Řešení digest problémů pomocí programování s omezujícími podmínkami

Hana Pařízková Ondřej Mička

Bioinformatické algoritmy, 2017

Cíle

- naprogramovat řešení digest problémů pomocí programování s omezujícími podmínkami (SICSTus Prolog)
- porovnat jejich efektivitu s klasickými algoritmy (v jazyce Python)

Restrikční mapování

- metoda sloužící k charakterizaci neznámého úseku DNA
- DNA naštěpena pomocí enzymů (tzv. **restrikčních endonukleáz**) na specifických pozicích, tzv. **restrikčních místech**
- cíl: určit přesné pozice restrikčních míst
- v praxi tři různé varianty:
  - Partial Digest Problem (PDP)
  - Simplified Partial Digest Problem (SPDP),
  - Double Digest Problem (DDP)
- všechny tři problémy jsou výpočetně „těžké“

Programování s omezujícími podmínkami (Constraint programming)

- druh deklarativního programování
- problém modelujeme pomocí proměnných a vztahů mezi nimi (podmínek)
- řešení nalezeno pomocí backtrackingu aj. prohledávacích metod
- vhodné k řešení NP úplných problémů
- např. knihovna clpfd pro jazyk Prolog

Partial Digest Problem

- cíl: nalézt množinu
  $RS = \{RS_1, RS_2, \ldots, RS_r\}, 0 = RS_1 < RS_2 < \cdots < RS_r$
  restrikčních míst
- vstup: množina $F = \{RS_j - RS_i | 1 \leq i < j \leq r\}$.
- klasické algoritmy: BruteForce, Skiena

PDP pomocí CSP

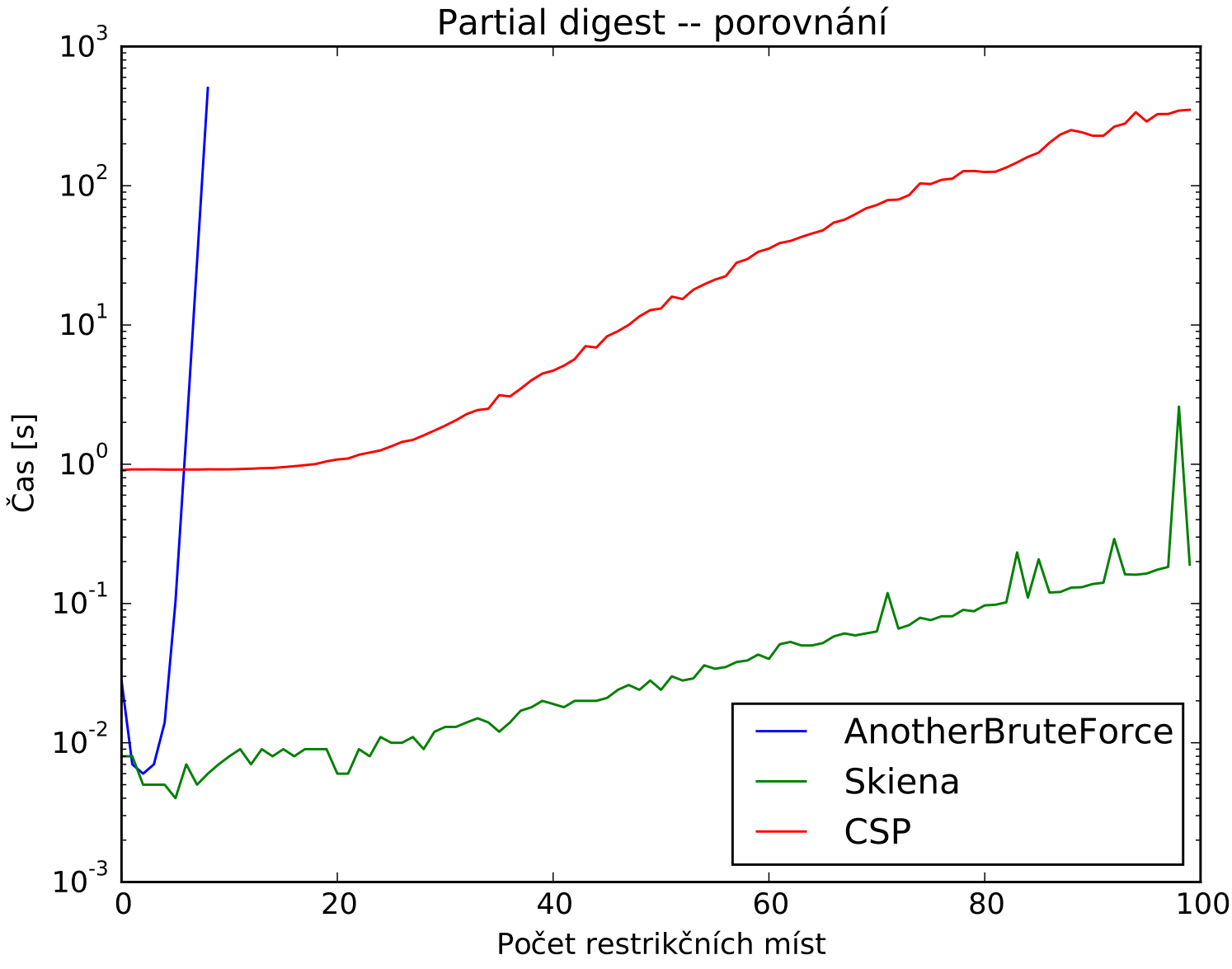
Proměnné:

- restrikční místa $RS = \{RS_1, RS_2, \ldots, RS_r\}$
- délka sekvence D

Podmínky:

- $|F| = \binom{|RS|}{2} = \frac{|RS|(|RS|-1)}{2}$, tzn. $|RS| = \frac{1+\sqrt{8*|F|+1}}{2}$
- $D = \max\{Fr | Fr \in F\}$, $0 = RS_1 < RS_2 < \cdots < RS_r = D$
- $F = \{RS_i - RS_j | 1 \leq j < i \leq r\}$
- $RS_2 \leq RS_r - RS_{r-1}$ (odstranění symetrie)

Efektivita algoritmů pro PDP



Simplified Partial Digest Problem

- cíl: nalézt množinu
  $RS = \{RS_1, RS_2, \ldots, RS_r\}, 0 = RS_1 < RS_2 < \cdots < RS_r$
  restrikčních míst
- vstup:
  - $LFrag = \{RS_i - RS_{i-1} | 1 \leq i \leq L\}$
  - $SFrag = \{RS_i, RS_L - RS_i | 1 \leq i < L\}$

SPDP pomocí CSP

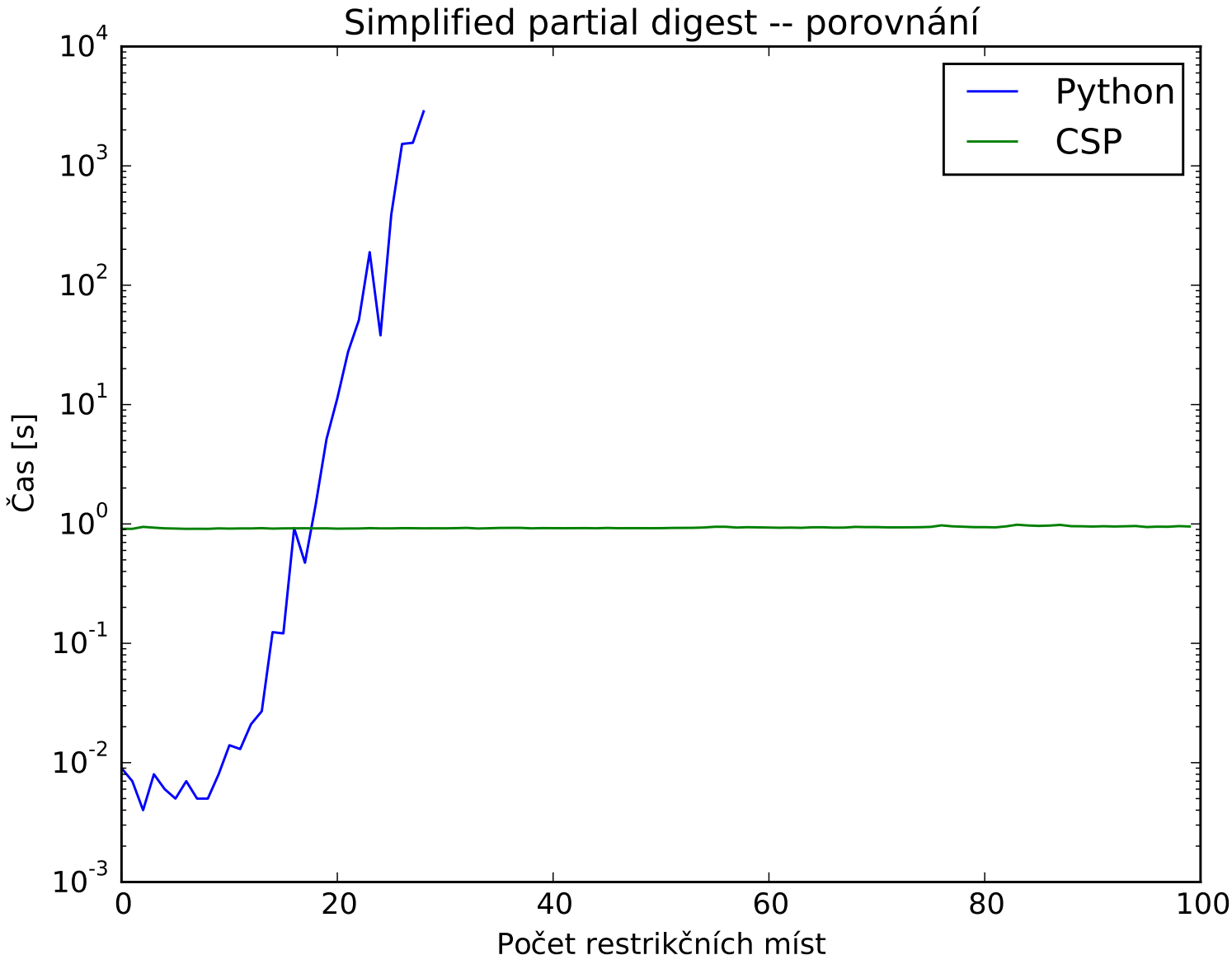
Proměnné:

- restrikční místa $RS = \{RS_1, RS_2, \ldots, RS_r\}$
- délka sekvence D, počet restrikčních míst (bez nuly) L
- $LOrder$ - vektor správného pořadí framentů z $LFrag$

Podmínky:

- $|LFrag| = L$, $|SFrag| = 2L - 2$, $|RS| = L + 1$
- $D = \min_{S \in SFrag} S + \max_{S \in SFrag} S = \sum_{d \in LFrag} d$
- $0 = RS_0 < RS_1 < RS_2 < \cdots < RS_{L-1} < RS_L = D$
- $\forall k \in [1, L] : \sum_{i=1}^{k} LOrder_i = RS_k$
- $\{RS_i, D - RS_i | 1 \leq i \leq L - 1\} = SFrag$
- $RS_1 \leq RS_L - RS_{L-1}$

Efektivita algoritmů pro SPDP



Double Digest Problem

- cíl: nalézt množiny
  $RA = \{RA_0, RA_1, \ldots, RA_n\}, 0 = RA_0 < RA_1 < \cdots < RA_n$ a
  $RB = \{RB_0, RB_1, \ldots, RB_m\}, 0 = RB_0 < RB_1 < \cdots < RB_m$,
  $RC = RA \cup RB$ restrikčních míst
- vstup:
  - $AFrag = \{RA_i - RA_{i-1} | 1 \leq i \leq n\}$
  - $BFrag = \{RB_i - RB_{i-1} | 1 \leq i \leq m\}$
  - $CFrag = \{RC_i - RC_{i-1} | 1 \leq i \leq k\}$

DDP pomocí CSP

Proměnné:

- restrikční místa $RA = \{RA_0, RA_1, \ldots, RS_n\}$, RB a RC analogicky
- fragmenty $FA = \{FA_1, FA_2, \ldots, FA_n\}$, FB a FC analogicky
- zobrazení restrikčních míst z A do C
  $MA = \{MA_1, MA_2, \ldots, MA_n\}$, MB analogicky

Podmínky:

- $\forall i \, \exists j : FA_i = AFrag_j$, analogicky pro FB a FC
- $RA_i = \sum_{j=1}^{i} FA_j$, analogicky pro FB a FC
- $MA_1 < MA_2 < \ldots MA_n$, $MB_1 < MB_2 < \ldots MB_n$
- $MA_i = j \Leftrightarrow RA_i = RC_j$, analogicky pro MB
- $|MA \cup MB| = |RC|$

Závěr

\+

- CSP může dosahovat výsledků srovnatelných (i lepších) než klasické algoritmy
- dobrý poměr programátorská náročnost/efektivita
- použitelné i na problémy, kde klasické algoritmy selhávají

−

- u CSP nelze teoreticky stanovit časovou náročnost
- na velká data spíše vhodnější klasické algoritmy

Zdrojové kódy ke stažení:
`https://ulozto.cz/!jHY1rrHxBahw/cspdigest-zip`

Dotazy?