



Iskanje po variabilnih okolich

Seminarska naloga pri predmetu Izbrana poglavja iz optimizacije

Katja Ciglar

26. maj 2011

Uvodna motivacija

Uvod

Spust po
variabilnih
okolihah

Posplošitev
iskanja po
variabilnih
okolihah

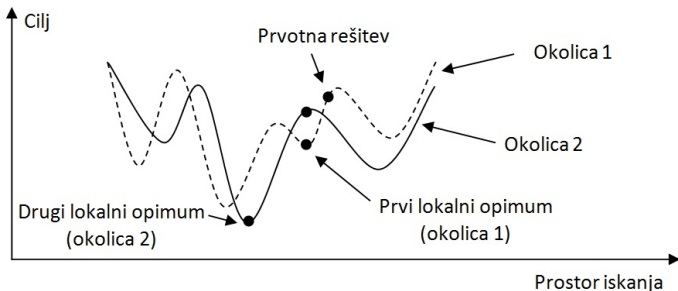
Primer

Iskanje po variabilnih okolihah (ang. *variable neighborhood search*, v nadaljevanju VNS)

Osnovna ideja

- Raziskovanje množice v naprej določenih okolic z namenom iskanja boljše optimalne rešitve.
- Raziskovanje množice okolic lahko poteka slučajno ali pa sistematično, da s tem pridemo do lokalnega optimuma in ga zapustimo.
- Raziskovanje različnih okolic pri lokalnem iskanju lahko generira različne lokalne optime, globalni optimum je lokalni optimum za dano okolico.

Ideja iskanja po variabilnih okolihah



SLIKA 1: Iskanje po variabilnih okolihah z uporabo dveh okolih. Prvi lokalni optimum dobimo glede na okolico 1. Glede na okolico 2 dobimo drugi lokalni optimum iz prvega lokalnega optimuma.

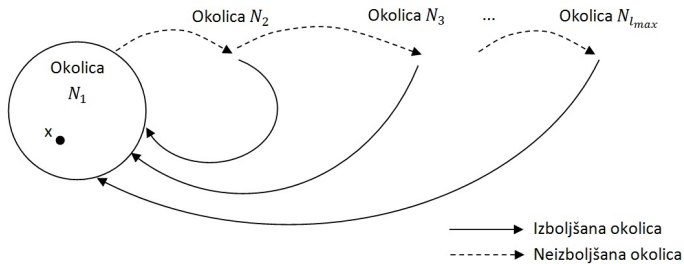
Spust po variabilnih okolihah

Spust po variabilnih okolihah (ang. *variable neighborhood descent*, v nadaljevanju VND)

Osnovna ideja

- Algoritem je deterministična verzija VNS.
- Algoritem uporablja zaporedne okolice pri spustu v lokalni optimum.
- V prvem koraku definiramo množico okolih N_l ($l = 1, \dots, l_{max}$).
- Začnemo v okolici N_1 in naj bo x prvotna rešitev.
- V primeru, da izboljšanje rešitve x v njeni trenutni okolici $N_l(x)$ ne obstaja, se okolica spremeni iz N_l v N_{l+1} .
- Če v postopku najdemo izboljšanje rešitve x , se okolica vrne v prvo N_1 in ponovno začnemo postopek (Slika 2).
- Strategija bo uspešna, če bodo različne okolice komplementarne.

Spust po variabilnih okolihah



SLIKA 2: Princip delovanja algoritma spust po variabilnih okolihah.

Spust po variabilnih okolihah

ALGORITEM 1: Predloga algoritma spust po variabilni okolici.

Input: množica okolih N_l za $l = 1, \dots, l_{max}$.

$x = x_0$; /*Generiramo prvotno rešitev*/

$l = 1$;

While $l \leq l_{max}$ **do**

 Poišči najboljšega soseda x' od x v $N_l(x)$;

If $f(x') \leq f(x)$ **Then** $x = x'$; $l = 1$;

Otherwise $l = l + 1$;

Output: Najboljša najdena rešitev.

Spust po variabilnih okolihah

Uporaba algoritma

- Algoritem VND je povezan z izbiro okolic za namen njihove nadaljne uporabe.
- Pri tem je potrebno upoštevati kompleksnost okolic, saj večje kot so okolice, večja je časovna zahtevnost algoritma.
- Najbolj popularna strategija je naraščajoča razvrstitev okolic glede na njihovo kompleksnost, to pomeni glede na njihovo velikost $|N_I(x)|$.

Posplošitev iskanja po variabilnih okolihah

Opis algoritma

- VNS je stohastični algoritem pri katerem najprej definiramo množico okolih N_k ($k = 1, \dots, n$).
- Vsaka iteracija algoritma je sestavljena iz treh korakov: pretresenje, lokalno iskanje in premik.
- Pri vsaki iteraciji je prvotna rešitev pretresena iz trenutne okolice N_k , kar pomeni, da je rešitev x' naključno generirana v trenutni okolici $N_k(x)$.
- Postopek lokalnega iskanja apliciramo na rešitev x' , da generiramo rešitev x'' . Novo rešitev x'' obdržimo, če in samo če smo našli boljšo rešitev (to pomeni $f(x'') \leq f(x)$).
- Enak proces ponovno začnemo z rešitvijo x'' v prvi okolici N_1 .
- V primeru, da ne najdemo boljše rešitve (to pomeni $f(x'') \geq f(x)$), se algoritem premakne v naslednjo okolico N_{k+1} , slučajno generira novo rešitev v tej okolici in jo poskuša izboljšati.

Algoritem iskanja variabilnih okolich

ALGORITEM 2: Predloga algoritma iskanja po variabilnih okolich.

Input: množica okolich N_l za $l = 1, \dots, l_{max}$.

$x = x_0$; /*Generiramo prvotno rešitev*/
 $l = 1$;

Repeat

 Pretresi: izberi slučajno rešitev x' iz k -te okolice $N_k(x)$ od x ;

$x'' = \text{lokalno iskanje}(x')$;

If $f(x'') \leq f(x)$ **Then** $x = x''$;

 Nadaljuj iskanje z N_1 ; $k = 1$;

Otherwise $k = k + 1$;

Until $k = k_{max}$;

Until Kriterij končanja

Output: Najboljša najdena rešitev.

Posplošitev iskanja po variabilnih okolihah

Opombe

- Bolj splošni algoritem VNS dobimo, če enostavno lokalno iskanje nadomestimo z algoritmom VND.
- Namen uporane algoritma VNS je povezano z izbiro okolic v koraku pretresenja. Običajno vzamemo vgnezdene okolice, kjer vsaka naslednja okolica $N_k(x)$ vsebuje prejšno $N_{k-1}(x)$:

$$N_1(x) \subset N_2(x) \subset \dots \subset N_k(x), \forall x \in S.$$

- Potrebno je poiskati kompromis med intenzivnostjo iskanja in njegovo diverzifikacijo preko porazdelitve dela med fazi lokalno iskanje in pretresenje. Več dela v fazi lokalnega iskanja (več intenzivnosti) nam generira boljši lokalni optimum, medtem ko več dela v fazi pretresenja vrne potencialno boljše regije prostora iskanja (več diverzifikacije).

Primer

Uporaba algoritma na primeru iz resničnega življenja

Primer prikazuje uporabo več okolic ($k = 9$) za rešitev problema razporeda. Veliko nahajališč nafte na kopenskih področjih se zanaša na umetne metode dviganja. Vzdrževanje, ki vključuje čiščenje, stimuliranje in drugo, je ključnega pomena za delovanje nahajališča. Vzdrževanje opravljajo s pomočjo ploščadi. Število razpoložljivih ploščadi je omejeno glede na potrebe po vzdrževanju nahajališč. Odločitev katero od ploščadi bodo poslali za izvedbo vzdrževanja je odvisno od več faktorjev, kot so produktivnost nahajališča, trenutno nahajanje ploščadi glede na zahtevano nahajališče in vrsta potrebnega vzdrževanja. Problem razporeditve ploščadi je poiskati najboljšo razporeditev S^* za razpoložljivih m ploščadi, tako da minimiziramo izgubo proizvodnje nahajališč, ki čakajo na vzdrževanje. Razpored je opredeljen z urejeno množico vzdrževanja nahajališč s ploščadmi. Algoritem iskanja po variabilnih okolich je bil predlagan za rešitev tega problema z uporabo naslednjih okolic v koraku pretresenja:

- 1 Zamenjava poti, kjer nahajališča povezana z dvema ploščadma zamenjamo. (SS)
- 2 Zamenjava nahajališč z isto ploščadjo, kjer zamenjamo vzdrževanje dveh nahajališč z isto ploščadjo. (SWSW)
- 3 Zamenjava nahajališč z različnima ploščadma, kjer dve nahajališči vzdrževani z dvema različnima ploščadma zamenjamo. (SWDW)
- 4 Dodaj/odvzemi (AD), kjer je nahajališče prerazporejeno na razpored druge ploščadi.
- 5 Dve uporabi transformacije SWSW.
- 6 Dve uporabi transformacije SWDW.
- 7 Tri uporabe transformacije SWDW.
- 8 Zaporedna uporaba dveh AD transformacij.
- 9 Zaporedna uporaba treh AD transformacij.