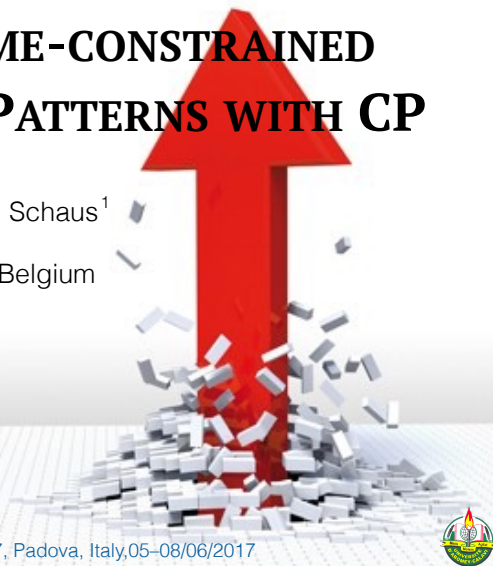


MINING TIME-CONSTRAINED SEQUENTIAL PATTERNS WITH CP

J. AOGA¹, T. Guns², P. Schaus¹

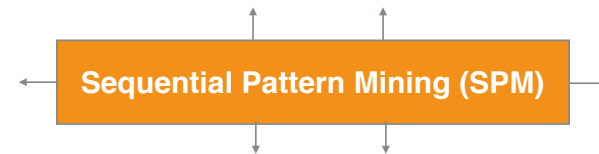
¹UCLouvain, ²VUB — Belgium



CPAIOR 2017, Padova, Italy, 05-08/06/2017



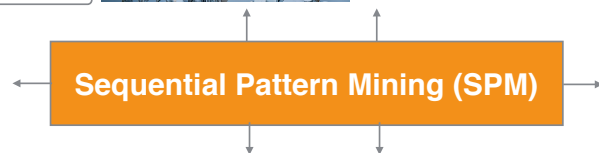
GENERAL OVERVIEW



GENERAL OVERVIEW

Telecom

- Network analysis
- People behavior
- Inter-Connection



GENERAL OVERVIEW

Telecom

- Network analysis
- People behavior
- Inter-Connection

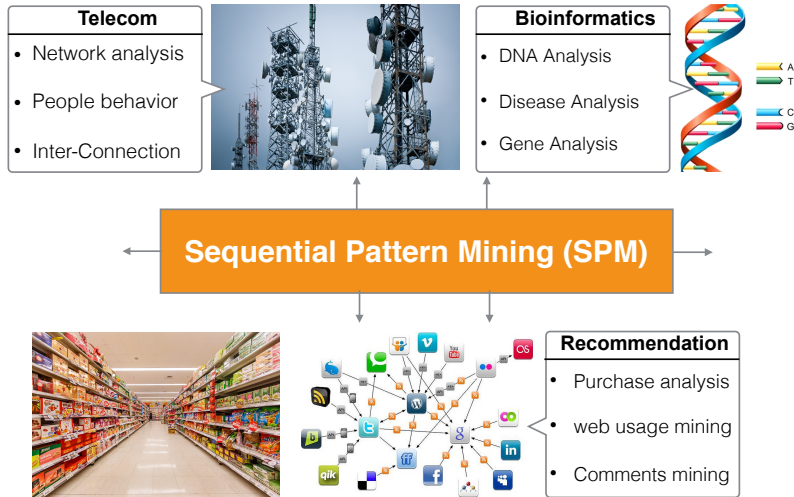


Bioinformatics

- DNA Analysis
- Disease Analysis
- Gene Analysis



GENERAL OVERVIEW



SPM PROBLEM

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

Sequence Database (SDB)



SPM PROBLEM

Sequence

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

Sequence Database (SDB)

- Sequence : < Milk Coffee Sugar Coffee Sugar >

SPM PROBLEM

Sub-sequence

Sequence

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

Sequence Database (SDB)

- Sequence : < Milk Coffee Sugar Coffee Sugar >
- Sub-sequence : < Coffee Sugar >



SPM PROBLEM

Sub-sequence
Sequence

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

Sequence Database (SDB)

- Sequence : < Milk Coffee Sugar Coffee Sugar >
- Sub-sequence : < Coffee Sugar >
- Support (< Coffee Sugar >) = 3

X

SPM PROBLEM

Sub-sequence
Sequence

Client1	Milk	Coffee	Sugar	Coffee	Sugar
Client2	Coffee	Milk	Coffee	Sugar	
Client3	Milk	Coffee			
Client4	Coffee	Sugar	Egg		

Sequence Database (SDB)

- Sequence : < Milk Coffee Sugar Coffee Sugar >
- Sub-sequence : < Coffee Sugar >
- Support (< Coffee Sugar >) = 3

Problem : Find all subsequences with support \geq Given Threshold

X

Related Work

Timeline

Specialized Methods

X

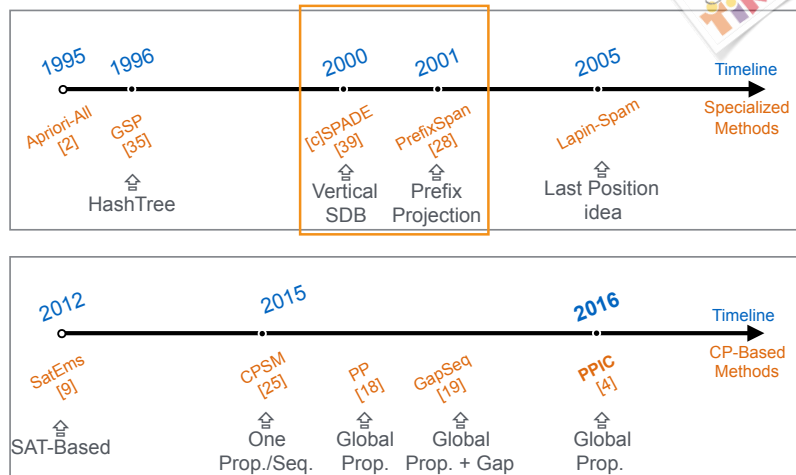
Related Work

Timeline

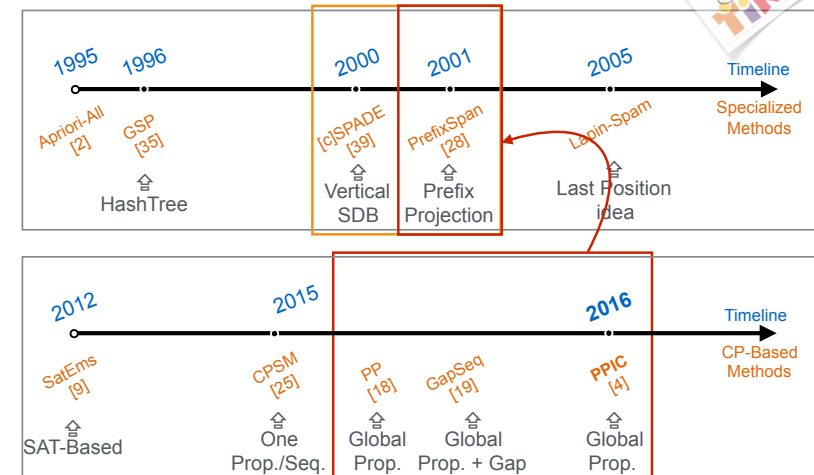
Specialized Methods

X

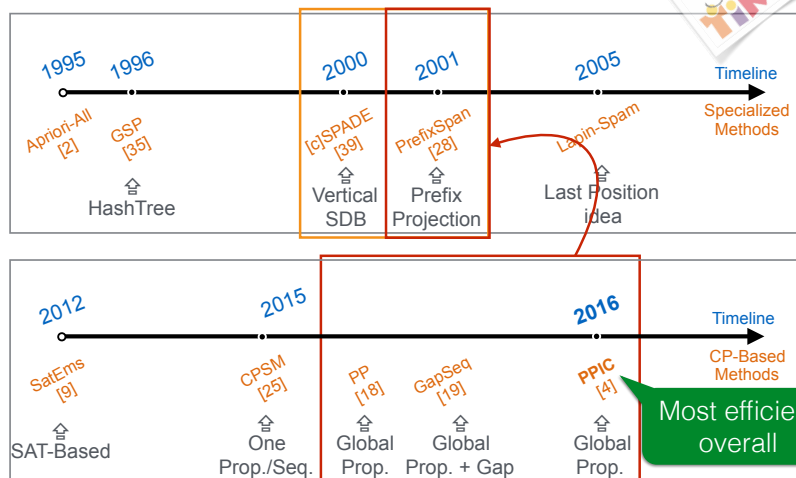
Related Work



Related Work



Related Work



PPICT : CONTRIBUTION

Goal: Capture the most common time-related constraints: namely timed events, minimum/maximum gap and span

- ✓ Adapt trailed-based data structure to efficiently capture **all** valid embeddings (previously only smallest needed)
- ✓ Algorithmic improvements to avoid scanning overlapping time windows, and to efficiently compute the frequency of symbols
- ✓ Can be combined with many other constraints: Regular/ Grammar, Gcc, Among, ...



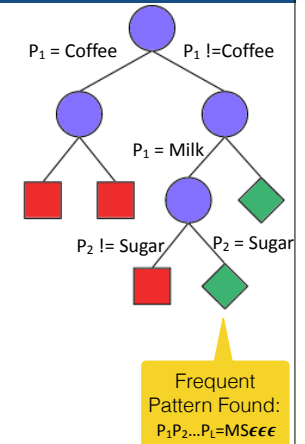
CP : Filtering + DFSearch

Vi	P1	P2	P3	P4	P5
	ε	ε	ε	ε	ε
	Milk	Milk	Milk	Milk	Milk
Di	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



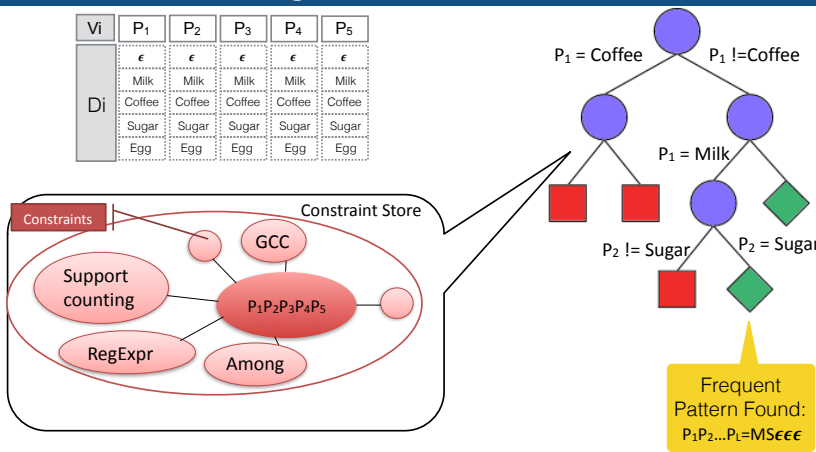
CP : Filtering + DFSearch

Vi	P1	P2	P3	P4	P5
	ε	ε	ε	ε	ε
	Milk	Milk	Milk	Milk	Milk
Di	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



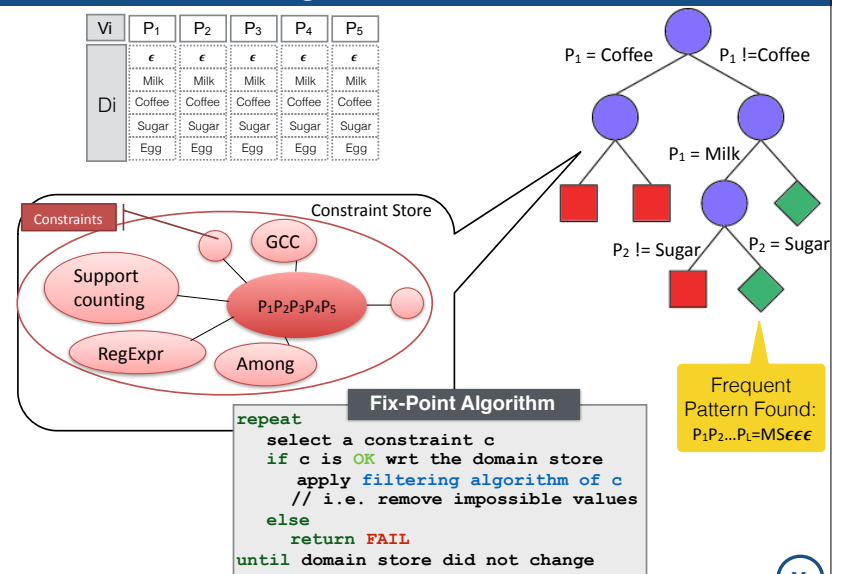
CP : Filtering + DFSearch

Vi	P1	P2	P3	P4	P5
	ε	ε	ε	ε	ε
	Milk	Milk	Milk	Milk	Milk
Di	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



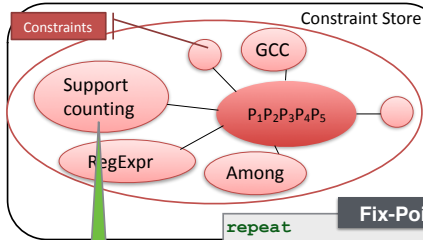
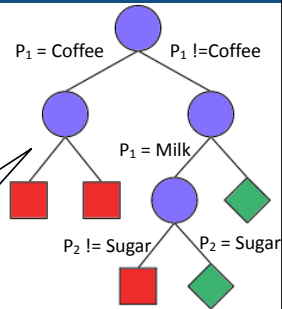
CP : Filtering + DFSearch

Vi	P1	P2	P3	P4	P5
	ε	ε	ε	ε	ε
	Milk	Milk	Milk	Milk	Milk
Di	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



CP : Filtering + DFSearch

Vi	P ₁	P ₂	P ₃	P ₄	P ₅
	ϵ	ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
D _i	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



```

Fix-Point Algorithm
repeat
  select a constraint c
  if c is OK wrt the domain store
    apply filtering algorithm of c
    // i.e. remove impossible values
  else
    return FAIL
until domain store did not change
    
```

This is main bottleneck

Frequent Pattern Found:
P₁P₂...P_i=MSEϵϵ



MinSup=3
(75%)

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		



MinSup=3
(75%)

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

	P ₁	P ₂	P ₃	P ₄	P ₅
	ϵ	ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



MinSup=3
(75%)

	0	1	2	3	4
1	M	C	S	C	S
2	C	M	C	S	
3	M	C			
4	C	S	E		

Supports
M:

	P ₁	P ₂	P ₃	P ₄	P ₅
	ϵ	ϵ	ϵ	ϵ	ϵ
	Milk	Milk	Milk	Milk	Milk
	Coffee	Coffee	Coffee	Coffee	Coffee
	Sugar	Sugar	Sugar	Sugar	Sugar
	Egg	Egg	Egg	Egg	Egg



MinSup=3
(75%)

0 1 2 3 4
1 MCSCS
2 CMCS
3 MC
4 CSE

Supports

M : 3
C : 4
S : 3
E : 1

	P ₁	P ₂	P ₃	P ₄	P ₅
	€	€	€	€	€
Milk	Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg	Egg



MinSup=3
(75%)

0 1 2 3 4
1 MCSCS
2 CMCS
3 MC
4 CSE

Supports

M : 3
C : 4
S : 3
~~E : 1~~

	P ₁	P ₂	P ₃	P ₄	P ₅
	€	€	€	€	€
Milk	Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg	Egg



MinSup=3
(75%)

0 1 2 3 4
1 MCSCS
2 CMCS
3 MC
4 CSE

Supports

M : 3
C : 4
S : 3
~~E : 1~~

	P ₁	P ₂	P ₃	P ₄	P ₅
	€	€	€	€	€
Milk	Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg	Egg



MinSup=3
(75%)

0 1 2 3 4
1 MCSCS
2 CMCS
3 MC
4 CSE

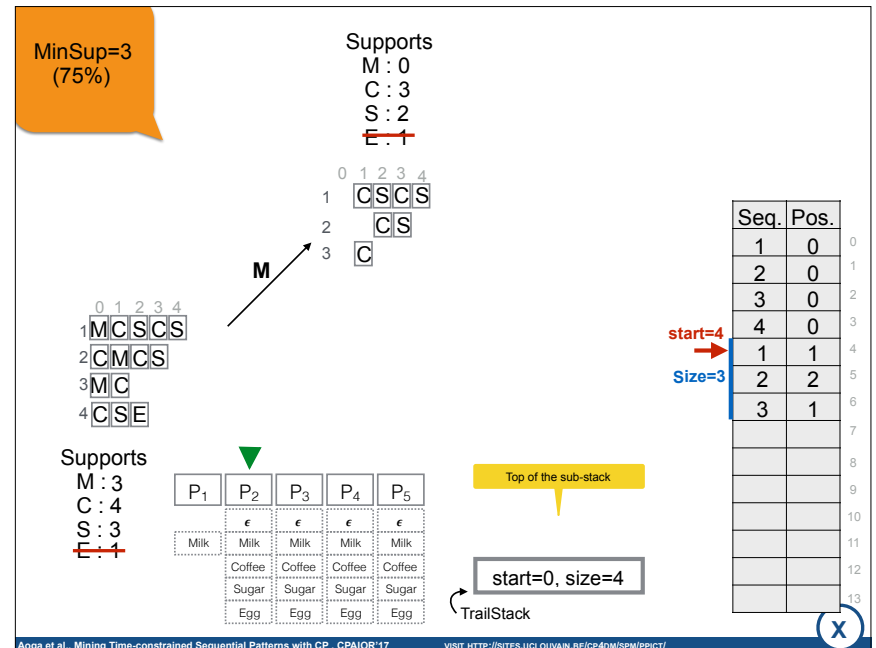
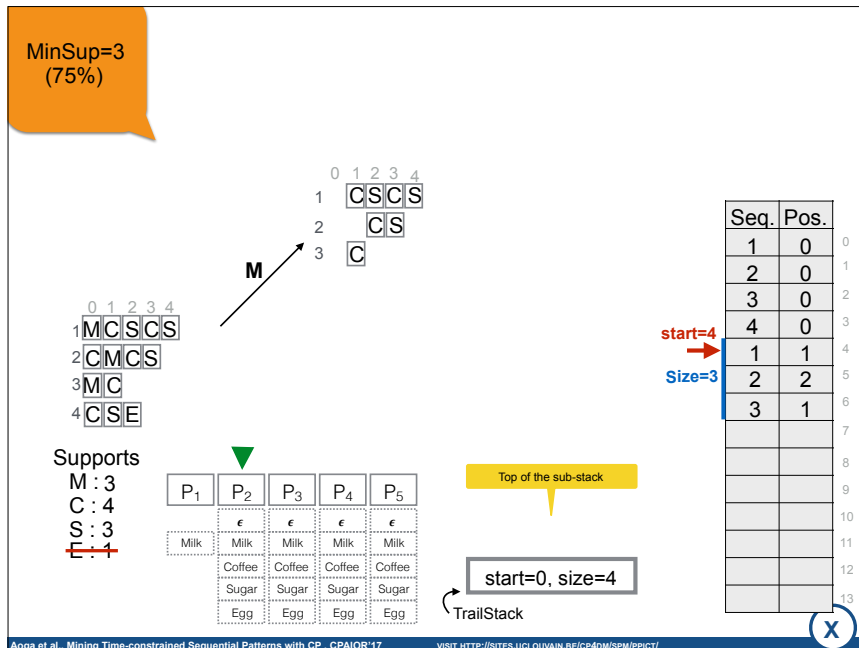
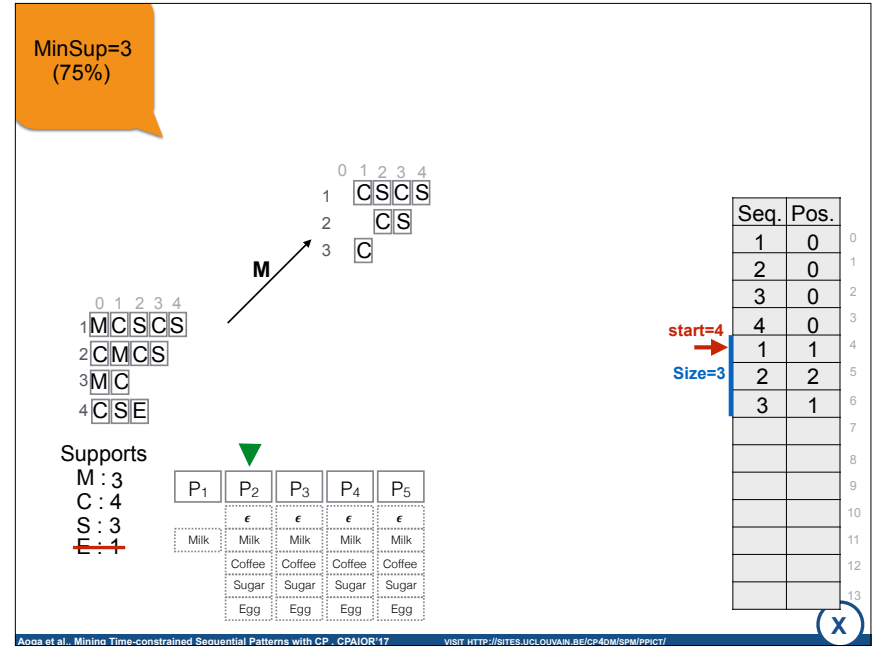
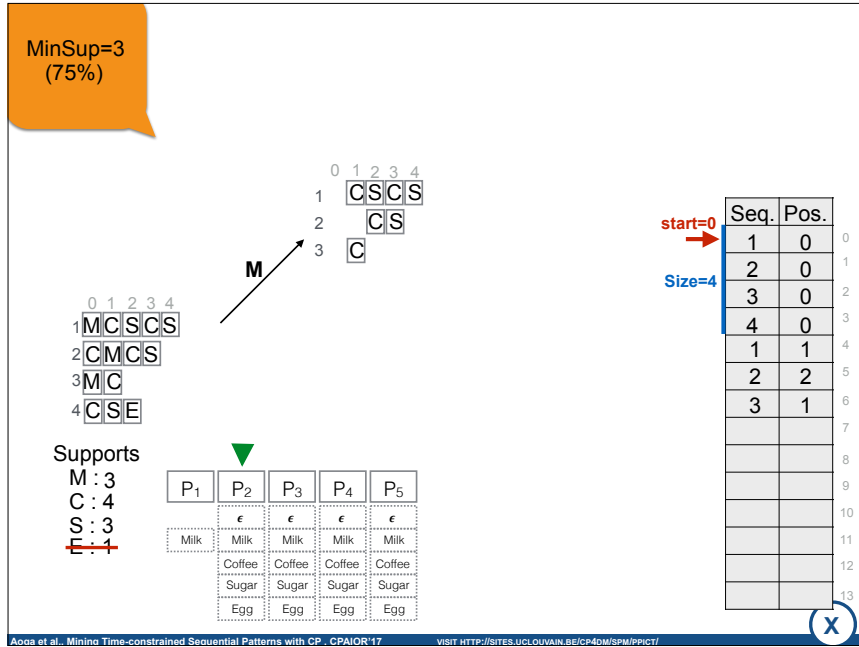
Supports

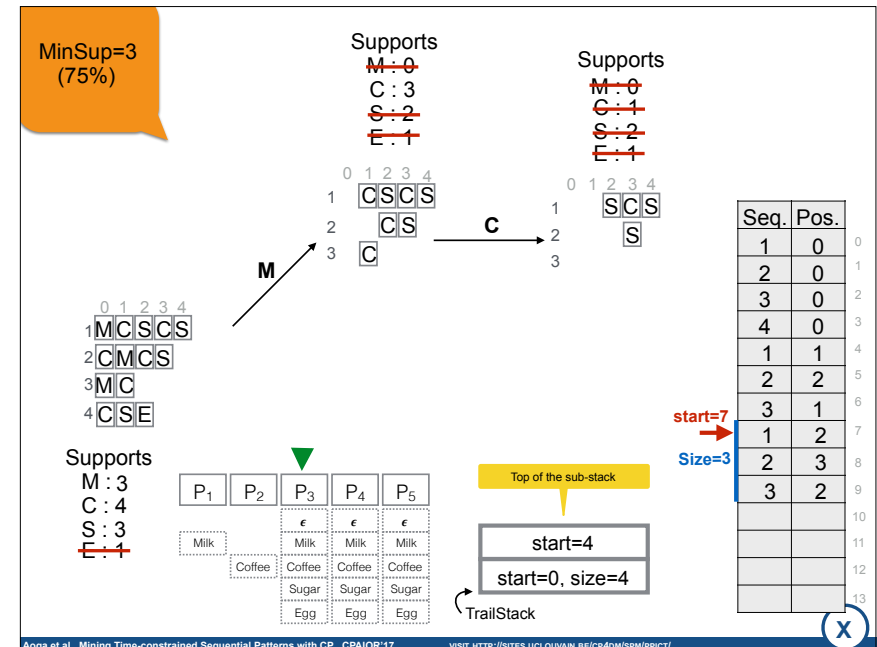
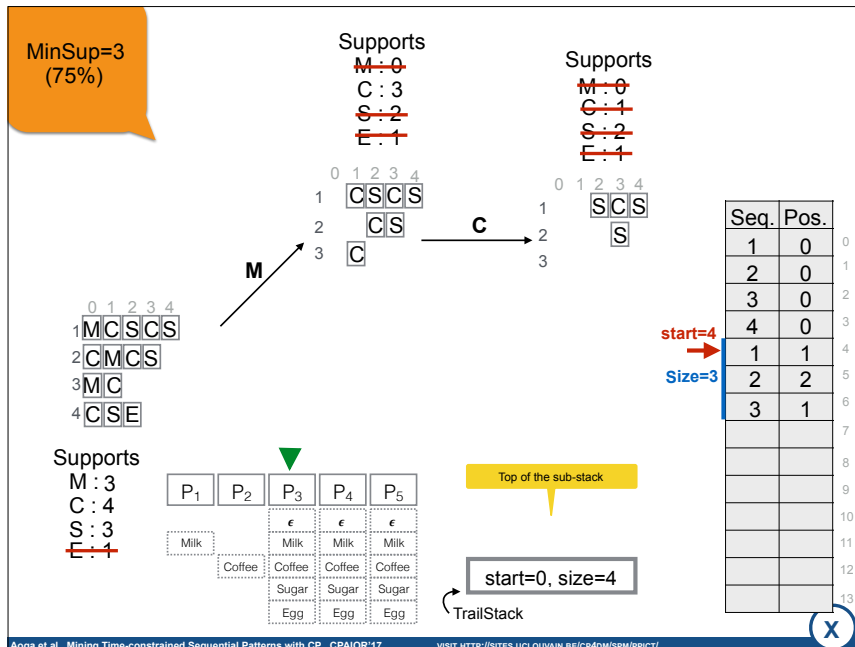
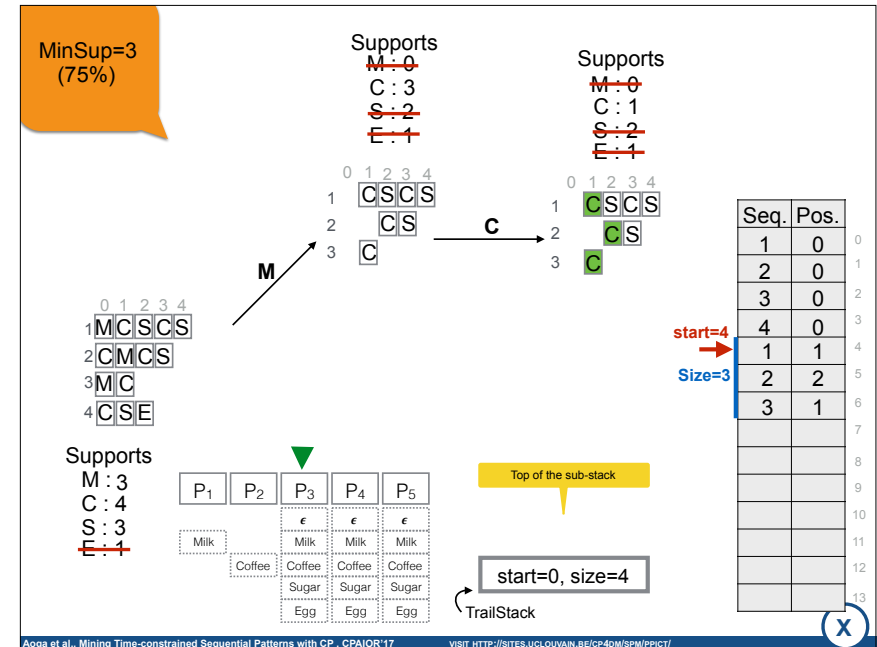
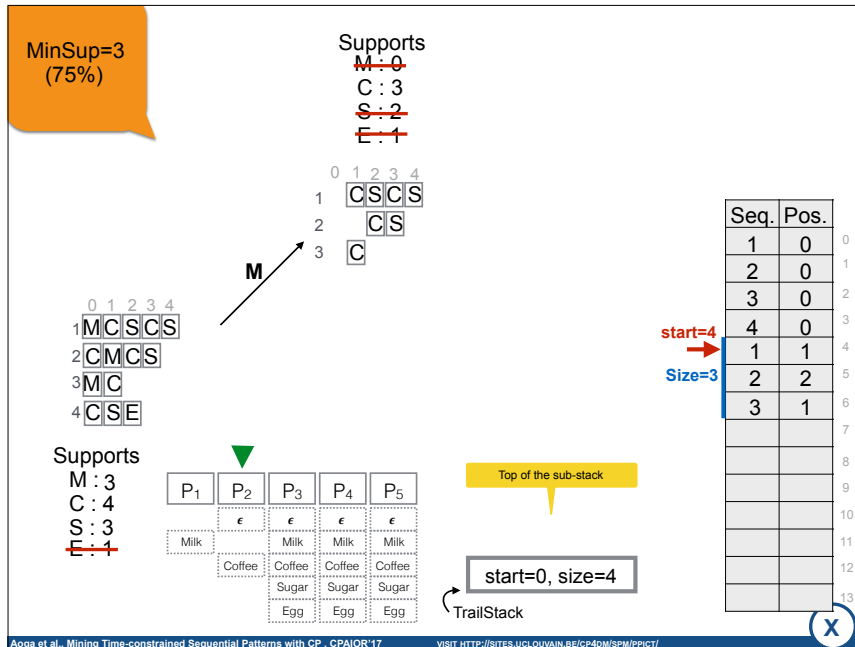
M : 3
C : 4
S : 3
~~E : 1~~

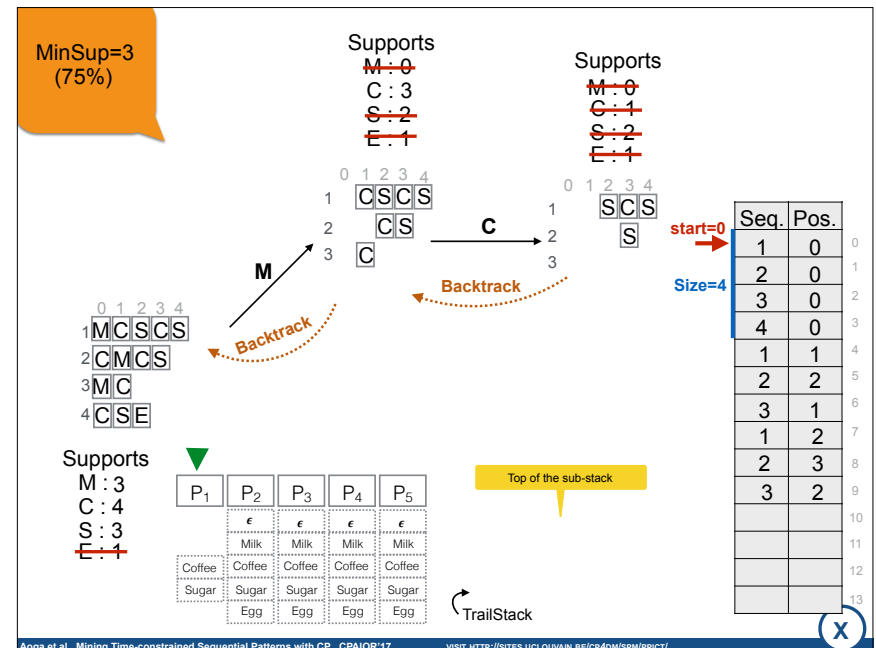
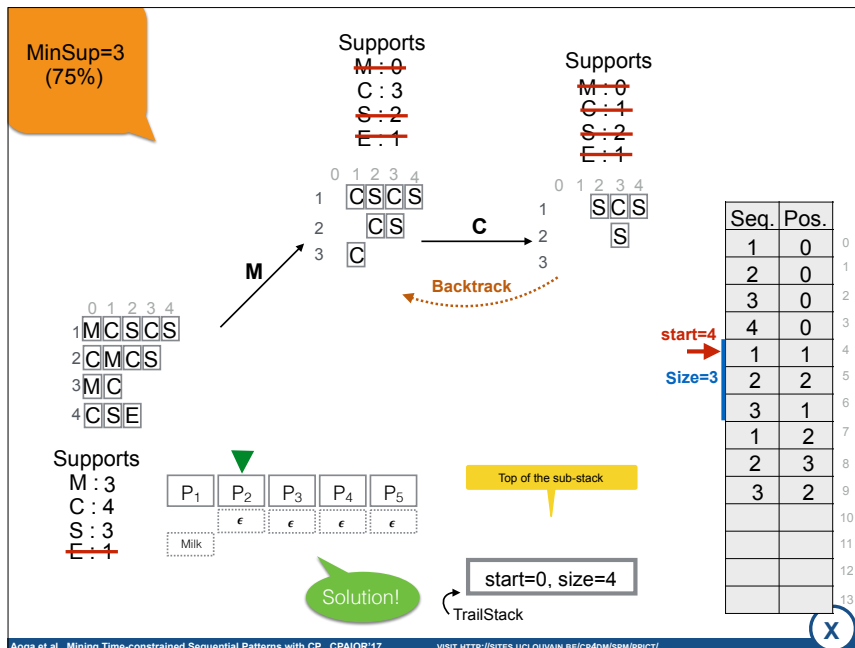
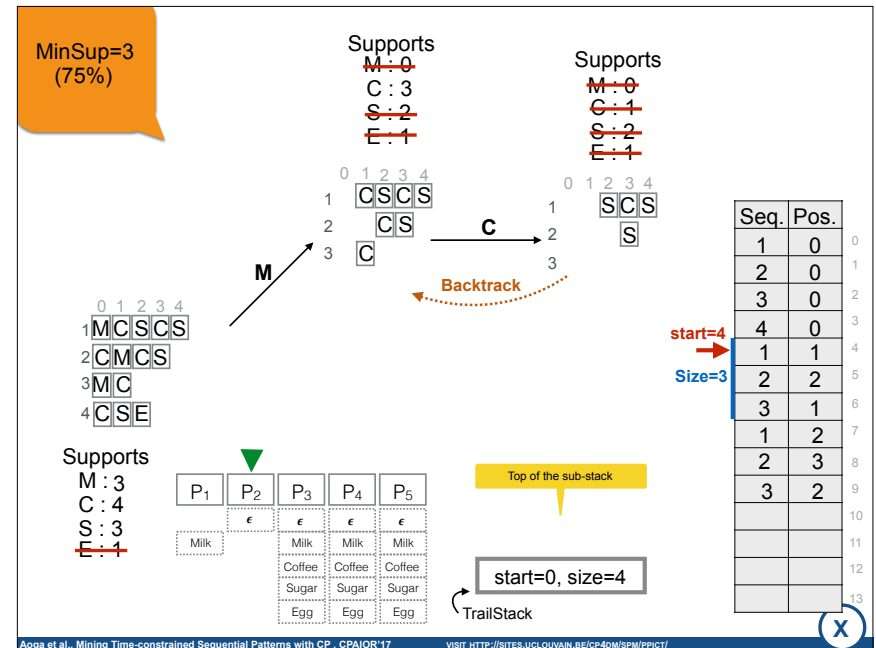
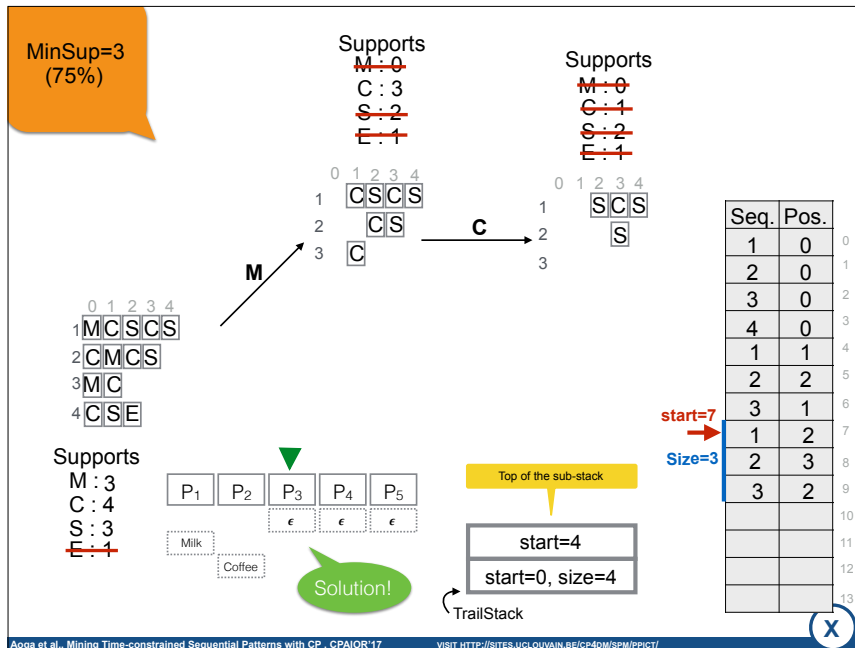
	P ₁	P ₂	P ₃	P ₄	P ₅
	€	€	€	€	€
Milk	Milk	Milk	Milk	Milk	Milk
Coffee	Coffee	Coffee	Coffee	Coffee	Coffee
Sugar	Sugar	Sugar	Sugar	Sugar	Sugar
Egg	Egg	Egg	Egg	Egg	Egg

	Seq.	Pos.	
start=0 →	1	0	0
	2	0	1
Size=4	3	0	2
	4	0	3
			4
			5
			6
			7
			8
			9
			10
			11
			12
			13









TIME DATABASE CHALLENGES

Gap[M,N] =
Minimum(M) and
Maximum(N) gap time
restriction

Span[Y,W] =
Minimum(Y) and
Maximum(W) span
time restriction

	gap=3		span=8			gap=5		
Client1	(2, Milk)	(5, Coffee)	(6, Egg)	(10, Sugar)	(11, Coffee)			
Client2	(1, Coffee)	(2, Milk)	(9, Milk)	(12, Egg)	(15, Sugar)	(18, Milk)	(24, Coffee)	
Client3	(2, Milk)	(4, Coffee)	(6, Egg)	(8, Egg)	(10, Coffee)	(12, Wine)	(14, Sugar)	
Client4	(2, Milk)	(5, Sugar)	(6, Sugar)	(10, Coffee)				

Sequence Database (SDB)

- Client1
- gap[3,7] (<(2, Milk)(6, Egg)(10, Sugar)>)
 - gap[3,7] (<(2, Milk)(10, Sugar)>)



TIME DATABASE CHALLENGES

Gap[M,N] =
Minimum(M) and
Maximum(N) gap time
restriction

Span[Y,W] =
Minimum(Y) and
Maximum(W) span
time restriction

	gap=3		span=8			gap=5		
Client1	(2, Milk)	(5, Coffee)	(6, Egg)	(10, Sugar)	(11, Coffee)			
Client2	(1, Coffee)	(2, Milk)	(9, Milk)	(12, Egg)	(15, Sugar)	(18, Milk)	(24, Coffee)	
Client3	(2, Milk)	(4, Coffee)	(6, Egg)	(8, Egg)	(10, Coffee)	(12, Wine)	(14, Sugar)	
Client4	(2, Milk)	(5, Sugar)	(6, Sugar)	(10, Coffee)				

Sequence Database (SDB)

- Client1
- gap[3,7] (<(2, Milk)(6, Egg)(10, Sugar)>)
 - gap[3,7] (<(2, Milk)(10, Sugar)>)

non anti-monotone



TIME DATABASE CHALLENGES

Gap[M,N] =
Minimum(M) and
Maximum(N) gap time
restriction

Span[Y,W] =
Minimum(Y) and
Maximum(W) span
time restriction

	gap=3		span=8			gap=5		
Client1	(2, Milk)	(5, Coffee)	(6, Egg)	(10, Sugar)	(11, Coffee)			
Client2	(1, Coffee)	(2, Milk)	(9, Milk)	(12, Egg)	(15, Sugar)	(18, Milk)	(24, Coffee)	
Client3	(2, Milk)	(4, Coffee)	(6, Egg)	(8, Egg)	(10, Coffee)	(12, Wine)	(14, Sugar)	
Client4	(2, Milk)	(5, Sugar)	(6, Sugar)	(10, Coffee)				

Sequence Database (SDB)

- Client1
- gap[3,7] (<(2, Milk)(6, Egg)(10, Sugar)>)
 - gap[3,7] (<(2, Milk)(10, Sugar)>)
- Client2
- gap[3,7] (<(2, Milk)(12, Egg)(15, Sugar)>)
 - gap[3,7] (<(9, Milk)(12, Egg)(15, Sugar)>)

non anti-monotone



TIME DATABASE CHALLENGES

Gap[M,N] =
Minimum(M) and
Maximum(N) gap time
restriction

Span[Y,W] =
Minimum(Y) and
Maximum(W) span
time restriction

	gap=3		span=8			gap=5		
Client1	(2, Milk)	(5, Coffee)	(6, Egg)	(10, Sugar)	(11, Coffee)			
Client2	(1, Coffee)	(2, Milk)	(9, Milk)	(12, Egg)	(15, Sugar)	(18, Milk)	(24, Coffee)	
Client3	(2, Milk)	(4, Coffee)	(6, Egg)	(8, Egg)	(10, Coffee)	(12, Wine)	(14, Sugar)	
Client4	(2, Milk)	(5, Sugar)	(6, Sugar)	(10, Coffee)				

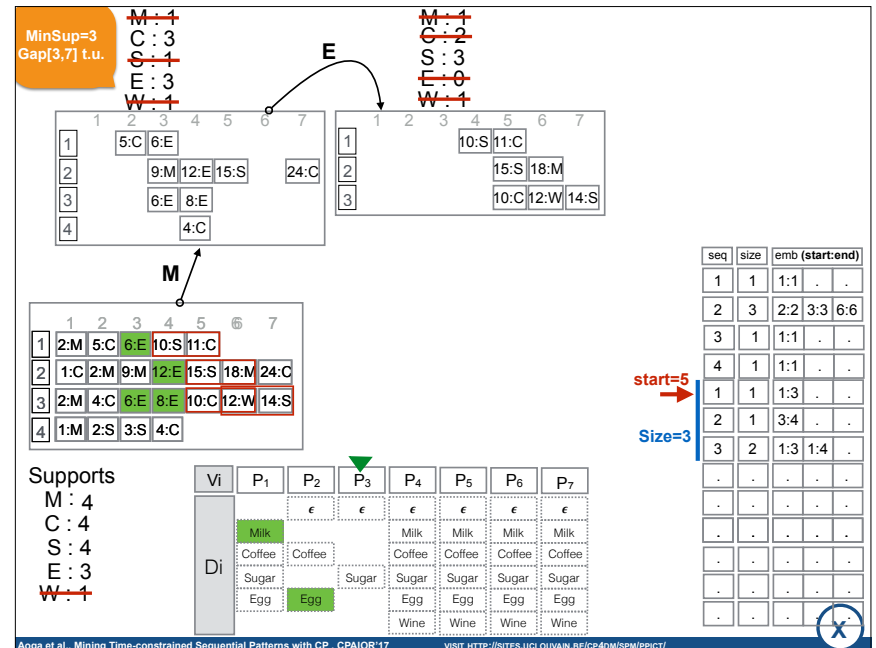
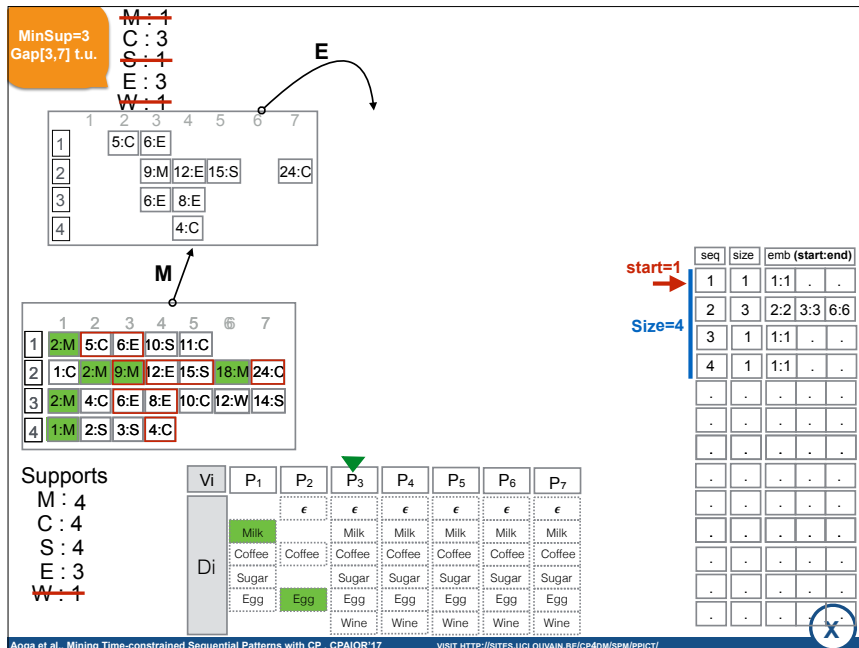
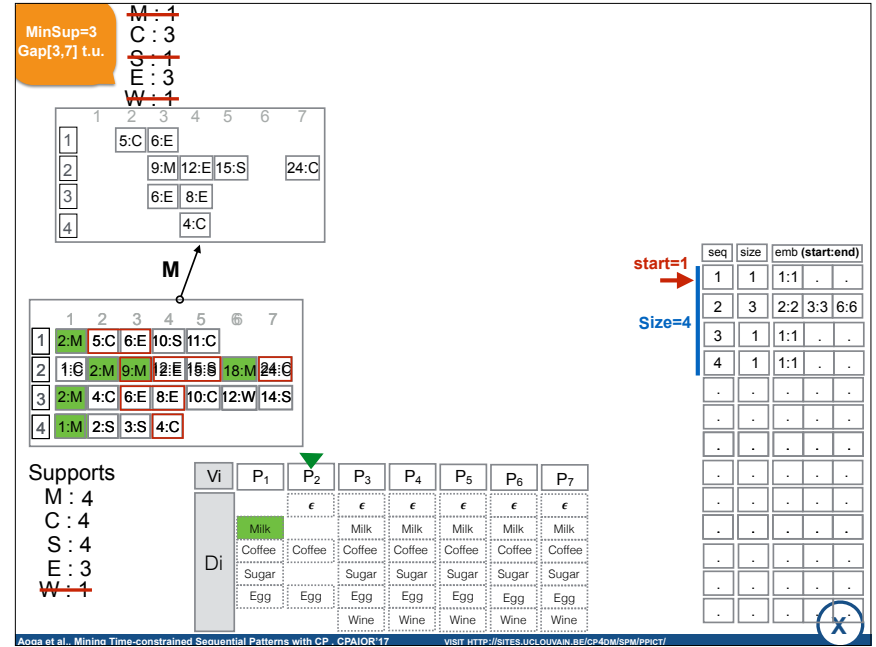
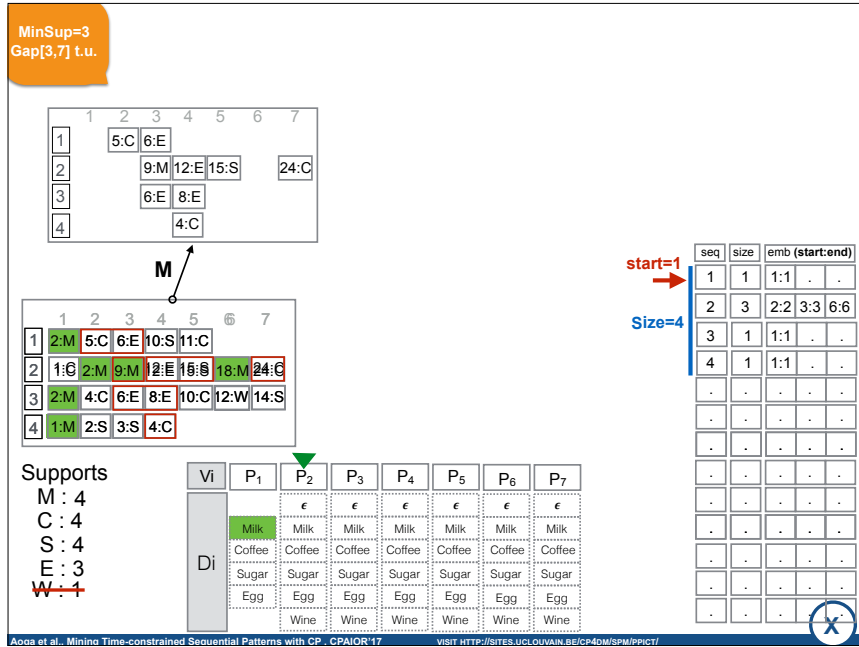
Sequence Database (SDB)

- Client1
- gap[3,7] (<(2, Milk)(6, Egg)(10, Sugar)>)
 - gap[3,7] (<(2, Milk)(10, Sugar)>)
- Client2
- gap[3,7] (<(2, Milk)(12, Egg)(15, Sugar)>)
 - gap[3,7] (<(9, Milk)(12, Egg)(15, Sugar)>)

non anti-monotone

Prefix notion non-applicable

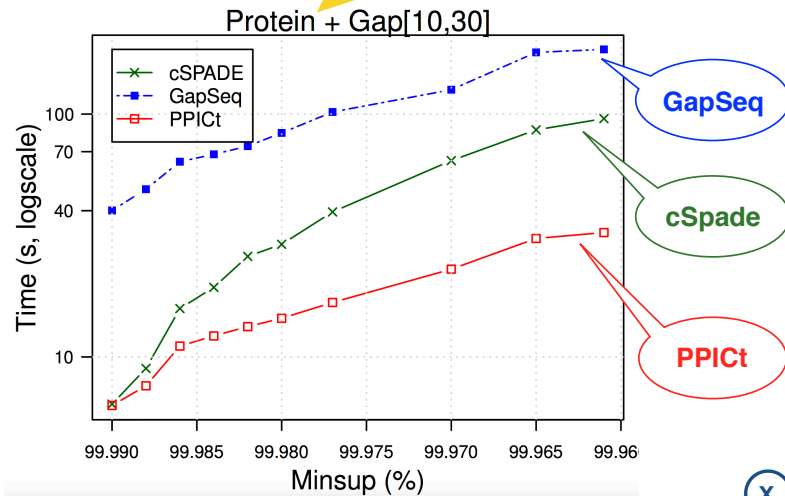




COMPARED WITH EXISTING METHODS

Time limit = 3600s (1Hour)

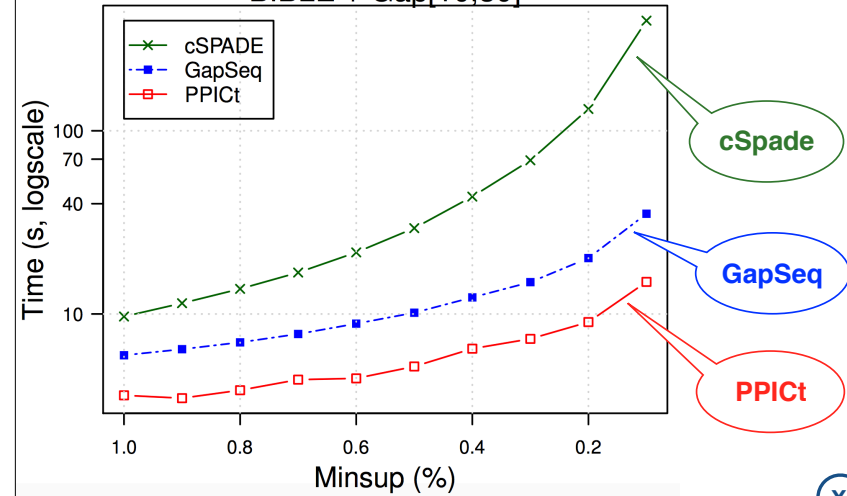
Largest and densest dataset (49,729,890 symbols) 600 variables



COMPARED WITH EXISTING METHODS

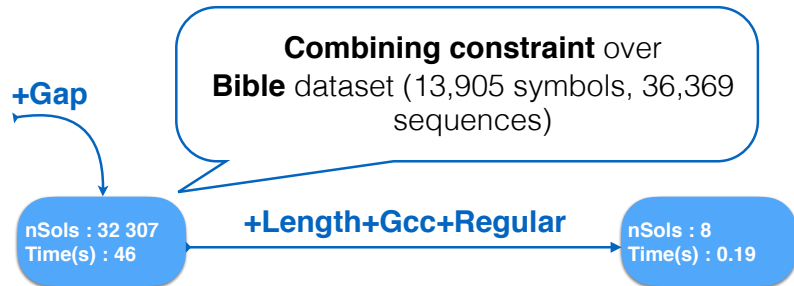
sparse dataset (787,066 symbols) 100 variables

BIBLE + Gap[10,30]



Handling of different additional constraints

Methods	Frequency	Gap	Span	Regular/ Grammar	Among/ Gcc	Length
PPICt	x	x	x	x	x	x
GapSeq	x	x*			x	x
cSPADE	x	x	x**			x



Take-Away message

- Combining both SPM and CP techniques can lead to very efficient, modular and flexible approaches.
- Many kind of existing modules (in CP-Solvers) are reusable for free
- **Efficient memory using Trail-based backtracking aware data structure** really speed up search in DFSearch (not only for data mining)
- Code, data and apps are open
<http://sites.uclouvain.be/cp4dm/spm/>



