Prize-Collecting Data Fusion for Cost-Performance Tradeoff in Distributed Inference

Anima Anandkumar[1] **Meng Wang**[1] **Lang Tong**[1]
Ananthram Swami[2]

[1]School of ECE, Cornell University, Ithaca, NY.

[2]Army Research Laboratory, Adelphi MD.

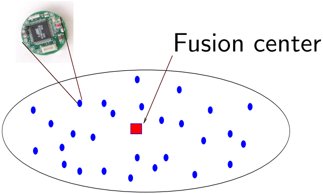
INFOCOM 2009
April 23, 2009

Supported by Army Research Laboratory CTA.

Distributed Statistical Inference

Sensor Network Applications: Statistical Inference

- Sensors: take measurements, e.g., Target, Temperature
- Fusion center: make a final decision

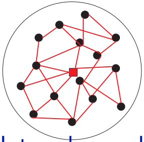


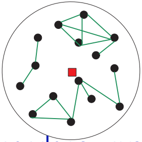
Wireless sensor networks for inference

- Energy constraints
- Measurement selection, inference accuracy
- In-network data fusion

Sensor selection, routing and fusion policies

Optimal Node Selection For Tradeoff

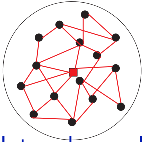




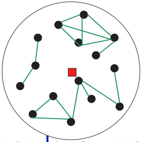
Cost-Performance Tradeoff

- Cost $\equiv$ Total cost of routing with fusion
- Performance degradation $\equiv$ Inference error probability
- Objective $\equiv$Cost $+\ \mu$ Performance degradation, $\quad \mu > 0$

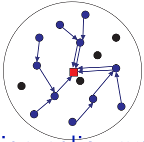
Optimal Node Selection For Tradeoff



Network graph



Dependency graph

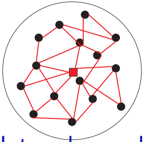


Fusion policy graph

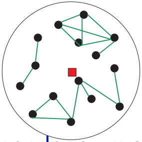
Cost-Performance Tradeoff

- Cost $\equiv$ Total cost of routing with fusion
- Performance degradation $\equiv$ Inference error probability
- Objective $\equiv$Cost $+ \mu$ Performance degradation, $\mu > 0$

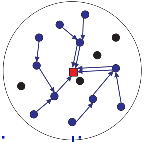
Optimal Node Selection For Tradeoff



Network graph



Dependency graph



Fusion policy graph

Cost-Performance Tradeoff

- Cost $\equiv$ Total cost of routing with fusion
- Performance degradation $\equiv$ Inference error probability
- Objective $\equiv$Cost $+\ \mu$ Performance degradation, $\quad \mu > 0$

Challenges

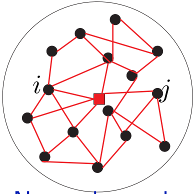
- Presence of Correlation
- Multi-Hop Routing & Fusion
- Optimality: NP-hard, Brute Force?

Outline

Network and Inference Model

Network Model

- Fixed location $\mathbf{V} \stackrel{\Delta}{=} (1, \cdots, n)$.
- Feasible links with cost $C(i, j)$ for link (i, j).

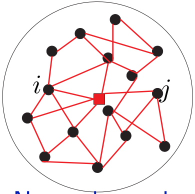


Network graph

Network and Inference Model

Network Model

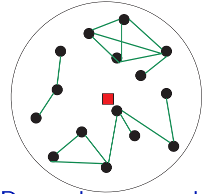
- Fixed location $\mathbf{V} \stackrel{\Delta}{=} (1, \cdots, n)$.
- Feasible links with cost $C(i, j)$ for link (i, j).



Network graph

Inference model

- Sensor measurements $\mathbf{Y_V}$.
- Binary hypothesis: $\mathcal{H}_0$ vs. $\mathcal{H}_1$: $\mathcal{H}_k : \mathbf{Y_V} \sim f(\mathbf{y_v}|\mathcal{H}_k)$



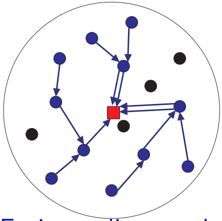
Dependency graph

Optimal Cost-Performance Tradeoff

Problem Statement

- Select $V_s \subset V$ and design a fusion scheme $\Gamma(V_s)$.
- Minimize the total routing costs $C(\Gamma(V_s))$ plus a penalty π based on the error prob. $P_M(V_s)$.

$$\pi(V \backslash V_s) \stackrel{\Delta}{=} \log \frac{P_M(V_s)}{P_M(V)} > 0$$



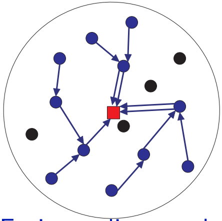
Fusion policy graph

Optimal Cost-Performance Tradeoff

Problem Statement

- Select $V_s \subset V$ and design a fusion scheme $\Gamma(V_s)$.
- Minimize the total routing costs $C(\Gamma(V_s))$ plus a penalty π based on the error prob. $P_M(V_s)$.

$$\pi(V \backslash V_s) \stackrel{\Delta}{=} \log \frac{P_M(V_s)}{P_M(V)} > 0$$



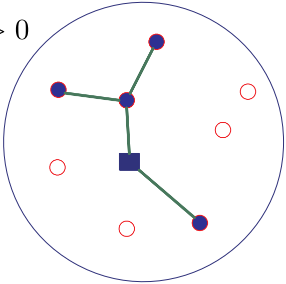
Fusion policy graph

$$\min_{V_s \subset V, \Gamma(V_s)} \Big[C(\Gamma(V_s)) + \mu \pi(V \backslash V_s) \Big], \; \mu > 0$$

Prize-Collecting Data Fusion

Main Results

$$\min_{V_s \subset V, \Gamma(V_s)} \left[C(\Gamma(V_s)) + \mu \log \frac{P_M(V_s)}{P_M(V)}) \right], \ \mu > 0$$



IID measurements

$2 - (|V| - 1)^{-1}$ approximation via Prize-Collecting Steiner Tree

Correlated data: component and clique selection heuristics

- Provable approximation guarantee for special dependency graphs.
- Substantially better than no data fusion.
- Performance under different node placements.

Simplification of the Problem

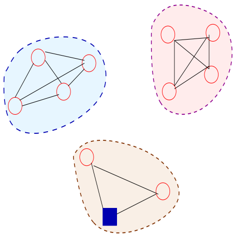
Simplification of the fusion scheme

- Minimal sufficient statistic for $V_s \subset V$
  Log-Likelihood Ratio:
  $\text{LLR}(\mathbf{Y}_{V_s}) = \log \frac{f(\mathbf{Y}_{V_s};\mathcal{H}_0)}{f(\mathbf{Y}_{V_s};\mathcal{H}_1)}$
- Limit to schemes delivering $\text{LLR}(\mathbf{Y}_{V_s})$

Simplification of the Problem

Simplification of the fusion scheme

- Minimal sufficient statistic for $V_s \subset V$
  Log-Likelihood Ratio:
  $\mathsf{LLR}(\mathbf{Y}_{V_s}) = \log \dfrac{f(\mathbf{Y}_{V_s};\mathcal{H}_0)}{f(\mathbf{Y}_{V_s};\mathcal{H}_1)}$
- Limit to schemes delivering $\mathsf{LLR}(\mathbf{Y}_{V_s})$
- $\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c\in\mathcal{C}} \Psi_c(\mathbf{Y}_c)$
  $\mathcal{C}$: the set of maximal cliques

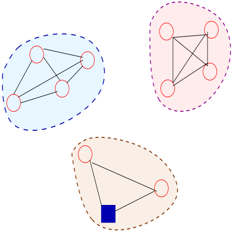


dependency graph

Simplification of the Problem

Simplification of the fusion scheme

- Minimal sufficient statistic for $V_s \subset V$
  Log-Likelihood Ratio:
  $\text{LLR}(\mathbf{Y}_{V_s}) = \log \frac{f(\mathbf{Y}_{V_s}; \mathcal{H}_0)}{f(\mathbf{Y}_{V_s}; \mathcal{H}_1)}$
- Limit to schemes delivering $\text{LLR}(\mathbf{Y}_{V_s})$
- $\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$
  $\mathcal{C}$: the set of maximal cliques



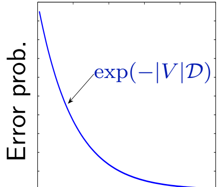
dependency graph

Simplification of the penalty function

$$\pi(V \backslash V_s) \stackrel{\Delta}{=} \log \frac{P_M(V_s)}{P_M(V)}$$

Error exponent approx. in a large network

$$\mathcal{D} \stackrel{\Delta}{=} - \lim_{|V| \to \infty} \frac{1}{|V|} \log P_M(V)$$



Outline

PCDF: IID case

$$\min_{V_s \subset V, \Gamma(V_s)} \left[C(\Gamma(V_s)) + \mu \log \frac{P_M(V_s)}{P_M(V)} \right], \; \mu > 0$$

Simplifications of IID measurements

- $\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$
- $\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \log \frac{f(Y_i;\mathcal{H}_0)}{f(Y_i;\mathcal{H}_1)} = \sum_{i \in V_s} \text{LLR}(\mathbf{Y}_i)$
- Error exponent $\mathcal{D} = D(f(Y;\mathcal{H}_0) || f(Y;\mathcal{H}_1))$

PCDF: IID case

$$\min_{V_s \subset V, \Gamma(V_s)} \left[C(\Gamma(V_s)) + \mu \log \frac{P_M(V_s)}{P_M(V)} \right], \ \mu > 0$$

Simplifications of IID measurements

- $\mathcal{H}_k : \mathbf{Y_V} \sim \prod_{i \in \mathbf{V}} f_k(Y_i)$
- $\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \log \frac{f(Y_i; \mathcal{H}_0)}{f(Y_i; \mathcal{H}_1)} = \sum_{i \in V_s} \text{LLR}(\mathbf{Y}_i)$
- Error exponent $\mathcal{D} = D(f(Y; \mathcal{H}_0) || f(Y; \mathcal{H}_1))$

Modified cost-performance tradeoff for IID

$$\min_{V_s \subset V, \Gamma(V_s)} \left[C(\Gamma(V_s)) + \mu [|V| - |V_s|] D \right]$$

- Asymptotic convergence to the original problem.
- The optimal solution is the Prize Collecting Steiner Tree.

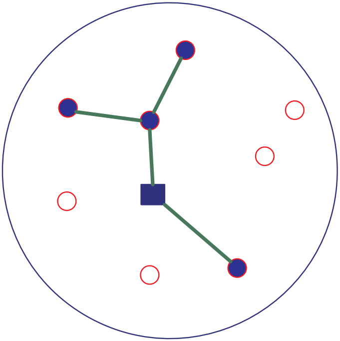
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \Big[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \Big].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



Approx. PCST

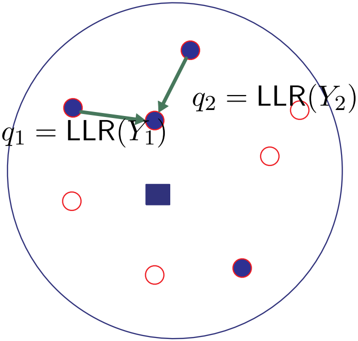
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \Big[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \Big].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



Fusion of IID measurements

$$\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \text{LLR}(\mathbf{Y}_i)$$

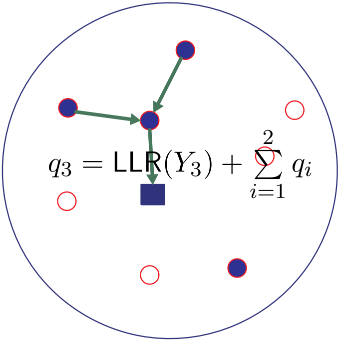
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \left[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \right].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



Fusion of IID measurements

$$\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \text{LLR}(\mathbf{Y}_i)$$

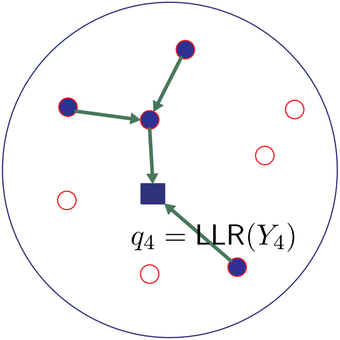
Prize Collecting Steiner Tree (PCST)

Definition

- Tree with minimum sum edge costs plus node penalties not spanned

$$T_* = \arg \min_{T=(V',E')} \left[\sum_{e \in E'} c_e + \sum_{i \notin V'} \pi_i \right].$$

- NP-hard, Goemans-Williamson algorithm has approx. ratio of $2 - \frac{1}{|V|-1}$



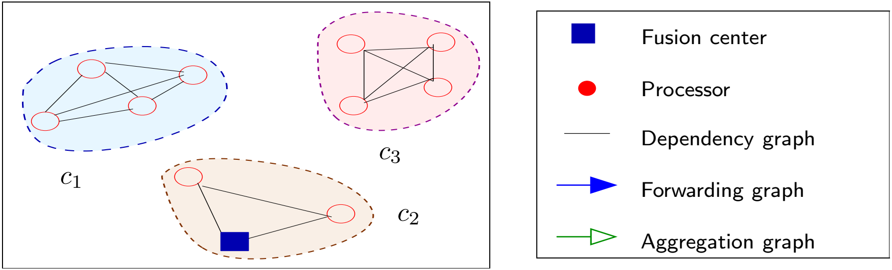
Fusion of IID measurements

$$\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{i \in V_s} \text{LLR}(\mathbf{Y}_i)$$

Outline

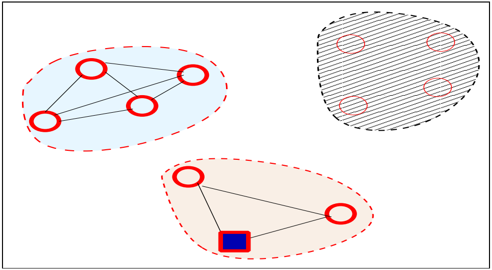
Structure of Fusion Schemes for fixed selection

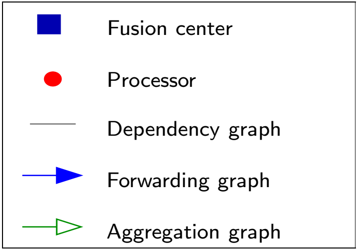
$$\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$



Structure of Fusion Schemes for fixed selection

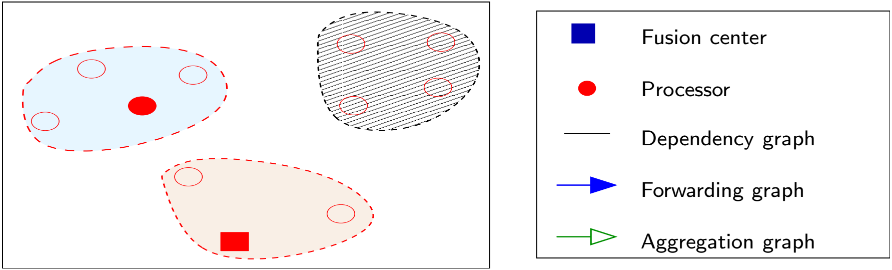
$$\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$





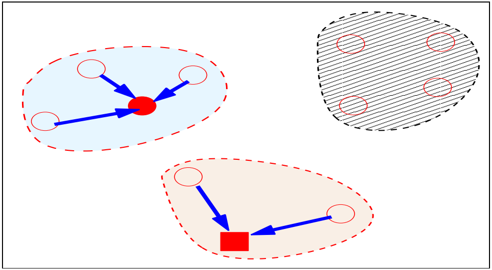
Structure of Fusion Schemes for fixed selection

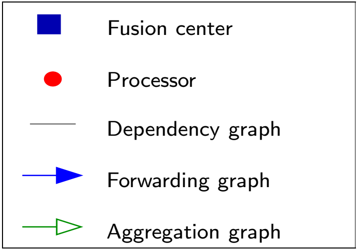
$$\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$



Structure of Fusion Schemes for fixed selection

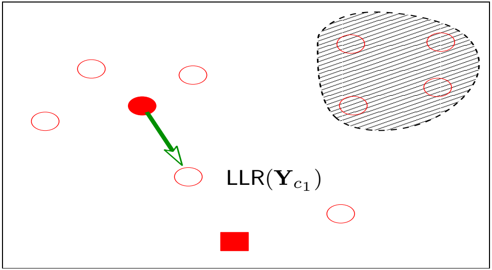
$$\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

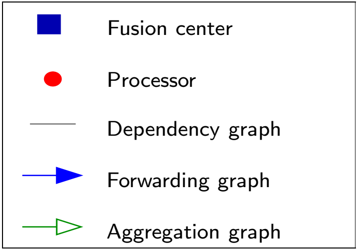




Structure of Fusion Schemes for fixed selection

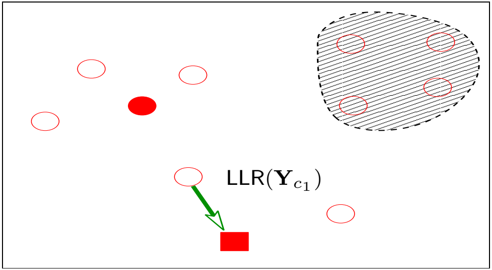
$$\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

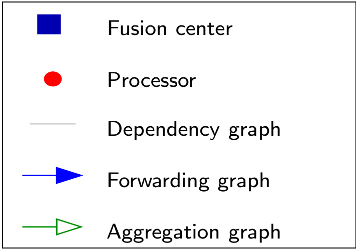




Structure of Fusion Schemes for fixed selection

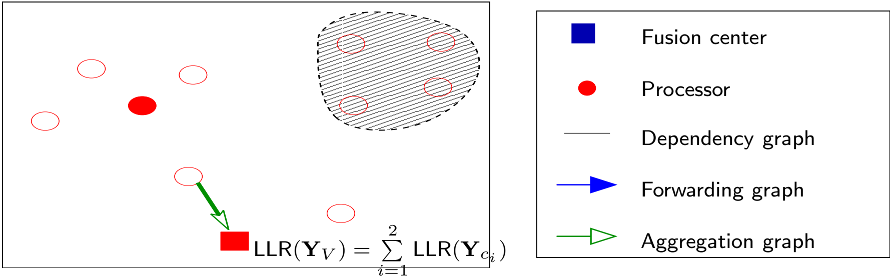
$$\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$





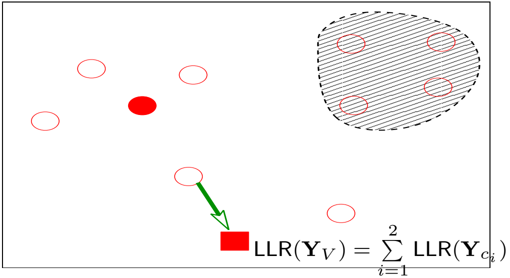
Structure of Fusion Schemes for fixed selection

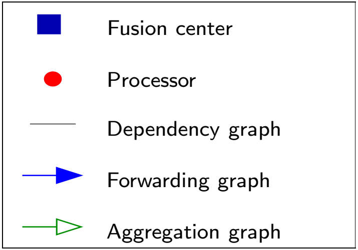
$$\mathrm{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$



Structure of Fusion Schemes for fixed selection

$$\mathsf{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}} \Psi_c(\mathbf{Y}_c)$$

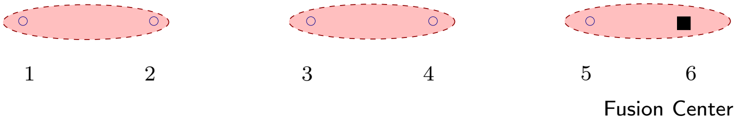




How to select useful groups?

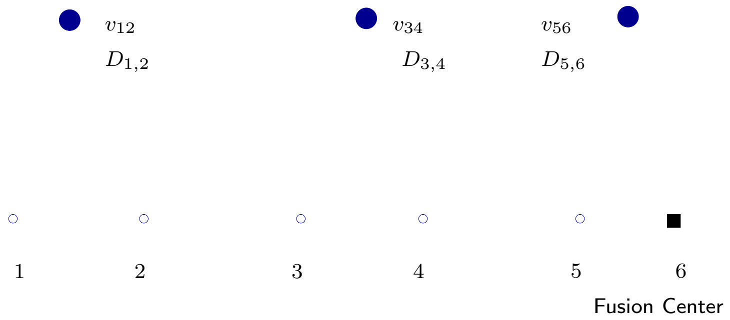
Dependency graph of V_s may be different!

Prize-Collecting Steiner tree (PCST) Reduction



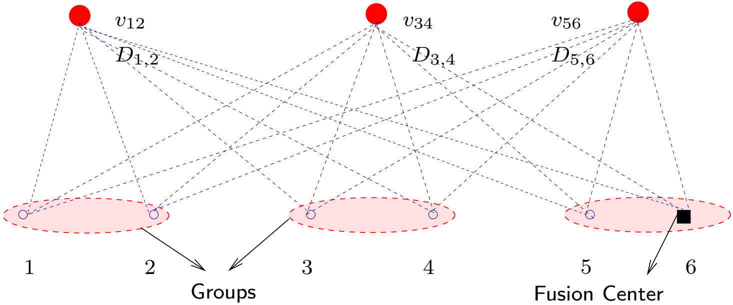
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



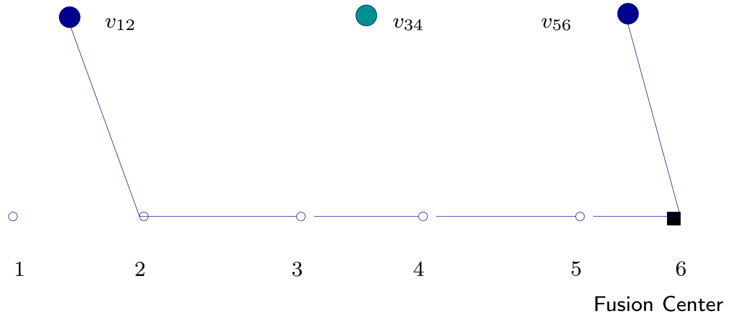
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



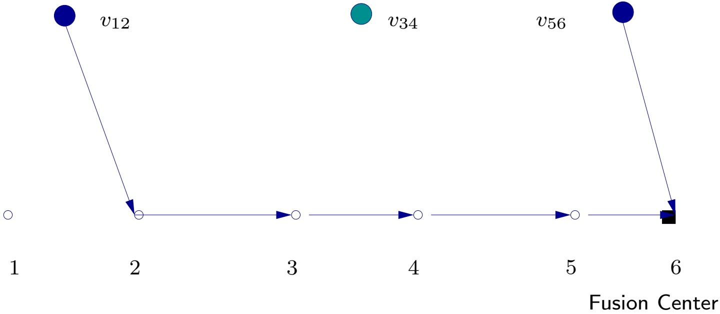
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



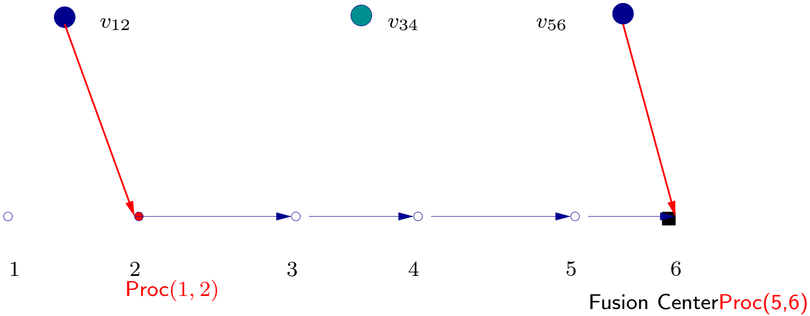
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



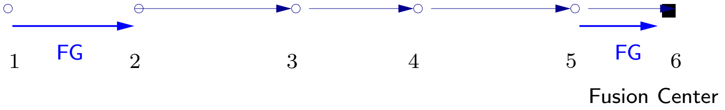
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



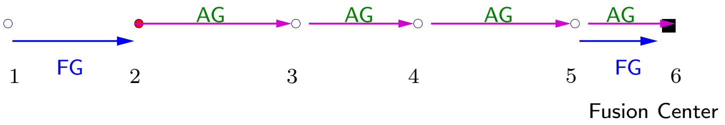
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



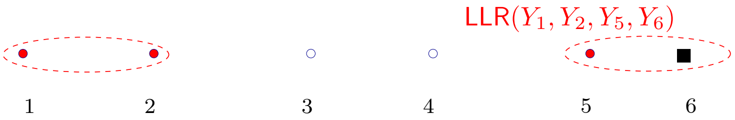
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



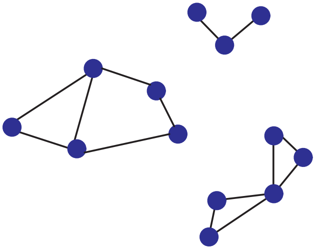
Node selection and data fusion via PCST reduction

Prize-Collecting Steiner tree (PCST) Reduction



Node selection and data fusion via PCST reduction

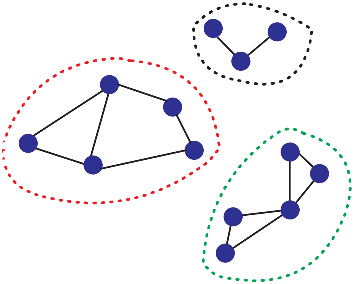
Two Selection Heuristics



Component Selection Heuristic

- Groups = components.
- No new cliques.
- Approximation guarantee.

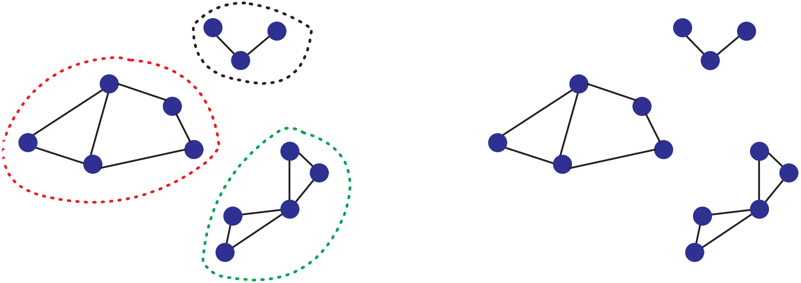
Two Selection Heuristics



Component Selection Heuristic

- Groups = components.
- No new cliques.
- Approximation guarantee.

Two Selection Heuristics



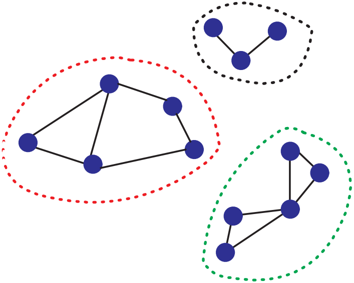
Component Selection Heuristic

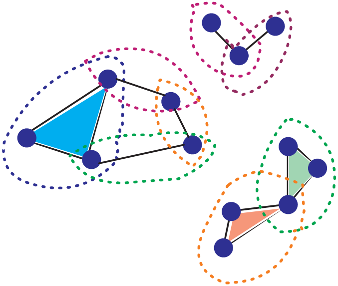
- Groups = components.
- No new cliques.
- Approximation guarantee.

Clique Selection Heuristic

- Groups = cliques.
- New produced cliques.
- No approximation guarantee.

Two Selection Heuristics





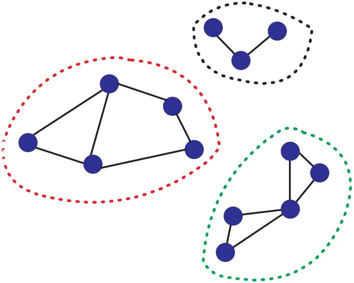
Component Selection Heuristic

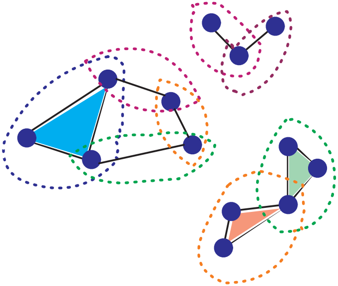
- Groups = components.
- No new cliques.
- Approximation guarantee.

Clique Selection Heuristic

- Groups = cliques.
- New produced cliques.
- No approximation guarantee.

Two Selection Heuristics





Component Selection Heuristic

- Groups = components.
- No new cliques.
- Approximation guarantee.

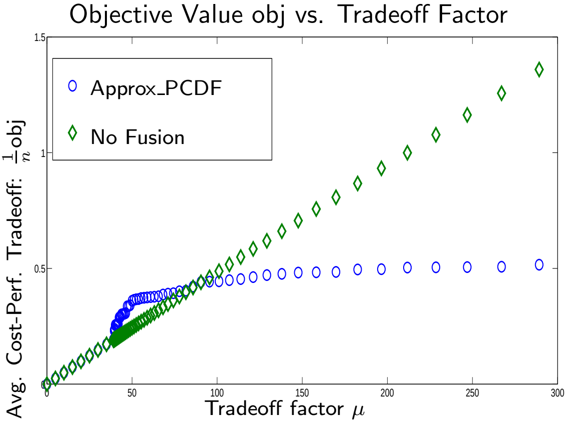
Clique Selection Heuristic

- Groups = cliques.
- New produced cliques.
- No approximation guarantee.

Component selection = clique selection for disjoint cliques.

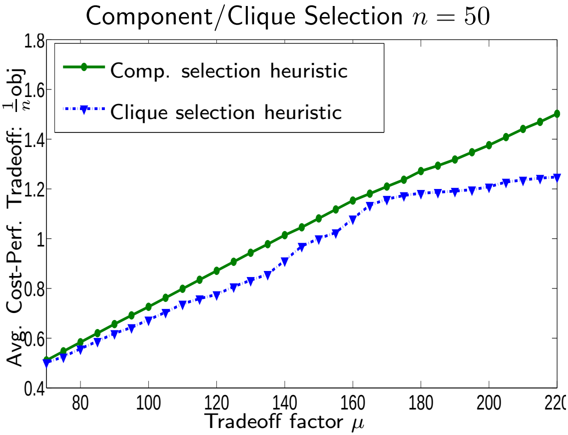
Outline

Simulation: IID Measurements

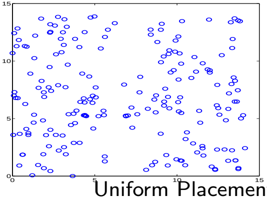


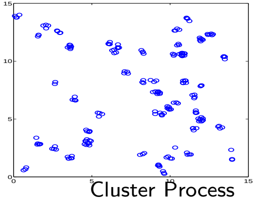
Significant saving compared with no data fusion

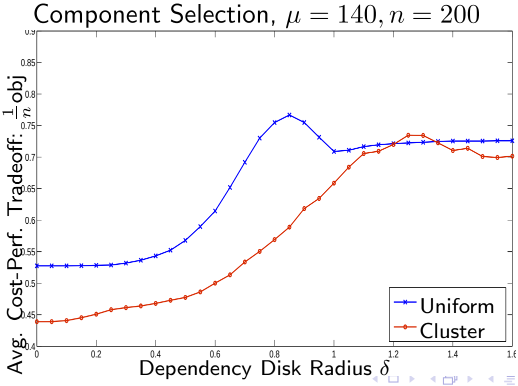
Simulation: Correlated Measurements



Simulation: Correlated Measurements (cont.)







Outline

Conclusion

Summary of Results

- Prize-Collecting Data Fusion for cost-performance tradeoff
- PCST for IID data
- Component and clique selection heuristics for correlated data

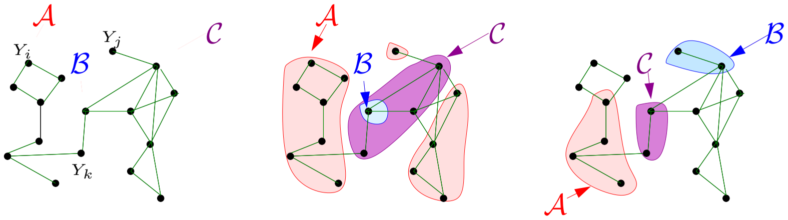
Future directions

- Local selection and coordination
- Realtime measures and delay

http://acsp.ece.cornell.edu/members/anima/pubs/Anandkumar09TR.pdf

Dependency Graph and Markov Random Field

- Consider an undirected graph $\mathcal{G}(\mathbf{V})$, each vertex $V_i \in \mathbf{V}$ is associated with a random variable Y_i
- For any disjoint sets $\mathcal{A}, \mathcal{B}, \mathcal{C}$ such that $\mathcal{C}$ separates $\mathcal{A}$ and $\mathcal{B}$,



$$\mathbf{Y}_{\mathcal{A}} \perp\!\!\!\perp \mathbf{Y}_{\mathcal{B}} | \mathbf{Y}_{\mathcal{C}}$$