

Metahirističke metode 6. nedelja

Metoda promenljivih okolina (Variable Neighborhood Search)

- S heuristika zasnovana na lokalnom pretraživanju (jedna od najefikasnijih)
- Osnovna ideja je sistematska promena unapred definisanih okolina čime se izbegava zaglavljivanje pretrage u lokalnom optimumu
- Oslanja se na 3 činjenice:
 - 1 Lokalni optimum u odnosu na jednu okolinu ne mora biti lokalni optimum u odnosu na drugu.
 - 2 Globalni optimum je lokalni optimum u odnosu na sve okoline.
 - 3 Za većinu problema lokalni optimumi u odnosu na različite okoline su relativno blizu.

Metoda promenljivog spusta (Variable Neighborhood Descent)

- Deterministička varijanta VNS-a
- Cilj VND metode je nalaženje rešenje koje će biti lokalni optimum u odnosu na sve okoline iz unapred definisanog skupa okolina
- Inicijalizacija: odabir okolina N_k , $k = 1, \dots, k_{max}$ i generisanje početnog rešenja koje postaje tekuće rešenje x^*
- U svakoj VND iteraciji, nalazi se najbolji sused iz N_k okoline tekućeg rešenja x^* . Ukoliko novo rešenje x'' ne dovedi do poboljšanja (u odnosu na vrednost funkcije cilja), algoritam nastavlja sa pretraživanjem N_{k+1} okoline nepromenjenog tekućeg rešenja x^* . U suprotnom, rešenje x' postaje tekuće ($x^* = x'$) i pretraživanje se nastavlja u okolini N_1 tekućeg rešenja.
- Algoritam se završava ukoliko se ne naidje na poboljšanje nakon pretraživanja svih okolina tekućeg rešenja.

VND Algoritam

Algoritam 1 VND

- 1: Inicijalizacija: izabrati okoline N_k , $k = 1, 2, \dots, k_{max}$ i parametar k_{max}
 - 2: Generisati početno rešenje x^* , $f^* = f(x^*)$..
 - 3: Set $k = 1$
 - 4: **while** $k \leq k_{max}$ **do**
 - 5: Naći najbolje rešenje x' iz okoline $N_k(x^*)$
 - 6: **if** $f(x') < f^*$ **then**
 - 7: $x^* = x'$ and $k = 1$,
 - 8: **else**
 - 9: $k = k + 1$.
-

Redukovana metoda promenljivih okolina (Reduced Variable Neighborhood Search)

- Stohastička varijanta VNS-a
- Karakteriše je odsustvo lokalnog pretraživanja i slučajan odabir rešenja iz trenutne okoline tekućeg rešenja - dobija se na vremenu, ali gubi na kvalitetu dobijenog rešenja.
- Inicijalizacija: odabir okolina N_k , $k = 1, \dots, k_{max}$ i generisanje početnog rešenja koje postaje tekuće rešenje x^* i izbor kriterijuma zaustavljenja.
- U svakoj iteraciji nasumično se bira sused x' iz $N_k(x^*)$. Ukoliko novo rešenje x' ne dovedi do poboljšanja (u odnosu na vrednost funkcije cilja), algoritam nastavlja sa pretragom N_{k+1} okoline nepromenjenog tekućeg rešenja x^* . Inače, rešenje x' postaje tekuće ($x^* = x'$) i pretraga se nastavlja u okolini N_1 tekućeg rešenja.
- Kada k postane veće od k_{max} , njegova vrednost se postavlja na 1 i opisani postupak se ponavlja do zadovoljenja kriterijuma zaustavljanja (maksimalan broj iteracija, maksimalan broj iteracija između dva poboljšanja, maksimalno vreme izvršavanja itd)

RVNS Algoritam

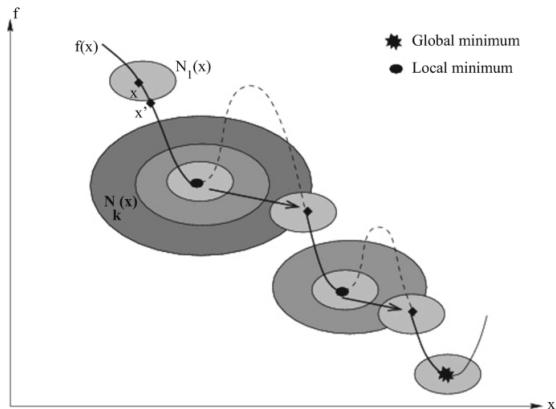
Algoritam 2 RVNS

- 1: Inicijalizacija: izabrati okoline N_k , $k = 1, 2, \dots, k_{max}$, parametar k_{max} i odabrati kriterijum zaustavljanja
 - 2: Generisati početno rešenje x^* , $f^* = f(x^*)$..
 - 3: **while** nije ispunjen kriterijum zaustavljanja **do**
 - 4: Set $k = 1$
 - 5: **while** $k \leq k_{max}$ **do**
 - 6: Naći nasumično rešenje x' iz okoline $N_k(x^*)$
 - 7: **if** $f(x') < f^*$ **then**
 - 8: $x^* = x'$ and $k = 1$,
 - 9: **else**
 - 10: $k = k + 1$.
-

Modifikacija RVNS metode: slučajno se bira m rešenja iz okoline tekućeg rešenja od kojih se bira ono sa najboljom vrednošću funkcije cilja (pokušaj da se poboljša kvalitet rešenja u svakoj iteraciji).

Osnovna metoda promenljivih okolina (Basic Variable Neighborhood Search)

- Najčešće korišćena varijanta: kombinuje deterministički i stohastički pristup (tako se doprinosi balansu između diverzifikacije i intenzifikacije procesa pretrage)
- Dve osnovne faze: faza razmrđavanja i faza lokalnog pretraživanja



- Inicijalizacija: odabir okolina za fazu razmrdavanja i fazu lokalne pretrage (ne moraju biti iste), generisanje početnog rešenja koje postaje tekuće rešenje x^* i izbor kriterijuma zaustavljenja.
- Svaka iteracija počinje fazom razmrdavanja u kojoj se vrši slučajan odabir rešenja x' iz okoline $N_k(x^*)$. Potom se, počevši od x' , primenjuje neka metoda lokalne pretrage unapred definisane okoline u cilju nalaženja boljeg rešenja
- Rezultat lokalne pretrage x'' se poredi sa tekućim rešenjem (korak pomeraja). Ukoliko je x'' dovodi do boljšanja vrednosti funkcije cilja, x'' postaje tekuće rešenje ($x^* = x''$), k se postavlja na 1 i proces pretrage se nastavlja fazom razmrdavanja za okolinu $N_1(x^*)$.
- U suprotnom, algoritam nastavlja pretraživanje u sledećoj okolini N_{k+1} nepromenjenog tekućeg rešenja.
- Kada k postane veće od k_{max} , njegova vrednost se postavlja na 1 i opisani postupak se ponavlja do zadovoljenja kriterijuma zaustavljanja.

Algoritam 3 BVNS

- 1: Initialization: izabrati okoline za fazu razmrđavanja i fazu lokalne pretrage, ako i kriterijum zaustavljanja.
 - 2: Generisati početno rešenje x^* , $f^* = f(x^*)$.
 - 3: **while** kriterijum zaustavljanja nije ispunjen **do**
 - 4: Set $k = 1$
 - 5: **while** $k \leq k_{max}$ **do**
 - 6: Na slučajan način izabrati tačku x' iz okoline $N_k(x^*)$ (**Shaking**)
 - 7: Naći x'' kao rezultat lokalne pretrage počevši od rešenja x'
 (**Local search**)
 - 8: **if** $f(x'') < f^*$ **then**
 - 9: $x^* = x''$ and $k = 1$,
 - 10: **else** $k = k + 1$.
-

Opšta metoda promenljivih okolina (General Variable Neighborhood Search)

Ukoliko se u BVNS metodi lokalna pretraga zameni metodom promenljivog spusta, dobija se GVNS. VND može raditi nad posebnim skupom okolina N_l , $l = 1, 2, \dots, l_{max}$, a može i pretraživati iste okoline koje se koriste u razmrdavanju.

Metoda promenljivih okolina sa dekompozicijom (Variable Neighborhood Decomposition Search)

- Razbija polazni problem na manje potprobleme, koji su lakši za rešavanje, a zatim kombinovanjem njihovih rešenja dobija rešenje polaznog problema.
- Glavna razlika izmedju VNDS i BVNS algoritama je u tome što umesto primene lokalne pretrage u čitavoj okolini tekućeg rešenja, VNDS u svakoj iteraciji rešava potproblem u nekom potprostoru $V_k \subset N_k(x^*)$ čime se smanjuje vreme trajanja lokalne pretrage.
- Inicijalizacija i faza razmrđavanja iste su kao kod BVNS metode
- Neka je sa y označen skup svih atributa (elementa rešenja) koja karakterišu rešenje x' dobijeno u fazi razmrđavanja, ali ne i rešenje x^* , $y = x' \setminus x^*$. Lokalnom pretragom dozvoljava se jedino promena atributa iz skupa y , tj. zabranjuje se izmena skupa $x^* \cap x' = x' \setminus y$. Ukoliko lokalna pretraga naidje na bolji deo rešenja y' , onda skup atributa $(x' \setminus y) \cup y'$ odredjuje novo rešenje x'' . Kriterijum prihvatanja novog rešenja i ažuriranja parametra k isti je kao u BVNS metodi.

Algoritam 4 VNDS

- 1: Initialization: izabrati okoline za fazu razmrđavanja i fazu lokalne pretrage, kao i kriterijum zaustavljanja.
 - 2: Generisati početno rešenje x^* , $f^* = f(x^*)$.
 - 3: **while** kriterijum zaustavljanja nije ispunjen **do**
 - 4: Set $k = 1$
 - 5: **while** $k \leq k_{max}$ **do**
 - 6: Na slučajan način izabrati tačku x' iz okoline $N_k(x^*)$ (**Shaking**)
 - 7: $y = x' \setminus x^*$
 - 8: $y' = \text{lokalna pretraga}(y)$
 - 9: $x'' = (x' \setminus y) \cup y'$
 - 10: **if** $f(x'') < f^*$ **then**
 - 11: $x^* = x''$ and $k = 1$,
 - 12: **else** $k = k + 1$.
-

Iskošana metoda promenljivih okolina (Skewed VNS)

- Dozvoljava se prihvatanje nekih lošijih rešenja sa ciljem pretraživanja delova prostora koji se nalaze daleko od tekućeg rešenja.
- Pored adekvatnog izbora okolina za fazu razmrđavanja $N_k, k = 1, 2, \dots, k_{max}$ i generisanja početnog rešenja koje postaje tekuće i trenutno najbolje, potrebno je definisati rastojanje izmedju dva rešenja kao i vrednost parametra metode α .
- Kriterijum prelaska iz jednog rešenja u drugo, pored vrednosti funkcije cilja tih rešenja i njihove udaljenosti, zavisi i od parametra α .
- U svakoj iteraciji se nakon faze razmrđavanja i faze lokalne pretrage dobijeni lokalni optimum x'' poredi sa najboljim posećenim rešenjem koje se ažurira ukoliko je došlo do poboljšanja.
- Dalja pretraga se recentrira ukoliko je vrednost funkcije cilja rešenja x'' malo lošija od x^* i x'' je dovoljno daleko od x . U tom slučaju x'' postaje novo tekuće rešenje, k se postavlja na 1 i nastavlja se sa pretraživanjem okoline za razmrđavanje $N_1(x^*)$. U suprotnom, algoritam nastavlja pretraživanje u sledećoj okolini N_{k+1} nepromenjenog tekućeg rešenja.

Algoritam 5 SVNS

- 1: Inicijalizacija: Izaberi okoline za fazu razmrđavanja N_k , $k = \overline{1, k_{max}}$ i fazu lokalne pretrage, generisati početno rešenje x^* , $f^* = f(x^*)$ i izaberi parametar α
 - 2: $x = x^*$, $F = f^*$
 - 3: **while** kriterijum zaustavljanja nije ispunjen **do**
 - 4: $k = 1$
 - 5: **while** $k \leq k_{max}$ **do**
 - 6: $x_{shake} = Shaking(x, k)$ ▷ Shaking step
 - 7: $x'' = LS(x_{shake})$ ▷ Local Search step
 - 8: **if** $f(x'') < f^*$ **then**
 - 9: $x^* = x''$, $f^* = f(x'')$
 - 10: **if** $f(x'') < F + \alpha \rho(x, x'')$ **then** ▷ Move or Not step
 - 11: $x = x''$, $F = f(x'')$, $k = 1$
 - 12: **else**
 - 13: $k = k + 1$
-

Primena varijanti VNS-a na rešavanje problema p-centra

Problem p-centra (3. nedelja): Neka je $I = \{1, 2, \dots, n\}$ dati skup lokacija. Neka je sa c_{ij} označeno rastojanje od lokacije i do lokacije j . Zadatak je naći lokacije tačno p centara kojima se minimizuje maksimalno rastojanje svih ne-centar lokacija (korisnika) do njima najbližeg otvorenog centra.

- Reprezentacija rešenja: celobrojni vektor dužine p , koji sadrži lokacije centara. Skup dopustivih rešenja ima $\binom{n}{p}$ elemenata.
- Kreiranje inicijalnog rešenja:
 - 1 Nasumično inicijalno rešenje: na slučajan način bira se p od n potencijalnih lokacija korisnika za otvaranje centara.
 - 2 Pohlepno inicijalno rešenje: inicijalno rešenje je konstruisano pohlepnim algoritmom. Lokacija prvog centra se bira na slučajan način, nakon čega se u svakoj iteraciji (narednih $p - 1$) parcijalnom rešenju dodaje centar koji najviše doprinosi smanjenju funkcije cilja.

Okoline

- Za definisanje okolina korišćena je Hamingova metrika:
 $d(x, x') = \text{broj različitih elemenata vektora } x \text{ i } x' = p - |x \cap x'|$
- Za fazu razmrđavanja su korišćene okoline
 $N_k(x) = \{x' | d(x, x') = k\}, k = 1, 2, \dots, \lceil \frac{p}{2} \rceil$.
- Za BVNS i SVNS: U LS koraku se pretražuju N_1 okolina.

Okoline za LS za VND

- Kako vrednost parametra k raste, okolina N_k postaje sve veća i teža za pretraživanje, ($|N_k| = \binom{p}{k} \binom{n-p}{k}$) zato u okviru VND metode (u svakom koraku zahteva nalaženje najboljeg rešenja u odredjenoj okolini tekućeg rešenja) koristimo drugi skup okolina LK_l , $l = 1, \dots, n - p$. Susedi LK_l okoline tekućeg rešenja x dobijaju se na sledeći način:

- 1 Naći par indeksa (i_{in}^*, i_{out}^*) za koji rešenje $x' = (x \setminus i_{out}) \cup i_{in}$ ima najmanju vrednost funkcije cilja.
- 2 Kreirati rešenje $x' = (x \setminus i_{out}) \cup i_{in}$.
- 3 Ponoviti prethodne korake l puta pri čemu čvor i_{out} koji je izbačen u nekoj od prethodnih iteracija ne sme ponovo postati deo rešenja, tj. dobiti status i_{in} čvora u narednim iteracijama.

Ovim postupkom kreirano je l rešenja (elementi okoline LK_l tekućeg rešenja x). Kao krajnji rezultat lokalnog pretraživanja bira se rešenje sa najmanjom vrednošću funkcije cilja.