

# Integrating Constraint Programming And Itemset Mining



Siegfried Nijssen and *Tias Guns*

DTAI, K.U.Leuven, Belgium

# Constraint-based Itemset Mining

Analysing purchases (e.g. items = books),  
to find interesting patterns (sets of items)



Interestingness: constraints

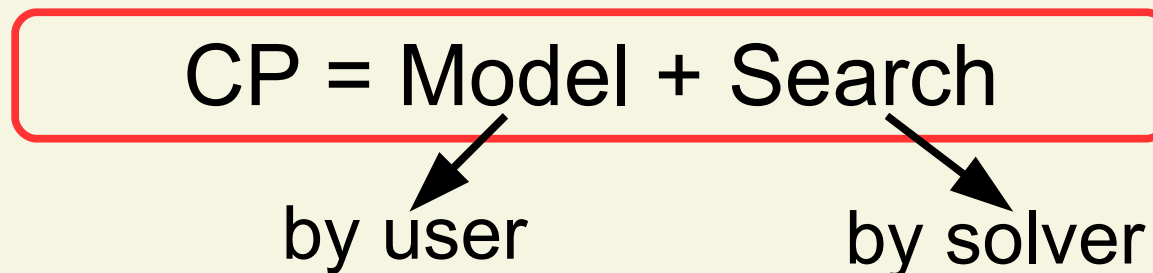
- Frequent sets, closed sets, correlated sets, ...
- Maximum size, minimum price, average cost, ...

Many constraints, many algorithms

# Constraint Programming

General methodology for solving constraint satisfaction problems.

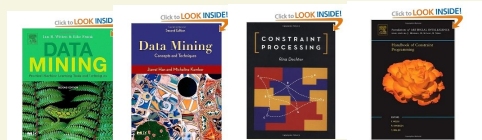
- Wide range of applications
- Each constraint is independent
- Freely combine constraints



# Constraint Programming (CP) for IM

- Variables: booleans

Model



:  $[i_1 \dots i_n]$

 :  $[t_1 \dots t_m]$

- Constraints: Many + combinations

Search

- Out-of-the-box Gecode CP solver (2008)

Integrating solver/miner principles:  
practical AND theoretical benefits

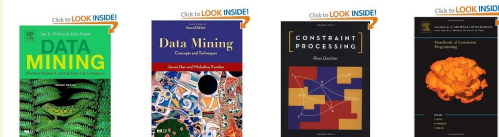
# Overview

1. Motivation
- 2. Principles of IM & CP**
3. Integrating IM & CP
4. Theoretical benefits
5. Practical benefits
6. Conclusions




# Principles of IM and CP

Will compare them on following implementations:

- **Eclat**: Simple and effective itemset miner
- **FIMCP**: our 2008 system, using the out-of-the-box Gecode CP solver



# Itemset Mining principles

	1	0	1	1
	1	1	0	1
	0	0	1	1

- Search strategy *Level-wise, BFS, DFS*
- Representation of data

1	0	1	1
1	1	0	1
0	0	1	1

1	0	1	1
1	1	0	1
0	0	1	1

1	0	1	1
1	1	0	1
0	0	1	1

- Representation of sets

1	0	1	1
---	---	---	---

  
{1, 3, 4}

0	1	0	0
---	---	---	---

  
{2}

# Comparison IM principles

	Eclat Miner	Gecode CP Solver												
Search Strategy	DFS	DFS (binary)												
Repres. of data	Shared, vertical <table border="1" data-bbox="1059 991 1300 1086"><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr></table>	1	0	1	1	1	1	0	1	0	0	1	1	In constraints (up to 4 copies)
1	0	1	1											
1	1	0	1											
0	0	1	1											
Repres. of sets	Sparse or Dense	Sparse {1, 3, 4}												



# Constraint Programming principles

- Types of variables *Bool, Int, Set, ...*
- Supported constraints *Clause, Sum, AllDifferent, ...*
- Constraint activation

*On change of domain, on change of upper/lower bound, on change of specific value, ...*

# Comparing CP principles

	Eclat Miner	Gecode CP Solver
Types of vars.	Boolean vector (set)	Bool, Int, Set, ...
Constraints	Few, hard to combine	Many, easy to add/combine
Constraint activation	Fixed order (in algorithm)	On domain change

# Overview

1. Motivation
2. Principles of IM & CP
- 3. Integrating IM & CP**
4. Theoretical benefits
5. Practical benefits
6. Conclusions

# Integrating IM & CP

We created a new CP solver called DMCP,

- using principles of both IM and CP
- implementing constraints for itemset mining

# Integration 1/3

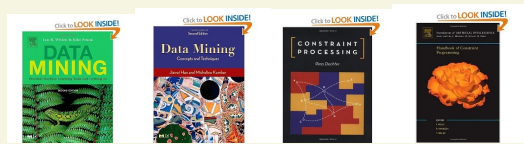
	Eclat Miner	Gecode CP Solver	Our <b>DMCP</b> CP Solver
Search strategy	DFS	DFS (binary)	DFS (binary)
Repres. of data	Shared, vertical	In constraints (up to 4 copies)	Shared matrix (default: vertical)

- Data shared (read-only) by constraints
- Horizontal, positive and negative *views* available

# Integration 2/3

	Eclat Miner	Gecode CP Solver	Our <b>DMCP</b> CP Solver
Repres. of sets	Sparse or Dense	Sparse	Sparse or Dense
Types of vars.	Boolean vector (set)	Bool, Int, Set, ...	Boolean vector (set)

- Represented by lower and upper bound:



0, 0/1, 0/1, 1

Min: {0, 0, 0, 1}

Max: {0, 1, 1, 1}

# Integration 3/3

	Eclat Miner	Gecode CP Solver	Our <b>DMCP</b> CP Solver
Constraints	Few, hard to combine	Many, easy to add/combine	Some, easy add/combine
Constraint activ.	Strict order (in algorithm)	On domain change	Change of lower/upper bound

- General matrix constraint:

$$\mathbf{X} \geq_1 \mathbf{1} \geq_2 \theta(\mathcal{A} \cdot \mathbf{Y});$$

Boolean vectors

Data representation (matrix)

# Overview

1. Motivation
2. Principles of IM & CP
3. Integrating IM & CP
- 4. Theoretical benefits**
5. Practical benefits
6. Conclusions





# Theoretical benefits

Can prove polynomial delay of

- Frequent itemset mining
- Closed itemset mining
- A related graph mining problem\*
- more ?

\* M. Boley et al. Efficient closed pattern mining in strongly accessible set systems. PKKD 2007

# Overview

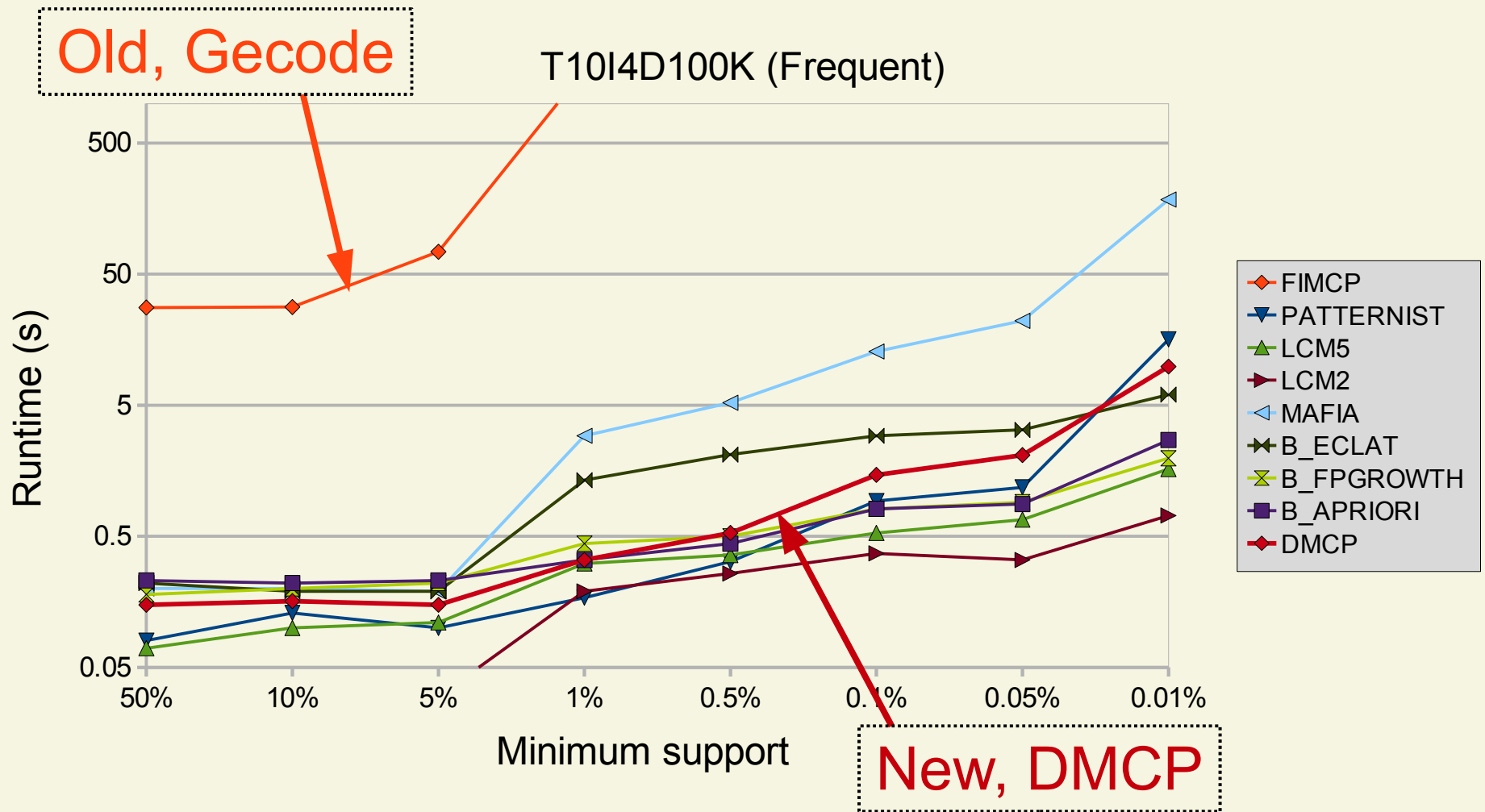
1. Motivation
2. Principles of IM & CP
3. Integrating IM & CP
4. Theoretical benefits
- 5. Practical benefits**
6. Conclusions

# Mining systems

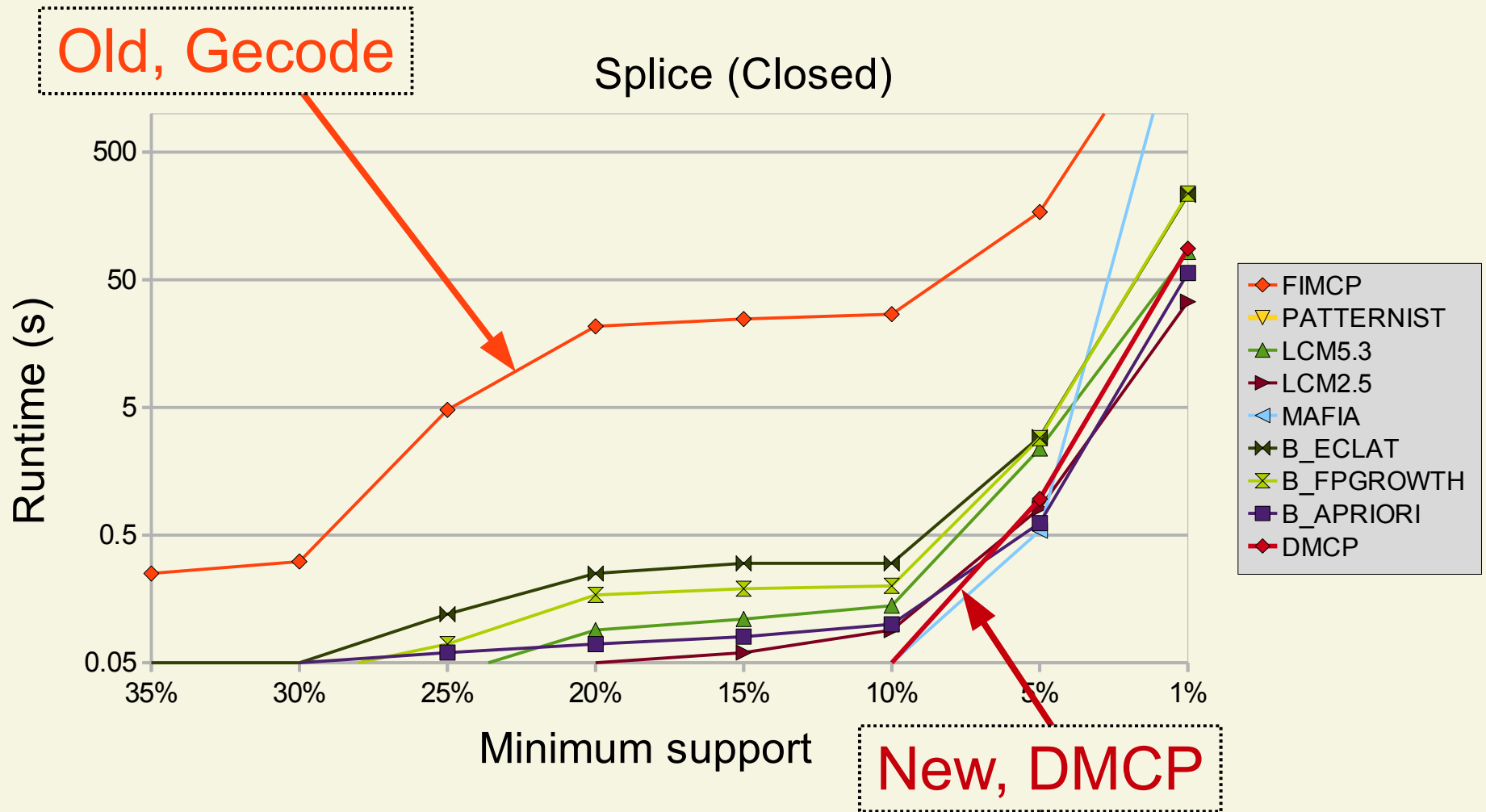
- DMCP: our new integrated CP/IM system.
- FIMCP: our Gecode based system
- PATTERNIST: constraint-based itemset miner
- LCM: 'winner' of the FIMI competition
- ECLAT, FPGrowth and APRIORI,  
as implemented by C. Borgelt



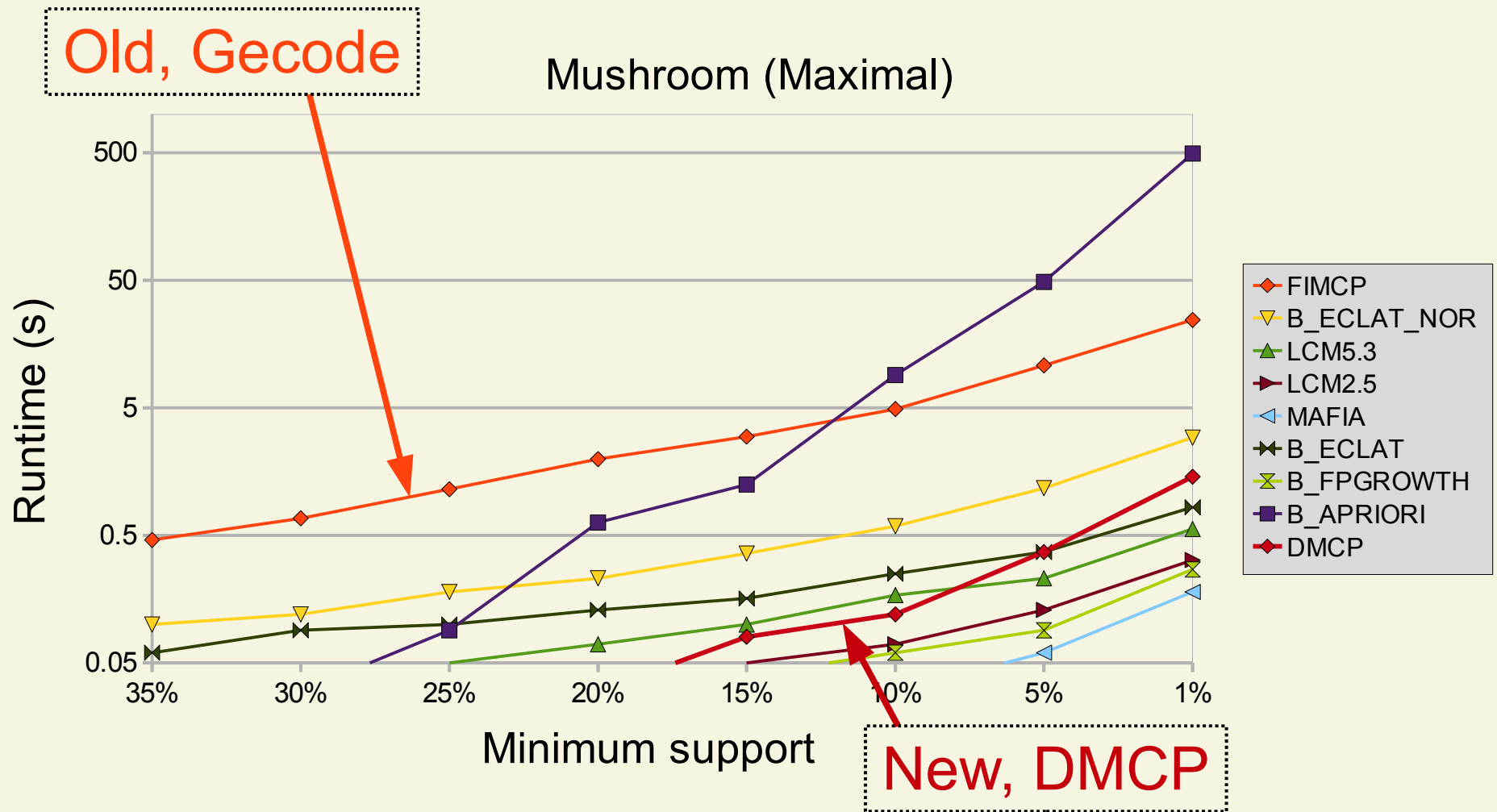
# Frequent Itemset Mining, scaling



# Closed Itemset Mining

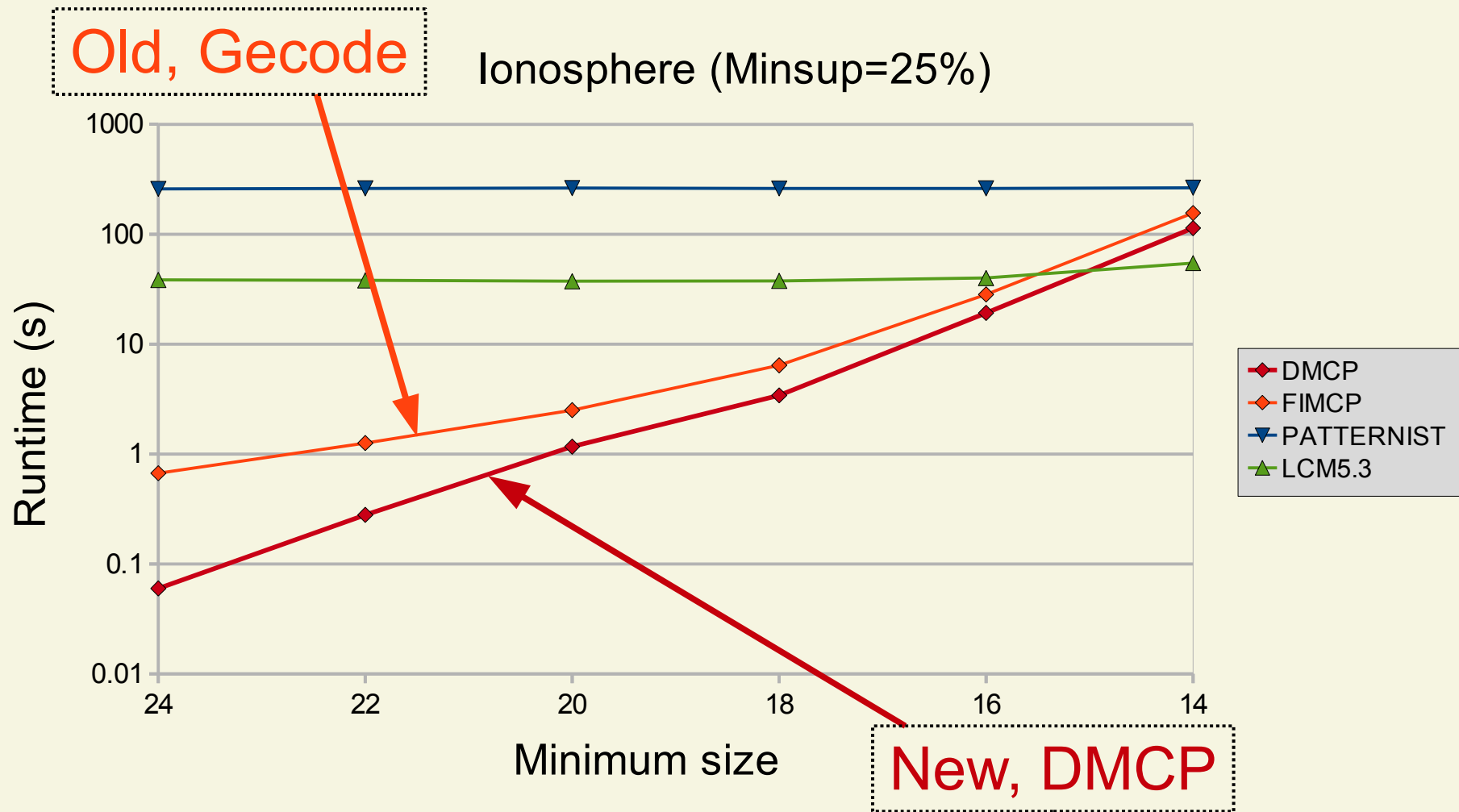


# Maximal Itemset Mining





# Minimum size (monotone)



# Overview

1. Motivation
2. Principles of IM & CP
3. Integrating IM & CP
4. Theoretical benefits
5. Practical benefits
- 6. Conclusions**

# Conclusion

## Advantages of CP modelling:

Model

- Easily add new constraints
- Freely combine constraints

Search

## Advantage of IM/CP solver integration:

- Theoretical: polynomial delay analysis
- Practical: remove efficiency/scalability gap

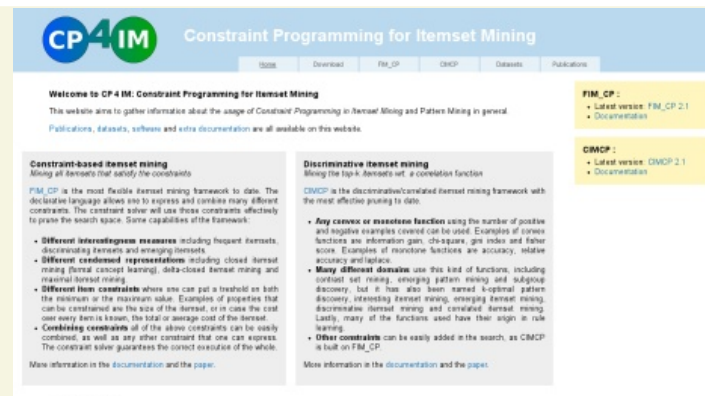
# Open questions

- Integrate IM principles in existing CP solver ?
- More efficient solving of typical CP problems ?
- Other mining strategies (e.g. FPgrowth) in CP ?
- Adding CP principles to other pattern domains (e.g. sequences, trees, graphs) ?

# Thank you for listening

## Questions?

<http://dtai.cs.kuleuven.be/CP4IM>



The screenshot shows the homepage of the CP4IM project. At the top, there is a navigation bar with the CP4IM logo and the text "Constraint Programming for Itemset Mining". Below this, there are links for "Home", "Download", "FM\_CP", "DMCP", "Datasets", and "Publications". The main content area is divided into several sections:

- Welcome to CP4IM: Constraint Programming for Itemset Mining**: A introductory paragraph stating the website's purpose and providing links to publications, datasets, software, and documentation.
- Constraint-based itemset mining**: A section describing the FM\_CP framework, which allows for expressing and combining various constraints. It lists several types of constraints: different interdependencies (frequent itemsets, decreasing itemsets, emerging itemsets), different condensed representations (closed itemset mining, formal concept learning, delta-closed itemset mining, maximal itemset mining), different item constraints (thresholds on both minimum and maximum values, cost constraints), and combining constraints.
- Discriminative itemset mining**: A section describing the DMCP framework, which is a discriminative/constrained itemset mining framework. It lists several types of constraints: any convex or monotone function (information gain, chi-square, gini index, Fisher score, accuracy, relative accuracy, and lift), many different domains (constraint set mining, emerging pattern mining, subgroup discovery, interesting itemset mining, emerging itemset mining, discriminative itemset mining, and correlated itemset mining), and other constraints.

On the right side of the page, there are two yellow boxes with links to the latest versions of the frameworks: "FM\_CP" (latest version: FM\_CP 2.1) and "DMCP" (latest version: DMCP 2.1).