

**UNIVERSITETI I PRISHTINËS**  
**FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE**  
**PROGRAMI I STUDIMEVE: INXHINIERI KOMPJUTERIKE DHE SOFTUERIKE**



**PUNIM DIPLOME**

**Niveli i studimeve: Master**

**Tema:**

**Optimizimi i orarit të skanimit të librave në një rrjet  
bibliotekash përmes kërkimit lokal të përsëritur**

**Mentor:**  
**Prof. Dr. Kadri Sylejmani**

**Kandidati:**  
**Driton Alija**

*Prishtinë, Korrik 2026*

## Abstrakt

Ky punim trajton problemin e skanimit të librave nga Google Hash Code 2020 si problem të optimizimit kombinatorik me kufizime kohore, kapacitete ditore dhe mbivendosje të inventarit ndërmjet bibliotekave. Qëllimi është të përcaktohet renditja e regjistrimit të bibliotekave dhe lista e librave që skanohen nga secila bibliotekë, në mënyrë që të maksimizohet shuma e pikëve të librave unikë të skanuar brenda horizontit kohor.

Metoda e propozuar mbështetet në kërkimin lokal të përsëritur (Iterated Local Search, ILS) dhe është përshtatur enkas për strukturën e këtij problemi. Ajo kombinon disa strategji për ndërtimin e zgjidhjeve fillestare, vlerësimin e shpejtë të tyre, mekanizma për alokimin e librave, përmirësime lokale, perturbim të kontrolluar, pranim të zgjidhjeve alternative dhe rifillim të kërkimit në rast stagnimi. Vlerësimi eksperimental përfshin 324 instanca nga katër familje instancash, të ndara në grup trajnimi dhe grup testimi. Parametrat akordohen me iRace vetëm mbi grupin e trajnimit, ndërsa të gjitha rezultatet finale raportohen dhe interpretohen mbi instancat e testimit.

Për secilën instancë testimi kryhen 10 ekzekutime të pavarura me vlera të ndryshme *seed*. Rezultati më i mirë (*Best*) shërben si vlerë kryesore e krahasimit, ndërsa mesatarja (*AVG*) raportohet si tregues i qëndrueshmërisë. Metoda arrin gjithsej 477,429,716 pikë sipas vlerës *Best* dhe 477,309,618 pikë sipas *AVG*. Kolona *Baseline*, e cila përfaqëson vlerën më të mirë të metodave për krahasim, arrin 466,414,103 pikë në të njëjtin grup testimi. Pra, ILS e tejkalon *Baseline* me 11,015,613 pikë, që përfaqëson një epërsi relative prej 2.362%. Përfitimi ka mbështetje statistikore, por nuk është i njëtrajtshëm. ILS barazon shumë metoda të forta për krahasim, sjell përmirësime të qarta në një pjesë të instancave dhe mbetet nën metodat ekzakte ose të specializuara në një nëngrup të kufizuar. Kontributi kryesor i punimit qëndron në një kornizë ILS të riprodhueshme dhe të analizueshme empirikisht, jo në pretendimin e një optimumi teorik të ri.

**Fjalë kyçe:** skanimi i librave, Google Hash Code, kërkim lokal i përsëritur, optimizim kombinatorik, metaheuristika, iRace, analiza statistikore.

# Abstract

This thesis studies the Google Hash Code 2020 Book Scanning problem as a combinatorial optimization problem with time limits, daily scanning capacities, and overlapping book inventories across libraries. The goal is to determine the library sign-up order and the list of books scanned by each active library so that the total score of uniquely scanned books is maximized within the available time horizon.

The proposed method is a problem-specific Iterated Local Search (ILS) system. It combines several strategies for constructing initial solutions, fast evaluation, mechanisms for book allocation, local improvements, controlled perturbation, acceptance of alternative solutions, and search restart in case of stagnation. The experimental evaluation covers 324 instances from four dataset families, split into a training set and a test set. The parameters are tuned with iRace exclusively on the training set, while all final results are reported and interpreted on the test instances, using 10 seeds per instance for the stochastic ILS configuration.

The main results are reported on the test dataset and use the Best score over 10 runs for each instance, while AVG is kept for transparency. The method achieves a Best total score of 477,429,716 and an AVG total score of 477,309,618. The Baseline column, which represents the best available comparison result, reaches 466,414,103 on the same test dataset. ILS therefore exceeds Baseline by 11,015,613 points, representing a relative advantage of 2.362%. The gain is statistically supported but not uniform. ILS ties many strong baselines, improves clearly on part of the instances, and remains below exact or specialized methods on a limited subset. The main contribution is a reproducible and interpretable ILS framework with detailed empirical analysis, rather than a claim of a new theoretical optimum for the problem.

**Keywords:** book scanning, Google Hash Code, iterated local search, combinatorial optimization, metaheuristics, iRace, statistical analysis.

# **Deklaratë mbi originalitetin dhe përdorimin e inteligjencës artificiale**

Deklaroj se ky punim diplome është punim original dhe nuk është paraqitur më parë për vlerësim në ndonjë institucion tjetër. E gjithë literatura e përdorur dhe burimet e tjera të shfrytëzuara gjatë punimit janë të listuara në referenca dhe plotësisht të cituara.

Jam plotësisht i vetëdijshëm se plagjiatura përbën shkelje akademike dhe mund të rezultojë me notë negative, si dhe me masa disiplinore sipas rregullave të Universitetit të Prishtinës.

## **Përdorimi i inteligjencës artificiale**

Gjatë përgatitjes së këtij punimi janë përdorur mjete të inteligjencës artificiale gjenerative, përfshirë ChatGPT, bazuar në modelin GPT-5.5, dhe Claude Opus 4.8. Këto mjete janë përdorur si ndihmë për strukturimin e materialit, përmirësimin gjuhësor, kontrollin e konsistencës, zhvillimin dhe rishikimin e pjesëve të implementimit të algoritmit, si dhe për përpunimin dhe paraqitjen e tabelave dhe figurave nga rezultatet ekzistuese të repozitorit.

Përdorimi i këtyre mjeteve ka qenë ndihmës dhe nuk zëvendëson punën kërkimore, analizën, interpretimin ose vendimmarrjen akademike të autorit. Përmbajtja shkencore, interpretimi i rezultateve, korrektësia e të dhënave dhe vendimet përfundimtare mbeten përgjegjësi e plotë e autorit.

Autori: Driton Alija

Nënshkrimi: \_\_\_\_\_

FIEK, Prishtinë, \_\_\_\_\_

# Përmbajtja

|                                                                        |             |
|------------------------------------------------------------------------|-------------|
| <b>Abstrakt</b>                                                        | <b>i</b>    |
| <b>Abstract</b>                                                        | <b>ii</b>   |
| <b>Deklaratë</b>                                                       | <b>iii</b>  |
| <b>Lista e shkurtesave</b>                                             | <b>vii</b>  |
| <b>Lista e figurave</b>                                                | <b>viii</b> |
| <b>Lista e tabelave</b>                                                | <b>ix</b>   |
| <b>1 Hyrje</b>                                                         | <b>1</b>    |
| 1.1 Konteksti i problemit . . . . .                                    | 1           |
| 1.2 Motivimi . . . . .                                                 | 2           |
| 1.3 Qëllimi dhe objektivat . . . . .                                   | 2           |
| 1.4 Pyetjet kërkimore . . . . .                                        | 3           |
| 1.5 Kontributet kryesore . . . . .                                     | 4           |
| 1.6 Marrëdhënia me punën paraprake dhe kontributi individual . . . . . | 4           |
| 1.7 Organizimi i punimit . . . . .                                     | 5           |
| <b>2 Problemi i skanimit të librave</b>                                | <b>6</b>    |
| 2.1 Përkufizimi i problemit . . . . .                                  | 6           |
| 2.2 Regjistrimi sekuencial dhe skanimi paralel . . . . .               | 6           |
| 2.3 Format i të dhënave hyrëse . . . . .                               | 7           |
| 2.4 Format i zgjidhjes dalëse dhe vlefshmëria . . . . .                | 8           |
| 2.5 Formulimi matematikor . . . . .                                    | 8           |
| 2.6 Kompleksiteti i problemit dhe kufiri i sipërm . . . . .            | 11          |
| 2.7 Familjet e instancave . . . . .                                    | 11          |
| <b>3 Rishikimi i literaturës</b>                                       | <b>13</b>   |
| 3.1 Klasifikimi teorik i problemit . . . . .                           | 13          |
| 3.2 Heuristikat konstruktive . . . . .                                 | 13          |

|          |                                                                              |           |
|----------|------------------------------------------------------------------------------|-----------|
| 3.3      | Kërkimi lokal dhe alokimi i librave . . . . .                                | 14        |
| 3.4      | Kërkimi lokal i përsëritur . . . . .                                         | 14        |
| 3.5      | GRASP dhe diversifikimi i ndërtimit fillestar . . . . .                      | 15        |
| 3.6      | Akordimi automatik i parametrave . . . . .                                   | 15        |
| 3.7      | Modelet e mëdha gjuhësore dhe kërkimi automatik i funksioneve heuristike . . | 16        |
| 3.8      | Metodat për krahasim të përdorura në këtë punim . . . . .                    | 16        |
| <b>4</b> | <b>Metodologjia e propozuar</b>                                              | <b>17</b> |
| 4.1      | Vështrim i përgjithshëm . . . . .                                            | 17        |
| 4.2      | Përfaqësimi i zgjidhjes . . . . .                                            | 17        |
| 4.3      | Alokimi i librave . . . . .                                                  | 18        |
| 4.4      | Vlerësimi i kandidatëve . . . . .                                            | 19        |
| 4.5      | Ndërtimi i zgjidhjes fillestare . . . . .                                    | 20        |
| 4.6      | Operatorët e kërkimit lokal . . . . .                                        | 23        |
| 4.7      | Perturbimi, pranimi dhe rifillimi . . . . .                                  | 26        |
| 4.8      | Akordimi i konfigurimit final . . . . .                                      | 26        |
| 4.9      | Pseudokodi i metodës . . . . .                                               | 28        |
| <b>5</b> | <b>Dizajni eksperimental dhe vlerësimi</b>                                   | <b>29</b> |
| 5.1      | Mjedisi eksperimental . . . . .                                              | 29        |
| 5.2      | Ndarja në grupe trajnimi dhe testimi . . . . .                               | 29        |
| 5.3      | Vlerat seed dhe riprodhueshmëria . . . . .                                   | 32        |
| 5.4      | Validimi i zgjidhjeve . . . . .                                              | 32        |
| 5.5      | Metodat për krahasim . . . . .                                               | 32        |
| 5.6      | Parametrat për krahasim . . . . .                                            | 33        |
| 5.7      | Analiza statistikore . . . . .                                               | 34        |
| 5.8      | Analiza e komponentëve të algoritmit . . . . .                               | 34        |
| 5.9      | Analiza e konvergjencës së algoritmit . . . . .                              | 34        |
| <b>6</b> | <b>Rezultatet eksperimentale dhe diskutimi</b>                               | <b>35</b> |
| 6.1      | Përmirësimi nga zgjidhja fillestare . . . . .                                | 35        |
| 6.2      | Analiza e komponentëve të algoritmit . . . . .                               | 36        |
| 6.3      | Analiza e grupeve të operatorëve . . . . .                                   | 38        |
| 6.4      | Ndikimi i zgjidhjes fillestare . . . . .                                     | 40        |
| 6.5      | Analiza e konvergjencës së algoritmit . . . . .                              | 43        |

|          |                                                          |           |
|----------|----------------------------------------------------------|-----------|
| 6.6      | Qëndrueshmëria ndaj vlerave seed . . . . .               | 45        |
| 6.7      | Krahasimi me baseline dhe metodat për krahasim . . . . . | 46        |
| 6.8      | Interpretimi i rezultateve . . . . .                     | 49        |
| <b>7</b> | <b>Përfundimet</b>                                       | <b>50</b> |
| 7.1      | Përmbledhje e punimit . . . . .                          | 50        |
| 7.2      | Përgjigje ndaj pyetjeve kërkimore . . . . .              | 50        |
| 7.3      | Kufizimet . . . . .                                      | 51        |
| 7.4      | Puna e ardhshme . . . . .                                | 52        |
| 7.5      | Diskutim përmbyllës . . . . .                            | 52        |
| <b>8</b> | <b>Conclusions</b>                                       | <b>53</b> |
| 8.1      | Thesis Summary . . . . .                                 | 53        |
| 8.2      | Answers to the Research Questions . . . . .              | 53        |
| 8.3      | Limitations . . . . .                                    | 54        |
| 8.4      | Future Work . . . . .                                    | 55        |
| 8.5      | Concluding Remarks . . . . .                             | 55        |
|          | <b>Referencat</b>                                        | <b>56</b> |

## Lista e shkurtesave

| Shkurtesa | Kuptimi                                                                                                     |
|-----------|-------------------------------------------------------------------------------------------------------------|
| AI/IA     | Artificial Intelligence / Inteligjenca artificiale                                                          |
| AVG       | Mesatarja aritmetike mbi ekzekutimet                                                                        |
| Baseline  | Vlerë referuese për krahasim. Në tabelat kryesore përfaqëson rezultatin më të mirë të metodave për krahasim |
| Best      | Rezultati më i mirë i vërejtur mbi 10 ekzekutime                                                            |
| BSP       | Book Scanning Problem, problemi i skanimit të librave                                                       |
| CSV       | Comma-Separated Values, format tekstual për ruajtjen e të dhënave tabelore                                  |
| CV        | Coefficient of Variation, koeficienti i variacionit                                                         |
| GRASP     | Greedy Randomized Adaptive Search Procedure                                                                 |
| ILS       | Iterated Local Search, kërkim lokal i përsëritur                                                            |
| iRace     | Paketë për akordim automatik të parametrave përmes garimit iterativ                                         |
| MILP      | Mixed-Integer Linear Programming, programim linear me numra të përzier të plotë                             |
| NP        | Nondeterministic Polynomial time                                                                            |
| RAM       | Random Access Memory, memoria operative e sistemit                                                          |
| SA        | Simulated Annealing, ftohje e simuluar                                                                      |
| SAT       | Boolean Satisfiability Problem, problem i plotësueshmërisë booleane                                         |
| SCIP      | Solving Constraint Integer Programs, zgjidhës për modele optimizimi me kufizime dhe numra të plotë          |
| Seed      | Vlerë hyrëse që kontrollon gjeneratorin e rastësisë në ekzekutime stokastike                                |
| UB        | Upper Bound, kufi i sipërm diagnostik                                                                       |
| vCPU      | Virtual Central Processing Unit, njësi virtuale procesori                                                   |



## Lista e figurave

|     |                                                                                                                                     |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Regjistrimi i bibliotekave kryhet në mënyrë sekuenciale, ndërsa skanimi kryhet paralelisht pas aktivizimit të bibliotekave. . . . . | 7  |
| 2.2 | Shkalla strukturore e familjeve të instancave sipas numrit të librave dhe bibliotekave. . . . .                                     | 12 |
| 4.1 | Rrjedha e përgjithshme e metodës së propozuar ILS. . . . .                                                                          | 17 |
| 4.2 | Përfaqësimi i zgjidhjes si renditje bibliotekash, bibliotekat jashtë planit efektiv të skanimit dhe librat e paskanuar. . . . .     | 18 |
| 4.3 | Ilustrim skematik i grupeve kryesore të operatorëve lokalë. . . . .                                                                 | 24 |
| 6.1 | Përmirësimi i ILS nga zgjidhja fillestare te zgjidhja finale në instancat e testimit. . . . .                                       | 35 |
| 6.2 | Numri i pranimeve sipas operatorit gjatë analizës së konvergjencës. . . . .                                                         | 40 |
| 6.3 | Pjesa e përmirësimit final e arritur gjatë fazës ILS, sipas familjes së instancave. . . . .                                         | 44 |
| 6.4 | Pjesa e rezultatit final e arritur gjatë fazës ILS, sipas familjes së instancave. . . . .                                           | 44 |
| 6.5 | Koha mesatare për arritjen e 95% të përmirësimit final sipas instancës. . . . .                                                     | 45 |
| 6.6 | Koeficienti i variacionit i rezultatit final sipas instancës. . . . .                                                               | 45 |
| 6.7 | Krahasimi i rezultateve totale sipas familjes së instancave. . . . .                                                                | 46 |

## Lista e tabelave

|      |                                                                                                                                               |    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.1  | Përmbledhje e strategjive konstruktive për zgjidhjen fillestare. . . . .                                                                      | 22 |
| 4.2  | Operatorët e kërkimit lokal dhe roli i tyre. . . . .                                                                                          | 25 |
| 4.3  | Parametrat kryesorë të konfigurimit të akorduar dhe hapësira e kërkimit në iRace. . . . .                                                     | 27 |
| 5.1  | Ndarja trajnim/testim dhe numri i ekzekutimeve finale ILS në instancat e testimit. . . . .                                                    | 30 |
| 5.2  | Përmbledhje e instancave të grupit të trajnimit, të përdorura për akordimin e parametrave. . . . .                                            | 30 |
| 5.3  | Instancat e grupit të testimit, të përdorura për raportimin final. . . . .                                                                    | 31 |
| 6.1  | Instancat me përmirësimin më të madh relativ të ILS nga zgjidhja fillestare te rezultati final. . . . .                                       | 36 |
| 6.2  | Eksperimentimi me komponentët e algoritmit në instancat e testimit. . . . .                                                                   | 36 |
| 6.3  | Testet Wilcoxon për krahasimet në çifte të varianteve të komponentëve kundrejt konfigurimit të plotë. . . . .                                 | 37 |
| 6.4  | Gap-i relativ i varianteve të komponentëve ndaj ILS për secilën instancë. . . . .                                                             | 37 |
| 6.5  | Eksperimenti me grupin e operatorëve në instancat e testimit. . . . .                                                                         | 38 |
| 6.6  | Testet Wilcoxon për krahasimet në çifte të varianteve të grupeve të operatorëve kundrejt konfigurimit të plotë. . . . .                       | 39 |
| 6.7  | Gap-i relativ i varianteve të grupeve të operatorëve ndaj ILS për secilën instancë. . . . .                                                   | 39 |
| 6.8  | Krahasimi i konfigurimit të plotë me nisjen e thjeshtë në instancat e testimit. . . . .                                                       | 41 |
| 6.9  | Testi Wilcoxon për krahasimin në çift të nisjes së thjeshtë kundrejt konfigurimit të plotë. . . . .                                           | 41 |
| 6.10 | Gap-i relativ i nisjes së thjeshtë ndaj ILS për secilën instancë. . . . .                                                                     | 41 |
| 6.11 | Përmbledhje e analizës së konvergjencës së algoritmit për 10 instancat e zgjedhura. . . . .                                                   | 43 |
| 6.12 | Konvergenca e ILS në 10 instancat e zgjedhura. . . . .                                                                                        | 43 |
| 6.13 | Testet Wilcoxon për krahasimet në çifte ndërmjet ILS dhe metodave për krahasim në instancat e testimit. . . . .                               | 46 |
| 6.14 | Rezultatet konkrete në instancat e testimit. ILS raporton rezultatin më të mirë nga 10 ekzekutime. UB është kufi diagnostik i sipërm. . . . . | 47 |
| 6.15 | Tabela konkrete e rezultateve për secilën instancë test. Vlerat me bold janë rezultati më i mirë në rresht. . . . .                           | 48 |

# 1. Hyrje

## 1.1. Konteksti i problemit

Digjitalizimi në shkallë të gjerë i librave nuk është vetëm sfidë teknologjike, por edhe problem planifikimi, ku vendimet algoritmike ndikojnë drejtpërdrejt në shfrytëzimin e kohës, të kapaciteteve dhe të burimeve operationale. Në një rrjet bibliotekash, secila bibliotekë ka inventar të vetin, kërkon një kohë të caktuar për t'u regjistruar në procesin e skanimit dhe mund të skanojë vetëm një numër të kufizuar librash në ditë. Ndërkohë, i njëjti libër mund të gjendet në disa biblioteka, por vlera e tij numërohet vetëm një herë. Pikërisht këtu lind ndërvlerësia që e bën problemin interesant. Sapo zgjidhet një bibliotekë, ndryshon vlera marginale e të gjitha bibliotekave të tjera.

Problemi i skanimit të librave u formalizua si detyrë konkurruese në Google Hash Code 2020 [1]. Edhe pse formulimi kuptohet lehtë, hapësira e zgjidhjeve rritet me shpejtësi. Duhet përcaktuar renditja e bibliotekave, duhet vendosur cilat prej tyre regjistrohen, duhet shmangur skanimi i të njëjtit libër më shumë se një herë dhe i gjithë plani duhet të përfundojë brenda horizontit kohor. Për instanca të mëdha, kërkimi ekzakt del jashtë mundësive praktike, prandaj përdorimi i metodave heuristike dhe metaheuristike është zgjedhje e natyrshme [2–4].

Ky punim ndjek një qasje të bazuar në kërkimin lokal të përsëritur, e cila kombinon ndërtimin e shpejtë të zgjidhjeve fillestare, kërkimin lokal intensiv dhe strategji diversifikimi që e drejtojnë kërkimin drejt rajoneve të reja të hapësirës së zgjidhjeve.

Për të vlerësuar cilësinë e metodës ILS, rezultatet e saj krahasohen me metoda për krahasim (*baseline*) të bazuara në dy zgjidhje të pavarura me kod të hapur. Burimi i parë është zgjidhja origjinale e Nawid Sayed për Google Hash Code 2020. Ekipi u rendit i treti nga 4 700 në fazën shtesë të garës (*extended round*) [5]. Burimi i dytë është zgjidhja origjinale e ekipit Errecto, i renditur në vendin e pestë në raundin kualifikues [6]. Për eksperimentet tona, të dyja zgjidhjet janë zgjeruar dhe përshtatur për strukturën hyrëse/dalëse dhe rrjedhën e ekzekutimit të repozitorit tonë, pa ndryshuar logjikën bazë të kodeve burimore [7, 8]. Në këtë punim, këto algoritme nuk trajtohen vetëm si rezultate të raportuara, por ekzekutohen mbi të gjithë grupin e instancave dhe rezultatet e prodhuara krahasohen me ILS. Në grupin e metodave për krahasim përfshihen Errecto, Greedy, MILP1, MILP2, SAT dhe SA.

Për secilën instancë ndërtohet vlera më e mirë e metodave për krahasim, duke marrë rezultatin më të lartë të arritur nga cilado prej këtyre metodave. Në këtë mënyrë, ILS nuk krahasohet vetëm me një algoritëm të vetëm, por me grupin më të fortë të metodave ekzistuese për atë instancë.

## 1.2. Motivimi

Motivimi i këtij punimi mbështetet në dy aspekte. Nga njëra anë, problemi është shembull i qartë i optimizimit kombinatorik me kufizime kohore, ku duhet vendosur njëkohësisht si renditen bibliotekat dhe si shpërndahen librat ndërmjet tyre. Nga ana tjetër, ekziston një boshllëk kërkimor ndërmjet zgjidhjeve shumë të specializuara për pesë instancat zyrtare të Google Hash Code dhe një metode më të përgjithshme, e cila vlerësohet në disa familje instancash.

Në shumë zgjidhje konkurruese, performanca e lartë arrihet duke përdorur kritere të veçanta për renditjen dhe përzgjedhjen e bibliotekave, të përshtatura enkas për instanca specifike. Në kontekst gare kjo qasje funksionon, por e vështirëson analizën e përgjithshme shkencore. Prandaj ky punim nuk mbështetet në një rregull të vetëm lakmitar, por propozon një arkitekturë metaheuristike të riprodhueshme, e cila bashkon disa strategji për ndërtimin e zgjidhjeve fillestare, rindërtim të shpejtë të alokimit të librave, operatorë të kërkimit lokal, perturbim, rifillim dhe akordim automatik të parametrave, duke u ndërtuar mbi themelin e vendosur në punën tonë paraprake (shih Seksionin 1.6).

Kontributi metodologjik nuk qëndron në propozimin e koncepteve krejtësisht të reja, por në integrimin e kujdesshëm të disa teknikave ekzistuese në një arkitekturë të përshtatur për këtë problem. Punimi nuk pretendon se ILS, GRASP, operatorët lokalë ose iRace janë risi në vetvete. Risia qëndron te mënyra se si këta komponentë lidhen me strukturën specifike të problemit: përfaqësimi përmes renditjes së bibliotekave dhe rindërtimit të alokimit të librave, vlerësimi i shpejtë i kandidatëve, strategjitë e ndryshme për ndërtimin e zgjidhjeve fillestare të orientuara nga kapaciteti, rrallësia dhe mbulimi, si dhe analiza e hollësishme e rasteve kur ky integrim sjell përfitim real ndaj metodave të forta për krahasim. Në këtë kuptim, punimi paraqitet si kontribut inxhinierik dhe eksperimental i riprodhueshëm, pa pretenduar një qasje të re teorike.

Një drejtim i rëndësishëm i kërkimeve të fundit është përdorimi i modeleve të mëdha gjuhësore për propozimin dhe përmirësimin automatik të programeve ose funksioneve të vlerësimit. Në këtë kontekst, Romera-Paredes et al. prezantojnë FunSearch, një metodë në të cilën modelet gjuhësore gjenerojnë programe të vogla, ndërsa një mekanizëm automatik vlerësimi e udhëheq kërkimin drejt zgjidhjeve me performancë më të lartë [9]. Duke ndjekur të njëjtën ide, Veličković et al. e lidhin këtë drejtim me probleme kombinatorike në garat e programimit, ku funksionet e vlerësimit gjenerohen, testohen dhe përzgjidhen automatikisht për të ndërtuar heuristika më efektive [10]. Ky punim nuk synon të gjenerojë automatikisht formula të tilla. Përkundrazi, ai shqyrton se sa larg mund të arrijë një metaheuristikë klasike, e ndërtuar me kujdes inxhinierik dhe e vlerësuar në mënyrë statistikore.

## 1.3. Qëllimi dhe objektivat

Qëllimi i përgjithshëm i punimit është zhvillimi, analizimi dhe vlerësimi eksperimental i një metode të bazuar në kërkimin lokal të përsëritur për problemin e skanimit të librave. Për ta bërë këtë qëllim konkret dhe të matshëm, punimi ndjek objektivat e mëposhtme:

## 1. Hyrje

1. Formalizimi i problemit të skanimit të librave, duke identifikuar vendimet kryesore, kufizimet dhe funksionin objektiv.
2. Ndërtimi i zgjidhjeve fillestare përmes disa strategjive konstruktive, të cilat balancojnë shpejtësinë, mbulimin e librave dhe vlerën marginale të çdo zgjedhjeje.
3. Implementimi i një mekanizmi të kërkimit lokal që përmirëson renditjen e bibliotekave përmes zhvendosjes, shkëmbimit, futjes dhe diversifikimit të kontrolluar.
4. Integrimi i perturbimit adaptiv, i rifillimit dhe i pranimit të kontrolluar të zgjidhjeve me vlerë më të ulët, me qëllim shmangien e ngecjes në optimume lokale.
5. Akordimi i parametrave me iRace dhe vlerësimi i konfigurimit final përmes një dizajni eksperimental të riprodhueshëm.
6. Krahasimi statistikor me metodat për krahasim, analiza e kontributit të komponentëve të algoritmit dhe analiza e konvergencës së tij.

## 1.4. Pyetjet kërkimore

Punimi u përgjigjet këtyre pyetjeve kërkimore:

1. Sa i përshtatshëm është kërkimi lokal i përsëritur për problemin e optimizimit të renditjes së regjistrimit të bibliotekave dhe skanimit të librave?
2. A është konfigurimi final i akorduar me iRace efektiv dhe i qëndrueshëm?
3. A arrin metoda e propozuar ILS rezultate më të mira se metodat për krahasim, duke përfshirë qasjet lakmitare, SA, SAT, MILP dhe Errecto, në instanca me karakteristika të ndryshme?
4. Cilët operatorë lokalë dhe cilët komponentë të tjerë të ILS kontribuojnë më shumë në performancën përfundimtare të metodës?
5. Si ndryshon performanca e algoritmit në familje instancash me madhësi dhe strukturë të ndryshme, dhe sa i qëndrueshëm është ai ndaj vlerave *seed*?
6. Sa ndikon cilësia e zgjidhjes fillestare në rezultatin përfundimtar të ILS?
7. Si zhvillohet cilësia e zgjidhjes së konfigurimit final ILS gjatë buxhetit kohor dhe sa shpejt arrihet pjesa kryesore e përmirësimit?

## 1.5. Kontributet kryesore

Kontributet kryesore të punimit janë:

- Zgjerimi i implementimit fillestar ILS nga puna paraprake [11] në një arkitekturë më të plotë për problemin Book Scanning, e cila mbështetet në përfaqësimin përmes renditjes së bibliotekave, rindërtimin e alokimit të librave, strategji të ndryshme për ndërtimin e zgjidhjeve fillestare, vlerësim të shpejtë dhe operatorë lokalë të specializuar.
- Ndërtimi i një dizajni eksperimental të riprodhueshëm, i cili i ndan 324 instancat në grup trajnimi dhe grup testimi dhe përdor 10 ekzekutime të pavarura për secilën instancë testuese të konfigurimit final.
- Realizimi i një analize krahasuese në instancat e testimit, ku metoda e propozuar arrin rezultatin total më të lartë prej 477,429,716 pikësh dhe një epërsi relative prej 2.362% ndaj Baseline.
- Kryerja e një analize statistikore me teste Wilcoxon për krahasime në çifte, korrigjim Holm për krahasime të shumëfishta dhe tabela përmbledhëse sipas familjeve të instancave.
- Analiza e konvergjencës së algoritmit përmes 100 gjurmëve të ekzekutimit, ku maten koha për arritjen e 95% të përmirësimit, qëndrueshmëria ndaj vlerave *seed* dhe aktiviteti i operatorëve.
- Paraqitja e tabelave konkrete të rezultateve brenda dokumentit, me nga një rresht për secilën instancë testuese dhe me skedarët burimorë CSV të ruajtur në repozitorin kryesor të punimit për riprodhueshmëri [12].

## 1.6. Marrëdhënia me punën paraprake dhe kontributi individual

Një version paraprak i problemit dhe i qasjeve metaheuristike është botuar në vitin 2025 në bashkëautorësi, me kandidatin si një prej autorëve [11]. Ai paraqet formulimin MILP, ndërtimin fillestar me GRASP, variante fillestare të ILS-it dhe algoritmit gjenetik, si dhe një studim paraprak me 20 instance dhe pesë ekzekutime për secilën metodë. Kontributi i kandidatit në atë punë lidhej me zhvillimin e variantit fillestar ILS dhe të ndërtimit GRASP.

Teza vazhdon vetëm drejtimin ILS dhe e zgjeron ndjeshëm punën paraprake. Kontributi individual i kandidatit në kuadër të kësaj teze përfshin:

- zgjerimin dhe riorganizimin e implementimit ILS, duke përfshirë ndërtimin e shumë zgjidhjeve fillestare, vlerësimin e shpejtë, operatorët e parametrizuar, perturbimin adaptiv dhe rifillimin pas stagnimit.

## 1. Hyrje

- integrimin e iRace dhe hartimin e dizajnit eksperimental me 324 instanca të ndara në trajnim dhe testim, me 10 vlera *seed* për çdo instancë testuese.
- realizimin e analizave të komponentëve, operatorëve dhe konvergjencës, krahasimin statistikor dhe përgatitjen e artefakteve të riprodhueshmërisë.

Baza e formulimit matematikor dhe konceptet fillestare ILS/GRASP trajtohen si punë paraprake. Teza zgjeron vetëm ILS-in dhe raporton eksperimente të reja. GA-ja dhe rezultatet e vitit 2025 nuk ripërdoren.

## 1.7. Organizimi i punimit

Kapitulli 2 formalizon problemin e skanimit të librave dhe përshkruan të dhënat hyrëse, zgjidhjen dalëse dhe modelin matematik. Kapitulli 3 rishikon literaturën dhe bazat teorike, duke përfshirë ILS, metodat për krahasim dhe qasjet moderne të bazuara në gjenerimin dhe përmirësimin automatik të funksioneve të vlerësimit. Kapitulli 4 paraqet metodologjinë e propozuar dhe mekanizmat e kërkimit. Kapitulli 5 përshkruan dizajnin eksperimental, ndarjen në grupe trajnimi dhe testimi, parametrat për krahasim dhe metodat statistikore. Kapitulli 6 paraqet rezultatet eksperimentale, analizën e kontributit të komponentëve, eksperimentin me grupet e operatorëve, analizën e konvergjencës dhe krahasimin final me metodat për krahasim. Kapitulli 7 përmbledh punimin, u përgjigjet pyetjeve kërkimore, diskuton kufizimet dhe identifikon drejtimet e punës së ardhshme, ndërsa Kapitulli 8 paraqet përfundimet kryesore në gjuhën angleze.

## 2. Problemi i skanimit të librave

### 2.1. Përkufizimi i problemit

Problemi i skanimit të librave u prezantua në raundin kualifikues të Google Hash Code 2020 dhe modelon planifikimin e digjitalizimit të një koleksioni të madh librash nën kufizime kohe dhe kapaciteti [1]. Nga këndvështrimi algoritmik, sfida nuk lidhet me skanimin fizik të një libri të vetëm. Thelbi i problemit qëndron te ndërveprimi ndërmjet vonesave të regjistrimit, kapaciteteve ditore dhe koleksioneve të librave që mbivendosen ndërmjet bibliotekave.

Në problem jepen një bashkësi librash dhe një bashkësi bibliotekash. Çdo libër ka një vlerë numerike në pikë, e cila fitohet vetëm nëse libri skanohet brenda horizontit kohor të lejuar. I njëjti libër mund të gjendet në më shumë se një bibliotekë, por vlera e tij llogaritet vetëm një herë, pavarësisht se nga cila bibliotekë skanohet. Çdo bibliotekë kërkon një numër të caktuar ditësh për t'u regjistruar në procesin e skanimit dhe, pas përfundimit të regjistrimit, mund të skanojë vetëm një numër të kufizuar librash në ditë.

Prandaj, zgjidhja e problemit kërkon dy vendime kryesore: përcaktimin e renditjes me të cilën regjistrohen bibliotekat dhe zgjedhjen e librave që do të skanohen nga secila bibliotekë e regjistruar. Qëllimi është maksimizimi i shumës së pikëve të librave unikë që skanohen me sukses brenda afatit kohor.

### 2.2. Regjistrimi sekuencial dhe skanimi paralel

Horizonti kohor përbëhet nga ditët  $0, 1, \dots, D - 1$ . Regjistrimi i bibliotekave është sekuencial. Në çdo moment vetëm një bibliotekë mund të jetë në proces regjistrimi. Regjistrimi i bibliotekës së parë mund të fillojë në ditën 0, ndërsa regjistrimi i një biblioteke tjetër mund të fillojë vetëm pasi të ketë përfunduar procesi i mëparshëm. Kjo e bën renditjen e bibliotekave vendim kritik, sepse çdo ditë e shpenzuar për regjistrim zvogëlon kohën e mbetur për skanim.

Skanimi, në anën tjetër, është paralel. Sapo një bibliotekë përfundon regjistrimin, ajo mund të fillojë të dërgojë libra në qendrën e skanimit nga dita pasuese. Të gjitha bibliotekat që janë regjistruar mund të skanojnë njëkohësisht, secila sipas kapacitetit të vet ditor. Specifikimi zyrtar supozon se dërgimi, skanimi dhe kthimi i librave realizohen brenda një dite, ndërsa qendra e skanimit nuk ka kufi global të kapacitetit. Kufizimi i vetëm ditor është kapaciteti individual i secilës bibliotekë [1].

Figura 2.1 paraqet këtë strukturë operacionale. Blloqet e regjistrimit nuk mbivendosen, ndërsa blloqet e skanimit mund të mbivendosen pasi bibliotekat aktivizohen. Ky dallim është burimi kryesor i kompromisit. Një bibliotekë me koleksion të vlefshëm por regjistrim të gjatë mund të jetë më pak tërheqëse sesa një bibliotekë më e vogël që aktivizohet shpejt dhe skanon libra të rrallë.



## 2. Problemi i skanimit të librave

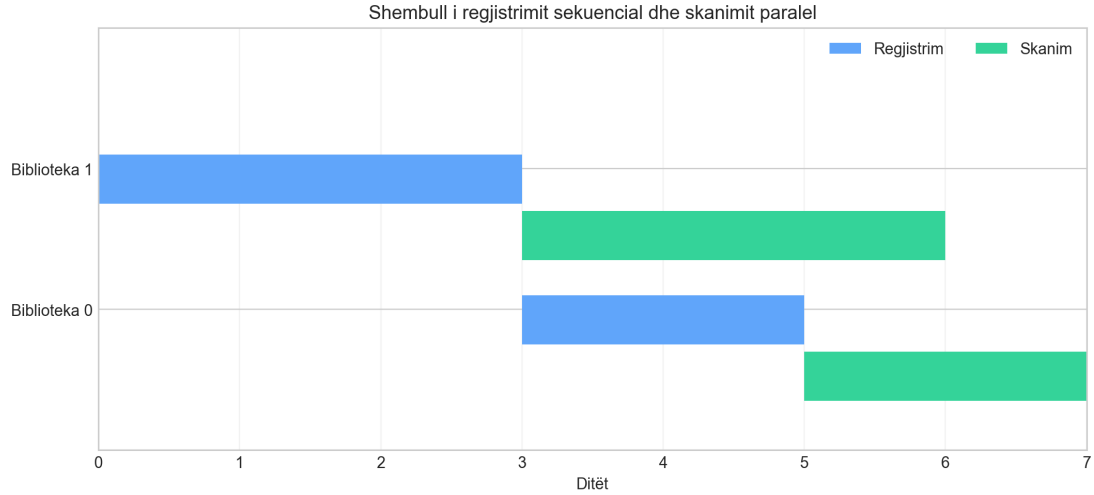


Figura 2.1: Regjistrimi i bibliotekave kryhet në mënyrë sekuenciale, ndërsa skanimi kryhet paralelisht pas aktivizimit të bibliotekave.

### 2.3. Formati i të dhënave hyrëse

Çdo instancë jepet si skedar tekstual. Skedari përmban fillimisht parametrat globalë të problemit, pastaj vlerat e librave dhe, në fund, nga dy rreshta për secilën bibliotekë. Skematikisht, formati hyrës është:

| Rreshti                                | Kuptimi                                                             |
|----------------------------------------|---------------------------------------------------------------------|
| $B \ L \ D$                            | numri i librave, numri i bibliotekave dhe horizonti kohor           |
| $s_0, s_1, \dots, s_{B-1}$             | vlerat e librave                                                    |
| $N_0 \ T_0 \ M_0$                      | numri i librave, koha e regjistrimit dhe kapaciteti i bibliotekës 0 |
| $b_{0,0}, b_{0,1}, \dots, b_{0,N_0-1}$ | identifikuesit e librave në bibliotekën 0                           |
| $\vdots$                               | $\vdots$                                                            |
| $N_{L-1} \ T_{L-1} \ M_{L-1}$          | parametrat e bibliotekës $L - 1$                                    |
| $b_{L-1,0}, \dots, b_{L-1,N_{L-1}-1}$  | identifikuesit e librave në bibliotekën $L - 1$                     |

Këtu  $B$  është numri i librave,  $L$  numri i bibliotekave dhe  $D$  numri i ditëve të disponueshme. Për bibliotekën  $j$ , madhësia  $N_j$  tregon numrin e librave që ndodhen në bibliotekë,  $T_j$  tregon kohën e regjistrimit në ditë, ndërsa  $M_j$  tregon numrin maksimal të librave që biblioteka mund të skanojë në ditë pas regjistrimit. Secili libër paraqitet më së shumti një herë brenda së njëjtës bibliotekë, ndërsa numri total i paraqitjeve të librave në të gjitha bibliotekat është i kufizuar nga specifikimi zyrtar për të ruajtur madhësinë praktike të instancave [1].

Siç shihet, skedari hyrës përmban dy lloje informacioni: informacion global mbi librat dhe horizontin kohor, si dhe informacion lokal për secilën bibliotekë. Algoritmi duhet t'i lidhë vazhdimisht këto dy nivele, sepse vlera e një biblioteke nuk varet vetëm nga librat e saj, por edhe nga librat që tashmë janë skanuar prej bibliotekave të tjera.

## 2.4. Formati i zgjidhjes dalëse dhe vlefshmëria

Zgjidhja dalëse fillon me numrin  $A$  të bibliotekave që propozohen për regjistrim. Pastaj, për secilën bibliotekë të zgjedhur, jepen dy rreshta: rreshti i parë përmban identifikuesin e bibliotekës dhe numrin e librave të planifikuar për skanim, ndërsa rreshti i dytë përmban listën e librave në rendin e skanimit. Bibliotekat në zgjidhjen dalëse lexohen sipas rendit të fillimit të regjistrimit. Në formë skematike, zgjidhja dalëse ka strukturën:

| Rreshti                                | Kuptimi                                                        |
|----------------------------------------|----------------------------------------------------------------|
| $A$                                    | numri i bibliotekave të përfshira në zgjidhje                  |
| $Y_0 \ K_0$                            | biblioteka e parë dhe numri i librave që planifikohen prej saj |
| $k_{0,0}, k_{0,1}, \dots, k_{0,K_0-1}$ | librat e skanuar nga biblioteka $Y_0$                          |
| $\vdots$                               | $\vdots$                                                       |
| $Y_{A-1} \ K_{A-1}$                    | biblioteka e fundit e përfshirë në zgjidhje                    |
| $k_{A-1,0}, \dots, k_{A-1,K_{A-1}-1}$  | librat e skanuar nga biblioteka $Y_{A-1}$                      |

Një zgjidhje mund të mos regjistrojë të gjitha bibliotekat. Bibliotekat që nuk përfshihen thjesht injorohen. Nëse një bibliotekë e përfshirë në zgjidhjen dalëse përfundon regjistrimin pas skadimit të horizontit kohor, kontributi i saj praktik është zero. Po ashtu, librat që do të dërgoheshin pas ditës së fundit nuk kontribuojnë në rezultat. Nëse i njëjti libër dërgohet nga më shumë se një bibliotekë, zgjidhja dalëse mund të mbetet e pranueshme, por pikët e librit numërohen vetëm herën e parë që libri skanohet brenda afatit [1].

Këto rregulla janë të rëndësishme për interpretimin e zgjidhjeve. Nuk mjafton që algoritmi të gjejë renditje me vlerë të lartë teorike. Zgjidhja dalëse duhet të jetë edhe e vlefshme, duke respektuar kapacitetet ditore, rendin e regjistrimit, përkatësinë e librave në bibliotekë dhe mosnumërimin e dyfishtë të tyre.

## 2.5. Formulimi matematikor

Bazuar në specifikimin zyrtar të problemit Book Scanning [1], për ta mbajtur notacionin të lidhur me formatin hyrës, le të jenë  $\mathcal{B} = \{0, \dots, B-1\}$  bashkësia e librave dhe  $\mathcal{L} = \{0, \dots, L-1\}$  bashkësia e bibliotekave. Çdo libër  $b \in \mathcal{B}$  ka vlerë  $s_b \geq 0$ . Çdo bibliotekë  $\ell \in \mathcal{L}$  karakterizohet nga:

- $\mathcal{B}_\ell \subseteq \mathcal{B}$ , bashkësia e librave që ndodhen në bibliotekë, me  $|\mathcal{B}_\ell| = N_\ell$ .
- $T_\ell$ , numri i ditëve të nevojshme për regjistrim.
- $M_\ell$ , kapaciteti ditor i skanimit pas regjistrimit.

Një zgjidhje përcakton një permutim të pjesshëm të bibliotekave:

## 2. Problemi i skanimimit të librave

$$\pi = (\pi_1, \pi_2, \dots, \pi_k), \quad k \leq L. \quad (2.1)$$

Në lidhje me formatin dalës,  $k = A$  dhe biblioteka  $Y_{i-1}$  në zgjidhjen dalëse korrespondon me bibliotekën  $\pi_i$  në formulimin matematikor, për  $i = 1, \dots, k$ . Listat e librave të dhëna pas secilës bibliotekë përfaqësohen nga bashkësitë  $X_{\pi_i}$ .

Dita në të cilën biblioteka  $\pi_i$  bëhet aktive është:

$$a_{\pi_i} = \sum_{j=1}^i T_{\pi_j}. \quad (2.2)$$

Biblioteka kontribuon vetëm nëse  $a_{\pi_i} < D$ . Kapaciteti efektiv i saj është:

$$c_{\pi_i} = (D - a_{\pi_i})M_{\pi_i}. \quad (2.3)$$

Për secilën bibliotekë të aktivizuar zgjidhet një nënbashkësi librash  $X_{\pi_i} \subseteq \mathcal{B}_{\pi_i}$ , me  $|X_{\pi_i}| \leq c_{\pi_i}$ . Rezultati i zgjidhjes është:

$$\text{score}(\pi, X) = \sum_{b \in \bigcup_{i=1}^k X_{\pi_i}} s_b. \quad (2.4)$$

Objektivi është:

$$\max_{\pi, X} \text{score}(\pi, X), \quad X_{\pi_i} \subseteq \mathcal{B}_{\pi_i}, \quad |X_{\pi_i}| \leq c_{\pi_i}. \quad (2.5)$$

Ky formulim ndan qartë dy nivelet e vendimmarrjes: renditjen e bibliotekave dhe alokimin e librave. Në këtë punim, kërkimi eksploron kryesisht renditjet e bibliotekave, ndërsa alokimi i librave rindërtohet për çdo renditje përmes procedurave të shpejta të rindërtimit.

Për të lidhur këtë përfaqësim algoritmik me një model optimizimi më formal, problemi mund të shkruhet edhe si formulim MILP. Një model i tillë është paraqitur fillimisht në punën e përbashkët paraprake [11]. Versioni i mëposhtëm e riorganizon notacionin në përputhje me këtë tezë dhe përdor konstante të tipit *big-M* për të trajtuar bibliotekat joaktive. Le të jenë variablat:

- $y_\ell \in \{0, 1\}$ , që tregon nëse biblioteka  $\ell$  regjistrohet.
- $z_{\ell b} \in \{0, 1\}$ , që tregon nëse libri  $b$  skanohet nga biblioteka  $\ell$ .
- $u_b \in \{0, 1\}$ , që tregon nëse libri  $b$  skanohet të paktën një herë.
- $p_{\ell q} \in \{0, 1\}$ , që tregon nëse biblioteka  $\ell$  regjistrohet para bibliotekës  $q$ .
- $\tau_\ell \in \{0, \dots, D - 1\}$ , dita kur fillon regjistrimi i bibliotekës  $\ell$ .

## 2. Problemi i skanimit të librave

Le të jenë  $H^T = D + \max_{\ell \in \mathcal{L}} T_\ell$  dhe  $H_\ell^C = |\mathcal{B}_\ell| + M_\ell D$ . Madhësitë  $H^T$  dhe  $H_\ell^C$  janë konstante të tipit *big-M*, jo variabla vendimmarrëse. Konstantja  $H^T$  bën që kushti për përfundimin e regjistrimit të zbatohet vetëm kur  $y_\ell = 1$  dhe që kushti i renditjes ndërmjet bibliotekave  $\ell$  dhe  $q$  të zbatohet vetëm kur  $p_{\ell q} = 1$ . Kur këto variabla binare kanë vlerën zero, termi me  $H^T$  e zgjeron anën e djathtë të kufizimit, kështu që koha e një biblioteke ose një marrëdhënie renditjeje joaktive nuk e kufizon planin. Në mënyrë të ngjashme,  $H_\ell^C$  bën që kufizimi i kapacitetit të skanimit të zbatohet vetëm kur biblioteka  $\ell$  zgjidhet, pra kur  $y_\ell = 1$ . Kur  $y_\ell = 0$ , kufizimi  $z_{\ell b} \leq y_\ell$  ndalon skanimin e librave nga ajo bibliotekë, ndërsa termi me  $H_\ell^C$  ruan realizueshmërinë e kufizimit të kapacitetit. Duke përdorur këto konstante, merret formulimi i mëposhtëm:

$$\max \sum_{b \in \mathcal{B}} s_b u_b \quad (2.6)$$

$$\text{s.t. } z_{\ell b} \leq u_b, \quad \forall \ell \in \mathcal{L}, b \in \mathcal{B}_\ell, \quad (2.7)$$

$$u_b \leq \sum_{\ell: b \in \mathcal{B}_\ell} z_{\ell b}, \quad \forall b \in \mathcal{B}, \quad (2.8)$$

$$z_{\ell b} \leq y_\ell, \quad \forall \ell \in \mathcal{L}, b \in \mathcal{B}_\ell, \quad (2.9)$$

$$\sum_{b \in \mathcal{B}_\ell} z_{\ell b} \leq M_\ell(D - \tau_\ell - T_\ell) + H_\ell^C(1 - y_\ell), \quad \forall \ell \in \mathcal{L}, \quad (2.10)$$

$$\tau_\ell + T_\ell \leq D + H^T(1 - y_\ell), \quad \forall \ell \in \mathcal{L}, \quad (2.11)$$

$$p_{\ell q} + p_{q\ell} \geq y_\ell + y_q - 1, \quad \forall \ell, q \in \mathcal{L}, \ell \neq q, \quad (2.12)$$

$$p_{\ell q} + p_{q\ell} \leq 1, \quad \forall \ell, q \in \mathcal{L}, \ell \neq q, \quad (2.13)$$

$$p_{\ell q} \leq y_\ell, \quad p_{\ell q} \leq y_q, \quad \forall \ell, q \in \mathcal{L}, \ell \neq q, \quad (2.14)$$

$$\tau_q \geq \tau_\ell + T_\ell - H^T(1 - p_{\ell q}), \quad \forall \ell, q \in \mathcal{L}, \ell \neq q. \quad (2.15)$$

Funksioni objektiv (2.6) maksimizon shumën e pikëve të librave unikë të skanuar. Kufizimet (2.7)–(2.8) lidhin vendimin për skanimin e librit me variablën  $u_b$ . Libri mund të skanohet nga më shumë se një bibliotekë, por në funksionin objektiv numërohet vetëm një herë. Kufizimi (2.9) lejon skanimin e librave vetëm nga bibliotekat e regjistruara. Kufizimet (2.10)–(2.11) vendosin kapacitetin efektiv të bibliotekës dhe kushtin që regjistrimi të përfundojë brenda horizontit. Kufizimet (2.12)–(2.15) imponojnë renditje sekuenciale ndërmjet bibliotekave të zgjedhura.

Ky MILP është i dobishëm për formalizim dhe për instanca të vogla, por nuk është mjeti kryesor i zgjidhjes në këtë punim. Një implementim i këtij formulimi me Google OR-Tools është realizuar në repozitorin BORT [13]. Për instancat e mëdha, numri i variablave  $z_{\ell b}$  dhe  $p_{\ell q}$  rritet shpejt. Prandaj metoda e propozuar përdor përfaqësimin me renditje bibliotekash dhe rindërtim të shpejtë të alokimit të librave.

## 2.6. Kompleksiteti i problemit dhe kufiri i sipërm

Problemi kombinon përzgjedhje nënbashkësish, renditje sekuenciale dhe kapacitete të kufizuara, duke krijuar nënstruktura të lidhura me probleme *NP-hard* të njohura (*set packing*, *knapsack*), siç diskutohet në §3.1. Në praktikë, numri i kombinimeve rritet shumë shpejt me madhësinë e instancës, prandaj kërkimi ekzakt bëhet i papërshtatshëm për instancat e mëdha të trajtuara në këtë punim.

Le të jetë  $L$  numri i bibliotekave. Nëse një zgjidhje përmban saktësisht  $k$  biblioteka, numri i përzgjedhjeve të renditura është:

$$P(L, k) = \frac{L!}{(L - k)!}. \quad (2.16)$$

Kur  $k$  mund të marrë çdo vlerë nga 0 deri në  $L$ , numri i permutimeve të pjesshme është:

$$N_L = \sum_{k=0}^L \frac{L!}{(L - k)!}. \quad (2.17)$$

Në përfaqësimin e përdorur nga implementimi, kërkimi vepron mbi permutime të plota të bibliotekave, prandaj vetëm shtresa e renditjes përmban  $L!$  renditje potenciale para rindërtimit të planit efektiv. Si ilustrim i shkallëzimit, për  $L = 100\,000$  biblioteka, numri  $100\,000!$  ka afërsisht 456 574 shifra dhjetore. Kjo ilustron madhësinë praktike të hapësirës së kërkimit dhe motivon përdorimin e metodave metaheuristike.

Për analizë diagnostike përdoret edhe një kufi i sipërm i thjeshtë:

$$UB = \sum_{b \in \bigcup_{\ell \in L} \mathcal{B}_\ell} s_b. \quad (2.18)$$

Ky kufi tregon maksimumin teorik nëse do të ishte e mundur të skanoheshin të gjithë librat unikë që janë të pranishëm në të paktën një bibliotekë. Megjithatë, ai nuk është kufi i ngushtë i optimumit, sepse injoron kohën e regjistrimit, kapacitetin ditor dhe ndërveprimin ndërmjet bibliotekave. Për këtë arsye, në rezultatet kryesore krahasimi bëhet kundrejt metodave për krahasim dhe konfigurimit të plotë të metodës, jo kundrejt këtij kufiri të sipërm.

## 2.7. Familjet e instancave

Vlerësimi eksperimental përdor katër familje instancash. Instancat zyrtare të Google Hash Code shërbejnë si referencë zyrtare e garës [1]. Instancat real-world-based janë marrë nga Book Scanning Problem Input Generator [14], instancat sintetike nga repozitori ndihmës Book.Scanning.Dataset [15], ndërsa instancat seed-based janë gjeneruar me parametra dhe vlera seed të kontrolluara për të mundësuar riprodhueshmëri [16]. Shtrirja numerike e familjeve të përdorura në eksperimente jepet në Tabelën 5.1.

## 2. Problemi i skanimit të librave

Figura 2.2 paraqet shkallën strukturore të instancave të përdorura në punim sipas numrit të librave dhe bibliotekave. Grafiku përdor shkallë logaritmike në të dy boshtet dhe madhësia e pikës tregon horizontin kohor. Instancat *seed-based* ruajnë kryesisht shkallën e instancave zyrtare të Google Hash Code, ndërsa instancat sintetike mbulojnë një zonë të ngjashme, por me shpërndarje më të gjerë. Në dallim prej tyre, familja *real-world-based* formon një spektër nga instancat mikro deri te rastet me dy milionë libra, duke e zgjeruar hapësirën strukturore përtej instancave zyrtare, veçanërisht në dimensionin e numrit të librave.

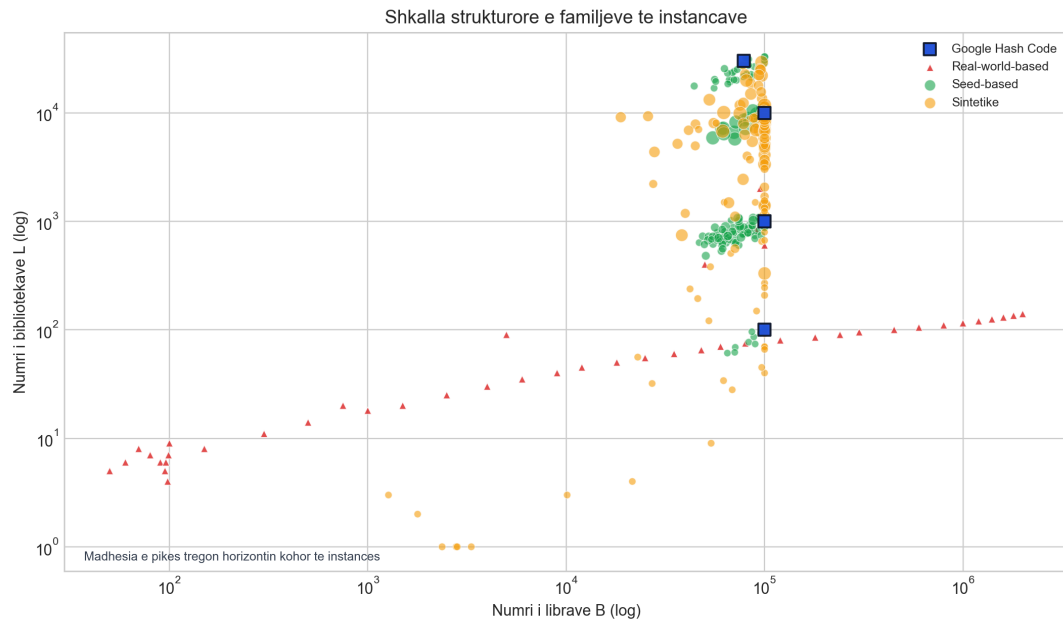


Figura 2.2: Shkalla strukturore e familjeve të instancave sipas numrit të librave dhe bibliotekave.

### 3. Rishikimi i literaturës

#### 3.1. Klasifikimi teorik i problemit

Problemi i skanimit të librave përmban nënstruktura të lidhura me probleme *NP-hard* të njohura dhe në praktikë paraqet një hapësirë kërkimi shumë të madhe. Kjo vështirësi rrjedh nga ndërthurja e tri vendimeve algoritmike: renditja sekuenciale e regjistrimit të bibliotekave, përzgjedhja e librave unikë dhe alokimi i kapaciteteve ditore të skanimit. Regjistrimi sekuencial e afron problemin me problemet e planifikimit me kohë përgatitore, ndërsa zgjedhja e librave unikë e afron atë me variante të paketimit të bashkësive (*set packing*) dhe me optimizimin nën kufizime kapaciteti të tipit *knapsack* [2]. Këto ngjashmëri strukturore shpjegojnë pse kërkimi ekzakt bëhet i papërshtatshëm në praktikë për instanca të mëdha. Renditja e bibliotekave dhe përzgjedhja e librave nuk janë dy probleme të veçanta që zgjidhen njëri pas tjetrit, por janë ngushtë të ndërrvarura. Zgjedhja e një biblioteke ndryshon librat e mbetur, kapacitetin efektiv të bibliotekave pasuese dhe vlerën marginale të vendimeve të ardhshme.

Për instanca të vogla mund të ndërtohen modele matematikore dhe formulime ekzakte, por për instanca me mijëra ose qindra mijëra libra hapësira e kërkimit bëhet tepër e madhe për zgjidhje ekzakte brenda një kohe të arsyeshme. Për këtë arsye, literatura praktike dhe zgjidhjet konkurruese të Google Hash Code dominohen nga heuristika konstruktive, kërkim lokal, rifillime dhe kombinime me metoda ekzakte për nënprobleme të vogla. Qasje të tilla nuk garantojnë optimalitet global, por synojnë zgjidhje shumë të mira nën kufizime të forta kohore.

#### 3.2. Heuristikat konstruktive

Pikënisja më e natyrshme është renditja lakmitare e bibliotekave. Në çdo hap vlerësohet kontributi i pritshëm i bibliotekave ende të paregjistruara dhe zgjidhet ajo me raportin më të mirë ndërmjet vlerës së librave, kohës së regjistrimit dhe kapacitetit ditor. Në formën më të thjeshtë, bibliotekat renditen sipas regjistrimit të shkurtër dhe totalit të lartë të librave. Format më të përpunuara i rivlerësojnë bibliotekat pas çdo zgjedhjeje, duke hequr nga konsiderata librat tashmë të skanuar dhe duke ulur vlerën e librave që shfaqen shumë shpesh.

Kjo ide është shfrytëzuar gjerësisht në zgjidhjet publike të Hash Code 2020. Kostka et al. përdorin një procedurë dyfazore: fillimisht kërkojnë një renditje të mirë të bibliotekave përmes lëvizjeve të rastësishme lokale, pastaj ndërtojnë planin final me rregulla lakmitare mbi librat e mbetur [6]. Indzhev, nga ana e tij, i vlerëson bibliotekat me peshë sipas frekuencës së librave, në mënyrë që librat e rrallë dhe me vlerë të mos humbasin nga zgjedhje të hershme të pakujdesshme [17]. Zgjidhja e Sayed-it, e zhvilluar me kontribut idesh nga Weltsch, ndjek një qasje më të gjerë, duke kombinuar një ndërtues lakmitar me metoda ekzakte (MILP, SAT) dhe me *Simulated Annealing* për nën-istanca të veçanta, duke arritur vendin e tretë nga 4 700 ekipe

në fazën shtesë (*extended round*) [5]. Të tria metodat ilustrojnë parimin themelor të problemit. Vlera e një biblioteke është dinamike dhe varet nga vendimet e mëparshme.

Heuristikat konstruktive janë shumë të shpejta dhe shpesh japin zgjidhje fillestare të forta, por vendimet e tyre janë të pakthyeshme. Një bibliotekë e zgjedhur herët mund të duket e mirë në pamje lokale, por të bllokojë skanimin e librave më të vlefshëm në biblioteka të tjera. Prandaj, në këtë punim strategjia konstruktive trajtohet si pikënisje, jo si zgjidhje finale.

### 3.3. Kërkimi lokal dhe alokimi i librave

Kërkimi lokal e përmirëson një zgjidhje duke aplikuar lëvizje të vogla mbi renditjen ose përzgjedhjen aktuale. Për Book Scanning, lëvizjet më të zakonshme janë ndërrimi i dy bibliotekave, zhvendosja e një biblioteke në pozicion tjetër, zëvendësimi i një biblioteke aktive me një joaktive dhe zhvendosja e një blloku të tërë bibliotekash. Secila prej tyre mund të ndryshojë ditët e aktivizimit të shumë bibliotekave dhe, rrjedhimisht, kapacitetin e tyre efektiv.

Një tjetër qasje e trajtuar në literaturë kombinon renditjen heuristike të bibliotekave me një alokim më të saktë të librave. Rodrigues e formulon alokimin e librave si një problem të optimizimit në rrjet, ku bibliotekat dhe librat përfaqësohen si nyje, ndërsa caktimi i librave realizohet përmes lidhjeve ndërmjet tyre. Për këtë qëllim, qasja përdor algoritmin e rrjedhës maksimale me kosto minimale (*minimum-cost maximum-flow*) për të përmirësuar alokimin e librave pasi është përcaktuar renditja e bibliotekave. Sipas tezës përkatëse, kjo qasje arriti vendin e 33-të në garë [18, 19]. Megjithëse përmirëson cilësinë e alokimit për një renditje të dhënë, zgjidhja e përsëritur e këtij nënproblemi mund të rrisë ndjeshëm kohën e ekzekutimit, veçanërisht në instanca me përmasa të mëdha.

Zgjidhja origjinale Errecto i përket ekipit të renditur në vendin e pestë në raundin kualifikues të Google Hash Code 2020 [6]. Ajo kombinon rregulla të specializuara për grupe të caktuara instancash, kërkim lokal të rastësishëm mbi renditjen e bibliotekave dhe faza më të kujdesshme alokimi për librat. Implementimi i përdorur si metodë për krahasim në këtë punim është zgjerim i kësaj zgjidhjeje dhe është përshtatur për infrastrukturën tonë eksperimentale [8]. Për këtë arsye, Errecto shërben si pikë referuese e fortë për krahasim, sidomos në instancat zyrtare dhe në ato sintetike.

### 3.4. Kërkimi lokal i përsëritur

Kërkimi lokal i përsëritur është një metaheuristikë që balancon eksplotimin e thelluar të një zgjidhjeje ekzistuese (intensifikimi) me eksplorimin e rajoneve të reja të hapësirës së kërkimit (diversifikimi) [3, 4]. Fillimisht ndërtohet një zgjidhje dhe përmirësohet me kërkim lokal derisa të mos gjendet më përmirësim i lehtë. Zgjidhja e arritur më pas perturbohet për të dalë nga optimumi lokal, përmirësohet sërish me kërkim lokal dhe pranohet ose refuzohet sipas një kriteri të kontrolluar.



Në kontekstin e Book Scanning, ILS përshtatet natyrshëm, sepse renditja e bibliotekave ka strukturë të qartë fqinjësie. Ndryshime të vogla në rend mund të sjellin ndryshime të mëdha në kapacitetet e mbetura. Një metaheuristikë me perturbim dhe rifillim mund të eksplorojë rajone të ndryshme të hapësirës pa e humbur përfitimin e përmirësimit lokal. Aplikimet e fundit të ILS në probleme të planifikimit tregojnë se performanca e kësaj metaheuristike varet nga kombinimi i nisjeve të shumëfishta, mekanizmave të perturbimit, kontrollit të realizueshmërisë dhe vlerësimit të shpejtë të lëvizjeve [20]. Në këtë punim, e njëjta logjikë përshtatet për Book Scanning duke u përqendruar vetëm në variantin ILS, me analizë të gjerë të vlerave *seed*, të kontributit të komponentëve, të konvergencës dhe të krahasimit me metodat për krahasim.

Një punë paraprake e botuar në bashkëautorësi nga kandidati trajtoi ILS, GA dhe ndërtimin GRASP për problemin Book Scanning [11]. Teza vazhdon vetëm drejtimin ILS. Marrëdhënia me punën paraprake dhe kontributi individual sqarohen në §1.6.

## 3.5. GRASP dhe diversifikimi i ndërtimit fillestar

GRASP ndërton zgjidhje duke kombinuar qasjen lakmitare me një element të kontrolluar rastësie [21]. Në vend që të zgjedhë gjithmonë bibliotekën me prioritetin më të lartë, metoda krijon një listë të kufizuar kandidatësh me cilësi të lartë dhe përzgjedh një prej tyre në mënyrë rastësore. Në këtë mënyrë gjenerohen zgjidhje fillestare të ndryshme, duke ruajtur njëkohësisht cilësinë e tyre.

Në këtë punim, parimi i diversifikimit shfrytëzohet gjatë ndërtimit të zgjidhjeve fillestare. Synimi nuk është të zëvendësohet kërkimi lokal, por të rritet mundësia që ILS të nisë kërkimin nga një rajon premtues i hapësirës së zgjidhjeve. Kjo është veçanërisht e rëndësishme kur disa biblioteka kanë vlera marginale të ngjashme, ndërsa përzgjedhja e njëres prej tyre ndikon në vlerësimin e bibliotekave të tjera.

## 3.6. Akordimi automatik i parametrave

Metaheuristikat janë të ndjeshme ndaj parametrave: forca e perturbimit, kriteri i pranimit, pragu i stagnimit, peshat e operatorëve dhe madhësia e listës së kandidatëve ndikojnë drejtpërdrejt në sjelljen e tyre. Akordimi manual mund të jetë subjektiv dhe rrezikon mbipërshtatje ndaj një numri të vogël instancash.

iRace përdor një proces garues iterativ ndërmjet konfigurimeve për të zgjedhur parametra me performancë të mirë në një grup trajnimi [22]. Kjo qasje i shkon për shtati ILS-it, sepse shumë parametra ndikojnë në mënyrë jolineare dhe nuk mund të optimizohen lehtë një nga një. Në këtë punim, konfigurimi final është rezultat i akordimit automatik dhe vlerësohet më pas në instanca testuese të ndara.

### 3.7. Modelet e mëdha gjuhësore dhe kërkimi automatik i funksioneve heuristike

Një zhvillim i rëndësishëm në kërkimet e fundit është përdorimi i modeleve të mëdha gjuhësore për gjenerimin dhe përmirësimin automatik të funksioneve të vlerësimit. Në këtë drejtim, Romera-Paredes et al. prezantojnë FunSearch, një qasje që kombinon gjenerimin e programeve nga modele gjuhësore me vlerësimin automatik të performancës së tyre [9]. Në një drejtim të ngjashëm, Veličković et al. e zbatojnë këtë ide për të përforcuar performancën e programuesve njerëzorë në probleme kombinatorike të garave të programimit, ku funksionet e vlerësimit evoluojnë automatikisht për t'u ofruar atyre heuristika më efektive [10]. Parimet e kësaj qasjeje lidhen konceptualisht edhe me problemin Book Scanning të Google Hash Code, pasi funksionet e prioritetit dhe rregullat heuristike mund të kërkohen në hapësirën e programeve. Megjithatë, kjo qasje mbetet e ndryshme nga metaheuristika klasike që trajtohet në këtë punim.

Ndryshe nga FunSearch, i cili fokusohet në gjenerimin dhe përmirësimin automatik të funksioneve të prioritetit, ky punim synon projektimin dhe analizën e një sistemi ILS me komponentë të kontrollueshëm. Kjo mundëson vlerësimin e veçantë të ndikimit të ndërtimit të zgjidhjes fillestare, kërkimit lokal, mekanizmave të perturbimit dhe rifillimit, operatorëve të përdorur, si edhe të variacionit ndërmjet ekzekutimeve me vlera të ndryshme të parametrut *seed*. Si rezultat, qasja e propozuar ofron një bazë më të përshtatshme për analizimin shkencor të kontributit të secilit komponent.

### 3.8. Metodat për krahasim të përdorura në këtë punim

Krahasimi eksperimental përdor disa metoda burimore, të përshkruara në hollësi në Kapitullin 5: Errecto, Greedy, MILP1, MILP2, SAT dhe SA [5–8]. Këto lejojnë që ILS të krahasohet jo me një rezultat të vetëm, por me grupin më të fortë të disponueshëm për secilën instancë.

Ky pozicionim e vendos punimin ndërmjet zgjidhjeve praktike të garës dhe analizës akademike të metaheuristikave. Qëllimi nuk është thjesht arritja e një rezultati të lartë, por të kuptuarit se pse dhe kur komponentët e metodës sjellin përmirësim.

## 4. Metodologjia e propozuar

### 4.1. Vështrim i përgjithshëm

Metoda e propozuar është një sistem i bazuar në kërkimin lokal të përsëritur. Ajo nis me ndërtimin e disa zgjidhjeve fillestare, prej të cilave zgjedh kandidatin më të mirë dhe e përmirëson me kërkim lokal. Më pas vazhdon me cikle të kontrolluara perturbimi, përmirësimi dhe rifillimi. Figura 4.1 paraqet rrjedhën kryesore të metodës.

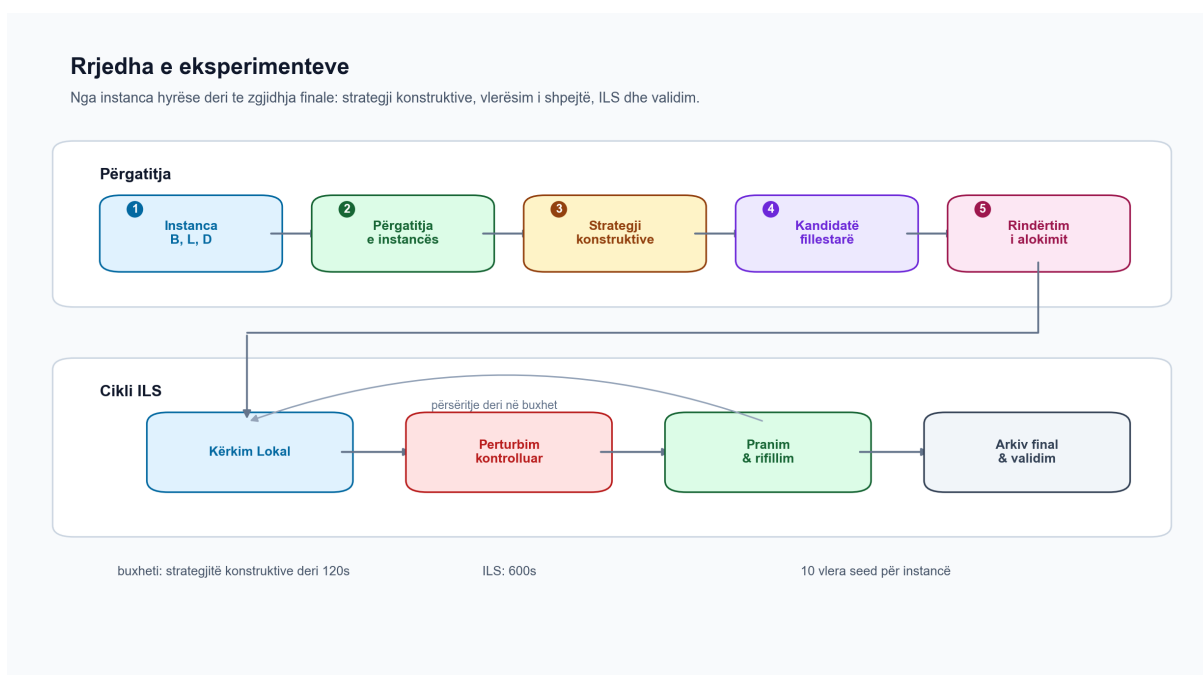


Figura 4.1: Rrjedha e përgjithshme e metodës së propozuar ILS.

Parimi kryesor është ndarja e problemit në dy nivele. Niveli i parë eksploron renditjet e bibliotekave. Niveli i dytë rindërton zgjidhjen dalëse për një renditje të dhënë, duke llogaritur cilat biblioteka përfundojnë regjistrimin në kohë dhe cilët libra mund të skanohen prej tyre. Kjo ndarje e bën kërkimin më efikas, sepse operatorët punojnë mbi një strukturë të thjeshtë, ndërsa detajet e alokimit të librave rindërtohen vetëm kur duhet vlerësuar një kandidat.

### 4.2. Përfaqësimi i zgjidhjes

Një zgjidhje përfaqësohet si renditje e bibliotekave. Pjesa e parë e renditjes përmban bibliotekat që arrijnë të regjistrohen dhe të kontribuojnë brenda horizontit kohor, ndërsa pjesa tjetër përmban bibliotekat që mbeten jashtë planit efektiv të skanimit. Ky përfaqësim nuk ruan vendime ditore të skanimit, sepse ato mund të rindërtohen nga renditja, kapaciteti ditor dhe librat e mbetur.

## 4. Metodologjia e propozuar

Për një renditje të dhënë, metoda lexon bibliotekat një nga një. Dita kur aktivizohet secila bibliotekë varet nga sa kohë regjistrimi është shpenzuar deri në atë pikë. Nëse biblioteka aktivizohet para përfundimit të horizontit, llogaritet kapaciteti i saj total për ditët e mbetura. Në të kundërt, biblioteka anashkalohet dhe shqyrtohet biblioteka e radhës në renditje, e cila mund të ketë kohë më të shkurtër regjistrimi dhe të hyjë ende në afat. Për bibliotekat e aktivizuara zgjidhen librat më të vlefshëm që gjenden në to dhe që nuk janë skanuar nga bibliotekat e mëparshme.

Figura 4.2 paraqet këtë ide në mënyrë vizuale. Pjesa aktive e renditjes përmban bibliotekat e regjistruara dhe librat që u caktohen pas rindërtimit të planit të skanimit. Pjesa joaktive përmban bibliotekat që nuk hyjnë në planin efektiv të skanimit, ndërsa librat e paskanuar mbeten jashtë rezultatit. Kërkimi lokal ndryshon kryesisht renditjen dhe kufirin ndërmjet këtyre dy pjesëve. Procedura e rindërtimit cakton listat e librave për çdo kandidat.

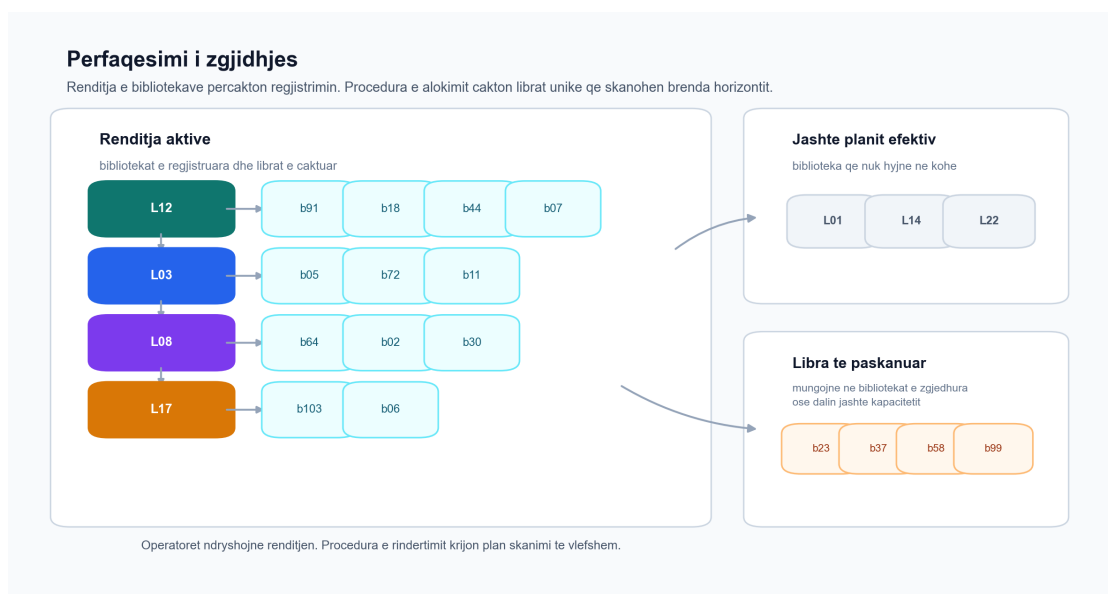


Figura 4.2: Përfaqësimi i zgjidhjes si renditje bibliotekash, bibliotekat jashtë planit efektiv të skanimit dhe librat e paskanuar.

Ky përfaqësim ka tri përparësi. Së pari, është kompakt, sepse një renditje bibliotekash është shumë më e vogël se një plan i plotë ditor. Së dyti, garanton që nga çdo kandidat mund të rindërtohet një zgjidhje dalëse e vlefshme. Së treti, lejon lëvizje lokale të interpretuara qartë, si zhvendosja e një biblioteke më herët, zëvendësimi i një biblioteke të dobët me një më premtuese ose riorganizimi i një blloku bibliotekash.

### 4.3. Alokimi i librave

Pasi të jepet renditja e bibliotekave, duhet vendosur cilët libra skanohen nga secila bibliotekë e regjistruar. Kjo fazë rindërton zgjidhjen dalëse nga renditja dhe kapacitetet efektive të bibliotekave. Metoda përdor disa procedura alokimi, sepse e njëjta renditje bibliotekash mund

të japë rezultate të ndryshme varësisht nga mënyra se si caktohen librat.

Alokimi sekuencial ndjek logjikën më të drejtpërdrejtë të problemit. Bibliotekat shqyrtohen sipas renditjes së regjistrimit. Për çdo bibliotekë aktive merren librat ende të paskanuar me vlerë më të lartë, deri në kapacitetin e saj. Ky variant është shumë i shpejtë dhe përputhet mirë me interpretimin e zgjidhjes dalëse zyrtare.

Alokimi global e shikon vendimin nga këndvështrimi i librave. Librat me vlerë të lartë shqyrtohen të parët dhe secili libër i caktohet bibliotekës aktive ku mund të skanohet pa tejkaluar kapacitetin. Kjo mund të përmirësojë rastet ku një libër me vlerë të madhe ndodhet në disa biblioteka, sepse alokimi nuk varet vetëm nga biblioteka që shfaqet e para në renditje.

Alokimi i balancuar synon të shmangë mbushjen e parakohshme të disa bibliotekave kur ka shumë libra kandidatë me vlerë të ngjashme. Ai favorizon shpërndarje më të ekuilibruar të librave ndërmjet bibliotekave aktive, sidomos në instanca ku bibliotekat kanë kapacitete të mëdha dhe koleksione që mbivendosen. Në instanca shumë të mëdha, metoda kufizon përdorimin e procedurave më të shtrenjta të alokimit për të ruajtur buxhetin e kërkimit.

Algoritmi 1 jep skemën e alokimit sekuencial, e cila përdoret si bazë për vlerësimin e një renditjeje. Në këtë variant, librat kandidatë të secilës bibliotekë renditen sipas pikëve në rend zbritës, prandaj librat me vlerë më të lartë merren të parët. Në algoritëm,  $\mathcal{P}$  shënon bashkësinë e librave tashmë të skanuar,  $S$  planin dalës dhe  $z$  rezultatin e akumuluar. Variantet globale dhe të balancuara ndryshojnë mënyrën e shpërndarjes së kapacitetit. Kufizimet kohore dhe mosnumërimi i dyfishtë i librave mbeten të njëjta.

---

**Algoritmi 1** Rindërtimi i zgjidhjes nga renditja  $\pi$  për instancën  $I$

---

- 1:  $t \leftarrow 0, \mathcal{P} \leftarrow \emptyset, S \leftarrow [], z \leftarrow 0.$
  - 2: **për** çdo bibliotekë  $\ell$  në renditjen  $\pi$  **bëj**
  - 3:   **nëse**  $t + T_\ell \geq D$  **atëherë** vazhdo.
  - 4:    $t \leftarrow t + T_\ell.$
  - 5:    $c_\ell \leftarrow (D - t) \cdot M_\ell.$
  - 6:    $K \leftarrow \{b \in \mathcal{B}_\ell \mid b \notin \mathcal{P}\}.$
  - 7:   Rendit  $K$  sipas pikëve  $s_b$  në rend zbritës.
  - 8:    $Q \leftarrow$  librat e parë  $\min(c_\ell, |K|)$  nga  $K$ .
  - 9:   **nëse**  $Q \neq \emptyset$  **atëherë**
  - 10:     Shto  $(\ell, Q)$  në planin  $S$ .
  - 11:      $\mathcal{P} \leftarrow \mathcal{P} \cup Q$  dhe  $z \leftarrow z + \sum_{b \in Q} s_b.$
  - 12: **kthe** planin e vlefshëm  $S$  dhe rezultatin  $z$ .
- 

#### 4.4. Vlerësimi i kandidatëve

Kërkimi lokal kërkon shumë vlerësime kandidatësh. Nëse çdo lëvizje do të rindërtonte zgjidhjen dalëse të plotë në mënyrë të ngadaltë, pjesa më e madhe e kohës do të shpenzohej vetëm në vlerësim. Për këtë arsye, të dhënat e instancës ruhen si vargje numerike, në vend të objekteve

komplekse Python, ndërsa llogaritjet intensive kryhen me ekzekutim të përshpejtuar mbi to [23].

Vlerësimi ndahet në dy shkallë. Shkalla e parë është vlerësim paraprak i shpejtë, i cili simulon kontributin e një renditjeje pa rindërtuar të gjitha detajet e zgjidhjes dalëse. Ai përdoret për të filtruar lëvizje që duket qartë se nuk kanë potencial. Shkalla e dytë është vlerësim i plotë i përshpejtuar, ku kandidati rindërtohet më saktë dhe krahasohet me rezultatin aktual.

Kjo ndarje nuk e ndryshon objektivin e algoritmit. Kandidati nuk pranohet vetëm sepse duket mirë në vlerësimin paraprak. Ai duhet të kalojë edhe kontrollin më të saktë. Roli i vlerësimit paraprak është të kursejë kohë, jo të zëvendësojë funksionin real të vlerësimit.

## 4.5. Ndërtimi i zgjidhjes fillestare

Faza fillestare krijon disa renditje kandidatësh. Kjo është e rëndësishme sepse ILS është më efektiv kur fillon nga një zgjidhje që tashmë ka strukturë të mirë. Në vend të një rregulle të vetme ndërtimi, përdoren disa strategji konstruktive, ku secila i jep përparësi një aspekti tjetër të problemit.

Strategjia **Sorted** është varianti bazë deterministik. Ajo rendit bibliotekat sipas kohës më të shkurtër të regjistrimit dhe, në rast barazimi, sipas vlerës totale më të madhe të librave. Ky variant është i thjeshtë, shumë i shpejtë dhe shërben si pikë reference për të matur sa përfitim sjellin strategjitë më adaptive.

Strategjitë **Static Greedy**, **Adaptive Heap Greedy** dhe **Fill Greedy** ndjekin të njëjtin parim: prioriteti i një biblioteke rritet me potencialin e saj dhe zvogëlohet me kohën e regjistrimit, ngritur në fuqinë  $\alpha$ :

$$p_{\theta}(\ell) = \frac{\text{potenciali i bibliotekës } \ell}{\max(1, T_{\ell})^{\alpha}}. \quad (4.1)$$

Potenciali përafron vlerën që biblioteka  $\ell$  do të sillte nëse zgjidhet, pra sa libra të vlefshëm e ende të paskanuar mund të skanojë me kapacitetin e saj. Ato dallojnë vetëm në mënyrën si llogaritet potenciali. **Static Greedy** e llogarit atë ( $P_{\theta}(\ell)$ ) një herë, para ndërtimit të renditjes. Në variantin *cap/raw* potenciali kufizohet nga kapaciteti efektiv i bibliotekës, ndërsa në variantin *cap/rare* vlera e librave peshohet edhe me rrallësinë e tyre. Këto renditje ndërtohen shpejt, por ende marrin parasysh se një bibliotekë nuk mund t'i skanojë të gjithë librat e saj.

**Adaptive Heap Greedy** përdor një heap maksimal, ku prioritetet nuk rillogariten për të gjitha bibliotekat në çdo hap. Në vend të kësaj, një bibliotekë rivlerësohet vetëm kur del në krye të heap-it. Nëse pas rivlerësimit mbetet ende kandidatja më e mirë, ajo zgjidhet. Përndryshe, kthehet në heap me prioritetin e përditësuar,  $G_{\theta}(\ell \mid \mathcal{P}, t)$ , i cili mat fitimin e librave ende të paskanuar që biblioteka  $\ell$  mund t'i skanojë me kapacitetin e mbetur në ditën  $t$ . Variantet *top/raw*, *top/rare*, *cap/raw* dhe *cap/rare* dallojnë nga fakti nëse përdorin vetëm librat më të vlefshëm, kapacitetin efektiv dhe/ose peshën sipas rrallësisë. Kjo strategji është më e shtrenjtë se renditja statike, por kap më mirë efektin e mbivendosjes së librave ndërmjet bibliotekave.

**Fill Greedy** përdor të njëjtën ide të prioritetit, por i jep përparësi bibliotekave që mund

ta mbushin mirë kapacitetin e tyre të mbetur. Një bibliotekë mund të ketë shumë libra me vlerë, por nëse ka pak ditë skanimi ose pak libra të dobishëm të mbetur, kontributi i saj real mund të jetë më i vogël. Për këtë arsye, metoda merr parasysh jo vetëm vlerën e librave, por edhe sa mirë mund të shfrytëzohet kapaciteti i bibliotekës. Varianti *raw* përdor pikët origjinale të librave, ndërsa varianti *rare* u jep peshë më të madhe librave që gjenden në më pak biblioteka.

Për instanca jo shumë të mëdha përdoret edhe **Branch Fill Greedy**. Kjo strategji mban një kufi të sipërm të lirë për bibliotekat, pastaj llogarit vlerën reale vetëm për kandidatët që mund të jenë konkurrues. Kjo ndjek të njëjtën logjikë si Adaptive Heap Greedy: shtyn rivlerësimin vetëm te kandidatët që dalin premtues, në vend që ta bëjë për të gjitha bibliotekat. Kështu shmangen shumë rivlerësime të panevojshme, por zgjidhet biblioteka që provon vlerë të lartë reale pasi hiqen librat e planifikuar më parë.

Për diversifikim përdoren dy mekanizma. **Noisy Heap** shton zhurmë të vogël të kontrolluar mbi prioritetin e bibliotekave, që krijon renditje të ndryshme pa humbur strukturën lakmitare. **GRASP** fillon nga një renditje adaptivisht e mirë, formon një listë të kufizuar kandidatësh premtues dhe zgjedh rastësisht një prej tyre. Kjo krijon pika nisjeje të ndryshme për kërkimin lokal, por shmang renditje plotësisht arbitrare.

Meqenëse përdoren disa strategji konstruktive me variante të ndryshme, Tabela 4.1 përmbledh rolin praktik të secilës. Variantet me parametra të ndryshëm nuk trajtohen si metoda të pavarura teorike, por si konfigurime të së njëjtës familje konstruktive që synojnë balancime të ndryshme ndërmjet kohës së regjistrimit, kapacitetit ditor, vlerës së librave dhe rrallësisë së tyre.

#### 4. Metodologjia e propozuar

Tabela 4.1: Përmbledhje e strategjive konstruktive për zgjidhjen fillestare.

| Strategjia           | Variantet                                                                                                | Çfarë bën                                                                                                                                                                                                          | Tipi                                   |
|----------------------|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| Sorted               | renditje sipas $t_l$ , me lidhje sipas vlerës totale të librave                                          | Jep një pikë nisjeje të thjeshtë dhe shumë të shpejtë, duke favorizuar bibliotekat me kohë të shkurtër regjistrimi. Përdoret edhe si variant për krahasim i dobët.                                                 | deterministik                          |
| Static Greedy        | $cap/raw$ , $cap/rare$                                                                                   | Llogarit një prioritet të vetëm për çdo bibliotekë para ndërtimit të renditjes. Varianti $cap$ përdor kapacitetin maksimal të skanimit, ndërsa $rare$ rrit peshën e librave që shfaqen në më pak biblioteka.       | deterministik                          |
| Adaptive Heap Greedy | $top/raw$ për disa vlera të $\alpha$ , $cap/raw$ , $cap/rare$ , dhe variante të rastësishme në rifillime | Rivlerëson bibliotekat gjatë ndërtimit të renditjes, pasi librat tashmë të mbuluar humbin vlerë. Kjo e bën më të përshtatshëm për instanca me mbivendosje të madhe të librave.                                     | deterministik / stokastik në rifillime |
| Fill Greedy          | $raw$ , $rare$                                                                                           | Favorizon bibliotekat që mund ta mbushin mirë kapacitetin e mbetur të skanimit, jo vetëm ato me vlerë të lartë teorike. Varianti $rare$ penalizon më pak librat që janë të vështirë për t'u zëvendësuar.           | deterministik                          |
| Branch Fill Greedy   | kandidatët më të mirë sipas kufirit të sipërm                                                            | Përdor një kufi të sipërm të lirë për të filtruar kandidatët dhe pastaj llogarit vlerën reale vetëm për bibliotekat premtuese. Kjo ul koston e rivlerësimit në krahasim me kontrollimin e plotë të çdo biblioteke. | deterministik                          |
| Noisy Heap           | $cap/raw$ , $cap/rare$ me zhurmë të vogël                                                                | Ndjek logjikën e Adaptive Heap Greedy, por shton perturbim të kontrolluar në vlerësim për të gjeneruar renditje alternative pranë zgjidhjeve lakmitare.                                                            | stokastik                              |
| GRASP                | listë kandidatësh e kufizuar me raport $\rho$                                                            | Ndërton renditjen duke zgjedhur rastësisht nga një grup i vogël bibliotekash premtuese. Përdoret për diversifikim, sidomos kur strategjitë deterministike japin renditje shumë të ngjashme.                        | stokastik                              |

Të gjithë kandidatët e gjeneruar vlerësohen fillimisht me një funksion të shpejtë. Vetëm kandidatët më premtues rindërtohen me procedurën e plotë të alokimit të librave. Kështu, faza fillestare mbetet efikase, ndërsa zgjidhja që i dorëzohet ILS-it nuk mbështetet vetëm në vlerësim të përafërt. Algoritmi 2 e përmbledh këtë proces.



**Algoritmi 2** Ndërtimi fillestar me disa strategji konstruktive  $(L, \mathcal{B}, D, \Theta, \rho)$ 

- 
- 1: Llogarit statistikatat bazë: vlerën totale të bibliotekave, rrallësinë e librave, kohën e regjistrimit dhe kapacitetin efektiv.
  - 2:  $\mathcal{C} \leftarrow []$ .
  - 3: **për çdo** strategji ndërtimi  $\theta \in \Theta$  **bëj**
  - 4:    $L' \leftarrow L, t \leftarrow 0, \mathcal{P} \leftarrow \emptyset, \pi_\theta \leftarrow []$ .
  - 5:   **përderisa** ekziston bibliotekë që mund të regjistrohet brenda  $D$  **bëj**
  - 6:     Për çdo  $\ell \in L'$ , vlerëso prioritetin  $p_\theta(\ell \mid \mathcal{P}, t)$ .
  - 7:     Formo listën e kufizuar  $R_\theta$  me bibliotekat më premtuese.
  - 8:     Zgjidh  $\ell^*$ : më të mirën për strategji deterministike, ose një element nga  $R_\theta$  për strategji të rastësishme lakmitare.
  - 9:     **nëse**  $t + T_{\ell^*} \geq D$  **atëherë** hiqe  $\ell^*$  nga  $L'$  dhe vazhdo.
  - 10:     $t \leftarrow t + T_{\ell^*}$  dhe shto  $\ell^*$  në  $\pi_\theta$ .
  - 11:    Përditëso përafërsisht librat e skanueshëm dhe  $\mathcal{P}$  për të ulur vlerën e mbivendosjes së librave.
  - 12:    Hiqe  $\ell^*$  nga  $L'$ .
  - 13:    Vlerëso  $\pi_\theta$  me procedurë të shpejtë dhe shto kandidatin në  $\mathcal{C}$ .
  - 14:    Rindërto me alokim më të plotë kandidatët më premtues nga  $\mathcal{C}$ .
  - 15: **kthe** kandidatin më të mirë fillestar dhe arkivin e kandidatëve të mirë.
- 

## 4.6. Operatorët e kërkimit lokal

Operatorët lokalë ndahen në tri grupe. Grupi i parë riorganizon bibliotekat që tashmë janë në pjesën aktive të renditjes. Këtu përfshihen ndërrimi i dy bibliotekave, zhvendosja e një biblioteke në pozicion tjetër dhe lëvizja e një blloku të shkurtër bibliotekash. Këto operatorë synojnë të gjejnë rend më të mirë për bibliotekat që algoritmi tashmë i konsideron të dobishme.

Grupi i dytë ndryshon kufirin ndërmjet bibliotekave aktive dhe atyre joaktive. Një bibliotekë e regjistruar mund të zëvendësohet me një bibliotekë jashtë planit efektiv të skanimit, ose një bibliotekë e re mund të futet në një pozicion ku koha e mbetur i lejon të kontribuojë. Këto lëvizje janë të rëndësishme sepse një optimum lokal mbi të njëjtin grup bibliotekash mund të mbetet i dobët nëse mungon një bibliotekë me vlerë të lartë marginale.

Grupi i tretë përdor informacion strukturor. Ajo kërkon biblioteka me kohë regjistrimi të shkurtër, libra të rrallë ose kontribut të lartë në pjesët kritike të renditjes. Qëllimi është të provohen lëvizje më të informuara se një shkëmbim i rastësishëm, sidomos kur hapësira e kandidatëve është e madhe.

Figura 4.3 paraqet grupet e operatorëve sipas rolit që luajnë në kërkim. Ky organizim thekson dallimin ndërmjet lëvizjeve që riorganizojnë bibliotekat aktive, lëvizjeve që ndryshojnë përbërjen e pjesës aktive dhe lëvizjeve që shfrytëzojnë informacion strukturor të instancës.

## 4. Metodologjia e propozuar



Figura 4.3: Ilustrim skematik i grupeve kryesore të operatorëve lokalë.

Tabela 4.2 i lidh emrat e operatorëve të implementuar me rolin e tyre metodologjik. Emrat në kolonën e parë ruhen në formën e implementimit për ta bërë lidhjen me eksperimentet dhe statistikat e operatorëve të drejtpërdrejtë.

#### 4. Metodologjia e propozuar

Tabela 4.2: Operatorët e kërkimit lokal dhe roli i tyre.

| Operatori                 | Grupi            | Lëvizja që kryen                                                                           | Qëllimi praktik                                                            |
|---------------------------|------------------|--------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| swap_signed               | Rirenditje       | Ndërron dy biblioteka aktive në renditje.                                                  | Korrigjon rendin relativ të bibliotekave të zgjedhura.                     |
| move_signed               | Rirenditje       | Zhvendos një bibliotekë aktive në një pozicion tjetër aktiv.                               | Gjen pozicion më të mirë për një bibliotekë pa ndryshuar përbërjen aktive. |
| swap_neighbor_libraries   | Rirenditje       | Ndërron dy biblioteka fqinje në pjesën aktive.                                             | Bën përmirësime lokale shumë të imëta.                                     |
| reverse_segment           | Rirenditje       | Kthen rendin e një segmenti aktiv.                                                         | Provon ndryshim më të gjerë se shkëmbimi i dy bibliotekave.                |
| block_reinsert            | Rirenditje       | Heq një bllok bibliotekash aktive dhe e vendos në pozicion tjetër.                         | Ruan strukturë të pjesshme, por ndryshon kohën e nisjes së bllokut.        |
| swap_signed_with_unsigned | Futje/zëvendësim | Shkëmben një bibliotekë aktive me një joaktive.                                            | Ndryshon përbërjen e planit efektiv të skanimit.                           |
| insert_library            | Futje/zëvendësim | Merr një bibliotekë joaktive dhe e fut në pjesën aktive.                                   | Teston kandidatë që ndërtimi fillestar i ka lënë jashtë.                   |
| critical_path_insert      | Futje/zëvendësim | Zgjedh bibliotekë të shpejtë me libra të vlefshëm ose të rrallë dhe e fut herët.           | Shfrytëzon hapësira ku koha e regjistrimit është kufizuese.                |
| remove_library            | Futje/zëvendësim | Largon një bibliotekë aktive drejt pjesës joaktive.                                        | Liron kohë për riorganizim ose futje të mëvonshme.                         |
| diversity_swap            | Strategjik       | Zëvendëson një bibliotekë me kontribut të dobët me kandidat që shton libra të rinj.        | Rrit mbulimin dhe shmang mbivendosjen e librave.                           |
| sampled_best_exchange     | Strategjik       | Mostron disa zëvendësime aktive/joaktive dhe mban më të mirin sipas vlerësimit të shpejtë. | Gjen shkëmbime me potencial të lartë pa kontroll të plotë të hapësirës.    |
| coverage_exchange         | Strategjik       | Përdor mbulimin e librave për instanca me strukturë uniforme.                              | Maksimizon libra të mbuluar kur vlerat dhe kapacitetet janë të ngjashme.   |
| paired_choice_flip        | Strategjik       | Ndryshon zgjedhjen në çifte bibliotekash kur struktura e instancës e lejon.                | Shfrytëzon instanca me zgjedhje komplementare të tipit çift.               |

Kërkimi lokal aplikohet në dy faza. Fillimisht provohen shumë lëvizje me vlerësim paraprak të lirë dhe verifikim të shpejtë. Më pas, kur gjendet një rajon premtues ose kur mbetet kohë, aplikohen lëvizje më të shtrenjta me vlerësim më të plotë. Kjo kombinon shpejtësinë me saktësinë.

Faza e dytë nuk fut një grup të ri operatorësh. Ajo përdor të njëjtat ide bazë me rindërtim më të plotë të alokimit, sidomos shkëmbime ndërmjet bibliotekave të para në renditje, zhvendosje të shkurtra dhe shkëmbime në kufirin ndërmjet bibliotekave aktive dhe joaktive. Kjo fazë shërben si rindërtim final pas eksplorimit më të shpejtë.

## **4.7. Perturbimi, pranimi dhe rifillimi**

Kërkimi lokal mund të ngeçë në optimum lokal. Për këtë arsye, pas përmirësimit lokal aplikohet perturbim. Renditja aktuale ndryshohet në mënyrë të kontrolluar, për shembull duke zhvendosur një segment, duke zëvendësuar disa biblioteka ose duke riorganizuar një pjesë të renditjes. Perturbimi nuk synon ta shkatërrojë zgjidhjen, por ta largojë atë mjaftueshëm nga optimumi lokal ku ka ngecur.

Kandidati i perturbuar përmirësohet sërish me kërkim lokal. Nëse rezultati del më i mirë se zgjidhja bazë, ai pranohet. Nëse del më i dobët, mund të pranohet me probabilitet të kufizuar, sidomos në fazat e hershme të kërkimit. Ky mekanizëm hap rrugë për eksplorim më të gjerë dhe shmang kthimin e menjëhershëm në të njëjtën zgjidhje lokale.

Kur nuk ka përmirësim për disa raunde, aktivizohet rifillimi. Rifillimi mund të kthehet te një zgjidhje e mirë e ruajtur më parë ose të ndërtojë një zgjidhje të re përmes strategjive konstruktive. Ky mekanizëm është veçanërisht i dobishëm në instanca ku disa renditje kanë rezultat të ngjashëm, por vetëm disa prej tyre hapin rrugë për përmirësime të mëvonshme.

## **4.8. Akordimi i konfigurimit final**

Parametrat e metodës janë akorduar me iRace [22], mbi ndarjen trajnim/testim të përshkruar në §5.2. Tabela 4.3 përmbledh parametrat kryesorë në formë metodologjike, duke raportuar vlerën finale dhe intervalin ose bashkësinë e kërkimit të përdorur gjatë akordimit. Ajo tregon se konfigurimi final i jep peshë të konsiderueshme lëvizjeve që fusin ose shkëmbejnë biblioteka dhe lëvizjeve strategjike, ndërsa rirenditja mbetet aktive me peshë më të ulët.

#### 4. Metodologjia e propozuar

Tabela 4.3: Parametrat kryesorë të konfigurimit të akorduar dhe hapësira e kërkimit në iRace.

| Parametri                         | Vlera            | Intervali                               | Roli                                                                                                  |
|-----------------------------------|------------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------|
| Pranimi fillestar i dobësimit     | 0.0320           | (0.001, 0.15)                           | Shkalla fillestare e tolerancës për pranimin probabilistik të një kandidati rreth 1% më të dobët      |
| Ftohja drejt fundit të kërkimit   | 0.1699           | (0.01, 0.25)                            | Raporti ndërmjet temperaturës finale dhe asaj fillestare në kriterin e pranimi                        |
| Pragu i stagnimit                 | 3                | [2, 12]                                 | Numri i raundeve pa përmirësim para se të aktivizohet rifillimi                                       |
| Forca bazë e perturbimit          | 2                | [2, 15]                                 | Madhësia minimale e ndryshimit që aplikohet mbi renditjen aktuale                                     |
| Rritja e perturbimit              | 0                | [0, 6]                                  | Në konfigurimin final perturbimi nuk rritet automatikisht me stagnimin                                |
| Lista e kufizuar e kandidatëve    | 0.1308           | (0.01, 0.25)                            | Pjesa e bibliotekave më premtuese prej nga zgjedh strategjia e rastësishme lakmitare                  |
| Buxheti i rindërtimit në rifillim | 0.1491           | (0.10, 0.50)                            | Pjesa e kohës së mbetur që mund t'i jepet një rindërtimi të freskët                                   |
| Prirja drejt zëvendësimit         | 0.6794           | (0.30, 0.90)                            | Sa shpesh perturbimi zëvendëson një nënrenditje në vend se vetëm ta riorganizojë atë                  |
| Rifillim nga zgjidhje e re        | 0.4035           | (0.10, 0.70)                            | Probabiliteti që rifillimi të ndërtojë një renditje të re, jo vetëm të kthehet te arkivi i zgjidhjeve |
| Kufiri lokal i stagnimit          | 316              | [50, 500]                               | Numri maksimal i provave lokale pa përmirësim para ndërprerjes së fazës lokale                        |
| Pesha e rirenditjes               | 0.6317           | (0.40, 2.50)                            | Pesha relative e lëvizjeve që ndryshojnë rendin e bibliotekave të regjistruara                        |
| Pesha e futjes dhe shkëmbimit     | 2.3634           | (0.40, 2.50)                            | Pesha relative e lëvizjeve që fusin biblioteka të reja ose shkëmbejnë biblioteka aktive me joaktive   |
| Pesha e lëvizjeve strategjike     | 1.9964           | (0.40, 2.50)                            | Pesha relative e lëvizjeve që synojnë kufirin kritik të kapacitetit dhe vlerën marginale              |
| Eksponentët e ndërtimit lakmitar  | gama eksploruese | {default, wide, dense, signup, explore} | Bashkësia e vlerave për balancimin e rezultetit, kohës së regjistrimit dhe rrallësisë së librave      |

Suksesi i akordimit me iRace në këtë punim vlerësohet si validim i procesit të zgjedhjes së parametrave, jo si krahasim i izoluar statistikor ndaj një konfigurimi manual. Për një krahasim të tillë do të duhej që konfigurimi manual dhe konfigurimi i akorduar të ekzekutoheshin me të njëjtin protokoll, të njëjtat instanca testuese, të njëjtin buxhet kohor dhe të njëjtat vlera *seed*. Në këtë punim, iRace verifikohet në dy nivele: së pari, përmes një procesi të kontrolluar trajnimi me hapësirë të dokumentuar kërkimi dhe buxhet prej 5000 eksperimenteve. Së dyti, përmes përdorimit të konfigurimit final pa ndryshime në grupin e ndarë të testimit, sipas protokollit të përsëritjeve me *seed* nga §5.3. Prandaj, rezultatet finale interpretojnë iRace si mekanizëm të kontrolluar për zgjedhjen e parametrave, ndërsa jo si provë se çdo parametër i akorduar është domosdoshmërisht më i mirë se çdo vlerë manuale e mundshme.

Variantet e eksperimentimit me komponentët e algoritmit nuk duhen lexuar si konfigurime “pa iRace”. Ato nisen nga konfigurimi i akorduar dhe çaktivizojnë vetëm një komponent ose grup operatorësh, për të matur rolin e tij. Rrjedhimisht, këto rezultate përbëjnë analizë të kontributit të komponentëve brenda konfigurimit final, jo analizë të efektit të akordimit.

## 4.9. Pseudokodi i metodës

Algoritmi 3 përmbledh metodën pa hyrë në detaje implementimi. Ai tregon qartë ndarjen ndërmjet ndërtimit fillestar, përmirësimit lokal, perturbimit, pranimit dhe rifillimit.

---

### Algoritmi 3 Kërkimi lokal i përsëritur për Book Scanning ( $I, T_0, T$ )

---

- 1: Lexo instancën  $I$  dhe ndërto strukturat numerike për bibliotekat, librat, vlerat e librave, kohët e regjistrimit dhe kapacitetet ditore.
  - 2:  $(S, A) \leftarrow$  ndërtimi fillestar me strategji konstruktive brenda buxhetit  $T_0$ .
  - 3:  $S \leftarrow$  përmirësimi lokal i zgjidhjes fillestare  $S$ .
  - 4:  $S^* \leftarrow S, S_b \leftarrow S, h \leftarrow 0$ .
  - 5: **përderisa** buxheti  $T$  nuk ka përfunduar **bëj**
  - 6:    $S' \leftarrow$  perturbim i kontrolluar i  $S_b$ .
  - 7:    $S' \leftarrow$  kërkim lokal mbi  $S'$  me vlerësim paraprak dhe verifikim të plotë.
  - 8:   **nëse**  $\text{score}(S') > \text{score}(S^*)$  **atëherë**  $S^* \leftarrow S'$  dhe  $h \leftarrow 0$ .
  - 9:   **përndryshe**  $h \leftarrow h + 1$ .
  - 10:   Prano  $S'$  si  $S_b$  nëse është më i mirë se  $S_b$  ose plotëson kriterin probabilistik të pranimit.
  - 11:   **nëse**  $h$  tejkalon pragun e rifillimit **atëherë**
  - 12:      $S_b \leftarrow$  zgjidhje nga arkivi  $A$  ose zgjidhje e re përmes strategjive konstruktive.
  - 13:      $h \leftarrow 0$ .
  - 14: Rindërto  $S^*$  me procedurën e Algoritmit 1, eksporto zgjidhjen dalëse dhe valido kufizimet.
  - 15: **kthe** zgjidhjen më të mirë  $S^*$  dhe rezultatin përkatës.
- 

Ky pseudokod pasqyron logjikën shkencore të metodës. Detajet praktike, si ruajtja e vlerave *seed*, validimi i zgjidhjes dalëse dhe regjistrimi i metadatave, trajtohen në dizajnin eksperimental.

## 5. Dizajni eksperimental dhe vlerësimi

Ky kapitull përshkruan eksperimentet e kryera për të vlerësuar metodën e propozuar në Kapitullin 4: mjedisin eksperimental, ndarjen e instancave, protokollin e vlerave *seed*, metodat për krahasim dhe analizat statistikore e strukturore të përdorura.

### 5.1. Mjedi eksperimental

Eksperimentet janë kryer me implementimin e metodës së propozuar në Python, duke përdorur Numba për shpejtësi më të madhe llogaritëse [23]. Ekzekutimet kryesore janë realizuar në një instancë Linux me 8 vCPU dhe 32 GB RAM të Google Cloud Compute Engine, me Python 3.11.2 dhe platformën `Linux-6.1.0-44-cloud-amd64-x86_64-with-glibc2.36`. Eksperimentet janë ekzekutuar në mënyrë paralele me deri në 6 procese punuese. Paralelizimi është përdorur vetëm për të zvogëluar kohën totale të eksperimenteve dhe nuk ndryshon buxhetin kohor të një ekzekutimi individual. Çdo rresht rezultati ruan statusin e validimit, metodën, variantin eksperimental, instancën, familjen e instancës, vlerën *seed*, rezultatin e zgjidhjes fillestare, rezultatin final, kufirin diagnostik të sipërm, kohën reale të ekzekutimit dhe metadat teknike.

Konfigurimi final, ILS-i i propozuar me parametrat e akorduar, vlerësohet me buxhet kohor të ndarë në dy pjesë: deri në 120 sekonda për ndërtimin e zgjidhjes fillestare dhe 600 sekonda për fazën ILS. Koha reale e raportuar në tabela përfshin ndërtimin, kërkimin ILS, eksportimin dhe validimin e zgjidhjes dalëse.

### 5.2. Ndarja në grupe trajnimi dhe testimi

Vlerësimi përdor 324 instanca unike nga katër familje: Google Hash Code, real-world-based, seed-based dhe sintetike. Grupi i trajnimit (*train dataset*) shërben për akordimin e parametrave dhe zgjedhjen e konfigurimit, ndërsa grupi i testimit (*test dataset*) përdoret vetëm për raportimin final të rezultateve. Kjo ndarje shmang raportimin e performancës finale në të njëjtat instanca ku janë akorduar parametrat.

Ndarja e instancave u realizua në mënyrë deterministe brenda secilës familje të gjeneruar. Instancat u morën sipas rendit të përcaktuar në katalogun e të dhënave dhe çdo e pesta instancë u caktua në grupin e testimit, ndërsa katër instancat e tjera në grupin e trajnimit. Pesë instancat zyrtare të Google Hash Code u përfshinë vetëm në grupin e vlerësimit si instanca publike referuese për krahasim.

## 5. Dizajni eksperimental dhe vlerësimi

Tabela 5.1: Ndarja trajnim/testim dhe numri i ekzekutimeve finale ILS në instancat e testimit.

| Familja          | Trajnim | Testim | Gjithsej | Ekzekutime ILS (10× Testim) |
|------------------|---------|--------|----------|-----------------------------|
| Google Hash Code | 0       | 5      | 5        | 50                          |
| Real-world-based | 36      | 9      | 45       | 90                          |
| Seed-based       | 108     | 27     | 135      | 270                         |
| Sintetike        | 112     | 27     | 139      | 270                         |
| Total            | 256     | 68     | 324      | 680                         |

Instancat zyrtare të Google Hash Code mbahen vetëm në grupin e testimit, sepse shumë metoda për krahasim janë dizajnuar drejtpërdrejt për to. Tabela 5.2 përmbledh instancat e trajnimit në grupe sipas karakteristikës më përfaqësuese të secilës familje. Instancat real-world-based grupohen sipas madhësisë së koleksionit, pra numrit të librave  $B$ . Instancat seed-based grupohen sipas familjes bazë nga e cila janë zgjeruar, me shënimet b, c, d, e dhe f. Instancat sintetike ndahen në katër grupe me numër të barabartë sipas kuartileve të numrit të bibliotekave  $L$ , të shënuara si Q1–Q4. Intervalet min–max në rreshtat Q1–Q4 janë vlerat reale brenda secilit grup, prandaj kufijtë nuk janë domosdoshmërisht të njëpasnjëshëm. Për këto grupe raportohen intervalet e librave, bibliotekave dhe ditëve, si dhe medianat e kopjeve, vlerës së librave, frekuencës së librave, kohës së regjistrimit dhe kapacitetit ditor. Mediana përdoret sepse disa instanca kanë vlera ekstreme dhe mesatarja mund ta shtrembërojë përfaqësimin e grupit. Tabela 5.3 jep nga një rresht për secilën instancë të testimit dhe raportin të njëjtat karakteristika strukturore në nivel instance.

Tabela 5.2: Përmbledhje e instancave të grupit të trajnimit, të përdorura për akordimin e parametrave.

| Familja          | Grupi    | Nr. | Libra<br>min–max  | Bibl.<br>min–max | Ditë<br>min–max | Kopje<br>med. | Vlera<br>med. | Freq.<br>med. | Signup<br>med. | Kap.<br>med. |
|------------------|----------|-----|-------------------|------------------|-----------------|---------------|---------------|---------------|----------------|--------------|
| Real-world-based | Mikro    | 12  | 50–750            | 4–20             | 4–20            | 202           | 63.22         | 2.54          | 7.46           | 15.79        |
| Real-world-based | Të vogla | 8   | 1 500–25 000      | 20–90            | 21–110          | 35 348        | 487.96        | 5.25          | 21.70          | 206.36       |
| Real-world-based | Të mesme | 8   | 48 000–180 000    | 65–2 000         | 28–170          | 260 381       | 612.48        | 2.97          | 29.70          | 596.70       |
| Real-world-based | Të mëdha | 8   | 300 000–1 800 000 | 95–135           | 190–270         | 649 124       | 744.98        | 1.37          | 39.33          | 1566.78      |
| Seed-based       | Tipi b   | 8   | 64 950–100 000    | 61–102           | 59–229          | 60 843        | 100.00        | 1.77          | 11.17          | 1.00         |
| Seed-based       | Tipi c   | 15  | 54 838–100 000    | 5 746–10 925     | 99 999–100 000  | 92 805        | 300.77        | 1.96          | 505.72         | 49011.99     |
| Seed-based       | Tipi d   | 21  | 44 181–100 000    | 16 877–32 747    | 2 025–9 842     | 116 653       | 100.00        | 2.57          | 1.50           | 1.00         |
| Seed-based       | Tipi e   | 29  | 46 779–100 000    | 603–1 079        | 238–1 078       | 287 806       | 100.00        | 4.35          | 4.99           | 1.25         |
| Seed-based       | Tipi f   | 35  | 49 720–100 000    | 481–1 081        | 7 891–33 678    | 281 410       | 125.49        | 4.26          | 165.71         | 6.96         |
| Sintetike        | Q1       | 28  | 1 783–100 000     | 1–667            | 110–86 428      | 251 460       | 354.25        | 5.02          | 25.00          | 476.50       |
| Sintetike        | Q2       | 28  | 27 464–100 000    | 795–5 181        | 160–85 923      | 952 454       | 290.92        | 522.99        | 258.00         | 21127.50     |
| Sintetike        | Q3       | 28  | 41 394–100 000    | 5 332–8 504      | 6 435–100 000   | 999 888       | 291.00        | 1215.60       | 275.00         | 25110.00     |
| Sintetike        | Q4       | 28  | 52 700–100 000    | 8 649–29 302     | 13 312–100 000  | 252 410       | 263.33        | 516.85        | 276.50         | 27278.00     |



## 5. Dizajni eksperimental dhe vlerësimi

Tabela 5.3: Instancat e grupit të testimit, të përdorura për raportimin final.

| Familja          | Instanca                     | Libra     | Bibl.  | Ditë    | Kopje   | Vlera AVG | Freq. AVG | Signup AVG | Kap. AVG |
|------------------|------------------------------|-----------|--------|---------|---------|-----------|-----------|------------|----------|
| Google Hash Code | b_read_on.txt                | 100 000   | 100    | 1 000   | 100 000 | 100.00    | 1.00      | 11.81      | 1.00     |
| Google Hash Code | c_incunabula.txt             | 100 000   | 10 000 | 100 000 | 150 282 | 300.76    | 1.94      | 506.08     | 49511.38 |
| Google Hash Code | d_tough_choices.txt          | 78 600    | 30 000 | 30 001  | 220 800 | 65.00     | 2.81      | 2.00       | 1.00     |
| Google Hash Code | e_so_many_books.txt          | 100 000   | 1 000  | 200     | 491 942 | 125.49    | 4.96      | 5.43       | 1.49     |
| Google Hash Code | f_libraries_of_the_world.txt | 100 000   | 1 000  | 700     | 509 342 | 401.11    | 5.12      | 166.19     | 7.47     |
| Real-world-based | B1000k_L115_D230.txt         | 1 000 000 | 115    | 230     | 653 038 | 750.18    | 1.36      | 41.38      | 1574.59  |
| Real-world-based | B100_L9_D18.txt              | 100       | 9      | 18      | 388     | 62.46     | 3.88      | 9.44       | 20.00    |
| Real-world-based | B1k_L18_D17.txt              | 1 000     | 18     | 17      | 4 307   | 285.16    | 4.33      | 15.94      | 64.61    |
| Real-world-based | B2000k_L140_D280.txt         | 2 000 000 | 140    | 280     | 895 057 | 800.03    | 1.24      | 42.99      | 1701.08  |
| Real-world-based | B240k_L90_D180.txt           | 240 000   | 90     | 180     | 460 268 | 674.81    | 2.24      | 33.69      | 1289.23  |
| Real-world-based | B35k_L60_D120.txt            | 35 000    | 60     | 120     | 192 290 | 589.68    | 5.51      | 30.53      | 534.25   |
| Real-world-based | B6k_L35_D70.txt              | 6 000     | 35     | 70      | 36 458  | 489.69    | 6.08      | 25.17      | 185.46   |
| Real-world-based | B90_L6_D11.txt               | 90        | 6      | 11      | 155     | 50.32     | 2.09      | 5.00       | 11.50    |
| Real-world-based | B90k_L850_D21.txt            | 90 000    | 850    | 21      | 25 183  | 19.97     | 1.20      | 17.51      | 54.16    |
| Seed-based       | b_075x_012t.txt              | 89 643    | 74     | 98      | 55 768  | 100.00    | 1.69      | 11.04      | 1.00     |
| Seed-based       | bc_060x_008t.txt             | 60 743    | 7 110  | 100 000 | 60 713  | 100.00    | 1.82      | 503.72     | 49028.59 |
| Seed-based       | bc_075x_020t.txt             | 91 720    | 8 885  | 100 000 | 95 960  | 100.00    | 1.87      | 506.13     | 49232.23 |
| Seed-based       | bd_060x_012t.txt             | 66 395    | 20 186 | 3 623   | 79 186  | 100.00    | 2.24      | 1.50       | 1.00     |
| Seed-based       | bd_090x_008t.txt             | 88 685    | 26 623 | 3 190   | 162 867 | 100.00    | 2.73      | 1.50       | 1.00     |
| Seed-based       | be_060x_020t.txt             | 64 150    | 629    | 620     | 187 171 | 100.00    | 3.41      | 4.93       | 1.27     |
| Seed-based       | be_090x_012t.txt             | 100 000   | 1 014  | 608     | 450 815 | 100.00    | 4.87      | 5.00       | 1.24     |
| Seed-based       | bf_075x_008t.txt             | 77 459    | 770    | 10 139  | 293 849 | 100.00    | 4.19      | 164.59     | 6.93     |
| Seed-based       | bf_090x_020t.txt             | 84 605    | 909    | 29 937  | 412 032 | 100.00    | 5.17      | 164.67     | 6.95     |
| Seed-based       | c_075x_012t.txt              | 91 207    | 8 367  | 100 000 | 89 990  | 300.77    | 1.74      | 506.60     | 48850.90 |
| Seed-based       | cd_060x_008t.txt             | 65 305    | 18 474 | 2 214   | 72 171  | 300.77    | 2.11      | 1.50       | 1.00     |
| Seed-based       | cd_075x_020t.txt             | 85 411    | 22 535 | 6 754   | 113 408 | 300.78    | 2.22      | 1.50       | 1.00     |
| Seed-based       | ce_060x_012t.txt             | 69 662    | 649    | 396     | 192 923 | 300.77    | 3.17      | 5.09       | 1.25     |
| Seed-based       | ce_090x_008t.txt             | 100 000   | 971    | 387     | 427 607 | 300.77    | 4.52      | 4.99       | 1.25     |
| Seed-based       | cf_060x_020t.txt             | 57 420    | 680    | 22 392  | 206 366 | 300.78    | 3.89      | 164.65     | 6.98     |
| Seed-based       | cf_090x_012t.txt             | 99 017    | 899    | 18 013  | 403 963 | 300.78    | 4.34      | 166.98     | 7.05     |
| Seed-based       | d_075x_008t.txt              | 68 867    | 23 422 | 2 818   | 117 900 | 65.00     | 2.67      | 1.50       | 1.00     |
| Seed-based       | d_090x_020t.txt              | 82 871    | 27 543 | 8 253   | 168 401 | 65.00     | 2.94      | 1.50       | 1.00     |
| Seed-based       | de_075x_012t.txt             | 63 095    | 755    | 458     | 280 759 | 65.00     | 4.83      | 5.06       | 1.28     |
| Seed-based       | df_060x_008t.txt             | 48 721    | 729    | 9 732   | 220 756 | 65.00     | 4.89      | 166.88     | 6.90     |
| Seed-based       | df_075x_020t.txt             | 63 536    | 895    | 29 690  | 337 407 | 65.00     | 5.62      | 165.87     | 6.95     |
| Seed-based       | e_060x_012t.txt              | 66 585    | 672    | 414     | 202 940 | 125.49    | 3.35      | 5.14       | 1.25     |
| Seed-based       | e_090x_008t.txt              | 100 000   | 955    | 386     | 424 400 | 125.49    | 4.44      | 5.06       | 1.26     |
| Seed-based       | ef_060x_020t.txt             | 58 606    | 701    | 23 498  | 215 793 | 125.49    | 3.91      | 167.61     | 6.93     |
| Seed-based       | ef_090x_012t.txt             | 74 028    | 1 005  | 20 075  | 461 449 | 125.49    | 6.33      | 166.46     | 6.99     |
| Seed-based       | f_075x_008t.txt              | 61 188    | 841    | 11 102  | 321 719 | 401.12    | 5.38      | 165.02     | 7.02     |
| Seed-based       | f_090x_020t.txt              | 73 488    | 1 064  | 35 524  | 486 927 | 401.12    | 6.71      | 166.94     | 6.95     |
| Sintetike        | synthetic_10.txt             | 84 402    | 8 802  | 482     | 999 983 | 142.12    | 2754.77   | 426.00     | 20498.00 |
| Sintetike        | synthetic_100.txt            | 100 000   | 5 327  | 28 757  | 999 868 | 266.76    | 1599.79   | 336.00     | 35204.00 |
| Sintetike        | synthetic_105.txt            | 27 116    | 32     | 716     | 119 838 | 318.61    | 4.46      | 26.00      | 90.00    |
| Sintetike        | synthetic_110.txt            | 18 843    | 9 111  | 46 751  | 999 983 | 785.74    | 53.07     | 400.00     | 48268.00 |
| Sintetike        | synthetic_115.txt            | 100 000   | 7 054  | 88 048  | 999 875 | 437.44    | 1240.54   | 480.00     | 28867.00 |
| Sintetike        | synthetic_120.txt            | 100 000   | 5 081  | 82 258  | 999 869 | 398.43    | 2096.16   | 188.00     | 21200.00 |
| Sintetike        | synthetic_125.txt            | 100 000   | 6 706  | 54 523  | 999 889 | 344.13    | 1062.58   | 235.00     | 27124.00 |
| Sintetike        | synthetic_130.txt            | 100 000   | 3 352  | 91 540  | 999 871 | 284.66    | 1298.53   | 53.00      | 19313.00 |
| Sintetike        | synthetic_135.txt            | 100 000   | 6 227  | 36 992  | 740 157 | 233.59    | 805.39    | 455.00     | 41302.00 |
| Sintetike        | synthetic_15.txt             | 1 271     | 3      | 130     | 2 433   | 41.34     | 2.03      | 4.00       | 166.00   |
| Sintetike        | synthetic_20.txt             | 38 270    | 745    | 73 522  | 566 103 | 810.93    | 14.79     | 485.00     | 18770.00 |
| Sintetike        | synthetic_25.txt             | 100 000   | 4 832  | 60 494  | 999 845 | 340.77    | 1390.61   | 235.00     | 13181.00 |
| Sintetike        | synthetic_30.txt             | 3 324     | 1      | 1 519   | 2 565   | 890.94    | 1.00      | 151.00     | 10.00    |
| Sintetike        | synthetic_35.txt             | 100 000   | 1 111  | 64 501  | 239 657 | 247.75    | 361.47    | 35.00      | 44471.00 |
| Sintetike        | synthetic_40.txt             | 100 000   | 11 426 | 50 649  | 241 386 | 363.50    | 382.55    | 520.00     | 34595.00 |
| Sintetike        | synthetic_45.txt             | 25 859    | 9 307  | 44 152  | 999 994 | 542.88    | 48.19     | 145.00     | 40128.00 |
| Sintetike        | synthetic_5.txt              | 46 509    | 7 033  | 6 753   | 747 354 | 193.86    | 4982.36   | 52.00      | 26366.00 |
| Sintetike        | synthetic_50.txt             | 96 585    | 22 085 | 88 451  | 288 681 | 294.67    | 596.45    | 294.00     | 14331.00 |
| Sintetike        | synthetic_55.txt             | 100 000   | 7 248  | 56 396  | 766 722 | 272.52    | 3262.65   | 404.00     | 6604.00  |
| Sintetike        | synthetic_60.txt             | 77 949    | 2 437  | 64 294  | 999 991 | 431.39    | 12.83     | 436.00     | 17324.00 |
| Sintetike        | synthetic_65.txt             | 95 490    | 24 914 | 52 437  | 225 156 | 224.33    | 386.20    | 152.00     | 4129.00  |
| Sintetike        | synthetic_70.txt             | 100 000   | 1 544  | 8 584   | 501 031 | 365.80    | 690.13    | 329.00     | 37090.00 |
| Sintetike        | synthetic_75.txt             | 100 000   | 7 578  | 33 769  | 999 995 | 276.46    | 1915.70   | 112.00     | 31409.00 |
| Sintetike        | synthetic_80.txt             | 57 016    | 8 031  | 8 548   | 999 957 | 646.13    | 4566.01   | 360.00     | 6673.00  |
| Sintetike        | synthetic_85.txt             | 2 823     | 1      | 1 671   | 2 823   | 920.18    | 1.00      | 52.00      | 8.00     |
| Sintetike        | synthetic_90.txt             | 78 361    | 7 931  | 58 347  | 175 090 | 386.14    | 383.13    | 429.00     | 34318.00 |
| Sintetike        | synthetic_95.txt             | 99 079    | 10 191 | 100 000 | 171 070 | 348.01    | 328.98    | 395.00     | 38226.00 |

Katër familjet luajnë role të ndryshme në dizajnin e eksperimenteve:

- **Google Hash Code:** pesë instancat zyrtare të raundit kualifikues 2020. Ato përdoren si referencë zyrtare e garës dhe mbahen vetëm në grupin e testimit.
- **Real-world-based:** instanca të gjeneruara me Book Scanning Problem Input Generator [14], me madhësi dhe afate kohore të ndryshme, që synojnë të imitojnë raste praktike të digjitalizimit.

- **Seed-based:** instanca të gjeneruara nga vlerat *seed* zyrtare të Google Hash Code përmes një gjeneratori të kontrolluar [16].
- **Sintetike:** instanca nga repozitori ndihmës Book.Scanning.Dataset [15], të cilat e zgjerojnë hapësirën strukturore të testimit përtej pesë instancave zyrtare.

### 5.3. Vlerat seed dhe riprodhueshmëria

Për konfigurimin final përdoren 10 vlera *seed* për çdo instancë:

$$\{1, 12, 23, 34, 45, 56, 67, 78, 89, 100\}. \quad (5.1)$$

Vlerat *seed* kontrollojnë zgjedhjet stokastike të strategjive konstruktive, kërkimit lokal, perturbimit dhe rifillimit. Në krahasimet finale përdoret vlera *Best* nga 10 ekzekutimet për secilën instancë, pra rezultati më i lartë i arritur, sepse synohet të vlerësohet cilësia më e mirë që arrin konfigurimi final me nisje të shumëfishta. Në të njëjtat tabela raportohet edhe *AVG*, pra mesatarja e 10 ekzekutimeve, si tregues i stabilitetit dhe i performancës së pritur.

### 5.4. Validimi i zgjidhjeve

Çdo zgjidhje dalëse kontrollohet kundrejt skedarit hyrës përkatës. Validimi verifikon formatin, përsëritjen e bibliotekave, përsëritjen e librave, përkatësinë e librave në bibliotekë, respektimin e kapacitetit ditor dhe respektimin e horizontit kohor. Në analizë hyjnë vetëm ekzekutimet me zgjidhje dalëse të vlefshme.

### 5.5. Metodat për krahasim

Krahasimi mbështetet në metoda për krahasim (*baseline*) të bazuara në dy zgjidhje të pavarura me kod të hapur. Burimi i parë është zgjidhja origjinale e Sayed-it, e zhvilluar me kontribut idesh nga Weltsch. Ekipi u rendit i treti nga 4 700 ekipe në fazën shtesë të garës (*extended round*) [5]. Burimi i dytë është zgjidhja origjinale e ekipit Errecto, i renditur në vendin e pestë në raundin kualifikues për problemin Book Scanning [6]. Për eksperimentet tona, të dy implementimet janë zgjeruar dhe përshtatur teknikisht për strukturën hyrëse/dalëse dhe rrjedhën e ekzekutimit të këtij punimi, pa ndryshuar logjikën bazë të kodeve burimore [7, 8]. Kështu ruhet lidhja e qartë ndërmjet zgjidhjeve origjinale dhe varianteve të përdorura në krahasim.

Errecto ruhet edhe si kategori e veçantë në tabelat e rezultateve. Për çdo instancë ndërtohet gjithashtu kolona *Baseline*, e barabartë me rezultatin më të lartë të arritur ndërmjet Errecto, Greedy, MILP1, MILP2, SAT dhe SA. Kjo e bën krahasimin më të rreptë, sepse ILS nuk krahasohet me një metodë të vetme, por me rezultatin më të mirë të prodhuar nga grupi i metodave për krahasim për atë instancë.

Metodat Greedy, MILP1, MILP2, SAT dhe SA janë ekzekutuar si variante për krahasim dhe rezultatet e tyre janë ruajtur në tabelat e analizës. Greedy përdor një ndërtim lakmitar të bibliotekave dhe shërben si bazë e thjeshtë konstruktive. Etiketat MILP1 dhe MILP2 përfaqësojnë dy variante të formulimeve MILP dhe kanë mbulim të ndryshëm të instancave të vlefshme. MILP1 formulon në mënyrë ekzakte problemin e përzgjedhjes së bibliotekave, ndërsa MILP2 është një optimizues më i përgjithshëm me buxhet të madh përsëritjesh, i përshtatshëm për instanca më të mëdha. Të dyja përdorin programin SCIP për zgjidhjen e modeleve të optimizimit [24]. SAT përdor një formulim me variabla booleane, ku kërkohet një caktim që përmbush kufizimet, kurse SA aplikon ftohje të simuluar me rafinim adaptiv të zgjidhjes.

Metodat për krahasim trajtohen si ekzekutime me një vlerë për instancë, sepse janë deterministike për një vlerë të dhënë *seed*. Në veçanti, SA përdor rastësi të kontrolluar gjatë kërkimit, por, meqë ekzekutohet me një vlerë fikse *seed*, prodhon të njëjtin rezultat në çdo ekzekutim të përsëritur, prandaj një ekzekutim i vetëm mjafton. Metoda e propozuar ILS, në anën tjetër, vlerësohet me 10 vlera të ndryshme *seed* për të matur qëndrueshmërinë e saj ndaj rastësisë. Prandaj ekzekutohet 10 herë për instancë dhe në tabela raporton si *Best*, ashtu edhe *AVG*.

## 5.6. Parametrat për krahasim

Parametri kryesor i krahasimit është rezultati final, pra shuma e pikëve të librave unikë të skanuar brenda afatit. Krahas tij, në analizë përdoren edhe:

- rezultati i zgjidhjes fillestare, që mat cilësinë para fazës ILS.
- rezultati final *Best* dhe *AVG* nga 10 ekzekutimet.
- diferenca relative ndaj metodave për krahasim.
- koha reale e ekzekutimit.
- koeficienti i variacionit ndërmjet vlerave *seed*.

Në të gjitha tabelat e rezultateve, përfshirë krahasimet me metodat e tjera dhe eksperimentet me komponentët dhe grupet e operatorëve, ILS mbahet si bazë e njëjtë referuese. Kolona *Gap ndaj ILS* llogaritet si:

$$\text{Gap}_{\text{ILS}}(B) = 100 \cdot \frac{s_B(i) - s_{\text{ILS}}(i)}{s_B(i)}, \quad (5.2)$$

ku  $s_{\text{ILS}}(i)$  është rezultati i ILS në instancën  $i$ , ndërsa  $s_B(i)$  është rezultati i metodës për krahasim ose i variantit eksperimental dhe shërben si emërues. Me këtë konventë, vlera negative tregon qartë epërsi të ILS, zeroja tregon barazim, ndërsa vlera pozitive tregon se metoda për krahasim ose varianti eksperimental arrin rezultat më të lartë se ILS.

## 5.7. Analiza statistikore

Për krahasimet në çifte përdoret testi Wilcoxon signed-rank [25]. Çdo instancë formon një çift rezultatesh ndërmjet ILS *Best* dhe metodës për krahasim përkatëse. P-vlerat korrigjohen me metodën Holm-Bonferroni [26]. Në interpretim, testet statistikore lexohen së bashku me numrin e fitoreve, humbjeve dhe barazimeve, si dhe me madhësinë praktike të diferencës.

## 5.8. Analiza e komponentëve të algoritmit

Për të kuptuar rolin e secilës pjesë të algoritmit, në instancat e testimit shqyrtohen disa konfigurime: vetëm kërkim lokal, pa perturbim, pa rifillim, pa pranim probabilistik, pa grupe të caktuara operatorësh dhe nisje me renditje të thjeshtë. Secili konfigurim ekzekutohet 10 herë me të njëjtat vlera seed, njësoj si konfigurimi i plotë ILS. Këto eksperimente nuk krahasohen me metodat për krahasim, por me konfigurimin e plotë ILS.

## 5.9. Analiza e konvergjencës së algoritmit

Për analizën e konvergjencës janë zgjedhur 10 instanca të ndryshme dhe secila është ekzekutuar 10 herë. Gjurmët regjistrojnë kohën, fazën e kërkimit, rezultatin aktual, rezultatin më të mirë dhe ngjarjet kryesore të kërkimit. Kjo analizë nuk synon renditjen e metodave për krahasim. Ajo shpjegon dinamikën e brendshme të ILS.

## 6. Rezultatet eksperimentale dhe diskutimi

Ky kapitull fillon me analizën e sjelljes së metodës ILS dhe të komponentëve të saj. Fillimisht shqyrtohet sa përmirëson kërkimi lokal mbi zgjidhjen fillestare, pastaj analizohet ndikimi i komponentëve, grupeve të operatorëve, cilësisë së zgjidhjes fillestare, konvergjencës dhe vlerave *seed*. Më pas paraqitet krahasimi me baseline dhe me metodat për krahasim, ndërsa pjesa e fundit jep interpretimin përmbljedhës të rezultateve.

### 6.1. Përmirësimi nga zgjidhja fillestare

Figura 6.1 tregon përmirësimin e ILS për secilën instancë testuese, nga zgjidhja fillestare e ndërtuar para ILS deri te rezultati final. Çdo pikë paraqet një instancë testuese, ndërsa vija horizontale tregon medianën brenda familjes së instancave. Tabela 6.1 liston instancat ku ILS jep përmirësimin më të madh relativ nga zgjidhja fillestare.

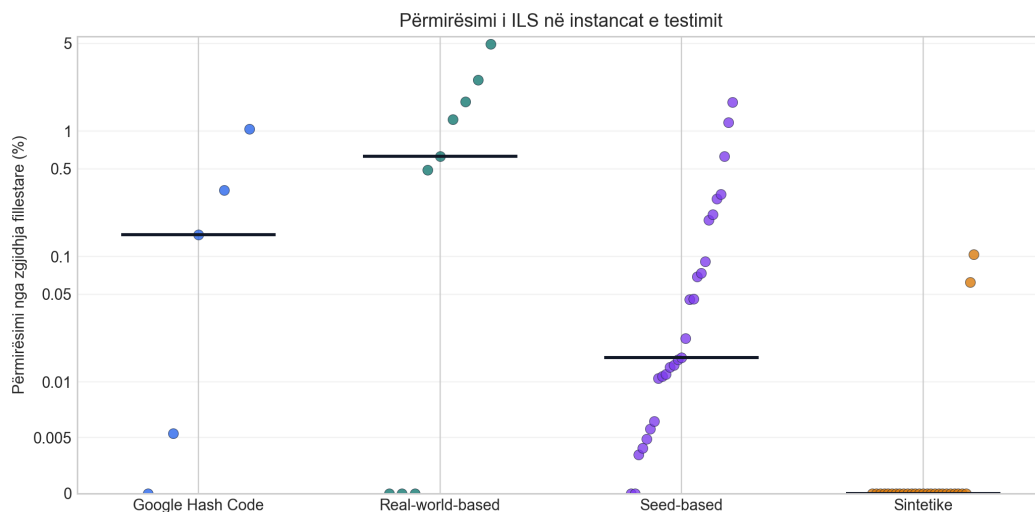


Figura 6.1: Përmirësimi i ILS nga zgjidhja fillestare te zgjidhja finale në instancat e testimit.

Tabela 6.1: Instancat me përmirësimin më të madh relativ të ILS nga zgjidhja fillestare te rezultati final.

| Familja          | Instanca             | Zgjidhja fillestare | ILS final  | Përmirësim (%) | ILS AVG final |
|------------------|----------------------|---------------------|------------|----------------|---------------|
| Real-world-based | B90k_L850_D21.txt    | 828                 | 869        | 4.952          | 869           |
| Real-world-based | B35k_L60_D120.txt    | 12 544 648          | 12 864 753 | 2.552          | 12 864 753    |
| Real-world-based | B240k_L90_D180.txt   | 42 359 386          | 43 085 291 | 1.714          | 43 085 291    |
| Seed-based       | ce_090x_008t.txt     | 20 368 337          | 20 713 720 | 1.696          | 20 692 587    |
| Real-world-based | B1000k_L115_D230.txt | 56 328 452          | 57 027 193 | 1.240          | 57 027 193    |
| Seed-based       | e_090x_008t.txt      | 8 809 568           | 8 912 388  | 1.167          | 8 910 345     |
| Google Hash Code | d_tough_choices.txt  | 5 041 660           | 5 094 050  | 1.039          | 5 092 841     |
| Seed-based       | e_060x_012t.txt      | 5 613 967           | 5 649 393  | 0.631          | 5 631 492     |
| Real-world-based | B90_L6_D11.txt       | 2 551               | 2 567      | 0.627          | 2 567         |
| Real-world-based | B2000k_L140_D280.txt | 60 713 936          | 61 011 720 | 0.490          | 60 966 934    |
| Google Hash Code | e_so_many_books.txt  | 5 222 608           | 5 240 181  | 0.336          | 5 240 005     |
| Seed-based       | ce_060x_012t.txt     | 12 973 556          | 13 014 383 | 0.315          | 13 013 811    |

Në Tabelën 6.1, kolona *Zgjidhja fillestare* tregon rezultatin e arritur nga faza konstruktive para fillimit të ILS, ndërsa kolona *ILS final* tregon rezultatin më të mirë të arritur pas kërkimit lokal të përsëritur.

## 6.2. Analiza e komponentëve të algoritmit

Tabela 6.2 përmbledh analizën e komponentëve të algoritmit në instancat e testimit. Konfigurimi i plotë krahasohet me variantet ku hiqet ose thjeshtohet një komponent i algoritmit. Krahasimi përdor vlerën *Best* nga 10 ekzekutime, ndërsa *AVG* raportohet për transparencë. Në kolonën *Gap ndaj ILS*, vlerat negative tregojnë epërsi të ILS ndaj variantit, ndërsa vlerat pozitive tregojnë se varianti e tejkalon ILS. Varianti *Vetëm kërkim lokal* aplikon një përmirësim lokal të vetëm mbi zgjidhjen fillestare dhe nuk hyn në ciklin e plotë ILS.

Tabela 6.2: Eksperimentimi me komponentët e algoritmit në instancat e testimit.

| Konfigurimi             | Best total         | AVG total          | Gap ndaj ILS (%) |
|-------------------------|--------------------|--------------------|------------------|
| Metoda e propozuar ILS  | <b>477 429 716</b> | <b>477 309 618</b> | 0.000            |
| Vetëm kërkim lokal      | 476 181 564        | 475 600 821        | -0.262           |
| Pa pranim probabilistik | 477 415 465        | 477 319 444        | -0.003           |
| Pa perturbim            | 477 465 158        | 477 330 571        | 0.007            |
| Pa rifillim             | 477 359 379        | 477 303 271        | -0.015           |

## 6. Rezultatet eksperimentale dhe diskutimi

Tabela 6.3: Testet Wilcoxon për krahasimet në çifte të varianteve të komponentëve kundrejt konfigurimit të plotë.

| Konfigurimi               | Fitore ILS | Humbje ILS | Barazime | $P_{\text{Wilcoxon}}$<br>(Holm) |
|---------------------------|------------|------------|----------|---------------------------------|
| Vetëm kërkim lokal        | 37         | 0          | 31       | 3.421e-07                       |
| Pa pranim probabilitistik | 15         | 16         | 37       | 1.000e+00                       |
| Pa perturbim              | 12         | 20         | 36       | 1.000e+00                       |
| Pa rifillim               | 18         | 13         | 37       | 6.164e-01                       |

Tabela 6.4 jep përparësinë relative të ILS ndaj secilit variant për çdo instancë. Prej saj duket se dallimet e vogla në total nuk shpërndahen njësoj në të gjitha instancat.

Tabela 6.4: Gap-i relativ i varianteve të komponentëve ndaj ILS për secilën instancë.

| Instanca                     | Vetëm kërkim lokal | Pa pranim probabilitistik | Pa perturbim | Pa rifillim |
|------------------------------|--------------------|---------------------------|--------------|-------------|
| b_read_on.txt                | 0.000              | 0.000                     | 0.000        | 0.000       |
| c_incunabula.txt             | -0.002             | 0.000                     | 0.000        | -0.001      |
| d_tough_choices.txt          | -0.476             | -0.008                    | 0.004        | 0.003       |
| e_so_many_books.txt          | -0.199             | 0.000                     | 0.002        | -0.002      |
| f_libraries_of_the_world.txt | 0.000              | 0.000                     | 0.000        | 0.000       |
| B1000k_L115_D230.txt         | -1.240             | 0.000                     | 0.000        | 0.000       |
| B100_L9_D18.txt              | 0.000              | 0.000                     | 0.000        | 0.000       |
| B1k_L18_D17.txt              | 0.000              | 0.000                     | 0.000        | 0.000       |
| B2000k_L140_D280.txt         | -0.082             | 0.000                     | 0.000        | -0.082      |
| B240k_L90_D180.txt           | 0.000              | 0.000                     | 0.000        | 0.000       |
| B35k_L60_D120.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| B6k_L35_D70.txt              | 0.000              | 0.000                     | 0.000        | 0.000       |
| B90_L6_D11.txt               | 0.000              | 0.000                     | 0.000        | 0.000       |
| B90k_L850_D21.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| b_075x_012t.txt              | -0.001             | 0.000                     | 0.000        | 0.000       |
| bc_060x_008t.txt             | -0.010             | 0.000                     | 0.001        | -0.004      |
| bc_075x_020t.txt             | -0.012             | 0.000                     | 0.001        | 0.000       |
| bd_060x_012t.txt             | -0.092             | -0.022                    | 0.038        | -0.029      |
| bd_090x_008t.txt             | -0.013             | 0.007                     | -0.012       | 0.014       |
| be_060x_020t.txt             | -0.169             | -0.017                    | -0.006       | -0.006      |
| be_090x_012t.txt             | -0.124             | 0.055                     | 0.049        | 0.050       |
| bf_075x_008t.txt             | -0.011             | 0.002                     | 0.019        | 0.007       |
| bf_090x_020t.txt             | 0.000              | 0.000                     | 0.000        | 0.000       |
| c_075x_012t.txt              | -0.006             | 0.001                     | -0.004       | -0.006      |
| cd_060x_008t.txt             | -0.045             | -0.005                    | 0.040        | 0.004       |
| cd_075x_020t.txt             | -0.069             | -0.019                    | 0.082        | 0.012       |
| ce_060x_012t.txt             | -0.083             | 0.088                     | 0.152        | 0.001       |
| ce_090x_008t.txt             | -1.375             | -0.077                    | 0.014        | -0.037      |
| cf_060x_020t.txt             | -0.014             | 0.000                     | -0.007       | -0.002      |
| cf_090x_012t.txt             | -0.003             | 0.000                     | -0.002       | -0.002      |
| d_075x_008t.txt              | -0.073             | -0.010                    | -0.003       | -0.032      |
| d_090x_020t.txt              | -0.046             | -0.021                    | -0.027       | 0.061       |
| de_075x_012t.txt             | -0.154             | 0.004                     | -0.013       | 0.052       |
| df_060x_008t.txt             | -0.006             | 0.000                     | -0.006       | -0.006      |
| df_075x_020t.txt             | -0.004             | 0.002                     | 0.002        | -0.001      |
| e_060x_012t.txt              | -0.385             | 0.074                     | -0.023       | -0.351      |
| e_090x_008t.txt              | -0.900             | 0.022                     | 0.021        | 0.026       |
| ef_060x_020t.txt             | -0.001             | 0.000                     | 0.000        | -0.001      |
| ef_090x_012t.txt             | 0.000              | 0.000                     | 0.001        | 0.000       |
| f_075x_008t.txt              | 0.000              | 0.000                     | 0.001        | 0.005       |
| f_090x_020t.txt              | -0.022             | -0.003                    | -0.002       | 0.006       |
| synthetic_10.txt             | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_100.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_105.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_110.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_115.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_120.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_125.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_130.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_135.txt            | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_15.txt             | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_20.txt             | -0.054             | -0.019                    | -0.011       | -0.003      |
| synthetic_25.txt             | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_30.txt             | 0.000              | 0.000                     | 0.000        | 0.000       |
| synthetic_35.txt             | 0.000              | 0.000                     | 0.000        | 0.000       |

Vazhdon në faqen tjetër

Tabela 6.4: Vazhdim i tabelës

| Instanca         | Vetëm kërkim lokal | Pa pranim probabilistik | Pa perturbim | Pa rifillim |
|------------------|--------------------|-------------------------|--------------|-------------|
| synthetic_40.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_45.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_5.txt  | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_50.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_55.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_60.txt | -0.063             | -0.053                  | 0.015        | -0.021      |
| synthetic_65.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_70.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_75.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_80.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_85.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_90.txt | 0.000              | 0.000                   | 0.000        | 0.000       |
| synthetic_95.txt | 0.000              | 0.000                   | 0.000        | 0.000       |

Gjetja kryesore është se kërkimi lokal mbetet shumë i fortë edhe pa ciklin e plotë ILS, ndërsa dallimet ndërmjet konfigurimit të plotë dhe disa varianteve janë shumë të vogla në total. Megjithatë, konfigurimi i plotë e tejkalon variantin me vetëm kërkim lokal në 37 instance dhe nuk humb në asnjë. Vlera e korrigjuar  $p = 3.421 \times 10^{-7}$  në Tabelën 6.3 tregon se ky dallim është statistikisht i rëndësishëm. Varianti pa perturbim arrin vlerë pak më të lartë në *Best* total, ndërsa variantet pa perturbim dhe pa pranim probabilistik janë pak më të larta në *AVG* total. Për variantet pa pranim probabilistik, pa perturbim dhe pa rifillim, vlerat  $p > 0.05$  nuk tregojnë dallime statistikisht të rëndësishme ndaj konfigurimit të plotë pas korrigjimit Holm. Për këtë arsye, ato interpretohen si variante praktikisht shumë të afërta me konfigurimin e plotë, jo si përmirësime të qëndrueshme të metodës. Ky rezultat nuk e zvogëlon rëndësinë e ILS. Përkundrazi, ai tregon se kërkimi lokal përbën komponentin kryesor të metodës, ndërsa mekanizmat tipikë të ILS, si perturbimi, rifillimi dhe pranimi, veprojnë kryesisht si mekanizma diversifikimi, stabilizimi dhe rikuperimi në nivel instance.

### 6.3. Analiza e grupeve të operatorëve

Tabela 6.5 tregon ndikimin e heqjes së grupeve të operatorëve. Krahasimi përdor vlerën *Best* nga 10 ekzekutime, ndërsa *AVG* raportohet për transparencë. Edhe këtu përdoret e njëjta konventë si te analiza e komponentëve: vlerat negative në kolonën *Gap ndaj ILS* tregojnë epërsi të ILS ndaj variantit, ndërsa vlerat pozitive tregojnë se varianti e tejkalon ILS.

Tabela 6.5: Eksperimenti me grupin e operatorëve në instancat e testimit.

| Konfigurimi                          | Best total         | AVG total          | Gap ndaj ILS (%) |
|--------------------------------------|--------------------|--------------------|------------------|
| Metoda e propozuar ILS               | <b>477 429 716</b> | <b>477 309 618</b> | 0.000            |
| Pa futje dhe zëvendësim bibliotekash | 477 293 236        | 477 108 533        | -0.029           |
| Pa rirenditje të regjistrimit        | 477 334 136        | 477 267 413        | -0.020           |
| Pa lëvizje strategjike strukturore   | 477 413 931        | 477 221 144        | -0.003           |

Tabela 6.6 paraqet testet Wilcoxon për krahasime në çifte. Rezultatet tregojnë se dallimi mbetet statistikisht i rëndësishëm vetëm kur hiqen operatorët e futjes dhe të zëvendësimit, ndërsa



## 6. Rezultatet eksperimentale dhe diskutimi

dy grupet e tjera nuk japin dallim të qëndrueshëm pas korrigjimit Holm.

Tabela 6.6: Testet Wilcoxon për krahasimet në çifte të varianteve të grupeve të operatorëve kundrejt konfigurimit të plotë.

| Konfigurimi                          | Fitore ILS | Humbje ILS | Barazime | $p_{\text{Wilcoxon}}$<br>(Holm) |
|--------------------------------------|------------|------------|----------|---------------------------------|
| Pa futje dhe zëvendësim bibliotekash | 28         | 2          | 38       | 3.530e-06                       |
| Pa rirenditje të regjistrimit        | 15         | 16         | 37       | 1.000e+00                       |
| Pa lëvizje strategjike strukturore   | 15         | 16         | 37       | 1.000e+00                       |

Tabela 6.7 jep përparësinë relative të ILS ndaj secilit variant të grupeve të operatorëve për çdo instancë.

Tabela 6.7: Gap-i relativ i varianteve të grupeve të operatorëve ndaj ILS për secilën instancë.

| Instanca                     | Pa futje dhe zëvendësim bibliotekash | Pa rirenditje të regjistrimit | Pa lëvizje strategjike strukturore |
|------------------------------|--------------------------------------|-------------------------------|------------------------------------|
| b_read_on.txt                | 0.000                                | 0.000                         | 0.000                              |
| c_incunabula.txt             | 0.000                                | 0.002                         | -0.001                             |
| d_tough_choices.txt          | -0.005                               | -0.004                        | -1.035                             |
| e_so_many_books.txt          | -0.001                               | -0.006                        | -0.001                             |
| f_libraries_of_the_world.txt | 0.000                                | 0.000                         | 0.000                              |
| B1000k_L115_D230.txt         | 0.000                                | 0.000                         | 0.000                              |
| B100_L9_D18.txt              | 0.000                                | 0.000                         | 0.000                              |
| B1k_L18_D17.txt              | 0.000                                | 0.000                         | 0.000                              |
| B2000k_L140_D280.txt         | 0.000                                | -0.082                        | 0.000                              |
| B240k_L90_D180.txt           | 0.000                                | 0.000                         | 0.000                              |
| B35k_L60_D120.txt            | 0.000                                | 0.000                         | 0.000                              |
| B6k_L35_D70.txt              | 0.000                                | 0.000                         | 0.000                              |
| B90_L6_D11.txt               | 0.000                                | 0.000                         | 0.000                              |
| B90k_L850_D21.txt            | 0.000                                | 0.000                         | 0.000                              |
| b_075x_012t.txt              | 0.000                                | 0.000                         | 0.000                              |
| bc_060x_008t.txt             | 0.000                                | 0.000                         | 0.000                              |
| bc_075x_020t.txt             | -0.004                               | 0.000                         | 0.000                              |
| bd_060x_012t.txt             | -0.080                               | -0.031                        | 0.018                              |
| bd_090x_008t.txt             | -0.011                               | -0.006                        | 0.027                              |
| be_060x_020t.txt             | -0.089                               | -0.051                        | 0.016                              |
| be_090x_012t.txt             | -0.002                               | 0.051                         | 0.060                              |
| bf_075x_008t.txt             | -0.011                               | 0.007                         | 0.001                              |
| bf_090x_020t.txt             | 0.000                                | 0.000                         | 0.000                              |
| c_075x_012t.txt              | -0.001                               | -0.003                        | -0.005                             |
| cd_060x_008t.txt             | -0.022                               | -0.007                        | 0.032                              |
| cd_075x_020t.txt             | -0.027                               | 0.008                         | 0.103                              |
| ce_060x_012t.txt             | 0.001                                | -0.006                        | 0.001                              |
| ce_090x_008t.txt             | -0.297                               | -0.134                        | 0.047                              |
| cf_060x_020t.txt             | -0.008                               | 0.003                         | 0.000                              |
| cf_090x_012t.txt             | -0.002                               | 0.000                         | -0.002                             |
| d_075x_008t.txt              | -0.061                               | -0.028                        | 0.014                              |
| d_090x_020t.txt              | -0.039                               | 0.006                         | 0.118                              |
| de_075x_012t.txt             | -0.102                               | -0.017                        | 0.031                              |
| df_060x_008t.txt             | 0.000                                | 0.000                         | 0.000                              |
| df_075x_020t.txt             | -0.004                               | 0.002                         | -0.001                             |
| e_060x_012t.txt              | -0.350                               | -0.235                        | 0.150                              |
| e_090x_008t.txt              | -0.185                               | -0.106                        | 0.055                              |
| ef_060x_020t.txt             | 0.000                                | 0.000                         | -0.001                             |
| ef_090x_012t.txt             | 0.000                                | 0.000                         | 0.000                              |
| f_075x_008t.txt              | 0.001                                | 0.005                         | 0.000                              |
| f_090x_020t.txt              | -0.007                               | 0.001                         | -0.007                             |
| synthetic_10.txt             | 0.000                                | 0.000                         | 0.000                              |
| synthetic_100.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_105.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_110.txt            | 0.000                                | 0.052                         | 0.007                              |
| synthetic_115.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_120.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_125.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_130.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_135.txt            | 0.000                                | 0.000                         | 0.000                              |
| synthetic_15.txt             | 0.000                                | 0.000                         | 0.000                              |
| synthetic_20.txt             | -0.029                               | 0.003                         | -0.015                             |

Vazhdon në faqen tjetër

## 6. Rezultatet eksperimentale dhe diskutimi

Tabela 6.7: Vazhdim i tabelës

| Instanca         | Pa futje dhe zëvendësim bibliotekash | Pa rirenditje të regjistrimit | Pa lëvizje strategjike strukturore |
|------------------|--------------------------------------|-------------------------------|------------------------------------|
| synthetic_25.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_30.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_35.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_40.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_45.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_5.txt  | 0.000                                | 0.000                         | 0.000                              |
| synthetic_50.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_55.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_60.txt | -0.063                               | 0.003                         | -0.014                             |
| synthetic_65.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_70.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_75.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_80.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_85.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_90.txt | 0.000                                | 0.000                         | 0.000                              |
| synthetic_95.txt | 0.000                                | 0.000                         | 0.000                              |

Heqja e operatorëve të futjes dhe të zëvendësimit dëmton më shumë sesa heqja e operatorëve të rirenditjes. Kjo sugjeron se ndryshimi i përbërjes së pjesës aktive të renditjes ka më shumë peshë sesa korrigjimi lokal i rendit relativ të bibliotekave ekzistuese.

Për ta plotësuar këtë analizë, Figura 6.2 përmbledh sa herë është pranuar secili operator gjatë analizës së konvergencës. Emrat në boshtin vertikal ruhen në formën e implementimit, njësoj si në Tabelën 4.2, për ta bërë lidhjen me log-et dhe statistikën e operatorëve të drejtpërdrejtë. Ngjyra tregon grupin metodologjik të operatorit. Grafiku tregon vetëm sa shpesh është përdorur secili operator, jo sa ka kontribuar në rezultatin final. Për këtë arsye, ai lexohet së bashku me Tabelën 6.5.

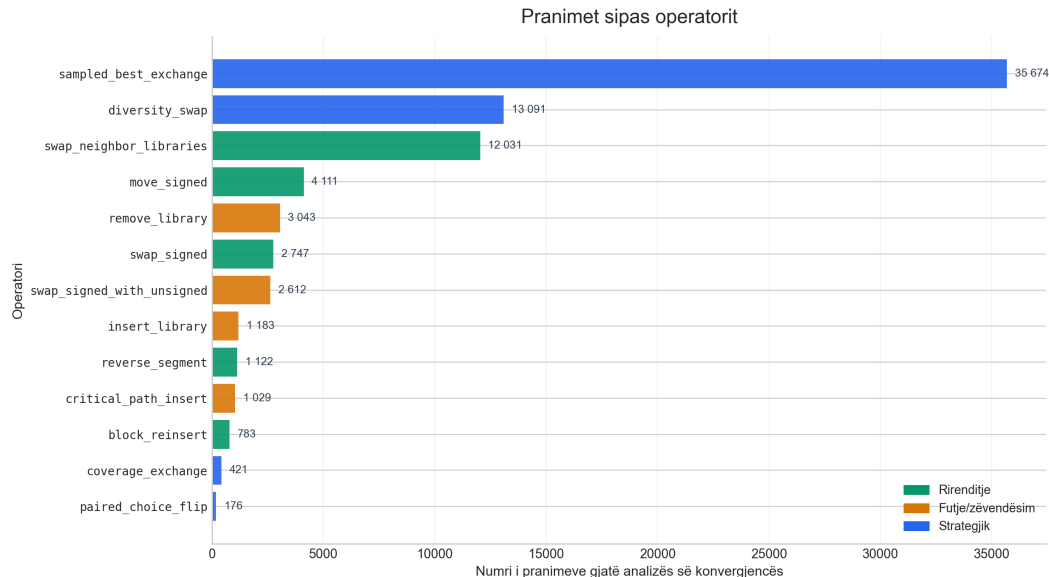


Figura 6.2: Numri i pranimeve sipas operatorit gjatë analizës së konvergencës.

### 6.4. Ndikimi i zgjidhjes fillestare

Tabela 6.8 mat ndikimin e nisjes nga një renditje e thjeshtë. Ky eksperiment kontrollon nëse ILS arrin ta rikuperojë cilësinë edhe kur zgjidhja fillestare nuk është aq e fortë sa në konfigurimin

## 6. Rezultatet eksperimentale dhe diskutimi

final. Krahasimi përdor vlerën *Best* nga 10 ekzekutime, ndërsa *AVG* raportohet për transparencë. Kolonat *Gap Best ndaj ILS* dhe *Gap AVG ndaj ILS* raportojnë dallimin relativ ndaj konfigurimit të plotë për këto dy metrika. Vlerat negative tregojnë epërsi të konfigurimit të plotë ILS ndaj nisjes së thjeshtë, ndërsa vlerat pozitive tregojnë rastet ku nisja e thjeshtë arrin rezultat më të lartë.

Tabela 6.8: Krahasimi i konfigurimit të plotë me nisjen e thjeshtë në instancat e testimit.

| Konfigurimi            | Best total         | AVG total          | Gap Best ndaj ILS (%) | Gap AVG ndaj ILS (%) |
|------------------------|--------------------|--------------------|-----------------------|----------------------|
| Metoda e propozuar ILS | <b>477 429 716</b> | <b>477 309 618</b> | 0.000                 | 0.000                |
| Nisje e thjeshtë       | 477 228 453        | 476 696 335        | -0.042                | -0.129               |

Në terma mesatarë, nisja me renditje të thjeshtë rritet nga 421,718,944 pikë fillestare në 476,696,335 pikë finale, që përfaqëson përmirësim prej 13.04% gjatë kërimit.

Tabela 6.9 paraqet testin Wilcoxon për krahasimin në çift. Rezultati nuk kalon prapun e rëndësisë statistikore pas korrigjimit Holm, por tregon prirjen që nisja e thjeshtë të jetë më e dobët në mesatare.

Tabela 6.9: Testi Wilcoxon për krahasimin në çift të nisjes së thjeshtë kundrejt konfigurimit të plotë.

| Konfigurimi      | Fitore ILS | Humbje ILS | Barazime | $p_{\text{Wilcoxon}}$ (Holm) |
|------------------|------------|------------|----------|------------------------------|
| Nisje e thjeshtë | 20         | 12         | 36       | 6.688e-02                    |

Tabela 6.10 jep *Gap-in ndaj ILS* për nisjen e thjeshtë në secilën instancë. Vlerat negative tregojnë epërsi të konfigurimit të plotë ILS.

Tabela 6.10: Gap-i relativ i nisjes së thjeshtë ndaj ILS për secilën instancë.

| Instanca                     | Gap ndaj ILS (%) |
|------------------------------|------------------|
| b_read_on.txt                | 0.000            |
| c_incunabula.txt             | 0.000            |
| d_tough_choices.txt          | 0.003            |
| e_so_many_books.txt          | -0.003           |
| f_libraries_of_the_world.txt | 0.000            |
| B1000k_L115_D230.txt         | 0.000            |
| B100_L9_D18.txt              | 0.000            |
| B1k_L18_D17.txt              | 0.000            |
| B2000k_L140_D280.txt         | 0.000            |
| B240k_L90_D180.txt           | 0.000            |
| B35k_L60_D120.txt            | 0.000            |
| B6k_L35_D70.txt              | 0.000            |
| B90_L6_D11.txt               | 0.000            |
| B90k_L850_D21.txt            | 0.000            |
| b_075x_012t.txt              | 0.000            |
| bc_060x_008t.txt             | -0.001           |
| bc_075x_020t.txt             | 0.000            |

Vazhdon në faqen tjetër

## 6. Rezultatet eksperimentale dhe diskutimi

Tabela 6.10: Vazhdim i tabelës

| Instanca          | Gap ndaj ILS (%) |
|-------------------|------------------|
| bd_060x_012t.txt  | -1.289           |
| bd_090x_008t.txt  | -2.333           |
| be_060x_020t.txt  | 0.003            |
| be_090x_012t.txt  | 0.100            |
| bf_075x_008t.txt  | -0.006           |
| bf_090x_020t.txt  | 0.000            |
| c_075x_012t.txt   | -0.005           |
| cd_060x_008t.txt  | -1.558           |
| cd_075x_020t.txt  | -0.042           |
| ce_060x_012t.txt  | 0.355            |
| ce_090x_008t.txt  | 0.029            |
| cf_060x_020t.txt  | 0.000            |
| cf_090x_012t.txt  | 0.000            |
| d_075x_008t.txt   | -1.639           |
| d_090x_020t.txt   | -2.182           |
| de_075x_012t.txt  | 0.146            |
| df_060x_008t.txt  | 0.006            |
| df_075x_020t.txt  | 0.000            |
| e_060x_012t.txt   | 0.194            |
| e_090x_008t.txt   | 0.075            |
| ef_060x_020t.txt  | 0.000            |
| ef_090x_012t.txt  | 0.000            |
| f_075x_008t.txt   | 0.000            |
| f_090x_020t.txt   | -0.003           |
| synthetic_10.txt  | 0.000            |
| synthetic_100.txt | 0.000            |
| synthetic_105.txt | 0.000            |
| synthetic_110.txt | -0.058           |
| synthetic_115.txt | 0.000            |
| synthetic_120.txt | 0.000            |
| synthetic_125.txt | 0.000            |
| synthetic_130.txt | 0.000            |
| synthetic_135.txt | 0.000            |
| synthetic_15.txt  | 0.000            |
| synthetic_20.txt  | -0.101           |
| synthetic_25.txt  | 0.000            |
| synthetic_30.txt  | 0.000            |
| synthetic_35.txt  | 0.000            |
| synthetic_40.txt  | 0.000            |
| synthetic_45.txt  | 0.005            |
| synthetic_5.txt   | 0.000            |
| synthetic_50.txt  | 0.000            |
| synthetic_55.txt  | 0.000            |
| synthetic_60.txt  | -0.163           |
| synthetic_65.txt  | 0.000            |
| synthetic_70.txt  | 0.000            |
| synthetic_75.txt  | 0.000            |
| synthetic_80.txt  | 0.000            |
| synthetic_85.txt  | 0.000            |
| synthetic_90.txt  | 0.000            |
| synthetic_95.txt  | 0.000            |

Rezultatet tregojnë se ILS ka aftësi të mirë korrigjuese, pasi diferenca ndaj konfigurimit të plotë mbetet e vogël në *Best* total, ndërsa *AVG* bie më shumë, duke treguar rezultate më pak të qëndrueshme nga seed në seed.

## 6.5. Analiza e konvergjencës së algoritmit

Analiza e konvergjencës mbështetet në 10 instanca të zgjedhura, secila e ekzekutuar 10 herë. Tabela 6.11 përmbledh përmirësimin mesatar, kohën mesatare për arritjen e 95% të përmirësimit final, si dhe numrin mesatar të përmirësimeve të rekordit dhe rifillimeve. Në tabelë, *Përmir. AVG* tregon sa herë mesatarisht gjatë një ekzekutimi është gjetur një rezultat i ri më i mirë se rezultati më i mirë i deriatëhershëm. Pragu prej 95% përdoret si tregues praktik i afërsisë me rezultatin final. Ai mat momentin kur ekzekutimi ka arritur pothuajse gjithë përmirësimin që do të arrijë deri në fund, pa u ndikuar shumë nga përmirësimet e vogla të vonshme.

Tabela 6.11: Përmbledhje e analizës së konvergjencës së algoritmit për 10 instancat e zgjedhura.

| Familja          | Inst. | Përmirësimi<br>AVG (%) | $t_{95}$ AVG<br>(s) | Përmir.<br>AVG | Rifillime<br>AVG |
|------------------|-------|------------------------|---------------------|----------------|------------------|
| Google Hash Code | 3     | 0.509                  | 111.8               | 50.0           | 13.2             |
| Real-world-based | 3     | 2.640                  | 8.0                 | 1.3            | 81.8             |
| Seed-based       | 2     | 0.922                  | 233.0               | 236.4          | 0.8              |
| Sintetike        | 2     | 0.059                  | 257.1               | 11.8           | 1.4              |
| Të gjitha        | 10    | 1.141                  | 134.0               | 65.0           | 28.9             |

Tabela 6.12: Konvergjenca e ILS në 10 instancat e zgjedhura.

| Instanca                     | ILS AVG    | Përmirësimi<br>AVG (%) | $t_{95}$ AVG<br>(s) | CV (%) | Rifillime<br>AVG |
|------------------------------|------------|------------------------|---------------------|--------|------------------|
| d_tough_choices.txt          | 5 092 633  | 1.046                  | 260.1               | 0.0159 | 3.8              |
| e_so_many_books.txt          | 5 239 942  | 0.332                  | 71.1                | 0.0016 | 1.1              |
| f_libraries_of_the_world.txt | 5 348 248  | 0.149                  | 4.1                 | 0.0000 | 34.8             |
| B2000k_L140_D280.txt         | 60 966 934 | 0.417                  | 14.3                | 0.0258 | 24.7             |
| B35k_L60_D120.txt            | 12 864 753 | 2.552                  | 9.5                 | 0.0000 | 39.9             |
| B90k_L850_D21.txt            | 869        | 4.952                  | 0.3                 | 0.0000 | 180.7            |
| ce_060x_012t.txt             | 13 013 931 | 0.311                  | 85.8                | 0.0041 | 1.5              |
| ce_090x_008t.txt             | 20 680 564 | 1.533                  | 380.2               | 0.0529 | 0.0              |
| synthetic_20.txt             | 29 691 848 | 0.086                  | 258.0               | 0.0134 | 1.9              |
| synthetic_60.txt             | 18 670 432 | 0.032                  | 256.2               | 0.0202 | 0.9              |

Figura 6.3 tregon sa pjesë e përmirësimit final arrihet në pika të ndryshme kohore gjatë fazës ILS. Disa familje instancash e arrijnë përmirësimin pothuajse menjëherë, ndërsa te familjet seed-based dhe sintetike ka raste ku progresi vazhdon deri afër fundit të buxhetit kohor.

## 6. Rezultatet eksperimentale dhe diskutimi

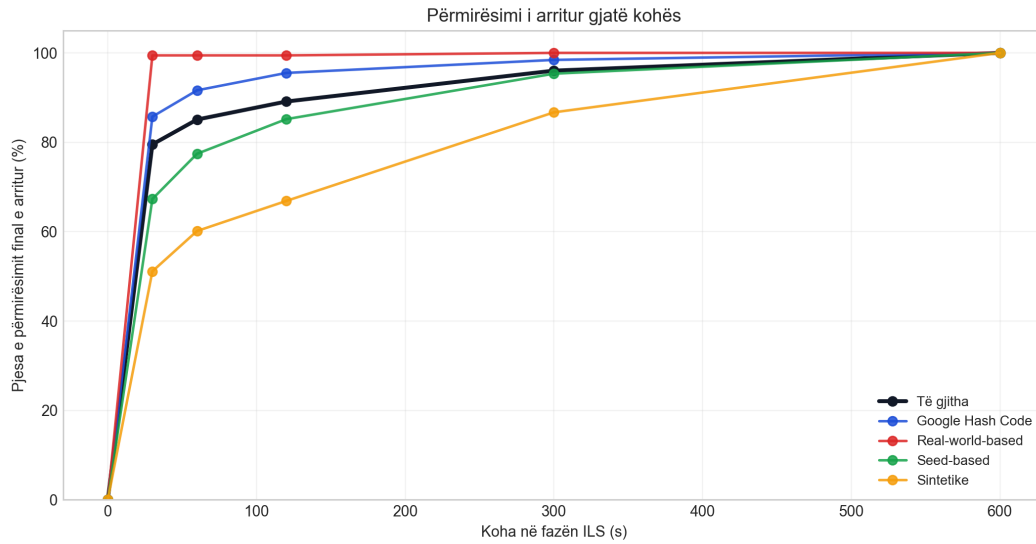


Figura 6.3: Pjesa e përmirësimit final e arritur gjatë fazës ILS, sipas familjes së instancave.

Figura 6.4 e shfaq të njëjtën dinamikë nga këndvështrimi i rezultatit absolut. Për çdo pikë kontrolli raportohet sa pjesë e rezultatit final të atij ekzekutimi është arritur.

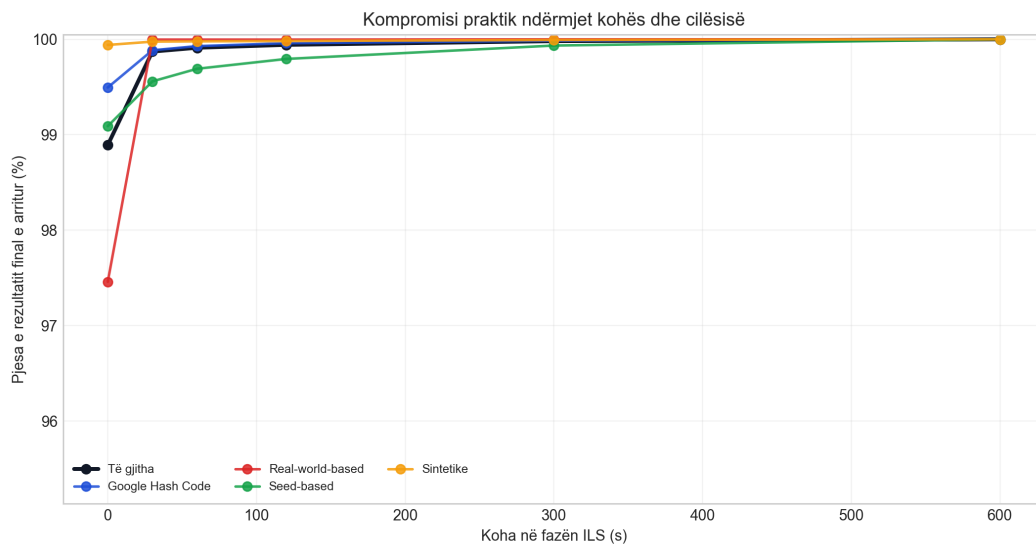


Figura 6.4: Pjesa e rezultatit final e arritur gjatë fazës ILS, sipas familjes së instancave.

Figura 6.5 tregon për secilën instancë kohën mesatare që i duhet ILS për të arritur 95% të përmirësimit final.

## 6. Rezultatet eksperimentale dhe diskutimi

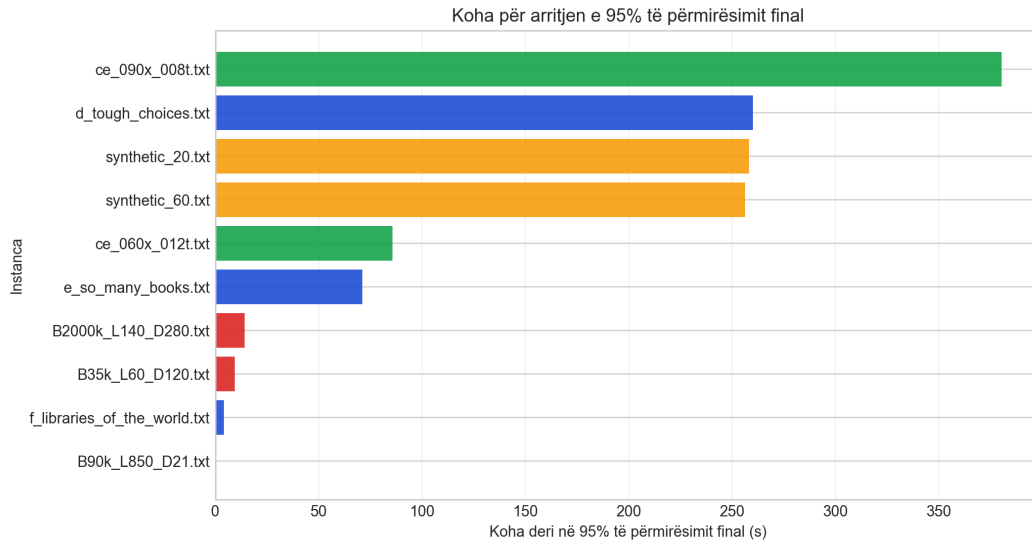


Figura 6.5: Koha mesatare për arritjen e 95% të përmirësimit final sipas instancës.

### 6.6. Qëndrueshmëria ndaj vlerave seed

Figura 6.6 paraqet koeficientin e variacionit të rezultatit final për 10 ekzekutimet e secilës instancë. Vlerat pranë zeros tregojnë se rezultati final ndryshon shumë pak nga një vlerë *seed* në tjetrën. Kjo e mbështet përdorimin e *Best* nga 10 ekzekutimet në krahasimin final, sepse tregon se rezultati nuk varet tepër nga një ekzekutim i vetëm.

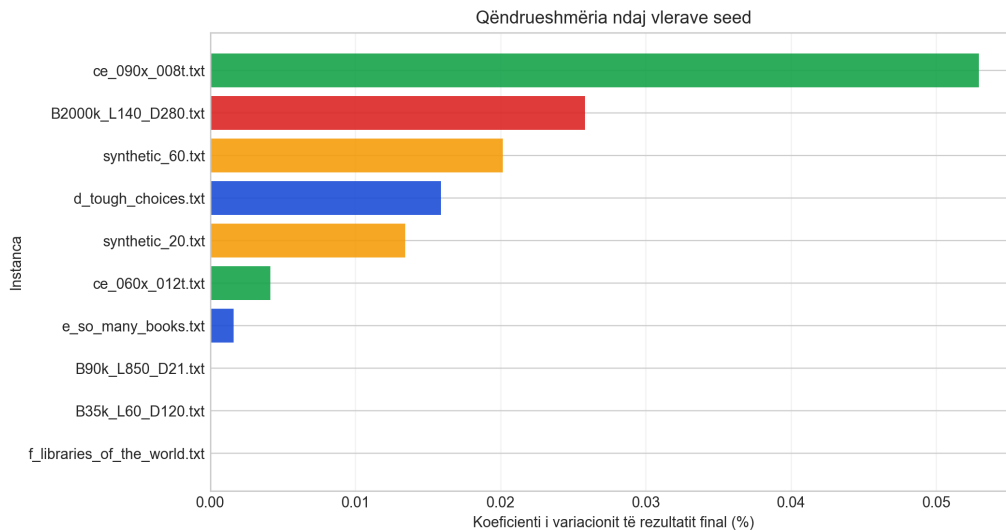


Figura 6.6: Koeficienti i variacionit i rezultatit final sipas instancës.

## 6.7. Krahاسimi me baseline dhe metodat për krahasim

Rezultatet finale raportohen në instancat e testimit. Kolona *ILS* tregon vlerën më të mirë nga 10 ekzekutime për secilën instancë, ndërsa kolona *ILS AVG* raporton mesataren e po atyre ekzekutimeve, pa shërbyer si vlerë kryesore e krahasimit final. Kjo ndarje e bën krahasimin më të qartë: *ILS* tregon cilësinë më të mirë që arrin konfigurimi final me zgjidhje fillestare të shumëfishta, ndërsa *ILS AVG* tregon stabilitetin e algoritmit.

Në totalin e instancave të testimit, kolona *ILS* arrin 477,429,716 pikë dhe kolona *ILS AVG* arrin 477,309,618 pikë, ndërsa *Baseline* arrin 466,414,103 pikë. Kështu, *ILS* e tejkalon *Baseline* me 11,015,613 pikë, që përfaqëson një epërsi relative prej 2.362%.

Figura 6.7 jep një pamje përmblledhëse të rezultateve totale sipas familjes së instancave. Ajo shërben si orientim vizual para tabelave të detajuara, ndërsa krahasimet instancë për instancë mbeten të raportuara në Tabelat 6.14 dhe 6.15.



Figura 6.7: Krahاسimi i rezultateve totale sipas familjes së instancave.

Tabela 6.13 paraqet testet Wilcoxon për krahasime në çifte në instancat e testimit. Fitoret, humbjet dhe barazimet numërohen duke krahasuar *ILS* me secilën metodë për krahasim në të njëjtën instancë.

Tabela 6.13: Testet Wilcoxon për krahasimet në çifte ndërmjet *ILS* dhe metodave për krahasim në instancat e testimit.

| Krahasimi | Fitore ILS | Humbje ILS | Barazime | Gap total ndaj ILS (%) | Gap AVG ndaj ILS (%) | Gap MED ndaj ILS (%) | $p_{\text{Wilcoxon}}$ (Holm) |
|-----------|------------|------------|----------|------------------------|----------------------|----------------------|------------------------------|
| Baseline  | 24         | 13         | 31       | <b>-2.362</b>          | <b>-0.576</b>        | 0.000                | 2.176e-04                    |
| Errecto   | 40         | 0          | 28       | <b>-4.202</b>          | <b>-3.513</b>        | <b>-0.320</b>        | 5.457e-12                    |
| Greedy    | 47         | 1          | 20       | <b>-16.826</b>         | <b>-13.835</b>       | <b>-6.647</b>        | 7.105e-14                    |
| SAT       | 48         | 1          | 19       | <b>-11.489</b>         | <b>-5.156</b>        | <b>-2.692</b>        | 7.816e-13                    |
| SA        | 50         | 1          | 17       | <b>-17.891</b>         | <b>-55.925</b>       | <b>-3.699</b>        | 1.306e-09                    |



## 6. Rezultatet eksperimentale dhe diskutimi

*MILP1* dhe *MILP2* nuk përfshihen në testin kryesor Wilcoxon për krahasime në çifte, sepse nuk prodhojnë rezultat të vlefshëm në të gjitha instancat e testimit. *MILP1* ka rezultate të vlefshme në 41 nga 68 instanca, ndërsa *MILP2* në 56 nga 68 instanca. Për të ruajtur krahasueshmërinë e testit statistikor, Tabela 6.13 raporton vetëm metodat me mbulim të plotë mbi të gjitha instancat e testimit. Rezultatet konkrete të *MILP1* dhe *MILP2* mbeten të raportuara në Tabelën 6.15.

Krahasimi me vlerën *Baseline* të metodave për krahasim është më i rreptë, sepse për çdo instancë merret rezultati më i lartë i disponueshëm nga i gjithë grupi i metodave. Për këtë arsye, fitoret ndaj kësaj kolone peshojnë më shumë sesa fitoret ndaj një metode të vetme.

Tabela 6.14 jep rezultatet konkrete për secilën instancë testuese. Kolona *UB* jep kufirin diagnostik të sipërm për instancën, kolona *ILS* është baza e krahasimit, ndërsa *Baseline* është rezultati më i mirë i disponueshëm nga metodat për krahasim për atë instancë. Kolona *Gap ndaj ILS* raporton dallimin relativ të *Baseline* ndaj *ILS*. Vlerat negative tregojnë qartë epërsi të *ILS*.

Tabela 6.14: Rezultatet konkrete në instancat e testimit. *ILS* raporton rezultatin më të mirë nga 10 ekzekutime. *UB* është kufi diagnostik i sipërm.

| Instanca                     | UB          | ILS AVG    | ILS               | Baseline   | Gap ndaj ILS (%) |
|------------------------------|-------------|------------|-------------------|------------|------------------|
| b_read_on.txt                | 10 000 000  | 5 822 900  | 5 822 900         | 5 822 900  | 0.000            |
| c_incunabula.txt             | 23 372 244  | 5 690 439  | 5 690 468         | 5 690 888  | 0.007            |
| d_tough_choices.txt          | 5 109 000   | 5 092 841  | 5 094 050         | 5 109 000  | 0.293            |
| e_so_many_books.txt          | 12 457 261  | 5 240 005  | 5 240 181         | 5 240 723  | 0.010            |
| f_libraries_of_the_world.txt | 39 885 586  | 5 348 248  | 5 348 248         | 5 348 248  | 0.000            |
| B1000k_L115_D230.txt         | 360 585 201 | 57 027 193 | <b>57 027 193</b> | 55 879 583 | <b>-2.054</b>    |
| B100_L9_D18.txt              | 6 246       | 4 916      | 4 916             | 4 916      | 0.000            |
| B1k_L18_D17.txt              | 283 465     | 127 579    | 127 579           | 127 579    | 0.000            |
| B2000k_L140_D280.txt         | 578 379 827 | 60 966 934 | <b>61 011 720</b> | 53 657 244 | <b>-13.706</b>   |
| B240k_L90_D180.txt           | 138 925 961 | 43 085 291 | <b>43 085 291</b> | 42 745 540 | <b>-0.795</b>    |
| B35k_L60_D120.txt            | 20 578 732  | 12 864 753 | 12 864 753        | 12 864 753 | 0.000            |
| B6k_L35_D70.txt              | 2 936 308   | 2 307 803  | 2 307 803         | 2 307 803  | 0.000            |
| B90_L6_D11.txt               | 3 704       | 2 567      | 2 567             | 2 567      | 0.000            |
| B90k_L850_D21.txt            | 418 275     | 869        | 869               | 869        | 0.000            |
| b_075x_012t.txt              | 3 374 450   | 185 334    | 185 334           | 185 341    | 0.004            |
| bc_060x_008t.txt             | 3 369 254   | 923 220    | 923 235           | 923 357    | 0.013            |
| bc_075x_020t.txt             | 5 168 217   | 1 299 742  | 1 299 768         | 1 299 813  | 0.003            |
| bd_060x_012t.txt             | 3 597 755   | 1 622 614  | 1 623 707         | 1 631 093  | 0.453            |
| bd_090x_008t.txt             | 6 073 672   | 2 374 322  | <b>2 375 036</b>  | 2 293 686  | <b>-3.547</b>    |
| be_060x_020t.txt             | 5 500 376   | 4 473 336  | <b>4 475 296</b>  | 4 469 247  | <b>-0.135</b>    |
| be_090x_012t.txt             | 9 261 438   | 7 351 926  | <b>7 353 771</b>  | 7 122 631  | <b>-3.245</b>    |
| bf_075x_008t.txt             | 7 011 170   | 4 493 613  | <b>4 494 063</b>  | 4 449 237  | <b>-1.007</b>    |
| bf_090x_020t.txt             | 7 969 327   | 6 872 910  | <b>6 872 910</b>  | 6 844 683  | <b>-0.412</b>    |
| c_075x_012t.txt              | 16 428 705  | 4 331 910  | 4 332 083         | 4 332 827  | 0.017            |
| cd_060x_008t.txt             | 11 362 607  | 4 428 199  | 4 428 885         | 4 460 167  | 0.701            |
| cd_075x_020t.txt             | 16 643 615  | 10 851 650 | 10 855 183        | 10 928 387 | 0.670            |
| ce_060x_012t.txt             | 18 615 160  | 13 013 811 | <b>13 014 383</b> | 12 787 403 | <b>-1.775</b>    |
| ce_090x_008t.txt             | 28 729 695  | 20 692 587 | <b>20 713 720</b> | 20 127 366 | <b>-2.913</b>    |
| cf_060x_020t.txt             | 16 143 221  | 13 441 428 | <b>13 442 088</b> | 13 382 904 | <b>-0.442</b>    |
| cf_090x_012t.txt             | 28 288 174  | 21 425 636 | <b>21 425 988</b> | 21 338 035 | <b>-0.412</b>    |
| d_075x_008t.txt              | 2 899 800   | 1 131 582  | 1 132 304         | 1 140 023  | 0.677            |
| d_090x_020t.txt              | 3 773 722   | 2 542 796  | 2 543 509         | 2 564 848  | 0.832            |
| de_075x_012t.txt             | 3 783 606   | 2 975 335  | <b>2 976 941</b>  | 2 874 196  | <b>-3.575</b>    |
| df_060x_008t.txt             | 2 933 572   | 2 007 479  | <b>2 007 570</b>  | 2 002 526  | <b>-0.252</b>    |
| df_075x_020t.txt             | 3 903 784   | 3 404 637  | <b>3 404 720</b>  | 3 390 357  | <b>-0.424</b>    |
| e_060x_012t.txt              | 7 723 605   | 5 631 492  | <b>5 649 393</b>  | 5 535 451  | <b>-2.058</b>    |
| e_090x_008t.txt              | 12 120 060  | 8 910 345  | <b>8 912 388</b>  | 8 704 950  | <b>-2.383</b>    |
| ef_060x_020t.txt             | 7 007 807   | 5 938 316  | <b>5 938 316</b>  | 5 918 919  | <b>-0.328</b>    |
| ef_090x_012t.txt             | 9 193 700   | 8 009 885  | <b>8 009 885</b>  | 7 978 463  | <b>-0.394</b>    |
| f_075x_008t.txt              | 24 117 937  | 18 243 051 | <b>18 243 051</b> | 18 186 218 | <b>-0.313</b>    |
| f_090x_020t.txt              | 29 232 950  | 27 578 316 | <b>27 579 675</b> | 27 429 014 | <b>-0.549</b>    |
| synthetic_10.txt             | 51 834      | 17 463     | 17 463            | 17 463     | 0.000            |
| synthetic_100.txt            | 156 794     | 156 794    | 156 794           | 156 794    | 0.000            |
| synthetic_105.txt            | 8 558 652   | 8 490 974  | 8 490 974         | 8 490 974  | 0.000            |
| synthetic_110.txt            | 14 805 607  | 3 350 131  | <b>3 351 703</b>  | 3 318 210  | <b>-1.009</b>    |
| synthetic_115.txt            | 345 277     | 345 277    | 345 277           | 345 277    | 0.000            |
| synthetic_120.txt            | 187 715     | 187 715    | 187 715           | 187 715    | 0.000            |
| synthetic_125.txt            | 330 735     | 330 735    | 330 735           | 330 735    | 0.000            |
| synthetic_130.txt            | 226 911     | 226 911    | 226 911           | 226 911    | 0.000            |

Vazhdon në faqen tjetër

6. Rezultatet eksperimentale dhe diskutimi

Tabela 6.14: Vazhdim i tabelës

| Instanca          | UB            | ILS AVG            | ILS                | Baseline    | Gap ndaj ILS (%) |
|-------------------|---------------|--------------------|--------------------|-------------|------------------|
| synthetic_135.txt | 217 206       | 217 206            | 217 206            | 217 206     | 0.000            |
| synthetic_15.txt  | 49 747        | 49 747             | 49 747             | 49 747      | 0.000            |
| synthetic_20.txt  | 31 034 336    | 29 691 957         | <b>29 698 214</b>  | 29 601 148  | <b>-0.328</b>    |
| synthetic_25.txt  | 244 295       | 244 295            | 244 295            | 244 295     | 0.000            |
| synthetic_30.txt  | 2 282 117     | 2 282 117          | 2 282 117          | 2 282 117   | 0.000            |
| synthetic_35.txt  | 163 375       | 163 375            | 163 375            | 163 375     | 0.000            |
| synthetic_40.txt  | 236 992       | 236 992            | 236 992            | 236 992     | 0.000            |
| synthetic_45.txt  | 11 254 939    | 1 063 158          | 1 063 158          | 1 063 451   | 0.028            |
| synthetic_5.txt   | 28 639        | 28 639             | 28 639             | 28 639      | 0.000            |
| synthetic_50.txt  | 142 479       | 142 479            | 142 479            | 142 479     | 0.000            |
| synthetic_55.txt  | 65 074        | 65 074             | 65 074             | 65 074      | 0.000            |
| synthetic_60.txt  | 33 626 312    | 18 673 734         | <b>18 682 072</b>  | 18 534 710  | <b>-0.795</b>    |
| synthetic_65.txt  | 132 057       | 132 057            | 132 057            | 132 057     | 0.000            |
| synthetic_70.txt  | 249 881       | 249 881            | 249 881            | 249 881     | 0.000            |
| synthetic_75.txt  | 140 755       | 140 755            | 140 755            | 140 755     | 0.000            |
| synthetic_80.txt  | 137 454       | 137 454            | 137 454            | 137 454     | 0.000            |
| synthetic_85.txt  | 2 597 661     | 2 597 661          | 2 597 661          | 2 597 661   | 0.000            |
| synthetic_90.txt  | 177 823       | 177 823            | 177 823            | 177 823     | 0.000            |
| synthetic_95.txt  | 171 405       | 171 405            | 171 405            | 171 405     | 0.000            |
| Total             | 1 594 785 261 | <b>477 309 618</b> | <b>477 429 716</b> | 466 414 103 | <b>-2.362</b>    |

Tabela 6.15 paraqet matricën e plotë të metodave për krahasim për secilën instancë testuese. Kjo tabelë shmang përmbledhjen e tepruar dhe e bën të dukshme se në cilat instancia ILS fiton, ku barazohet dhe ku mbetet nën metodën për krahasim më të mirë.

Tabela 6.15: Tabela konkrete e rezultateve për secilën instancë test. Vlerat me bold janë rezultati më i mirë në rresht.

| Instanca                     | ILS               | Errecto           | Greedy           | MILP1             | MILP2            | SA               | SAT              | Baseline          |
|------------------------------|-------------------|-------------------|------------------|-------------------|------------------|------------------|------------------|-------------------|
| b_read_on.txt                | <b>5 822 900</b>  | <b>5 822 900</b>  | <b>5 822 900</b> | 4 563 300         | <b>5 822 900</b> | 199 700          | <b>5 822 900</b> | <b>5 822 900</b>  |
| c_incunabula.txt             | 5 690 468         | 5 570 375         | 5 467 966        | <b>5 690 888</b>  | 3 943 983        | 5 388 615        | 5 436 542        | <b>5 690 888</b>  |
| d_tough_choices.txt          | 5 094 050         | 5 034 640         | 4 109 300        | -                 | 4 811 170        | 4 811 105        | <b>5 109 000</b> | <b>5 109 000</b>  |
| e_so_many_books.txt          | 5 240 181         | 5 183 726         | 4 395 880        | -                 | 5 165 502        | <b>5 240 723</b> | 5 119 283        | <b>5 240 723</b>  |
| f_libraries_of_the_world.txt | <b>5 348 248</b>  | 5 348 230         | 2 692 713        | -                 | <b>5 348 248</b> | 4 895 326        | 4 790 783        | <b>5 348 248</b>  |
| B1000k_L115_D230.txt         | <b>57 027 193</b> | 55 879 583        | 44 975 266       | -                 | 47 749 748       | 45 256 249       | 45 256 249       | 55 879 583        |
| B100_L9_D18.txt              | <b>4 916</b>      | <b>4 916</b>      | 4 838            | <b>4 916</b>      | 4 794            | 4 794            | 4 656            | <b>4 916</b>      |
| B1k_L18_D17.txt              | <b>127 579</b>    | <b>127 579</b>    | 116 176          | 110 125           | 69 397           | 102 814          | <b>127 579</b>   | <b>127 579</b>    |
| B2000k_L140_D280.txt         | <b>61 011 720</b> | 53 657 244        | 50 245 011       | -                 | 49 660 568       | 49 660 568       | 49 660 568       | 53 657 244        |
| B240k_L90_D180.txt           | <b>43 085 291</b> | 42 745 540        | 27 840 091       | -                 | 28 275 299       | 28 275 299       | 28 275 299       | 42 745 540        |
| B35k_L60_D120.txt            | <b>12 864 753</b> | <b>12 864 753</b> | 10 700 619       | -                 | 11 095 874       | 11 006 520       | 11 220 952       | <b>12 864 753</b> |
| B6k_L35_D70.txt              | <b>2 307 803</b>  | <b>2 307 803</b>  | <b>2 307 803</b> | <b>2 307 803</b>  | 2 076 043        | 2 076 043        | 2 232 682        | <b>2 307 803</b>  |
| B90_L6_D11.txt               | <b>2 567</b>      | <b>2 567</b>      | 1 770            | <b>2 567</b>      | 2 099            | 1 959            | 1 959            | <b>2 567</b>      |
| B90k_L850_D21.txt            | <b>869</b>        | <b>869</b>        | 383              | <b>869</b>        | 731              | 769              | 769              | <b>869</b>        |
| b_075x_012t.txt              | 185 334           | 185 277           | <b>185 341</b>   | 119 877           | 184 748          | 47 638           | 183 069          | <b>185 341</b>    |
| bc_060x_008t.txt             | 923 235           | 906 866           | 883 795          | <b>923 357</b>    | 552 841          | 870 975          | 880 863          | <b>923 357</b>    |
| bc_075x_020t.txt             | 1 299 768         | 1 277 606         | 1 241 021        | <b>1 299 813</b>  | 854 322          | 1 225 599        | 1 235 823        | <b>1 299 813</b>  |
| bd_060x_012t.txt             | 1 623 707         | 1 570 031         | 1 152 865        | <b>1 631 093</b>  | -                | 1 511 215        | 1 498 821        | <b>1 631 093</b>  |
| bd_090x_008t.txt             | <b>2 375 036</b>  | 2 293 686         | 1 631 894        | -                 | -                | 2 189 675        | 2 159 703        | 2 293 686         |
| be_060x_020t.txt             | <b>4 475 296</b>  | 4 424 008         | 4 097 154        | 4 469 247         | 4 400 132        | 2 654 337        | 4 368 656        | 4 469 247         |
| be_090x_012t.txt             | <b>7 353 771</b>  | 7 122 631         | 6 763 543        | -                 | 7 078 395        | 5 136 935        | 7 101 741        | 7 122 631         |
| bf_075x_008t.txt             | <b>4 494 063</b>  | 4 449 237         | 3 942 051        | -                 | 4 308 643        | 4 365 069        | 4 306 495        | 4 449 237         |
| bf_090x_020t.txt             | <b>6 872 910</b>  | 6 844 683         | 6 434 566        | -                 | 5 441 266        | 6 703 328        | 6 685 316        | 6 844 683         |
| c_075x_012t.txt              | 4 332 083         | 4 234 977         | 4 121 481        | <b>4 332 827</b>  | 2 852 536        | 4 407 416        | 4 125 628        | <b>4 332 827</b>  |
| cd_060x_008t.txt             | 4 428 885         | 2 940 560         | 2 802 396        | <b>4 460 167</b>  | -                | 4 016 724        | 3 966 757        | <b>4 460 167</b>  |
| cd_075x_020t.txt             | 10 855 183        | 4 844 762         | 8 343 070        | <b>10 928 387</b> | -                | 10 057 136       | 10 250 204       | <b>10 928 387</b> |
| ce_060x_012t.txt             | <b>13 014 383</b> | 12 787 403        | 11 452 572       | -                 | -                | 7 681 906        | 12 516 542       | 12 787 403        |
| ce_090x_008t.txt             | <b>20 713 720</b> | 20 127 366        | 18 490 154       | -                 | -                | 15 652 651       | 19 903 692       | 20 127 366        |
| cf_060x_020t.txt             | <b>13 442 088</b> | 13 382 904        | 12 579 883       | -                 | 10 414 100       | 13 246 860       | 13 226 601       | 13 382 904        |
| cf_090x_012t.txt             | <b>21 425 988</b> | 21 338 035        | 19 469 453       | -                 | 19 219 717       | 20 912 724       | 20 806 960       | 21 338 035        |
| d_075x_008t.txt              | 1 132 304         | 1 107 304         | 753 742          | <b>1 140 023</b>  | -                | 1 035 889        | 1 018 047        | <b>1 140 023</b>  |
| d_090x_020t.txt              | 2 543 509         | 2 482 618         | 2 055 273        | <b>2 564 848</b>  | -                | 2 401 922        | 2 406 977        | <b>2 564 848</b>  |
| de_075x_012t.txt             | <b>2 976 941</b>  | 2 873 496         | 2 726 980        | -                 | 2 842 247        | 2 118 299        | 2 874 196        | 2 874 196         |
| df_060x_008t.txt             | <b>2 007 570</b>  | 2 002 526         | 1 751 522        | -                 | 1 926 601        | 1 963 279        | 1 955 323        | 2 002 526         |
| df_075x_020t.txt             | <b>3 404 720</b>  | 3 390 357         | 3 210 910        | -                 | 2 739 159        | 3 337 290        | 3 322 203        | 3 390 357         |
| e_060x_012t.txt              | <b>5 649 393</b>  | 5 535 451         | 5 062 174        | -                 | -                | 3 510 702        | 5 459 039        | 5 535 451         |
| e_090x_008t.txt              | <b>8 912 388</b>  | 8 704 950         | 7 989 533        | -                 | -                | 6 761 321        | 8 570 668        | 8 704 950         |
| ef_060x_020t.txt             | <b>5 938 316</b>  | 5 918 919         | 5 493 347        | -                 | 4 597 635        | 5 881 370        | 5 859 563        | 5 918 919         |
| ef_090x_012t.txt             | <b>8 009 885</b>  | 7 978 463         | 7 451 779        | -                 | 7 204 118        | 7 869 923        | 7 873 076        | 7 978 463         |
| f_075x_008t.txt              | <b>18 243 051</b> | 18 186 218        | 16 271 547       | -                 | 17 421 370       | 17 903 514       | 17 761 379       | 18 186 218        |
| f_090x_020t.txt              | <b>27 579 675</b> | 27 429 014        | 26 562 032       | -                 | 23 478 277       | 27 261 783       | 27 249 240       | 27 429 014        |
| synthetic_10.txt             | <b>17 463</b>     | 17 442            | 16 142           | <b>17 463</b>     | 16 768           | <b>17 463</b>    | <b>17 463</b>    | <b>17 463</b>     |
| synthetic_100.txt            | <b>156 794</b>    | <b>156 794</b>    | <b>156 794</b>   | <b>156 794</b>    | <b>156 794</b>   | <b>156 794</b>   | <b>156 794</b>   | <b>156 794</b>    |
| synthetic_105.txt            | <b>8 490 974</b>  | 8 488 353         | 8 459 918        | <b>8 490 974</b>  | 8 487 090        | 8 478 835        | 8 487 090        | <b>8 490 974</b>  |
| synthetic_110.txt            | <b>3 351 703</b>  | 3 318 210         | 2 923 482        | -                 | 3 025 414        | 3 023 135        | 3 006 040        | 3 318 210         |
| synthetic_115.txt            | <b>345 277</b>    | <b>345 277</b>    | <b>345 277</b>   | <b>345 277</b>    | <b>345 277</b>   | <b>345 277</b>   | <b>345 277</b>   | <b>345 277</b>    |
| synthetic_120.txt            | <b>187 715</b>    | <b>187 715</b>    | <b>187 715</b>   | <b>187 715</b>    | <b>187 715</b>   | <b>187 715</b>   | <b>187 715</b>   | <b>187 715</b>    |
| synthetic_125.txt            | <b>330 735</b>    | <b>330 735</b>    | <b>330 735</b>   | <b>330 735</b>    | <b>330 735</b>   | <b>330 735</b>   | <b>330 735</b>   | <b>330 735</b>    |
| synthetic_130.txt            | <b>226 911</b>    | <b>226 911</b>    | <b>226 911</b>   | <b>226 911</b>    | -                | <b>226 911</b>   | <b>226 911</b>   | <b>226 911</b>    |
| synthetic_135.txt            | <b>217 206</b>    | <b>217 206</b>    | <b>217 206</b>   | <b>217 206</b>    | <b>217 206</b>   | <b>217 206</b>   | <b>217 206</b>   | <b>217 206</b>    |
| synthetic_15.txt             | <b>49 747</b>     | <b>49 747</b>     | <b>49 747</b>    | <b>49 747</b>     | <b>49 747</b>    | <b>49 747</b>    | <b>49 747</b>    | <b>49 747</b>     |

Vazhdon në faqen tjetër

Tabela 6.15: Vazhdim i tabelës

| Instanca         | ILS         | Errecto     | Greedy      | MILP1      | MILP2       | SA          | SAT         | Baseline    |
|------------------|-------------|-------------|-------------|------------|-------------|-------------|-------------|-------------|
| synthetic_20.txt | 29 698 214  | 29 601 148  | 28 983 692  | -          | 29 058 495  | 29 029 072  | 29 038 110  | 29 601 148  |
| synthetic_25.txt | 244 295     | 244 295     | 244 295     | 244 295    | 244 295     | 244 295     | 244 295     | 244 295     |
| synthetic_30.txt | 2 282 117   | 2 282 117   | 2 282 117   | 2 282 117  | 2 282 117   | 2 282 117   | 2 282 117   | 2 282 117   |
| synthetic_35.txt | 163 375     | 163 375     | 163 375     | 163 375    | -           | 163 375     | 163 375     | 163 375     |
| synthetic_40.txt | 236 992     | 236 992     | 228 204     | 236 992    | 223 718     | 224 885     | 223 713     | 236 992     |
| synthetic_45.txt | 1 063 158   | 1 044 638   | 865 559     | 1 063 451  | 938 744     | 928 704     | 927 380     | 1 063 451   |
| synthetic_5.txt  | 28 639      | 28 639      | 28 639      | 28 639     | 28 639      | 28 639      | 28 639      | 28 639      |
| synthetic_50.txt | 142 479     | 142 479     | 142 479     | 142 479    | 141 424     | 141 178     | 141 120     | 142 479     |
| synthetic_55.txt | 65 074      | 65 074      | 65 074      | 65 074     | 65 074      | 65 074      | 65 074      | 65 074      |
| synthetic_60.txt | 18 682 072  | 18 534 710  | 17 544 919  | -          | 17 930 950  | 17 887 435  | 17 867 164  | 18 534 710  |
| synthetic_65.txt | 132 057     | 132 057     | 131 164     | 132 057    | 125 642     | 125 968     | 126 212     | 132 057     |
| synthetic_70.txt | 249 881     | 249 881     | 249 881     | 249 881    | 249 881     | 249 881     | 249 881     | 249 881     |
| synthetic_75.txt | 140 755     | 140 755     | 140 755     | 140 755    | 140 755     | 140 755     | 140 755     | 140 755     |
| synthetic_80.txt | 137 454     | 137 454     | 137 454     | 137 454    | 137 454     | 137 454     | 137 454     | 137 454     |
| synthetic_85.txt | 2 597 661   | 2 597 661   | 2 597 661   | 2 597 661  | 2 597 661   | 2 597 661   | 2 597 661   | 2 597 661   |
| synthetic_90.txt | 177 823     | 177 823     | 177 427     | 177 823    | 177 746     | 176 769     | 176 853     | 177 823     |
| synthetic_95.txt | 171 405     | 171 405     | 171 405     | 171 405    | 170 867     | 170 966     | 170 946     | 171 405     |
| Total            | 477 429 716 | 458 178 966 | 408 668 821 | 62 406 387 | 346 872 671 | 404 974 010 | 428 231 530 | 466 414 103 |

## 6.8. Interpretimi i rezultateve

Nga rezultatet e mësipërme dalin tri përfundime kryesore. Së pari, strategjitë konstruktive janë të forta dhe në shumë instanca prodhojnë zgjidhje afër rezultatit final, prandaj përmirësimi pas fazës fillestare nuk është gjithmonë i madh në përqindje. Së dyti, përfitimi i ILS bëhet më i dukshëm në instancat ku renditja e bibliotekave lë hapësirë reale për riorganizim, veçanërisht në pjesë të familjeve real-world-based dhe seed-based. Së treti, kërkimi lokal del si bërthama kryesore e performancës. Perturbimi, rifillimi dhe pranimi probabilistik nuk japin përfitim të pavarur të qëndrueshëm në total dhe në disa përmbledhje dalin fare pak poshtë varianteve të thjeshtuara, por shërbejnë si mekanizma eksplorimi, stabilizimi dhe rikuperimi brenda konfigurimit të plotë.

Rastet ku ILS mbetet pas vlerës *Baseline* janë të kufizuara dhe të strukturuara. Ato shfaqen kryesisht në instanca ku formulimet MILP, SAT, SA ose zgjidhjet e raportuara nga gara janë veçanërisht të përshtatura me strukturën e problemit. Kjo nuk e dobëson rezultatin kryesor, por tregon kufirin praktik të ILS kundrejt metodave të specializuara për krahasim në nënfamilje të caktuara.

Nga këndvështrimi akademik, rezultati më i rëndësishëm nuk është vetëm tejkalimi i një vlere referuese për krahasim, por demonstrimi se një kornizë ILS e riprodhueshme, e akorduar në një grup të ndarë trajnimit dhe e vlerësuar në instanca testimi, arrin performancë konkurruese në familje instancash me karakteristika të ndryshme. Analizat e komponentëve, të grupeve të operatorëve, të konvergjencës dhe të qëndrueshmërisë ndaj vlerave *seed* e mbështesin këtë përfundim duke treguar jo vetëm nivelin e rezultatit, por edhe sjelljen e metodës gjatë ekzekutimit.

## 7. Përfundimet

### 7.1. Përmbledhje e punimit

Ky punim e trajtoi problemin e skanimit të librave si problem të optimizimit kombinatorik me renditje sekuenciale të bibliotekave, kapacitete paralele skanimit dhe mbivendosje të librave ndërmjet bibliotekave. U zhvillua dhe u analizua një qasje e bazuar në kërkimin lokal të përsëritur, e cila bashkon disa strategji konstruktive, rindërtim të shpejtë të alokimit të librave, operatorë lokalë, perturbim adaptiv, pranim të kontrolluar dhe rifillim pas stagnimit.

Konfigurimi final u akordua me iRace në grupin e trajnimit dhe u vlerësua në instancat e testimit, me 10 ekzekutime për çdo instancë testuese. Rezultatet kryesore përdorin vlerën më të mirë nga 10 ekzekutimet (*Best*) si bazë krahasimi dhe raportojnë *AVG* për transparencë. Metoda e propozuar arrin rezultat *Best* total prej 477,429,716 pikësh dhe *AVG* total prej 477,309,618 pikësh. Kolona *Baseline*, e cila përfaqëson vlerën më të mirë të metodave për krahasim, arrin 466,414,103 pikë në të njëjtin grup testimi. ILS e tejkalon kështu Baseline me 11,015,613 pikë, që përfaqëson një epërsi relative prej 2.362%. Testet Wilcoxon për krahasime në çift, me korrigjim Holm, mbështesin dallimet statistikore në favor të ILS në krahasimet kryesore me mbulim të plotë të instancave.

### 7.2. Përgjigje ndaj pyetjeve kërkimore

1. Kërkimi lokal i përsëritur rezulton i përshtatshëm për këtë problem, sepse renditja e bibliotekave është përfaqësim i natyrshëm dhe operatorët lokalë mund t'i ndryshojnë me efektivitet ditët e aktivizimit dhe kapacitetet efektive të bibliotekave. Kjo i lejon metodës të eksplorojë zgjidhje fqinje pa e rindërtuar problemin nga e para në çdo hap.
2. Konfigurimi i parametrave i akorduar me iRace jep një zgjedhje të qëndrueshme të peshave të operatorëve, të parametrave të perturbimit dhe të mekanizmave të pranimit. Punimi megjithatë nuk pretendon epërsi ndaj çdo konfigurimi të mundshëm të paakorduar. Ai tregon se konfigurimi final është efektiv brenda dizajnit eksperimental të përdorur.
3. Kundrejt Baseline, ILS arrin rezultat më të lartë në 24 instanca, barazohet në 31 dhe mbetet më poshtë në 13. Rrjedhimisht, ILS është të paktën po aq i mirë sa Baseline në 55 nga 68 instancat e testimit. Në total, ajo arrin *Best* 477,429,716 pikë dhe e tejkalon Baseline me 11,015,613 pikë, që përfaqëson një epërsi relative prej 2.362%. Përfitimet janë më të dukshme në familjet real-world-based dhe seed-based, ndërsa në instancat zyrtare të Google Hash Code dallimet janë më të vogla, sepse zgjidhjet e raportuara nga gara janë të specializuara pikërisht për ato instanca.
4. Eksperimentet me komponentët e algoritmit tregojnë se kërkimi lokal është komponenti

dominues i cilësisë. Lëvizjet që fusin ose shkëmbejnë biblioteka kanë ndikim më të madh se rirenditjet e thjeshta, ndërsa perturbimi, rifillimi dhe pranimi i kontrolluar vlejnë më shumë si mekanizma stabilizimi dhe rikuperimi sesa si burim i madh përmirësimi në çdo instancë.

5. Performanca ndryshon sipas strukturës së familjeve të instancave. Familjet real-world-based dhe seed-based ofrojnë më shumë hapësirë për përmirësim, ndërsa shumë instanca sintetike barazohen shpejt nga disa metoda. Vlerësimi me 10 ekzekutime për instancë mundëson raportimin e *Best* dhe *AVG*, pra të cilësisë më të mirë dhe të stabilitetit të konfigurimit stokastik. Koeficienti i variacionit mbetet i ulët nëpër shumicën e familjeve, çka tregon se konfigurimi final nuk varet ndjeshëm nga një vlerë e vetme seed.
6. Cilësia e zgjidhjes fillestare ka rëndësi, por nuk është faktori i vetëm. Me zgjidhjet fillestare të konfigurimit të plotë, përmirësimi i përgjithshëm nga rezultati fillestar te rezultati final është 0.533%. Me renditje fillestare të thjeshtë, përmirësimi gjatë kërkimit rritet në 13.04%, ndërsa rezultati final mbetet vetëm 0.042% nën konfigurimin e plotë sipas vlerës *Best* dhe 0.129% sipas *AVG*. Kjo tregon se ILS ka aftësi të mirë korrigjuese. Edhe kur zgjidhja fillestare është e dobët, kërkimi lokal i përsëritur arrin ta afrojë shumë rezultatin final me atë të konfigurimit të plotë.
7. Analiza e konvergjencës tregon se konfigurimi final ILS nis me më shumë se 97% të cilësisë së rezultatit përfundimtar. Koha mesatare për arritjen e 95% të përmirësimit final është 134 sekonda, por ndryshon sipas familjes së instancave. Disa instanca konvergojnë pothuajse menjëherë, ndërsa instanca të familjeve seed-based dhe sintetike vazhdojnë të përfitojnë deri afër fundit të buxhetit prej 600 sekondash.

### 7.3. Kufizimet

Punimi ka disa kufizime. Së pari, kufiri i sipërm i përdorur është i thjeshtë dhe shpesh shumë i lirë. Prandaj diferenca ndaj tij shërben vetëm si diagnostikë e brendshme dhe nuk interpretohet si afërsi e saktë ndaj optimumit. Së dyti, metodat për krahasim janë ekzekutuar si konfigurime me një vlerë për instancë, ndërsa ILS është stokastik dhe vlerësohet me 10 ekzekutime. Kjo është metodologjikisht e arsyeshme për natyrën e secilës metode, por nuk e zëvendëson një dizajn plotësisht simetrik, ku metodat krahasuese do të ekzekutoheshin me vlera të ndryshme *seed* dhe me të njëjtin buxhet kohor. Së treti, eksperimenti është i kushtueshëm në kohë, sepse çdo ekzekutim ka deri në 120 sekonda ndërtim fillestar dhe 600 sekonda kërkim ILS. Eksperimentimi me komponentët e algoritmit tregon se varianti me kërkim lokal të thjeshtë është shumë konkurrues me kohë dukshëm më të ulët, prandaj në praktikë konfigurimi duhet zgjedhur sipas buxhetit të disponueshëm. Së katërti, qasja e propozuar nuk izolon një operator të vetëm krejtësisht të ri. Kontributi i saj kryesor është integrimi i komponentëve të përshtatur enkas për problemin, vlerësimi i shpejtë, rindërtimi i alokimit të librave dhe analiza e gjerë

eksperimentale. Së pesti, qasja përqendrohet te renditja e bibliotekave dhe përdor procedura heuristike për alokimin e librave. Një alokim ekzakt i librave për çdo renditje do të ishte më i shtrenjtë, por potencialisht më i fortë në disa instanca. Së gjashti, MILP1 dhe MILP2 nuk kanë rezultate të vlefshme për të gjitha instancat e testimit, përkatësisht 41 nga 68 dhe 56 nga 68, prandaj përjashtohen nga testi kryesor Wilcoxon për krahasime në çifte dhe krahasohen vetëm në nivel instance, aty ku kanë rezultat.

## 7.4. Puna e ardhshme

Drejtimit më të rëndësishme për punë të ardhshme janë:

- zhvillimi i kufijve të sipërm më të ngushtë, për ta matur më saktë cilësinë absolute të zgjidhjeve.
- integrimi i procedurave më të avancuara për alokimin e librave.
- akordimi i parametrave për të balancuar cilësinë e rezultatit me kohën e ekzekutimit.
- shfrytëzimi i të dhënave nga ekzekutimet e mëparshme për të zgjedhur automatikisht cilët operatorë të kërkimit lokal duhen përdorur më shpesh në instanca me karakteristika të caktuara.
- krahasimi me qasje të bazuara në gjenerimin dhe përmirësimin automatik të funksioneve të vlerësimit, si FunSearch, brenda të njëjtit dizajn eksperimental.
- vlerësimi i metodave për krahasim me një dizajn eksperimental më simetrik, ku metodat krahasuese ekzekutohen me më shumë vlera *seed* dhe me buxhet kohor të kontrolluar.
- ekzekutimi paralel i disa kërkimeve lokale të pavarura, duke e ndarë buxhetin kohor ndërmjet disa zgjidhjeve fillestare, në vend që kërkimi të mbështetet në një nisje të vetme.

## 7.5. Diskutim përmbyllës

Rezultatet e këtij punimi tregojnë se një metaheuristikë klasike, e projektuar me kujdes dhe e mbështetur nga implementim efikas, mund të jetë konkurruese në një problem modern të optimizimit kombinatorik. Përfitimi ndaj metodave të forta për krahasim është real, por i përqendruar. Shumë instanca barazohen, disa humben ndaj zgjidhjeve ekzakte ose të specializuara, ndërsa fitimet më të qarta shfaqen në familjet real-world-based dhe seed-based. Prandaj vlera kryesore e qasjes nuk qëndron vetëm te rezultati final, por te riprodhueshmëria, analizueshmëria dhe besueshmëria e saj në shumë familje instancash. Në tërësi, rezultatet tregojnë se një qasje klasike metaheuristike, e projektuar me kujdes dhe e analizuar në mënyrë sistematike, mbetet një alternativë konkurruese për probleme moderne të optimizimit kombinatorik me shkallë të madhe.

## 8. Conclusions

### 8.1. Thesis Summary

This thesis addressed the Google Hash Code 2020 Book Scanning problem as a combinatorial optimization problem with sequential library sign-up, parallel scanning capacity, overlapping book inventories, and a finite time horizon. A problem-specific Iterated Local Search approach was developed and analyzed. The method combines several constructive strategies, fast reconstruction of book allocation, local operators, adaptive perturbation, controlled acceptance, and restart after stagnation.

The final configuration was tuned with iRace on the train dataset and evaluated on the test dataset, using 10 runs per test instance. The main results use the Best score over these runs as the comparison value and keep AVG for transparency. The proposed method achieves a Best total score of 477,429,716 and an AVG total score of 477,309,618. The Baseline column, which represents the best available comparison result, reaches 466,414,103 on the same test dataset. ILS therefore exceeds Baseline by 11,015,613 points, representing a relative advantage of 2.362%. Wilcoxon tests for paired comparisons, with Holm correction, support the statistical differences in favor of ILS in the main full-coverage comparisons.

### 8.2. Answers to the Research Questions

1. Iterated Local Search is suitable for this problem because the ordering of libraries is a natural representation, and local operators can effectively change activation days and effective scanning capacities. This allows the method to explore neighboring schedules without rebuilding the whole problem from scratch at every step.
2. The iRace-tuned parameter configuration provides a stable choice of operator weights, perturbation parameters, and acceptance mechanisms. However, the thesis does not claim superiority over every possible untuned configuration. It shows that the final configuration is effective within the experimental design used in this work.
3. Compared with Baseline, ILS achieves a higher score on 24 instances, ties on 31, and obtains a lower score on 13. Consequently, ILS is at least as good as Baseline on 55 of the 68 test instances. In total, it reaches a Best score of 477,429,716 and exceeds Baseline by 11,015,613 points, representing a relative advantage of 2.362%. The gains are most visible in the real-world-based and seed-based families, while on the official Google Hash Code instances the differences are smaller because the reported contest solutions are highly specialized for those cases.
4. The experiments with algorithm components show that local search is the dominant

quality component. Moves that insert or exchange libraries have a larger effect than simple reordering moves, while perturbation, restart, and controlled acceptance mainly contribute as stabilization and recovery mechanisms rather than as sources of large improvement on every instance.

5. Performance depends on the structure of the instance families. The real-world-based and seed-based families provide more room for improvement, whereas many synthetic instances are quickly tied by several methods. The evaluation with 10 runs per instance allows the thesis to report both *Best* and *AVG*, namely the best achieved quality and the stability of the stochastic configuration. The coefficient of variation remains low across most families, showing that the final configuration does not depend heavily on any single seed value.
6. The quality of the initial solution matters, but it is not the only factor. With the initial solutions of the full configuration, the overall improvement from the initial score to the final score is 0.533%. With a simple initial ordering, the improvement during search rises to 13.04%, while the final score remains only 0.042% below the full configuration in terms of *Best* and 0.129% in terms of *AVG*. This shows that ILS has strong corrective ability. Even when the initial solution is weak, iterated local search brings the final result very close to that of the full configuration.
7. The convergence analysis shows that the final ILS configuration starts with more than 97% of the final solution quality. The mean time required to reach 95% of the final improvement is 134 seconds, although it varies across instance families. Some instances converge almost immediately, whereas instances from the seed-based and synthetic families continue to benefit until near the end of the 600-second budget.

### 8.3. Limitations

Several limitations remain. First, the upper bound used in the thesis is simple and often loose, so the gap to it is used only as an internal diagnostic and is not interpreted as an accurate distance to the optimum. Second, the comparison methods are executed as single-value configurations per instance, while ILS is stochastic and evaluated with 10 runs. This is methodologically reasonable for the nature of each method, but it does not replace a fully symmetric experimental design where the comparison methods would be run with different *seed* values and the same time budget.

Third, the experiment is computationally expensive because each run uses up to 120 seconds for initial construction and 600 seconds for ILS search. The component experiments show that the simple local-search variant is highly competitive at much lower runtime, so practical use should choose the configuration according to the available budget. Fourth, the proposed approach does not isolate a completely new operator. Its main contribution is problem-specific integration, fast evaluation, book-allocation reconstruction, and broad



experimental analysis. Fifth, the approach focuses on library ordering and uses heuristic procedures for book allocation. Exact book allocation for every ordering would be more expensive, but potentially stronger on some instances. Sixth, MILP1 and MILP2 do not have valid results on all test instances, covering 41 of 68 and 56 of 68 respectively, so they are excluded from the main Wilcoxon test for paired comparisons and compared only at the instance level, where a result is available.

### 8.4. Future Work

The most important directions for future work are:

- developing stronger upper bounds to measure absolute solution quality more accurately.
- integrating more advanced procedures for book allocation.
- tuning parameters to balance solution quality with runtime.
- using data from previous runs to automatically choose which local-search operators should be applied more often on instances with certain characteristics.
- comparing the method with approaches that evolve scoring functions, such as FunSearch, under the same experimental design.
- evaluating the comparison methods under a more symmetric experimental design, where the comparison methods are run with more seed values and a controlled time budget.
- running several independent local searches in parallel, splitting the time budget across multiple initial solutions instead of relying on a single starting point.

### 8.5. Concluding Remarks

The results of this thesis show that a classical metaheuristic, carefully designed and supported by an efficient implementation, can be competitive on a modern combinatorial optimization problem. The gain over strong comparison methods is real but concentrated. Many instances are tied, some are lost against exact or specialized methods, and the clearest gains appear in the real-world-based and seed-based families. Therefore, the main value of the approach lies not only in the final score, but also in its reproducibility, analyzability, and reliability across many instance families. Overall, the results show that a carefully designed and systematically analyzed classical metaheuristic remains a competitive alternative for large-scale modern combinatorial optimization problems.

# Referencat

- [1] Google. Book Scanning: Problem Statement for the Online Qualification Round of Hash Code 2020. [https://github.com/pierreavn/google-hashcode-archive/blob/main/archive/2020/qualification/hashcode\\_2020\\_qualification\\_round.pdf](https://github.com/pierreavn/google-hashcode-archive/blob/main/archive/2020/qualification/hashcode_2020_qualification_round.pdf), 2020. Specifikimi zyrtar i problemit. Qasur më 13 korrik 2026.
- [2] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.
- [3] Helena R. Lourenco, Olivier C. Martin, and Thomas Stutzle. Iterated local search. In Fred Glover and Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, pages 320–353. Springer, Boston, MA, 2003. doi:10.1007/0-306-48056-5\_11.
- [4] Sean Luke. *Essentials of Metaheuristics*. Lulu, 2 edition, 2013. Versioni online 2.4, i disponueshëm në <https://people.cs.gmu.edu/~sean/book/metaheuristics/>. Qasur më 13 korrik 2026.
- [5] Nawid Sayed. Google-Hash-Code-Qualification-2020. <https://github.com/nawidsayed/Google-Hash-Code-Qualification-2020>, 2020. Zgjidhja origjinale, me kontribut idesh nga Andre Weltsch. Qasur më 13 korrik 2026.
- [6] Bartosz Kostka. GHC 2020: Google Hash Code 2020 Repository. <https://github.com/Kostero/ghc20>, 2020. Zgjidhja origjinale e ekipit Errecto, të renditur në vendin e pestë në raundin kualifikues. Dokumenton procedurën dyfazore (kërkim lokal i rastësishëm mbi renditjen + ndërtim lakmitar). Qasur më 13 korrik 2026.
- [7] Driton Alija. Google-Hash-Code-Qualification-2020: Adapted Fork. <https://github.com/dritonalija/Google-Hash-Code-Qualification-2020>, 2026. Zgjerim dhe përshtatje e zgjidhjes origjinale të Nawid Sayed për strukturën hyrëse/dalëse dhe infrastrukturën eksperimentale të këtij punimi. Qasur më 13 korrik 2026.
- [8] Uran Lajqi. Book.Scanning.Errecto. <https://github.com/uran-lajci/Book.Scanning.Errecto>, 2026. Zgjerim dhe përshtatje e zgjidhjes origjinale të ekipit Errecto për përdorim në infrastrukturën eksperimentale të këtij punimi. Qasur më 13 korrik 2026.
- [9] Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi:10.1038/s41586-023-06924-6.
- [10] Petar Velickovic, Alex Vitvitskyi, Larisa Markeeva, Borja Ibarz, Lars Buesing, Matej Balog, and Alexander Novikov. Amplifying human performance in combinatorial

- competitive programming. <https://arxiv.org/abs/2411.19744>, 2024. arXiv:2411.19744.
- [11] Driton Alija, Meriton Kryeziu, Endrit Hoda, Lirim Islami, Lorik Mustafa, Yll Berisha, Kadri Sylejmani, Arben Ahmeti, and Labeat Arbneshti. Optimization of the book scanning problem via iterated local search and genetic algorithms. In *Proceedings of the 6th International Conference on Recent Trends and Applications in Computer Science and Information Technology (RTA-CSIT 2025)*, volume 4044 of *CEUR Workshop Proceedings*, pages 127–135. CEUR-WS.org, 2025. I disponueshëm në <https://ceur-ws.org/Vol-4044/paper14.pdf>.
- [12] Driton Alija. ils\_book\_scanning. [https://github.com/dritonalija/ils\\_book\\_scanning](https://github.com/dritonalija/ils_book_scanning), 2026. Repozitori kryesor që përmban implementimin, skriptet eksperimentale, rezultatet e gjeneruara dhe artefaktet e punimit. Qasur më 13 korrik 2026.
- [13] Yll Berisha. BORT (Book Scanning OR-Tools): Book Scanning Problem Using OR-Tools. <https://github.com/yllberisha/Bort>, 2025. Implementim i formulimit MILP për problemin Book Scanning. Qasur më 13 korrik 2026.
- [14] Endrit Gjoka and Bleron Morina. Book Scanning Problem Input Generator. <https://github.com/endritgjoka/hashcodeproblem-inputgenerator>, 2026. Repozitori i mjetit online për gjenerimin e instancave real-world-based për problemin Book Scanning. Aplikacioni është i disponueshëm në <https://bookscanning-ig.netlify.app/>. Qasur më 13 korrik 2026.
- [15] Uran Lajqi. Book.Scanning.Dataset. <https://github.com/uran-lajci/Book.Scanning.Dataset>, 2026. Instanca sintetike për Book Scanning. Qasur më 13 korrik 2026.
- [16] Driton Alija. book\_scanning\_dataset. [https://github.com/dritonalija/book\\_scanning\\_dataset](https://github.com/dritonalija/book_scanning_dataset), 2026. Instanca të gjeneruara për Book Scanning me vlera seed të kontrolluara. Qasur më 13 korrik 2026.
- [17] Emil Indzhev. Book Scanning Solution: Google Hash Code 2020. [https://github.com/indjev99/Optimizational-Problems-Solutions/blob/master/book\\_scanning.cpp](https://github.com/indjev99/Optimizational-Problems-Solutions/blob/master/book_scanning.cpp), 2020. Repozitor që dokumenton zgjidhjen me peshim sipas frekuencës së librave, të renditur 28-ti në garë. Qasur më 13 korrik 2026.
- [18] Pedro Rodrigues. Google Hash Code Models. <https://github.com/pedromig/hashcode-models>, 2023. Repozitor që dokumenton një qasje të optimizimit në rrjet për alokimin optimal të librave, të bazuar në algoritmin *minimum-cost maximum-flow*. Qasur më 13 korrik 2026.
- [19] Pedro Miguel Duque Rodrigues. Principled Modeling of the Google Hash Code Problems for Meta-Heuristics. Master's thesis, Universidade de Coimbra, 2023. Trajtim akademik i plotë i problemeve Google Hash Code, përfshirë Book Scanning. Raporton vendin e 33-të

- në leaderboard për zgjidhjen përkatëse. I disponueshëm në <https://hdl.handle.net/10316/110471>. Qasur më 13 korrik 2026.
- [20] Ala Eddine Yahiaoui, Sohaib Afifi, and Hamid Allaoui. Enhanced iterated local search for the technician routing and scheduling problem. *Computers & Operations Research*, 160: 106385, 2023. doi:10.1016/j.cor.2023.106385.
- [21] Thomas A. Feo and Mauricio G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995. doi:10.1007/BF01096763.
- [22] Manuel Lopez-Ibanez, Jeremie Dubois-Lacoste, Leslie Perez Caceres, Mauro Birattari, and Thomas Stutzle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016. doi:10.1016/j.orp.2016.09.002. Faqja zyrtare e paketës është e disponueshme në <https://mlopez-ibanez.github.io/irace/>. Qasur më 13 korrik 2026.
- [23] Numba Developers. Numba: A High Performance Python Compiler. <https://numba.pydata.org/>, 2026. Bibliotekë Python për përshpejtim JIT (*just-in-time*) të kodit numerik, e përdorur për vlerësimin e shpejtë të kandidatëve. Qasur më 13 korrik 2026.
- [24] Tobias Achterberg. Scip: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009. doi:10.1007/s12532-008-0001-1. Faqja zyrtare e SCIP është e disponueshme në <https://scipopt.org/>. Qasur më 13 korrik 2026.
- [25] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6): 80–83, 1945. doi:10.2307/3001968.
- [26] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. Regjistrimi bibliografik është i disponueshëm në <https://www.jstor.org/stable/4615733>. Qasur më 13 korrik 2026.