# Iterative Forward Search: Combining Local Search with Maintaining Arc Consistency and a Conflict-based Statistics

Tomáš Müller[1],Roman Barták[1], Hana Rudová[2]

[1] Faculty of Mathematics and Physics, Charles University

{muller|bartak}@ktiml.mff.cuni.cz

[2] Faculty of Informatics, Masaryk University

hanka@fi.muni.cz

# Agenda

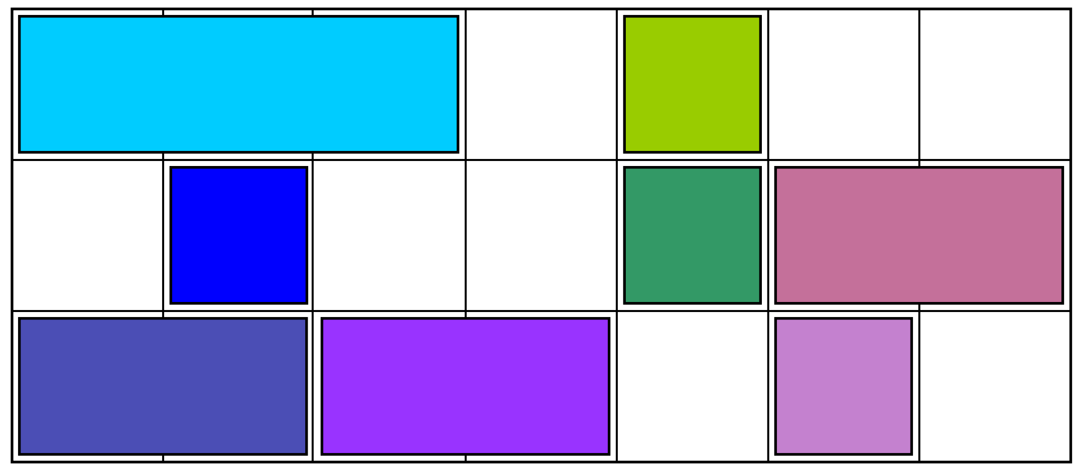
- Iterative Forward Search Algorithm
  - Extensions:
    - Conflict-based statistics
    - Maintaining arc consistency
    - Dynamic backtracking
- Experiments
  - Purdue University Timetabling Problem
  - Random binary CSP
- Conclusion

# Iterative Forward Search

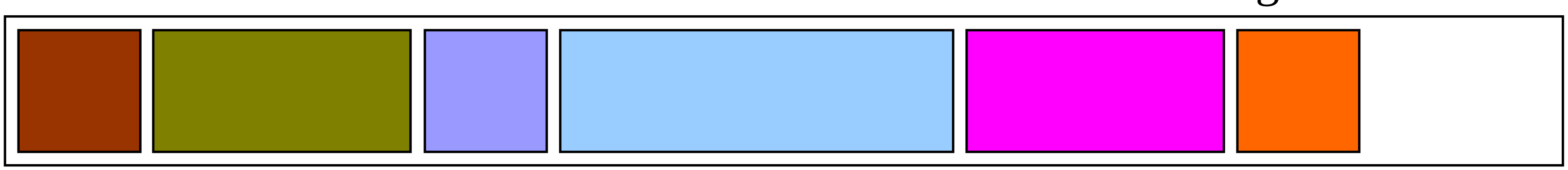
- Basic Approaches
  - Local search
  - Backtracking based search
- Iterative Forward Search Algorithm
  - Forward based search
  - Works in iterations
  - Extending a (partial) feasible solution
  - Interactivity

# Iterative Forward Search Algorithm

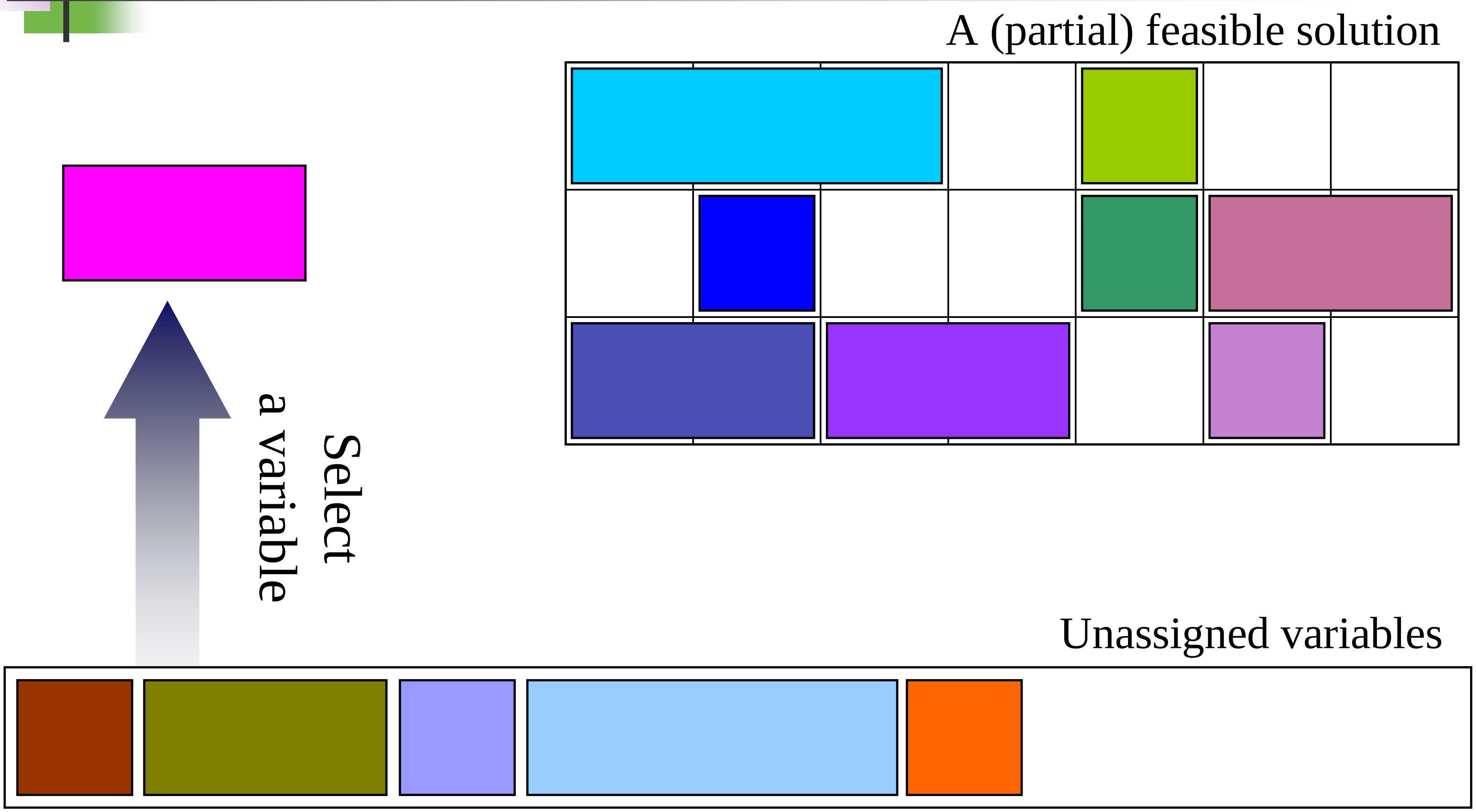
A (partial) feasible solution



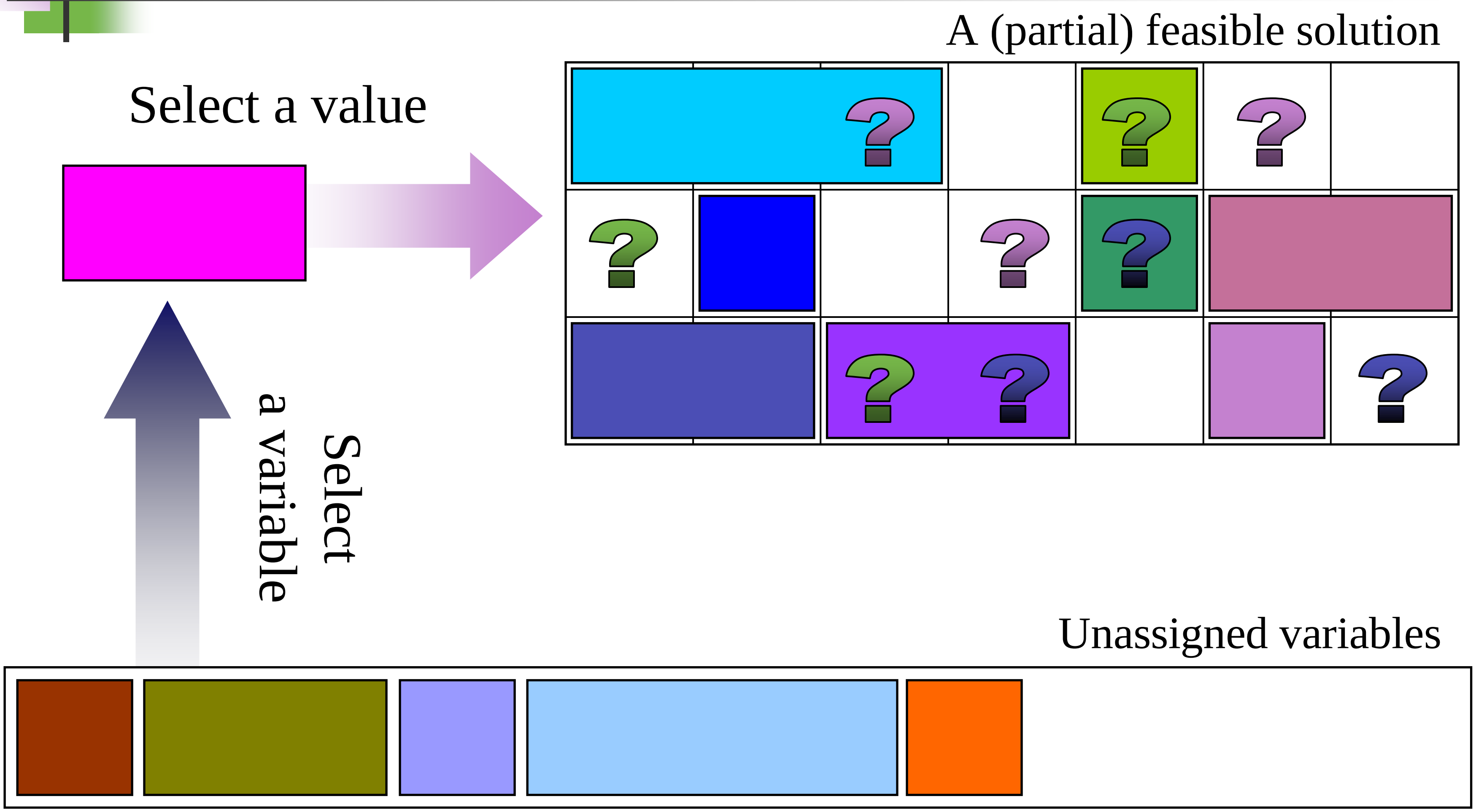
Unassigned variables



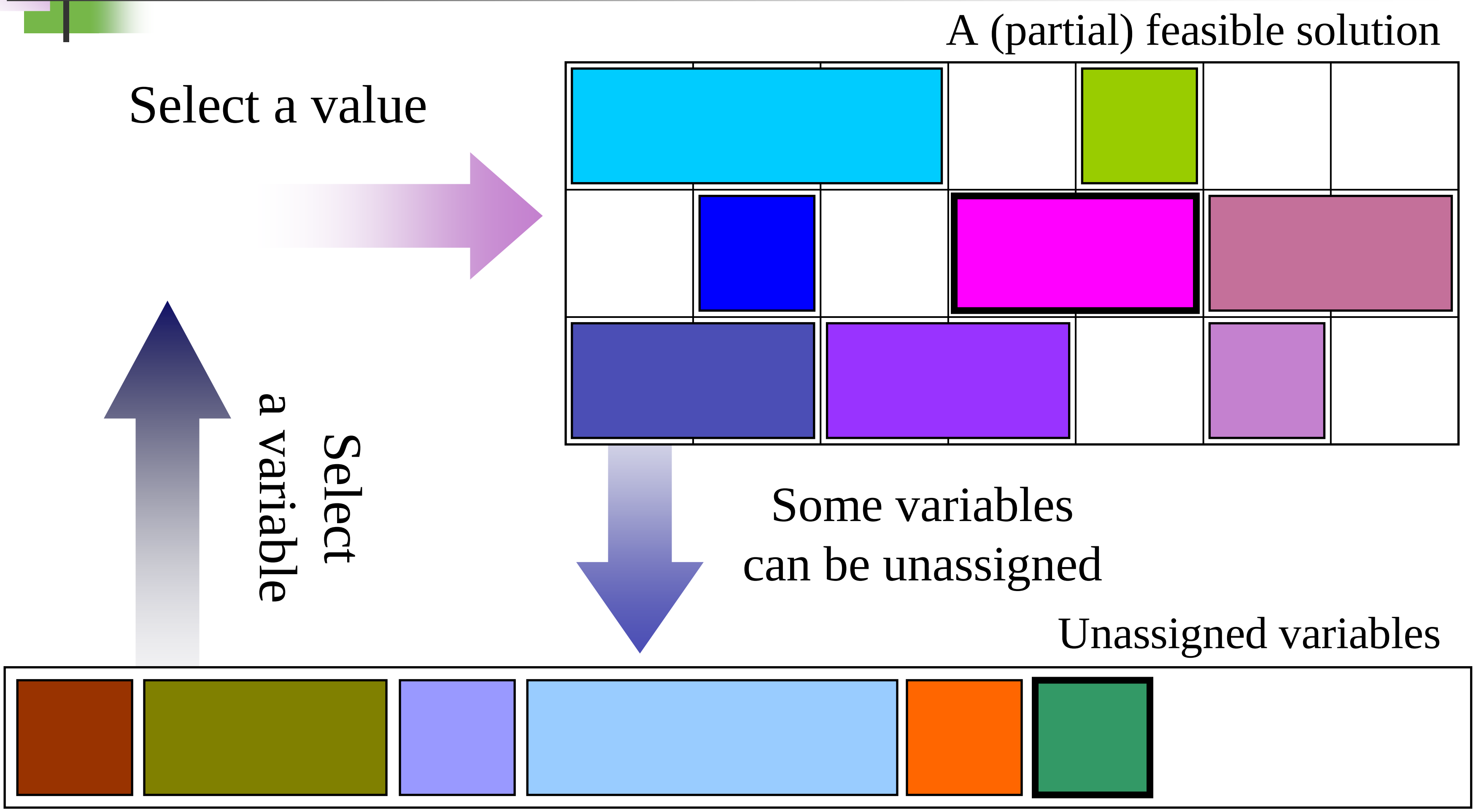
# Iterative Forward Search Algorithm



# Iterative Forward Search Algorithm



# Iterative Forward Search Algorithm



# Iterative Forward Search Algorithm

- Guided by
  - Variable selection
    - First-fail principle
  - Value selection
    - Best-fit value
  - Solution comparator
    - Less unassigned variables, less violated soft constraints, …
  - Termination condition
    - Solution is complete and good enough
    - Timeout or user interaction

# Conflict-based statistics

- Idea
  - Memorize conflicts and discourage their potential repetition

- If $B=c$ is unassigned because of the $A=a$
  - A counter Stat[$A=a,B\neq c$] is incremented

$$A=a \Rightarrow \begin{cases} 3\times B \neq a \\ 4\times B \neq c \\ 2\times C \neq a \\ 120\times D \neq a \end{cases}$$

# Conflict-based statistics

To be used e.g. in value selection

- If $a$ is being selected for variable $A$
- And there is $B=b$ in a conflict with $A=a$

$$\Downarrow$$

Value $a$ is weighted by Stat[$A=a, B\neq b$]+1

Conflicts are weighted by
their occurrences in the past

# Maintaining Arc Consistency

- Based on explanations
  - $V_i \neq v_i \Leftarrow (V_1 = v_1 \;\&\; V_2 = v_2 \ldots \& \; V_j = v_j)$
  - When a value is removed from a domain
    - An explanation is attached to the deleted value
  - When a variable is unassigned (e.g., $V_x = v_x$)
    - All deleted values which contain $V_x = v_x$ in their explanations have to be recomputed
  - Computation
    - FC: Explanation corresponds to the violated constraint
    - MAC: Union of explanations

# Dynamic Backtracking with MAC

- A special case of IFS with MAC
  - An unassigned variable is always selected
  - If there is a variable with an empty domain
    - A union of assignments of all values' explanations is computed
    - Fail if the computed union is empty
    - The last assignment from the union is unassigned
    - Explanation: all the other assignments in the computed union
  - If a value $v_x$ is assigned to $V_x$
    - An explanation $V_x \neq v'_x \Leftarrow (V_x = v_x)$ is attached to all values from the variables domain different from $v_x$

# Experiments: Purdue University Timetabling

- Timetabling Problem at Purdue University
  - Central timetable for large lecture classes
    - 826 classes (forming 1782 meetings)
      - some of them (25%) with multiple sections
    - 50 lecture rooms (with various equipment, up to 474 seats)
    - 89,633 course demands from 29,808 students
    - Utilization over 78% (~ 94% for the four largest rooms)
  - Timetables for individual departments
    - Done manually for the moment
      - An area for our future work

Fall 2004

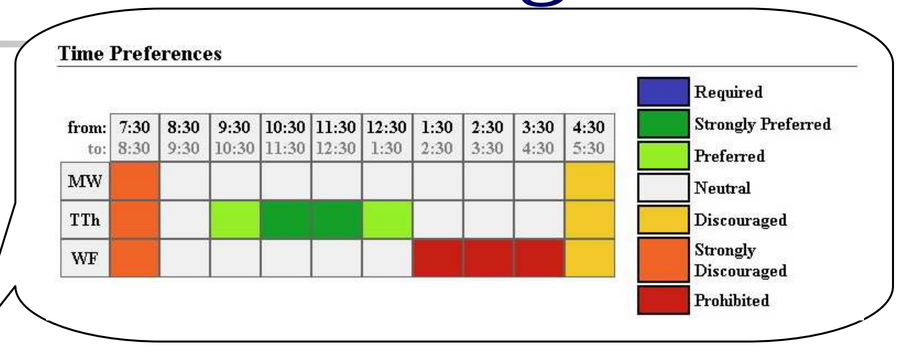
# Experiments: Purdue University Timetabling

- For each class
  - Student requirements
  - Time requirements & preferences
    - Meeting patterns (e.g., 3 x 50 min, 2 x 75 min)
  - Room requirements & preferences
    - Capacity
    - Required equipment
    - Room / building preference
  - Instructor
  - Additional (group) constraints
    - Between several classes (e.g. back-to-back, precedence)
  - Other …

Each student states which courses he or she wants to attend (soft constraint)

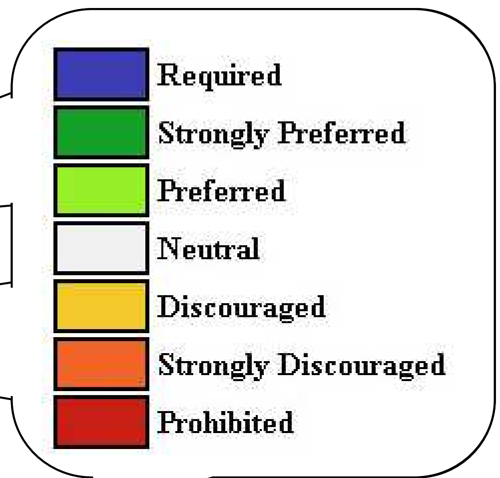
# Experiments: Purdue University Timetabling

- For each class
  - Student requirements
  - Time requirements & preferences
    - Meeting patterns (e.g., 3 x 50 min, 2 x 75 min)
  - Room requirements & preferences
    - Capacity
    - Required equipment
    - Room / building preference
  - Instructor
  - Additional (group) constraints
    - Between several classes (e.g. back-to-back, precedence)
  - Other …



# Experiments:
# Purdue University Timetabling

- For each class
  - Student requirements
  - Time requirements & preferences
    - Meeting patterns (e.g., 3 x 50 min, 2 x 75 min)
  - Room requirements & preferences
    - Capacity
    - Required equipment
    - Room / building preference
  - Instructor
  - Additional (group) constraints
    - Between several classes (e.g. back-to-back, precedence)
  - Other …



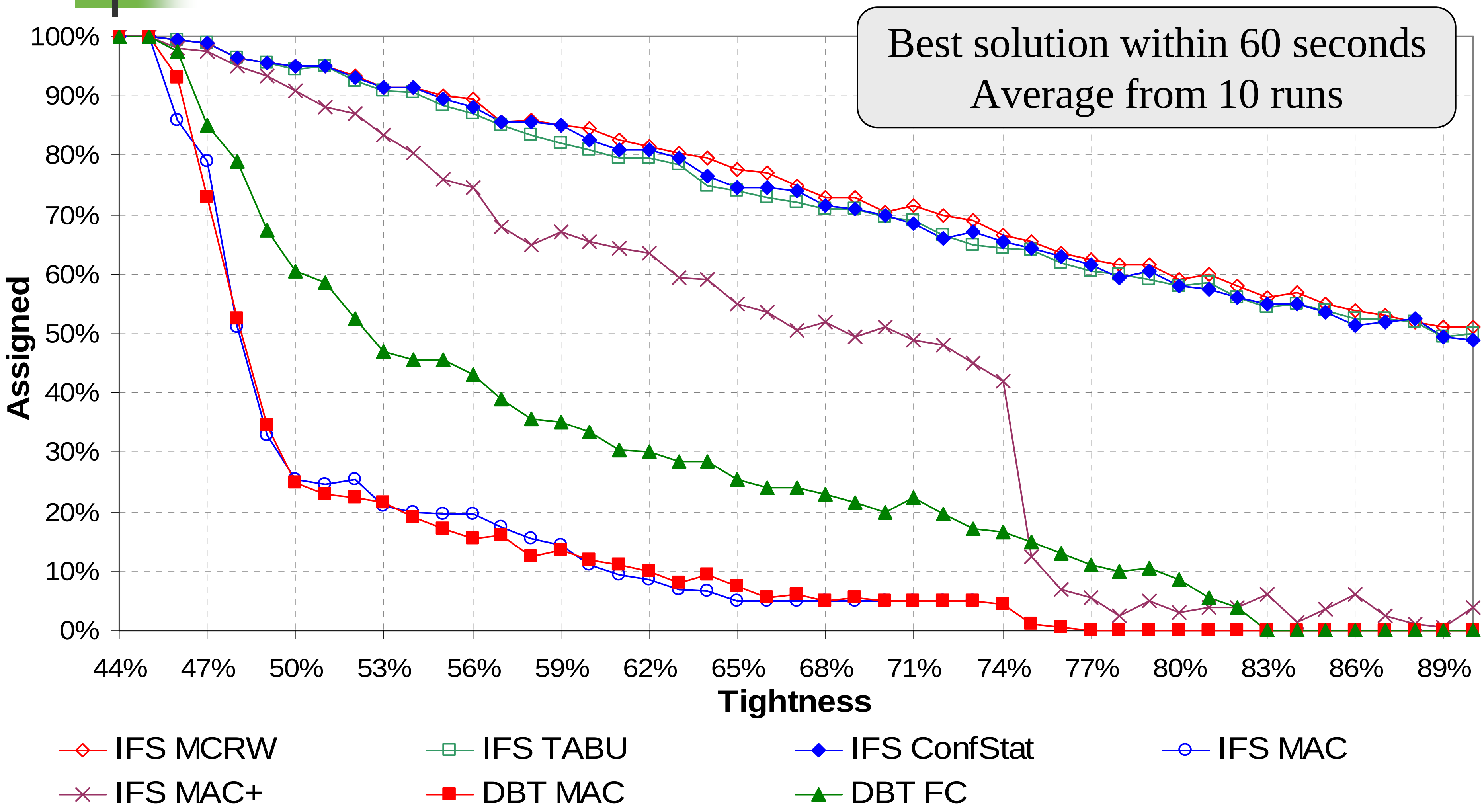
# Experiments: Purdue University Timetabling

| Test Case | IFS ConfStat | IFS TABU | IFS MCRW |
|---|---|---|---|
| Assigned classes [%] | 100.0 ± 0.00 | 97.67 ± 0.15 | 98.29 ± 0.16 |
| Time [min] | 24.11 ± 4.42 | 24.17 ± 3.62 | 24.52 ± 3.83 |
| Student conflicts [%] | 1.97 ± 0.06 | 1.97 ± 0.07 | 2.05 ± 0.19 |
| Preferred time [%] | 85.64 ± 1.57 | 89.86 ± 0.69 | 89.63 ± 1.06 |
| Preferred room [%] | 50.39 ± 5.34 | 66.48 ± 3.42 | 64.84 ± 3.86 |

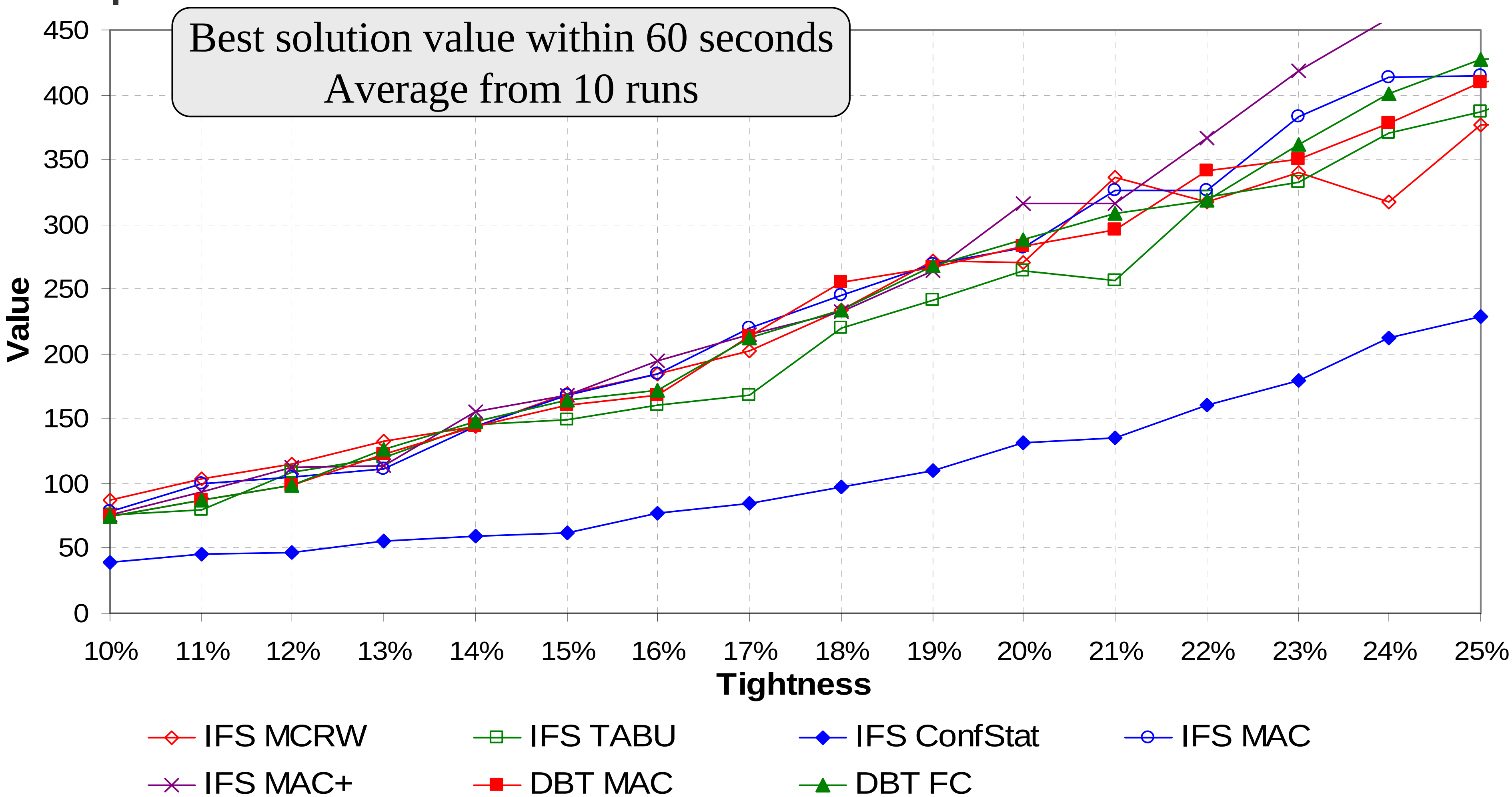
- DBT MAC was able to assign approx. 93% of variables
- IFS MAC was able to assign approx. 94% of variables

Best solution within 30 minutes, 10 runs
1 GHz Pentium III, Java 1.4.2

# Experiments: CSP(20,15,43%,$p_2$)



# Experiments: minCSP(40,30,43%,$p_2$)



# Conclusion And Future Work

- IFS algorithm with conflict-based statistics
  - Good results on Purdue University Timetabling Problem
- Future work
  - More results
    - Timetables for individual departments
    - On other (not only timetabling) problems
  - Solver improvements
  - Additional requirements from Purdue University
  - Application of conflict-based statistics in other search techniques