MIP-heuristike (Matheuristike)

Hibridi između metaheurističkih i egzaktnih metoda

Tatjana Davidović

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

21. januar 2013.

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)

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

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;
- $\overline{S}$ – Prostor rešenja za LP relaksaciju $LP(P)$ problema P;

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

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

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* $\leftarrow$ true, $k \leftarrow k^*$;

(2) *Main step.*

while (*proceed*) **do**

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

Set $x'' \leftarrow$ 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)

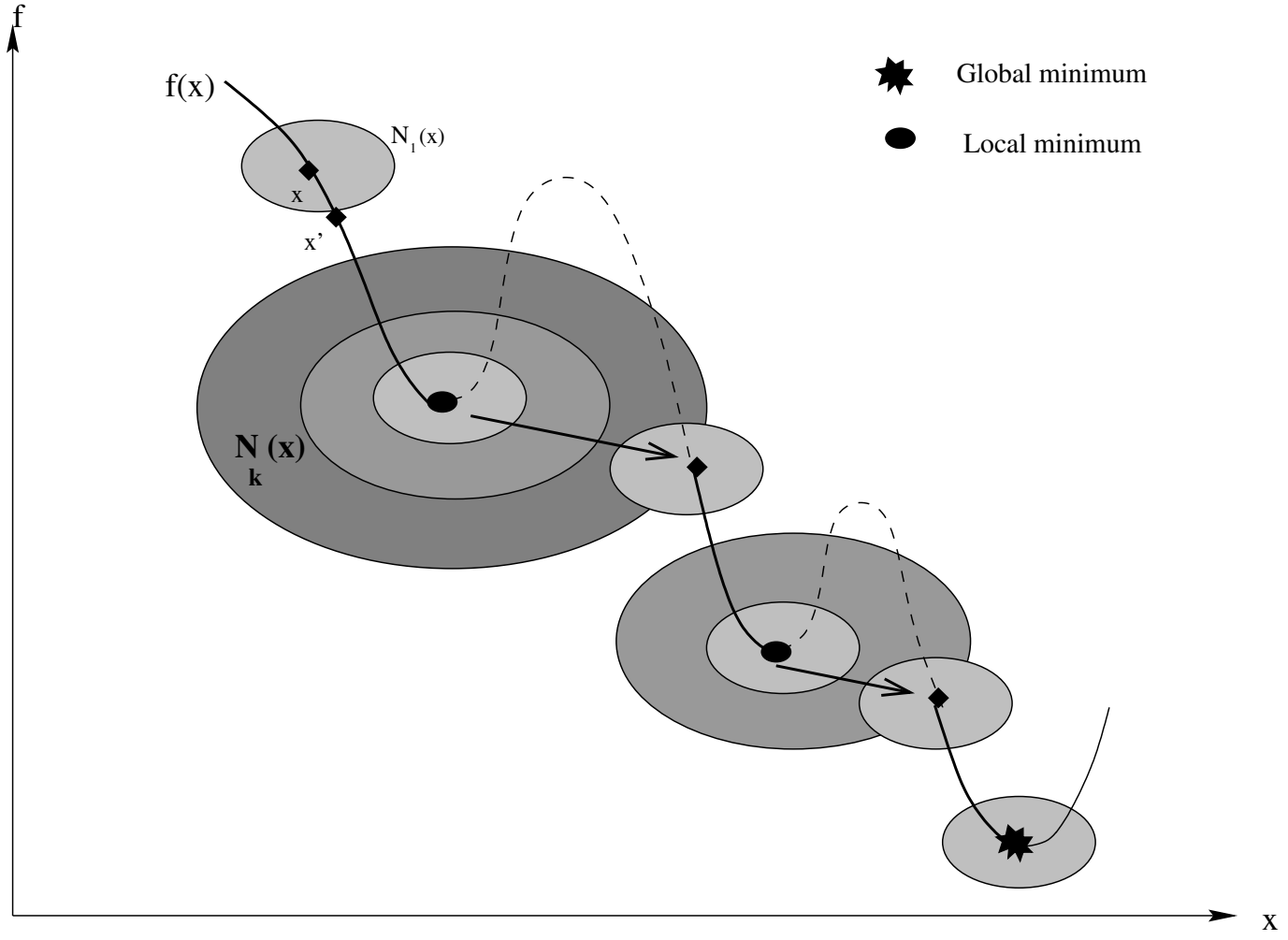
- Pravila VNS metode koriste se za definisanje podproblema;
- CPLEX se poziva da u vremenskom ograničenju reši podproblem;
- Dakle, CPLEX igra ulogu LS koraka u VNS metodi;

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 dve 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.

VNS koncept)



VNB - ideja metode

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

VNB - ideja metode

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

VNB - ideja metode

- Nađe se prvo dopustivo rešenje i proglasi 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 proglasi 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 proglasi 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 proglasi 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)
    Set proceed ← true; Set solutionLimit = 1;
    x' ← MIPSolve(P, solutionLimit);
    solutionLimit ← ∞; x* ← x';
    while (proceed) do
        Q ← P; x'' ← VND-MIP(Q, k_vnd, x');
        if (c^T x'' < c^T x*) then
            x* ← x''; k ← k_min;
        else k ← k + k_step;
        x' ← Shake(P, x*, k, k_step, k_max)
    //If no feasible solutions are found, return current (infeasible) solution.
        if (x' = x*) then break;
        Update proceed;
    endwhile
    return x*;
```

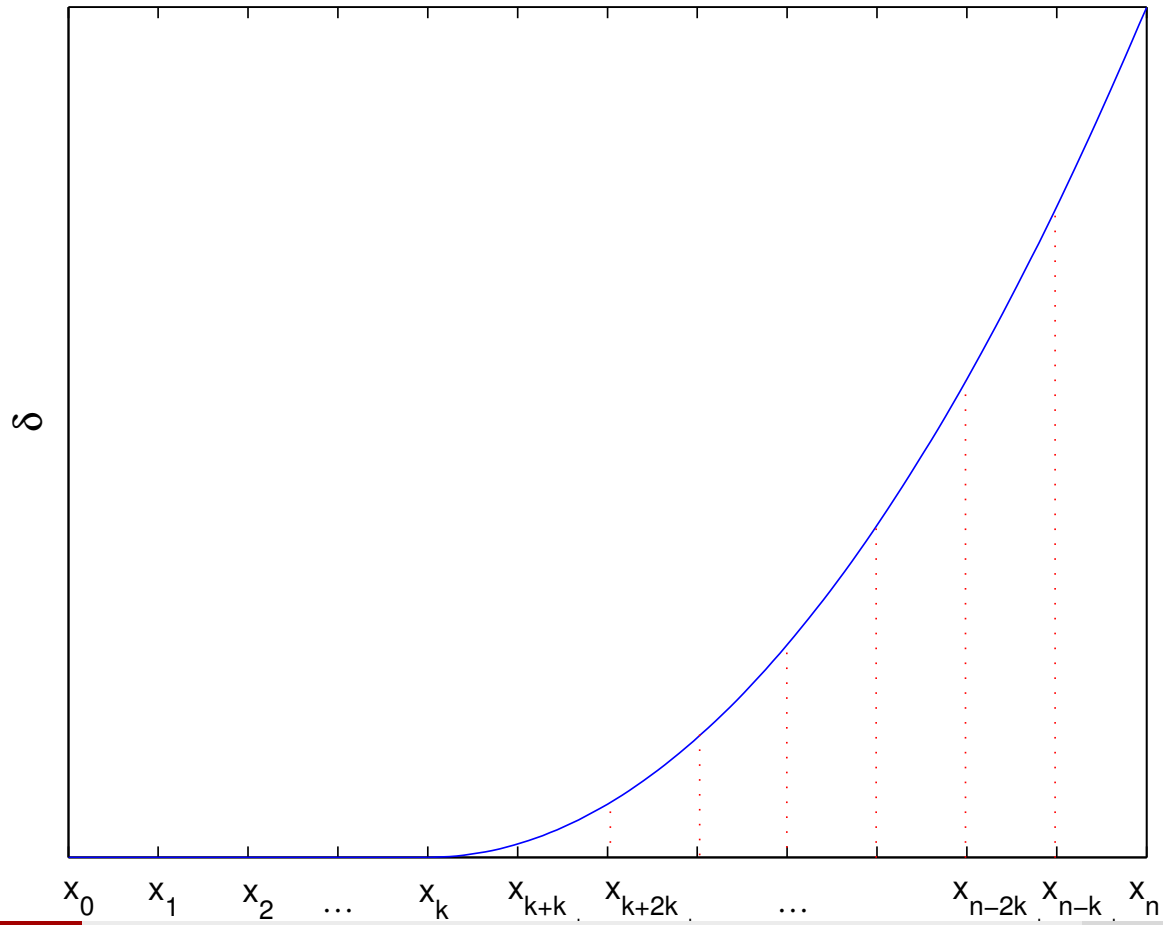
LB i VNB

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) $tsub = tlim/10.0$

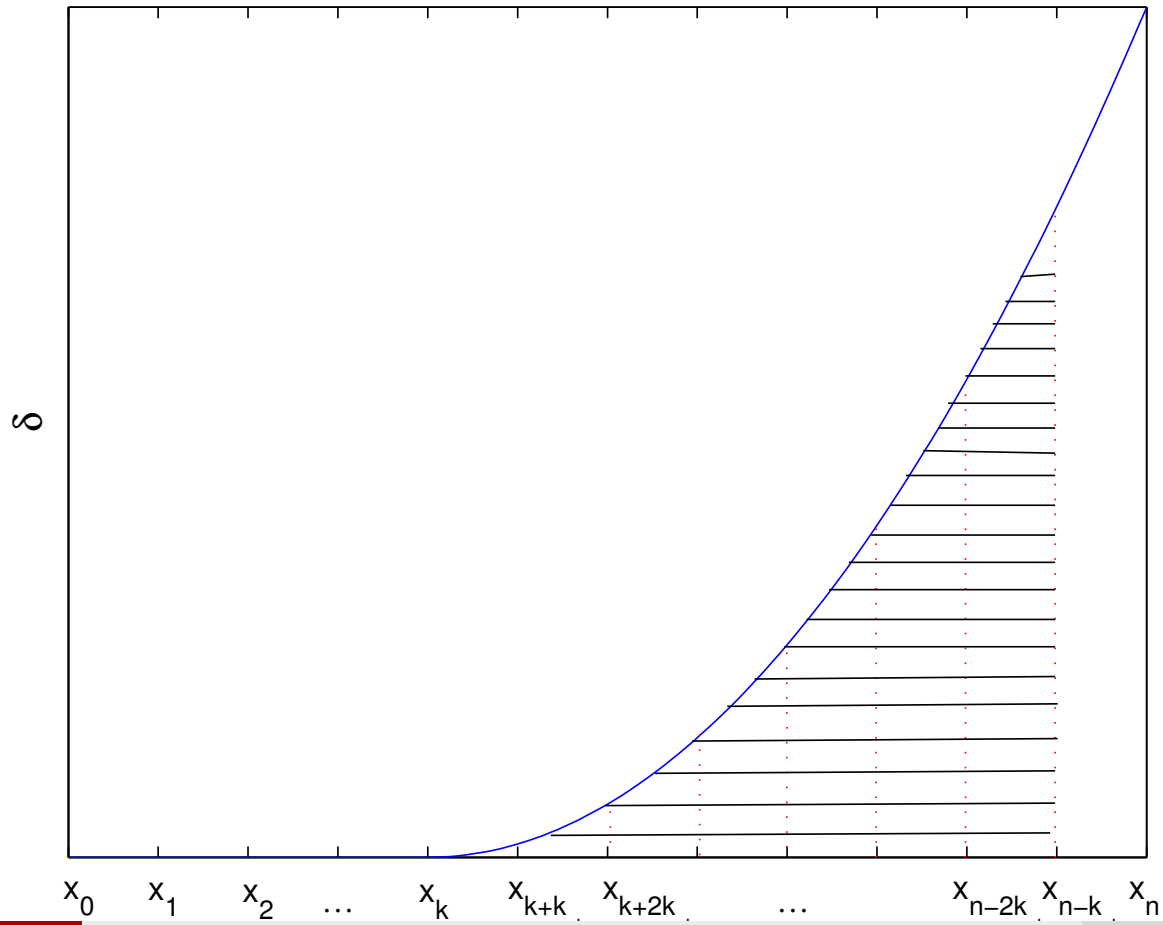
VNDS za 0-1 MIPs - ideja metode

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



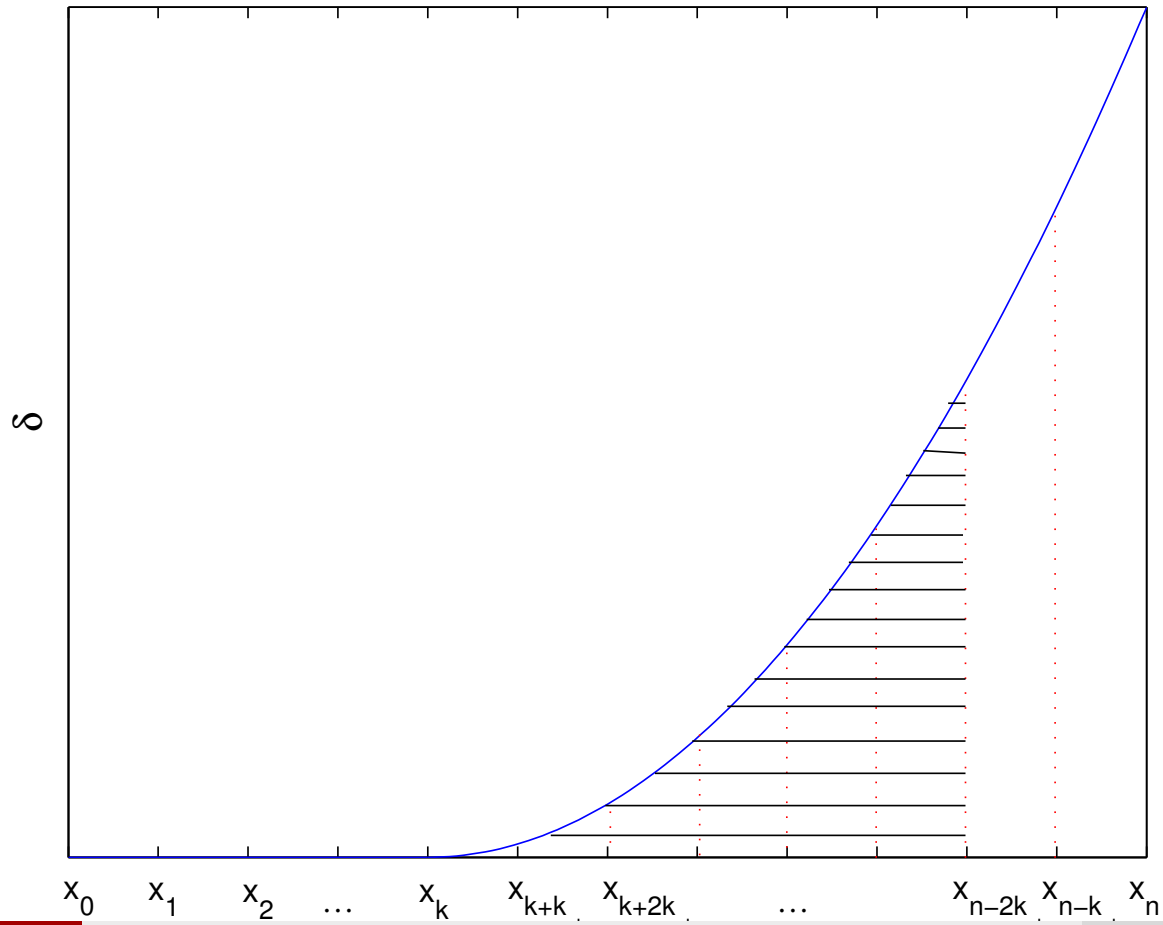
VNDS za 0-1 MIPs - ideja metode

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



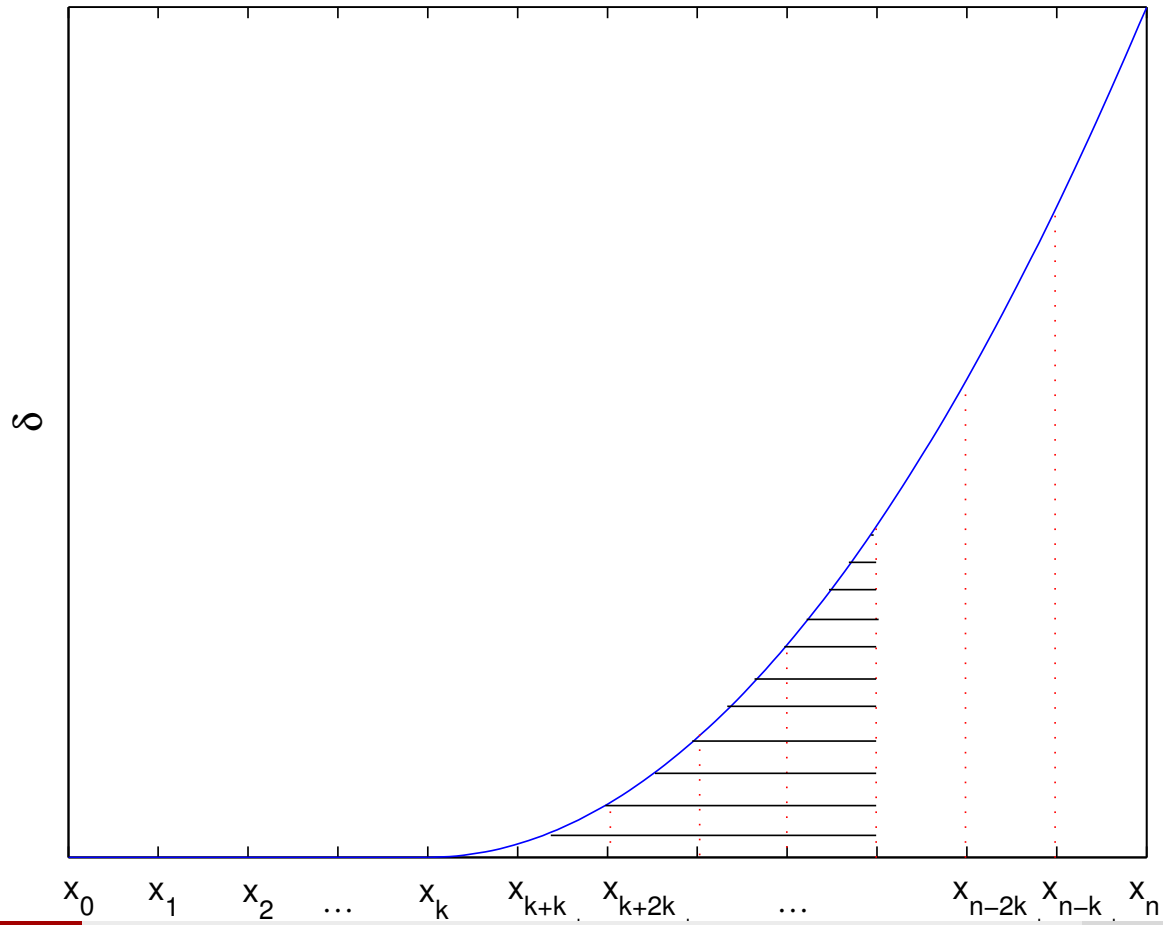
VNDS za 0-1 MIPs - ideja metode

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



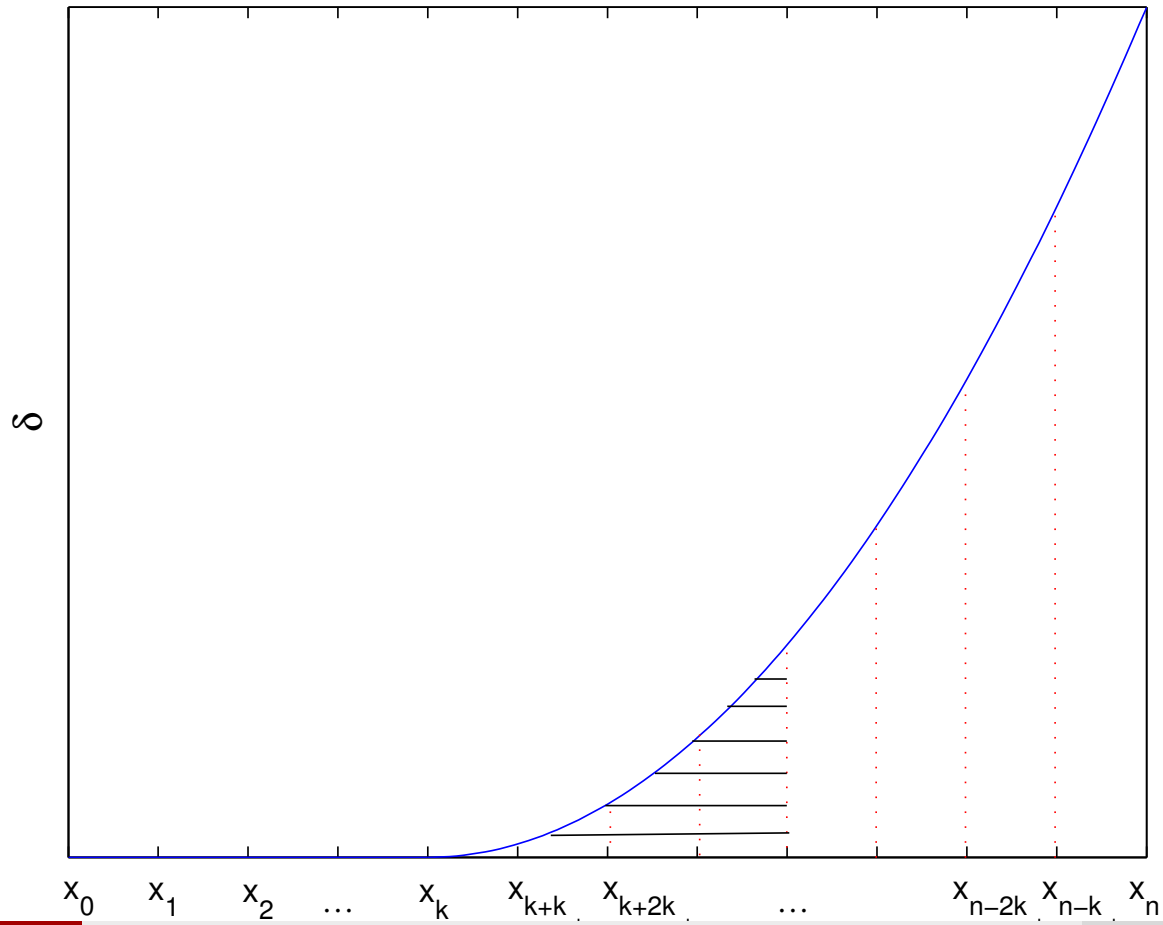
VNDS za 0-1 MIPs - ideja metode

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



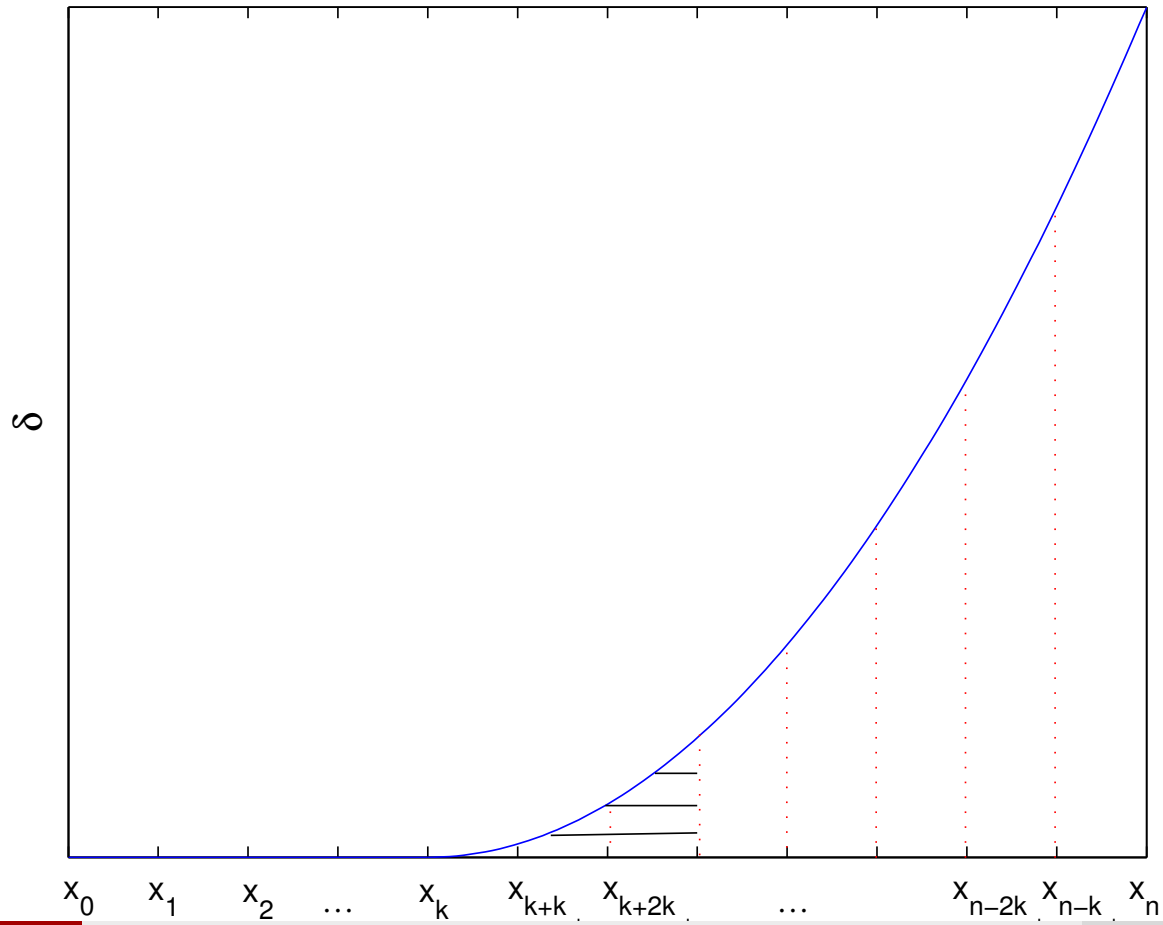
VNDS za 0-1 MIPs - ideja metode

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



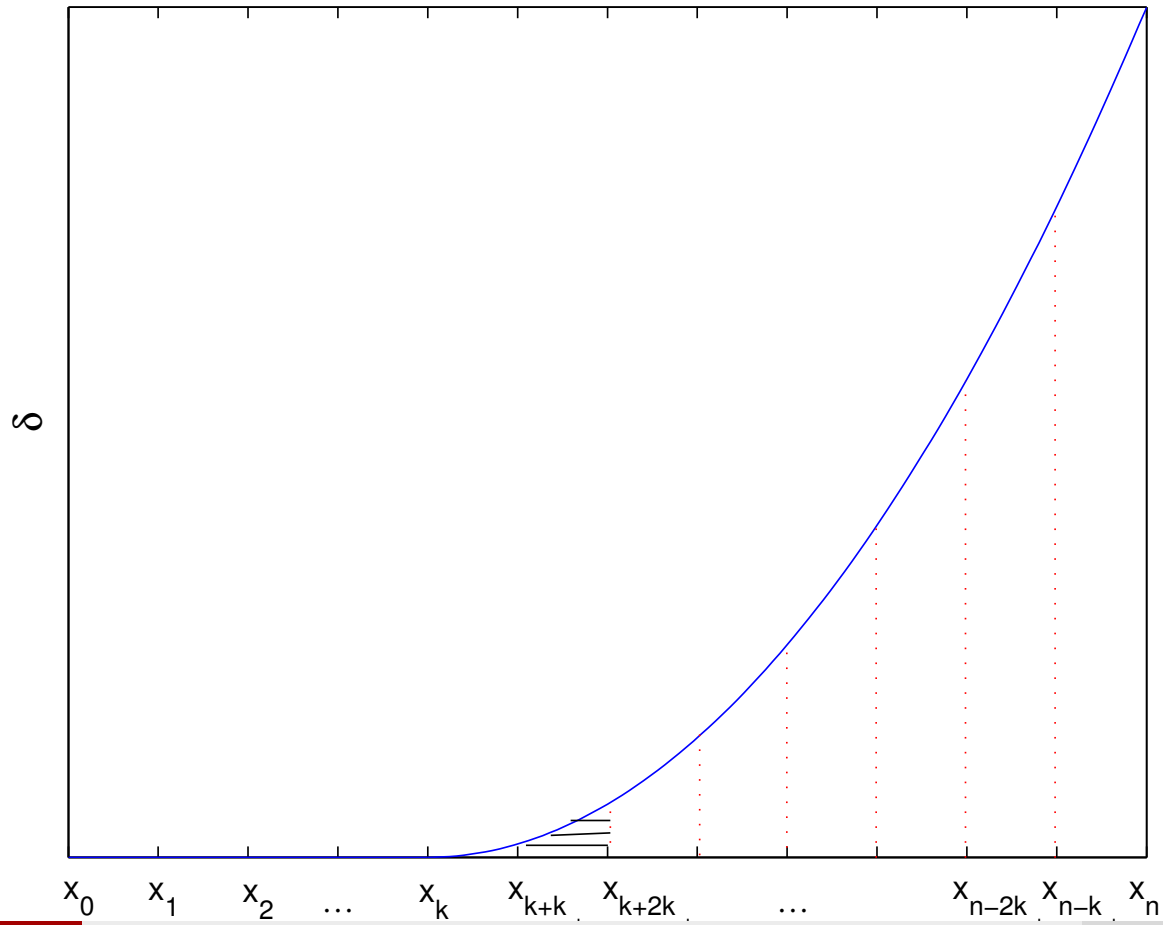
VNDS za 0-1 MIPs - ideja metode

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



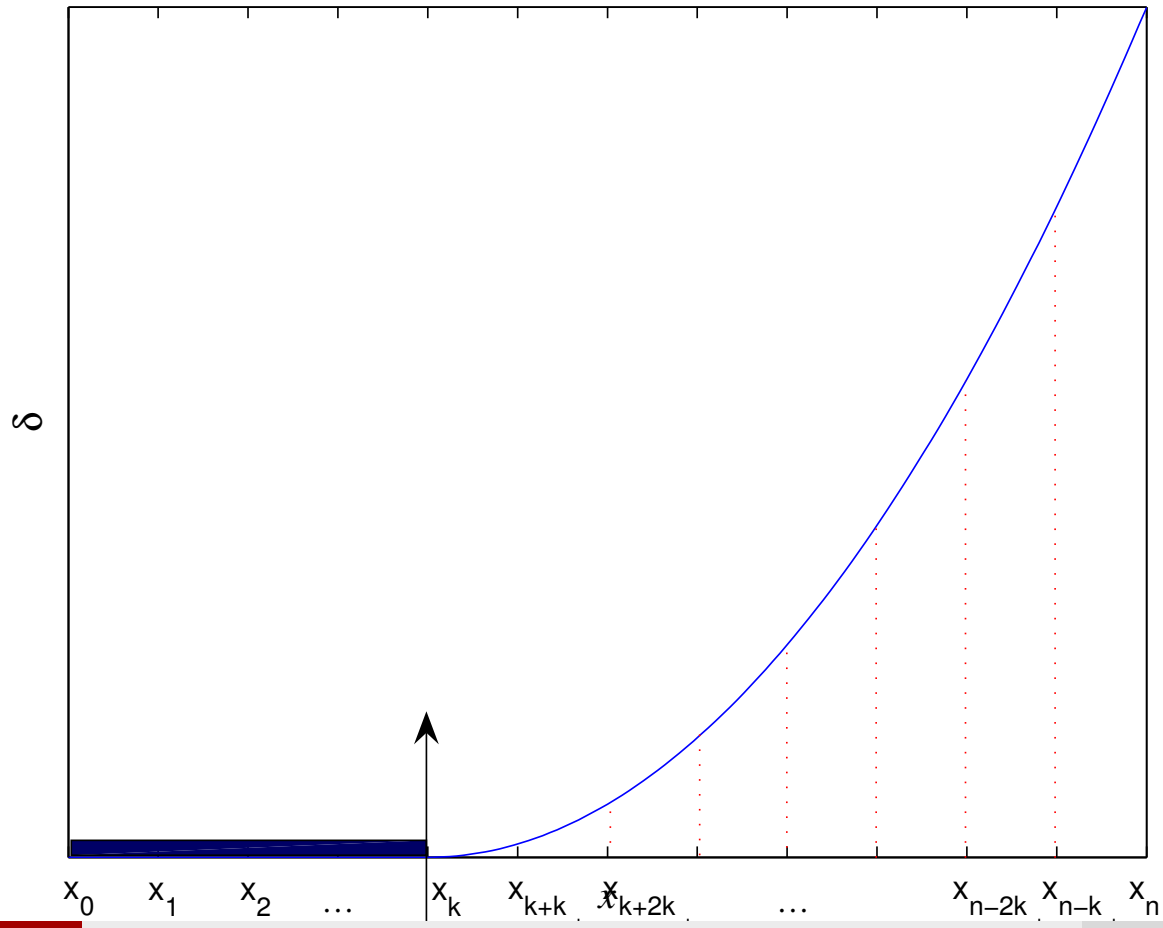
VNDS za 0-1 MIPs - ideja metode

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



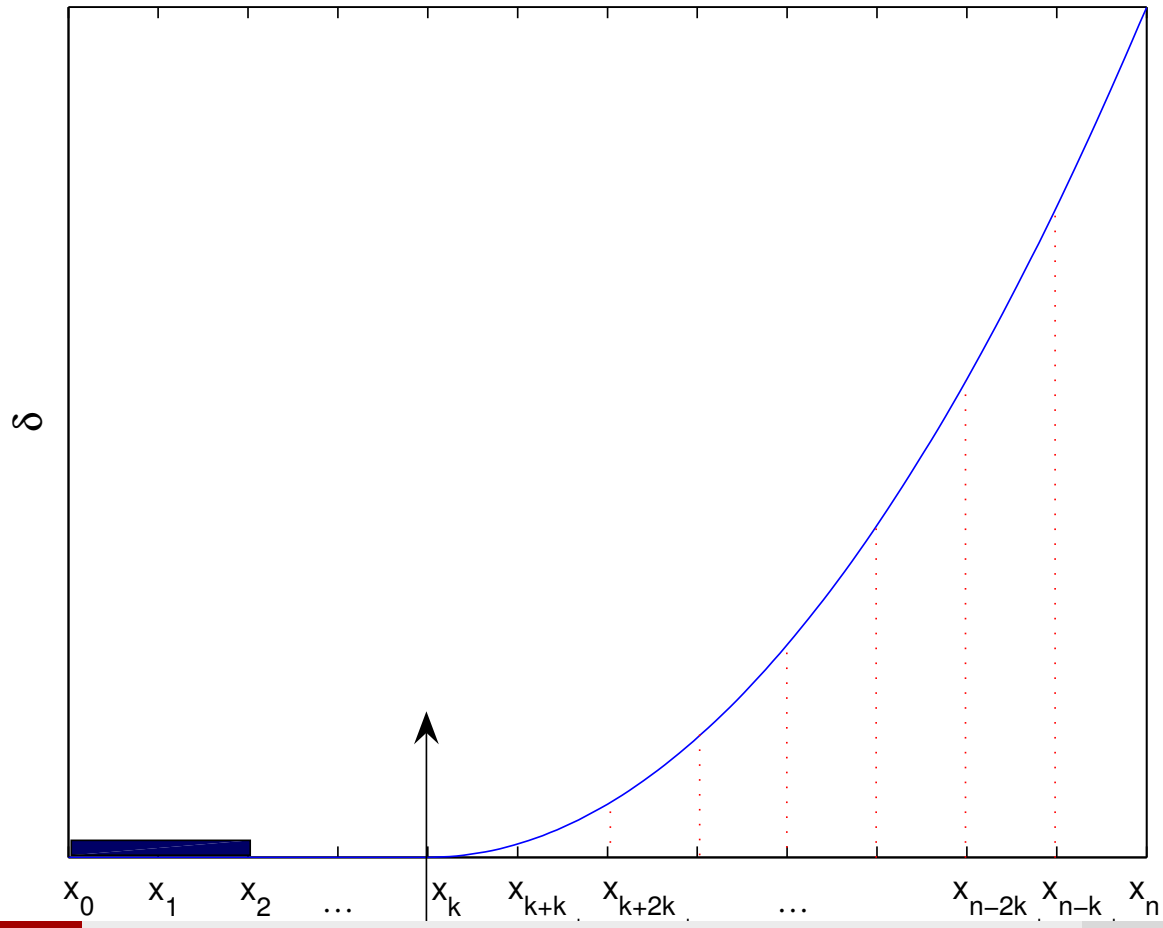
VNDS za 0-1 MIPs - ideja metode

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



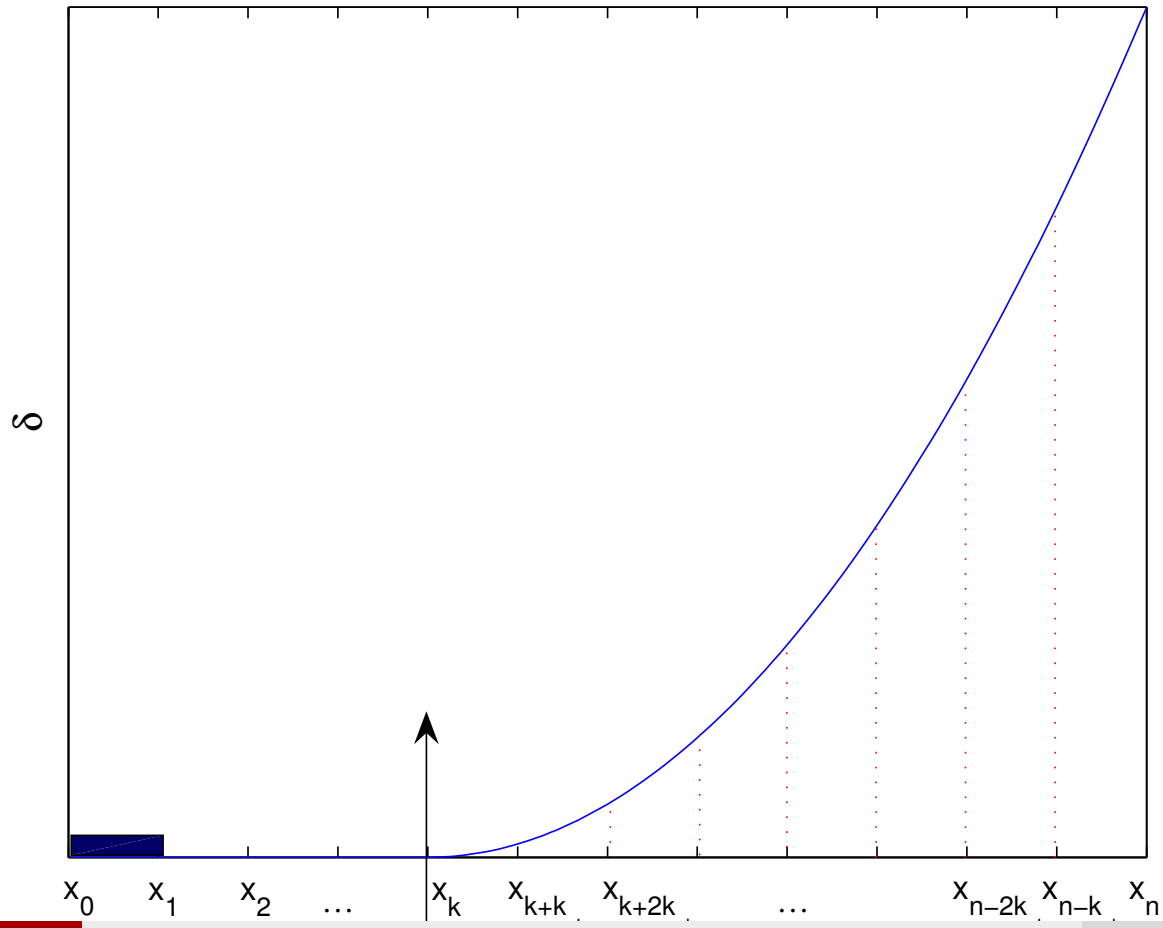
VNDS za 0-1 MIPs - ideja metode

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



VNDS za 0-1 MIPs - ideja metode

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



VNDS za 0-1 MIPs - pseudokod

```
VNDS-MIP(P, d, x, k_vnd)
    Find an optimal solution x̄ of LP(P);
    Set proceed1 ← true, proceed2 ← true;
    while (proceed1)
        δ_j ←| x_j − x̄_j |, j ∈ B; index variables x_j, j ∈ B,
        so that δ_1 ≤ δ_2 ≤ ... ≤ δ_p, p =| B |;
        Set n_d ←| {j ∈ B | δ_j ≠ 0} |, k_step ← [n_d/d], k ← p − k_step;
        while (proceed2 and k > 0)
            Fix values of x_1, ..., x_k;
            x' ← MIPSOLVE(P, x);
            Release x_1, ..., x_k;
            if (c^T x' > c^T x) then
                Q ← P; x ← VND-MIP(Q, k_vnd, x'); break;
            else
                if (k − k_step > p − n_d) then k_step ← max{[k/2], 1};
                Set k ← k − k_step;
            Update proceed1 and proceed2;
        endwhile
    endwhile
    return x.
```

Opšte napomene

- Efikasnost metoda zavisi od prirode problema i pravilnog podešavanja parametara;

Opšte napomene

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

Opšte napomene

- Efikasnost metoda zavisi od prirode problema i pravilnog podešavanja parametara;
- Metode ne zalaze u prirodu 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 metoda zavisi od prirode problema i pravilnog podešavanja parametara;
- Metode ne zalaze u prirodu 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 primera (VNDS-MIP);

Neke primene

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

Neke primene

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

Hvala na pažnji!

Pitanja?