

MIP-heuristike (Matheuristicke)

Hibridi između metaheurističkih i egzaktnih metoda

Tatjana Davidović

Matematički institut SANU
<http://www.mi.sanu.ac.rs/~tanjad>
(tanjad@mi.sanu.ac.rs)

14. jan. 2016.



Sadržaj

- 1 Matheuristike za 0-1 mešovito celobrojno programiranje
- 2 Opis nekih matheuristika
- 3 Primeri primena
- 4 Još o matheuristikama



Šta su MATHEURISTIKE?

- Naziv matheuristike je skraćenica za matematičke heuristike (*math-heuristic*);



Šta su MATHEURISTIKE?

- Naziv matheuristike je skraćenica za matematičke heuristike (*math-heuristic*);
- To su hibridi između metaheuristika i metoda zasnovanih na matematičkom programiranju;



Šta su MATHEURISTIKE?

- Naziv matheuristike je skraćenica za matematičke heuristike (*math-heuristic*);
- To su hibridi između metaheuristika i metoda zasnovanih na matematičkom programiranju;
- Nova rešenja dobijaju se manipulacijama nad matematičkim modelom datog problema;



Šta su MATHEURISTIKE?

- Naziv matheuristike je skraćenica za matematičke heuristike (*math-heuristic*);
- To su hibridi između metaheuristika i metoda zasnovanih na matematičkom programiranju;
- Nova rešenja dobijaju se manipulacijama nad matematičkim modelom datog problema;
- U opštem slučaju i to su egzaktne metode, tj. ako im se da "dovoljno" vremena pronaći će optimalno rešenje;



Šta su MATHEURISTIKE?

- Naziv matheuristike je skraćenica za matematičke heuristike (*math-heuristic*);
- To su hibridi između metaheuristika i metoda zasnovanih na matematičkom programiranju;
- Nova rešenja dobijaju se manipulacijama nad matematičkim modelom datog problema;
- U opštem slučaju i to su egzaktne metode, tj. ako im se da "dovoljno" vremena pronaći će optimalno rešenje;
- Soyster-ov algoritam za probleme 0-1 programiranja iz 1978. godine bio je jedna od prvih matheuristika.



Definicija problema 0-1 mešovitog celobrojnog programiranja (0-1 MIP)

$$\max(\min)\{\nu(\xi) = c^T \xi \mid \xi \in X\}, \quad (1)$$

gde je

$X = \{\xi \in \mathbb{R}^{|N|} \mid A\xi \leq b; \xi_j \in \{0, 1\} \text{ za } j \in \mathcal{B}; \xi_j \in \mathbb{Z}_0^+ \text{ za } j \in \mathcal{G}; \xi_j \geq 0 \text{ za } j \in \mathcal{C}\}$ dopustivi skup (skup dopustivih rešenja),

$N = \mathcal{C} \cup \mathcal{G} \cup \mathcal{B}$ skup indeksa svih promenljivih,

\mathcal{C} skup indeksa neprekidnih promenljivih,

\mathcal{G} skup indeksa celobrojnih promenljivih (koje nisu binarne), a

\mathcal{B} skup indeksa binarnih promenljivih,

$\mathcal{C} \cap \mathcal{G} = \emptyset, \mathcal{C} \cap \mathcal{B} = \emptyset, \mathcal{G} \cap \mathcal{B} = \emptyset, \mathcal{B} \neq \emptyset$.



Rastojanje u 0-1 MIP prostoru rešenja

- S – Prostor rešenja za 0-1 MIP problem P ;
- \bar{S} – Prostor rešenja za LP relaksaciju $LP(P)$ problema P ;

Definicija. Rastojanje između $x \in S$ i $y \in \bar{S}$ definiše se kao

$$\delta(x, y) = \sum_{j \in \mathcal{B}} |x_j - y_j|,$$

a može se linearizovati na sledeći način (Fischetti & Lodi, 2003):

$$\delta(x, y) = \sum_{j \in \mathcal{B}} x_j(1 - y_j) + y_j(1 - x_j).$$

Kada $x, y \in \{0, 1\}^n$, $\delta(x, y)$ je ekvivalentno Hamingovom rastojanju.



Lokalno grananje (Local Branching, LB)

(Fischetti & Lodi, 2003)

- Učita se problem i postave vrednosti parametara;



Lokalno grananje (Local Branching, LB)

(Fischetti & Lodi, 2003)

- Učita se problem i postave vrednosti parametara;
- Nađe se prvo dopustivo rešenje x' ;



Lokalno grananje (Local Branching, LB)

(Fischetti & Lodi, 2003)

- Učita se problem i postave vrednosti parametara;
- Nađe se prvo dopustivo rešenje x' ;
- Ograniči se pretraga na ona rešenja x za koja važi $\delta(x, x') \leq k$, gde je k početna vrednost parametra LB metode;



Lokalno grananje (Local Branching, LB)

(Fischetti & Lodi, 2003)

- Učita se problem i postave vrednosti parametara;
- Nađe se prvo dopustivo rešenje x' ;
- Ograniči se pretraga na ona rešenja x za koja važi $\delta(x, x') \leq k$, gde je k početna vrednost parametra LB metode;
- Egzaktni MIP solver (CPLEX) se poziva da reši dobijeni podproblem u zadatom vremenskom intervalu;



Lokalno grananje (Local Branching, LB)

(Fischetti & Lodi, 2003)

- Učita se problem i postave vrednosti parametara;
- Nađe se prvo dopustivo rešenje x' ;
- Ograniči se pretraga na ona rešenja x za koja važi $\delta(x, x') \leq k$, gde je k početna vrednost parametra LB metode;
- Egzaktni MIP solver (CPLEX) se poziva da reši dobijeni podproblem u zadatom vremenskom intervalu;
- Zavisno od rezultata koje vrati CPLEX, rastojanje k se povećava ili smanjuje i proces ponavlja do zadovoljenja kriterijuma zaustavljanja.



LB pseudokod

Procedure LocBra(P, x', k^*)

(1) Initialisation. Set $proceed \leftarrow \text{true}$, $k \leftarrow k^*$;

(2) Main step.

while ($proceed$) **do**

Add cuts: $c^T x \geq c^T x'$, $\delta(x', x) \leq k$;

Set $x'' \leftarrow \text{MIPSOLVE}(P, x')$;

switch status **do**

case "optSolFound": reverse last pseudo-cut into
 $\delta(x', x) \geq k + 1$; $x' \leftarrow x''$; $k \leftarrow k^*$;

case "feasibleSolFound": replace last pseudo-cut with
 $\delta(x', x) \geq 1$; $x' \leftarrow x''$; $k \leftarrow k^*$;

case "provenInfeasible": reverse last pseudo-cut into
 $\delta(x', x) \geq k + 1$; $k \leftarrow k + \lceil k^*/2 \rceil$;

case "noFeasibleSolFound": delete last pseudo-cut
 $\delta(x', x) \leq k$; $k \leftarrow k - \lceil k^*/2 \rceil$;

end

end

(3) Output. **return** x' ;



Matheuristike zasnovane na metodi promenljivih okolina

Variable Neighborhood Search (VNS)

- Pravila VNS metode koriste se za definisanje podproblema;



Matheuristike zasnovane na metodi promenljivih okolina

Variable Neighborhood Search (VNS)

- Pravila VNS metode koriste se za definisanje podproblema;
- CPLEX se poziva da u vremenskom ograničenju reši podproblem;



Matheuristike zasnovane na metodi promenljivih okolina

Variable Neighborhood Search (VNS)

- Pravila VNS metode koriste se za definisanje podproblema;
- CPLEX se poziva da u vremenskom ograničenju reši podproblem;
- Dakle, CPLEX igra ulogu LS koraka u VNS metodi;



Matheuristike zasnovane na metodi promenljivih okolina

Variable Neighborhood Search (VNS)

- Pravila VNS metode koriste se za definisanje podproblema;
- CPLEX se poziva da u vremenskom ograničenju reši podproblem;
- Dakle, CPLEX igra ulogu LS koraka u VNS metodi;
- Najpoznatije su sledeće tri matheuristike:
 - ① Grananje kroz promenljive okoline (Variable Neighborhood Branching, VNB) [Hansen & Mladenović & Urošević, 2006](#);
 - ② Dekompozicija kroz promenljive okoline za 0-1 MIP (VNDS for 0-1 MIPs) [Lazić et al., 2010](#).
 - ③ Intenzivno pretraživanje kroz promenljive okoline (The Variable Intensity Neighborhood Search for 0-1 MIP, VINS) [Jovanović et al., 2015](#).



VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglaši za trenutno najbolje;



VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglaši za trenutno najbolje;
- Razmatraju se samo rešenja na određenom rastojanju od trenutno najboljeg;



VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglaši za trenutno najbolje;
- Razmatraju se samo rešenja na određenom rastojanju od trenutno najboljeg;
- Rastojanja se menjaju u skladu sa VNS pravilima (od k_{min} do k_{max} sa korakom k_{step});



VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglaši za trenutno najbolje;
- Razmatraju se samo rešenja na određenom rastojanju od trenutno najboljeg;
- Rastojanja se menjaju u skladu sa VNS pravilima (od k_{min} do k_{max} sa korakom k_{step});
- Kao LS se koristi metoda promenljivog spusta (Variable neighborhood descent, VND), naravno opet na zadatom rastojanju;



VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglaši za trenutno najbolje;
- Razmatraju se samo rešenja na određenom rastojanju od trenutno najboljeg;
- Rastojanja se menjaju u skladu sa VNS pravilima (od k_{min} do k_{max} sa korakom k_{step});
- Kao LS se koristi metoda promenljivog spusta (Variable neighborhood descent, VND), naravno opet na zadatom rastojanju;
- Raspon rastojanja za VND se zadaje posebnim parametrom;



VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglaši za trenutno najbolje;
- Razmatraju se samo rešenja na određenom rastojanju od trenutno najboljeg;
- Rastojanja se menjaju u skladu sa VNS pravilima (od k_{min} do k_{max} sa korakom k_{step});
- Kao LS se koristi metoda promenljivog spusta (Variable neighborhood descent, VND), naravno opet na zadatom rastojanju;
- Raspon rastojanja za VND se zadaje posebnim parametrom;
- Razmrdavanje (shaking) se vrši nalaženjem prvog dopustivog rešenja u disk-prstenu veličine k do $k + k_{step}$.



VNB - pseudokod

Procedure VNB($P, k_{min}, k_{step}, k_{max}, k_{vnd}$)

```

1   Set proceed  $\leftarrow$  true; Set solutionLimit = 1;
2    $x' \leftarrow \text{MIPSolve}(P, \text{solutionLimit});$ 
3   solutionLimit  $\leftarrow \infty$ ;  $x^* \leftarrow x'$ ;
4   while (proceed) do
5      $Q \leftarrow P$ ;  $x'' \leftarrow \text{VND-MIP}(Q, k_{vnd}, x');$ 
6     if ( $c^T x'' < c^T x^*$ ) then
7        $x^* \leftarrow x''$ ;  $k \leftarrow k_{min}$ ;
8     else  $k \leftarrow k + k_{step}$ ;
9      $x' \leftarrow \text{Shake}(P, x^*, k, k_{step}, k_{max})$ 
//If no feasible solutions are found, return current (infeasible) solution.
10    if ( $x' = x^*$ ) then break;
11    Update proceed;
12  endwhile
13  return  $x^*$ ;
```



LB i VNB

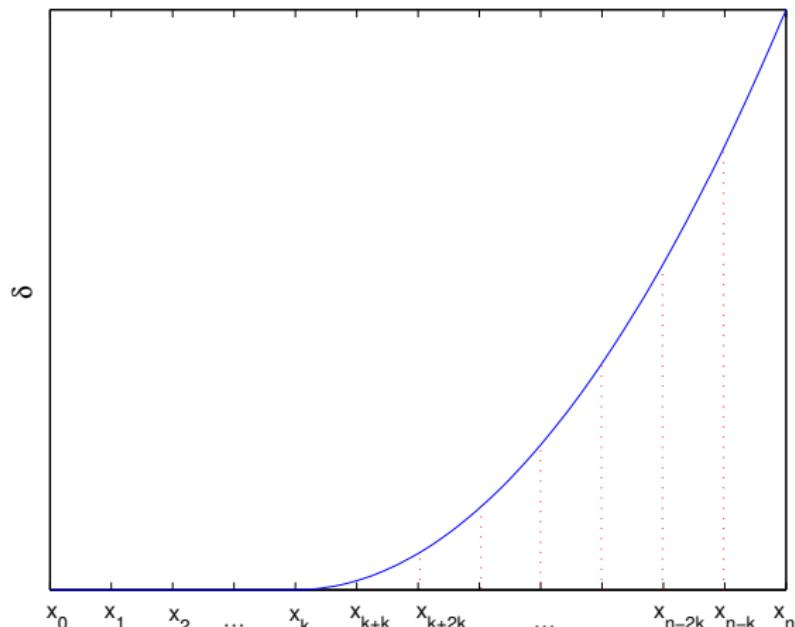
Podešavanje parametara koji zavise od problema

- LB
 - $k^* = \lfloor 0.20|\mathcal{B}| \rfloor$
- VNB
 - $k_{max} = \lfloor 0.5|\mathcal{B}| \rfloor$
 - $k_{min} = k_{step} = k_{vnd} = \lfloor 0.05|\mathcal{B}| \rfloor$
- $tlim$ - dozvoljeno vreme za rad svake od metoda (kriterijum zaustavljanja)
- $tsub$ - dozvoljeno vreme za podproblem (CPLEX) $tsub = tlim/10.0$



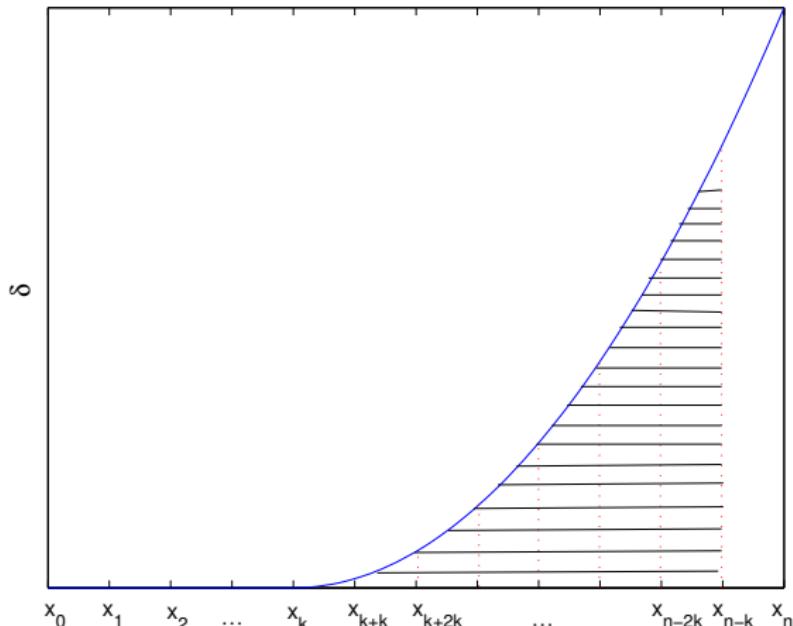
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



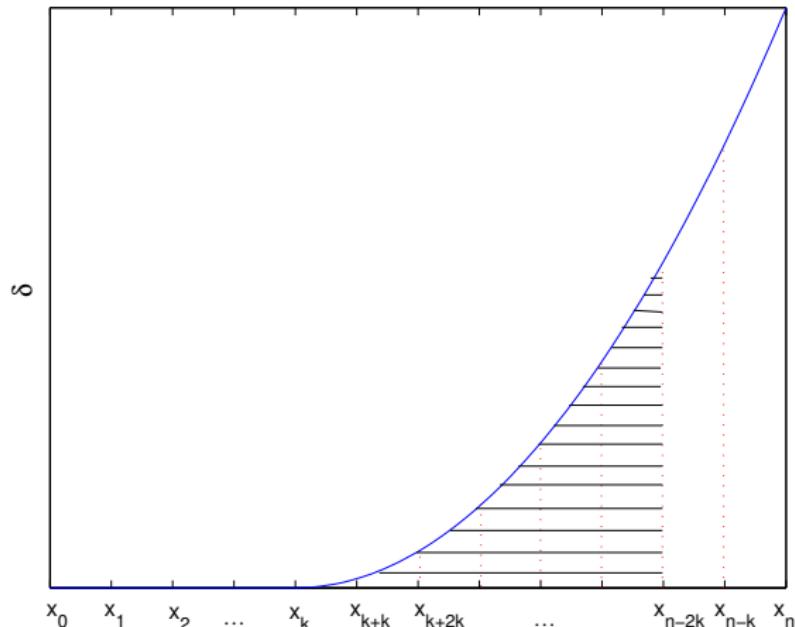
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



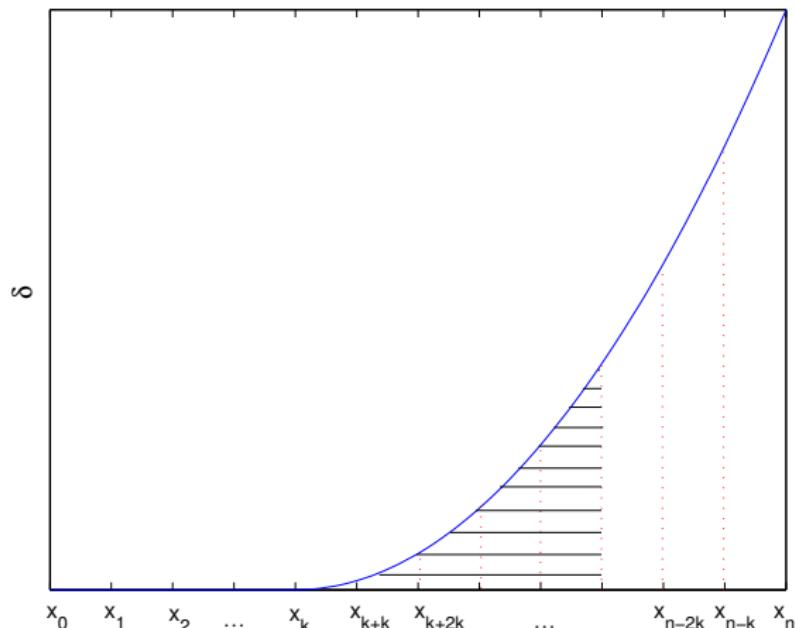
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



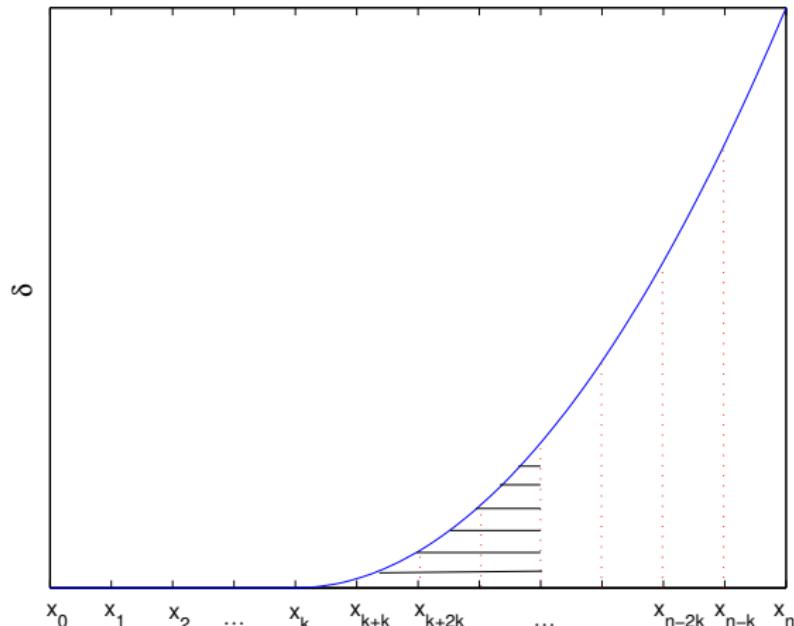
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



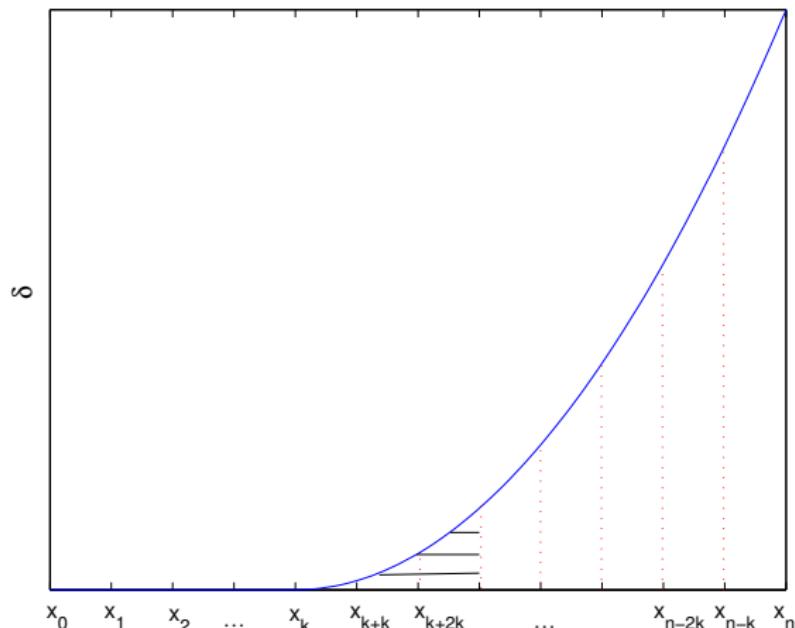
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



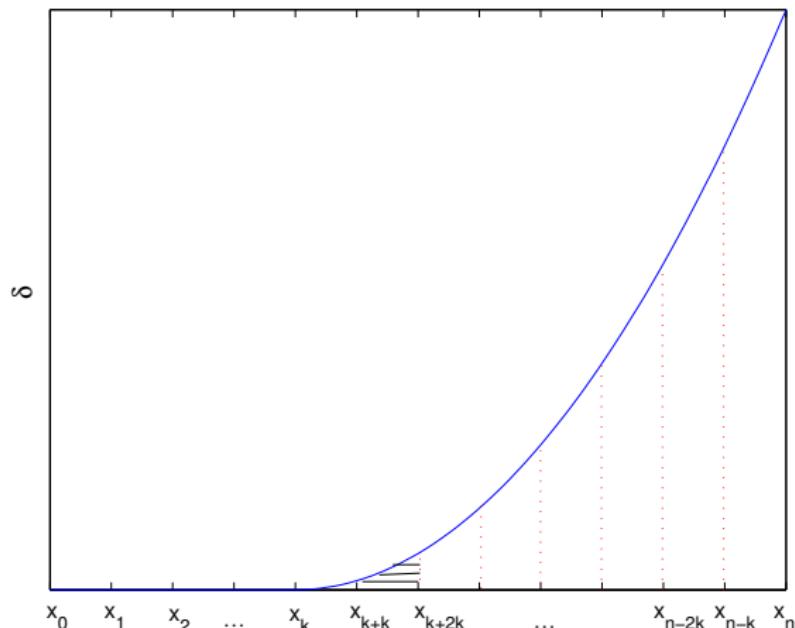
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



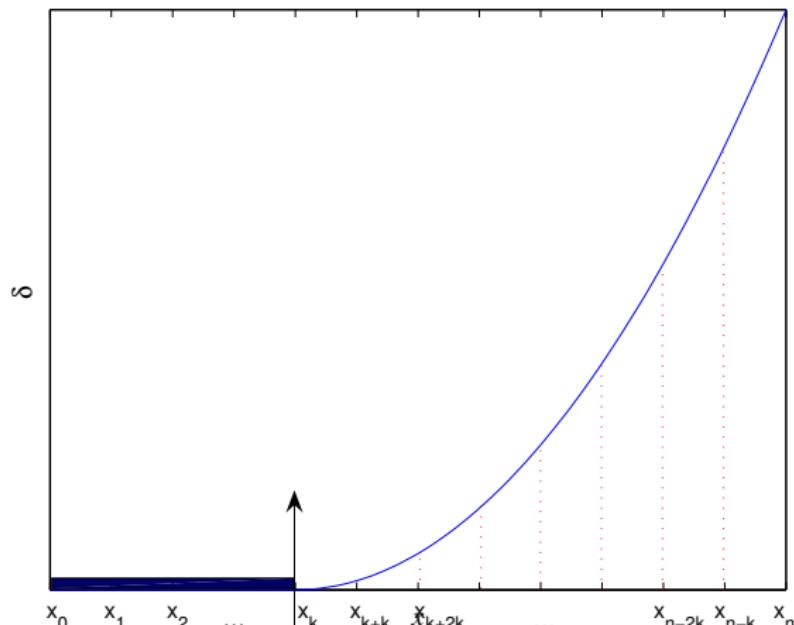
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



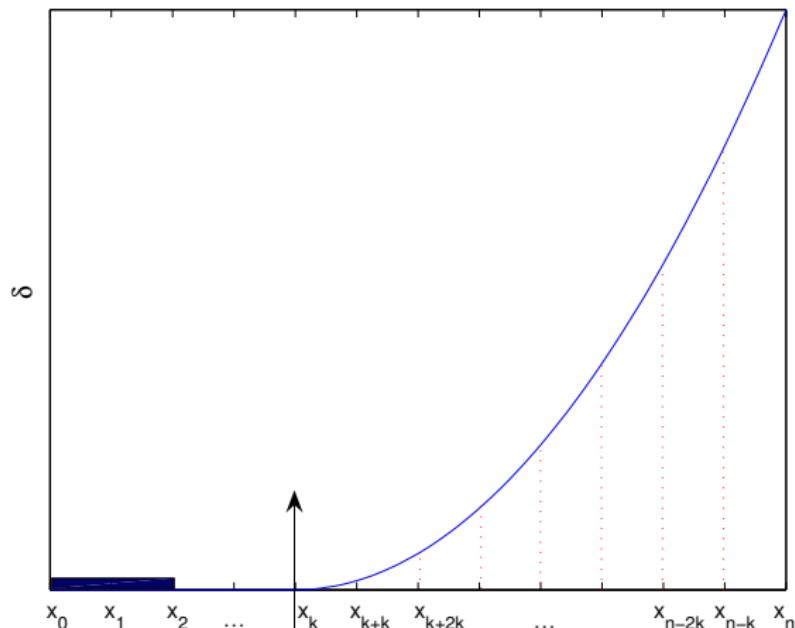
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



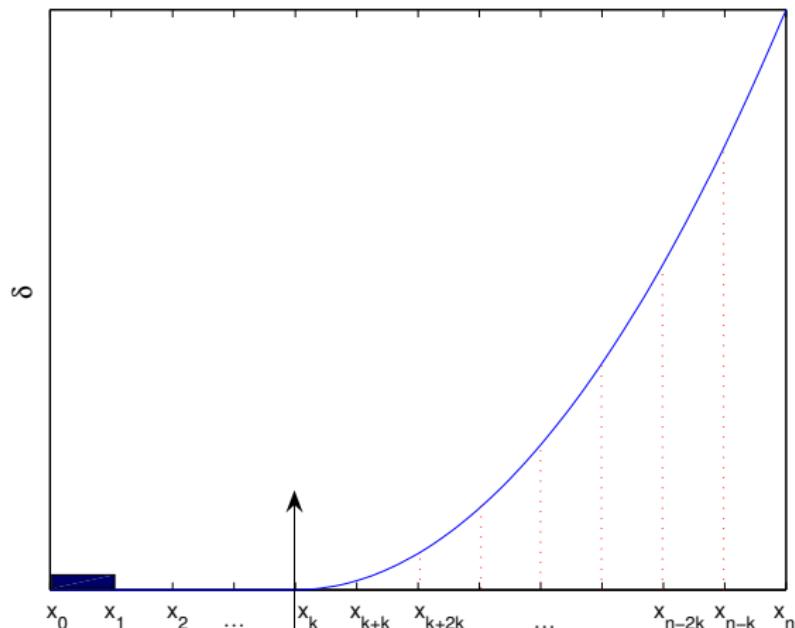
VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



VNDS za 0-1 MIPs - ideja metode

- Neka je \bar{x} optimalno rešenje za $LP(P)$, a $p = |\mathcal{B}|$.
- Neka je $\delta_j = |x_j - \bar{x}_j|$, $j \in \mathcal{B}$, pri čemu su (x_j) uređeni tako da važi $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$.



VNDS za 0-1 MIPs - pseudokod

VNDS-MIP(P, d, x, k_{vnd})

```

1   Find an optimal solution  $\bar{x}$  of LP( $P$ );
2   Set  $proceed1 \leftarrow \text{true}$ ,  $proceed2 \leftarrow \text{true}$ ;
3   while ( $proceed1$ )
4        $\delta_j \leftarrow |x_j - \bar{x}_j|$ ,  $j \in \mathcal{B}$ ; index variables  $x_j, j \in \mathcal{B}$ ,
        so that  $\delta_1 \leq \delta_2 \leq \dots \leq \delta_p$ ,  $p = |\mathcal{B}|$ ;
5       Set  $n_d \leftarrow |\{j \in \mathcal{B} \mid \delta_j \neq 0\}|$ ,  $k_{step} \leftarrow [n_d/d]$ ,  $k \leftarrow p - k_{step}$ ;
6       while ( $proceed2$  and  $k > 0$ )
7           Fix values of  $x_1, \dots, x_k$ ;
8            $x' \leftarrow \text{MIPSOLVE}(P, x)$ ;
9           Release  $x_1, \dots, x_k$ ;
10          if ( $c^T x' > c^T x$ ) then
11               $Q \leftarrow P$ ;  $x \leftarrow \text{VND-MIP}(Q, k_{vnd}, x')$ ; break;
12          else
13              if ( $k - k_{step} > p - n_d$ ) then  $k_{step} \leftarrow \max\{[k/2], 1\}$ ;
14              Set  $k \leftarrow k - k_{step}$ ;
15              Update  $proceed1$  and  $proceed2$ ;
16          endwhile
17      endwhile
18      return  $x$ .

```



VINS - ideja metode

- Kombinacija dve metode:
 - ① VNDS-MIP i
 - ② Variable Intensity Local Search (VILS), [Mitrović-Minić & Punnen, 2009](#);



VINS - ideja metode

- Kombinacija dve metode:
 - ① VNDS-MIP i
 - ② Variable Intensity Local Search (VILS), [Mitrović-Minić & Punnen, 2009](#);
- VILS je specijalna hibridna metoda za problem dodeljivanja;



VINS - ideja metode

- Kombinacija dve metode:
 - ① VNDS-MIP i
 - ② Variable Intensity Local Search (VILS), [Mitrović-Minić & Punnen, 2009](#);
- VILS je specijalna hibridna metoda za problem dodeljivanja;
- U njoj su uvedeni različiti tipovi i veličine okolina;



VINS - ideja metode

- Kombinacija dve metode:
 - ① VNDS-MIP i
 - ② Variable Intensity Local Search (VILS), [Mitrović-Minić & Punnen, 2009](#);
- VILS je specijalna hibridna metoda za problem dodeljivanja;
- U njoj su uvedeni različiti tipovi i veličine okolina;
- Okoline se pretražuju po round-robin principu za zadato vreme;



VINS - ideja metode

- Kombinacija dve metode:
 - ① VNDS-MIP i
 - ② Variable Intensity Local Search (VILS), [Mitrović-Minić & Punnen, 2009](#);
- VILS je specijalna hibridna metoda za problem dodeljivanja;
- U njoj su uvedeni različiti tipovi i veličine okolina;
- Okoline se pretražuju po round-robin principu za zadato vreme;
- Na te principe dograđene su ideje za različite tipove okolina u opštem slučaju.



VINS - ideja metode

- Sortiranje promenljivih, fiksiranje i relaksiranje je inspirisano VNDS-MIP-om;



VINS - ideja metode

- Sortiranje promenljivih, fiksiranje i relaksiranje je inspirisano VNDS-MIP-om;
- Definicije okolina preuzete su iz VILS-a;



VINS - ideja metode

- Sortiranje promenljivih, fiksiranje i relaksiranje je inspirisano VNDS-MIP-om;
- Definicije okolina preuzete su iz VILS-a;
- Na početku se odrede rešenje linearne relaksacije i prvo dopustivo rešenje;



VINS - ideja metode

- Sortiranje promenljivih, fiksiranje i relaksiranje je inspirisano VNDS-MIP-om;
- Definicije okolina preuzete su iz VILS-a;
- Na početku se odrede rešenje linearne relaksacije i prvo dopustivo rešenje;
- Promenljive se sortiraju u ne-rastućem redosledu u odnosu na rastojanje ta dva rešenja;



VINS - ideja metode

- Sortiranje promenljivih, fiksiranje i relaksiranje je inspirisano VNDS-MIP-om;
- Definicije okolina preuzete su iz VILS-a;
- Na početku se odrede rešenje linearne relaksacije i prvo dopustivo rešenje;
- Promenljive se sortiraju u ne-rastućem redosledu u odnosu na rastojanje ta dva rešenja;
- Podskup promenljivih se fiksira na trenutne dopustive vrednosti zavisno od definicije trenutne okoline;



VINS - ideja metode

- Sortiranje promenljivih, fiksiranje i relaksiranje je inspirisano VNDS-MIP-om;
- Definicije okolina preuzete su iz VILS-a;
- Na početku se odrede rešenje linearne relaksacije i prvo dopustivo rešenje;
- Promenljive se sortiraju u ne-rastućem redosledu u odnosu na rastojanje ta dva rešenja;
- Podskup promenljivih se fiksira na trenutne dopustive vrednosti zavisno od definicije trenutne okoline;
- Vrednosti ostalih promenljivih određuju se pomoću MILP solvera.



VINS - Definicije okolina

- N1: $\alpha\%$ najgorih promenljivih se otpušta;
- N2: skup promenljivih se deli na 10 delova i $\alpha/10\%$ najgorih promenljivih se otpušta u svakom delu;
- N3: počev od slučajne pozicije, otpušta se $\alpha\%$ promenljivih;
- N4: na 20 slučajnih pozicija otpušta se $\alpha/10\%$ promenljivih;
- N5: $\alpha\%$ najboljih promenljivih se otpušta;



VINS - Definicije okolina (nast.)

- N6: skup promenljivih se deli na 10 delova i $\alpha/10\%$ najboljih promenljivih se otpušta u svakom delu;
- N7: otpušta se $\alpha/2\%$ najboljih i najgorih promenljivih;
- N8: unutar 10 delova otpušta se $\alpha/20\%$ najboljih i najgorih promenljivih;
- N9: slučajnih $\alpha\%$ promenljivih se otpušta;
- N10: unutar 10 delova otpušta se slučajnih $\alpha/10\%$ promenljivih.



VINS - Pseudo-code

```

Initialization: Read MIP, alphas, time_limits;
    Solve linear relaxation of MIP to obtain linx;
    Calculate x, the first feasible solution for MIP;
    improvement  $\leftarrow$  true;
While (t < Tlim)
    If (improvement)
        bestx  $\leftarrow$  x;
        Add objective constraint  $f(x) < f(bestx)$ ;
        Sort variables according to (bestx – linx);
        improvement  $\leftarrow$  false;
    End If
    N  $\leftarrow$  NextNeighborhood;
    If (N = null)
         $\alpha \leftarrow NextAlpha$ ;
        If ( $\alpha = null$ ) break;
        N = N1
    End If
    Release(N,  $\alpha$ );
    x  $\leftarrow$  MIPSolve(MIP, time_limit);
End While

```



Opšte napomene

- Efikasnost ovih metoda zavisi od složenosti problema i pravilnog podešavanja parametara;



Opšte napomene

- Efikasnost ovih metoda zavisi od složenosti problema i pravilnog podešavanja parametara;
- Metode ne zalaze u prirodu problema niti koriste *a priori* znanje o pojedinim primerima;



Opšte napomene

- Efikasnost ovih metoda zavisi od složenosti problema i pravilnog podešavanja parametara;
- Metode ne nalaze u prirodi problema niti koriste *a priori* znanje o pojedinim primerima;
- Ulaz za sve metode su datoteka koja sadrži LP formulaciju problema (uključujući i ulazne podatke za konkretan primer) i izabrane vrednosti parametara;



Opšte napomene

- Efikasnost ovih metoda zavisi od složenosti problema i pravilnog podešavanja parametara;
- Metode ne nalaze u prirodi problema niti koriste *a priori* znanje o pojedinim primerima;
- Ulaz za sve metode su datoteka koja sadrži LP formulaciju problema (uključujući i ulazne podatke za konkretan primer) i izabrane vrednosti parametara;
- Jedina merila složenosti problema (tj. konkretnog primera) su njegova veličina iskazana kroz broj promenljivih i broj ograničenja.



Neke primene

- Standardni problemi za testiranje 0-1 MIP solvera: 29 grupa test problema ([VNDS-MIP](#), [VINS](#));



Neke primene

- Standardni problemi za testiranje 0-1 MIP solvera: 29 grupa test primera ([VNDS-MIP](#), [VINS](#));
- Problem rutiranja rečnih kontejnerskih brodova: 20 slučajnih test primera različitih dimenzija ([VNB](#), [VINS](#));



Neke primene

- Standardni problemi za testiranje 0-1 MIP solvera: 29 grupa test primera ([VNDS-MIP](#), [VINS](#));
- Problem rutiranja rečnih kontejnerskih brodova: 20 slučajnih test primera različitih dimenzija ([VNB](#), [VINS](#));
- Problem dodele vezova brodovima u luci (BAP): dve grupe primera različitih dimenzija ([VNDS-MIP](#));



Neke primene

- Standardni problemi za testiranje 0-1 MIP solvera: 29 grupa test primera ([VNDS-MIP](#), [VINS](#));
- Problem rutiranja rečnih kontejnerskih brodova: 20 slučajnih test primera različitih dimenzija ([VNB](#), [VINS](#));
- Problem dodele vezova brodovima u luci (BAP): dve grupe primera različitih dimenzija ([VNDS-MIP](#));
- Problem zajedničkih vožnji (Carpooling problem): dva realna primera sa Politehničkog fakulteta u Milanu ([VNB](#)).



Još neke matheuristike:

- HyperOpt, [Burke et al. 2001](#);
- MetaBoosting, [Pushinger et al. 2009](#);
- MIP-Based GRASP, [Dolgui et al. 2009](#);
- Kernel Search, [Enrico, Mansini, Speranca, 2010](#).

Knjiga:

Maniezzo, Vittorio, Thomas Stützle, and Stefan Voss, eds. Matheuristics: hybridizing metaheuristics and mathematical programming. Vol. 10. Springer, 2009.



Konferencije o matheuristikama

6-th Workshop on Model-Based Meta-heuristics Matheuristics 2016
September 4-7, 2016, Brussels, Belgium
<http://iwi.econ.uni-hamburg.de/mh14/prev.html>

Prethodne konferencije:

- Matheuristics 2006 in Cesena, Italy;
- Matheuristics 2008 in Cesena, Italy;
- Matheuristics 2010 in Vienna, Austria;
- Matheuristics 2012, in R4o de Janeiro, Brazil;
- Matheuristics 2014, in Hamburg, Germany.



Hvala na pažnji!

Pitanja?

Tatjana Davidović
tanjad@mi.sanu.ac.rs

