Discrete Inference and Learning

Lecture 4

Primal-dual schema, dual decomposition

D. Khuê Lê-Huu, Nikos Paragios
{khue.le, nikos.paragios}@centralesupelec.fr

Slides courtesy of Nikos Komodakis

At a glance

- **Last lecture:**

Tree-reweighted Message Passing (TRW)

⟹ LP relaxation and its dual.

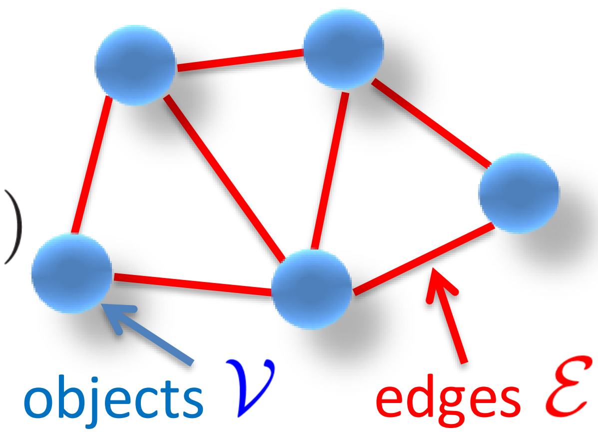
- **This lecture:**

More on **duality** theory: primal-dual schema, dual decomposition.

Part I
Recap: MRFs and Convex Relaxations

Discrete MRF optimization

- Given:
  - Objects $\mathcal{V}$ from a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - Discrete label set $\mathcal{L}$



- Assign labels (to objects) that minimize MRF energy:

$$\min_{\{x_p\}} \sum_{p \in \mathcal{V}} \underbrace{\bar{g}_p(x_p)}_{\text{unary potential}} + \sum_{pq \in \mathcal{E}} \underbrace{\bar{f}_{pq}(x_p, x_q)}_{\text{pairwise potential}}$$

Discrete MRF optimization

- Extensive research for more than 20 years

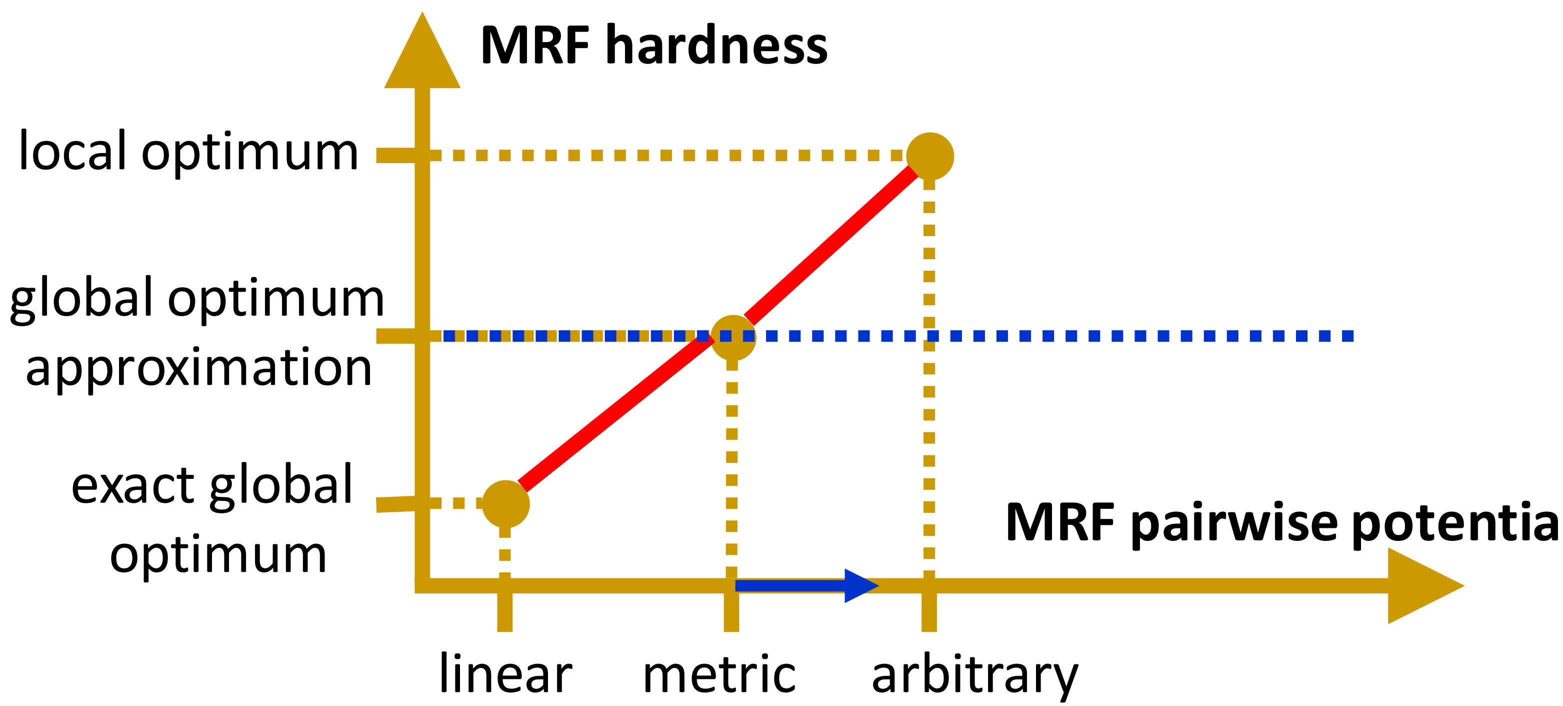
- MRF optimization ubiquitous in computer vision
  - segmentation — stereo matching
    optical flow — image restoration
    image completion — object detection/localization
    ...

- and beyond
  - medical imaging, computer graphics, digital communications, physics…

- Really powerful formulation

How to handle MRF optimization?

- Unfortunately, discrete MRF optimization is extremely hard (a.k.a. NP-hard)
  - E.g., highly non-convex energies



How to handle MRF optimization?

- Unfortunately, discrete MRF optimization is extremely hard (a.k.a. NP-hard)
  - E.g., highly non-convex energies
- So what do we do?
  - Is there a principled way of dealing with this problem?
- Well, first of all, we don't need to panic.
  Instead, we have to stay calm and **RELAX!**
- Actually, this idea of relaxing may not be such a bad idea after all…

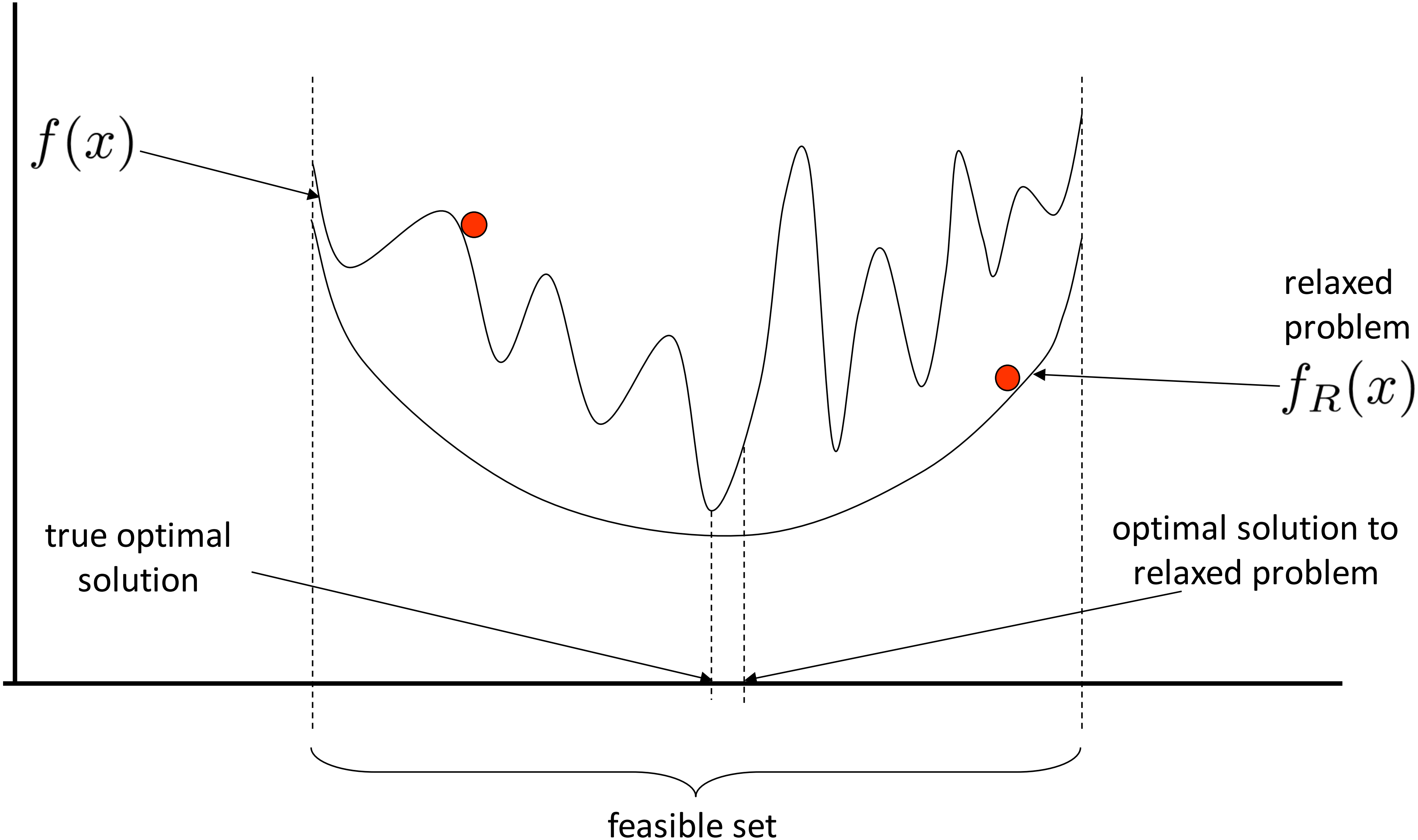
The relaxation technique

- Very successful technique for dealing with difficult optimization problems

- It is based on the following simple idea:
  - try to approximate your original difficult problem with another one (the so called **relaxed problem**) which is easier to solve

- Practical assumptions:
  - Relaxed problem must always be easier to solve
  - Relaxed problem must be related to the original one

The relaxation technique



How do we find easy problems?

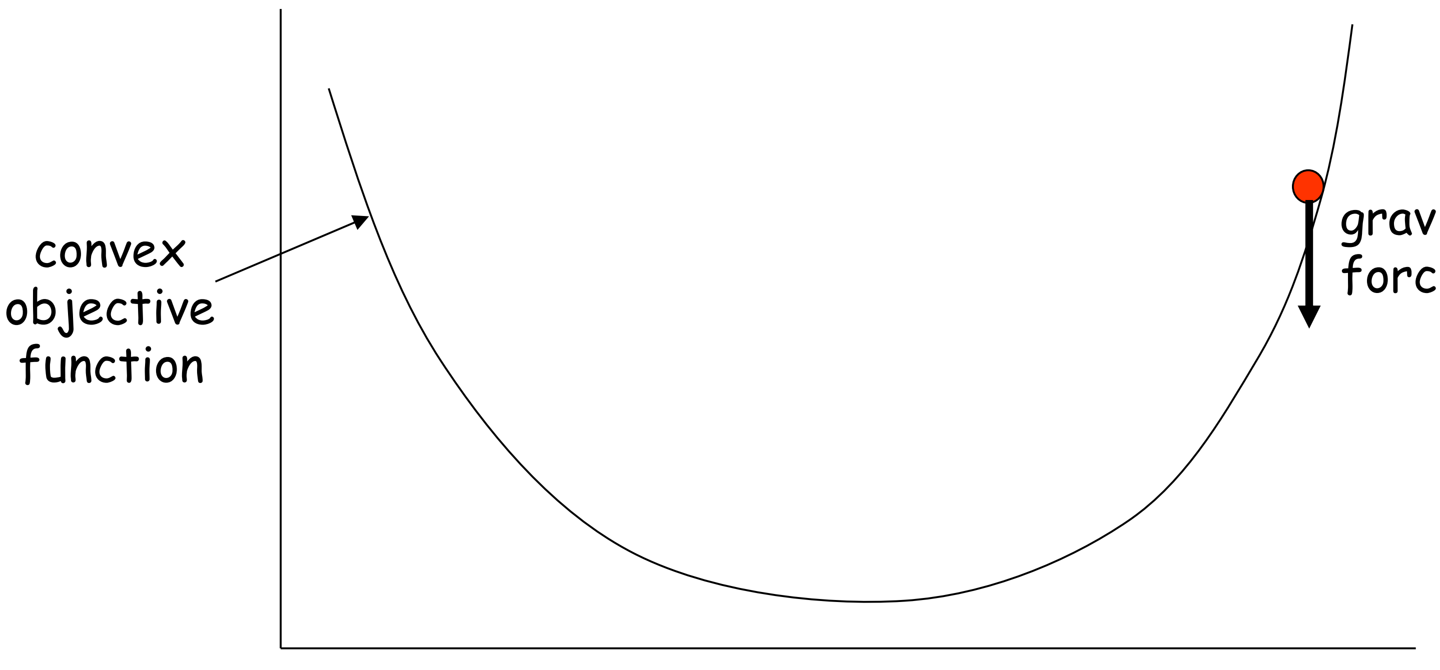
- Convex optimization to the rescue

  "…in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity" - R. Tyrrell Rockafellar, in SIAM Review, 1993

- Two conditions for an optimization problem to be convex:
  - convex objective function
  - convex feasible set

Why is convex optimization easy?

- Because we can simply let gravity do all the hard work for us



- More formally, we can let gradient descent do all the hard work for us

How do we get a convex relaxation?

- By dropping some constraints
  (so that the enlarged feasible set is convex)

- By modifying the objective function
  (so that the new function is convex)

- By combining both of the above

Linear programming (LP) relaxations

- Optimize linear function subject to linear constraints, i.e.:

$$\begin{aligned} &\min\ \mathbf{c}^T\mathbf{x} \\ &\text{s.t.}\quad \mathbf{A}\mathbf{x} = \mathrm{b} \end{aligned}$$

- Very common form of a convex relaxation, because:
    - Typically leads to very efficient algorithms (important due to large scale nature of problems in computer vision)
    - Also often leads to combinatorial algorithms
    - Surprisingly good approximation for many problems

MRFs and Linear Programming

- Tight connection between MRF optimization and Linear Programming (LP) recently emerged

- Active research topic with a lot of interesting work:

  – MRFs and LP-relaxations [Schlesinger] [Boros] [Wainwright et al. 05] [Kolmogorov 05] [Weiss et al. 07] [Werner 07] [Globerson et al. 07] [Kohli et al. 08]…

  – Tighter relaxations/alternative relaxations [Sontag et al. 07, 08] [Werner 08] [Kumar et al. 07, 08]

MRFs and Linear Programming

- E.g., state of the art MRF algorithms are now known to be directly related to LP:
  - Graph-cut based techniques such as *α*-expansion:
    generalized by primal-dual schema algorithms (Komodakis et al. 05, 07)
  - Message-passing techniques:
    generalized by TRW methods (Wainwright 03, Kolmogorov 05)
    further generalized by Dual-Decomposition (Komodakis 07)
- The above statement is more or less true for almost all state-of-the-art MRF techniques

What about MRFs and other relaxations?

- Many alternative types of **convex** relaxations (quadratic, SOCP, SDP, etc..)
- But:
  - Efficiency for large scale problems?
  - Moreover, many of them less powerful than LP [Kumar et al. 07]

- Beyond convex relaxations: what about **non-convex** relaxations? May be covered in the last lecture ('recent advances').

Outline

LP-based methods for MAP estimation can be…

Key ingredient: duality theory

the fastest → Primal-dual schema (Komodakis et al. 05, 07)

extremely general → Dual decomposition (Komodakis et al. 07)

very accurate → Cycle-repairing (beyond loose LPs)

NOTE: each **green box** may be linked to <u>many</u> **red ones**

Part II
Primal-dual schema
FastPD algorithm

The primal-dual schema

- Highly successful technique for exact algorithms. Yielded exact algorithms for cornerstone combinatorial problems:

matching	network flow
minimum spanning tree	minimum branching
shortest path	...

- Soon realized that it's also an extremely powerful tool for deriving approximation algorithms [Vazirani]:

set cover	steiner tree
steiner network	feedback vertex set
scheduling	...

The primal-dual schema

- **Conjecture:**

> Any approximation algorithm can be derived using the primal-dual schema

(has not been disproved yet)

The primal-dual schema

- Say we seek an optimal solution x^* to the following integer program (this is our **primal** problem):

$$\begin{aligned} &\min \mathbf{c}^T\mathbf{x} \\ &\text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathbb{N} \end{aligned}$$

(NP-hard problem)

- To find an approximate solution, we first relax the integrality constraints to get a primal & a dual linear program:

primal LP: $\min \mathbf{c}^T\mathbf{x}$

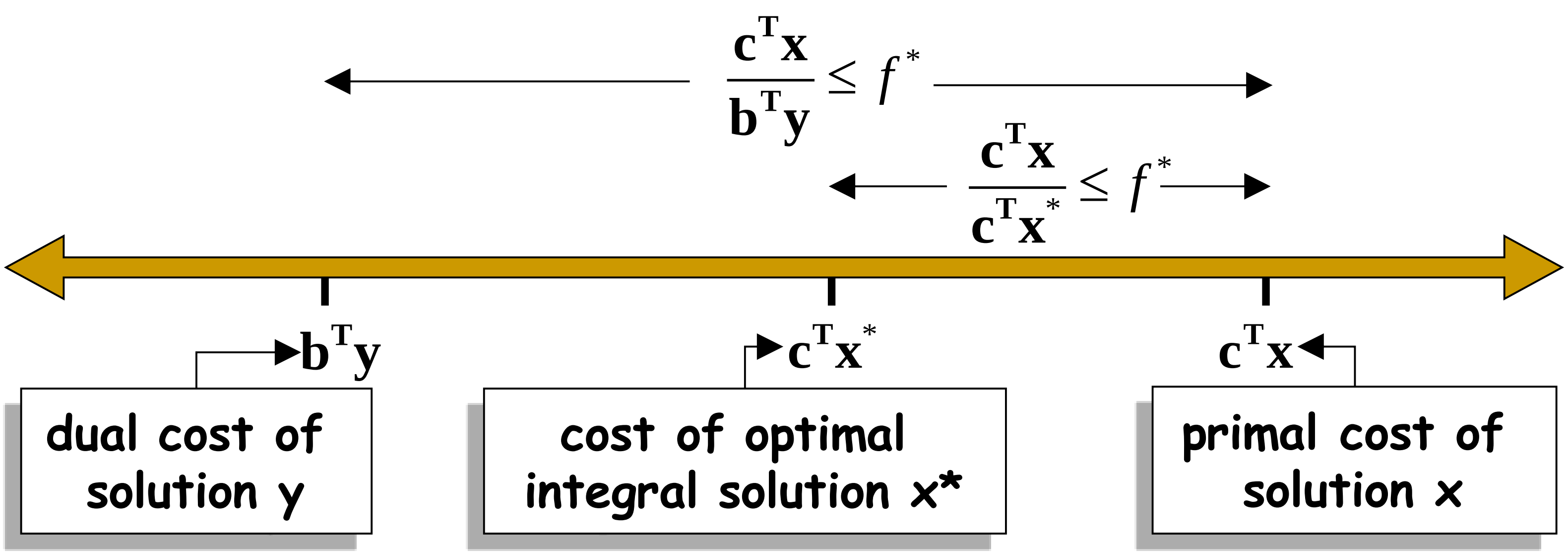
$$\text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$$

dual LP: $\max \mathbf{b}^T\mathbf{y}$

$$\text{s.t.} \quad \mathbf{A}^T\mathbf{y} \leq \mathbf{c}$$

The primal-dual schema

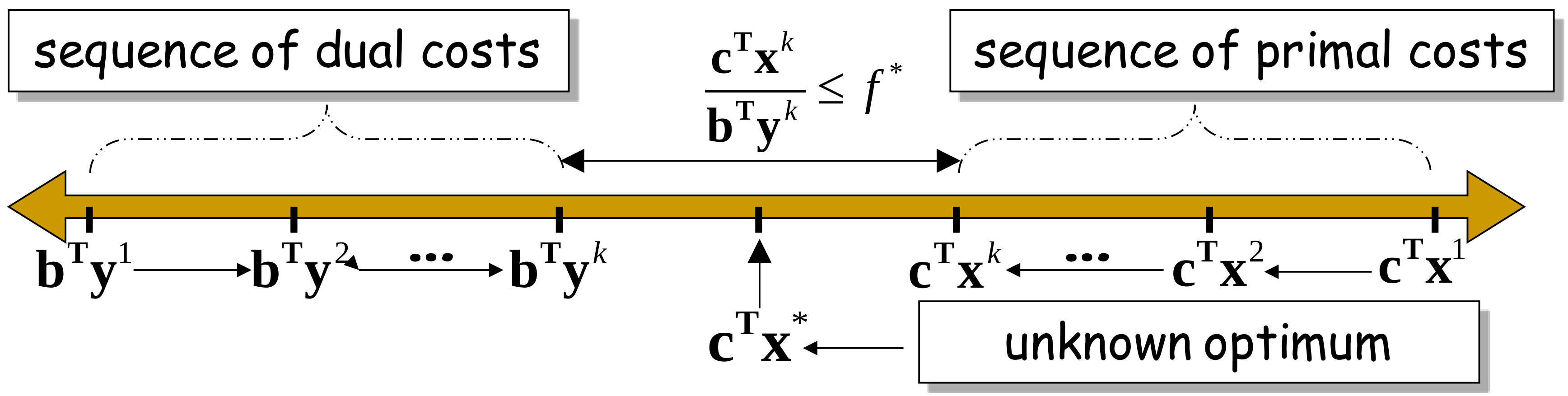
- Goal: find integral-primal solution x, feasible dual solution y such that their primal-dual costs are "close enough", e.g.,



Then x is an f^*-approximation to optimal solution x^*

The primal-dual schema

- The primal-dual schema works iteratively



- Global effects, through local improvements!

- Instead of working directly with costs (usually not easy), use <u>RELAXED complementary slackness conditions</u> (easier)

- Different relaxations of complementary slackness
  Different approximation algorithms!!!

Complementary slackness

primal LP: $\min \mathbf{c}^T\mathbf{x}$

$\text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$

dual LP: $\max \mathbf{b}^T\mathbf{y}$

$\text{s.t.} \quad \mathbf{A}^T\mathbf{y} \leq \mathbf{c}$

Complementary slackness conditions:

$$\forall 1 \leq j \leq n : \quad x_j > 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i = c_j$$

Theorem. If **x** and **y** are primal and dual feasible and satisfy the complementary slackness condition then they are both optimal.

Relaxed complementary slackness

primal LP: $\min \mathbf{c}^T \mathbf{x}$

s.t. $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$

dual LP: $\max \mathbf{b}^T \mathbf{y}$

s.t. $\mathbf{A}^T \mathbf{y} \leq \mathbf{c}$

Exact CS: $\forall 1 \leq j \leq n: \quad x_j > 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i = c_j$

Relaxed CS: $\forall 1 \leq j \leq n: \quad x_j > 0 \Rightarrow \sum_{i=1}^{m} a_{ij} y_i \geq c_j / f_j$

$f_j = 1 \forall j$ implies 'exact' complemetary slackness (why?)

Theorem. If **x, y** primal/dual feasible and satisfy the **relaxed CS** condition then **x** is an f-approximation of the optimal integral solution, where f = max_j f_j.

Complementary slackness and the primal-dual schema

Theorem (previous slide). If **x**, **y** primal/dual feasible and satisfy the relaxed CS condition then **x** is an f-approximation of the optimal integral solution, where f = max_j f_j.

Goal of the primal dual schema: find a pair (**x**,**y**) that satisfies:

- Primal feasibility
- Dual feasibility
- (Relaxed) complementary slackness conditions.

Relaxed complementary slackness for MRFs: example

(Reminder) LP relaxation for MRFs:

$$\min \left[\sum_{p \in G} \sum_{a \in L} V_p(a) x_{p,a} + \sum_{pq \in E} \sum_{a,b \in L} V_{pq}(a,b) x_{pq,ab} \right]$$

$$\text{s.t.} \sum\nolimits_{a \in L} x_{p,a} = 1$$

(only one label assigned per vertex)

$$\left.\begin{aligned} \sum\nolimits_{a \in L} x_{pq,ab} &= x_{q,b} \\ \sum\nolimits_{b \in L} x_{pq,ab} &= x_{p,a} \end{aligned}\right\}$$

enforce consistency between variables $x_{p,a}$, $x_{q,b}$ and variable $x_{pq,ab}$

$$x_{p,a} \geq 0, \; x_{pq,ab} \geq 0$$

Binary variables

$x_{p,a}=1 \Leftrightarrow$ label a is assigned to node p

$x_{pq,ab}=1 \Leftrightarrow$ labels a, b are assigned to nodes p, q

Relaxed complementary slackness for MRFs: example

Special case: $V_{pq}(a,b) = w_{pq} d(a,b) \quad \forall (p,q) \in E, \forall a,b \in L$

Primal (integer) LP

$$\min \sum_{p \in V, a \in L} c_{p,a} x_{p,a} + \sum_{(p,q) \in E} w_{pq} \sum_{a,b \in L} d_{ab} x_{pq,ab}$$

$$\begin{aligned}
\text{s.t. } & \textstyle\sum_a x_{p,a} = 1 && \forall\, p \in V \\
& \textstyle\sum_a x_{pq,ab} = x_{q,b} && \forall\, b \in L,\ (p,q) \in E \\
& \textstyle\sum_b x_{pq,ab} = x_{p,a} && \forall\, a \in L,\ (p,q) \in E \\
& x_{p,a},\ x_{pq,ab} \in \{0,1\} && \forall\, p \in V,\ (p,q) \in E,\ \ a,b \in L
\end{aligned}$$

Dual LP

$$\max \textstyle\sum_p y_p$$

$$\begin{aligned}
\text{s.t. } & y_p \le c_{p,a} + \sum_{q: q \sim p} y_{pq,a} && \forall p \in V,\ a \in L \\
& y_{pq,a} + y_{qp,b} \le w_{pq} d_{ab} && \forall a,b \in L, (p,q) \in E
\end{aligned}$$

Complementary slackness conditions

$$x_{p,a} > 0 \Rightarrow y_p \ge c_{p,a}/f_1 + \textstyle\sum_{q:q \sim p} y_{pq,a}$$

$$x_{pq,ab} > 0 \Rightarrow y_{pq,a} + y_{qp,b} \ge w_{pq} d_{ab} / f_2$$

Homework: check this slide yourself

FastPD: primal-dual schema for MRFs

- Regarding the PD schema for MRFs, it turns out that:

each update of primal and dual variables	⇄	solving max-flow in appropriately constructed graph

- Resulting flows tell us how to update both:
  - the dual variables, as well as
  - the primal variables

for each iteration of primal-dual schema

- Max-flow graph defined from current primal-dual pair (x^k, y^k)
  - (x^k, y^k) defines **connectivity** of max-flow graph
  - (x^k, y^k) defines **capacities** of max-flow graph

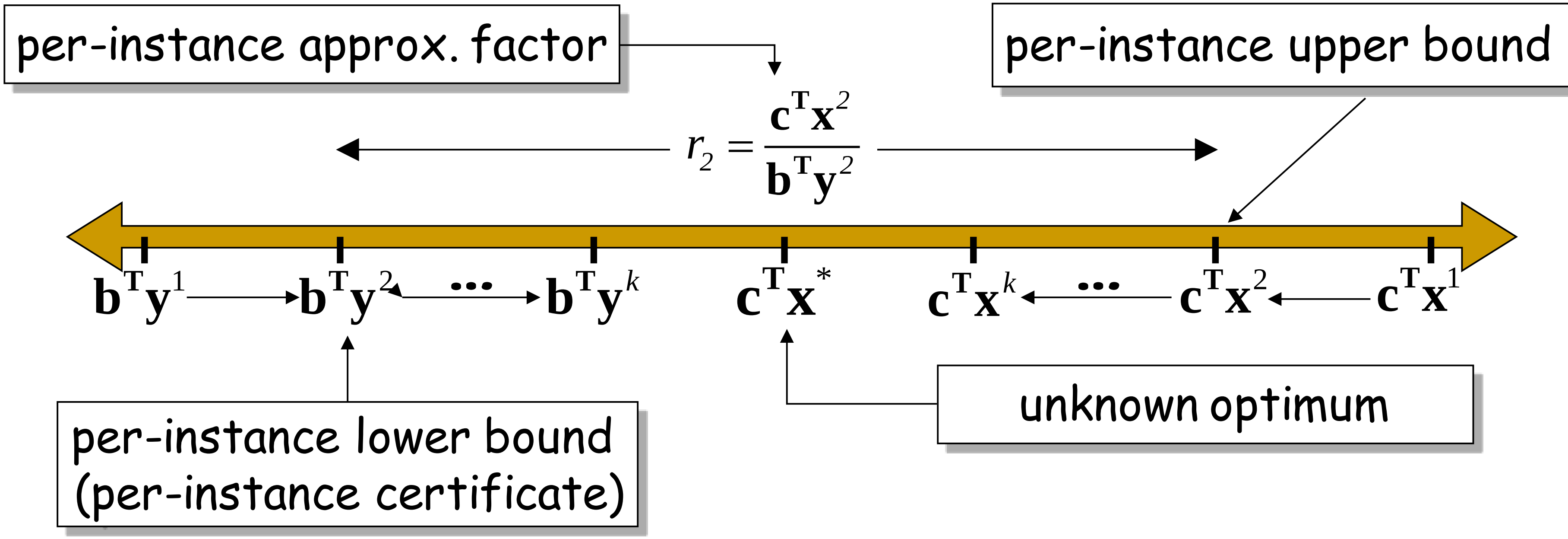
- Max-flow graph is thus continuously updated

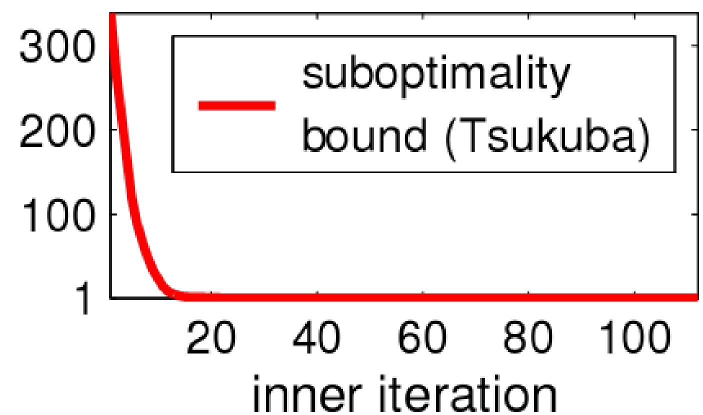
FastPD: primal-dual schema for MRFs

- Very general framework. Different PD-algorithms by RELAXING complementary slackness conditions differently.
- E.g., simply by using a particular relaxation of complementary slackness conditions (and assuming $V_{pq}(\cdot,\cdot)$ is a metric) THEN **resulting algorithm shown equivalent to a-expansion! [Boykov,Veksler,Zabih]**
- PD-algorithms for non-metric potentials $V_{pq}(\cdot,\cdot)$ as well
- **Theorem:** All derived PD-algorithms shown to satisfy certain relaxed complementary slackness conditions
- Worst-case optimality properties are thus **guaranteed**

Per-instance optimality guarantees

- Primal-dual algorithms can always tell you **(for free)** how well they performed for a particular instance

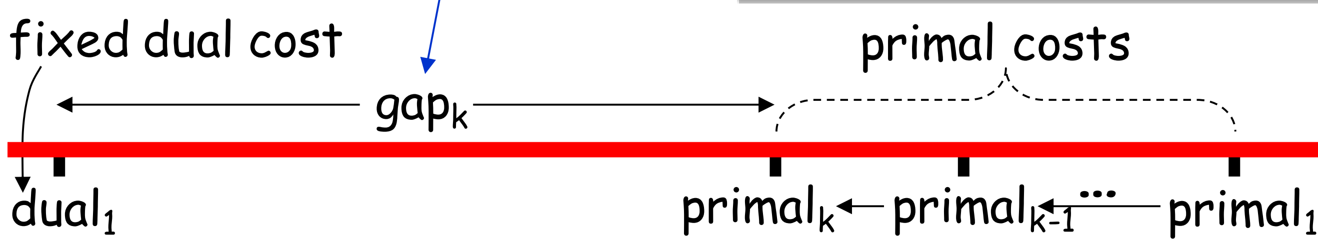




Computational efficiency (static MRFs)

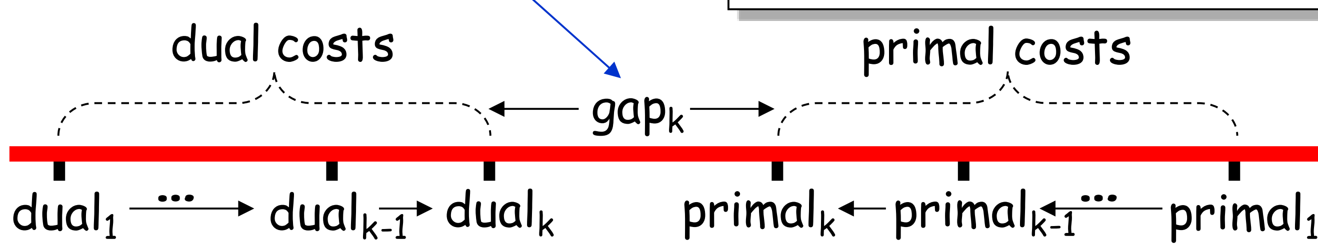
- MRF algorithm <u>only</u> in the primal domain (e.g., a-expansion)

STILL BIG → **Many augmenting paths per max-flow**



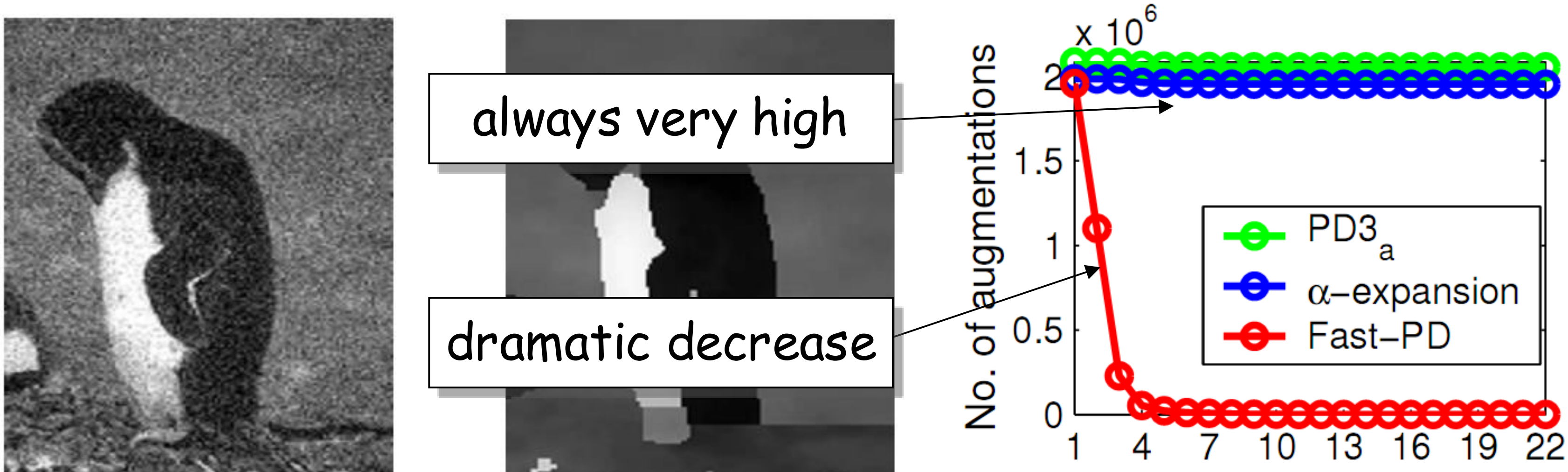
- MRF algorithm in the primal-dual domain (Fast-PD)

SMALL → **Few augmenting paths per max-flow**



Theorem: primal-dual gap = upper-bound on #augmenting paths (i.e., primal-dual gap indicative of time per max-flow)

Computational efficiency (static MRFs)



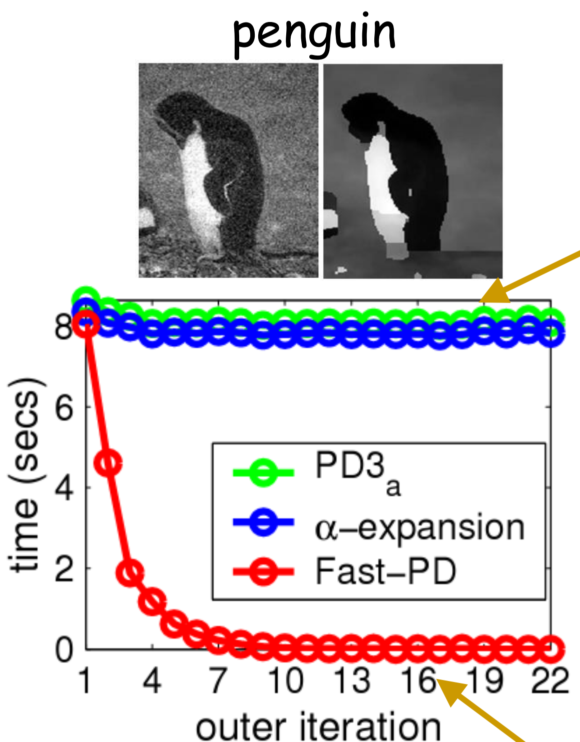
noisy image   denoised image

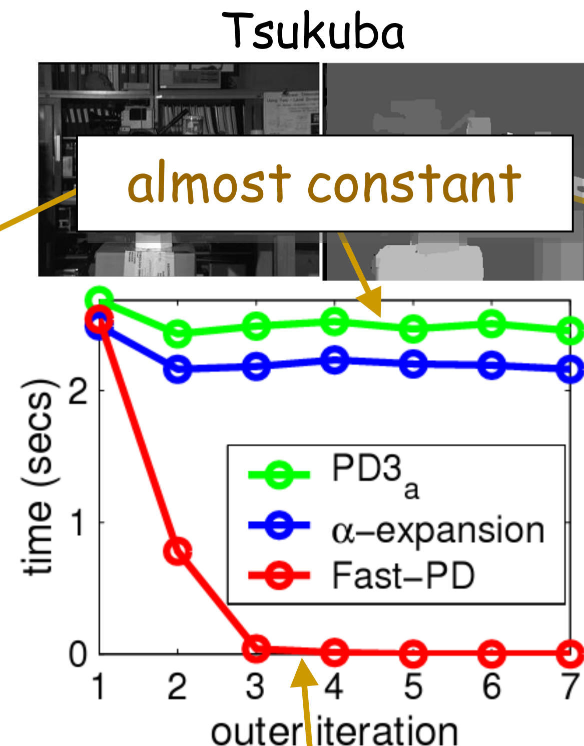
- Incremental construction of max-flow graphs (recall that max-flow graph changes per iteration)

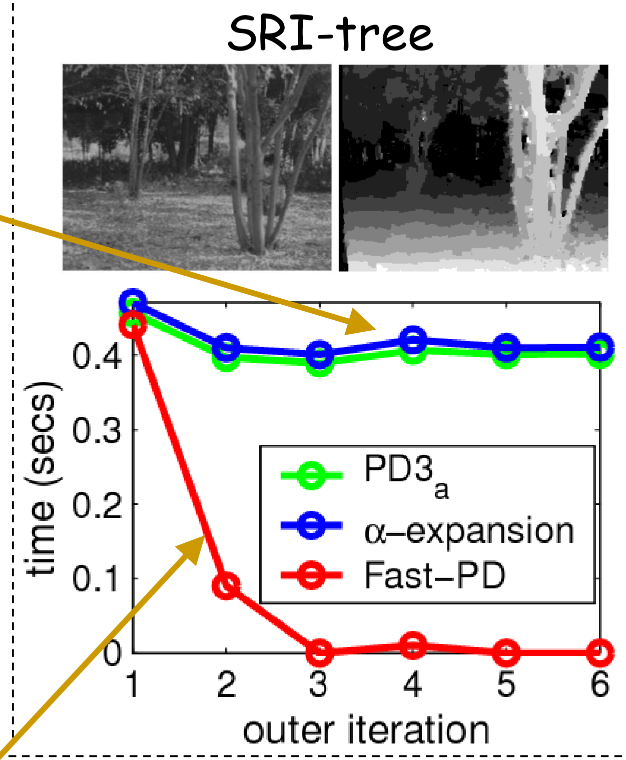
- Possible because we keep <u>both</u> **primal** and **dual** information

- Principled way for doing this construction via the primal-dual framework

Computational efficiency (static MRFs)







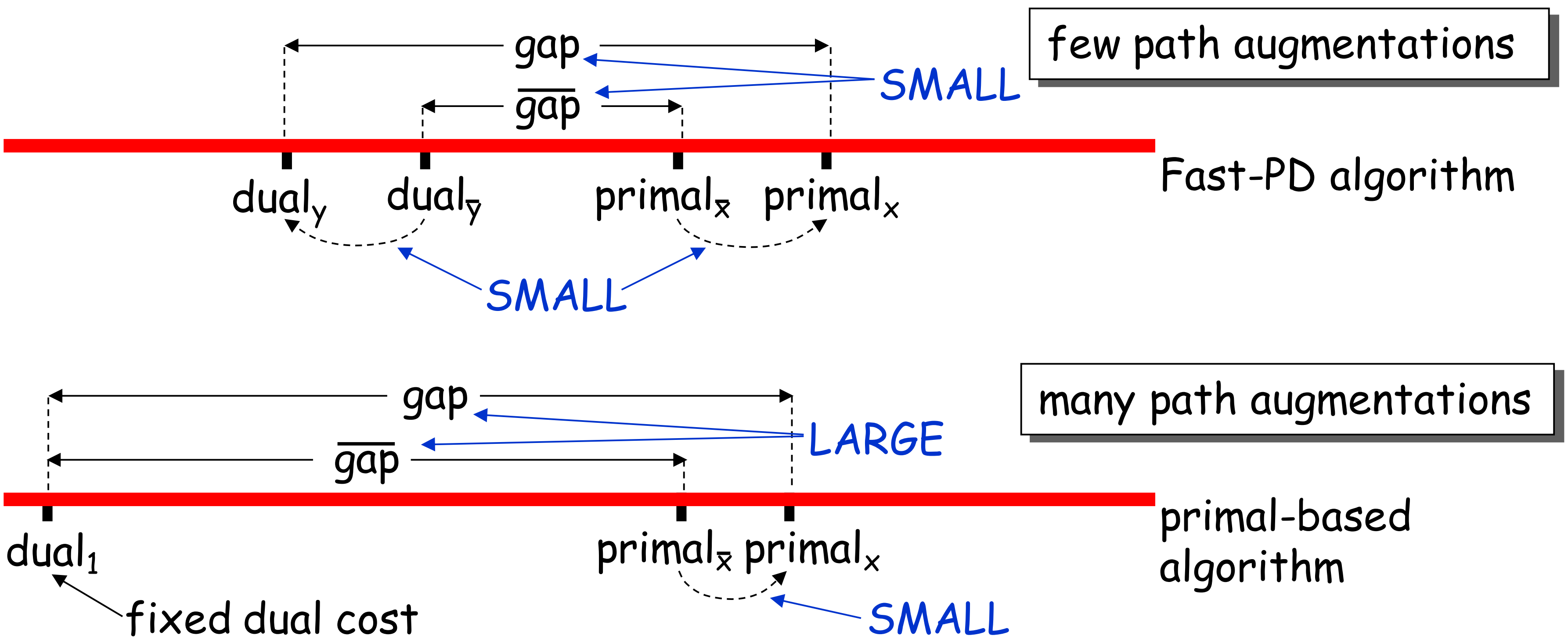
almost constant

dramatic decrease

total (secs)	Fast-PD	α-expansion
penguin	17.44	173.1
tsukuba	3.37	15.63
SRI tree	0.54	2.56

Computational efficiency (dynamic MRFs)

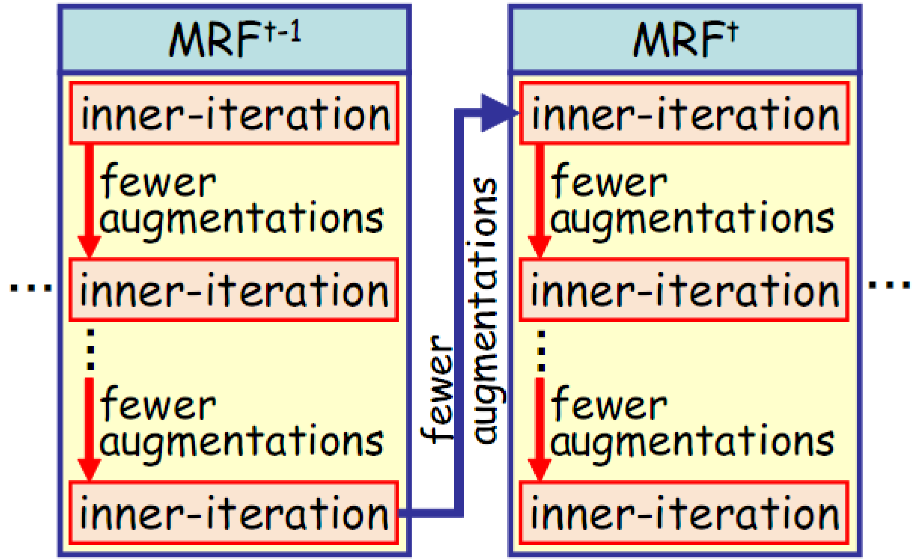
- Fast-PD can speed up dynamic MRFs [Kohli, Torr] as well (demonstrates the power and generality of this framework)



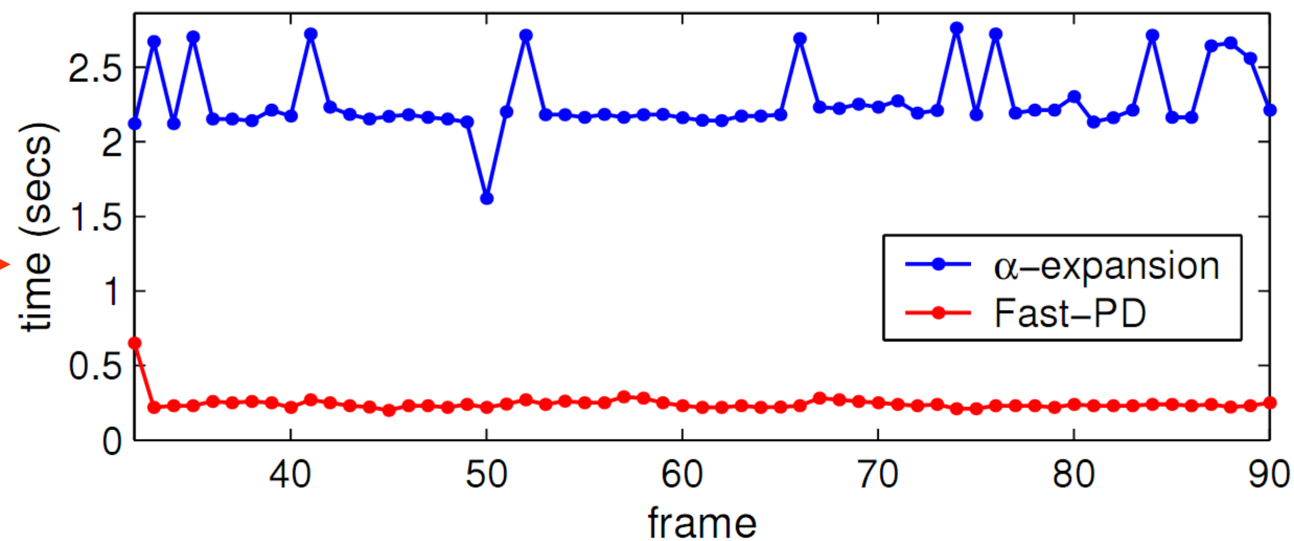
- Principled (and simple) way to update dual variables when switching between different MRFs

Computational efficiency (dynamic MRFs)

- Essentially, Fast-PD works along 2 different "axes"
  - reduces augmentations across <u>different iterations</u> of the same MRF
  - reduces augmentations across <u>different MRFs</u>



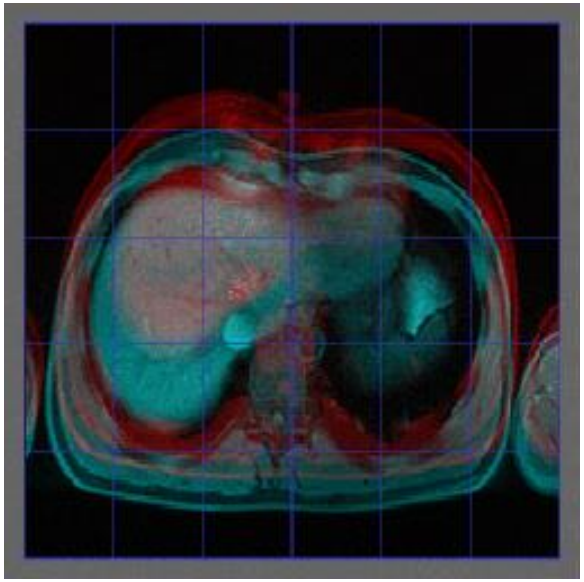
- Time per frame for SRI-tree stereo sequence

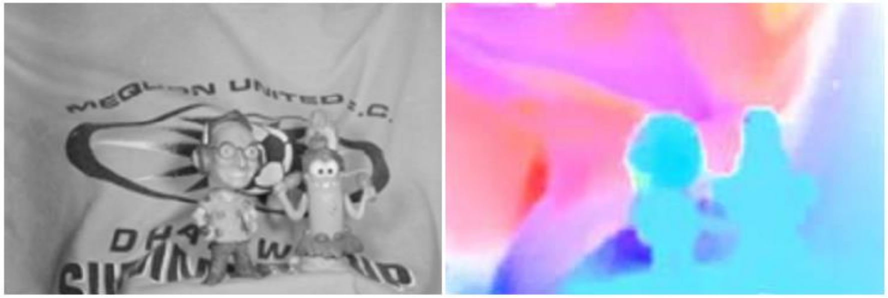


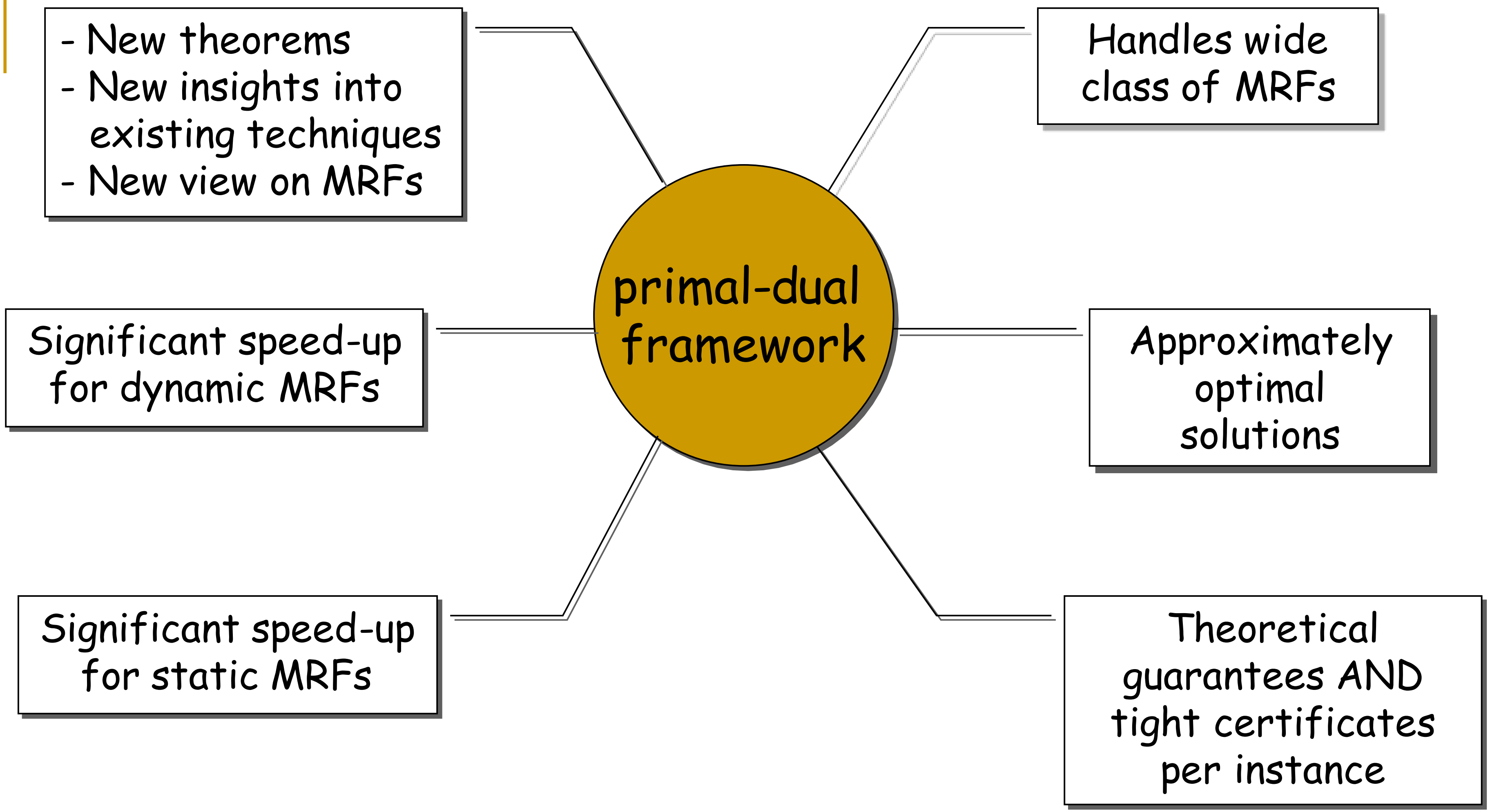
- Handles general (multi-label) dynamic MRFs

Drop: Deformable Registration using Discrete Optimization [Glocker et al. 07, 08]

- Easy to use GUI
- Main focus on medical imaging
- 2D-2D registration
- 3D-3D registration
- Publicly available:
  http://campar.in.tum.de/Main/Drop






primal-dual framework
- New theorems
- New insights into existing techniques
- New view on MRFs
Handles wide class of MRFs
Significant speed-up for dynamic MRFs
Approximately optimal solutions
Significant speed-up for static MRFs
Theoretical guarantees AND tight certificates per instance

Part III
Dual Decomposition

MRF optimization via dual-decomposition

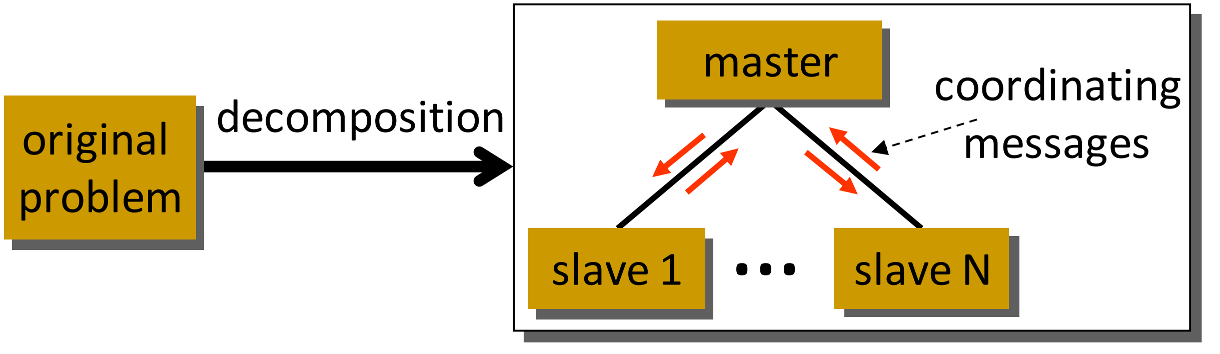
- Very general framework to address discrete MRF-based optimization [Komodakis et al. 07]

- Reduces MRF optimization to a simple **projected subgradient** method
  - Combines solutions from sub-problems in a principled and optimal manner
  - theoretical setting rests on the very powerful technique of **Dual Decomposition**
  - Applies to a wide variety of cases

Dual-decomposition

- Decomposition into subproblems (**slaves**)

- Coordination of slaves by a **master** process



- Proceeds in the dual domain

Dual-decomposition

Let's get the idea from a toy example

(Derivation on blackboard, no slides yet, sorry.

Don't forget to take notes.)

So, who are the slaves?

- Slaves can be MRFs corresponding to subgraphs of the original MRF graph

- To each graph T from a set of subgraphs $\mathcal{T}$, we can associate a slave MRF with parameters (i.e., potentials) θ^T
  - subgraphs must cover the original graph
  - sum of potentials of slave MRFs must reproduce original potentials:

    $$\sum_{T \in \mathcal{T}(p)} \theta_p^T = \theta_p, \qquad \sum_{T \in \mathcal{T}(pq)} \theta_{pq}^T = \theta_{pq},$$

    (Here $\mathcal{T}(p), \mathcal{T}(pq)$ denote all trees in $\mathcal{T}$ containing respectively p and pq)

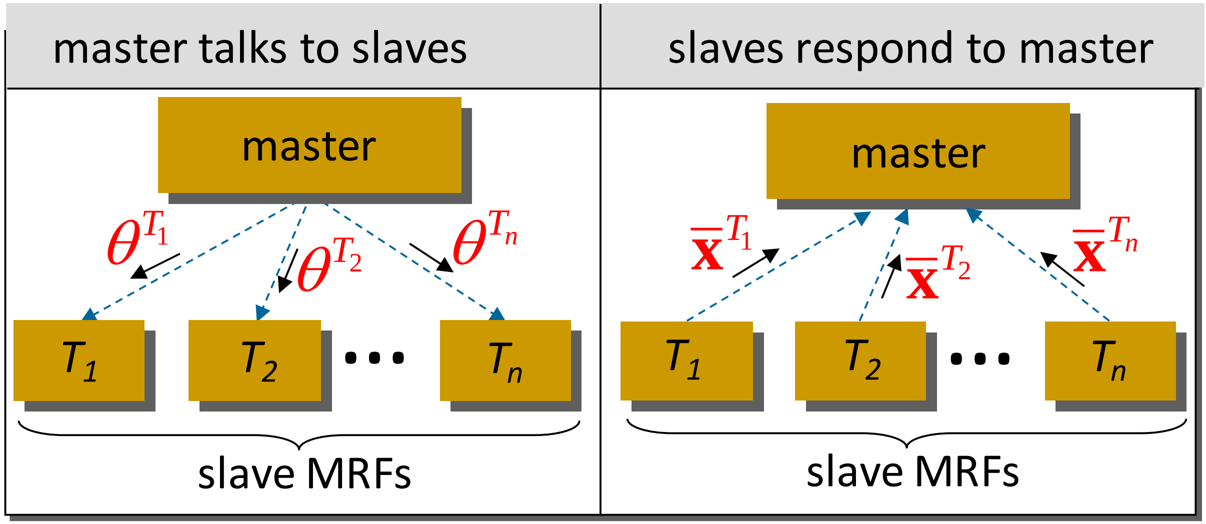
- Note that if, e.g., slave MRFs are tree-structured, then these are easy problems (solvable via max-product)

And who is the master?

- Master can be shown to globally optimize a relaxation
  - Dual relaxation

- E.g., if all slave MRFs are tree-structured then:
  master optimizes LP relaxation described earlier

“What is it that you seek, Master?...”



- Based on all collected minimizers, master readjusts the parameters of each slave MRF:

$$\theta_p^T \mathrel{+}= \alpha_t \cdot \left(\bar{\mathbf{x}}_p^T - \frac{\sum_{T' \in \mathcal{T}(p)} \bar{\mathbf{x}}_p^{T'}}{|\mathcal{T}(p)|} \right), \quad \theta_{pq}^T \mathrel{+}= \alpha_t \cdot \left(\bar{\mathbf{x}}_{pq}^T - \frac{\sum_{T' \in \mathcal{T}(pq)} \bar{\mathbf{x}}_{pq}^{T'}}{|\mathcal{T}(pq)|} \right)$$

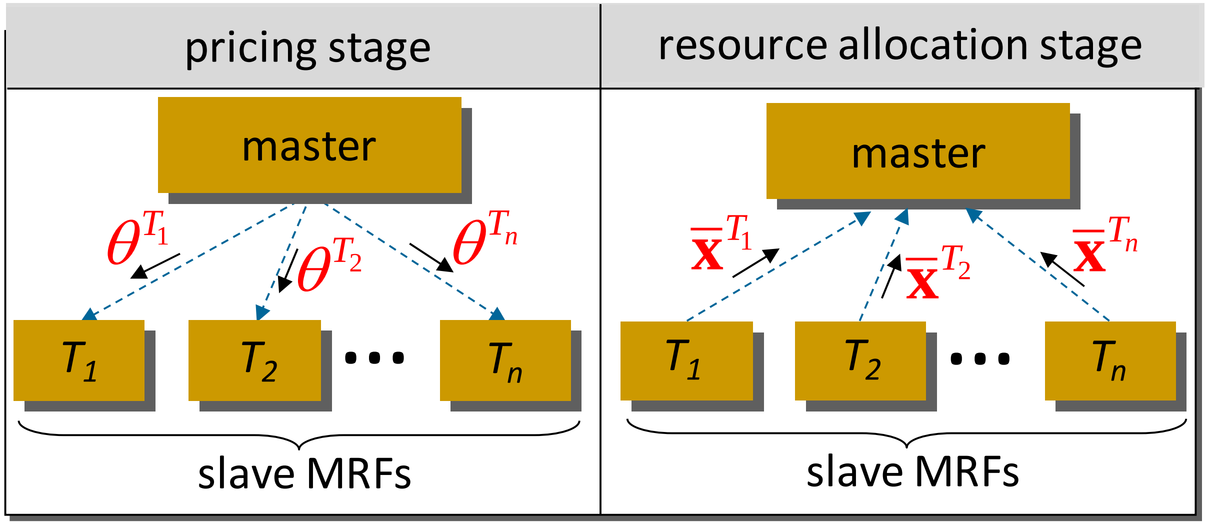
“What is it that you seek, Master?...”

- Master updates the parameters of the slave-MRFs by “averaging” the solutions returned by the slaves.

- Essentially, he tries to achieve consensus among all slave-MRFs
  - This means that minimizers should agree with each other, i.e., assign same labels to common nodes

- For instance, if a certain node is already assigned the same label by all minimizers, the master does not touch the MRF potentials of that node.

“What is it that you seek, Master?...”



Economic interpretation:

- Think of $\{\bar{\mathbf{x}}^T\}$ as amount of resources consumed by slave-MRFs
- Think of $\{\theta^T\}$ as corresponding prices
- Master naturally adjusts prices as follows:
  - prices for **overutilized** resources are **increased**
  - prices for **underutilized** resources are **decreased**
- Repeated until the market clears

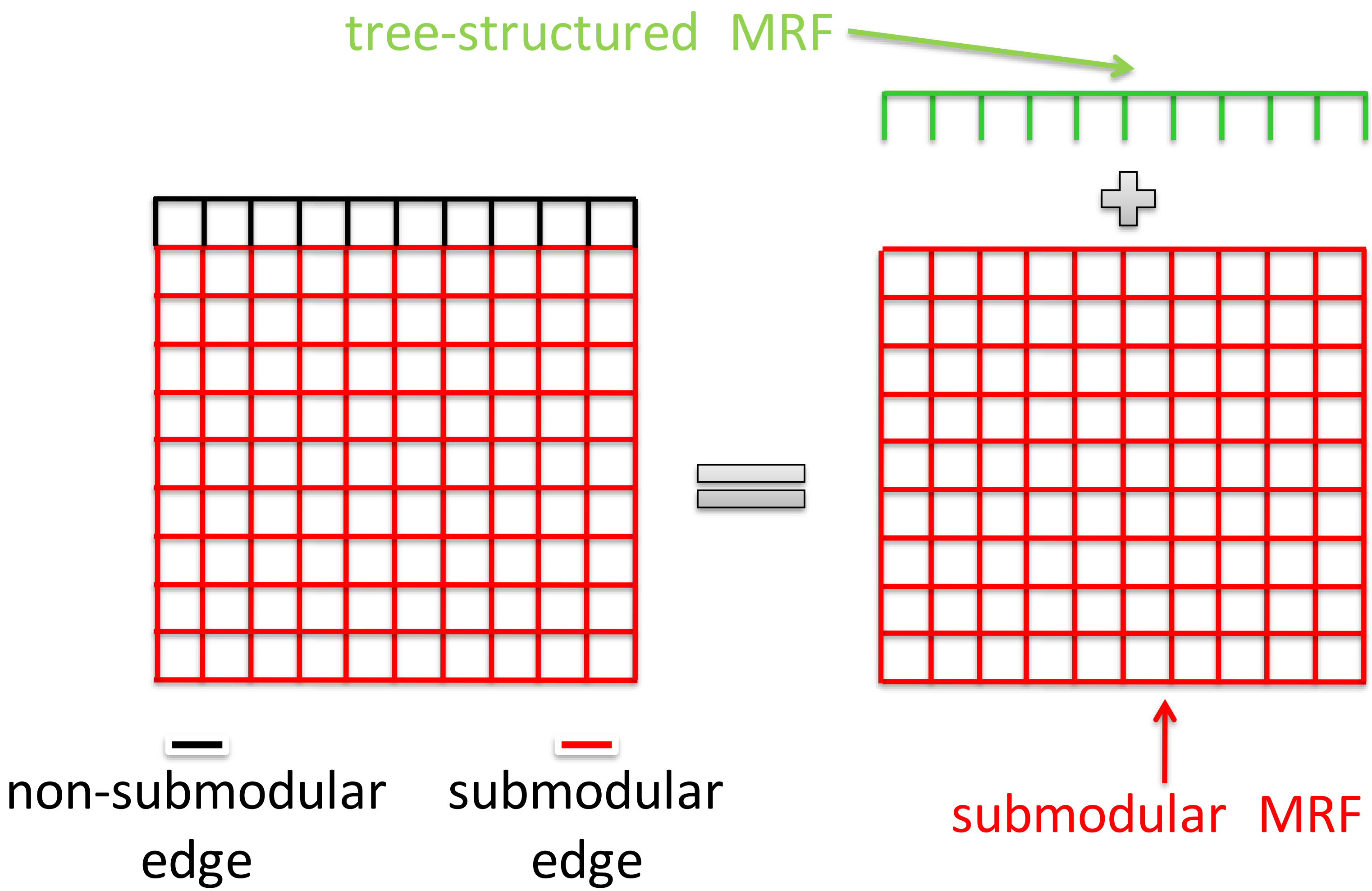
MRF optimization via dual-decomposition

- Extremely general and flexible framework
  - Generalizes and extends many state of the art techniques

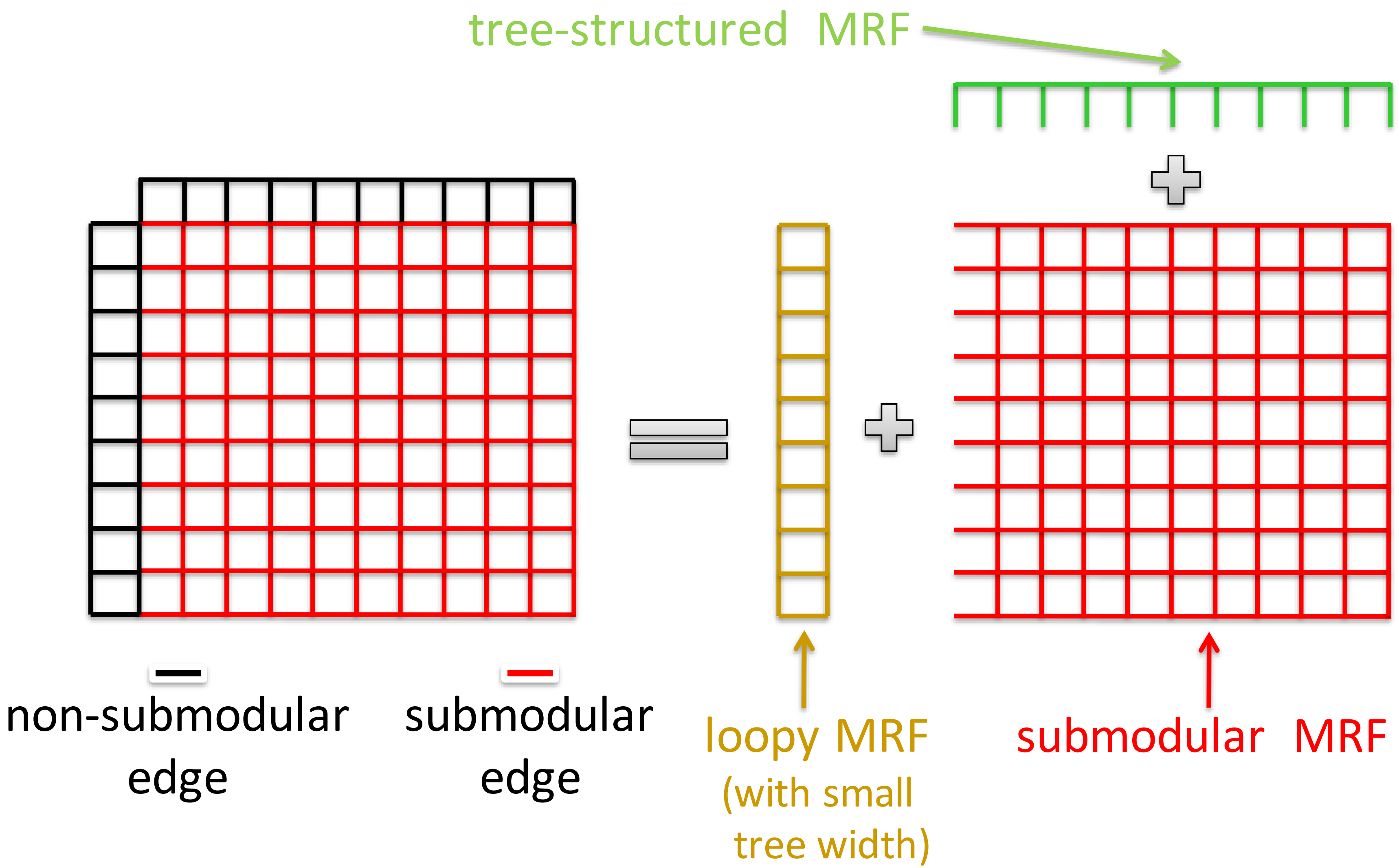
- E.g., if slaves MRFs are chosen to be trees then generalizes TRW methods [Wainwright et al. 03, Kolmogorov 05]
  - Resulting algorithms have stronger theoretical properties

- But can be applied to many other cases
  - Sub-problems need to satisfy very mild conditions
  - Exactly same framework still applies
  - E.g., algorithms for tighter relaxations

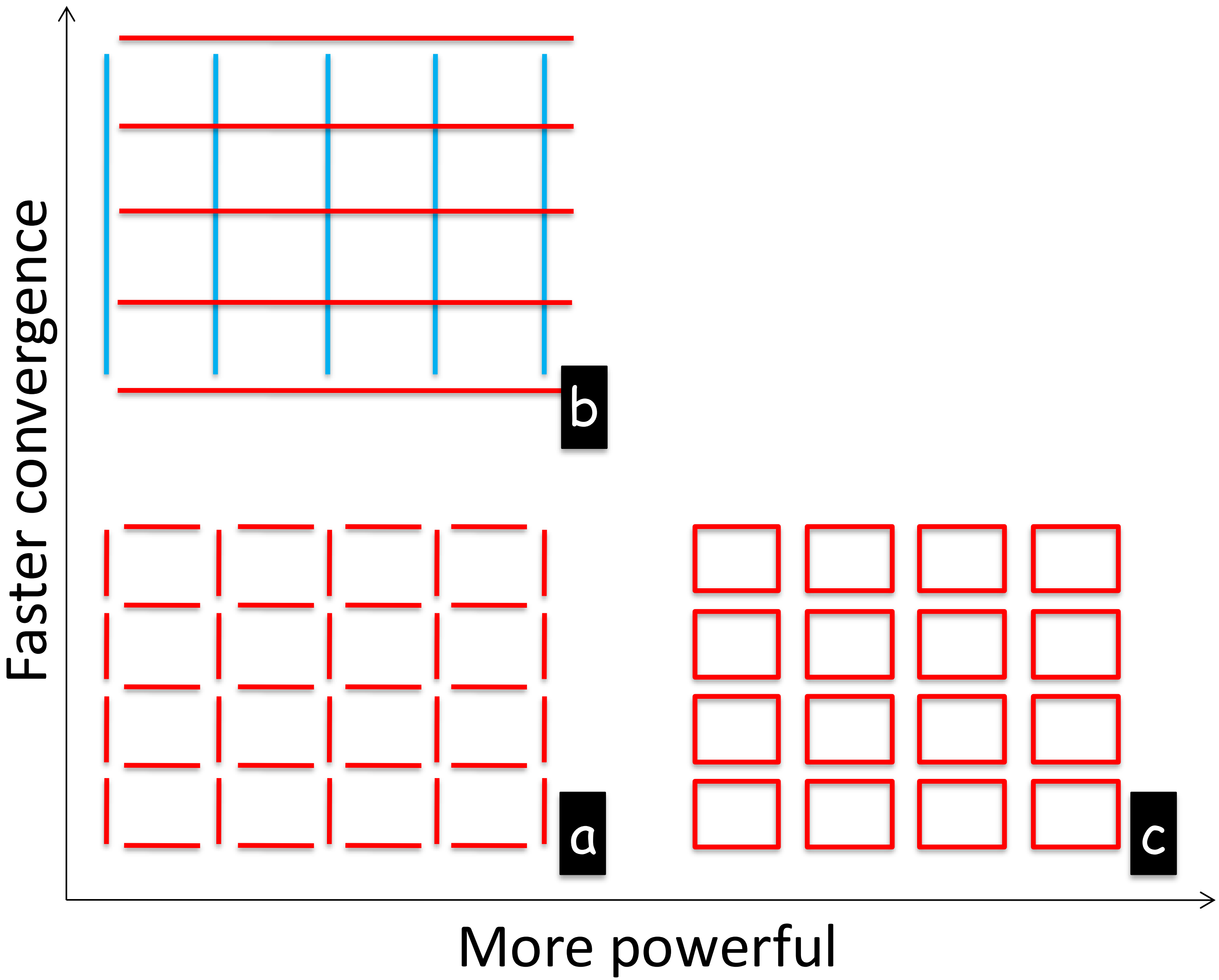
MRF optimization via dual-decomposition



MRF optimization via dual-decomposition

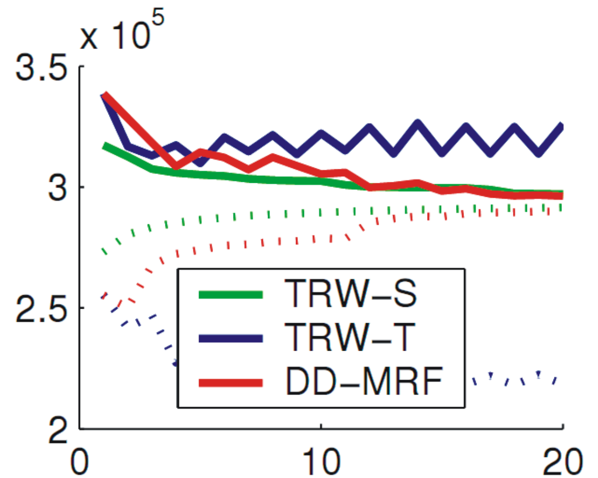


MRF optimization via dual-decomposition



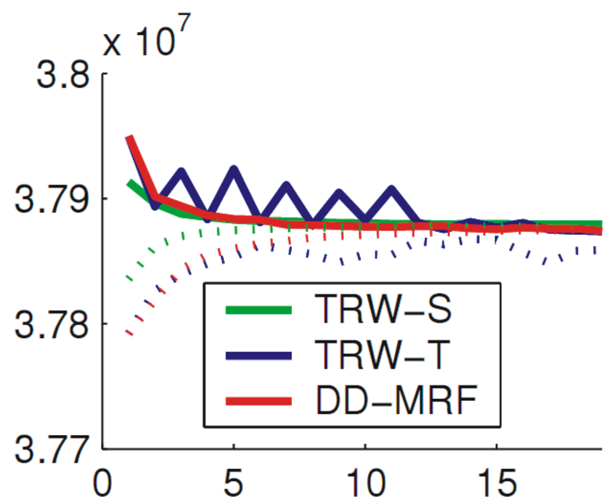
Results

estimated disparity for
Tsukuba stereo pair



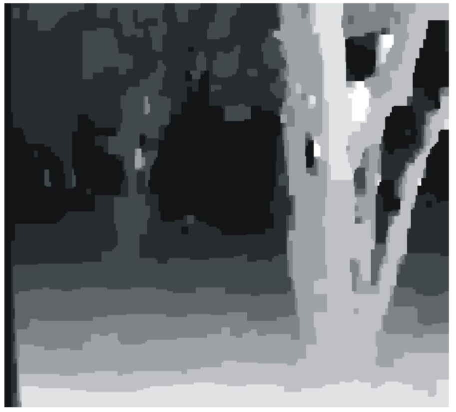
lower bounds (dual costs) and
MRF energies (primal costs)

estimated disparity for
Map stereo pair

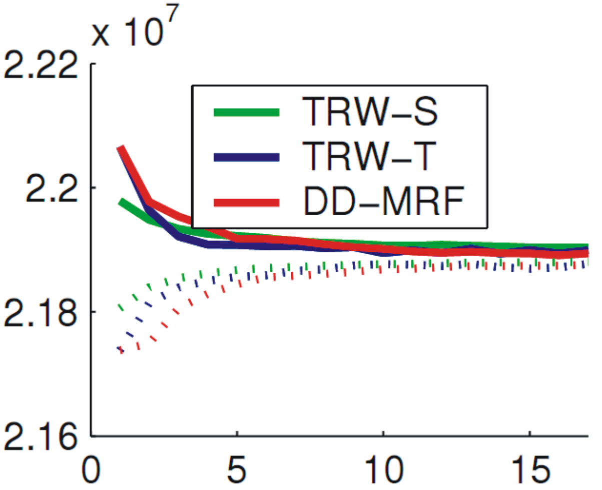


lower bounds (dual costs) and
MRF energies (primal costs)

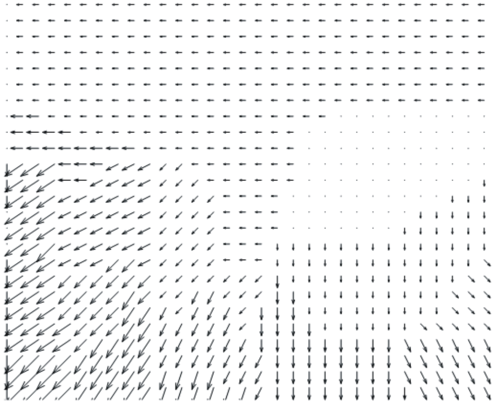
Results



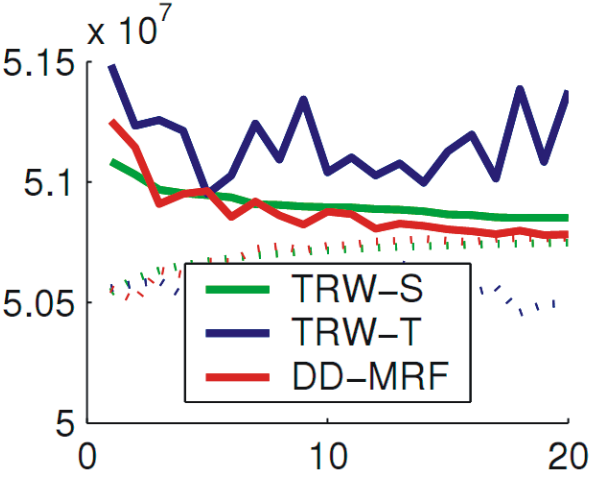
estimated disparity for SRI stereo pair



lower bounds (dual costs) and MRF energies (primal costs)

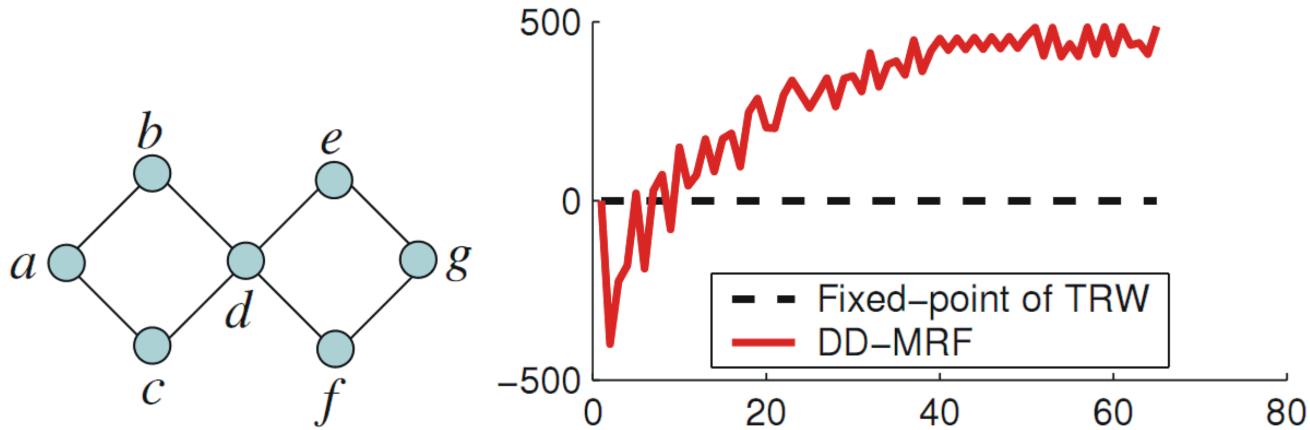


estimated optical flow for Yosemite sequence

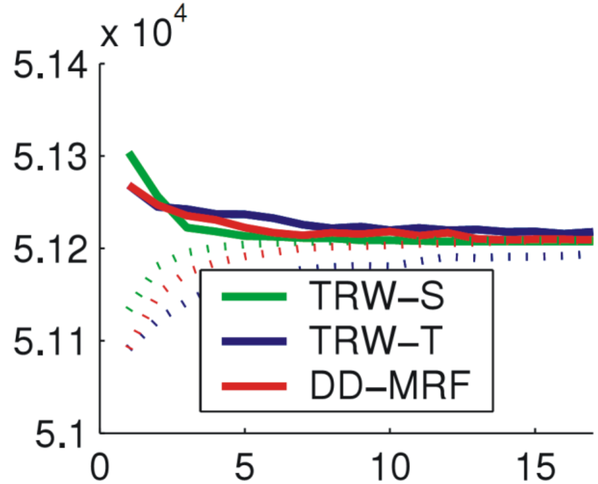


lower bounds (dual costs) and MRF energies (primal costs)

Results



a simple synthetic example illustrating that TRW methods are not able to maximize the dual lower bound, whereas DD-MRF can.



lower bounds (dual costs) and MRF energies (primal costs) for binary segmenation