Reduce, Reuse & Recycle
Efficiently Solving Multi-Label MRFs

Karteek Alahari   Pushmeet Kohli*   Philip H. S. Torr

Oxford Brookes University

*Now at Microsoft Research, Cambridge

Roadmap

Oxford Brookes Vision Group

- Energy Functions on MRFs
- Solving MRFs
  - Exact Methods
  - Approximate Methods
- Efficient Energy Minimization
  - Recycling
  - Reducing
  - Reusing
- Results

Roadmap

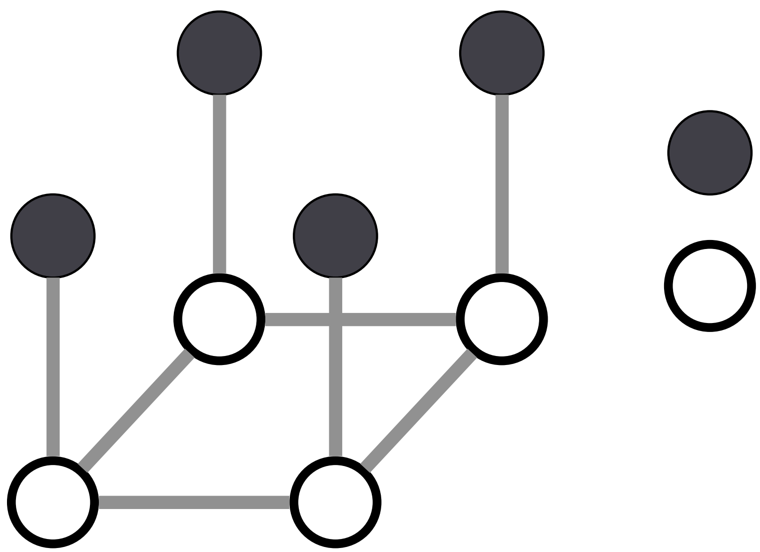
Oxford Brookes Vision Group

- Energy Functions on MRFs
- Solving MRFs
  - Exact Methods
  - Approximate Methods
- Efficient Energy Minimization
  - Recycling
  - Reducing
  - Reusing
- Results

Energy Functions on MRFs

Oxford Brookes Vision Group

MAP Inference



Observed Variables

Hidden Variables

Courtesy: Pushmeet Kohli

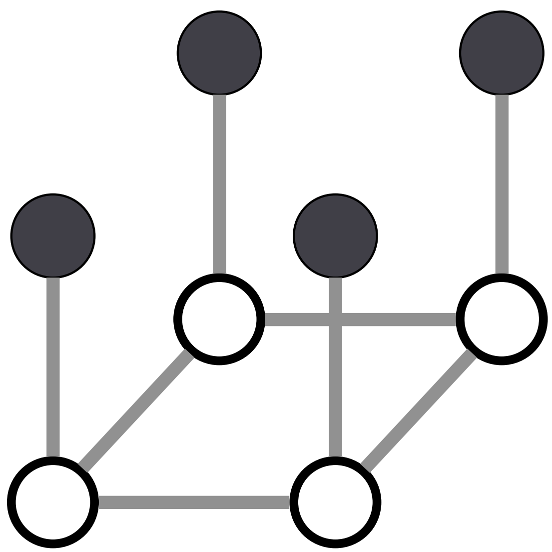
Energy Functions on MRFs

Oxford Brookes Vision Group

MAP Inference

↕

Energy Minimization



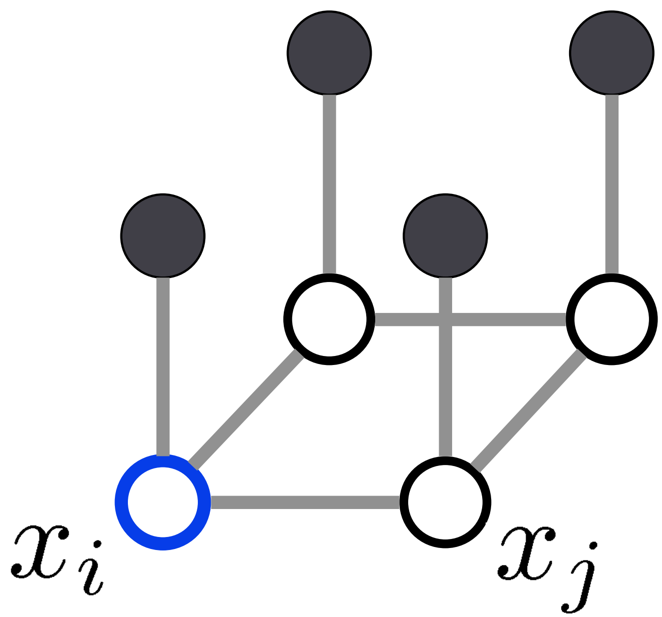
Energy Functions on MRFs

Oxford Brookes Vision Group

- **Pairwise** Energy Functions

$$E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)$$

Unary **Pairwise**



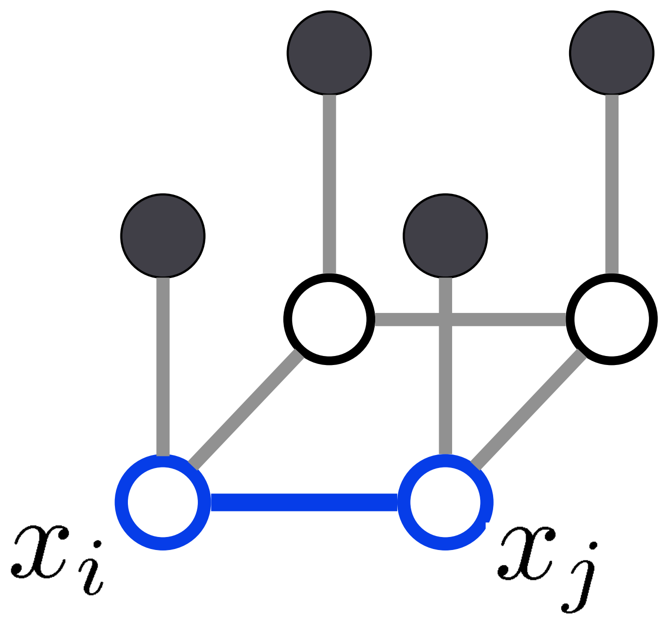
Energy Functions on MRFs

Oxford Brookes Vision Group

- **Pairwise** Energy Functions

$$E(x) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \psi_{ij}(x_i, x_j)$$

Unary **Pairwise**



Roadmap

- Energy Functions on MRFs
- Solving MRFs
  - Exact Methods
  - Approximate Methods
- Efficient Energy Minimization
  - Recycling
  - Reducing
  - Reusing
- Results

Solving MRFs

Oxford Brookes Vision Group

- Minimizing a general MRF energy -- NP-hard
- Exact solutions exist for certain sub-classes
  - Graphs with no loops [Felzenszwalb & Huttenlocher ‘04]
  - Submodular energy functions [Ishikawa ‘03, Kolmogorov & Zabih ‘04, Schlesinger & Flach ‘06]

$$f(a,b) + f(a+1,b+1) \leq f(a,b+1) + f(a+1,b), \quad \forall a,b \in \mathcal{L}$$

- What about the rest?

For instance, Potts model

$$f(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ \gamma & \text{otherwise,} \end{cases}$$

is not submodular

Approximate Methods

Oxford Brookes Vision Group

- Move making algorithms [Boykov et al. '01]
  - Expansion
  - Swap
- Message passing algorithms
  - Belief propagation (BP) [Pearl '98]
  - Tree reweighted message passing (TRW) [Wainwright et al. '05, Kolmogorov '06]
  - Dual decomposition [Komodakis et al. '07]

Roadmap

Oxford Brookes Vision Group

- Energy Functions on MRFs
- Solving MRFs
  - Exact Methods
  - Approximate Methods
- Efficient Energy Minimization
  - Recycling
  - Reducing
  - Reusing
- Results

Efficient Energy Minimization

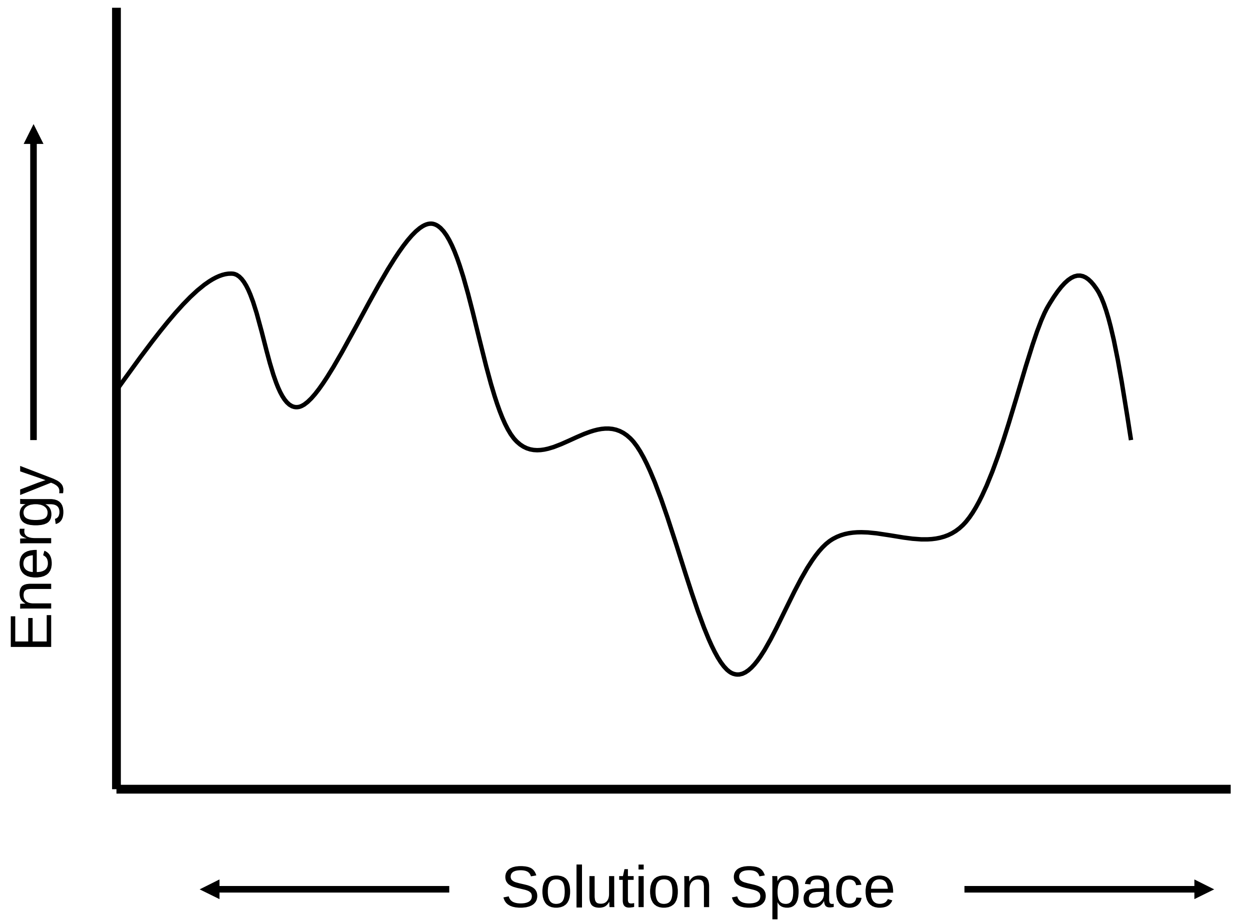
- Take considerable time for large problems
- Running time depends on
  - Initialization used for primal and dual variables
  - The number of variables in the problem
- Efficient methods do exist
  - Limited to submodular dynamic MRFs [Kohli & Torr ‘05, Juan & Boykov ‘06]
  - Fast-PD [Komodakis et al. ‘07]

Efficient Energy Minimization

Oxford Brookes Vision Group

- Our primary goals are
  - To generate a good initialization for the current problem instance **(Recyle & Reuse)**
  - To reduce the number of variables involved in the energy function **(Reduce)**

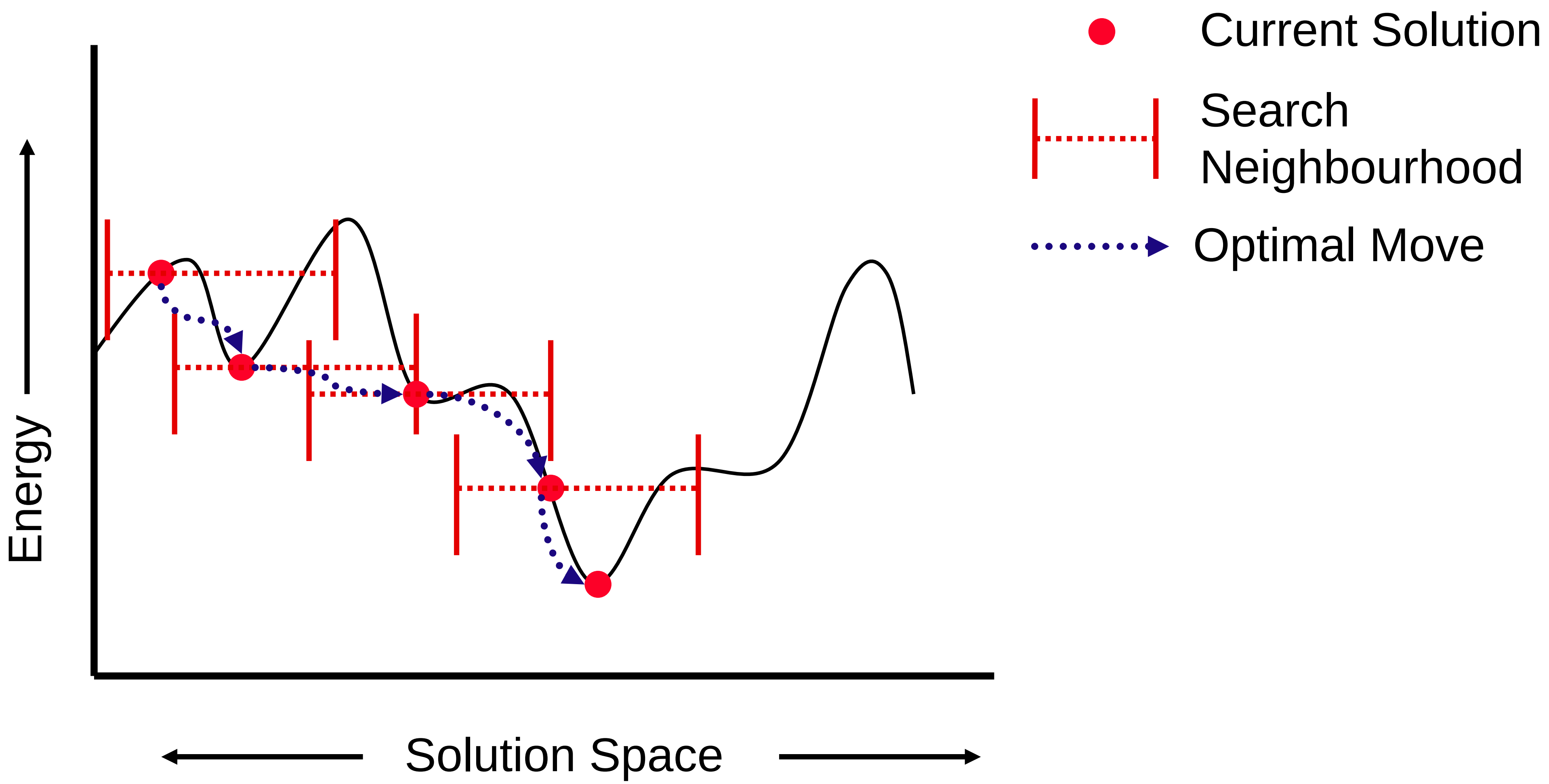
Move Making Algorithms



Courtesy: Pushmeet Kohli

Move Making Algorithms

Oxford Brookes Vision Group



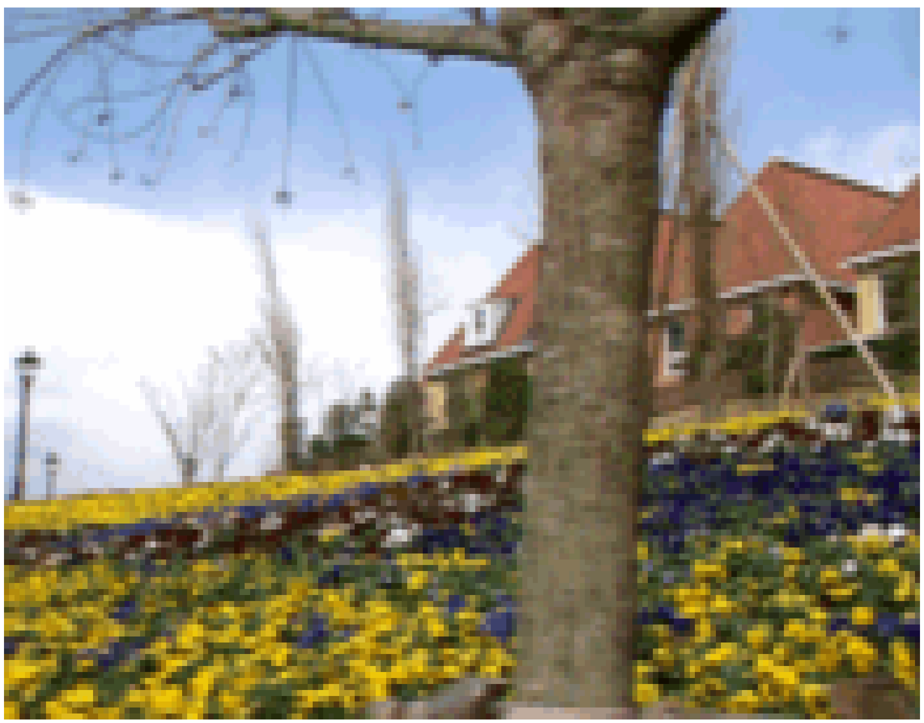
Example: α-expansion

Oxford Brookes Vision Group

Status:

Initialize with Tree
Expand Ground
Expand House
Expand Sky

- Tree
- Ground
- House
- Sky



Courtesy: Pushmeet Kohli

Example: α-expansion

- Variables take label α or retain current label $\bar{\alpha}$
- In one iteration, moves w.r.t. each label $\alpha \in \mathcal{L}$ are made
- Binary energy function for an α move is

$$E^{\alpha}(x^{\alpha}) = \sum_{i \in \mathcal{V}} \phi_i^{\alpha}(x_i^{\alpha}) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}_i} \phi_{ij}^{\alpha}(x_i^{\alpha}, x_j^{\alpha}),$$

where $x_i^{\alpha}, x_j^{\alpha} \in \{0, 1\}$,

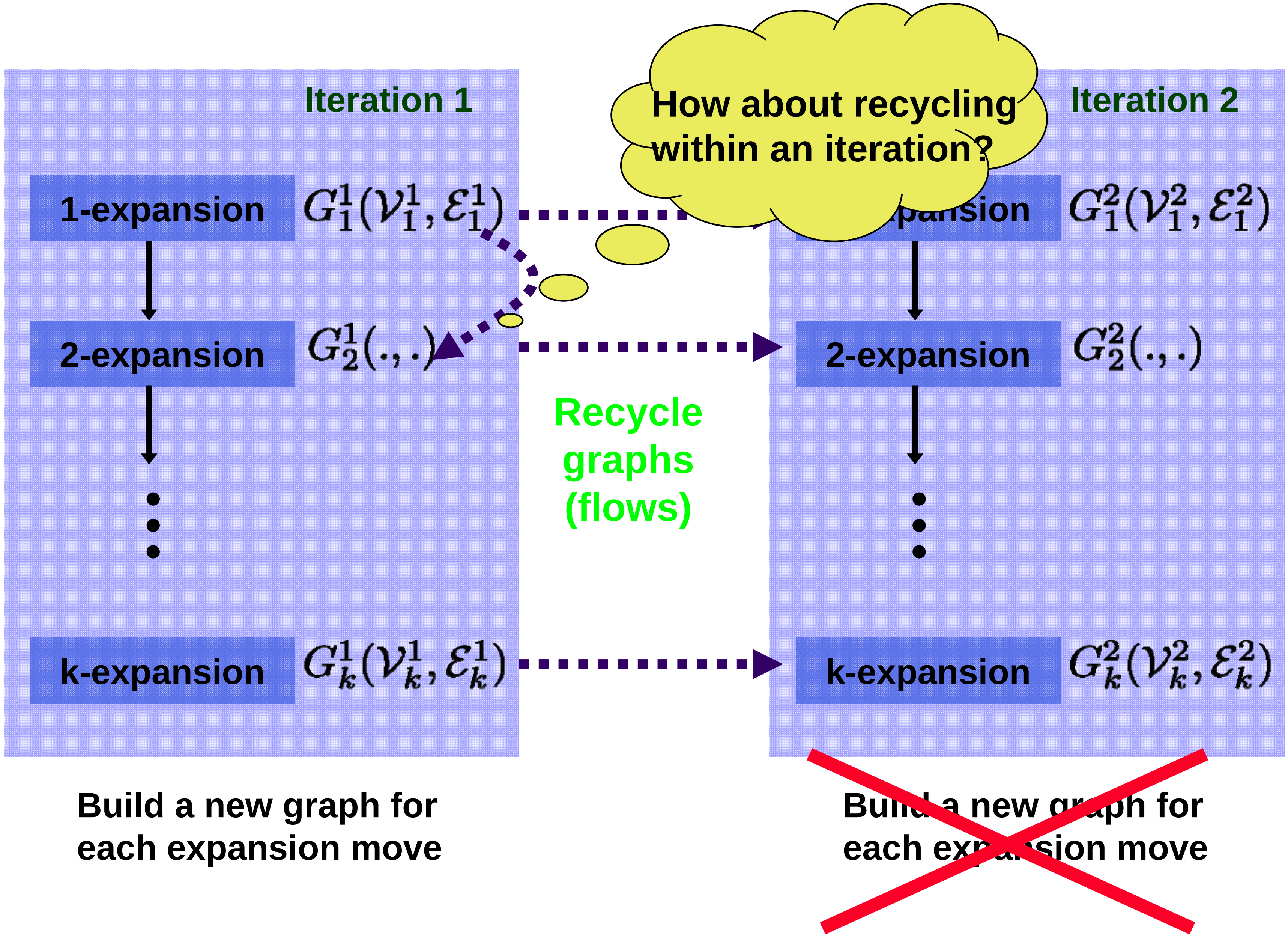
$$\phi_i^{\alpha}(x_i^{\alpha}) = \begin{cases} \phi_i(x_i = \alpha) & \text{if } x_i^{\alpha} = 0, \\ \phi_i(x_i = \bar{\alpha}) & \text{if } x_i^{\alpha} = 1, \end{cases}$$

$$\phi_{ij}^{\alpha}(x_i^{\alpha}, x_j^{\alpha}) = \begin{cases} 0 & \text{if } x_i^{\alpha} = 0, x_j^{\alpha} = 0, \\ \gamma(1 - \delta(x_i - x_j)) & \text{if } x_i^{\alpha} = 1, x_j^{\alpha} = 1, \\ \gamma & \text{otherwise.} \end{cases}$$

Example: α-expansion

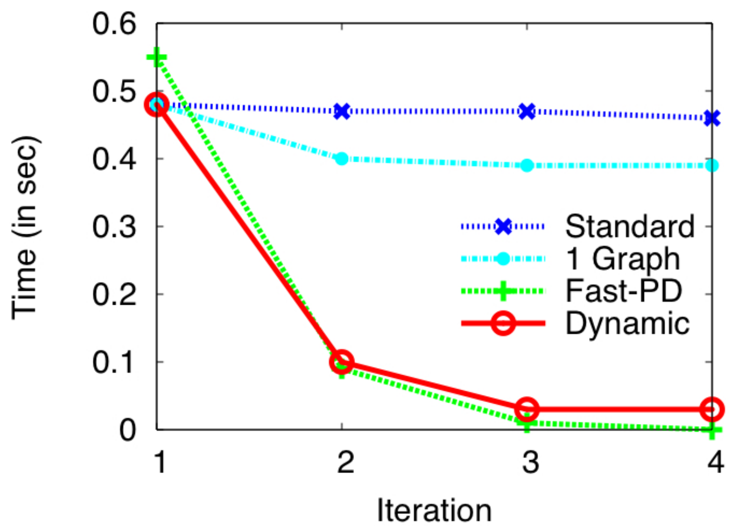
- $E^{\alpha}(x^{\alpha})$ is submodular if $E(x)$ is metric
- Can be solved by the st-mincut/maxflow algorithm
- Primal solution -- labels assigned to x_i^{α}
- Dual solution -- feasible flow solution of the maxflow problem
- How to recycle results
  - Single MRF
  - Dynamic MRF

α-expansion: Single MRF



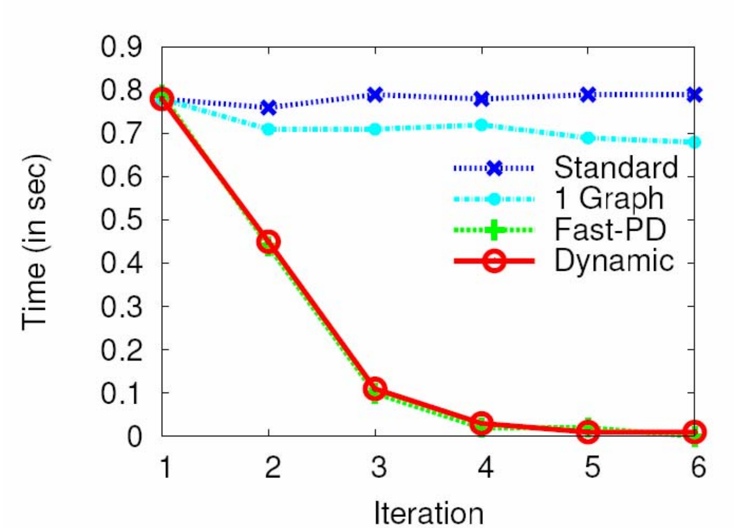
α-expansion: Single MRF

Oxford Brookes Vision Group



Object Segmentation

[Shotton et al. ‘06]



Stereo (Tsukuba)

Total times

Standard: 1.88s

1 Graph: 1.66s

Dynamic: 0.64s

Total times

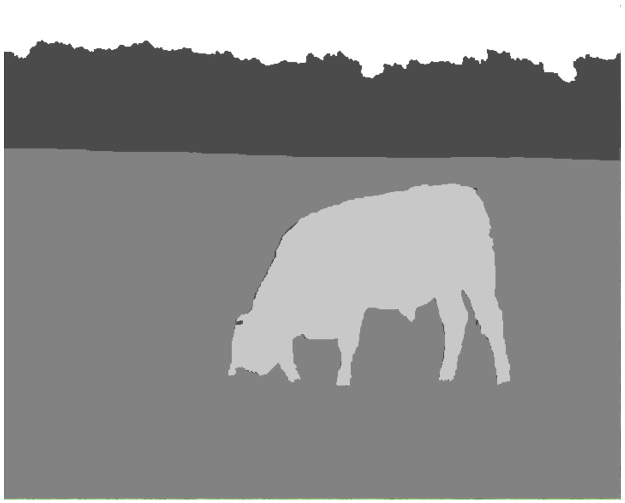
Standard 4.69s

1 Graph: 4.29s

Dynamic: 1.39s

α-expansion: Dynamic MRF

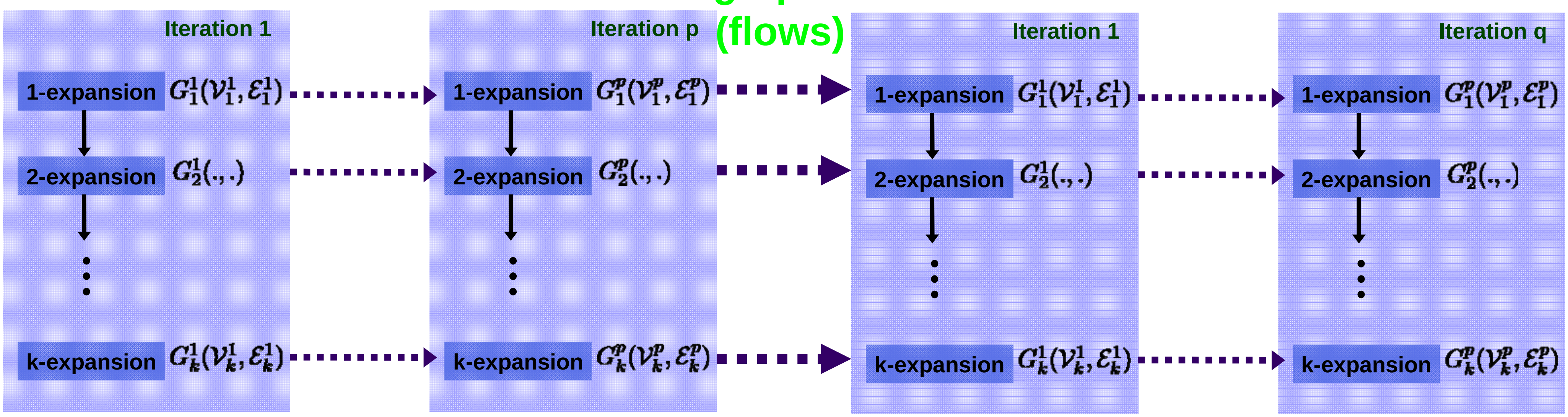
Frame t



Frame t+1

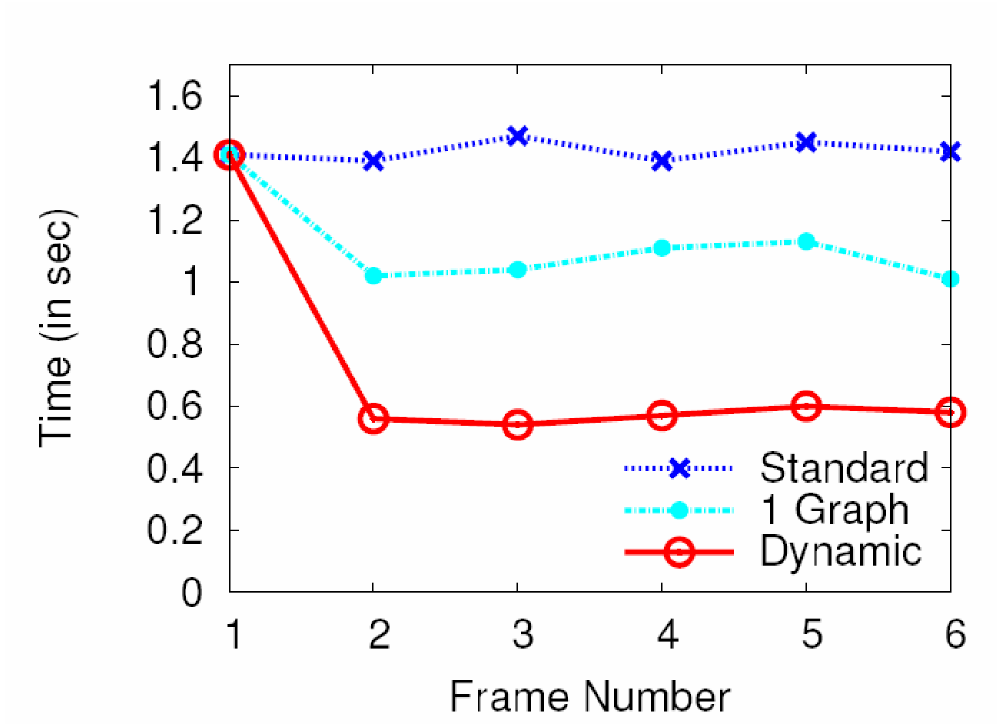
Recycle Labelling

Recycle graphs (flows)



α-expansion: Dynamic MRF

Oxford Brookes Vision Group



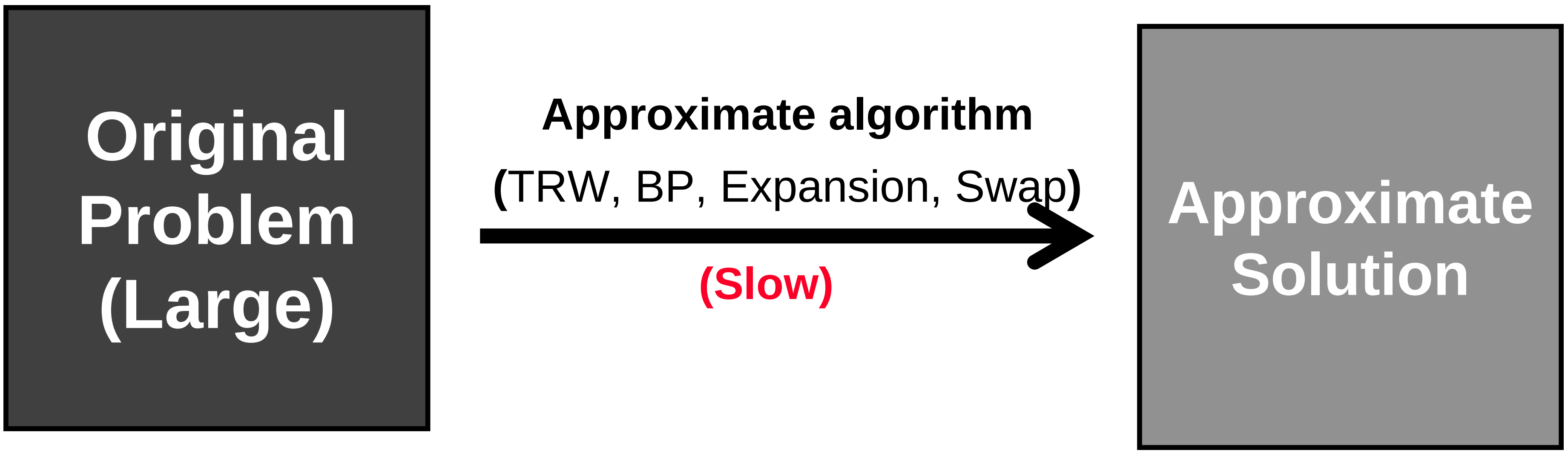
Multi-label Video Segmentation

Roadmap

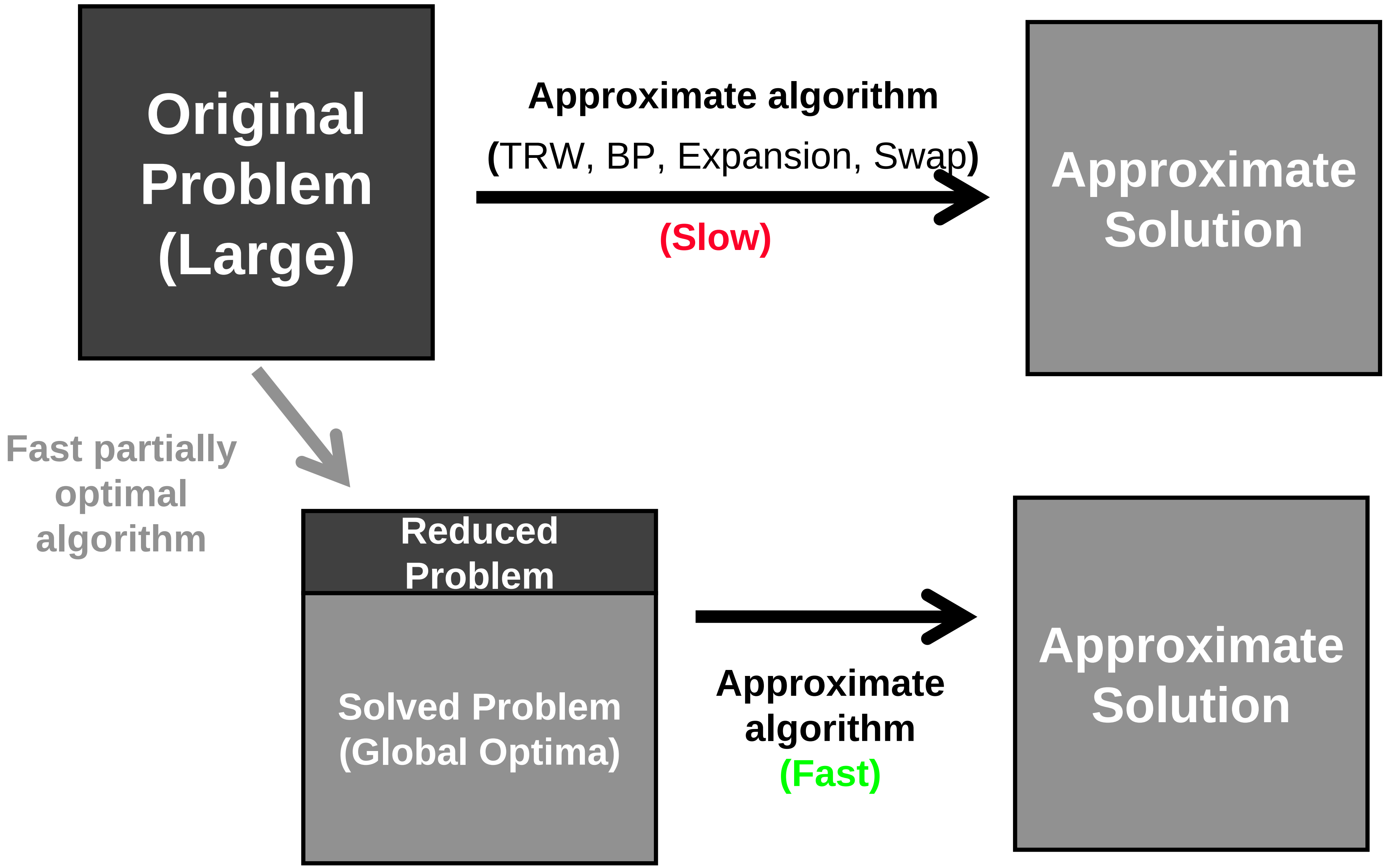
Oxford Brookes Vision Group

- Energy Functions on MRFs
- Solving MRFs
  - Exact Methods
  - Approximate Methods
- Efficient Energy Minimization
  - Recycling
  - Reducing
  - Reusing
- Results

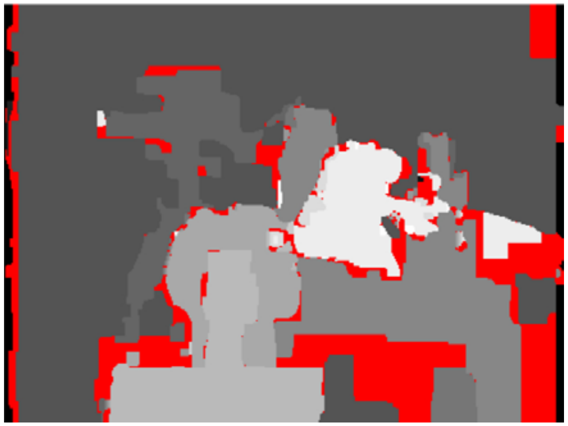
Reducing Energy Functions

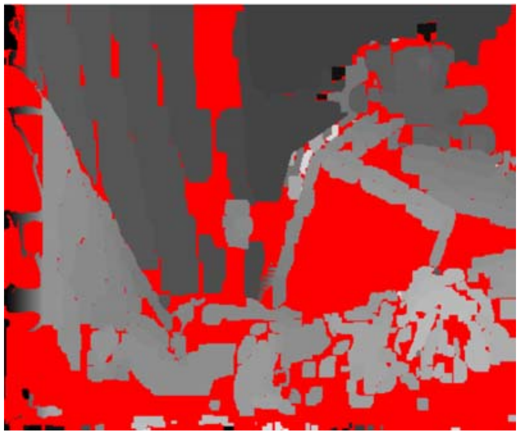


Reducing Energy Functions



Solved Problem : Examples





Partially Optimal Algorithm

- Construction of *k* auxiliary problems
- The energy corresponding to an auxiliary problem $\mathcal{P}_m$

$$E^m(\mathbf{x}^m) = \sum_{i \in \mathcal{V}} \phi_i^m(x_i^m) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}^m(x_i^m, x_j^m),$$

where $x_i^m, x_j^m \in \{0, 1\}$.

$$\phi_i^m(x_i^m) = \begin{cases} \phi_i(x_i = l_m) & \text{if } x_i^m = 0, \\ \phi_i(x_i = l_{\min}) & \text{if } x_i^m = 1, \end{cases}$$

where $l_{\min} = \arg\min_{l \in \mathcal{L} - \{l_m\}} \phi_i(x_i = l)$

$$\phi_{ij}^m(x_i^m, x_j^m) = \begin{cases} 0 & \text{if } x_i^m = 0, x_j^m = 0, \\ 0 & \text{if } x_i^m = 1, x_j^m = 1, \\ \gamma & \text{otherwise.} \end{cases}$$

Kovtun, DAGM'03

Partially Optimal Algorithm

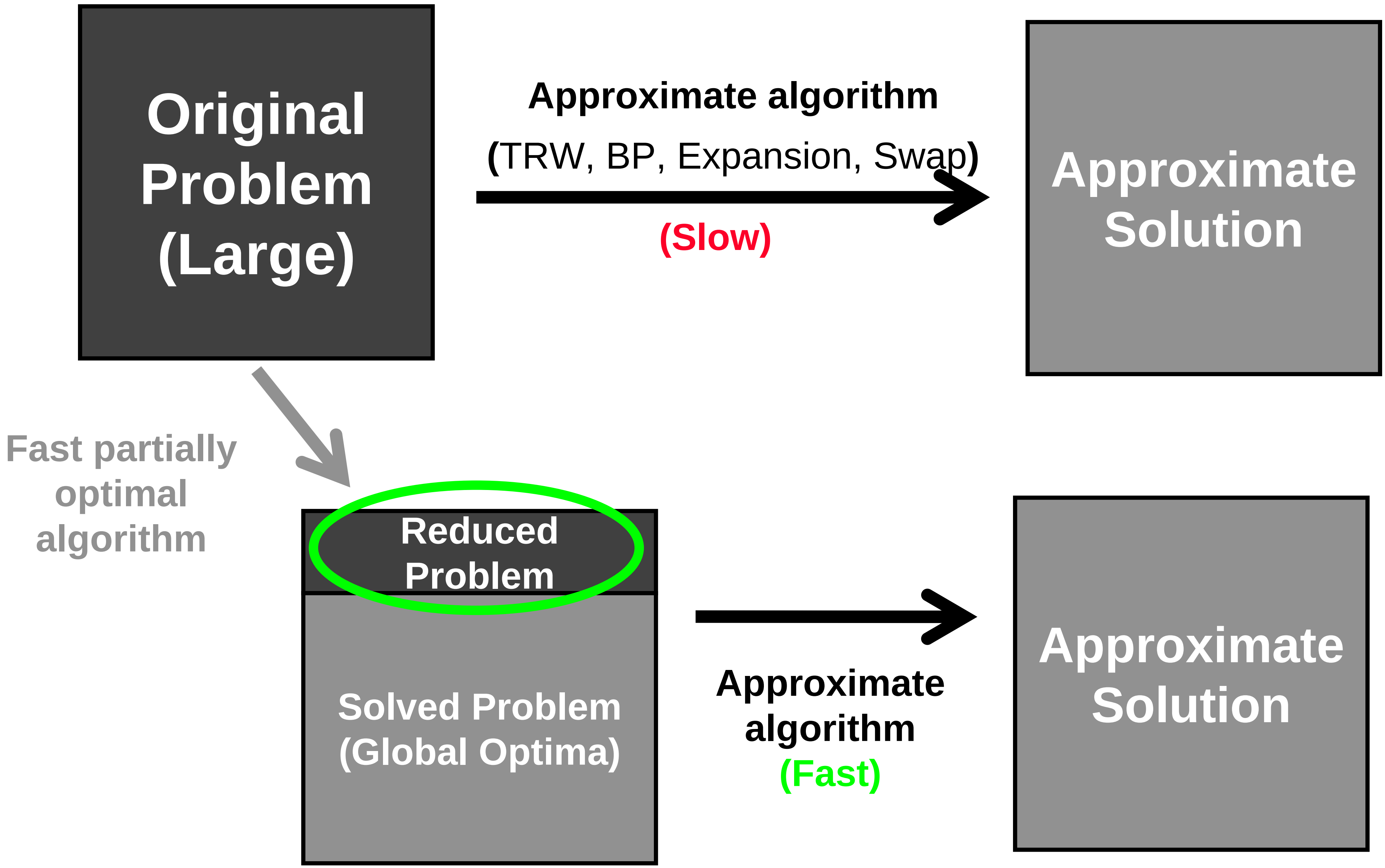
Oxford Brookes Vision Group

- $E^m(\mathbf{x}^m)$ defines a submodular energy function
- The partially optimal solution of $E(\mathbf{x})$w.r.t. $l_m \in \mathcal{L}$ is extracted as:

$$x_i = \begin{cases} l_m & \text{if } x_i^m = 0, \\ \epsilon & \text{otherwise.} \end{cases}$$

- Repeat this for all labels
- For further efficiency: *'Fix'* the partially optimal variables (Project)

Reducing Energy Functions



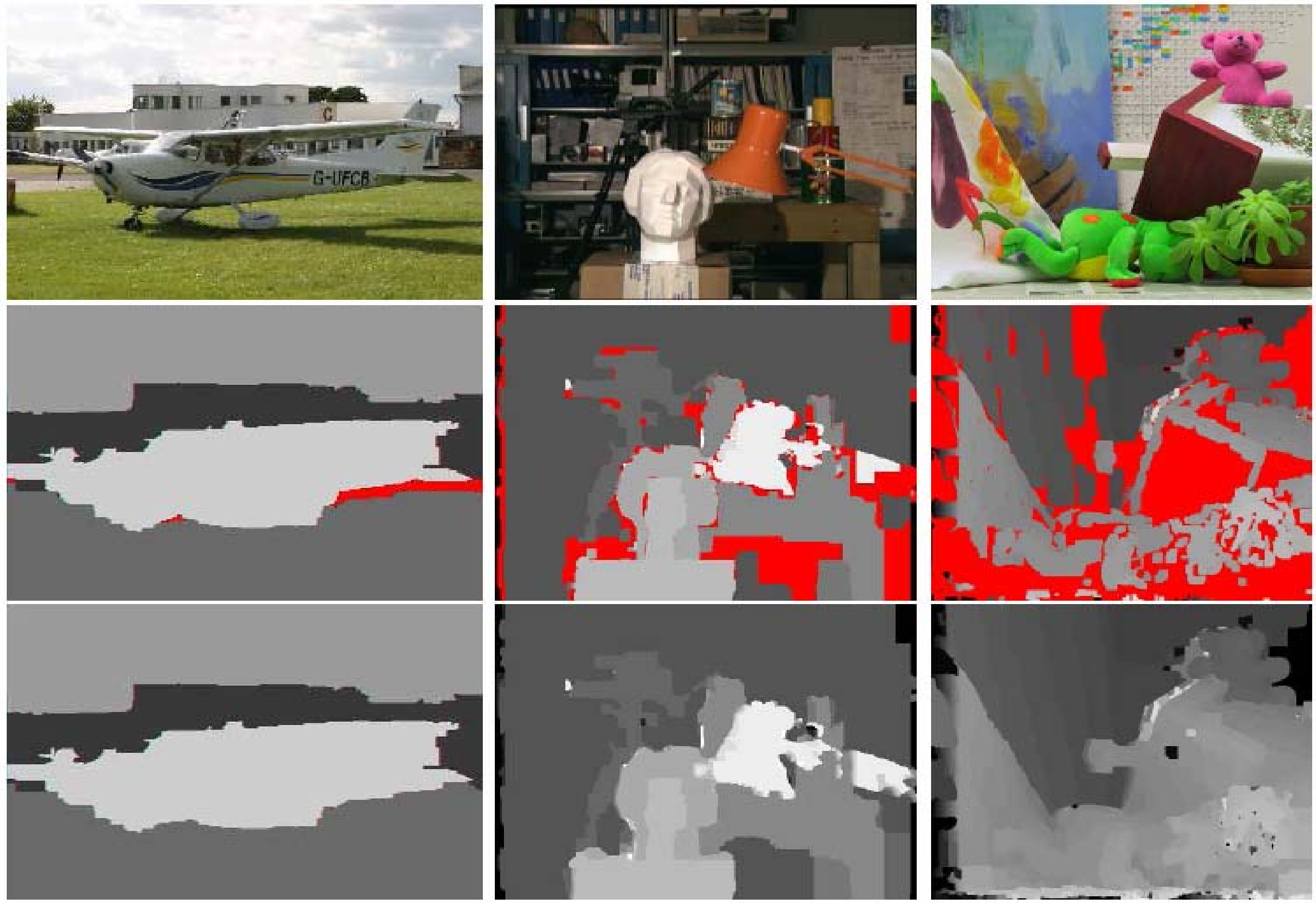
Reducing : Results

Oxford Brookes Vision Group

Image

Partial Labelling

Partial Labelling + TRW



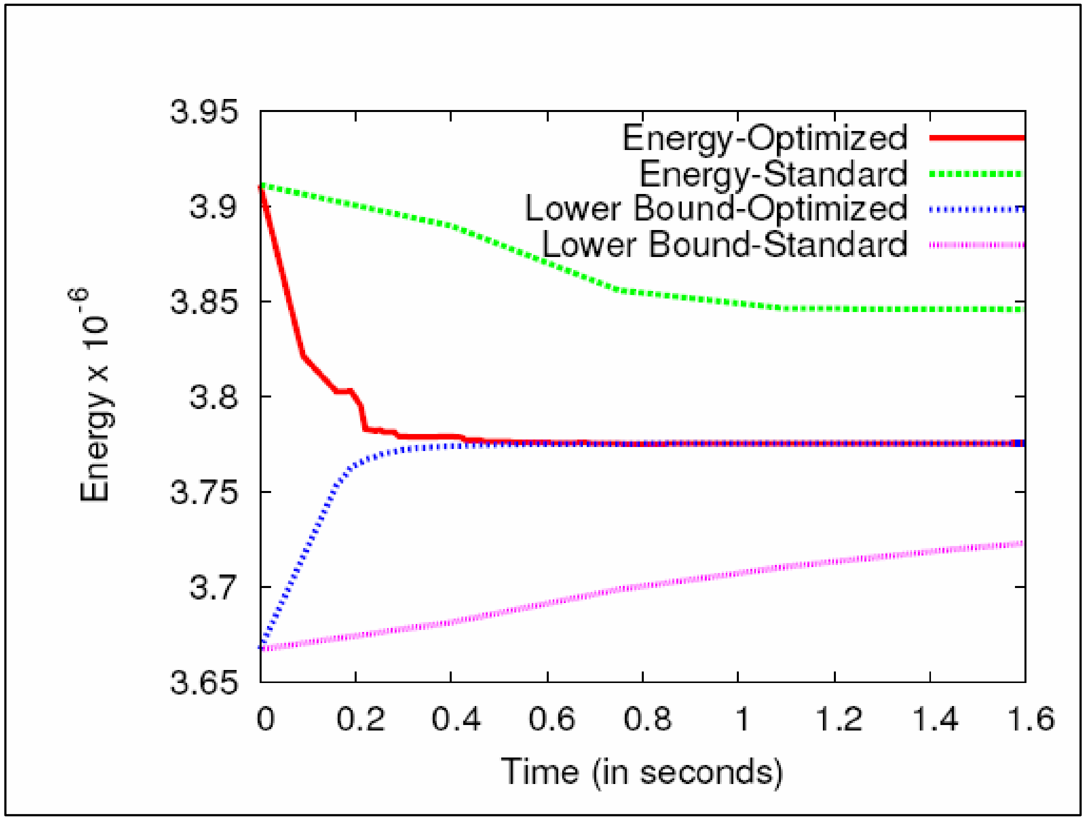
Times (s)	**0.30** 9.89	**4.67** 41.74	**63.77** 182.50

Reducing : Results

Oxford Brookes Vision Group

TRW-S

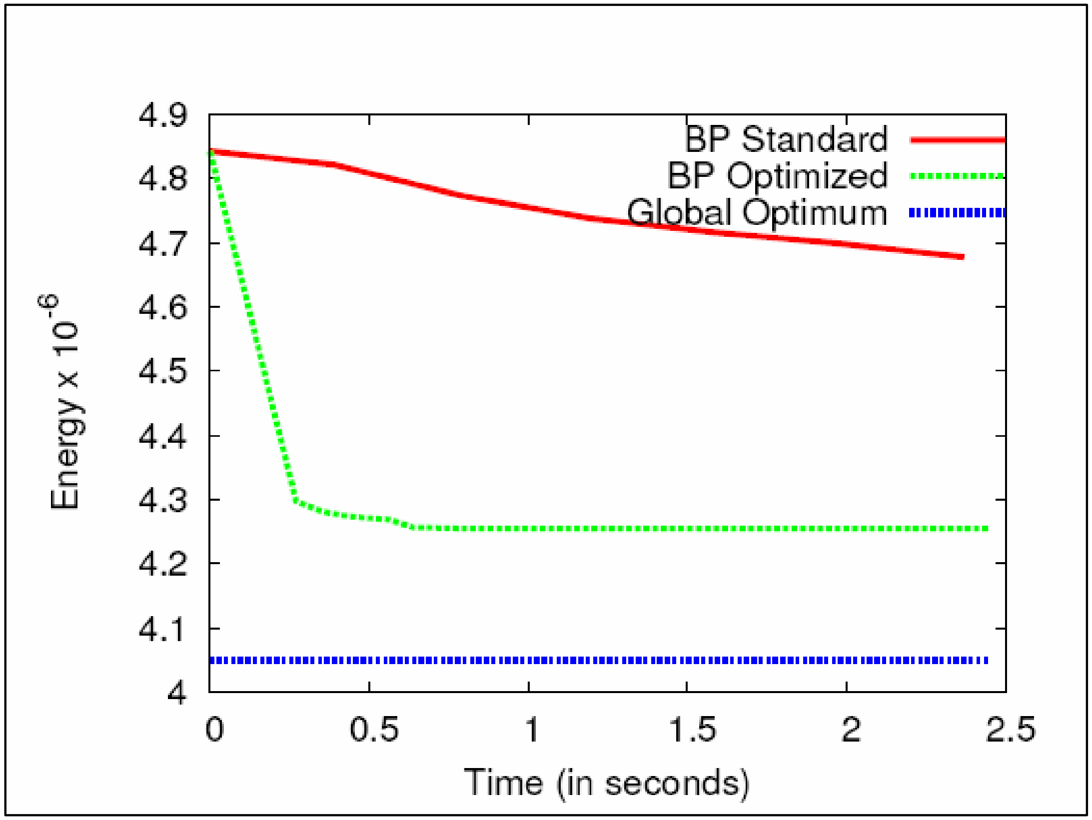
Object Segmentation Problem
(Labels: 5)



Reducing : Results

BP

Object Segmentation Problem
(Labels: 7)



Performance of Par. Opt.

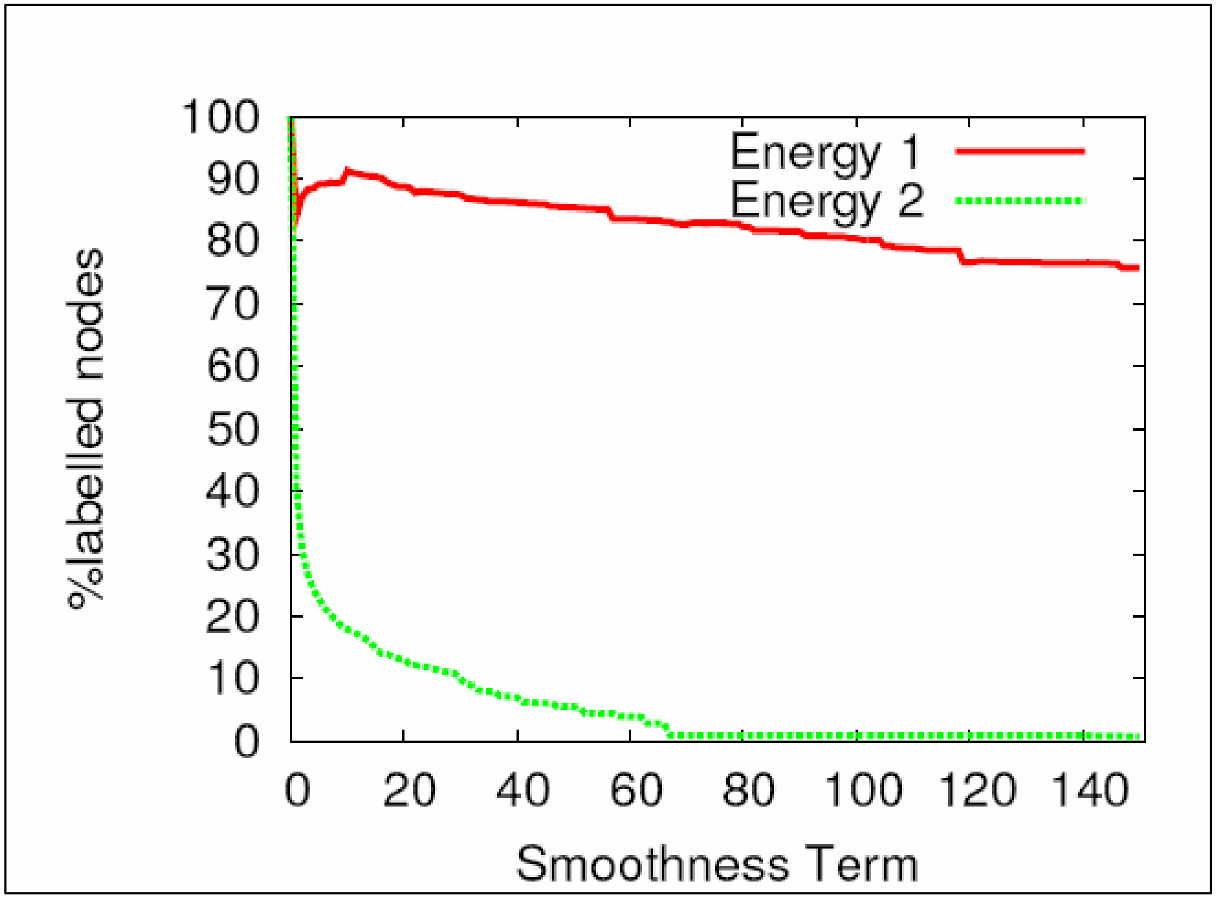
Oxford Brookes Vision Group

- Effect of increase in ‘*difficulty*’

Stereo Problem

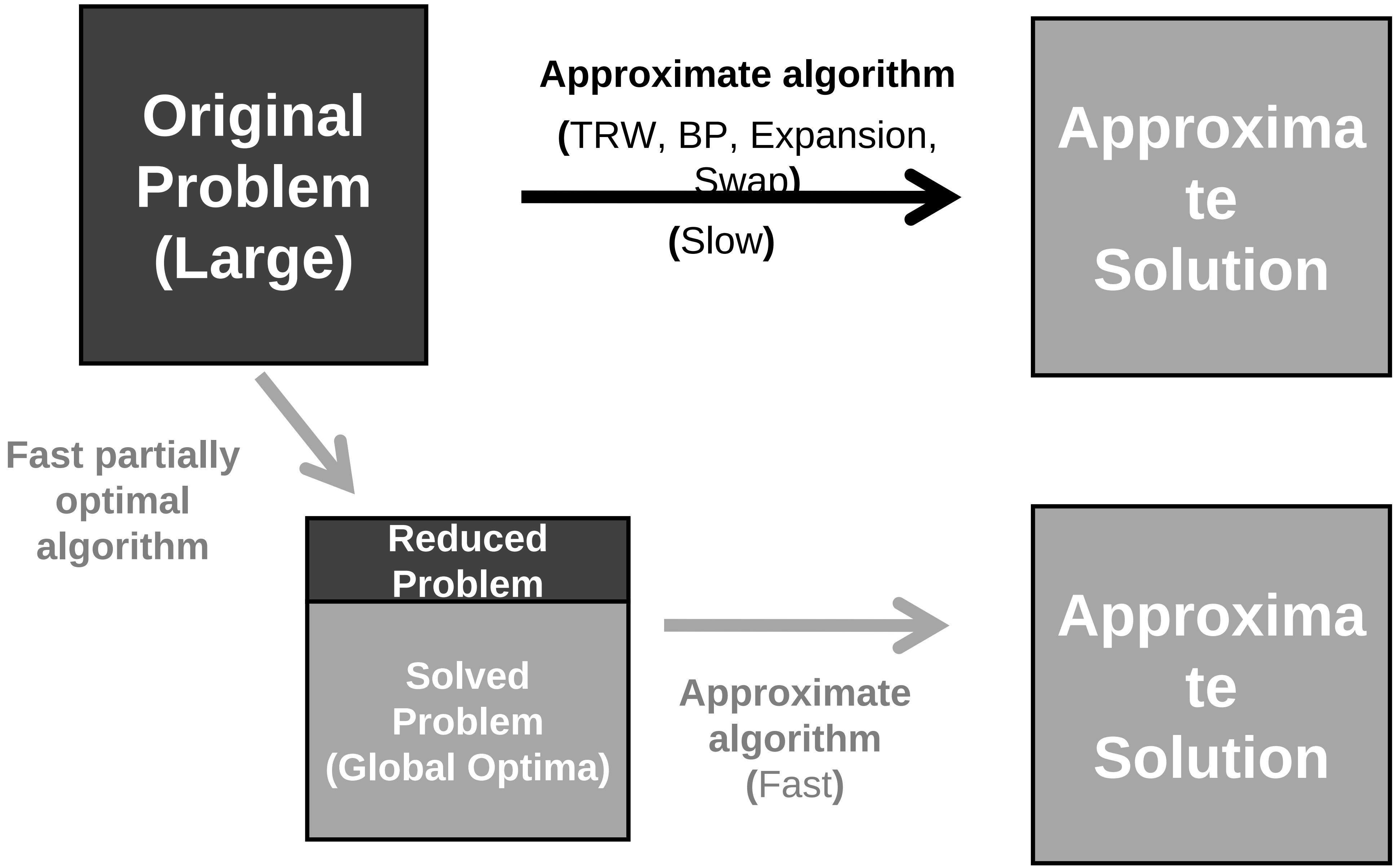
Energy 1:
Kovtun, DAGM’03

Energy 2:
Szeliski et al., ECCV’06



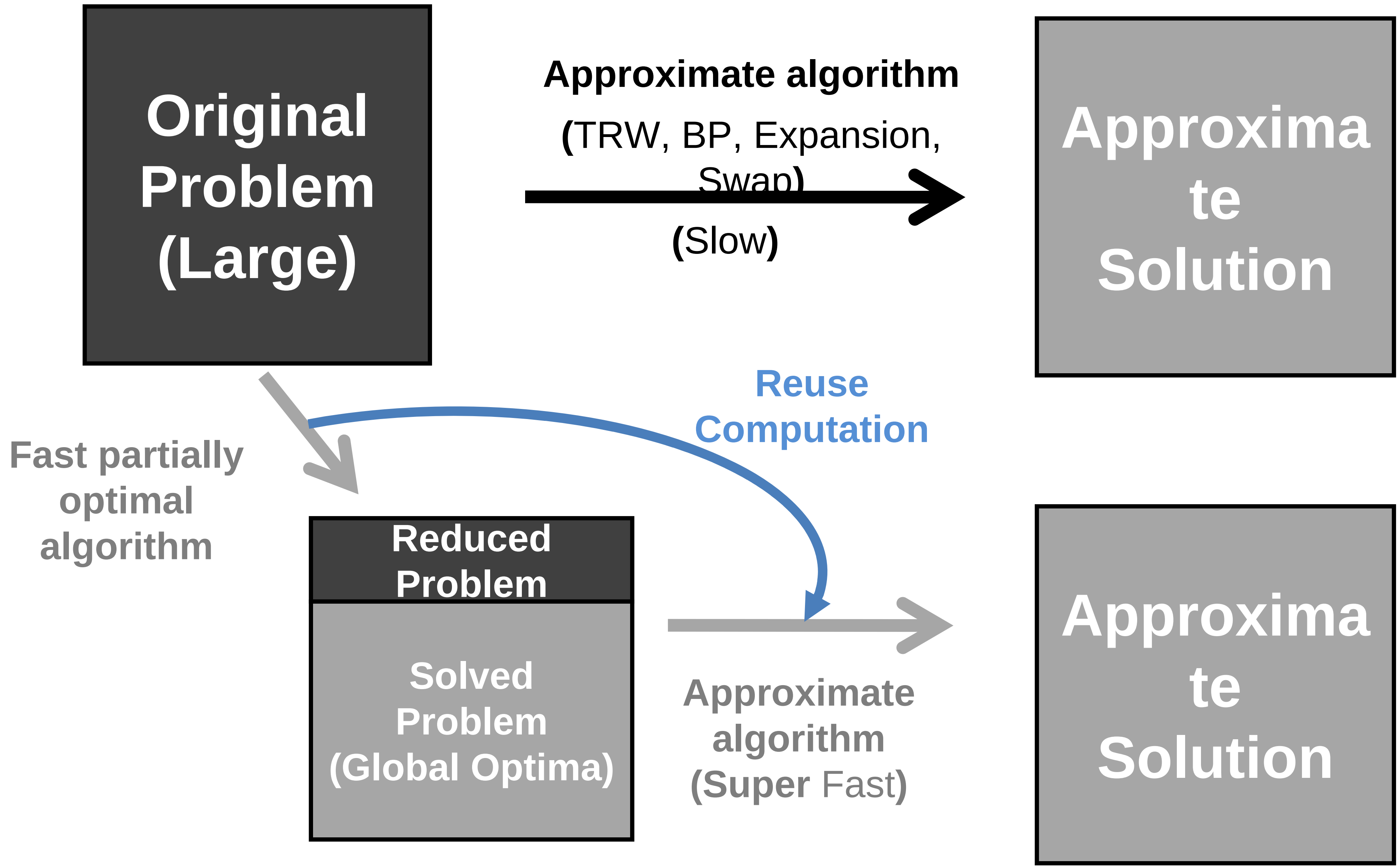
Reducing

Oxford Brookes Vision Group



Reducing

Oxford Brookes Vision Group



Reusing Flow from PO

Oxford Brookes Vision Group

- Key Observation
  - Sub-problems of **PO** and **Expansion** have same form
  - Can be made similar by choosing particular starting configurations for expansions (Reuse flow)
  - Example: Potts (again)

Expansion

$$\phi_i^\alpha(x_i^\alpha) = \begin{cases} \phi_i(\alpha) & \text{if } x_i^\alpha = 0, \\ \phi_i(x_i) & \text{if } x_i^\alpha = 1, \end{cases}$$

$$\phi_{ij}^\alpha(x_i^\alpha, x_j^\alpha) = \begin{cases} 0 & \text{if } x_i^\alpha = 0, x_j^\alpha = 0, \\ \gamma\delta(x_i - x_j) & \text{if } x_i^\alpha = 1, x_j^\alpha = 1, \\ \gamma & \text{otherwise,} \end{cases}$$

PO auxiliary problem

$$\phi_i^m(x_i^m) = \begin{cases} \phi_i(x_i = l_m) & \text{if } x_i^m = 0, \\ \phi_i(x_i = l_{\min}) & \text{if } x_i^m = 1, \end{cases}$$

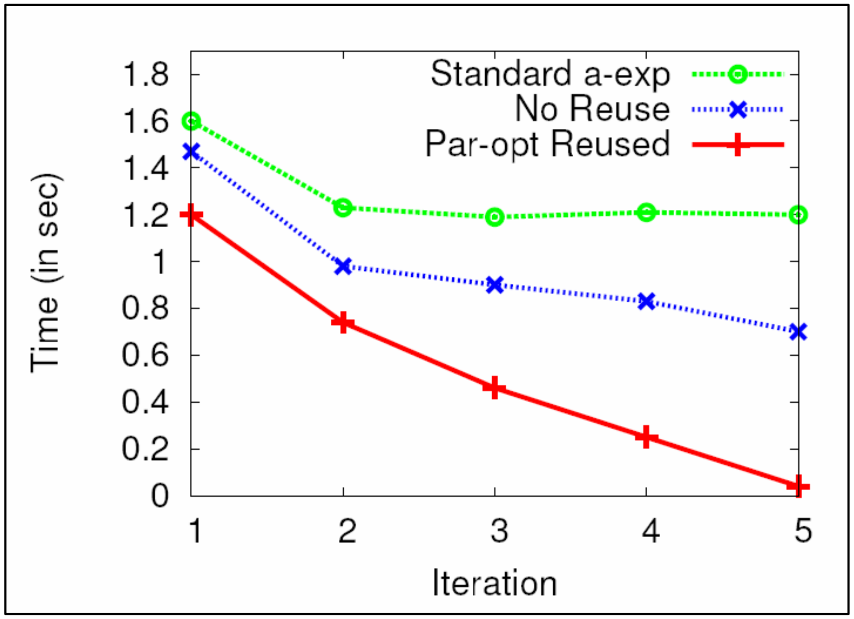
$$l_{\min} = \arg\min_{l \in \mathcal{L} - \{l_m\}} \phi_i(x_i = l)$$

$$\phi_{ij}^m(x_i^m, x_j^m) = \begin{cases} 0 & \text{if } x_i^m = 0, x_j^m = 0, \\ 0 & \text{if } x_i^m = 1, x_j^m = 1, \\ \gamma & \text{otherwise.} \end{cases}$$

Reusing Flow from PO

Oxford Brookes Vision Group

- Key Observation
  - Sub-problems of **PO** and **Expansion** have same form
  - Can be made similar by choosing particular starting configurations for expansions (Reuse flow)
  - Example: Potts (again)



Reducing & Reusing : Results

Oxford Brookes Vision Group

	Time (in seconds)						
	α-exp	Fast-PD	opt α-exp	BP	opt BP	TRW-S	opt TRW-S
Colour-based Segmentation:							
Cow (3)	2.53	1.31	**0.21**	95.93	**0.32**	98.36	**0.33**
Cow (4)	3.75	1.72	**0.38**	108.32	**0.42**	111.69	**0.43**
Garden (4)	0.28	0.14	**0.04**	5.59	**0.17**	5.89	**0.21**
Stereo:							
Tsukuba (16)	5.74	1.47	**0.84**	38.19	**4.47**	41.74	**4.67**
Venus (20)	11.87	3.07	**3.03**	67.04	**14.97**	71.46	**16.02**
Cones (60)	42.23	9.48	**4.36**	173.35	**29.41**	182.66	**30.70**
Teddy (60)	44.25	9.56	**8.27**	172.30	**60.35**	182.50	**63.77**
Texture-based Segmentation:							
Plane (4)	0.39	0.35	**0.15**	9.41	**0.29**	9.89	**0.30**
Bikes (5)	0.82	0.54	**0.22**	10.69	**0.64**	11.19	**0.70**
Road (5)	0.91	0.51	**0.18**	10.67	**0.60**	11.26	**0.62**
Building (7)	1.32	0.89	**0.38**	12.70	**2.57**	13.52	**2.66**
Car (8)	0.99	0.53	**0.11**	13.68	**0.23**	14.42	**0.24**

Thanks

- Questions?