

Hiperheurističke metode za rješavanje kombinatornih optimizacijskih problema

Marko Đurasević

Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

18. siječnja 2023.

Pregled

- 1 Motivacija
- 2 Postupci rješavanja
- 3 Hiperheuristike
- 4 Napredne teme
- 5 Zaključak

Motivacija

Heuristike, metaheuristike, hiper-heuristike

- Heuristika - problemski specifična metoda za rješavanje nekog problema
- Metaheuristika - Problemski neovisna metoda za rješavanje širokog skupa problema
- Hiperheuristika - ???

Uvod

- Kombinatorni optimizacijski problemi
- Primjena u stvarnom svijetu:
 - Raspoređivanje
 - Produkcija
 - Raspored predavanja i ispita
 - Logistika
 - Usmeravanje vozila
- Većinom NP problemi

Problem usmjeravanja vozilom

- Vehicle routing problem (VRP) [1]
- Široka primjena u stvarnom svijetu
- N kupaca potrebno je poslužiti s V vozila
- Karakteristike problema:
 - Vozila imaju ograničen kapacitet
 - Svaki kupac mora biti poslužen
 - Kupac može biti poslužen unutar određenog vremenskog prozora
- Kriteriji:
 - Prijeđena udaljenost
 - Ostvareni profit
 - Kašnjenje
 - Ukupni trošak
 - Potrebno vrijeme

Varijante problema raspoređivanja vozila

- VRP s prikupljanjem i dostavama [2]
- VRP sa zonama [3]
- VRP s električnim vozilima [4]
- ...

Postupci rješavanja

Metode

- Egzaktni postupci
 - Garantirano pronalaze optimalno rješenje
 - Primjenjivi samo za manje probleme
- Metaheuristike i heuristike koje pretražuju prostor stanja
 - Ne garantiraju optimalnost rješenja
 - Pronalaze rješenja koja su dovoljno dobra
 - Vrijeme izvršavanja je prihvatljivo

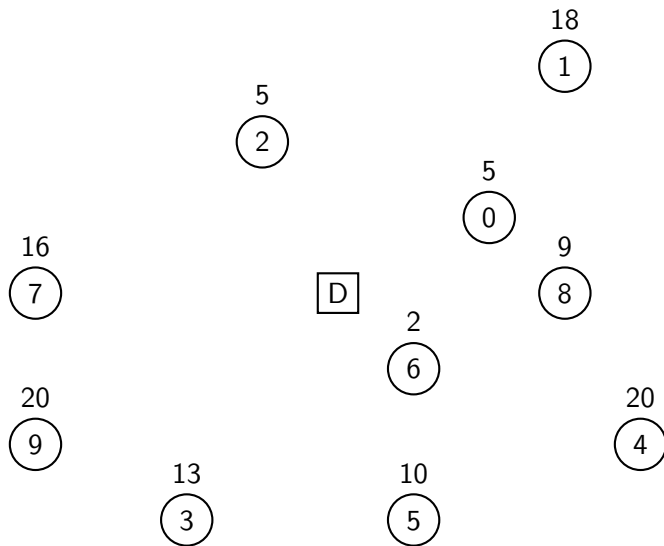
Nedostaci i ograničenja

- Veliki problemi
 - Povećava se vrijeme izvođenja
 - Opada efikasnost postupaka
- Dinamički problemi
 - Sve informacije o problemu su nedostupne u početku
 - Neočekivane promjene u sustavu
 - Potreba za brzom reakcijom
 - Kako pretražiti prostor rješenja kada unaprijed nije poznat?

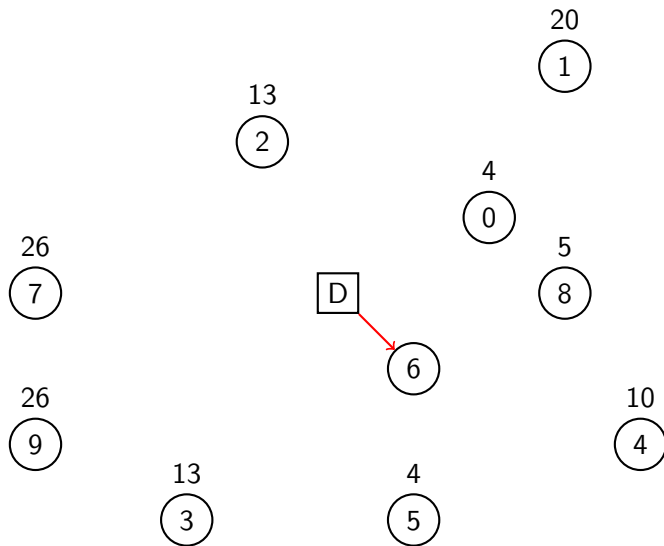
Konstruktivne heuristike

- Grade rješenja inkrementalno, korak po korak
- Ne izrađuju dio rješenja koji je u budućnosti
- Mogu brzo reagirati na promjene u sustavu
- Brzo donose odluku o idućem stanju

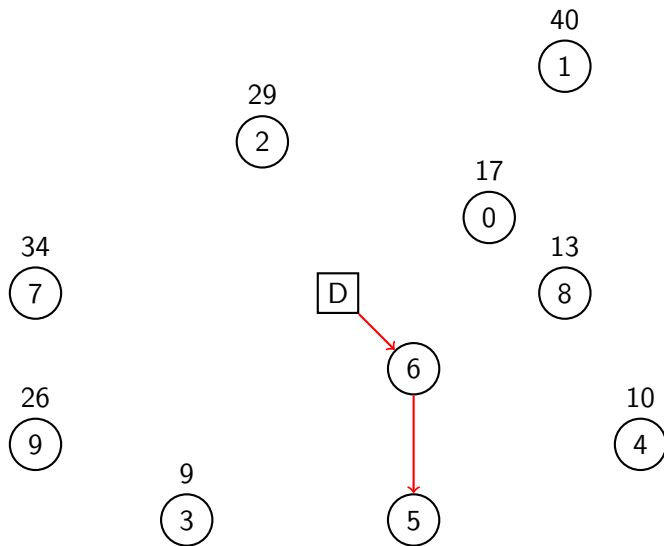
Najbliži susjed



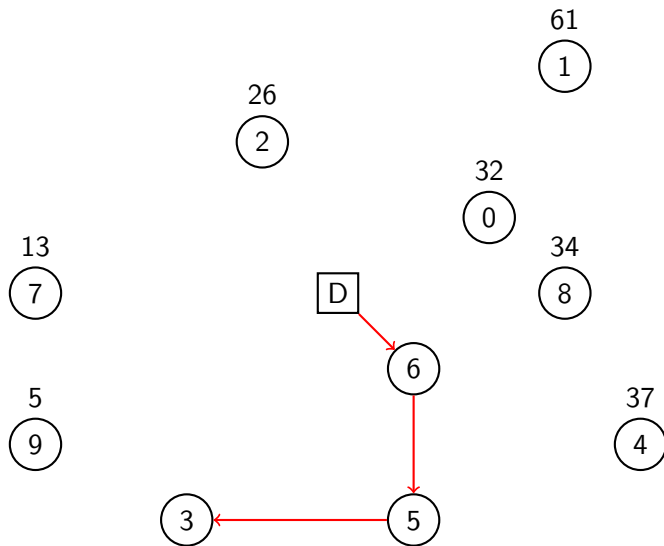
Najbliži susjed



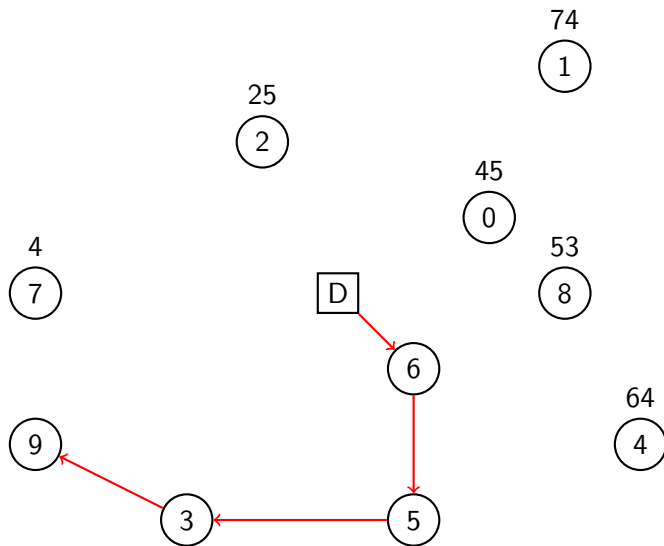
Najbliži susjed



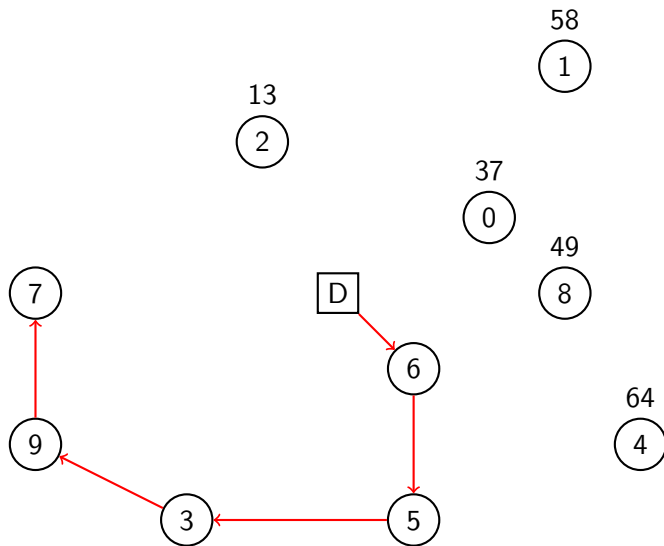
Najbliži susjed



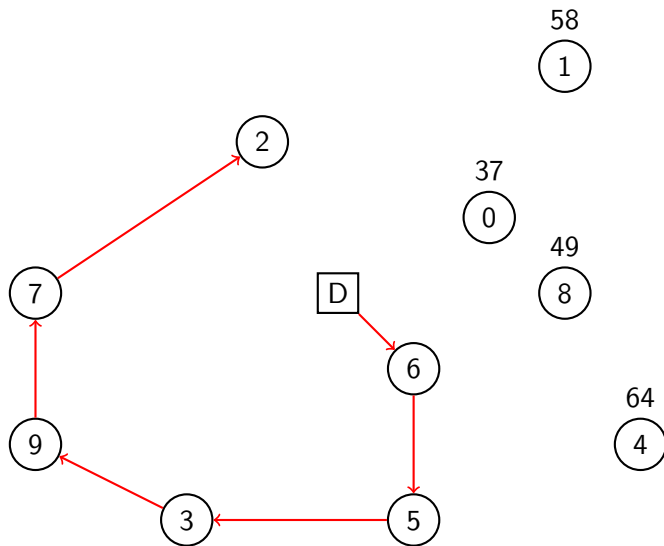
Najbliži susjed



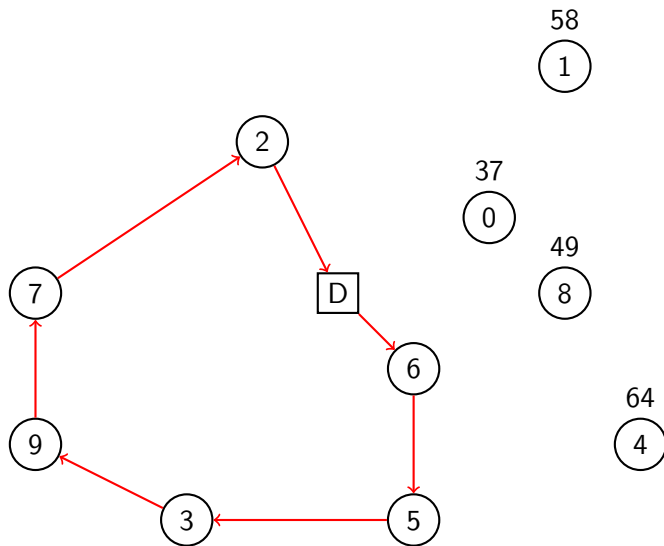
Najbliži susjed



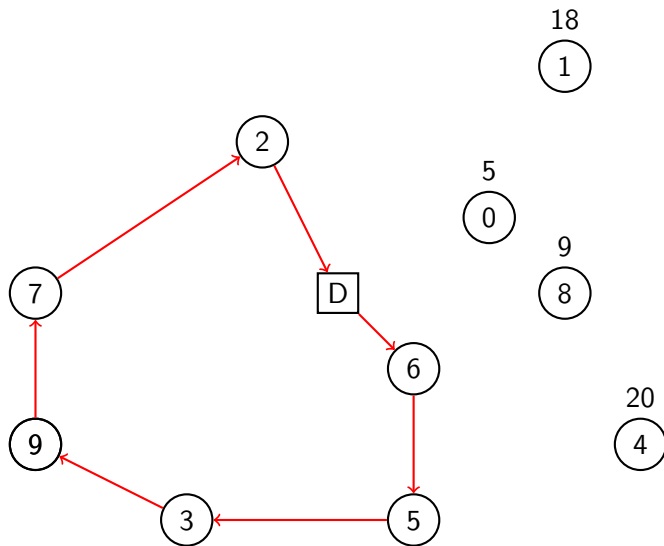
Najbliži susjed



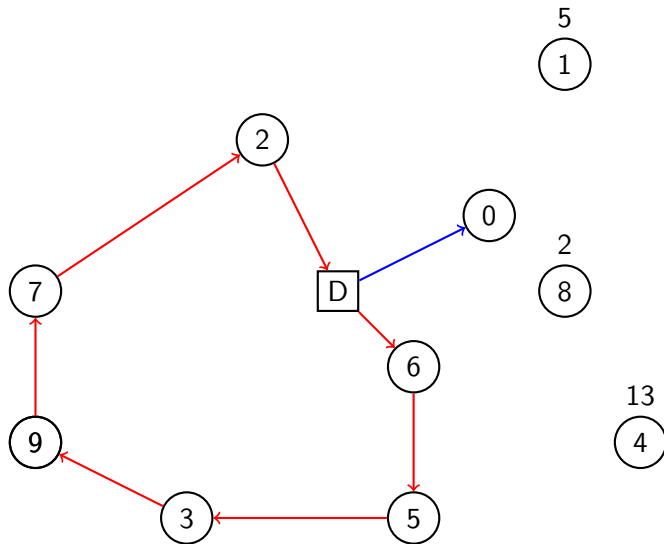
Najbliži susjed



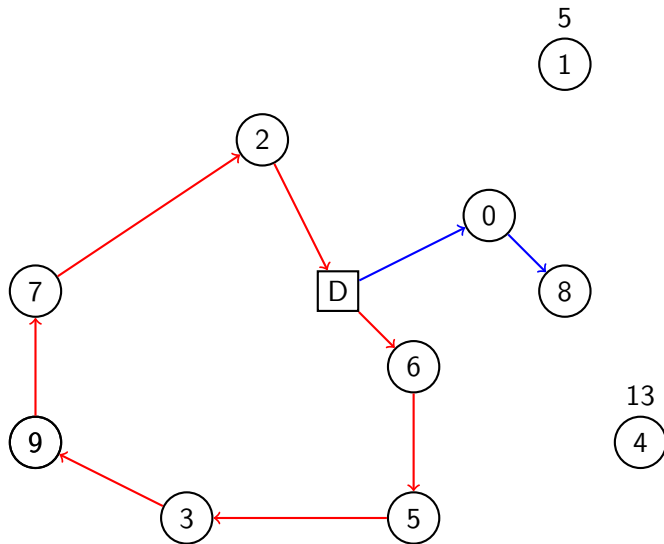
Najbliži susjed



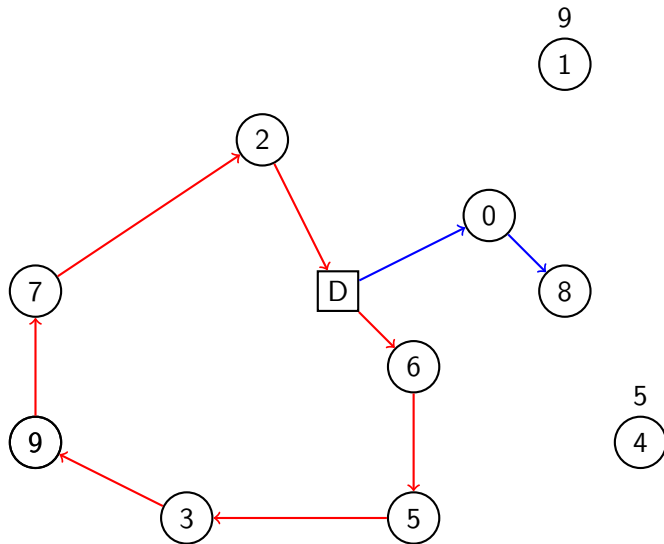
Najbliži susjed



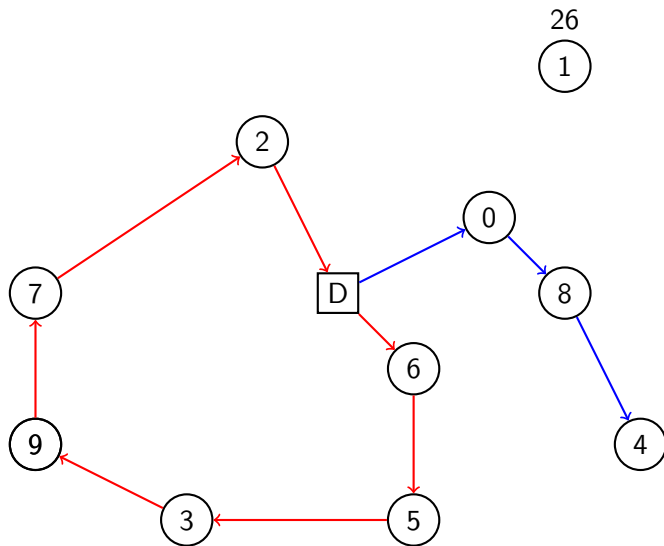
Najbliži susjed



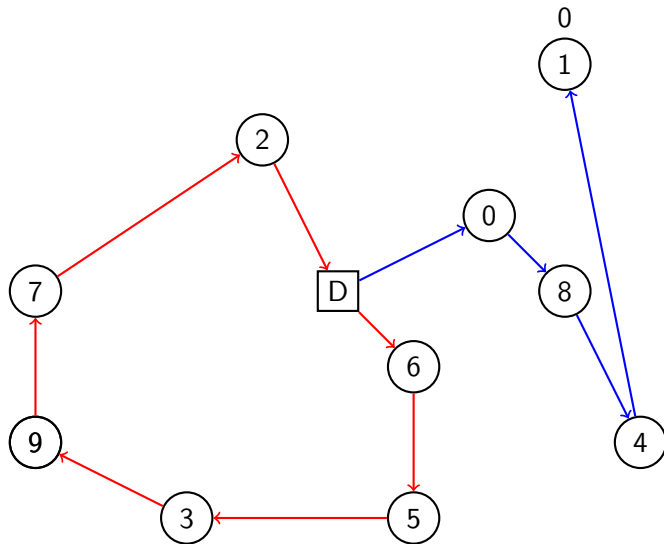
Najbliži susjed



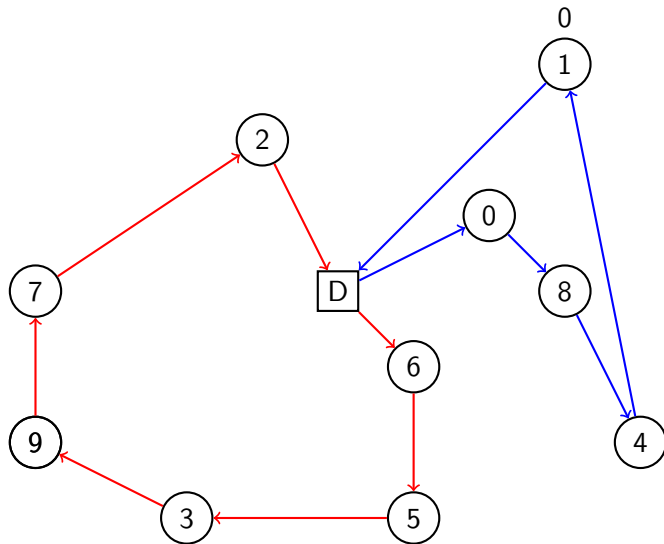
Najbliži susjed



Najbliži susjed



Najbliži susjed



Problemi

- Ograničena kvaliteta rješenja
- Izrada zahtjeva dobro poznavanje domene
- Proces izrade je dugotrajan i mukotrpan
 - Pokušaj i promašaj
- Problemski specifične
 - Jedna heuristika primjenjiva za konkretnu varijantu problema/kriterij
 - Postoji mnogo varijanti problema i kriterija
 - Potrebno izraditi heuristiku za svaki od njih
- Višekriterijska optimizacija

Hiperheuristike

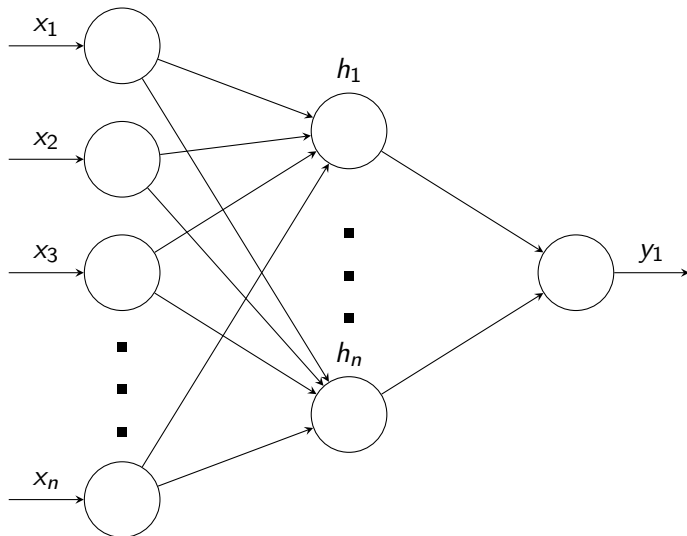
Hiperheuristike

- Metode koje izrađuju heuristike [5]
- Najčešće zasnovane na evolucijskim algoritmima
- Ideja:
 - Razviti prioritetnu funkciju koja ocjenjuje svaku odluku
 - Odabrati i izvesti odluku s najvećim/najmanjim prioritetom

Prioritetna funkcija

- $\pi(x)$
- Ocjenjuje svaku odluku
- $x \mapsto \pi(x)$
- Ulaz: karakteristike problema
- Izlaz: numerička ocjena
- Cilj hiperheuristika je pronaći optimalnu prioritetnu funkciju π
- Kako predstaviti prioritetnu funkciju?

Neuronske mreže



Neuronske mreže

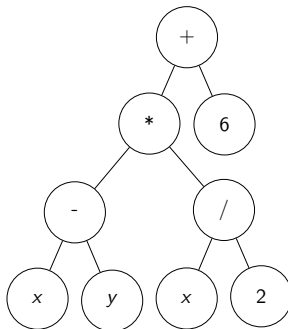
- Prednosti:
 - Popularan model
 - Odličan aproksimator
- Nedostaci:
 - Nadzirano učenje - što je očekivani izlaz?
 - Rješenje - optimizacija težina metaheuristikama
 - Potrebno optimirati mnogobrojne hiperparametre (broj slojeva, neurona, prijenosne funkcije)
 - Kako interpretirati mrežu?

Genetsko programiranje

- Metaheuristika slična genetskim algoritmima [6]
- Skup slučajno generiranih rješenja
- Modifikacija kroz razne operatore:
 - Križanje
 - Mutacija
- Razlika s obzirom na genetski algoritam?

Genetsko programiranje

- Jedinka predstavlja matematički izraz ili program
- $(x - y) * \frac{x}{2} + 6$

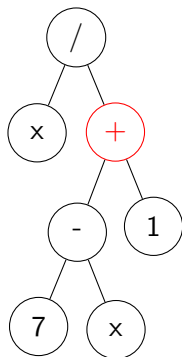


Genetsko programiranje

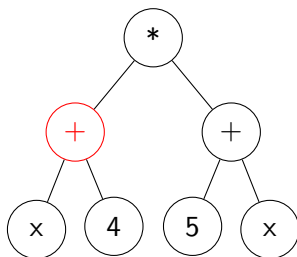
- Potrebno definirati građevne element, skup primitiva
- Funkcijski čvorovi
 - Razni matematički operatori ($+$, $-$, $*$)
 - Logički operatori (i , ili)
 - Funkcije grananja (ako)
- Terminalni čvorovi:
 - Odabir ovisi o domeni za koju se koriste
 - Varijable – predstavljaju informacije o problemu
 - Konstante

Križanje

Roditelj 1

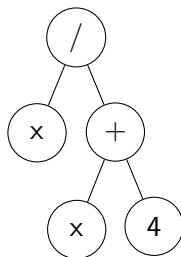


Roditelj 2



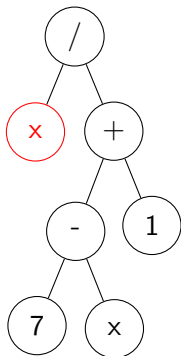
križanje →

Dijete



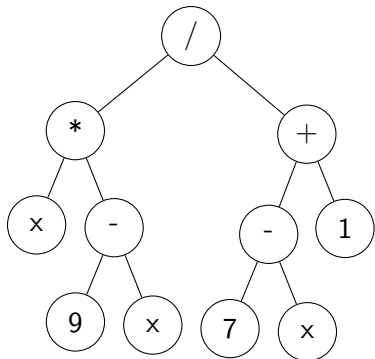
Mutacija

Prije mutacije



mutacija →

Nakon mutacije



Primjena genetskog programiranja na optimizacijski problem

- Potrebno definirati 2 stvari:
 - Heuristika za izradu rješenja
 - Informacije o problemu koje će koristiti prioritetna funkcija

Primjena genetskog programiranje za VRP

- Pretpostavke:
 - Razmatramo dinamično okruženje – može doći do promjena
 - Optimiramo trošak koji ovisi o više faktora
 - Prijeđeni put
 - Kašnjenje
 - Trajanje svake rute

Heuristika za izradu rasporeda

- 1: **dok** Postoje neposjećeni kupci **ponavljaj**
- 2: Čekaj dok barem jedno vozilo nije slobodno
- 3: **za** svako slobodno vozilo v **ponavljaj**
- 4: **za** svakog neposjećenog kupca i **ponavljaj**
- 5: Izračunaj prioritet π_{iv}
- 6: **kraj za**
- 7: **kraj za**
- 8: Odredi kupca s najvećim π_{iv} i usmjeri vozilo v prema njemu
- 9: **kraj dok**

Informacije o problemu

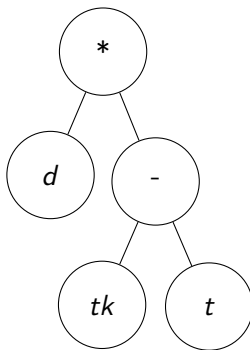
- Jednostavne karakteristike
 - Udaljenost do idućeg kupca – d
 - Kapacitet idućeg kupca – c
 - Početak prozora posluživanja – tp
 - Kraj prozora posluživanja – tk
 - Trenutno vrijeme – t
- Složenije karakteristike
 - Udaljenost do idućeg najbližeg neposječenog kupca
 - Udaljenost do centroida neposječenih kupaca
 - Vrijeme do početka prozora – $tp - t$
 - Vrijeme do kraja prozora – $tk - t$

Kako odabrati prikladne terminale

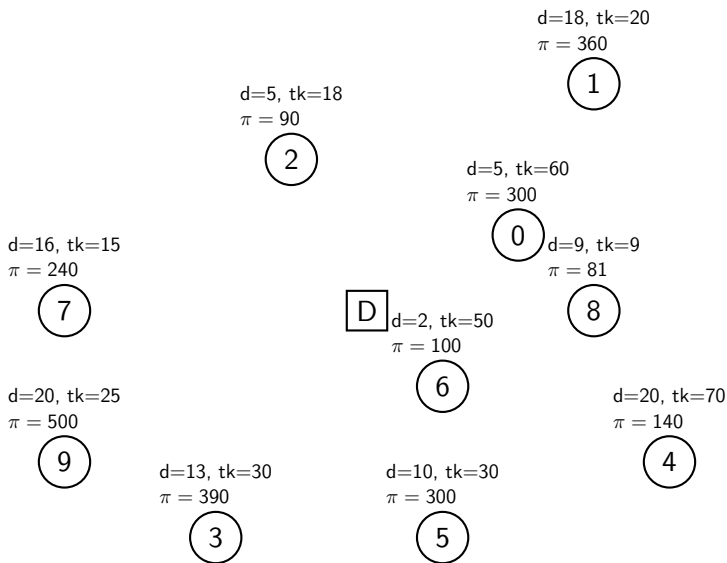
- Treba odabrati sve one koji su bitni za rješavanje problema
- Pokušaj i promašaj...
- Koristiti samo jednostavnije
 - Genetsko programiranje bi trebalo moći otkriti složenije ovisnosti
 - Treba mu jako dugo da ih otkrije i lako ih može izgubiti zbog genetskih operatora
- Koristiti složenije
 - Definirati smislene složenije terminale
 - Obično treba koristiti i jednostavnije da se izrazi mogu modificirati
- Općenito najteži dio u genetskom programiranju

Prioritetna funkccija

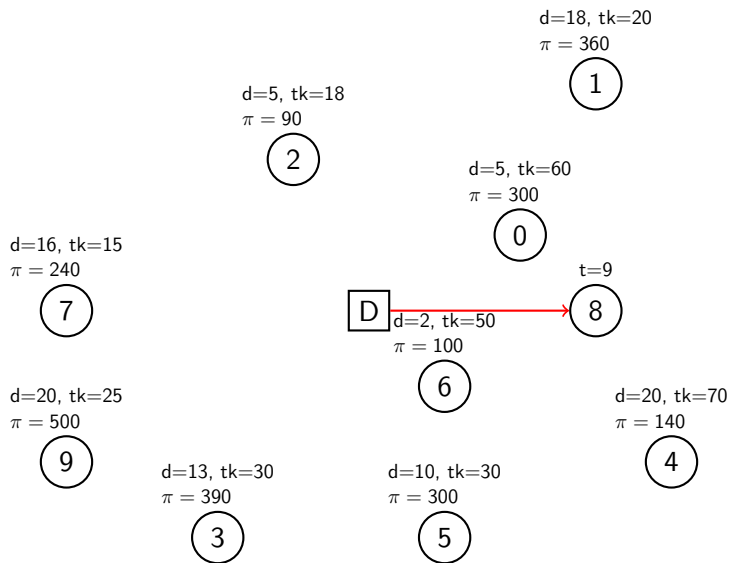
- Uzmi u obzir i distancu i kraj vremenskog prozora
- $d * (tk - t)$



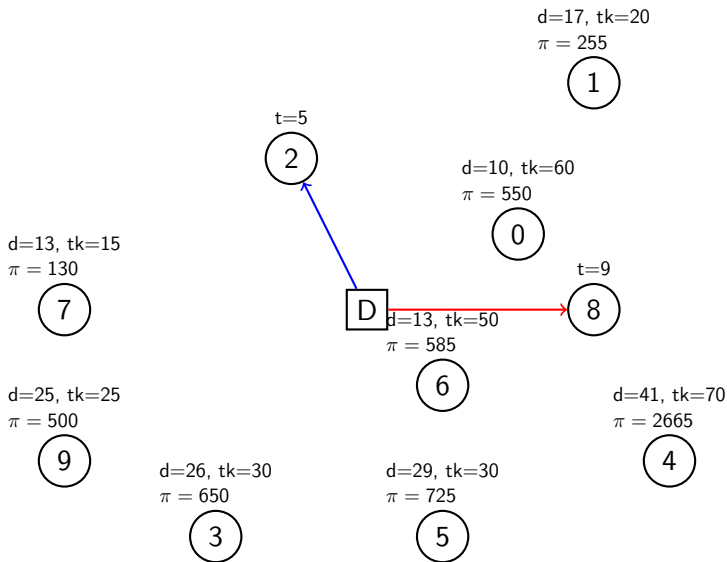
Vozilo 1



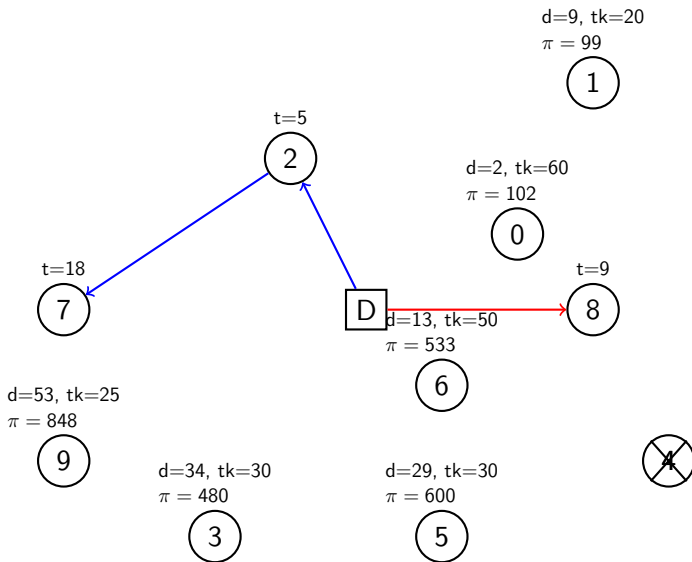
Vozilo 2



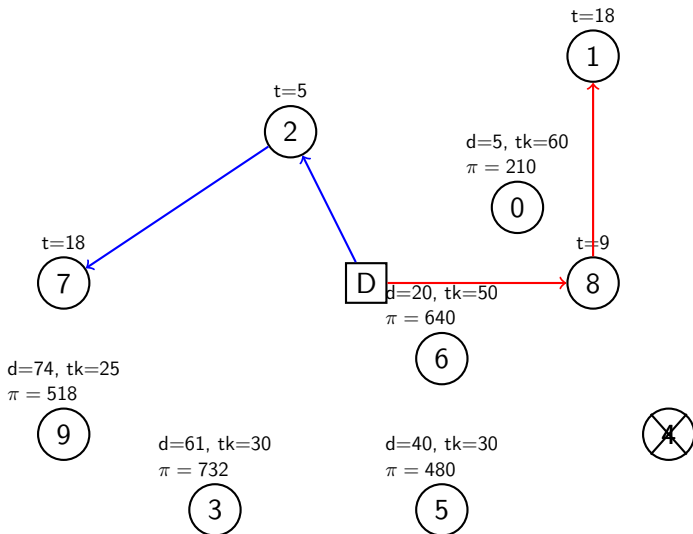
vozilo 2



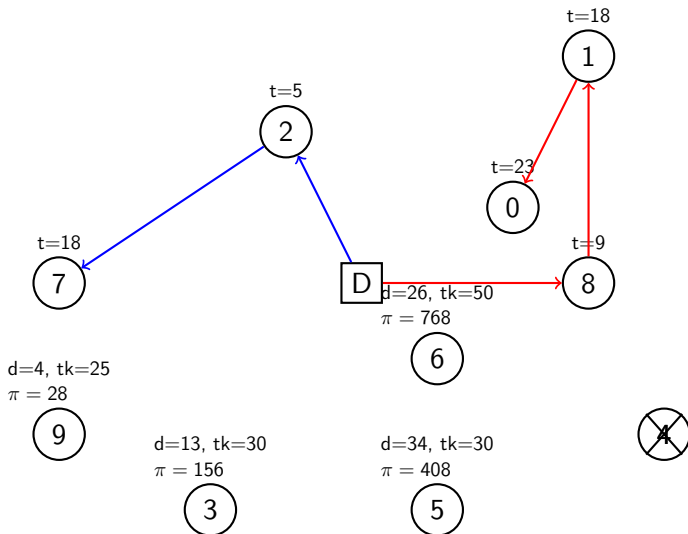
Vozilo 1



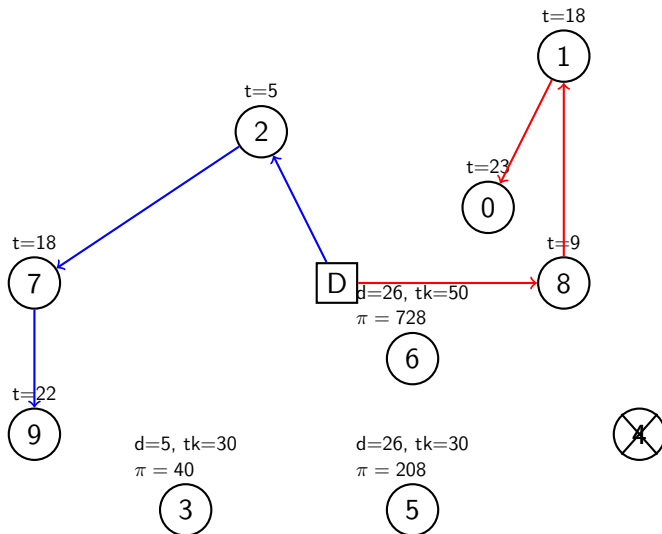
Vozilo 1



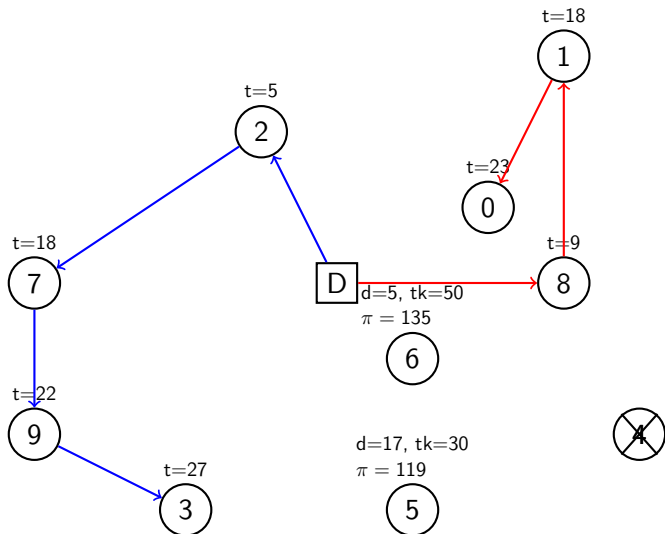
Vozilo 2



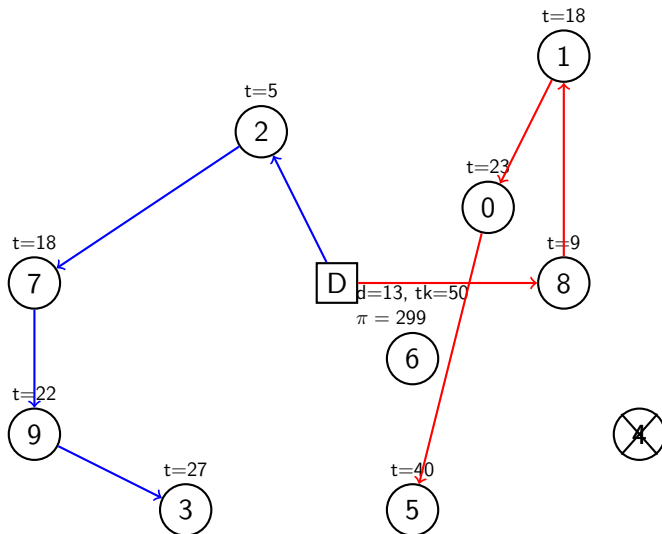
Vozilo 2



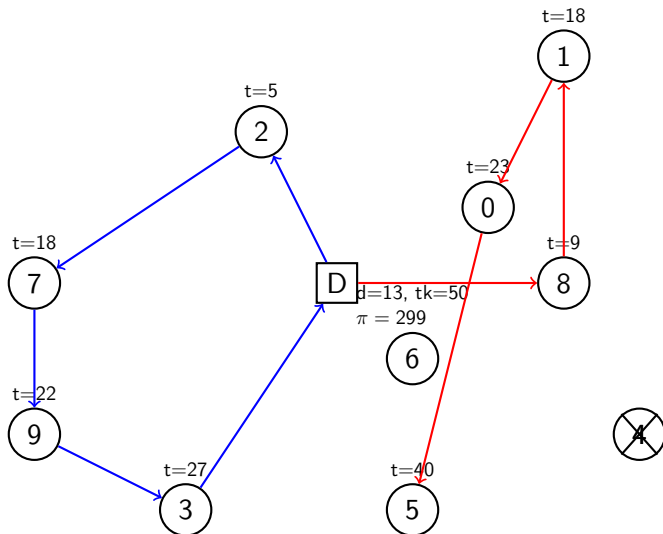
Vozilo 1



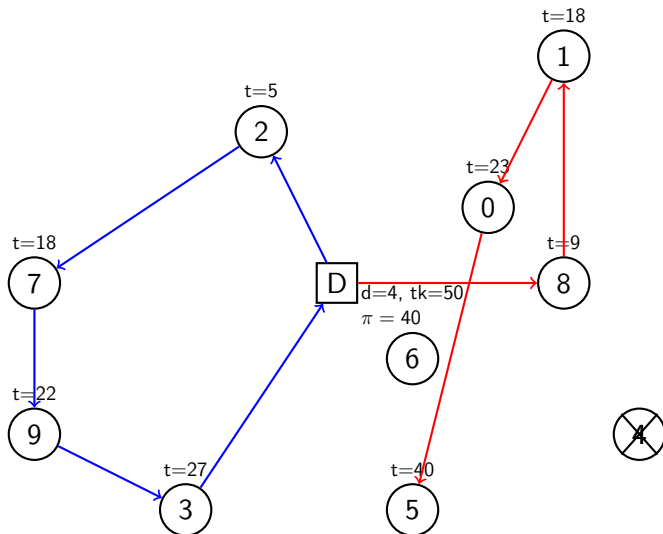
Vozilo 2



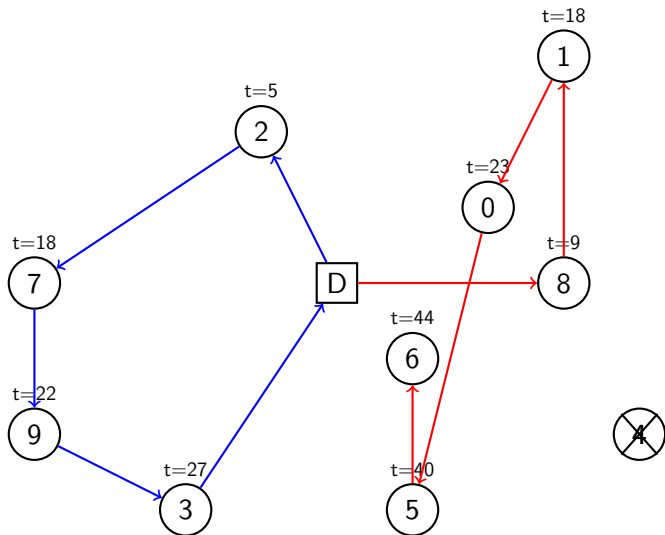
Vozilo 2



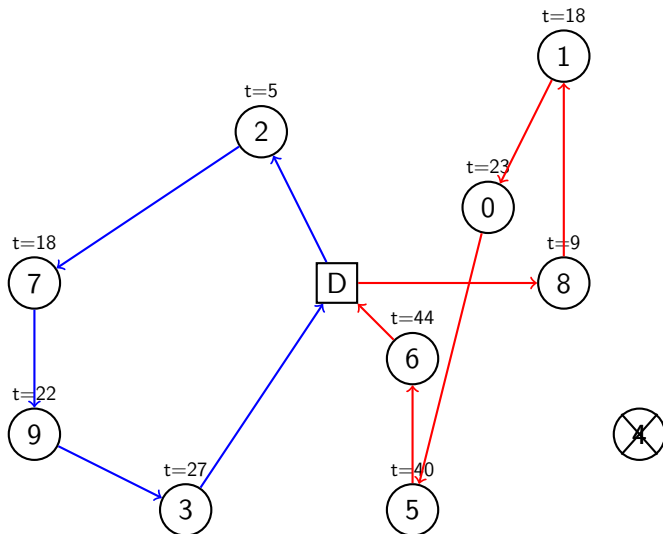
Vozilo 1



Vozilo 1



Vozilo 1



Prednosti

- Mogućnost izrade heuristike za gotovo proizvoljni problem
- S malo truda moguće je dobiti relativno dobre heuristike
- U većini slučajeva postižu bolje rezultate od postojećih heuristika
- Iznimno prikladna metoda za nestandardne probleme i kriterije
- Trenutno dosta istraživano područje – dosta novih metoda i pristupa.

Nedostaci

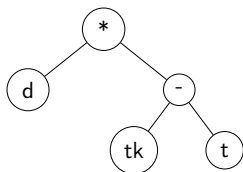
- I dalje nam je potrebno određeno znanje o domeni
- Izrada pravila traje dosta dugo
- Performanse u rangu konstruktivnih heuristika

Napredne teme

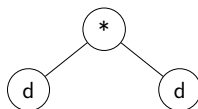
Skupno učenje

- Nemoguće je razviti heuristiku koja u svim situacijama radi dobro
- Očekivano je da će za određene probleme heuristika raditi loše (No free lunch)
- Koristiti grupu heuristika kako bi zajednički donijele odluku o idućoj akciji [7]

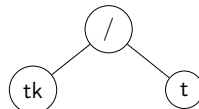
Skupno učenje



$\pi_1 = 15$
 $\pi_2 = 16$
 $\pi_3 = 20$



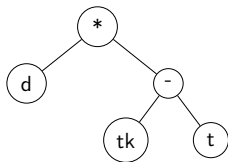
$\pi_1 = 17$
 $\pi_2 = 10$
 $\pi_3 = 13$



$\pi_1 = 7$
 $\pi_2 = 10$
 $\pi_3 = 12$

$\pi_1 = 39$
 $\pi_2 = 36$
 $\pi_3 = 45$

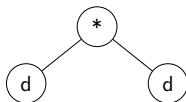
Skupno učenje



$$\begin{aligned}\pi_1 &= 15 \\ \pi_2 &= 16 \\ \pi_3 &= 20\end{aligned}$$



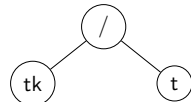
$$\begin{aligned}\pi_1 &= 1 \\ \pi_2 &= 0 \\ \pi_3 &= 0\end{aligned}$$



$$\begin{aligned}\pi_1 &= 17 \\ \pi_2 &= 10 \\ \pi_3 &= 13\end{aligned}$$



$$\begin{aligned}\pi_1 &= 0 \\ \pi_2 &= 1 \\ \pi_3 &= 0\end{aligned}$$



$$\begin{aligned}\pi_1 &= 7 \\ \pi_2 &= 10 \\ \pi_3 &= 12\end{aligned}$$

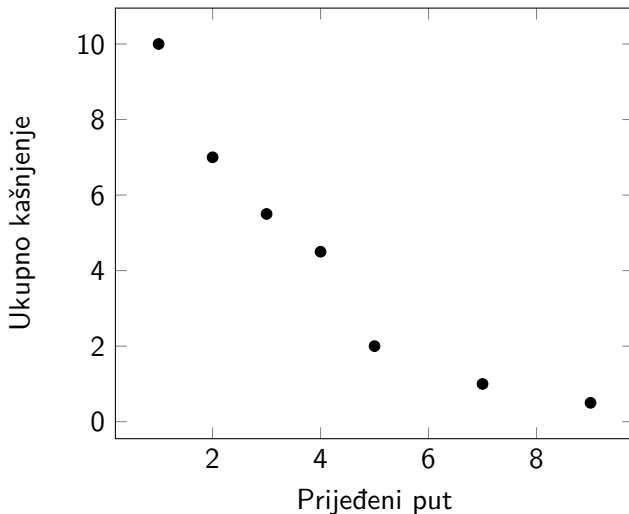


$$\begin{aligned}\pi_1 &= 1 \\ \pi_2 &= 0 \\ \pi_3 &= 0\end{aligned}$$

$$\begin{aligned}\pi_1 &= 2 \\ \pi_2 &= 1 \\ \pi_3 &= 0\end{aligned}$$

Višekriterijska optimizacija

- Lako ostvariva korištenjem višekriterijskih algoritama (NSGA-II) [8]



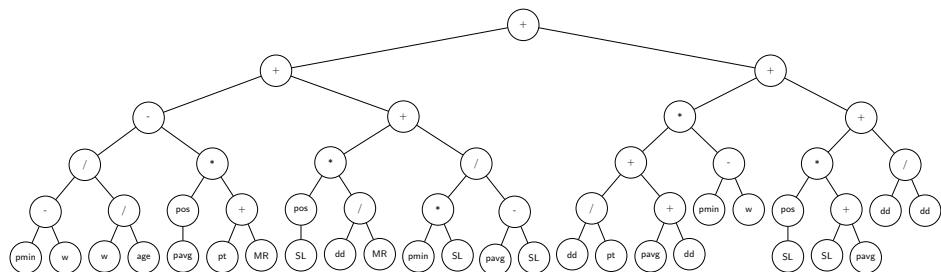
Surogatni modeli

- Razvoj pravila je dugotrajan proces
- Potrebno ih je testirati na velikom skupu problema
- Simulacija često veoma spora
- Rješenje – surogatni modeli [9]:
 - Za evaluaciju rješenja koristiti manje probleme
 - Moraju dobro predstavljati prave probleme
 - Koriste se u svrhu eliminacije loših jedinki

Interpretabilnost

- Prioritetna funkcija predstavljena matematičkim izrazom
- Možemo interpretirati na koji način se donose odluke?

Interpretabilnost



Interpretabilnost

- Rješenja
 - Algebarska ili numerička simplifikacija [10]
 - Uzeti u obzir kompleksnost rješenja prilikom evolucije
 - Alternativni prikazi: gene expression programming ili Kartezijevo genetsko programiranje [11]
 - Semantičko genetsko programiranje [12]
 - $\pi = d - t$

Zaključak

Zaključak

- Relativno nova metodologija
- Dobri rezultati u usporedbi s ručno izrađenim heuristikama
- Potrebno određeno poznavanje domene

Primjene

- Razni problemi raspoređivanja [13] [14]
- Problem trgovačkog putnika [15]
- Problemi usmjeravanja vozila [16]
- Problem relokacije kontejnera [17]
- Problem pakiranja kutija [18]

Otvorena područja

- Primjena na nove probleme, posebice iz stvarnog svijeta
- Poboljšanje performansi
 - Hibridni postupci
 - Kombinacija s metodama strojnog učenja
- Poboljšanje interpretabilnosti

Resursi

- Genetsko programiranje [6]
- Hiperheuristike [5]

Kontakt

marko.durasevic@fer.hr

Hvala na pažnji

Pitanja?

Reference I

- [1] B. Golden, S. Raghavan i E. Wasil, ur., *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, 2008. DOI: 10.1007/978-0-387-77778-8. adresa: <https://doi.org/10.1007/978-0-387-77778-8>.
- [2] P. Sitek i J. Wikarek, "Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach," *Annals of Operations Research*, sv. 273, br. 1-2, str. 257–277, prosinac 2017. DOI: 10.1007/s10479-017-2722-x. adresa: <https://doi.org/10.1007/s10479-017-2722-x>.

Reference II

- [3] H. M. Afsar, S. Afsar i J. J. Palacios, "Vehicle routing problem with zone-based pricing," *Transportation Research Part E: Logistics and Transportation Review*, sv. 152, str. 102-383, 2021., ISSN: 1366-5545. DOI: <https://doi.org/10.1016/j.tre.2021.102383>. adresa: <https://www.sciencedirect.com/science/article/pii/S1366554521001514>.
- [4] T. Erdelić i T. Carić, "A Survey on the Electric Vehicle Routing Problem: Variants and Solution Approaches," *Journal of Advanced Transportation*, sv. 2019, str. 1-48, svibanj 2019. DOI: [10.1155/2019/5075671](https://doi.org/10.1155/2019/5075671). adresa: <https://doi.org/10.1155/2019/5075671>.

Reference III

- [5] N. Pillay i R. Qu, *Hyper-Heuristics: Theory and Applications*. Springer International Publishing, 2018. DOI: 10.1007/978-3-319-96514-7. adresa: <https://doi.org/10.1007/978-3-319-96514-7>.
- [6] R. Poli, W. B. Langdon i N. F. McPhee, *A field guide to genetic programming*. Published via <http://lulu.com> i freely available at <http://www.gp-field-guide.org.uk>, 2008., (With contributions by J. R. Koza). adresa: <http://www.gp-field-guide.org.uk>.
- [7] J. Park, Y. Mei, S. Nguyen, G. Chen i M. Zhang, "An investigation of ensemble combination schemes for genetic programming based hyper-heuristic approaches to dynamic job shop scheduling," *Applied Soft Computing*, sv. 63, str. 72–86, veljača 2018. DOI: 10.1016/j.asoc.2017.11.020. adresa: <https://doi.org/10.1016/j.asoc.2017.11.020>.

Reference IV

- [8] S. Nguyen, M. Zhang i K. C. Tan, “Enhancing genetic programming based hyper-heuristics for dynamic multi-objective job shop scheduling problems,” *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015., str. 2781–2788. DOI: 10.1109/CEC.2015.7257234.
- [9] S. Nguyen, M. Zhang i K. C. Tan, “Surrogate-Assisted Genetic Programming With Simplified Models for Automated Design of Dispatching Rules,” *IEEE Transactions on Cybernetics*, sv. 47, br. 9, str. 2951–2965, rujan 2017. DOI: 10.1109/tcyb.2016.2562674. adresa: <https://doi.org/10.1109/tcyb.2016.2562674>.

Reference V

- [10] L. Planinic, M. Durasevic i D. Jakobovic, "Towards Interpretable Dispatching Rules: Application of Expression Simplification Methods," *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, prosinac 2021. DOI: 10.1109/ssci50451.2021.9659842. adresa: <https://doi.org/10.1109/ssci50451.2021.9659842>.
- [11] L. Planinic, H. Backovic, M. Durasevic i D. Jakobovic, "A comparative study of dispatching rule representations in evolutionary algorithms," *IEEE Access*, str. 1–1, 2022. DOI: 10.1109/access.2022.3151346. adresa: <https://doi.org/10.1109/access.2022.3151346>.

Reference VI

- [12] M. Đurasević, D. Jakobović i K. Knežević, “Adaptive scheduling on unrelated machines with genetic programming,” *Applied Soft Computing*, sv. 48, str. 419–430, studeni 2016. DOI: 10.1016/j.asoc.2016.07.025. adresa: <https://doi.org/10.1016/j.asoc.2016.07.025>.
- [13] J. Branke, T. Hildebrandt i B. Scholz-Reiter, “Hyper-heuristic Evolution of Dispatching Rules: A Comparison of Rule Representations,” *Evolutionary Computation*, sv. 23, br. 2, str. 249–277, lipanj 2015. DOI: 10.1162/evco_a_00131. adresa: https://doi.org/10.1162/evco_a_00131.

Reference VII

- [14] S. Nguyen, “A learning and optimizing system for order acceptance and scheduling,” *The International Journal of Advanced Manufacturing Technology*, sv. 86, br. 5-8, str. 2021–2036, siječanj 2016. DOI: 10.1007/s00170-015-8321-6. adresa: <https://doi.org/10.1007/s00170-015-8321-6>.
- [15] G. Duflo, E. Kieffer, M. R. Brust, G. Danoy i P. Bouvry, “A GP Hyper-Heuristic Approach for Generating TSP Heuristics,” *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019., str. 521–529. DOI: 10.1109/IPDPSW.2019.00094.

Reference VIII

- [16] J. Jacobsen-Grocott, Y. Mei, G. Chen i M. Zhang, “Evolving heuristics for Dynamic Vehicle Routing with Time Windows using genetic programming,” *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017., str. 1948–1955. DOI: 10.1109/CEC.2017.7969539.
- [17] M. Đurasević i M. Đumić, *Automated Design of Heuristics for the Container Relocation Problem*, 2021. DOI: 10.48550/ARXIV.2107.13313. adresa: <https://arxiv.org/abs/2107.13313>.

Reference IX

- [18] E. López-Camacho, H. Terashima-Marin, P. Ross i G. Ochoa, “A unified hyper-heuristic framework for solving bin packing problems,” *Expert Systems with Applications*, sv. 41, br. 15, str. 6876–6889, 2014., ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2014.04.043>. adresa: <https://www.sciencedirect.com/science/article/pii/S0957417414002668>.