Optimizing Streaming Applications with Self-Interested Users using M-DPOP

Boi Faltings

Adrian Petcu

Artificial Intelligence Laboratory EPFL, Switzerland

David Parkes

Jeff Shneidman

DEAS

Harvard University, USA

http://liawww.epfl.ch/Publications/Archive/Faltings2006.pdf





Overview

- Operator placement problems:
 - +description
 - model as Combinatorial Optimization (MCOP)
- M-DPOP: distributed algorithm for social choice
- Experimental evaluation
- M-DPOP extras, wrap up



Optimal Operator Placement Problem

Setting:

- data producers
- servers = computational resources







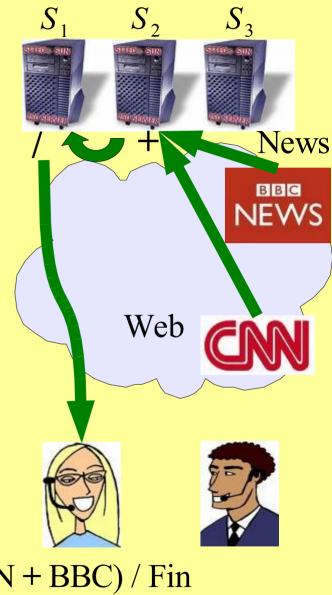


Optimal Operator Placement Problem

Setting:

- + data producers

- users have queries:
 - ▶"EU financial news" = operators on data



(CNN + BBC) / Fin

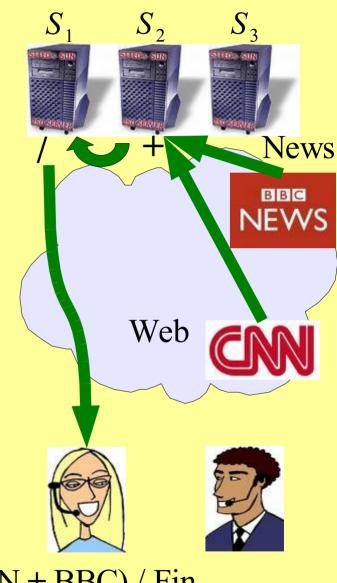




Optimal Operator Placement Problem

Setting:

- + data producers
- \$\phi\$ servers = computational resources
- users have queries:
 - ▶"EU financial news" = operators on data
 - ▶an operator could run on many servers
 - ►user has preferences on placements
- Resource allocation problem:
 - Goal: find the best placement
 - NP-hard



(CNN + BBC) / Fin





Issues in this setting

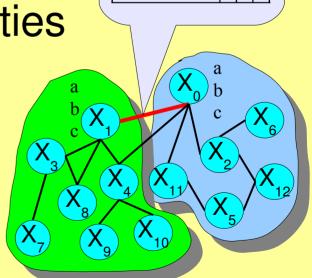
- Distribution: constraints, preferences
- Model: Multiagent Constraint Optimization Problems (MCOP)
- Solving algorithm (M-DPOP [Petcu, Faltings, Parkes '06])
 - distributed algorithm
 - implements VCG for incentive compatibility





Multi-agent Constraint Optimization (MCOP)

- Constraint Optimization Problems (COP):
 - Variables: decisions to take
 - Values: possible alternatives
 - Constraints: force/forbid combinations
 - Relations: utility functions
- Objective: maximize sum of utilities
- MCOP COP split among agents
- NP-hard



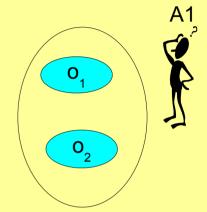
R(X0,X1)



MCOP: Operators-as-variables

Agents want to place operators:

$$\bigoplus$$
 e.g. A_1 : O_1 , O_2



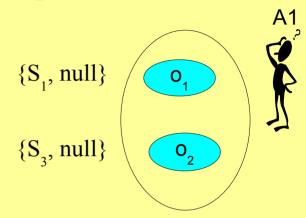
•Variables = where to place operators

MCOP: Operators-as-variables

Agents want to place operators:

$$\bigoplus$$
 e.g. A_1 : O_1 , O_2

- Servers can execute operators, e.g.
 - ϕ_1 could run on S_1 ; o_2 on S_3



- •Variables = where to place operators
- •Values = hosting servers (+ null)



MCOP: Operators-as-variables

Agents want to place operators:

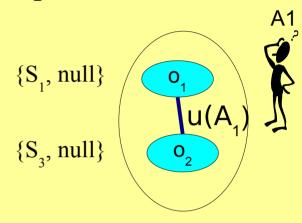
$$\bigoplus$$
 e.g. A_1 : O_1 , O_2

Servers can execute operators, e.g.

$$\phi_1$$
 could run on S_1 ; o_2 on S_3

😝 Utilities – e.g. u(A₁):

u(A1)	(A1) o1		
		null	S1
02	null	0	2
02	S3	3	6



- •Variables = where to place operators
- •Values = hosting servers (+ null)
- •Blue relations = preferences

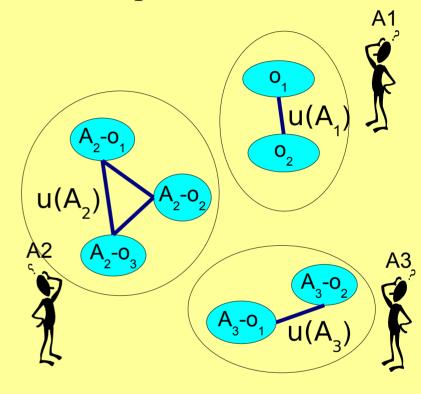


Agents want to place operators:

$$\bigoplus$$
 e.g. A_1 : O_1 , O_2

- Servers can execute operators, e.g.
 - \bigoplus o₁ could run on S₁; o₂ on S₃
- Each agent has its local problem

MCOP: Operators-as-variables



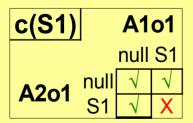
- •Variables = where to place operators
- •Values = hosting servers (+ null)
- •Blue relations = preferences



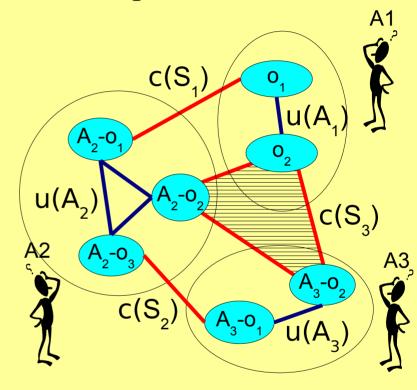
Agents want to place operators:

$$\bigoplus$$
 e.g. A_1 : O_1 , O_2

- Servers can execute operators, e.g.
 - \bigoplus o₁ could run on S₁; o₂ on S₃
- Each agent has its local problem
- Capacity constraints restrict allocations
 e.g. S1 can run a single service:



MCOP: Operators-as-variables



- •Variables = where to place operators
- •Values = hosting servers (+ null)
- •Blue relations = preferences
- Capacity constraints



Issues in this setting

- Distribution: constraints, preferences
- Model: Multiagent Constraint Optimization Problems (MCOP)
- Solving algorithm (M-DPOP [Petcu, Faltings, Parkes '06])
 - distributed algorithm
 - implements VCG for incentive compatibility



The VCG mechanism for incentive compatibility

- Centralized VCG (efficient, IC, IR social choice):
 - Center solves:
 - main economy MCOP(All)
 - ▶all marginal economies MCOP(-i)
 - Ai pays tax = its marginal effect on others
- Distributed VCG challenging, because agents do:
 - a)information revelation (also in centralized)
 - b)message passing
 - c)computation
 - d)report marginal effects of others



M-DPOP: the first distributed VCG mechanism

[Petcu, Faltings, Parkes '06]

- Distributed implementation of the VCG mechanism:
 - Only central entity required is the bank
 - •all agents solve the main economy (using DPOP)
 - For each agent Ai:

FÉDÉRALE DE LAUSANNE

- ▶the other agents solve MCOP(-i) (using DPOP)
- ▶the **other** agents report Ai's marginal effects
- Bank fines Ai with sum of reports
- Mechanism is faithful: truthfulness is ex-post Nash

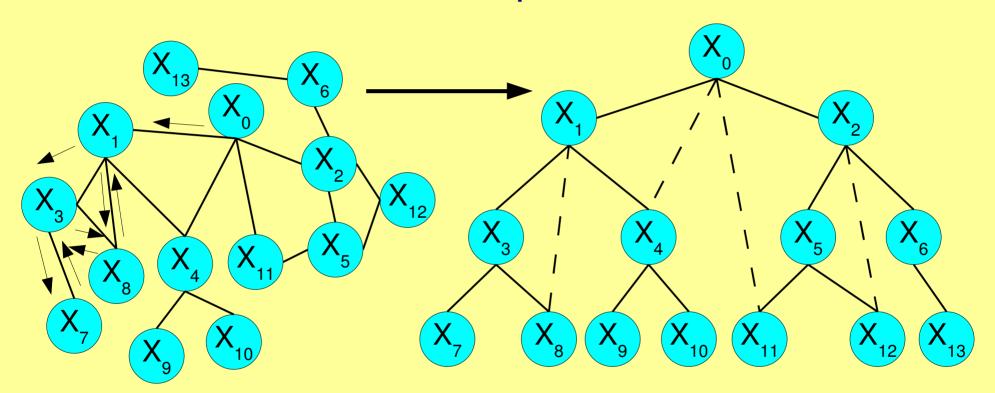


[Petcu&Faltings @ IJCAI'05]

© Complete algorithm, linear # of messages:

[Petcu&Faltings @ IJCAI'05]

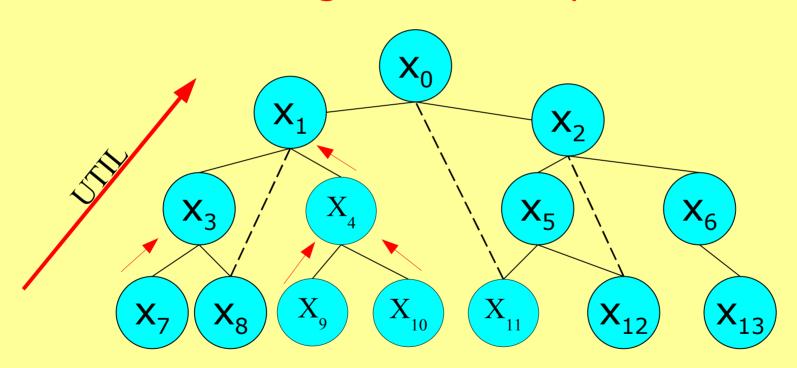
- © Complete algorithm, linear # of messages:
 - 1: DFS: distributed depth first traversal





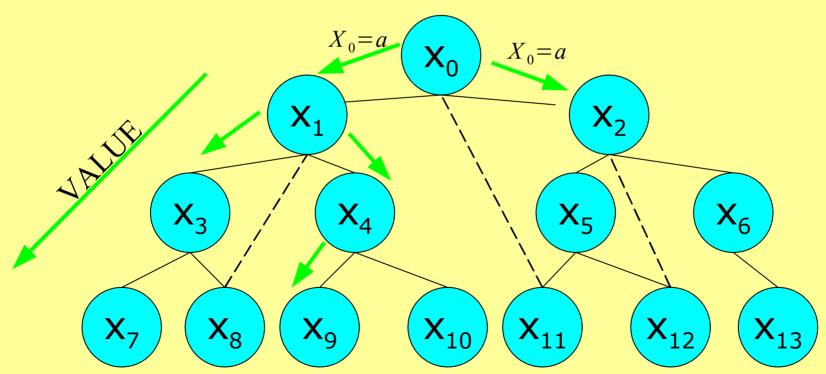
[Petcu&Faltings @ IJCAI'05]

- © Complete algorithm, linear # of messages:
 - 1: DFS: distributed depth first traversal
 - ◆2: **UTIL** messages bottom-up on the DFS



[Petcu&Faltings @ IJCAI'05]

- © Complete algorithm, linear # of messages:
 - 1: DFS: distributed depth first traversal
 - ◆2: UTIL messages bottom-up on the DFS
 - ⊕3: VALUE assignments go top-down





more difficult

Experimental results on random problems

- Model of [Faltings et al. 2006]:
 - several classes of users and servers;
 - users from the same class issue similar queries;
 - servers from the same class can handle the same queries
- Scales well when little overlap between user/server classes

	Agents	Srvs	Vars	Constr	Msgs	MaxMsgSize	TotalMsgSize
	20	10	32	42	31	94.62 KB	393.94 KB
	40	19	60	78	59	114.5 KB	377.25 KB
	60	29	92	117	91	296 KB	447.78 KB
	80	40	128	171	127	1.19 MB	2.18 MB
	100	49	156	210	155	1.07 MB	2.82 MB
,	120	59	188	255	187	19.1 MB	32.32 MB





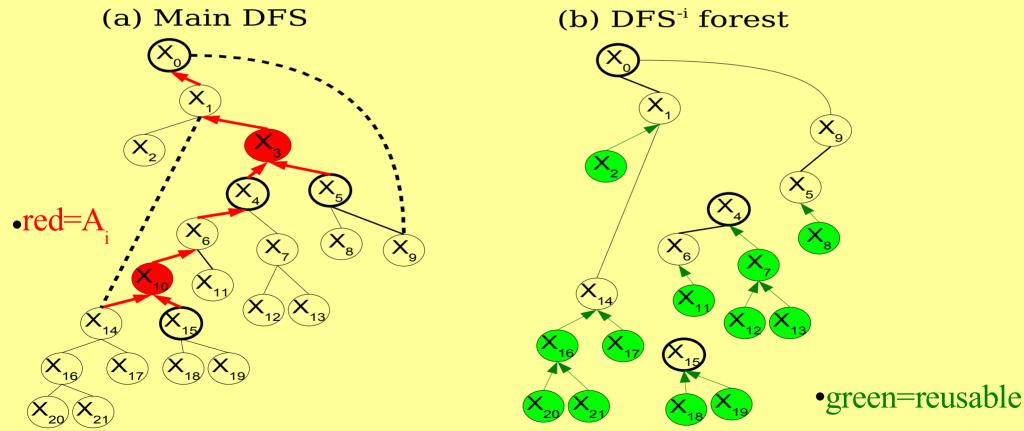
M-DPOP extras

- First faithful distributed VCG implementation
- Reuse computation from the main to the marginal economies faithfully
- Weak budget balance via limited redistribution of taxes
- Several techniques to increase its efficiency



M-DPOP: reusing computation from main econ.

- reuses computation from MCOP(All) to MCOP(-i):
 - adapt DFS(-i) from DFS(All)
 - reuse only messages not influenced by Ai



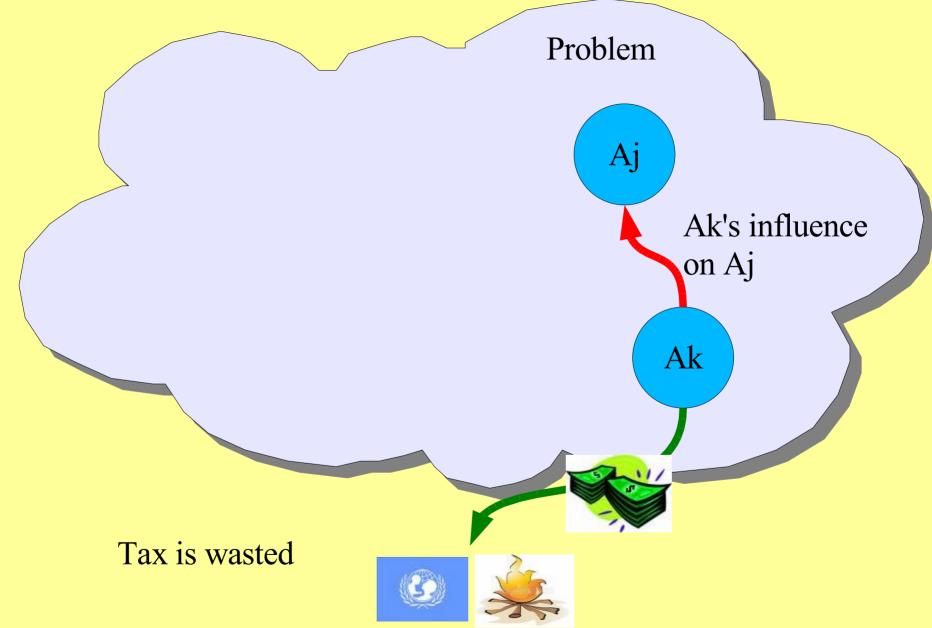


M-DPOP extras

- First faithful distributed VCG implementation
- Reuse computation from the main to the marginal economies faithfully
- Weak budget balance via limited redistribution of taxes
- Several techniques to increase its efficiency

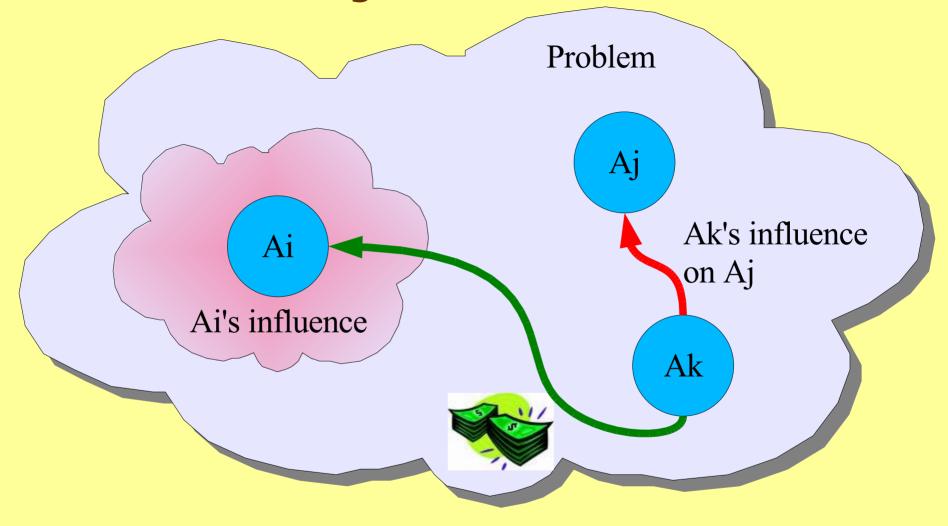


Ext1: Towards budget balance with M-DPOP





Ext1: Towards budget balance with M-DPOP



Ai can receive Aj's tax iff it cannot influence it!! (weak budget-balance)



M-DPOP extras

- First faithful distributed VCG implementation
- Reuse computation from the main to the marginal economies faithfully
- Weak budget balance via limited redistribution of taxes
- Several techniques to increase its efficiency



Ext2: Techniques to increase efficiency

- H-DPOP: reduces message size & memory by using Constraint Decision Diagrams [Kumar, Petcu & Faltings, DCR'07]
- MB-DPOP: memory bounded propagations using cycle-cuts [Petcu & Faltings, IJCAI'07]
- PC-DPOP: limited message size using partial centralization [Petcu, Faltings & Mailler, IJCAI'07]



Wrap up: our contributions

- MCOP model for operator placement
- M-DPOP for efficient social choice:
 - First faithful distributed VCG implementation
 - Reuse computation from the main to the marginal economies faithfully
 - •Weak budget balance via limited redistribution of taxes
 - Several techniques to increase efficiency

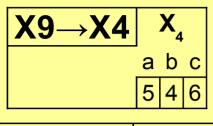


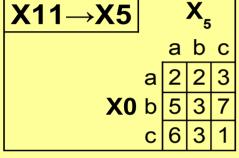
References and related work

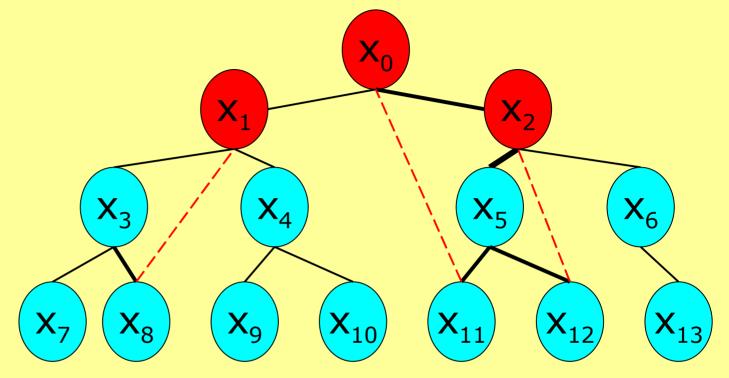
- Feigenbaum and Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pages 1–13. ACM Press, New York, 2002
- Yokoo, Durfee, Ishida, and Kuwabara. The distributed constraint satisfaction problem formalization and algorithms. IEEE Transactions on Knowledge and Data Engineering, 10(5):673–685, 1998.
- Nisan and Ronen. Computationally feasible VCG mechanisms. In ACM Conference on Electronic Commerce, pages 242–252, 2000
- Parkes and Shneidman. Distributed implementations of Vickrey-Clarke-Groves mechanisms. In Proceedings of AAMAS-04, pages 261–268, New York, USA, 2004
- Petcu, Faltings, Parkes. *M-DPOP: faithful implementation of efficient social choice problems.* In Proceedings of AAMAS'06, Hakodate, Japan.



DPOP - complexity is exponential in induced width







- •Linear # of messages! largest one is exponential in induced width
- Each back edge has its associated tree path
- Overlaps between tree paths increase complexity
- Space complexity = maximum no of overlaps = induced width

