Special Topics in AI: Intelligent Agents and Multi-Agent Systems

Distributed Constraint Optimization
(Heuristic approaches, Max-Sum)

Alessandro Farinelli

Approximate Algorithms: outline

- No guarantees
  - DSA-1, MGM-1 (exchange individual assignments)
  - Max-Sum (exchange functions)
- Off-Line guarantees
  - K-optimality and extensions
- On-Line Guarantees
  - Bounded max-sum

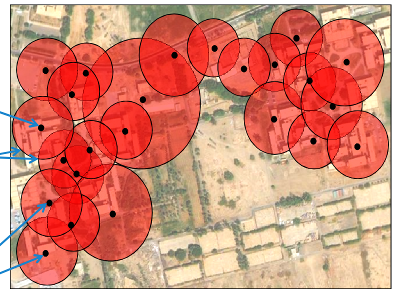
Why Approximate Algorithms

- Motivations
  - Often optimality in practical applications is not achievable
  - Fast good enough solutions are all we can have
- Example – Graph coloring
  - Medium size problem (about 20 nodes, three colors per node)
  - Number of states to visit for optimal solution in the worst case 3^20 = 3 billions of states
- Key problem
  - Provides guarantees on solution quality

Wide Area Surveillance Domain

Sensor detecting vehicles on a road network

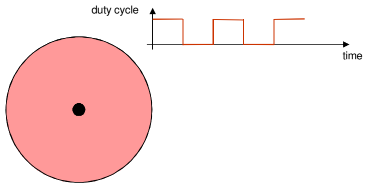
Heterogeneous Sensing range

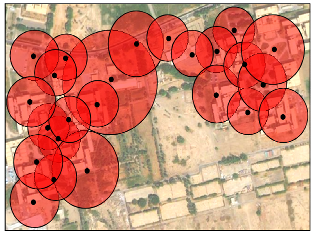


Neighbor agents can communicate

WAS: Model

- Energy Constraints
  - Sense/Sleep modes
  - Recharge when sleeping
  - Energy neutral operation
  - **Contrained on duty cycle**
- Sensor model
  - Activity can be detected by single sensor
- Environment
  - Roads have different traffic loads

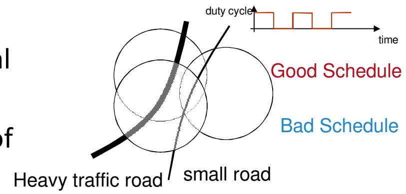




WAS: Goal

Coordinate sensors' duty cycles:

- Achieve energy neutral operations
- Minimize probability of missing vehicles



WAS: system wide utility

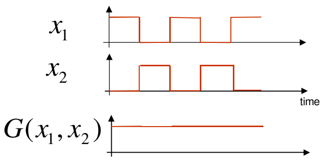
Weighted probability of event detection for each possible joint schedule

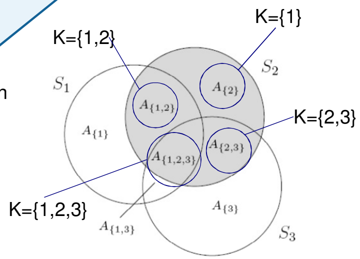
$$U(\mathbf{x}) = \sum_{\mathbf{k} \subseteq \mathcal{S}} A_{\mathbf{k}} \times P(detection | \lambda_d, G(\mathbf{x_k}))$$

$\{x_1, x_2, x_3\}$

Traffic load in the area

Assume Poisson process for event duration

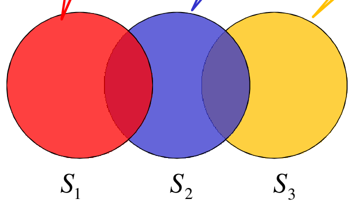




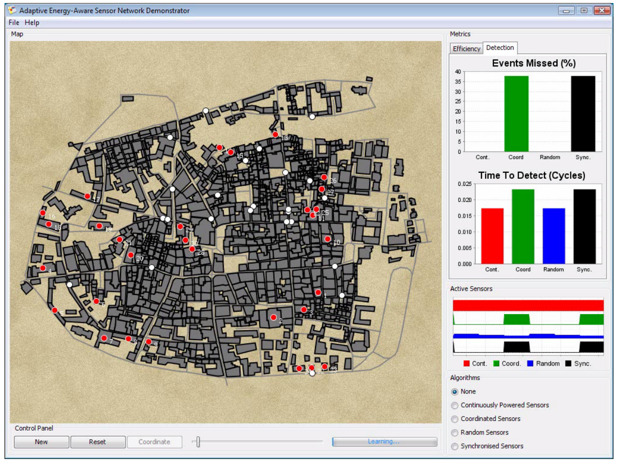
WAS: Interactions among sensors

System wide utility decomposition in individual utilities

$$U(x_1, x_2, x_3) = U_1(x_1, x_2) + U_2(x_2, x_3) + U_3(x_3)$$



Surveillance demo

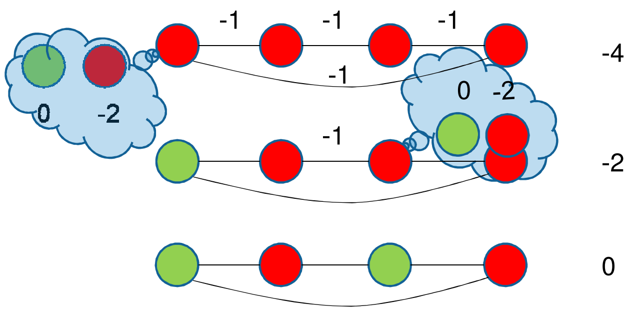


Heuristic approaches

- Local Greedy Search
  - DSA (Distributed Stochastic Algorithm)
  - MGM (Maximum Gain Message)
- Inference-based approaches
  - Max-Sum (GDL family)

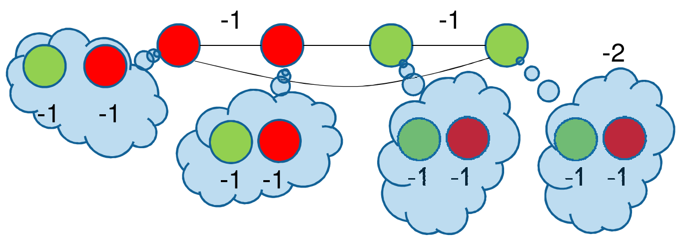
Centralized Local Greedy approaches

- Greedy local search
  - Start from random solution
  - Do local changes if global solution improves
  - Local: change the value of a subset of variables, usually one



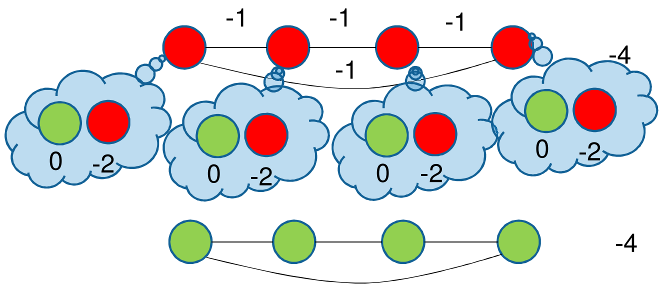
Centralized Local Greedy approaches

- Problems
  - Local minima
  - Standard solutions: RandomWalk, Simulated Annealing



Distributed Local Greedy approaches

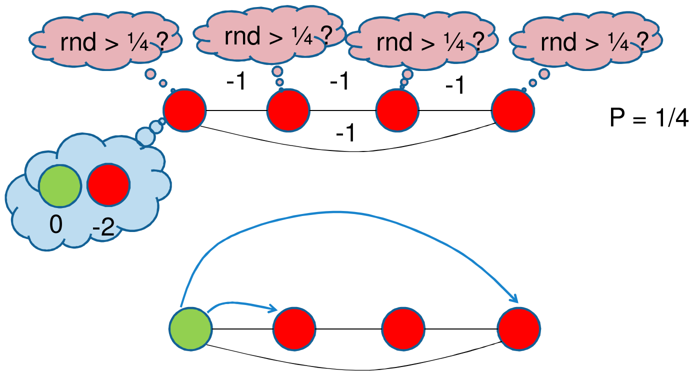
- Local knowledge
- Parallel execution:
  - A greedy local move might be harmful/useless
  - Need coordination



Distributed Stochastic Algorithm

- Greedy local search with activation probability to mitigate issues with parallel executions
- DSA-1: change value of one variable at time
- Initialize agents with a random assignment and communicate values to neighbors
- Each agent:
  - Generates a random number and execute only if rnd less than activation probability
  - When executing changes value maximizing local gain
  - Communicate possible variable change to neighbors

DSA-1: Execution Example



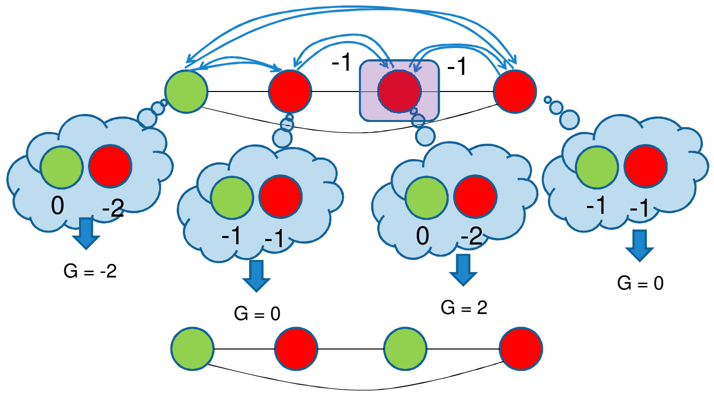
DSA-1: discussion

- Extremely “cheap” (computation/communication)
- Good performance in various domains
  - e.g. target tracking [Fitzpatrick Meertens 03, Zhang et al. 03],
  - Shows an anytime property (not guaranteed)
  - Benchmarking technique for coordination
- Problems
  - Activation probablity must be tuned [Zhang et al. 03]
  - No general rule, hard to characterise results across domains

Maximum Gain Message (MGM-1)

- Coordinate to decide who is going to move
  - Compute and exchange possible gains
  - Agent with maximum (positive) gain executes
- Analysis [Maheswaran et al. 04]
  - Empirically, similar to DSA
  - More communication (but still linear)
  - No Threshold to set
  - Guaranteed to be monotonic (Anytime behavior)

MGM-1: Example

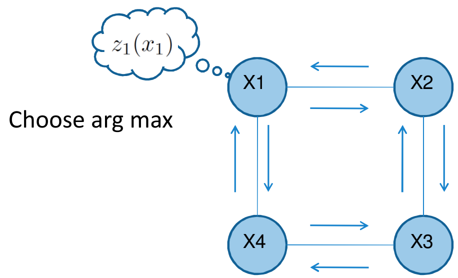


Local greedy approaches

- Exchange local values for variables
  - Similar to search based methods (e.g. ADOPT)
- Consider only local information when maximizing
  - Values of neighbors
- Anytime behaviors
- Could result in very bad solutions

Max-sum

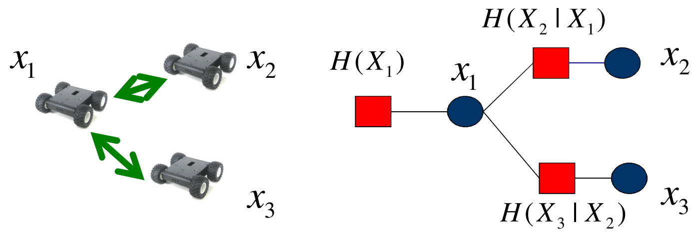
Agents iteratively computes local functions that depend only on the variable they control



Factor Graph

- Computational framework to represent factored computation
- Bipartite graph, Variable - Factor

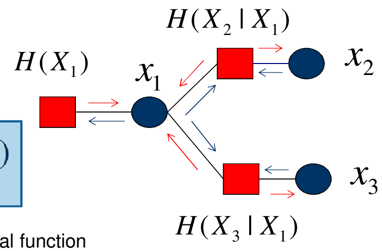
$$H(X_1, X_2, X_3) = H(X_1) + H(X_2 \mid X_1) + H(X_3 \mid X_1)$$



Max-Sum Messages

All incoming messages except xj

$$q_{i \to j}(x_i) = \sum_{k \in adj(i) \setminus j} r_{k \to i}(x_i)$$



Local function

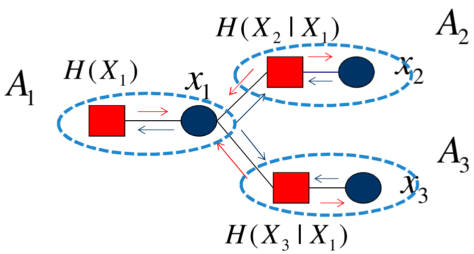
$$r_{j \to i}(x_i) = \max_{\mathbf{x}_j \setminus i} \left[U_j(\mathbf{x}_j) + \sum_{k \in adj(j) \setminus i} q_{k \to j}(x_k) \right]$$

Max over all variables except xi

All incoming messages except xi

Agents and the Factor Graph

- Variables and functions must be allocated to agents
- Allocation does not impact on solution quality
- Allocation does impact on load distribution/communication



Max-Sum Assignments

- At each iteration, each agent
  - Computes its local function

$$z_i(x_i) = \sum_{k \in adj(i)} r_{k \to i}(x_i)$$

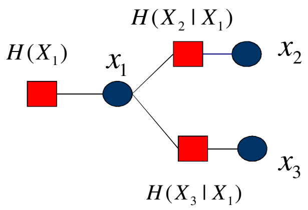
All incoming messages

- Sets its assignment as the value that maximizes its local function

$$\tilde{x}_i = \arg\max_{x_i} z_i(x_i)$$

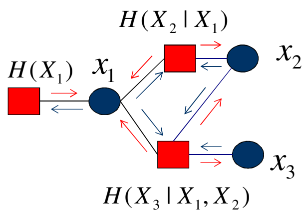
Max-Sum on Acyclic Graphs

- Convergence guaranteed in a polynomial number of cycles
- Optimal
  - Different branches are independent
  - Z functions provide correct estimation
  - Need Value propagation to break simmetries



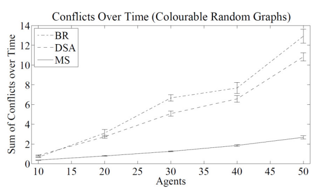
Max-Sum on Cyclic Graphs

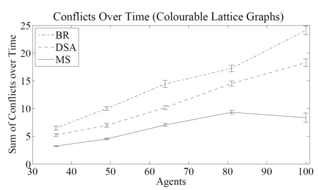
- Convergence NOT guaranteed
- When convergence it does converge to a neighbourhood maximum
- Neighbourhood maximum:
  - greater than all other maxima within a particular region of the search space



(Loopy) Max-sum Performance

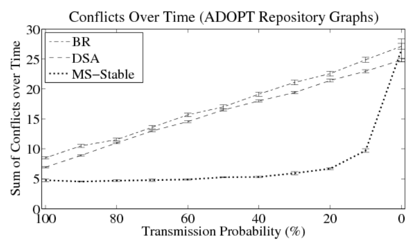
- Good performance on loopy networks [Farinelli et al. 08]
  - When it converges very good results
    - Interesting results when only one cycle [Weiss 00]
  - We could remove cycle but pay an exponential price (see DPOP)
  - Java Library for max-sum http://code.google.com/p/jmaxsum/



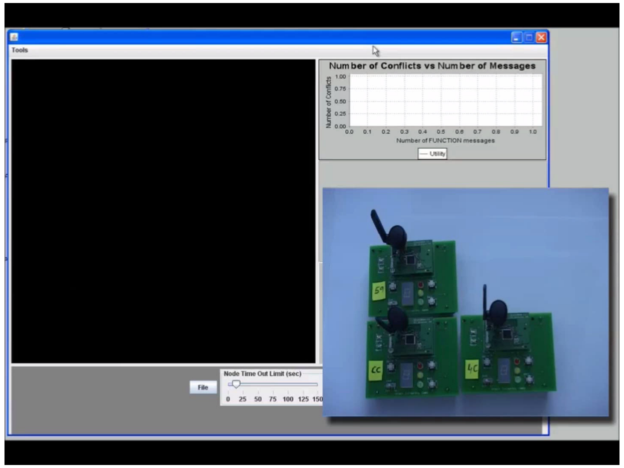


Max-Sum for low power devices

- Low overhead
  - Msgs number/size
- Asynchronous computation
  - Agents take decisions whenever new messages arrive
- Robust to message loss



Max-sum on hardware



GDL and Max-Sum

- Generalized Distributive Law (GDL)
  - Unifying framework for inference in Graphical Models
  - Based on mathematical properties of semi-rings
  - Widely used in
    - Information theory (turbo codes)
    - Probabilistic Graphical models (belief propagation)

	K	"$(+,0)$"	"$(\cdot,1)$"	short name
1.	A	$(+,0)$	$(\cdot,1)$	
2.	$A[x]$	$(+,0)$	$(\cdot,1)$	
3.	$A[x,y,\ldots]$	$(+,0)$	$(\cdot,1)$	
4.	$[0,\infty)$	$(+,0)$	$(\cdot,1)$	sum-product
5.	$(0,\infty]$	$(\min,\infty)$	$(\cdot,1)$	min-product
6.	$[0,\infty)$	$(\max,0)$	$(\cdot,1)$	max-product
7.	$(-\infty,\infty]$	$(\min,\infty)$	$(+,0)$	min-sum
8.	$[-\infty,\infty)$	$(\max,-\infty)$	$(+,0)$	max-sum
9.	$\{0,1\}$	$(\mathrm{OR},0)$	$(\mathrm{AND},1)$	Boolean
10.	2^S	$(\cup,\emptyset)$	$(\cap,S)$	
11.	Λ	$(\vee,0)$	$(\wedge,1)$	
12.	Λ	$(\wedge,1)$	$(\vee,0)$.	

GDL basic ideas

- Max-Sum semi-ring
  - Sum Distributes over Max
  - We can efficiently maximise a summation

Distributive Law

$$a+\max(b,c)=\max(a+b,a+c)$$

$$\max_{\bar{x}} \sum_i f_i(\bar{x}_i)$$

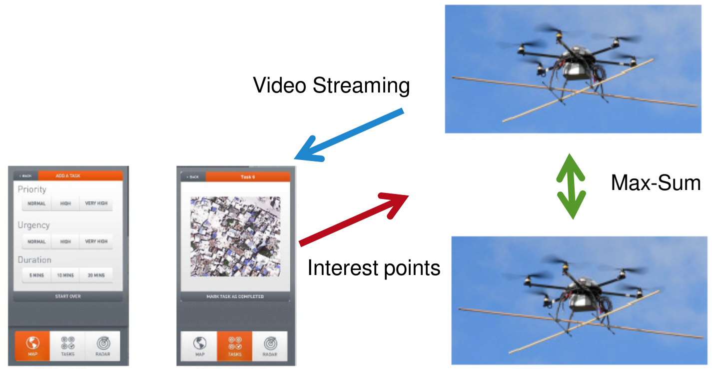
Generalized Distributive Law

$$\otimes(a,\oplus(b,c))=\oplus(\otimes(a,b),\otimes(a,c))$$

$$\oplus_{\bar{x}}(\otimes_i f_i(\bar{x}_i))$$

Max-Sum for UAVs

Task Assignment for UAVs [Delle Fave et al 12]



(a) Task Input (b) Task Monitor

Task utility

Priority

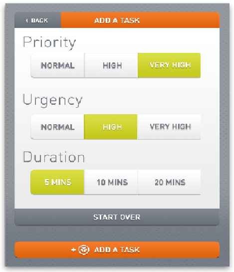
Task completion

$$U_j(X_j) = p_j \, u_j^{\,t-t_j^0} \, [1 - e^{-\lambda_j.(t_2 - t_1)}]$$

Urgency

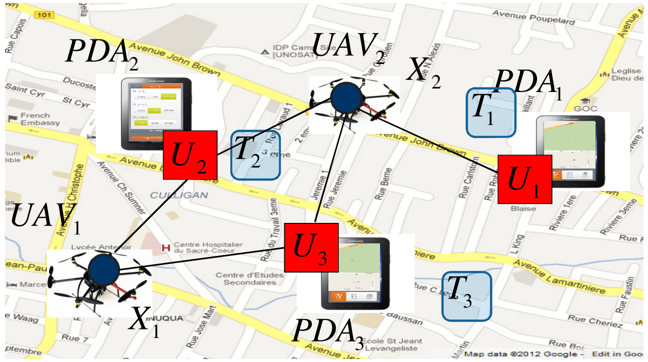
First assigned UAVs reaches task

Last assigned UAVs leaves task (consider battery life)



33

Factor Graph Representation



UAVs Demo

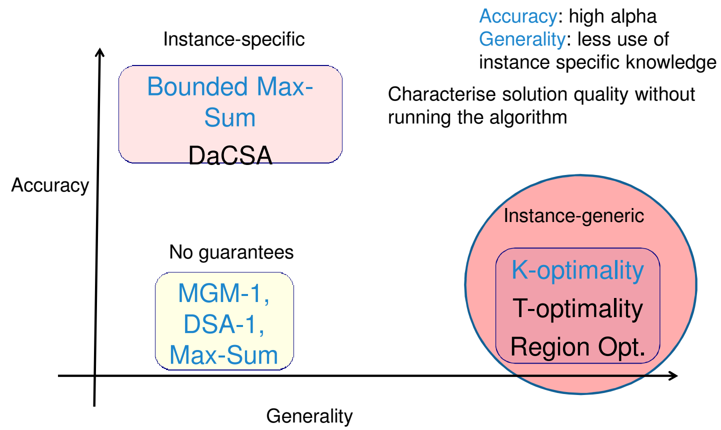
Quality guarantees for approx. techniques

- Key area of research
- Address trade-off between guarantees and computational effort
- Particularly important for many real world applications
  - Critical (e.g. Search and rescue)
  - Constrained resource (e.g. Embedded devices)
  - Dynamic settings

Terminology and notation

- Assume a maximization problem
- X^* optimal solution, $\tilde{X}$ a solution
- $F(\tilde{X}) \geq \alpha F(X^*)$
- α percentage of optimality
  - [0,1]
  - The higher the better
- $\rho = \frac{1}{\alpha}$ approximation ratio
  - >= 1
  - The lower the better
- $\rho F(\tilde{X})$ is the bound

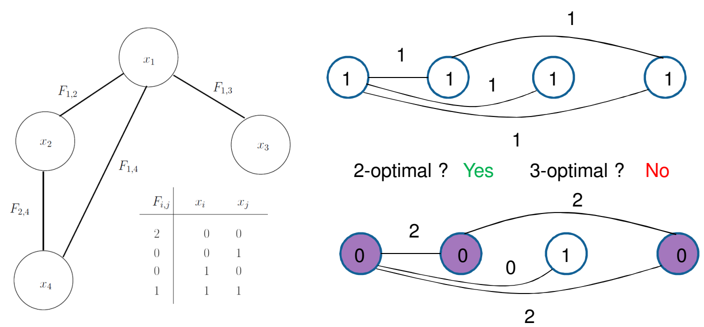
Instance-generic guarantees



K-Optimality framework

- Given a characterization of solution gives bound on solution quality [Pearce and Tambe 07]
- Characterization of solution: k-optimal
- K-optimal solution:
  - Corresponding value of the objective function can not be improved by changing the assignment of k or less variables.

K-Optimal solutions



$F_{i,j}$	x_i	x_j
2	0	0
0	0	1
0	1	0
1	1	1

Bounds for K-Optimality

For any DCOP with non-negative rewards [Pearce and Tambe 07]

Number of agents

Maximum arity of constraints

$$F(\tilde{X}) \geq \frac{\binom{n-m}{k-m}}{\binom{n}{k} - \binom{n-m}{k}} F(X^*)$$

K-optimal solution

α

Binary Network (m=2):

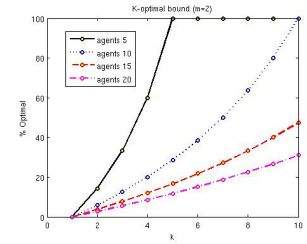
$$F(\tilde{X}) \geq \frac{k-1}{2n-k-1} F(X^*)$$

K-Optimality Discussion

- Need algorithms for computing k-optimal solutions
  - DSA-1, MGM-1 k=1; DSA-2, MGM-2 k=2 [Maheswaran et al. 04]
  - DALO for generic k (and t-optimality) [Kiekintveld et al. 10]
- The higher k the more complex the computation (exponential)

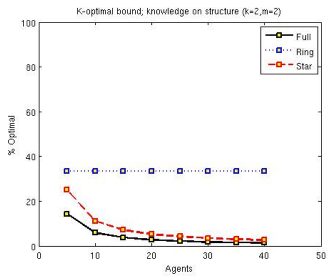
Percentage of Optimal:

- The higher k the better
- The higher the number of agents the worst



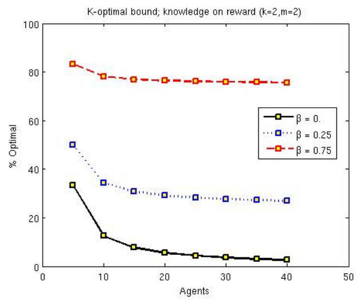
Trade-off between generality and solution quality

- K-optimality based on worst case analysis
- assuming more knowledge gives much better bounds
- Knowledge on structure [Pearce and Tambe 07]

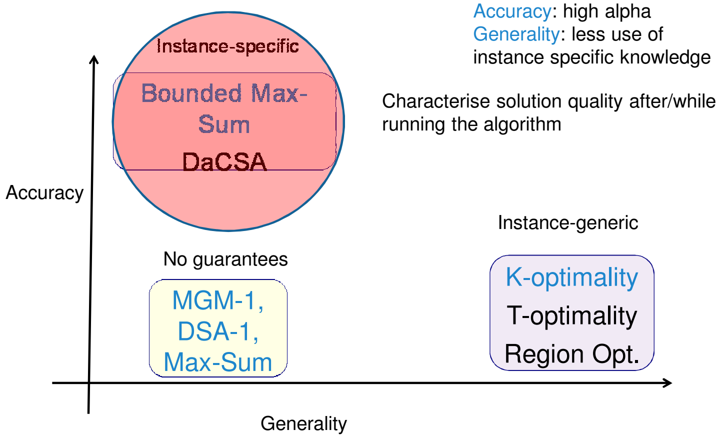


Trade-off between generality and solution quality

- Knowledge on reward [Bowring et al. 08]
- Beta: ratio of least minimum reward to the maximum

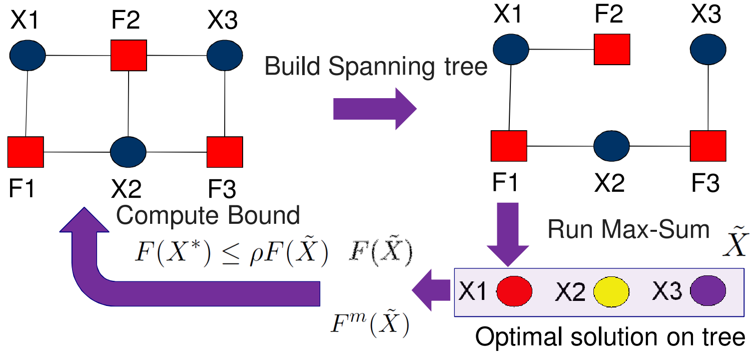


Instance-specific guarantees



Bounded Max-Sum

Aim: Remove cycles from Factor Graph avoiding exponential computation/communication (e.g. no junction tree)
Key Idea: solve a relaxed problem instance [Rogers et al.11]



$\tilde{X}$

$F(X^*) \le \rho F(\tilde{X})$ $F(\tilde{X})$

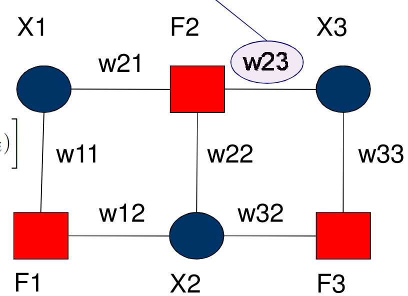
$F^m(\tilde{X})$

Factor Graph Annotation

- Compute a weight for each edge
  - maximum possible impact of the variable on the function

$$w_{ij} = \max_{\mathbf{x}_i \setminus j} \left[\max_{x_j} F_i(\mathbf{x}_i) - \min_{x_j} F_i(\mathbf{x}_i) \right]$$

$$w_{23} = \max_{x_1,x_2} \left[\max_{x_3} F_2(x_1,x_2,x_3) - \min_{x_3} F_2(x_1,x_2,x_3) \right]$$

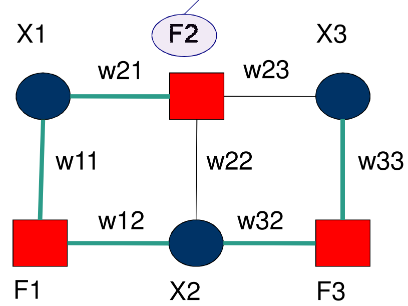


Factor Graph Modification

- Build a Maximum Spanning Tree
  - Keep higher weights
- Cut remaining dependencies
  - Compute $W = \sum_{e_{ij} \in C} w_{ij}$
- Modify functions $\min_{\mathbf{x}_i^c} F_i(\mathbf{x}_i)$
- Compute bound

$$F(\mathbf{x}^*) \le \rho F(\tilde{\mathbf{x}}) = F^m(\tilde{\mathbf{x}}) + W$$

$$F'_2(x_1) = \min_{x_2,x_3} F_2(x_1,x_2,x_3)$$



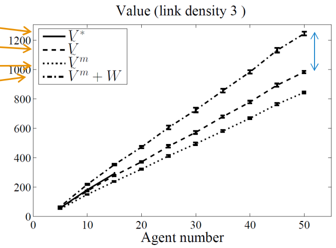
W = w22 + w23

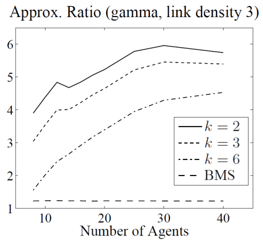
Results: Random Binary Network

Bound is significant

– Approx. ratio is typically 1.23 (81 %)

Optimal
Approx.
Lower Bound
Upper Bound





Comparison with k-optimal with knowledge on reward structure

Much more accurate less general

Discussion

- Discussion with other data-dependent techniques
  - BnB-ADOPT [Yeoh et al 09]
    - Fix an error bound and execute until the error bound is met
    - Worst case computation remains exponential
  - ADPOP [Petcu and Faltings 05b]
    - Can fix message size (and thus computation) or error bound and leave the other parameter free
- Divide and coordinate [Vinyals et al 10]
  - Divide problems among agents and negotiate agreement by exchanging utility
  - Provides anytime quality guarantees

Summary

- Approximation techniques crucial for practical applications: surveillance, rescue, etc.
- DSA, MGM, Max-Sum heuristic approaches
  - Low coordination overhead, acceptable performance
  - No guarantees (convergence, solution quality)
- Instance generic guarantees:
  - K-optimality framework
  - Loose bounds for large scale systems
- Instance specific guarantees
  - Bounded max-sum, ADPOP, BnB-ADOPT
  - Performance depend on specific instance

References I

DOCPs for MRS

- [Delle Fave et al 12] A methodology for deploying the max-sum algorithm and a case study on unmanned aerial vehicles. In, IAAI 2012
- [Taylor et al. 11] Distributed On-line Multi-Agent Optimization Under Uncertainty: Balancing Exploration and Exploitation, Advances in Complex Systems

MGM

- [Maheswaran et al. 04] Distributed Algorithms for DCOP: A Graphical Game-Based Approach, PDCS-2004

DSA

- [Fitzpatrick and Meertens 03] Distributed Coordination through Anarchic Optimization, Distributed Sensor Networks: a multiagent perspective.
- [Zhang et al. 03] A Comparative Study of Distributed Constraint algorithms, Distributed Sensor Networks: a multiagent perspective.

Max-Sum

- [Stranders at al 09] Decentralised Coordination of Mobile Sensors Using the Max-Sum Algorithm, AAAI 09
- [Rogers et al. 10] Self-organising Sensors for Wide Area Surveillance Using the Max-sum Algorithm, LNCS 6090 Self-Organizing Architectures
- [Farinelli et al. 08] Decentralised coordination of low-power embedded devices using the max-sum algorithm, AAMAS 08

References II

Instance-based Approximation

- [Yeoh et al. 09] Trading off solution quality for faster computation in DCOP search algorithms, IJCAI 09
- [Petcu and Faltings 05b] A-DPOP: Approximations in Distributed Optimization, CP 2005
- [Rogers et al. 11] Bounded approximate decentralised coordination via the max-sum algorithm, Artificial Intelligence 2011.

Instance-generic Approximation

- [Vinyals et al 10b] Worst-case bounds on the quality of max-product fixed-points, NIPS 10
- [Vinyals et al 11] Quality guarantees for region optimal algorithms, AAMAS 11
- [Pearce and Tambe 07] Quality Guarantees on k-Optimal Solutions for Distributed Constraint Optimization Problems, IJCAI 07
- [Bowring et al. 08] On K-Optimal Distributed Constraint Optimization Algorithms: New Bounds and Algorithms, AAMAS 08
- [Weiss 00] Correctness of local probability propagation in graphical models with loops, Neural Computation
- [Kiekintveld et al. 10] Asynchronous Algorithms for Approximate Distributed Constraint Optimization with Quality Bounds, AAMAS 10