Univerzitet u Beogradu

Matematički fakultet

Miloš Šošić

Heuristički pristup rešavanju problema minimalnog kašnjenja koristeći metode promenljivih okolina

master rad

Beograd

2014.

|  |  |
| --- | --- |
| Mentor: | **prof. dr Zorica Stanimirović**  Matematički fakultet u Beogradu |
| Članovi komisije: | **prof. dr Miroslav Marić**  Matematički fakultet u Beogradu  **prof. dr Miodrag Živković**  Matematički fakultet u Beogradu |
| Datum odbrane: |  |

Heuristički pristup rešavanju problema minimalnog kašnjenja koristeći metode promenljivih okolina

**Apstrakt:**

Problem minimalnog kašnjenja predstavlja varijaciju problema trgovačkog putnika, gde je cilj minimizovati sumu vremena potrebnog da se poseti svaki od čvorova, tj. sumu kašnjenja. Ovaj problem ima široku praktičnu primenu, koja uključuje distribuciju raznih dobara, pravljenje rasporeda glave diska itd. U ovom radu izložena metoda promenljivih okolina hibridizovana sa metodom simuliranog kaljenja za rešavanje problema minimalnog kašnjenja. Eksperimentalni rezultati pokazuju da ...

apstrakt engleski

Sadržaj

[1 Uvod 8](#_Toc398929194)

[2 Metoda promenljivih okolina 9](#_Toc398929195)

[3 Simulirano kaljenje 13](#_Toc398929196)

[4 Problem minimalnog kašnjenja 15](#_Toc398929197)

[4.1 Matematička formulacija 15](#_Toc398929198)

[4.2 Primer sa slikom, izračunavanje funkcije cilja 17](#_Toc398929199)

[4.3 Prethodna rešavanja 18](#_Toc398929200)

[5 Hibridini algoritam za problem minimalnog kašnjenja 19](#_Toc398929201)

[5.1 Kodiranje i prostor rešenja 20](#_Toc398929202)

[5.2 Konstrukcija početnog rešenja 20](#_Toc398929203)

[5.3 Strukture okoline 20](#_Toc398929204)

[5.4 20](#_Toc398929205)

[6 Rezultati 21](#_Toc398929206)

[6.1 Instance 21](#_Toc398929207)

[6.2 Poređenje sa TSP-om 21](#_Toc398929208)

[7 Zaključak 22](#_Toc398929209)

[Literatura 23](#_Toc398929210)

# Uvod

diskretna optimizacija, heuristike – hansen, mladenovic, moreno perez, 2008 ili H.o.M.ed2

tsp problem

# Metoda promenljivih okolina

Metode kombinatorne optimizacije koje koriste lokalnu pretragu dolaze do boljih rešenja iterativno primenjujući lokalne promene nad trenutno najboljim rešenjem sve dok se ne dostigne lokalni optimum. Ove lokalne promene pripadaju okolini trenutnog rešenja . Kod ovakvih metoda može se doći do stanja kada algoritam ostane u nekom lokalnom i ne dostigne globalni optimum. Ovaj problem se tada rešava na različite načine mehanizmom koji se naziva perturbacija.

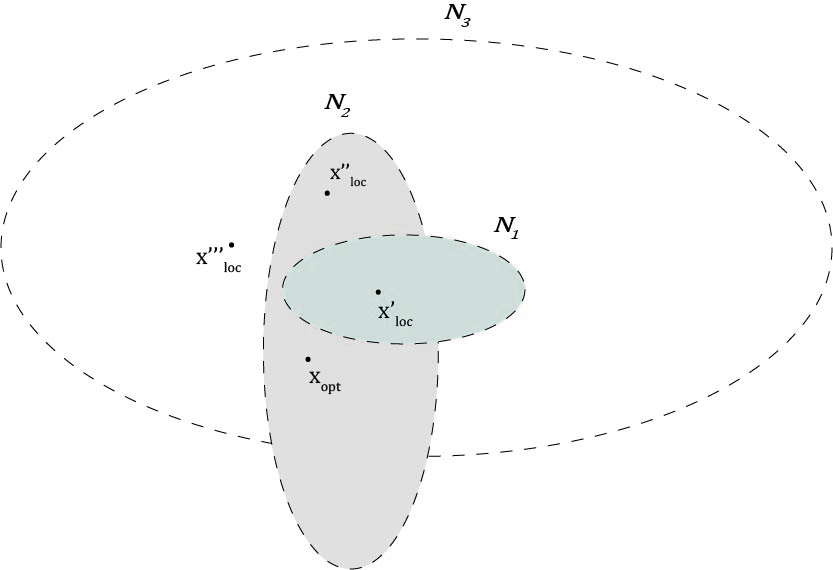
Metoda promenljivih okolina (VNS[[1]](#footnote-1)) je metaheuristika predstavljena u [1], koja se bazira na metodama lokalne pretrage, gde se koristi sistematska promena strukture okolina kako bi se poboljšalo trenutno rešenje u procesu lokalne pretrage i perturbacija. Ova metoda je osmišljena za približno rešavanje problema kombinatorne optimizacije ali je vremenom proširena i korisiti se pri rešavanju problema celobrojnog, mešovitog celobrojnog, nelinearnog programiranja i drugih. Stoga, neke od oblasti primene VNS metode su lokacijski problemi, problemi raspoređivanja i uparivanja, problemi rutiranja vozila, računarske mreže, i mnogi drugi (za detaljniji pregled oblasti primene, pogledati [2]). Sledi detaljniji opis VNS metaheuristike ( [2] ).

Neka je dati prostor rešenja i fukcija cilja za dati problem. Neka su pažljivo odabrane strukture okolina a skup rešenja iz -te okoline rešenja , gde . Označimo optimalno rešenje sa (globalni minimum). U okolini je lokalni minimum ako ne postoji tako da važi . Da bi različite okoline bile dobro definisane, treba da važi sledeće:

1. Lokalni minimum jedne okoline ne mora biti lokalni minimum druge okoline
2. Globalni minimum je lokalni minimum svih mogućih okolina
3. U mnogim problemima važi, lokalni minimumi različitih okolina su relativno blizu.

Poslednja pretpostavka podrazumeva da lokalni minimum često pruža korisne informacije o globalnom minimumu. Na primer, može se desiti da neke od promenljivih uzimaju iste vrednosti u oba rešenja ali kako se ne može utvrditi koje su te promenljive u toku izvršavanja algoritma, primenjuje se pažljiva pretraga okoline lokalnog minimuma u potrazi za boljim rešenjem. Primer struktura okolina može se videti na slici 1.

Osnovni VNS algoritam sastoji se iz dve faze koje se izvršavaju u svakoj iteraciji. Prva faza je „razmrdavanje“ gde se pravi slučajni potez u trenutnoj okolini nakon čega sledi druga faza, lokalna pretraga koja nalazi lokalni minimum za tekuću okolinu.

Slika 1. *Primer struktura okolina gde predstavljaju okoline, respektivno njihove lokalne optimume a globalni optimum.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  | | | |
|  |  | | | |
|  | |  | |
|  | |  | |
|  | | |  |
|  | | |  |
|  | | |  |
|  |  | | |

Početno rešenje može biti bilo koje dopustivo rešenje, obično dobijeno nekim pohlepnim ili slučajnim algoritmom. Kriterijum zaustavljanja može biti utrošeno procesorsko vreme, maksimalni broj iteracija, maksimalni broj iteracija bez poboljšanja rešenja itd. Lokalna pretraga je u ovom algoritmu može podrazumevati pretragu za najboljim rešenjem u trenutnoj okolini (eng. *best-improvement*) ili pretragu za prvim boljim rešenjem (eng. *first-improvement)*. Umesto lokalne pretrage može se koristiti i bilo koja druga s-heuristika.

Jedna značajna procedura lokalne pretrage sa promenljivim okolinama je metoda promenljivog spusta (VND[[2]](#footnote-2)) data sledećim algoritmom.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  | | | |
|  |  | | | |
|  |  | | | |
|  | |  | |
|  | |  |
|  |  | |

Ako se ova metoda koristi u drugoj fazi algoritma VNS, dobija se generalizovani VNS (GVNS[[3]](#footnote-3)) koji se za neke probleme istakao kao najefikasnija metoda za približno rešavanje.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  |  | | | |
|  |  | | | |
|  | |  | |
|  | |  | |
|  | | |  |
|  | | |  |
|  | | |  |
|  |  | | |

gde se u proceduri VND koriste iste strukture okolina kao iz dela inicijalizacije algoritma GVNS.

# Simulirano kaljenje

Simulirano kaljenje (SA – eng. *Simulated Annealing*) je veoma proučavana metaheuristika koja pripada grupi metoda lokalne pretrage. Koristi se za pri rešavanju diskretnih i u manjoj meri kontinualnih optimizacionih problema. Glavna odlika ove metode je to što dozvoljava rešenju da postane lošije kako bi se izvršila perturbacija i izbegla prerana konvergencija algoritma ka nekom lokalnom optimumu.

Motivacija za ovu metaheuristiku je proistekla iz procesa kaljenja metala u metalurgiji. Metal se zagreva i kontrolisano hladi kako bi se dostigla najbolja konfiguracija kristala i izbegli defekti kristala. Na ovaj način, ako je pogodno izabran način hlađenja, metal koji se na kraju dobije ima poboljšanu strukturu veza među atomima i povoljne osobine za dalju obradu.

Za dati problem diskretne optimizacije, neka je prostor rešenja, funkcija cilja i neka je sa definisana okolina tačke . Treba pronaći tako da važi za svako . Simulirano kaljenje u svakoj iteraciji upoređuje dva rešenja, trenutno, , i rešenje iz njegove okoline, , dobijeno na slučajan način. Ukoliko je bolje rešenje, trenutno rešenje postaje , a ako je lošije, se prihvata sa određenom verovatnoćom koja zavisi od parametra algoritma . Ovaj parametar predstavlja trenutnu temperaturu procesa kaljenja i smanjuje se u toku algoritma prema unapred određenoj šemi hlađenja koja podrazumeva skup temperatura . Za svako , bira se broj iteracija , na toj temperaturi. Osnovni deo algoritma je verovatnoća prihvatanja rešenja koja se izračunava na sledeći način:

|  |  |
| --- | --- |
|  | (4.1) |

gde je događaj prihvatanja rešenja .

Algoritam simuliranog kaljenja:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  |  | | | |
|  |  | | | |
|  |  | | | |
|  | |  | | | |
|  | |  | | | |
|  | | |  | |
|  | | |  | |
|  | | |  | |
|  | | |  | |
|  | |  | | | |
|  |  | | |

gde kriterijum zaustavljanja može biti kao i kod VNS algoritma.

Pri većoj temperatruti veća je i verovatnoća prihvatanja lošijeg rešenja a kako temperatura opada, ta verovatnoća postaje sve manja. Na taj način se na početku izbegava prerana konvergencija metode i omogućava pretraga različitih lanaca rešenja dok se na kraju nalazi lokalni minimum prihvatanjem isključivo boljih rešenja. Uspeh metode u velikoj meri zavisi od šeme hlađenja pa ovu šemu treba pažljivo konstruisati.

Više o simuliranom kaljenju i dokazu konvergencije metode ka globalnom rešenju može se naći u [3].

# Problem minimalnog kašnjenja

(problemi rutiranja?)

Neka je kompletan graf, gde je skup čvorova, tj. lokacija do kojih treba da se stigne, a skup grana grafa , gde je uz svaku granu poznato vremene potrebno za put između dva čvora. Postoji jedan istaknuti čvor (obično čvor ) koji predstavlja početni čvor sa kojeg obilazak počinje. Neka je kašnjenje do -tog čvora koje se izračunava kao potrebno vreme da se stigne od istaknutog do čvora . Cilj problema minimalnog kašnjenja (MLP[[4]](#footnote-4)) je pronaći Hamiltonov put koji minimizuje sumu . U ovom radu, svaki obilazak grafa počinje sa čvorom i završava se kada obiđe ostalih čvorova. MLP je u literaturi poznat i po drugim imenima (eng. Traveling Repairman Problem, Delivery Man Problem, Cumulative Traveling Salesman Problem, School Bus Driver Problem).

MLP ima široku primenu u praksi, posebno u distribuciji robe korisnicima, logistici u kriznim situacijama, raspoređivanju poslova itd. Kod problema putujućeg majstora, poznate su lokacije klijenata i majstora, kao i vremena puta između klijenata i potrebna vremena za servisiranje svakog klijenta. Potrebno je pronaći put kojim majstor obilazi klijente tako da je njihovo ukupno vreme čekanja minimalno. Sličan je problem kurira, gde kurir treba da pronađe put koji minimizuje ukupno vreme čekanja na dostavu pošiljke. Ovi problemi su klijentski orjentisani problemi rutiranja jer funkcija cilja daje prednost minimizaciji vremena čekanja klijenata u odnosu na dužinu puta vozila.

Kod raspoređivanja poslova na mašini[[5]](#footnote-5) javlja se ovaj problem u sledećem obliku. Postoji skup poslova koje mašina treba da obavi kao i vreme koje je potrebno da se mašina ponovo podesi da radi posao nakon završenog posla . Ovo vreme predstavlja vreme puta od čvora do čvora . Treba pronaći permutaciju zadatih poslova tako da se minimizuje potrebno vreme za završetak svih poslova. [4]

## Matematička formulacija

Neka je kompletan graf kao u prethodnom delu i neka je dat Hamiltonov put, tj. permutacija čvorova koja predstavlja neko rešenje problema. Neka je nenegativna matrica troškova gde odgovara trošku grane od čvora do čvora . Kašnjenje , koje odgovara čvoru na -toj poziciji u datom putu, može se izračunati kao , gde i . Funkcija cilja je predstavljena sumom , a kako važi

tako je

Kod problema kurira i putujućeg majstora, vrednost čine vreme puta i vreme servisiranja , tj.:

U ovom radu, rešava se problem gde je matrica simetrična.

U svakoj permutaciji čvorova, Hamiltonov put počinje od čvora i obilazi ostale čvorove po redu u kojem je data permutacija. Ako grana spaja -tog sa -im članom permutacije, tada je doprinos te grane funkciji cilja .

Uvodimo sledeće promenljive odlučivanja:

Promenljive se koriste pri izračunavanju kašnjenja od čvora do pozicije , tj. čvora , a promenljive za izračunavanje kašnjenja od čvora koji se nalazi na poziciji do čvora koji se nalazi na poziciji .

Sledi celobrojna linearna formulacija problema minimalnog kašnjenja na osnovu [4]:

|  |  |
| --- | --- |
|  | (2.1) |

pri ograničenjima:

|  |  |
| --- | --- |
|  | (2.2) |
|  | (2.3) |
|  | (2.4) |
|  | (2.5) |
|  | (2.6) |
|  | (2.7) |

Ograničenje (2) predstavlja uslov da se svaki čvor nalazi na tačno jednoj poziciji u permutaciji, (3) je uslov da se na svakoj poziciji nalazi tačno jedan čvor, (4) ograničava da sa svake pozicije vodi samo jedna grana ka sledećem čvoru a (5) da na svaku poziciju samo jedna grana vodi sa prethodnog. Ograničenja (6) i (7) su binarna ograničenja promenljivih odlučivanja.

## Primer sa slikom, izračunavanje funkcije cilja

primer + slika 2

## Prethodna rešavanja

Kako problem minimalnog kašnjenja pripada klasi NP-teških problema [5], tačni rešavači su primenjivi samo na instancama određene velične. Prvi takav algoritam [6], bio je enumeracioni, zasnovan na formulaciji nelinearnog celobrojnog programiranja koristeći Lagranžovu relaksaciju za dobijanje donjih ograničenja. U ovom radu rešene su instance sa manje od čvorova. Sledi [7] u kojem je predložena formulacija celobrojnog programiranja a algoritam za rešavanje je grananje sa ograničavanjem (eng. branch-and-bound) koristeći matroidne strukture problema za dobijanje donjih ograničenja. Rešene su instance do 60 čvorova. Kasnije, u [8] je izložena formulacija celobrojnog programiranja koja koristi prednosti veze sa problemom linearnog poretka a predstavljen je algoritam odsecanja ravni koji koristi validne nejednakosti u toj formulaciji. Dva rada, [9], [10], predstavila su branch-cut-and-price pristup kojim su rešene instance do čvorova. Ovo je trenutno jedini tačni algoritam koji rešava instance ovih dimenzija.

Problem minimalnog kašnjenja je rešavan i približnim algoritmima. Prvi ovakav algoritam [11] imao je aproksimacioni faktor 144. Trenutno najbolji približni algoritam [12] ima aproksimacioni faktor 3.59.

Ne postoji puno radova u kojima je MLP rešavan pomoću metaheuristika. Jedan od njih je [13], gde je korišćena metaheuristika koja spaja GRASP metodu i metodu promenljivih okolina. Za problem minimalnog kašnjenja sa profitom, razvijena je tabu pretraga u [14]. Varijacija problema rutiranja vozila [15] može da se prilagodi tako da rešava MLP. U tom radu, opisan je memetski algoritam kao i efikasna procedura provere novog poteza u okolini sa operacija. Najznačajniji rad u kojem se koristi metaheuristika za rešavanje MLP-a je [16] u kojem se spajaju metoda GRASP, iterativna lokalna pretraga i varijacija metode promenljivih okolina. Metaheuristika predložena u [16] dostiže optimalna rešenja za instance do čvorova gde su optimalna rešenja poznata, a za veće dimenzije postiže najbolja rešenja trenutno poznata.

U nekim od pomenutih radova ( [16] ), umesto Hamiltonovog puta treba pronaći Hamiltonov ciklus, tj. posle obilaska svih čvorova odlazi se do čvora .

# Hibridini algoritam za problem minimalnog kašnjenja

Ideja hibridnih metaheuristika postoji već duže vremena iako je termin „hibridizacija“ prihvaćen tek u poslednjih desetak godina ( [17]). Pod ovim terminom podrazumeva se kombinovanje različitih metaheuristka u složeniji algoritam. Uspešna hibridizacija predstavlja spoj dobrih osobina metaheuristika, koje se koriste u algoritmu, kako bi se postigla kvalitetnija rešenja za dati problem ili efikasniji algoritam.

Za problem minimalnog kašnjenja hibridni algoritam koristi metodu promenljivih okolina kao osnovni algoritam sa izmenjenom fazom lokalne pretrage. Ova faza se sastoji iz kombinovanja metode promenljivog spusta sa metodom simuliranog kaljenja. Ideja je da u početku dođe do dobrog rešenja koristeći metodu promenljivog spusta kao lokalnu pretragu a onda umesto nje se koristi metoda simuliranog kaljenja da bi se došlo do kvalitetnijeg rešenja. Algoritam je dat pseudokodom:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
|  | , | | | |
|  |  | | | |
|  |  | | | |
|  | |  | |
|  | |  | |
|  | | |  |
|  | | |  |
|  | | |  |
|  | |  | |
|  |  | | |

U ovom algoritmu se pri pozivanju procedure ne konstruiše iznova početno rešenje već se za početno uzima rešenje

Parametar zavisi od date instance i iznosi . Parametar α predstavlja broj iteracija posle kojeg treba započeti primenjivanje simuliranog kaljenja umesto metde VND. Empirijski je zaključeno da dobra vrednost za ∝ iznosi . Ova vrednost donosi dobar najbolji odnos kvaliteta rešenja i brzine izvršavanja.

## Kodiranje i prostor rešenja

Rešenje za MLP je Hamiltonov put pa se rešenje prirodno kodira nizom brojeva. Ovi brojevi predstavljaju indekse čvorova u tom putu a njihov redosled odgovara redosledu obilaska čvorova. Prostor rešenja je skup svih permutacija brojeva iz skupa koje počinju brojem (jer je prvi čvor fiksiran). Primer jednog rešenja za instancu MLP-a gde važi dat je na slici 3.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 7 | 8 | 5 | 1 | 3 | 9 | 2 | 10 | 6 | 4 |

Slika 3. *Primer rešenja za instancu gde*

Notacija za predstavljanje rute rešenja biće, za primer sa slike 3, .

## Konstrukcija početnog rešenja

Početno rešenje u fazi inicijalizacije algoritma VNS je pohlepni algoritam koji konstruiše rutu čvor po čvor. Počinje se od čvora i svaki naredni čvor je onaj koji ima najkraći put od poslednjeg čvora u trenutnom putu. Ovaj algoritam je dat sledećim pseudokodom:

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  | |
|  |  | |
|  |  | |
|  | |  |
|  | |  |
|  |  | |

Ovaj algoritam daje bolje početno rešenje od izbora rute na slučajan način pa se zato koristi u ovoj hibridizaciji.

## Strukture okoline

svaka okolina detaljno

složenost

## Dinamičko računanje funkije cilja

složenosti

funkcija cilja

dinamičko računanje funkcije cilja, strukture, složenosti

# Rezultati

rešavano na Intel ...

pisano u c++ visual studio

## Instance

tsplib instance

način zaokruživanja

random instance Sallyhapoor

## Poređenje sa TSP-om

tabela iz jednog rada

# Zaključak

pogledati neki word stil gotov

# Literatura

1. **Mladenovic N., Hansen E.** Variable neighborhood search. *Computers & Operations Research.* 1997, T. 24, 11, str. 1097-1100.

2. **Hansen, P., Mladenovic, N., Moreno Perez, J.A.** Variable neighborhood search: methods and applications. *4OR-Q J Oper. Res.* 2008, T. 6, str. 319-360.

3. **Glover F., Kochenberger G.A.** *Handbook of Metaheuristics.* [ur.] Potvin J.-Y. Gendreau M. 2nd. s.l. : Springer, 2010. T. 146.

4. **Francisco Angel-Bello Acosta, Ada Alvarez Socarras, Irma Garcia.** *Formulation for the minimum latency problem: an experimental evaluation.* 2011.

5. **Sahni, S., Gonzalez, T.** P-complete approximation problems. *Journal of the AXM.* 23, 1976, T. 3, str. 555-565.

6. **Lucena, A.** Time-dependent traveling salesman problem - the deliveryman case. *Networks.* 20, 1990, str. 753-763.

7. **Fischetti M., Laporte G., Martello S.** The delivery man problem and cumulative matroids. *Operations Research.* 41, 1993, T. 6, 1055-1064.

8. **Mendez-Diaz P., Zabala P.,Lucena A.** A new formulation for the traveling deliveryman problem. *Discrete Appled Math.* 156, 2008, str. 3223-3237.

9. **Abelado H., Fukasawa R., Pessoa A. Uchoa E.** The time dependent traveling salesman problem: Polyhedra and algorithm. *Tech. Rep. RPEP.* 10, 2010, T. 15.

10. **Abeledo H., Fukasawa R., Pessoa A., Uchoa E.** The time dependent traveling salesman problem: Polyhedra and branch-cut-and-price algorithm. *Proceedings of the 9th International Symposium on Experimental Algorithms.* 2010, str. 202-2013.

11. **Blum A., Chalasanit P., Pulleyblankt B., Raghavan P., Sudan M.** The minimum latency problem. *Proceedings of the 26th Annual ACM Symposium on Theory of Computing.* 1994, str. 163–171.

12. **Chaudhuri K., Godfrey B., Rao S., Talwar K.** Paths, trees, and minimum latency tours. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science.* 2003, str. 36-45.

13. **Salehipour, A., Sörensen, K., Goos, P., Bräysy, O.** Efficient GRASP + VND and GRASP + VNS metaheuristics for the traveling repairman problem. *A Quarterly Journal of Operations Research.* 9, 2011, T. 2, str. 189-209.

14. **Dewilde, T., Cattrysse, D., Coene, S., Spieksma, F.C.R., Vansteenwegen, P.** Heuristics for the traveling repairman problem with profits. *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2010.* 2010, str. 34-44.

15. **Ngueveu, S., Prins, C., Wolfler Calvo, R.** An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research.* 2010, T. 37, 11, str. 1877–1885.

16. **Melo Silva M., Subramanian A., Vidal T., Satoru Ochi L.** A simple and effective metaheuristic for the Minimum Latency Problem. *European Journal of Operational Research.* 2012, T. 221, str. 513-520.

17. **Blum C., Jose Blesa Aguilera M., Roli A., Sampels M.** Hybrid Metaheuristics: An Emerging Approach to Optimization. s.l. : Springer, 2008.

1. *eng. Variable Neighborhood Search* [↑](#footnote-ref-1)
2. eng. *Variable neighborhood descent* [↑](#footnote-ref-2)
3. eng. *General VNS* [↑](#footnote-ref-3)
4. *eng. Minimum latency problem* [↑](#footnote-ref-4)
5. *eng. Machine scheduling context* [↑](#footnote-ref-5)