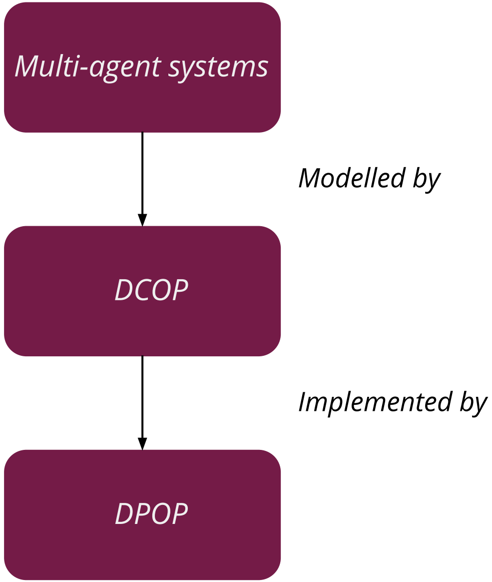
Distributed Constraint Optimization Problems

DPOP algorithm

Matthaios Zidianakis
Vasileios Papadopoulos
Dimitris Delikonstantis

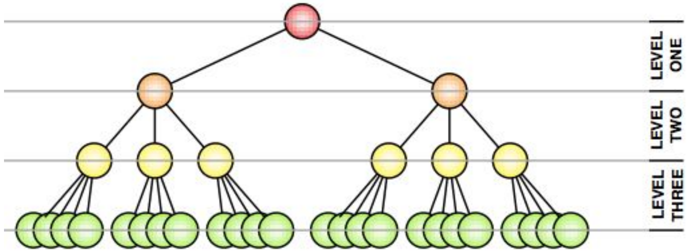
Distributed Constraint Optimization Problems

- Multi-agent systems
  - Communication
  - Cooperation
  - Compete
- DCOP framework model P = (A, X, D, F, α)
  - A = Agents
  - X = Variables
  - D = Domains
  - F = Cost functions/Utility/Rewards
  - α = mapping function from variables to agents
- DPOP algorithm (X, D, R)
  - Tree-shaped transformation



Problem Definition

- Meeting scheduling optimization
- <X, D, R> ⟶ series of tuples <A, M, U>
- A = agents, M = meetings, U = meeting utility
- Constraints
  - Equality(inter-agent constraint)
  - Inequality(intra-agent constraint)
- Preference
- Utility function
  - $\text{Preference}_{\text{timeslot}} * \text{Meeting}_{i}\text{\_Util}$
- Goal: find an assignment that maximizes the total utility, social welfare.

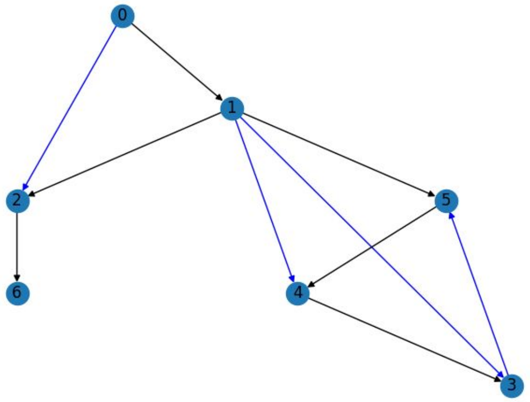


Thanks to **Andreas Oikonomakis**, generator.py

Constraint Graph to Pseudo-tree

Conventions

- Nodes represent agents
  - Inequality messages are virtual
  - Privacy
- DFS traversal
  - Start from any node and put to visited list
  - move to adjacent unvisited node and put it to visited
  - continue loop until there is no unvisited node.
  - Backtrack and check for other unvisited nodes
  - **Keep track of backedges**
- Edges represent equality constraint
  - Constraints within the same agent not shown(**inequality**)
  - Back-edges represented by **blue** color

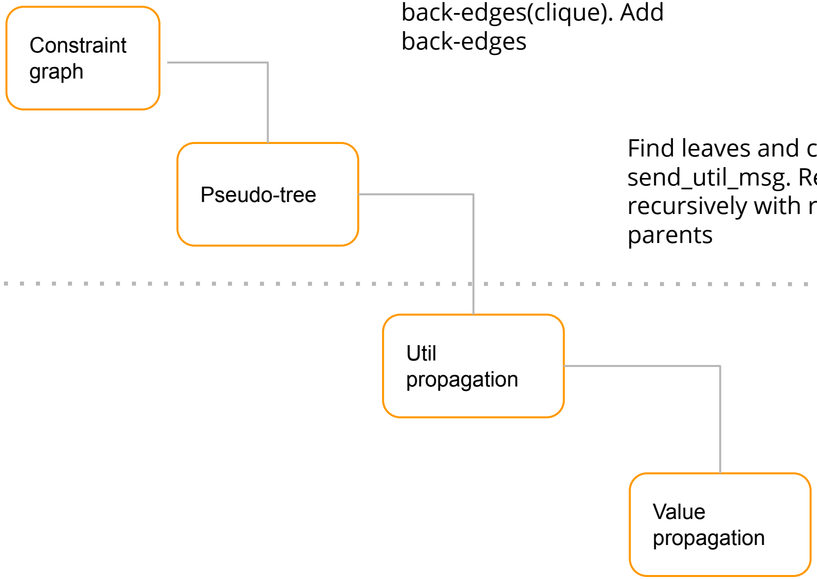


Pseudo-tree output, visualization hiccups

Impl. DPOP

high overview

Input_fille ->



DFS traversal. Keep track of back-edges(clique). Add back-edges

Find leaves and call send_util_msg. Repeat recursively with respective parents

Tree-shaped transformation

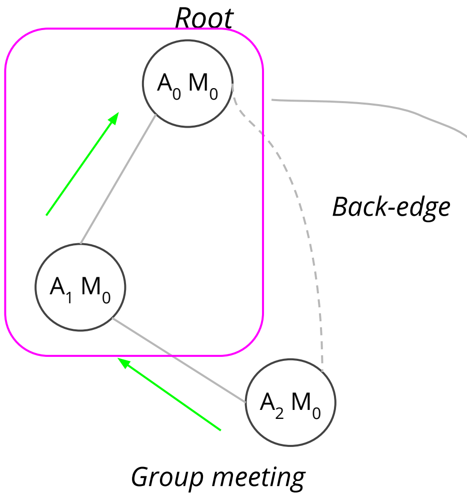
Reverse util_prop order and start value prop

Relations Representation

Equality Constraint

R(A2, A1) Meeting 0			
	A1=8	A1=9	A1=10
A2=8	100	-inf	-inf
A2=9	-inf	200	-inf
A2=10	-inf	-inf	500

R(A2, A0) Meeting 0			
	A0=8	A0=9	A0=10
A2=8	200	-inf	-inf
A2=9	-inf	500	-inf
A2=10	-inf	-inf	100



Inequality Constraint

R(m0, m1) Agent 2			
	A0=8	A0=9	A0=10
A2=8	-inf	230	210
A2=9	230	-inf	100
A2=10	210	100	-inf

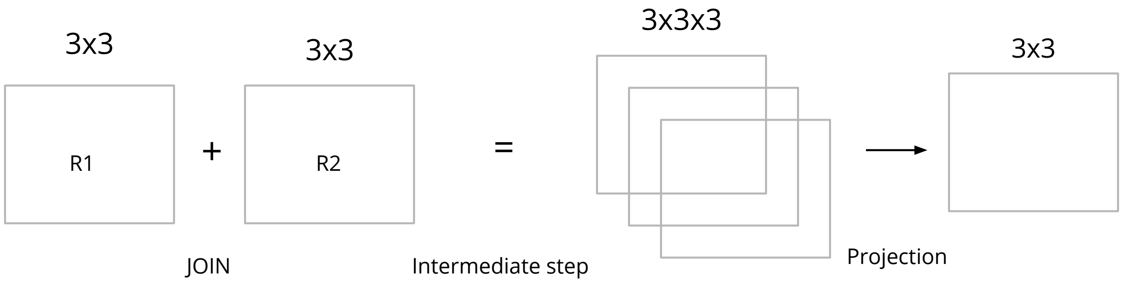
A2_TS(8) * UtilM(0,2) +
A1_TS(8) * UtilM(1,2)

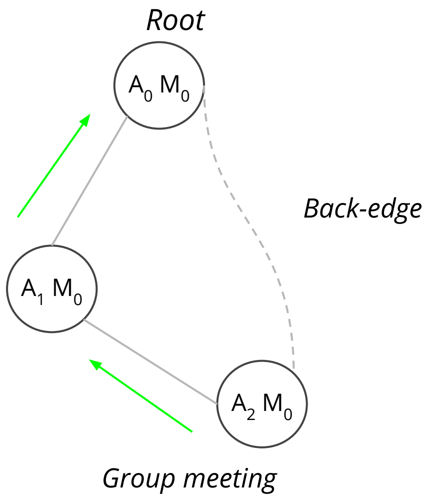
Implementation

```
relation->key:(1, 0, 0)
 [[1100.  -inf  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf 2200.  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf 3300.  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf 4400.  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf 5500.  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf 6600.  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf 7700.  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf  -inf 8800.]]
```

Utility Message Overview

2 Relations





Utility Propagation 1/3

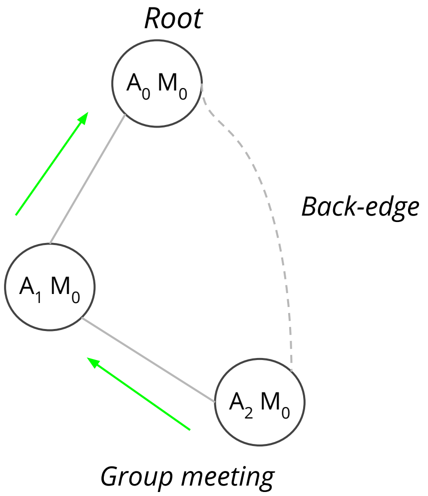
Current node: A2

R(A2, A1) Meeting 0			
	A1=8	A1=9	A1=10
A2=8	100	-inf	-inf
A2=9	-inf	200	-inf
A2=10	-inf	-inf	500

+

R(A2, A0) Meeting 0			
	A0=8	A0=9	A0=10
A2=8	200	-inf	-inf
A2=9	-inf	500	-inf
A2=10	-inf	-inf	100

A2 = 8			
	A0=8	A0=9	A0=10
A1=8	300	-inf	-inf
A1=9	-inf	-inf	-inf
A1=10	-inf	-inf	-inf



Implementation

```
relation->key:(1, 0, 0)
 [[1100.  -inf  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf 2200.  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf 3300.  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf 4400.  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf 5500.  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf 6600.  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf 7700.  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf  -inf 8800.]]
```

Utility Propagation 2/3

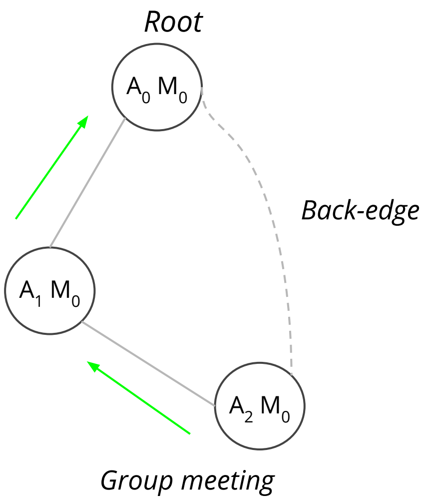
Current node: A2

R(A2, A1) Meeting 0			
	A1=8	A1=9	A1=10
A2=8	100	-inf	-inf
A2=9	-inf	200	-inf
A2=10	-inf	-inf	500

+

R(A2, A0) Meeting 0			
	A0=8	A0=9	A0=10
A2=8	200	-inf	-inf
A2=9	-inf	500	-inf
A2=10	-inf	-inf	100

A2 = 9			
	A0=8	A0=9	A0=10
A1=8	-inf	-inf	-inf
A1=9	-inf	700	-inf
A1=10	-inf	-inf	-inf



Implementation

```
relation->key:(1, 0, 0)
 [[1100.  -inf  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf 2200.  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf 3300.  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf 4400.  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf 5500.  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf 6600.  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf 7700.  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf  -inf 8800.]]
```

Utility Propagation 3/3

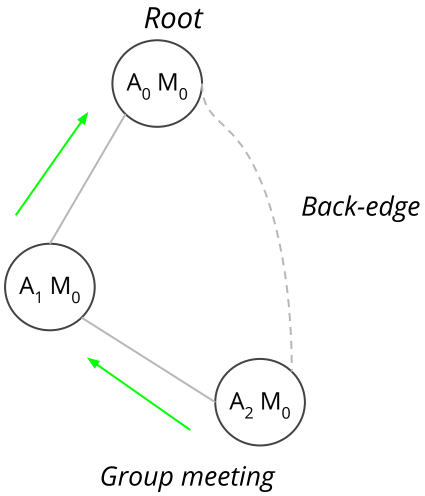
Current node: A2

R(A2, A1) Meeting 0			
	A1=8	A1=9	A1=10
A2=8	100	-inf	-inf
A2=9	-inf	200	-inf
A2=10	-inf	-inf	500

+

R(A2, A0) Meeting 0			
	A0=8	A0=9	A0=10
A2=8	200	-inf	-inf
A2=9	-inf	500	-inf
A2=10	-inf	-inf	100

A2 = 10			
	A0=8	A0=9	A0=10
A1=8	-inf	-inf	-inf
A1=9	-inf	-inf	-inf
A1=10	-inf	-inf	600



Implementation

```
relation->key:(1, 0, 0)
 [[1100.  -inf  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf 2200.  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf 3300.  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf 4400.  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf 5500.  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf 6600.  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf 7700.  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf  -inf 8800.]]
```

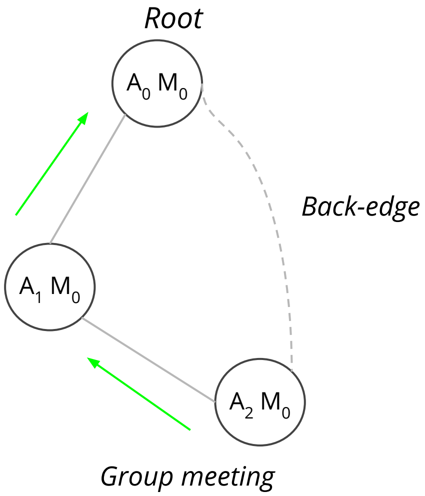
Utility Propagation

Current node: A2
Projection

A2 = 8			
	A0=8	A0=9	A0=10
A1=8	300	-inf	-inf
A1=9	-inf	-inf	-inf
A1=10	-inf	-inf	-inf

A2 = 9			
	A0=8	A0=9	A0=10
A1 = 8	-inf	-inf	-inf
A1=9	-inf	700	-inf
A1=10	-inf	-inf	-inf

A2 = 10			
	A0=8	A0=9	A0=10
A1=8	-inf	-inf	-inf
A1=9	-inf	-inf	-inf
A1=10	-inf	-inf	600



Projection A2_M0			
	A0=8	A0=9	A0=10
A1=8	300	-inf	-inf
A1=9	-inf	700	-inf
A1=10	-inf	-inf	600

Send message to A1 from A2

Implementation

```
relation->key:(1, 0, 0)
 [[1100.  -inf  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf 2200.  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf 3300.  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf 4400.  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf 5500.  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf 6600.  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf 7700.  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf  -inf 8800.]]
```

Utility Propagation

Current node: A1

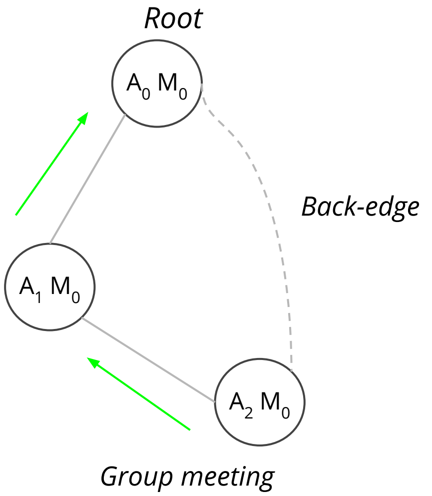
Projection A2_M0			
	A0=8	A0=9	A0=10
A1=8	300	-inf	-inf
A1=9	-inf	700	-inf
A1=10	-inf	-inf	600

Received msg

+

R(A1, A0) Meeting 0			
	A0=8	A0=9	A0=10
A1=8	200	-inf	-inf
A1=9	-inf	500	-inf
A=10	-inf	-inf	100

Relation



Projection A1_M0		
A0=8	A0=9	A0=10
500	1200	700

Send message to A0 from A1

Implementation

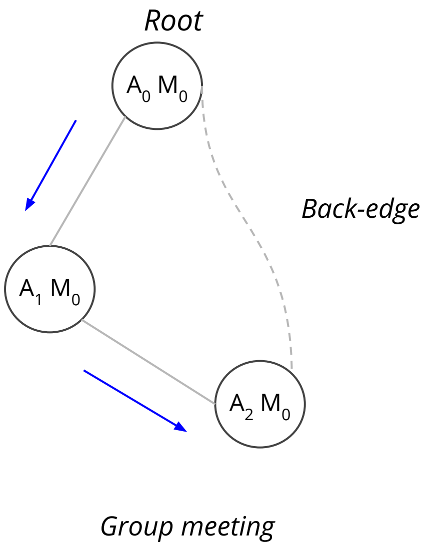
```
relation->key:(1, 0, 0)
 [[1100.  -inf  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf 2200.  -inf  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf 3300.  -inf  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf 4400.  -inf  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf 5500.  -inf  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf 6600.  -inf  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf 7700.  -inf]
 [ -inf  -inf  -inf  -inf  -inf  -inf  -inf 8800.]]
```

Value Propagation

Current node: A0

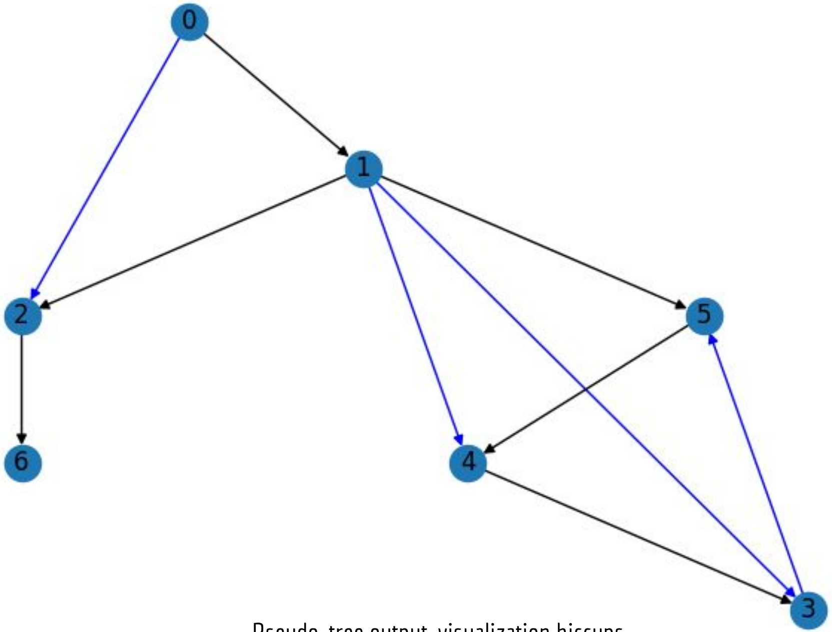
Projection A1_M0		
A0=8	A0=9	A0=10
500	1200	700

Chooses max and initiates value prop



Utility Order - Evaluation Impl.

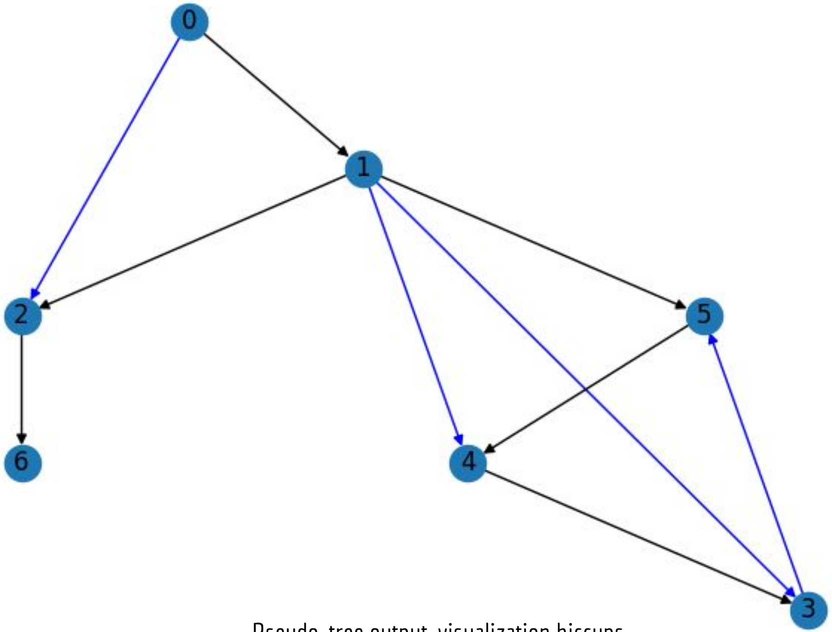
```
Leaves are: [6, 3]
Util Message from: 6 to 2
Util Message from: 3 to 4
Util Message from: 4 to 5
Util Message from: 5 to 1
Util Message from: 2 to 1
Util Message from: 1 to 0
root node:  {}
```



Pseudo-tree output, visualization hiccups

Value Order - Evaluation Impl.

```
Value Message from: 0 to 1
Value Message from: 1 to 2
Value Message from: 1 to 5
Value Message from: 5 to 4
Value Message from: 4 to 3
Value Message from: 2 to 6
```



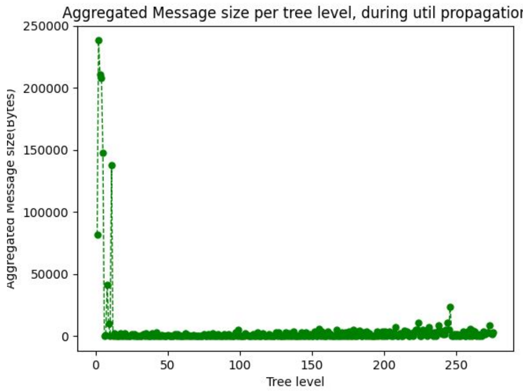
Pseudo-tree output, visualization hiccups

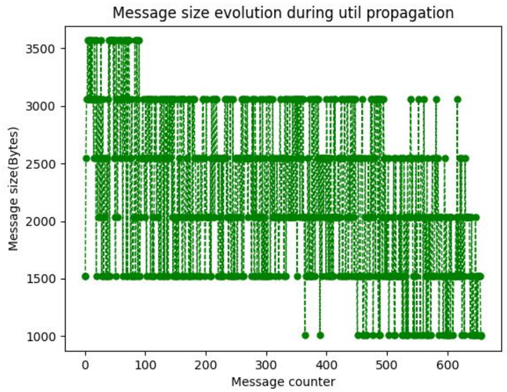
DPOP Evaluation - During Util Prop

Agents: 700
Meetings: 350

Execution

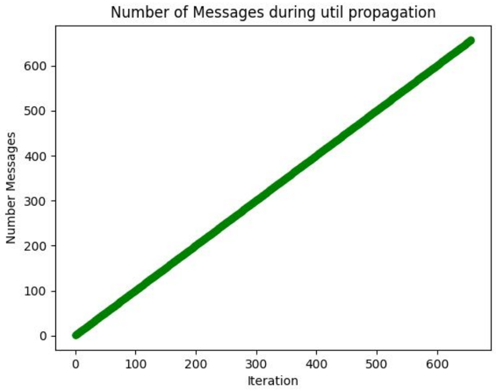
- Aggregated message size per cycle.
- Message size evolution.
- Rate of change of number of messages.





Inequality const calculation per agent

m in {m1,m2,m3,m4,m5}
m1 != (m2, m3, m4, m5) +
m2 != (m3, m4, m5) +
m3 != (m4, m5) +
m4 != (m5)



output

```
Number of agents:700
Number of meetings:350
Number of variables:1661
Total Constraints 3714
        Equality constraints 2200
        Inequality constraints 1514
Total number of messages:1314
Max message size:3568
Cycles:552
```

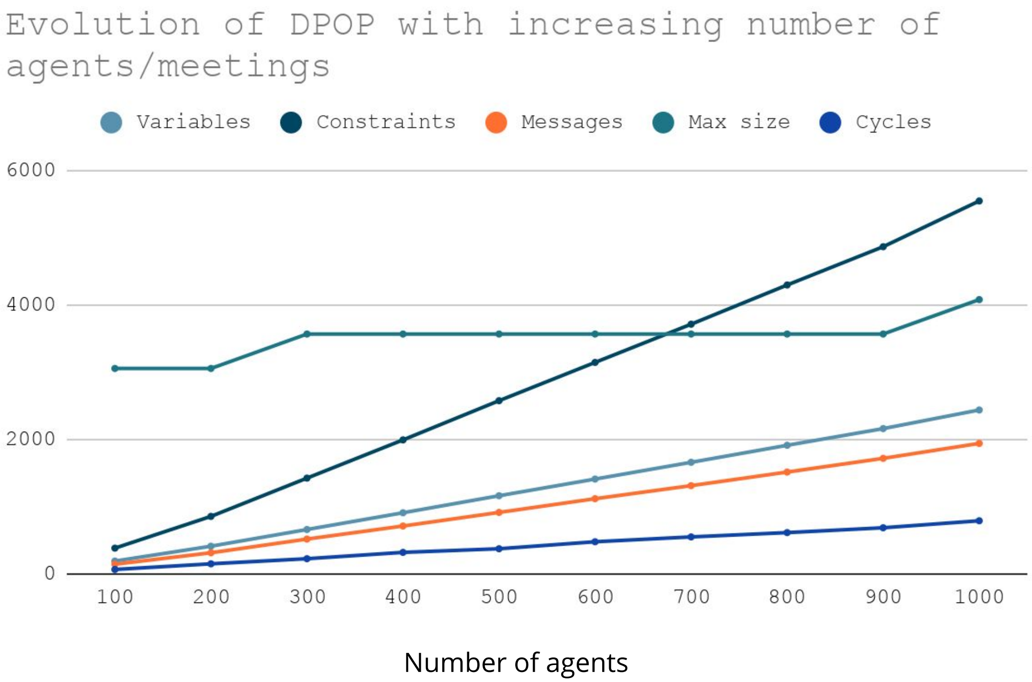
Results 1/2

Problems										
Agents	100	200	300	400	500	600	700	800	900	1000
Meetings	50	100	150	200	250	300	350	400	450	500
Variables	193	414	662	911	1164	1412	1661	1914	2162	2439
Constraints	385	857	1426	1994	2577	3146	3714	4297	4866	5546
Messages	146	316	520	714	916	1120	1314	1516	1720	1942
Max message size	3056	3056	3568	3568	3568	3568	3568	3568	3568	4080
Cycles	68	152	228	322	376	480	552	616	688	792

Results 2/2

Evaluation

- As expected all metrics has a linear relationship with size of the problem(agents,meetings).
- Max message size remains constant which is **incorrect** as our implementation lacks of proper hypercube calculation.
- Maybe use of *pandas.dataframe object*



References

1. Andrian Petcu, Boi Faltings, DPOP: A Scalable Method for Multiagent Constraint Optimization, IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005
2. Ferdinando Fioretto, Enrico Pontelli, William Yeoh, Distributed Constraint Optimization Problems and Applications: A Survey, Journal of Artificial Intelligence Research 61 (2018) 623-698
3. Rajiv T. Maheswaran, Milind Tambe, Emma Bowring, Jonathan P. Pearce, and Pradeep Varakantham, Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling, 2004
4. Andrian Petcu, A class of algorithms for distributed constraint optimization, Frontiers in Artificial Intelligence and Applications, 2009