Solving relaxations of MAP-MRF problems: Combinatorial in-face Frank-Wolfe directions

Vladimir Kolmogorov

Institute of Science and Technology Austria (ISTA)

WED-AM-358

MAP-MRF problem

- Goal: minimize function of discrete variables

$$f(\boldsymbol{x}) = \sum_{p} f_p(x_p) + \sum_{p,q} f_{pq}(x_p, x_q)$$

- Popular approach: solve LP relaxation
- Extensive literature on specialized LP solvers
  - Block-coordinate ascent on the dual (may get stuck…)
    - TRW-S [K 06] / SRMP [K '15], MPLP [Globerson-Jaakkola'08], DMM [Shekhovtsov'16], MPLP++ [Tourani et al.'18], SPAM [Tourani et al.'20], …
  - Converge to LP optimum
    - [Ravikumar et al.'10], [Jojic et al.'10], [Savchynskyy et al.'11,'12], [Schmidt et al.'11], [Komodakis et al.11], [Martins et al.'11], [Luong et al.'12], [Schwing et al.'12], …
  - Frank-Wolfe based approach [Swoboda-K.'19], [K.-Pock'21]
    - this work: efficient FW implementation
    - *(combinatorial) in-face FW directions*
    - state-of-the-art LP solver for some applications

MAP-MRF via Frank-Wolfe (FW)

- More generally: minimize

$$f(\boldsymbol{x}) = \sum_{A \subseteq [n]} f_A(x_A)$$

  – assumption: can solve

$$\arg\min_{x_A} \left[f_A(x_A) + \langle x_A, y_A \rangle \right] \qquad \forall\, y_A$$

“min-oracle”

    - e.g. tree-structured MAP-MRF

  – goal: solve certain Lagrangian relaxation

MAP-MRF via Frank-Wolfe (FW)

- More generally: minimize

$$f(\boldsymbol{x}) = \sum_{A \subseteq [n]} f_A(x_A)$$

  – assumption: can solve

$$\arg\min_{x_A} \; [f_A(x_A) + \langle x_A, y_A \rangle] \qquad \forall\, y_A$$

"min-oracle"

    - e.g. tree-structured MAP-MRF

  – goal: solve certain Lagrangian relaxation

- [Swoboda-K.'19]: proximal point method + FW algorithm

MAP-MRF via Frank-Wolfe (FW)

- More generally: minimize

$$f(\boldsymbol{x}) = \sum_{A \subseteq [n]} f_A(x_A)$$

  - assumption: can solve

$$\arg\min_{x_A} \; [f_A(x_A) + \langle x_A, y_A \rangle] \qquad \forall\, y_A \qquad \text{“min-oracle”}$$

    - e.g. tree-structured MAP-MRF
  - goal: solve certain Lagrangian relaxation

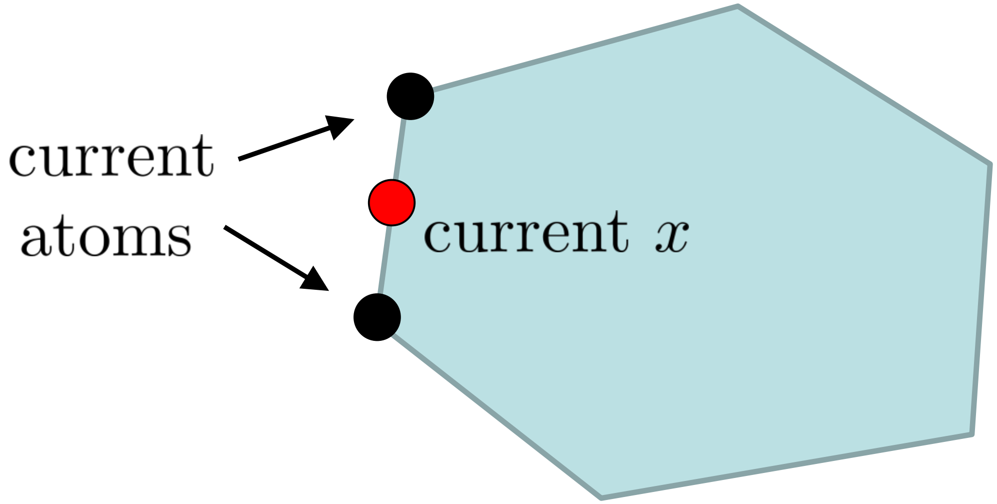
- [Swoboda-K.’19]: proximal point method + FW algorithm
- [K.-Pock’21]: accelerated version, convergence analysis
  - $O(1/n^2)$ dual gap after $O(n \log n)$ oracle calls
    (* *under technical assumptions*)
  - best known rate for iterative algorithms

This work: efficient implementation of FW

- Block-coordinate FW [Lacoste-Julien et al.'13]
- Cache atoms [Joachims et al.'09], [Shah et al.'15], [Osokin et al.16]
- Main improvement: *in-face FW directions* [Freund et al.'17]
  - implementation for combinatorial oracles
  - which data structures are needed?

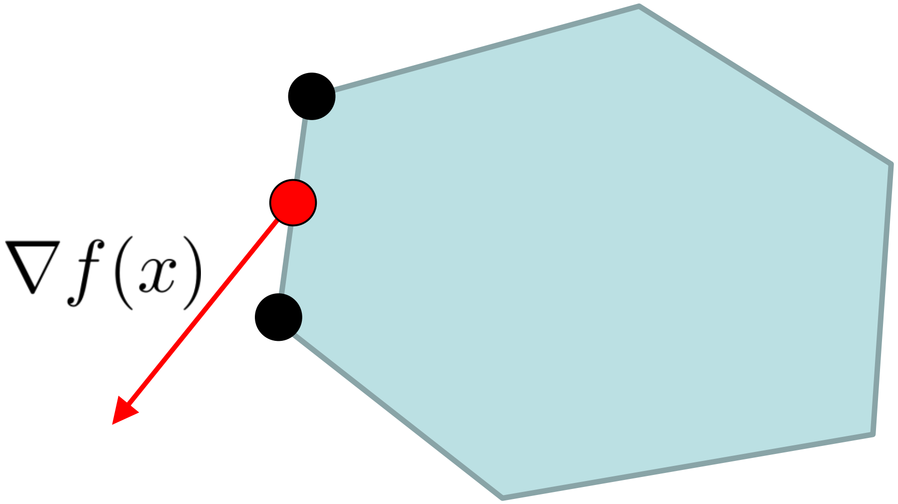
FW overview

- Main subproblem: optimize convex quadratic function $f(x)$ over combinatorial polytope



FW overview

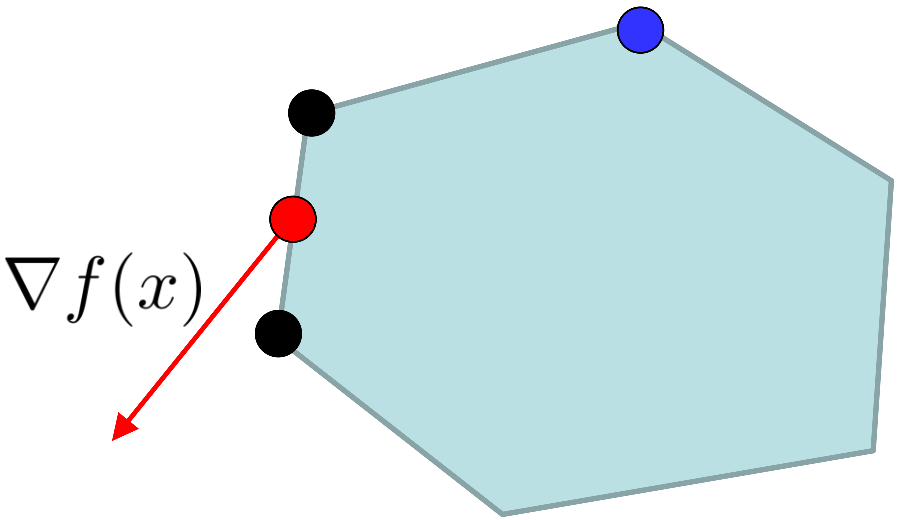
- Main subproblem: optimize convex quadratic function $f(x)$ over combinatorial polytope



- FW step:
  - linearize objective at x

FW overview

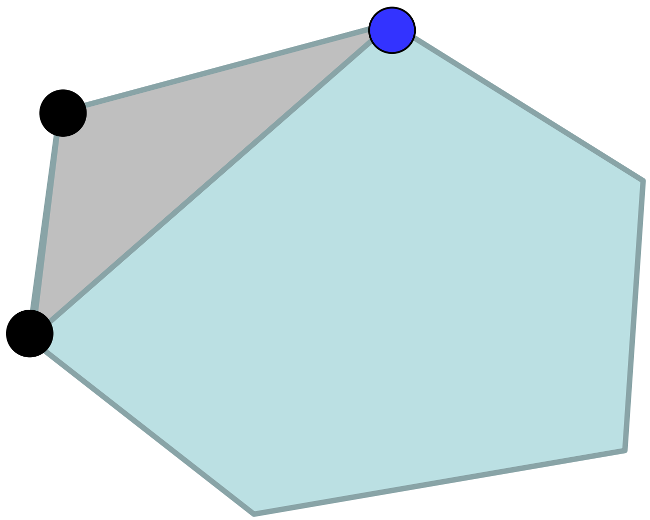
- Main subproblem: optimize convex quadratic function $f(x)$ over combinatorial polytope



- FW step:
  - linearize objective at x
  - call min-oracle, add result to cache

FW overview

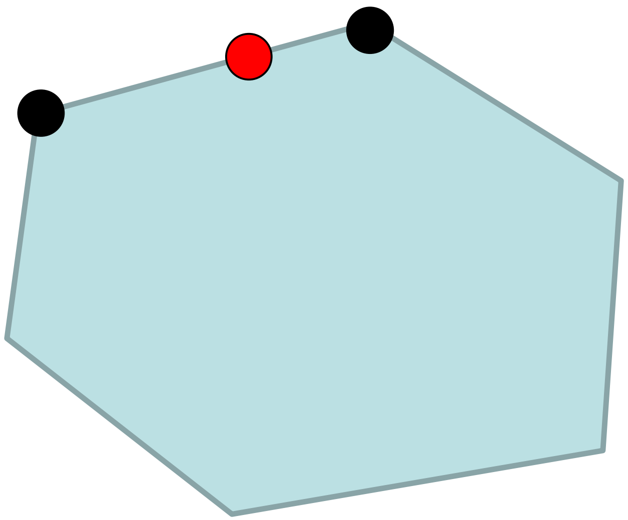
- Main subproblem: optimize convex quadratic function $f(x)$ over combinatorial polytope



- FW step:
  - linearize objective at x
  - call min-oracle, add result to cache
  - re-optimize f over atoms

FW overview

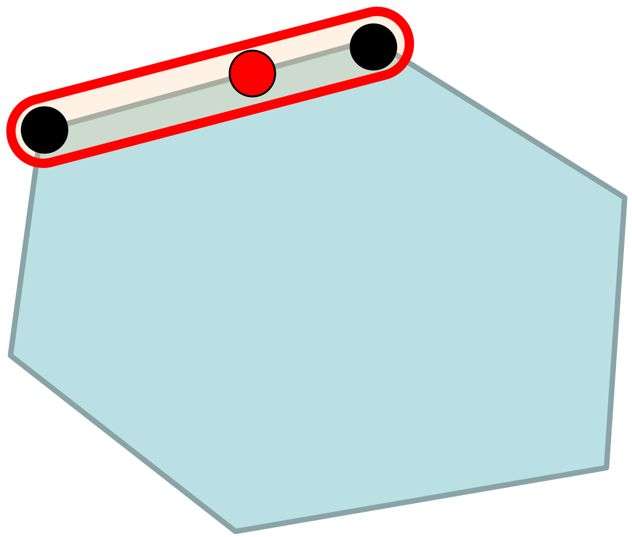
- Main subproblem: optimize convex quadratic function $f(x)$ over combinatorial polytope



- FW step:
  - linearize objective at x
  - call min-oracle, add result to cache
  - re-optimize f over atoms

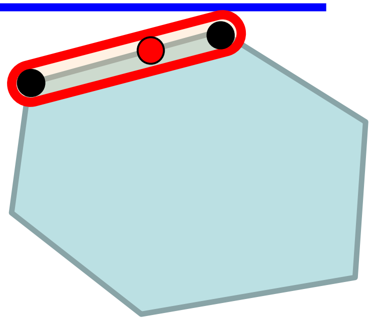
In-face FW directions [Freund et al.'17]

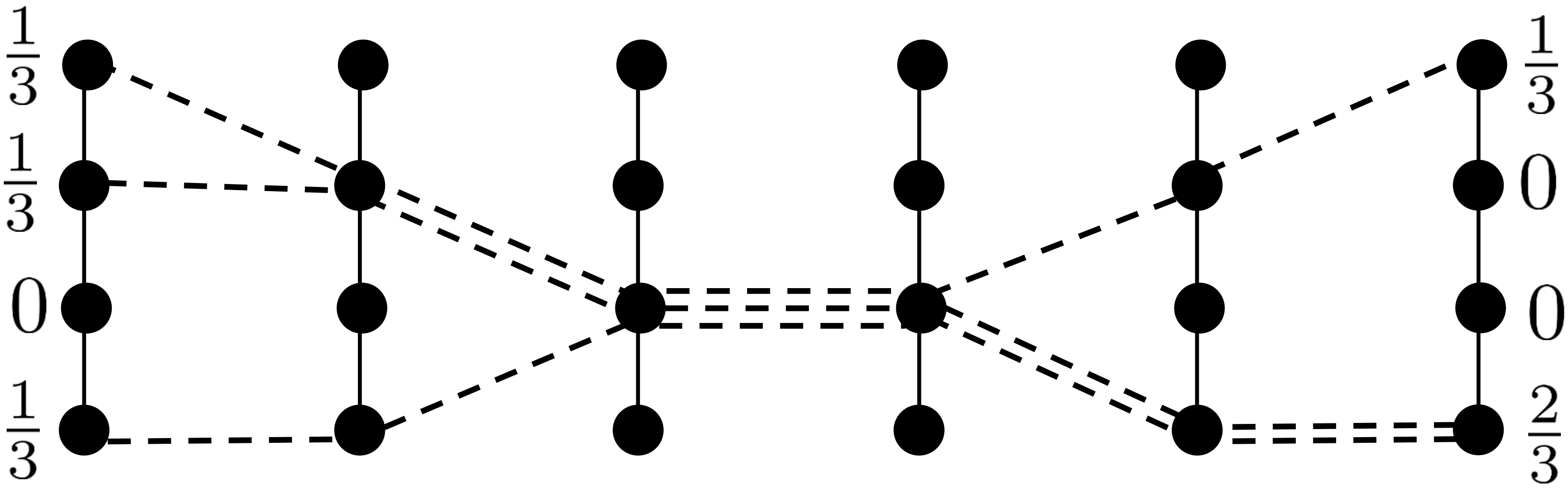
- Take (minimal) face of the polytope containing x
- Run several FW steps for smaller polytope



Implementation for tree-structured MAP-MRF

$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$

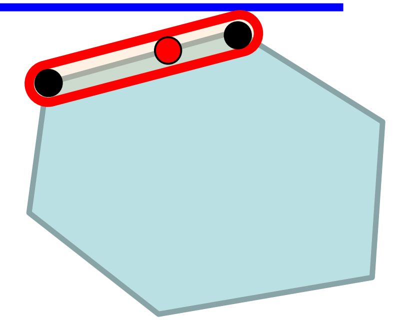


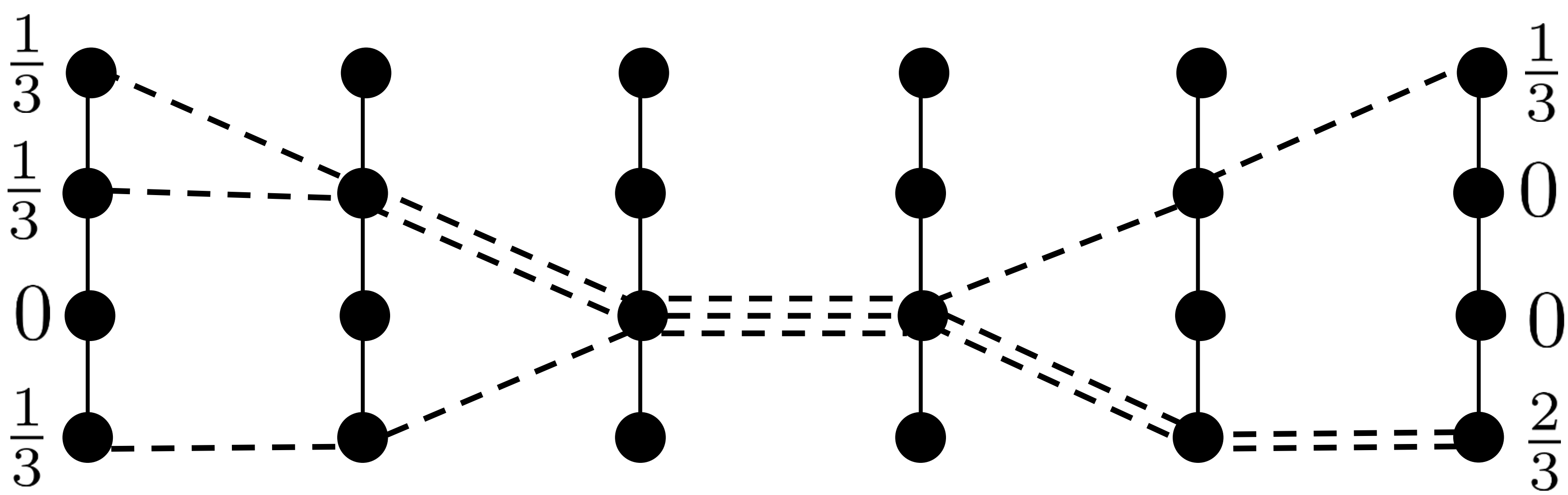


Implementation for tree-structured MAP-MRF

$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$

node variables



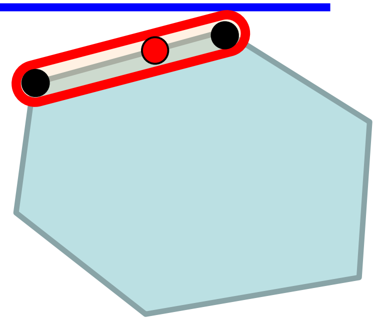


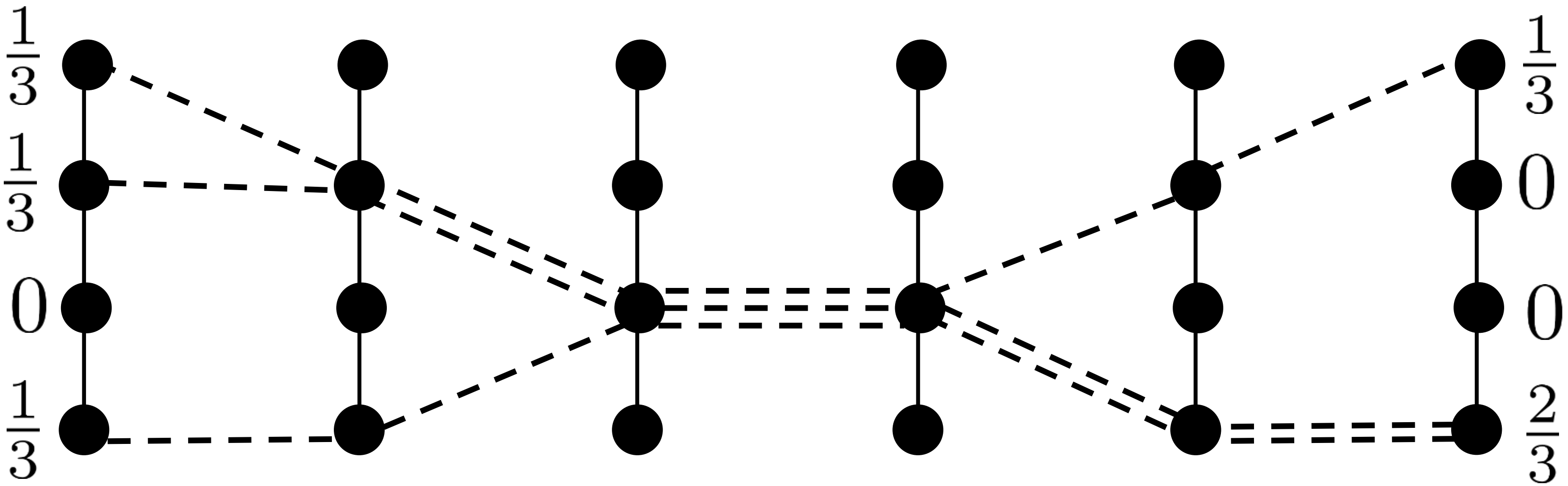
Implementation for tree-structured MAP-MRF

$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$

quadratic

node variables





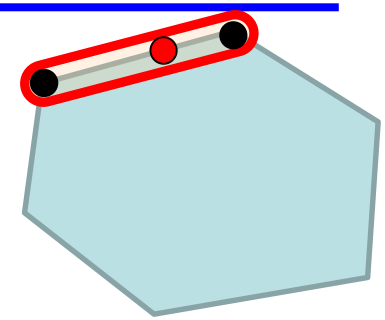
Implementation for tree-structured MAP-MRF

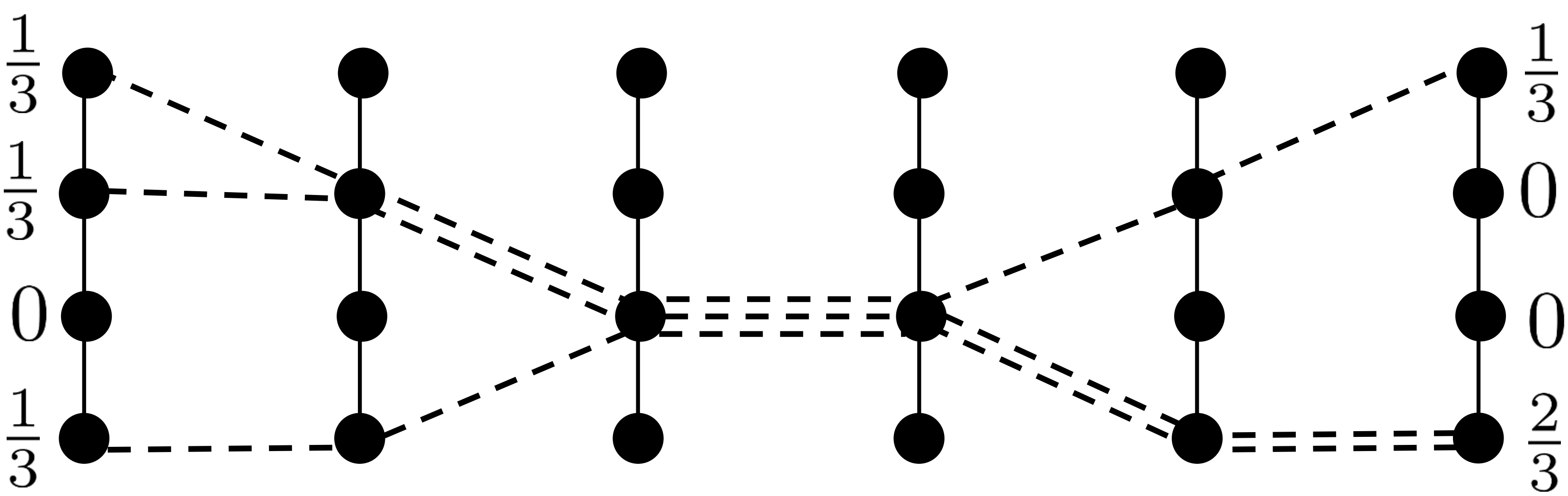
$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$

quadratic

node variables

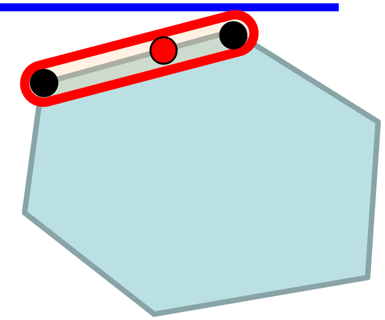
total cost of pairwise terms



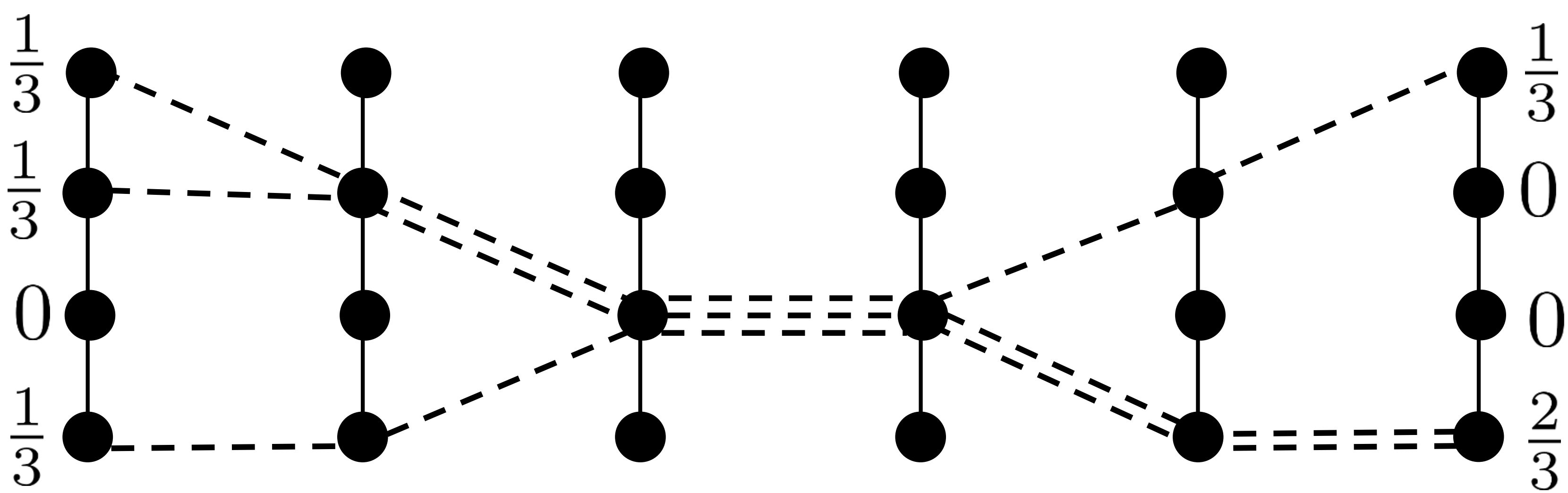


Implementation for tree-structured MAP-MRF

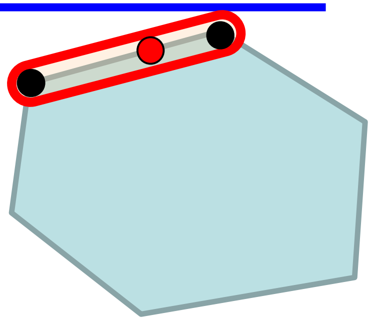
$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$



- Face: fix some vars to zero $(x_{p;a} = 0)$

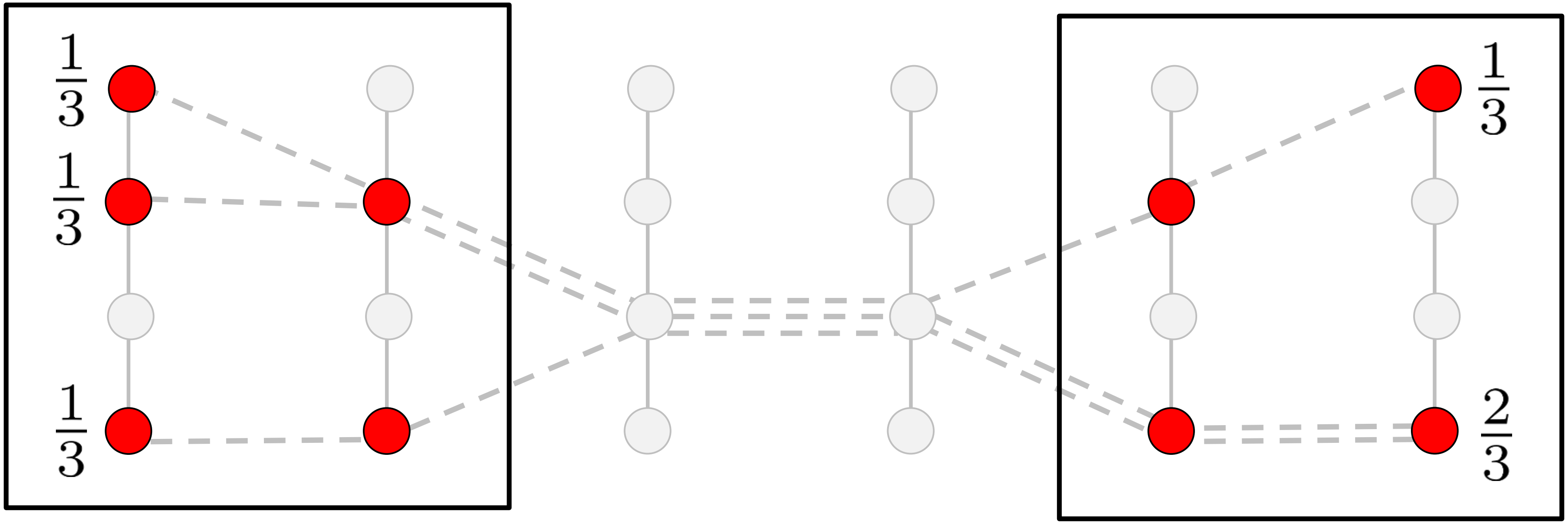


Implementation for tree-structured MAP-MRF



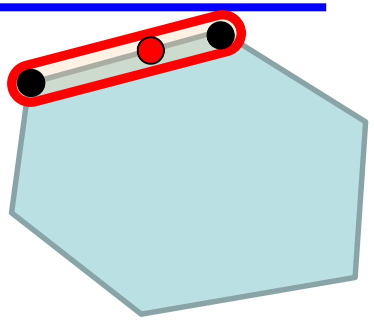
$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$

- Face: fix some vars to zero $(x_{p;a} = 0)$
- May split problem into independent subproblems

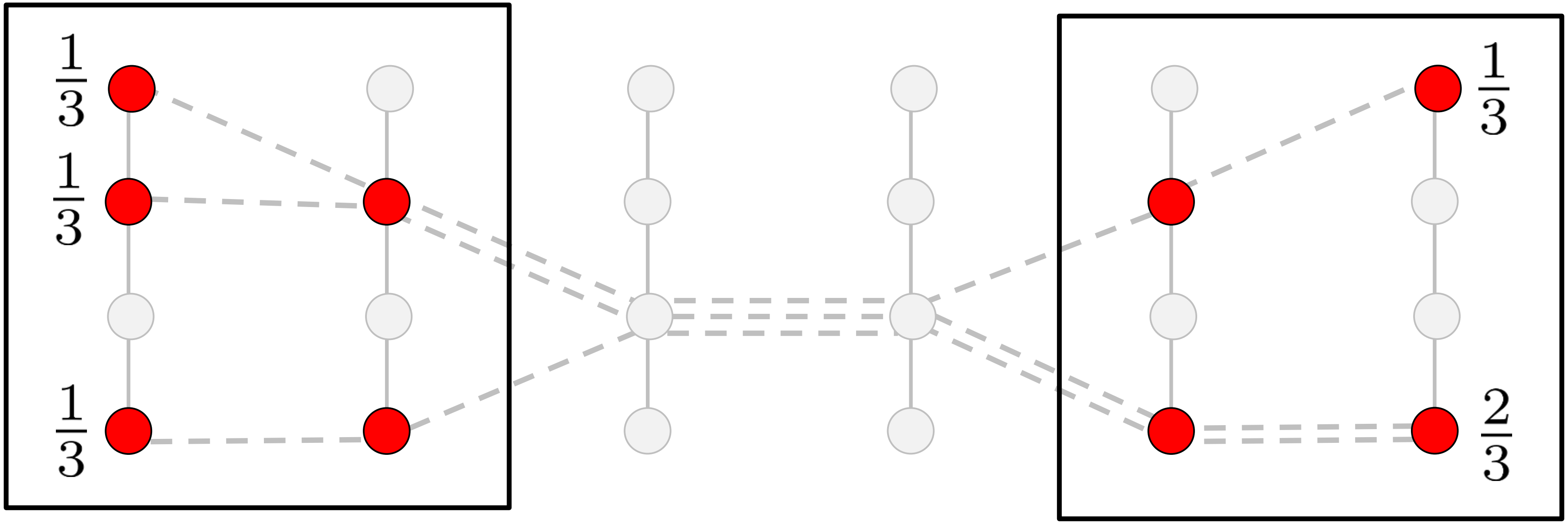


Implementation for tree-structured MAP-MRF

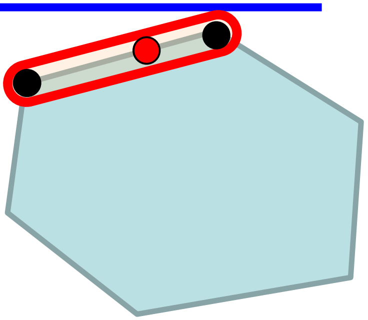
$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$



- Face: fix some vars to zero $(x_{p;a} = 0)$
- May split problem into independent subproblems
  – use block-coordinate FW for each new subproblem

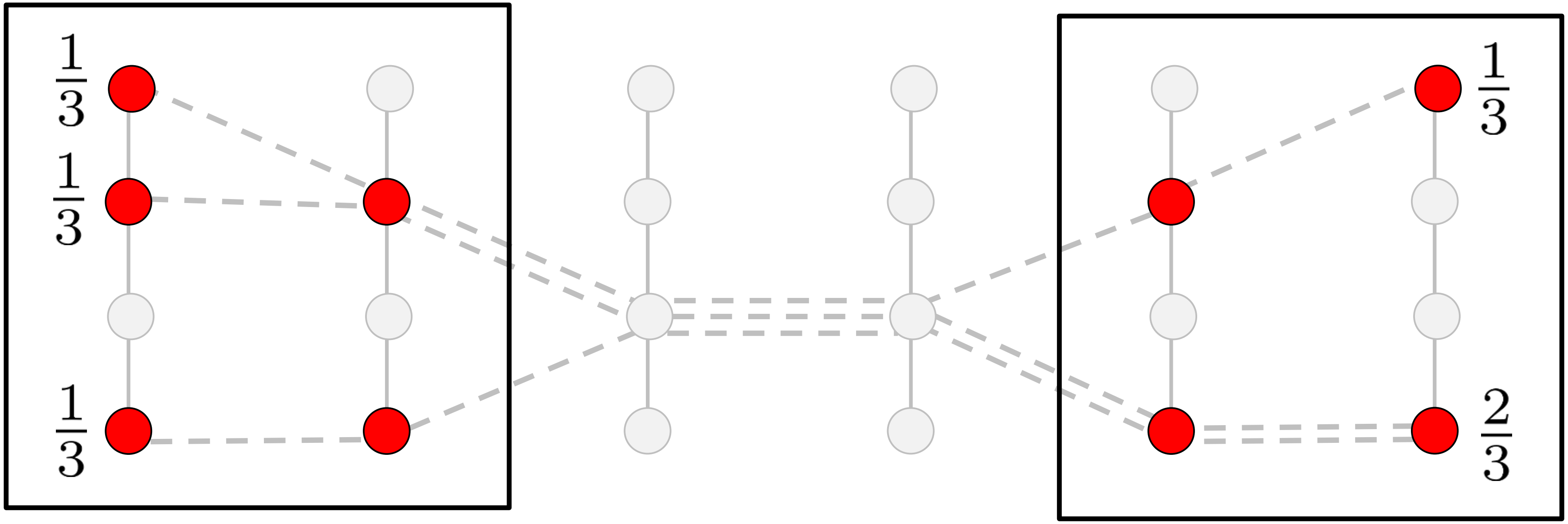


Implementation for tree-structured MAP-MRF



$$f(x) = \sum_{p;a} f_{p;a}(x_{pa}) + x_{\texttt{edge cost}}$$

- Face: fix some vars to zero $(x_{p;a} = 0)$
- May split problem into independent subproblems
  - use block-coordinate FW for each new subproblem
  - efficient transformation of atoms?



"Subproblem" data structure

- "Parent" and "child" subproblems
- Functions:

$\texttt{AtomToVector}(a)$

$\texttt{DotProduct}(a, g)$

$\ldots$

$\texttt{MinOracle}(a, g)$

$\ldots$

$\texttt{Contract}(x, s)$ — for parent subproblems

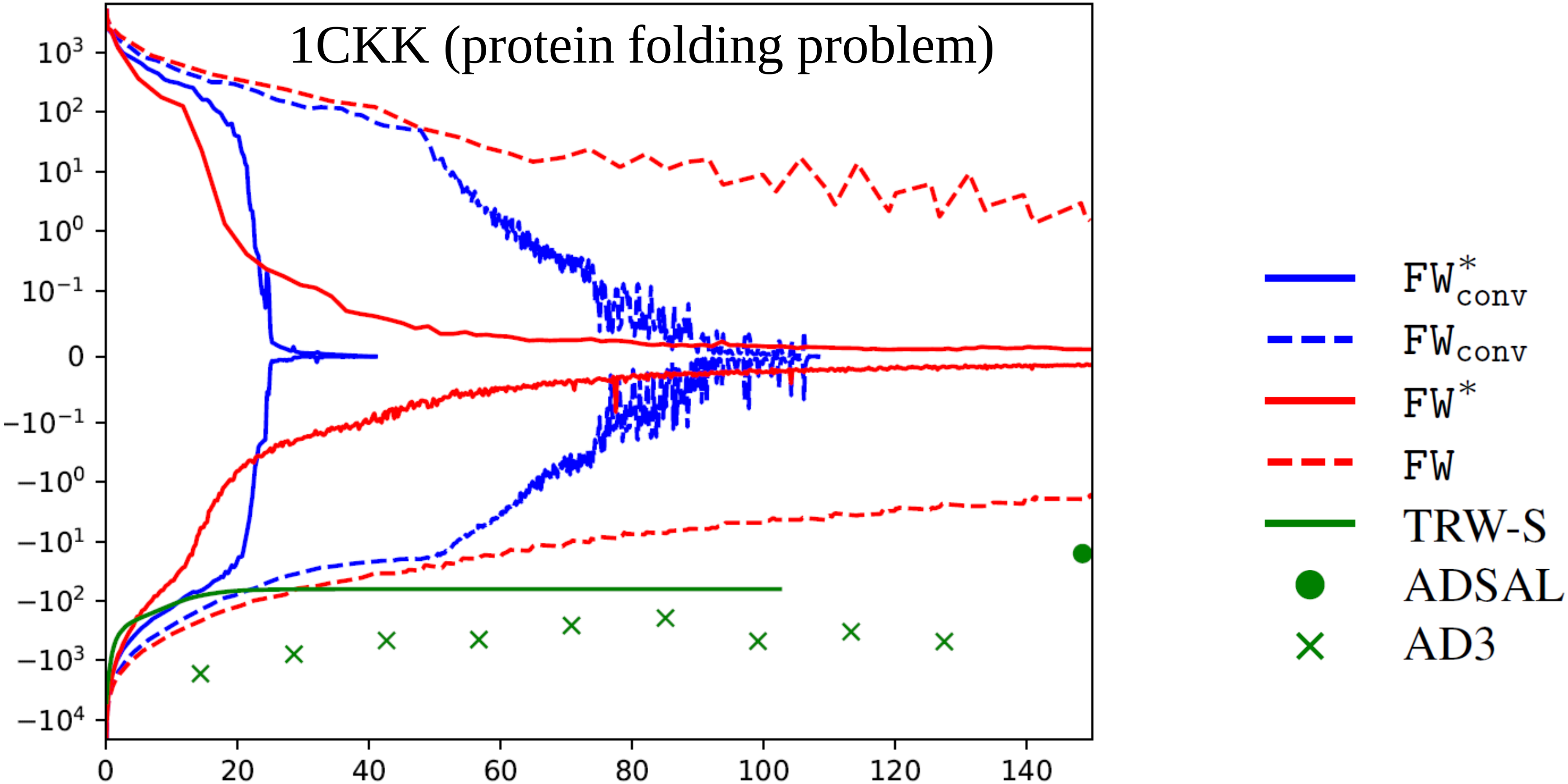
$\texttt{SetAtom}(a)$

$\texttt{GetAtom}(a)$

} for parent & child subproblems

$\ldots$

Results



1CKK (protein folding problem)
10³
10²
10¹
10⁰
10⁻¹
0
−10⁻¹
−10⁰
−10¹
−10²
−10³
−10⁴
0
20
40
60
80
100
120
140
FW*conv
FWconv
FW*
FW
TRW-S
ADSAL
AD3

Summary

Goal: solve Lagrangian relaxation of $$\min_{\boldsymbol{x}} \sum_{A \subseteq [n]} f_A(x_A)$$

- Added in-face directions to FW-based approach
  - specified abstract data structure for combinatorial subproblems
  - implementation for tree-structured MAP-MRF problems
  - current state-of-the-art LP solver for some classes of problems
  - C++ code publicly available