740.637	0.000	13782703.86	-9.46E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.038	0.984	742.512	0.000	13782703.86	-9.46E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.059	0.953	744.570	0.000	13782703.86	-1.49E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.045	0.938	742.660	0.000	13782703.86	-1.08E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_1000_031	LKH	13782703.86	1.038	0.953	741.699	0.000	13782703.86	-1.49E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8925, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.672	60.625	743.020	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.561	60.484	744.973	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.631	60.578	741.977	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.637	60.422	741.664	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.715	60.469	743.262	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.574	60.656	746.125	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.715	60.625	746.137	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.701	60.469	746.492	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.697	60.719	746.707	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_1000_031	OR-Tools (Routing TSP)	13782703.87	60.580	60.375	747.004	0.000	13782703.86	7.90E-09	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8925, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	20457492.07	61.346	60.813	3052.926	0.000	17387.07416	117559.1984	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	19582467.58	61.316	60.891	3052.707	0.000	17387.07416	112526.5834	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	21101243.86	61.375	60.984	3052.930	0.000	17387.07416	121261.6717	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	20018209.93	60.893	60.797	3054.004	0.000	17387.07416	115032.7115	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	19992113.63	62.248	61.859	3055.309	0.000	17387.07416	114882.6213	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	20664204.42	61.104	60.875	3055.684	0.000	17387.07416	118748.0836	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	19997324.33	61.293	61.359	3056.719	0.000	17387.07416	114912.5901	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	20925337.93	61.119	60.813	3056.133	0.000	17387.07416	120249.9665	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	19869555.86	60.915	60.891	3057.766	0.000	17387.07416	114177.7427	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_001.jsc	Genetic Algorithm	20380817.35	61.792	61.594	3058.527	0.000	17387.07416	117118.2115	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8925, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	65.625	65.141	3419.105	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	66.093	65.813	3419.383	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	65.420	65.281	3420.188	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	66.989	66.844	3420.609	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	66.737	66.250	3420.902	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	65.306	65.141	3421.430	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	65.964	65.547	3421.844	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	66.565	66.125	3422.438	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	65.579	65.125	3422.805	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_10000_001.jsc	Ant Colony Optimization	17387.07416	65.974	65.781	3423.516	2768.628	17387.07416	0	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8925, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	104.960	104.500	3041.523	0.000	17387.07416	-2.72E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	102.833	102.469	3041.391	0.000	17387.07416	-2.51E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}

dataset_nam	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_cost	relative_error_percent	params
spline_10000_001.jsc	LKH	17387.07416	100.393	100.156	3042.324	0.000	17387.07416	-1.26E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	104.156	103.766	3042.453	0.000	17387.07416	-2.51E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	105.213	104.672	3042.945	0.000	17387.07416	-2.30E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	103.805	103.203	3043.852	0.000	17387.07416	-1.88E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	104.828	104.234	3043.754	0.000	17387.07416	-2.09E-14	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	101.003	100.656	3044.578	0.000	17387.07416	-2.51E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	100.473	100.016	3045.199	0.000	17387.07416	-3.98E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_10000_001.jsc	LKH	17387.07416	101.204	100.938	3045.145	0.000	17387.07416	-3.77E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	147.799	147.328	3139.891	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	146.371	145.875	3141.336	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	148.912	148.250	3141.926	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	147.919	147.109	3142.582	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	147.273	146.734	3143.207	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	148.215	147.438	3140.645	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	147.466	146.828	3144.375	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	147.107	146.734	3144.813	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	146.564	146.188	3145.234	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_001.jsc	OR-Tools (Routing TSP)	17387.0783	147.904	147.328	3146.367	0.000	17387.07416	2.38E-05	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	150711944.6	61.455	61.281	3079.453	0.000	134113.2129	112276.6564	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	146151313.1	61.111	60.828	3077.531	0.000	134113.2129	108876.0732	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	149817234.3	61.321	60.891	3078.918	0.000	134113.2129	111609.5258	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	150401092.6	61.766	61.391	3077.461	0.000	134113.2129	112044.8658	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	147865207.1	61.567	61.375	3076.484	0.000	134113.2129	110154.0189	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	153426333.5	61.347	61.047	3080.023	0.000	134113.2129	114300.6098	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	148867646.7	61.117	61.047	3077.828	0.000	134113.2129	110901.4766	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	150570113.1	61.056	60.625	3078.660	0.000	134113.2129	112170.9014	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	155661157	61.240	60.813	3079.223	0.000	134113.2129	115966.9807	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_011	Genetic Algorithm	149215248.1	61.254	60.656	3078.281	0.000	134113.2129	111160.6617	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	69.552	69.500	3442.902	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.082	70.672	3442.840	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.091	70.719	3443.582	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.098	70.734	3444.164	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.052	70.609	3444.391	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	70.844	70.531	3445.012	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.328	71.219	3445.578	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.628	71.266	3445.355	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	71.749	71.453	3446.254	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_10000_011	Ant Colony Optimization	134113.2129	70.407	70.094	3446.559	2768.628	134113.2129	-1.95E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	107.838	107.469	3064.414	0.000	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	104.766	104.500	3064.590	0.000	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	104.551	104.234	3065.816	0.000	134113.2129	-2.17E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	105.621	105.516	3065.453	0.000	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	104.129	103.594	3066.406	0.000	134113.2129	-2.39E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	109.108	108.859	3066.477	0.000	134113.2129	-3.69E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	111.056	110.438	3067.168	0.000	134113.2129	-3.47E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	101.390	100.969	3067.320	0.000	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	105.411	104.922	3067.855	0.000	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_10000_011	LKH	134113.2129	104.360	103.828	3068.684	0.000	134113.2129	-1.95E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	153.553	153.016	3158.109	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	152.932	151.719	3159.441	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	152.074	151.578	3159.961	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	153.347	152.625	3161.211	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	153.548	152.781	3161.309	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	154.927	154.250	3162.137	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	154.270	153.734	3161.949	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	154.350	153.438	3163.078	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	153.546	152.750	3163.309	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}

dataset_name	method	cost	elapsed_s	cpu_time_s	peak_memory_mb	memory_mb	optimal_cost	relative_error_percent	params
spline_10000_011	OR-Tools (Routing TSP)	134113.2152	154.944	153.797	3163.773	0.000	134113.2129	1.68E-06	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1580592.847	63.062	61.203	2390.676	0.000	1580592.847	-4.42E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1580592.847	79.292	76.813	2391.957	0.000	1580592.847	-4.42E-14	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1704135018	61.653	60.891	2429.016	0.000	1580592.847	107716.1919	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1694419810	61.403	60.859	2524.125	0.000	1580592.847	107101.536	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1700959550	61.721	61.141	2521.816	0.000	1580592.847	107508.9616	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1695568031	63.250	62.906	2527.539	0.000	1580592.847	107174.181	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1768954848	61.624	60.344	2507.484	0.000	1580592.847	111817.174	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1726656030	61.667	60.766	2527.398	0.000	1580592.847	109141.0378	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1693888627	61.573	60.922	2524.211	0.000	1580592.847	107067.9294	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_021	Genetic Algorithm	1749935060	62.827	61.688	2522.383	0.000	1580592.847	110613.8415	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	83.074	81.109	4512.133	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	76.195	75.109	3632.230	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	75.424	74.313	2949.813	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	76.361	75.188	2898.531	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	78.750	77.734	2869.480	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	86.002	84.703	2851.445	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	79.871	78.016	2848.426	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 6964, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	79.698	78.094	2848.906	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 7951, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	80.016	78.875	2848.293	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 8938, "time_limit_sec": 60}
spline_10000_021	Ant Colony Optimization	1580592.847	78.250	76.969	2848.762	2768.628	1580592.847	-3.09E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 9925, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	112.642	110.875	2467.320	0.000	1580592.847	-6.33E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 1042, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	113.529	111.938	2471.664	0.000	1580592.847	-5.74E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 2029, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	109.632	107.375	2472.461	0.000	1580592.847	-4.12E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 3016, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	110.597	109.656	2473.066	0.000	1580592.847	-5.45E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4003, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	111.647	110.891	2472.453	0.000	1580592.847	-6.33E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 4990, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	112.676	111.578	2473.395	0.000	1580592.847	-5.45E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 5977, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	115.206	112.641	2474.008	0.000	1580592.847	-6.92E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	110.355	109.266	2474.500	0.000	1580592.847	-6.04E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	108.142	106.984	2475.039	0.000	1580592.847	-6.92E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_10000_021	LKH	1580592.847	113.817	112.703	2475.676	0.000	1580592.847	-6.63E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	165.254	164.047	2599.930	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	165.049	163.422	2601.074	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	165.122	163.547	2591.488	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	164.310	162.813	2602.043	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	165.241	163.828	2602.789	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	165.351	164.328	2603.379	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	165.711	164.109	2603.734	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	164.119	161.453	2600.949	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	164.910	163.047	2604.145	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_021	OR-Tools (Routing TSP)	1580592.841	166.888	165.422	2605.016	0.000	1580592.847	-3.90E-07	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14513755647	61.713	61.516	3107.039	0.000	13785028.63	105196.3656	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14563187399	60.950	60.781	3108.977	0.000	13785028.63	105588.4912	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14869450795	61.136	61.016	3104.461	0.000	13785028.63	107766.666	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14772317286	61.576	61.000	3108.543	0.000	13785028.63	107062.0356	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14605579731	61.555	61.203	3111.082	0.000	13785028.63	105852.4802	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14973679764	61.245	60.781	3106.766	0.000	13785028.63	108522.7687	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14977077586	61.159	60.875	3105.020	0.000	13785028.63	108547.4173	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14620965373	61.519	61.281	3106.668	0.000	13785028.63	105964.0914	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14470804778	61.233	61.078	3109.836	0.000	13785028.63	104874.7894	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_031	Genetic Algorithm	14957313272	61.274	60.906	3106.867	0.000	13785028.63	107678.6174	{"population_size": 200, "generations": 500, "crossover_rate": 0.75, "mutation_rate": 0.1, "selection": "roulette", "tournament_k": 5, "mutation_type": "mixed", "use_local_search": true, "ls_on_init": true, "ls_rate": 0.3, "ls_max_swaps": 200, "ls_elite_only": false, "seed": 9925, "time_limit_sec": 60}
spline_10000_031	Ant Colony Optimization	13785028.63	72.467	71.844	3471.527	2768.628	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 1042, "time_limit_sec": 60}
spline_10000_031	Ant Colony Optimization	13785028.63	71.075	70.953	3470.480	2768.628	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 2029, "time_limit_sec": 60}
spline_10000_031	Ant Colony Optimization	13785028.63	71.325	71.031	3472.488	2768.628	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 3016, "time_limit_sec": 60}
spline_10000_031	Ant Colony Optimization	13785028.63	71.937	71.656	3472.773	2768.628	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4003, "time_limit_sec": 60}
spline_10000_031	Ant Colony Optimization	13785028.63	71.489	71.094	3473.438	2768.628	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 4990, "time_limit_sec": 60}
spline_10000_031	Ant Colony Optimization	13785028.63	71.665	71.719	3474.137	2768.628	13785028.63	2.70E-13	{"ants": 0, "iterations": 200, "alpha": 1.0, "beta": 2.0, "rho": 0.1, "candidate_k": 25, "ants_cap": 100, "use_gpu": true, "seed": 5977, "time_limit_sec": 60}

dataset_nam	method	cost	elapse_s	cpu_tim_e_s	peak_me_morg_m	_memory_mb	optimal_co	relative_erro_percent	params
spline_10000_031	LKH	13785028.63	111.719	111.266	3096.180	0.000	13785028.63	-2.30E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 6964, "time_limit_sec": 60}
spline_10000_031	LKH	13785028.63	107.544	107.047	3096.281	0.000	13785028.63	-1.22E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 7951, "time_limit_sec": 60}
spline_10000_031	LKH	13785028.63	108.155	107.759	3096.703	0.000	13785028.63	-2.16E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 8938, "time_limit_sec": 60}
spline_10000_031	LKH	13785028.63	111.328	111.079	3097.328	0.000	13785028.63	-2.16E-13	{"runs": 10, "candidate_edges": 20, "candidate_set_type": "DELAUNAY", "move_type": "5", "backtracking": 5, "lkh_exe": "", "seed": 9925, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	161.492	161.094	3170.656	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 1042, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	159.664	159.188	3180.020	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 2029, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	159.017	158.547	3180.605	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 3016, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	160.532	159.891	3180.848	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4003, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	159.701	159.328	3182.016	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 4990, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	160.073	159.422	3179.746	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 5977, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	159.903	159.172	3182.488	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 6964, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	160.052	159.484	3183.082	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 7951, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	160.267	159.578	3183.930	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 8938, "time_limit_sec": 60}
spline_10000_031	OR-Tools (Routing TSP)	13785028.64	159.737	159.063	3184.758	0.000	13785028.63	2.22E-08	{"first_solution_strategy": "PATH_CHEAPEST_ARC", "local_search_metaheuristic": "GUIDED_LOCAL_SEARCH", "scale": 10000, "log_search": false, "seed": 9925, "time_limit_sec": 60}