asap research
automated scheduling optimisation & planning

EPSRC
Engineering and Physical Sciences Research Council

Theory LANCS Practice

The University of Nottingham

Hyper-heuristícas y Optimización Inter-dominio

Gabriela Ochoa
School of Computer Science, The University of Nottingham

Content

- Automated heuristic design
  - Different approaches
  - Motivation
- What are hyper-heuristics?
  - Definition
  - Origins of the term, and the ideas
  - Classification of approaches
- The first Cross-Domain Heuristic Search Challenge
  - Motivation
  - The HyFlex framework
  - Competition features
  - Competition results

Automated heuristic design

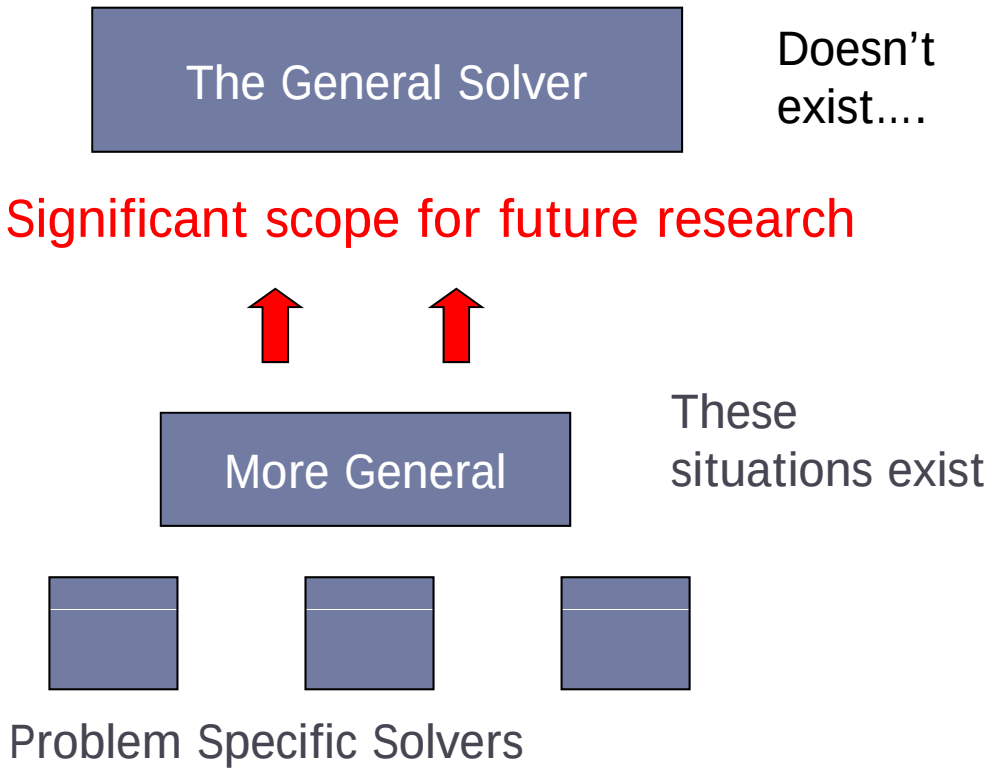
- Search and optimisation problems are everywhere, and search algorithms are getting increasingly powerful
- They are also getting increasingly complex
- Only autonomous self-managed systems that provide high-level abstractions can turn search algorithms into widely used methodologies
- **Research Goals**:
  - Reduce the role of the human expert in the process of designing optimisation algorithms and search heuristics
  - Software systems able to automatically tune, configure, generate and design optimisation algorithms and search heuristics.
  - Self-tuning, self-configuring and self-generating search heuristics (**Self-* Search**) → GECCO 2011, 2012 Track

Automated heuristic design: several approaches

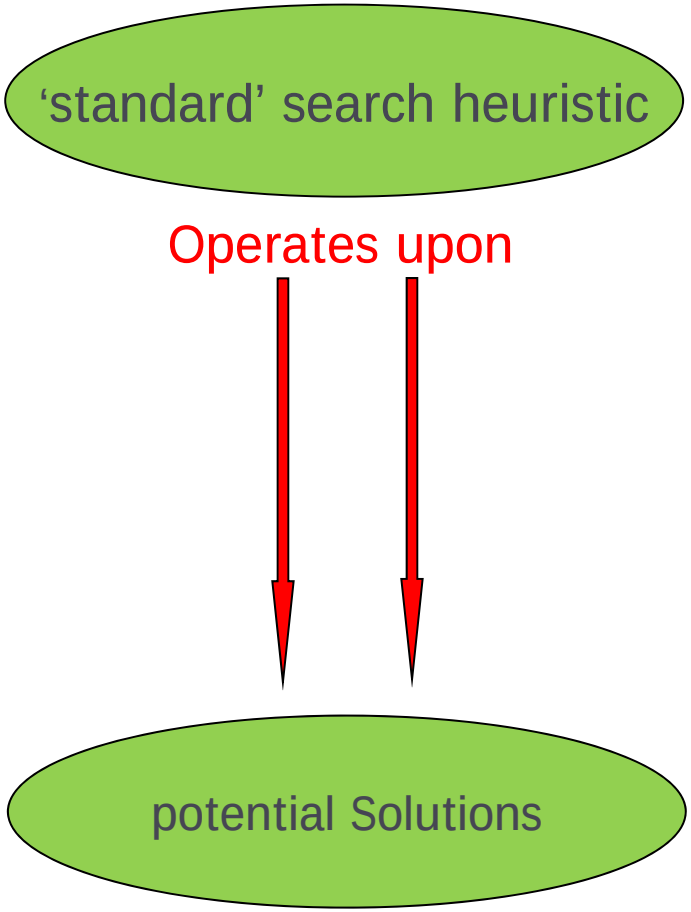
- ***Online* approaches**
  - Self-tuning and self-adapting heuristics on the fly, effectively learning by doing until a solution is found
  - **Examples:** adaptive memetic algorithms, adaptive operator selection, parameter control in evolutionary algorithms, adaptive and self-adaptive search algorithms, reactive search
- ***Offline* approaches**
  - Learn, from a set of training instances, a method that would generalise to unseen instances
  - **Examples:** automated algorithm configuration, meta-learning, performance prediction, experimental methods, SPO
- **Hyper-heuristics (offline and online)**

Motivation

- Decision support systems
  - *Off the peg vs. made to measure*
  - *Ford Model T vs. Ferraris*
- Develop the ability to automatically work well on different problems
- How general we could make hyper-heuristics ? (*no free lunch* theorem)

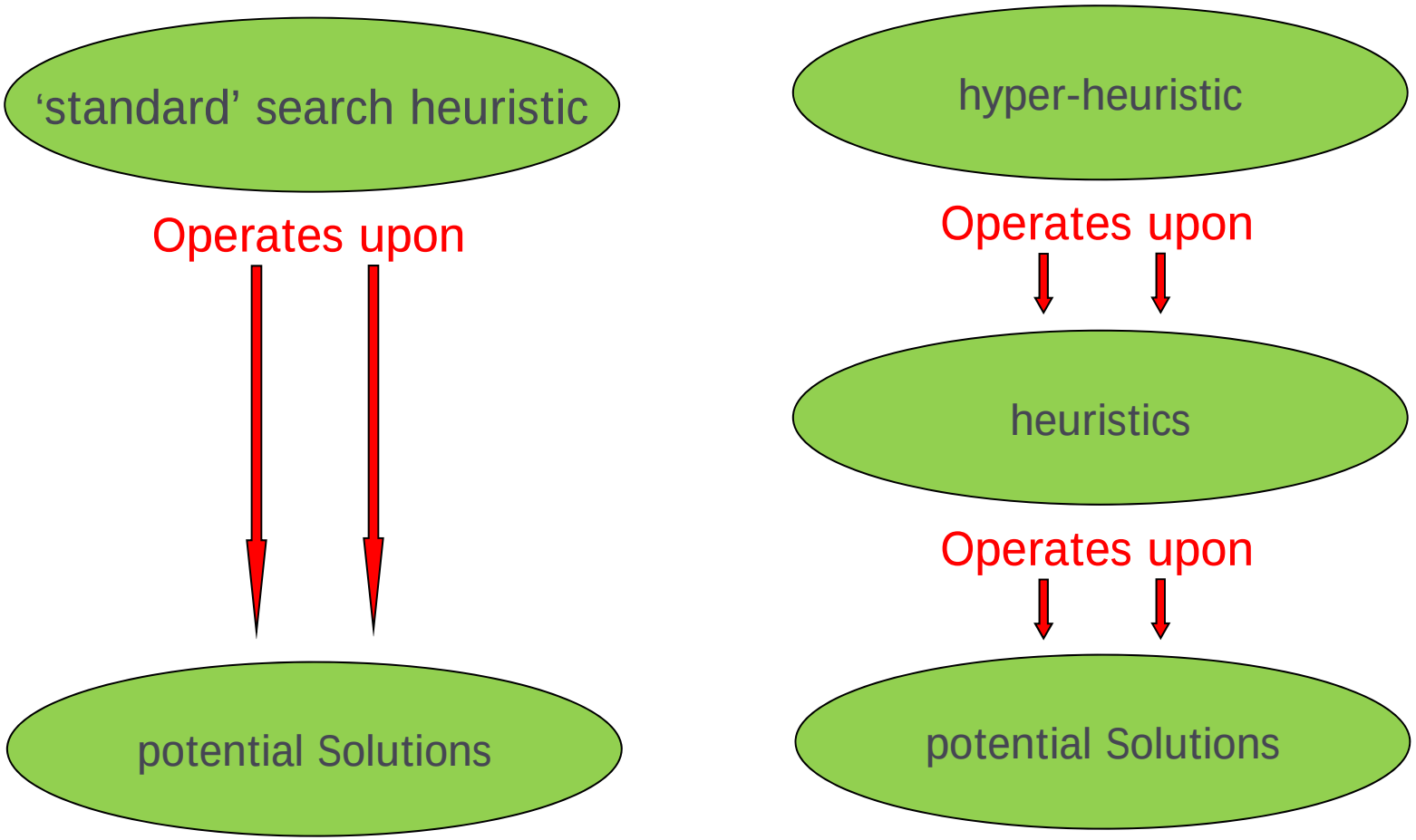


What is a hyper-heuristic?



Hyper-heuristics:

"Operate on a search space of heuristics"



What is a hyper-heuristic?

- Initially 'heuristics to choose heuristics'
- Recent research trend in hyper-heuristics
  - Automatically *generate* new heuristics suited to a given problem or class of problems
  - Combining, i.e. by GP, *components* or *building-blocks* of human designed heuristics
- New definition:

A hyper-heuristic is an automated methodology for selecting or generating heuristics to solve hard computational search problems

E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J. Woodward (2009). A Classification of Hyper-heuristics Approaches, *Handbook of Metaheuristics,* International Series in Operations Research & Management Science, M. Gendreau and J-Y Potvin (Eds.), Springer, pp.449-468.

What is a hyper-heuristic?

- Useful criteria for defining hyper-heuristics.
  A hyper-heuristic :
  1. Is a *higher level* heuristic which manages a *set* of l*ow-level* heuristics
  2. Searches for a good method to solve the problem rather than for a good solution
  3. Uses only limited problem-specific information

Chakhlevitch K, Cowling PI (2008) Hyperheuristics: Recent developments. In: Cotta C, Sevaux M, SÄorensen K (eds) Adaptive and Multilevel Metaheuristics, Studies in Computational Intelligence, vol 136, Springer, pp 3-29

The term hyper-heuristics

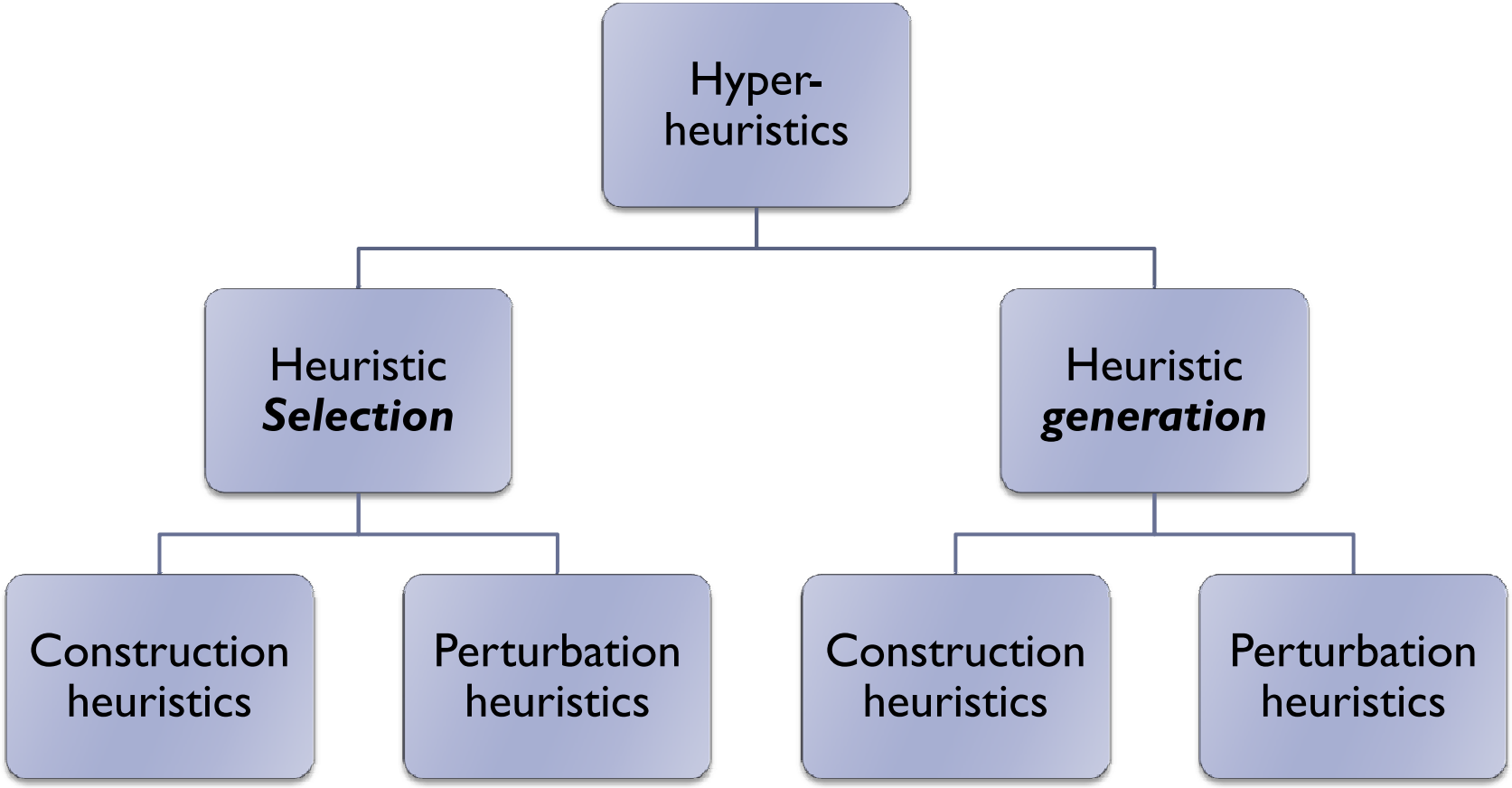
- First used in 2000 : 'heuristic to choose heuristics' in combinatorial optimisation
  - Cowling P.I., Kendall G. and Soubeiga E. (2001) A Hyperheuristic Approach to Scheduling a Sales Summit, Selected papers from the 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT 2000), Springer LNCS 2079, 176-190
- First journal paper to use the term published in 2003
  - Burke EK, Kendall G, Soubeiga E (2003b) A tabu-search hyperheuristic for timetabling, and rostering. *Journal of Heuristics* 9(6):451-470
- A claim in the Wikipidia page.
  - First used in 1997: Denzinger J, Fuchs M, Fuchs M (1997) High performance ATP systems by combining several ai methods. In: Proc. 15th *International Joint Conference on Artificial Intelligence (IJCAI 97)*, pp 102-107
  - Turns out not true: the term appears in an unpublished technical report, with the same title: Denzinger J, Fuchs M, Fuchs M (1996) High performance ATP systems by combining several ai methods. *Tech. Rep. SEKI-Report SR-96-09,* University of Kaiserslautern

Origins and early approaches

The ideas can be traced back to the 60s and 70s

- Automated heuristic sequencing (early 60s and 90s)
  - Fisher H, Thompson GL (1963) Probabilistic learning combinations of local job-shop scheduling rules. In: Muth JF, Thompson GL (eds) *Industrial Scheduling*, Prentice-Hall, Inc, New Jersey, pp 225-251
- Automated planning systems (90s)
  - Gratch J, Chien S (1996) Adaptive problem-solving for large-scale scheduling problems: a case study. Journal of *Artificial Intelligence Research* 4:365-396
- Automated parameter control in EAs (70s, 80s)
  - (Rechenberg, 1973), (Davis, 1989), (Grefenstette, 1986)
- Automated learning of heuristic methods (90s)
  - Minton S (1996) Automatically confguring constraint satisfaction problems: a case study. Constraints 1(1):7-43

Classification of hyper-heuristics
(*nature of the search space*)



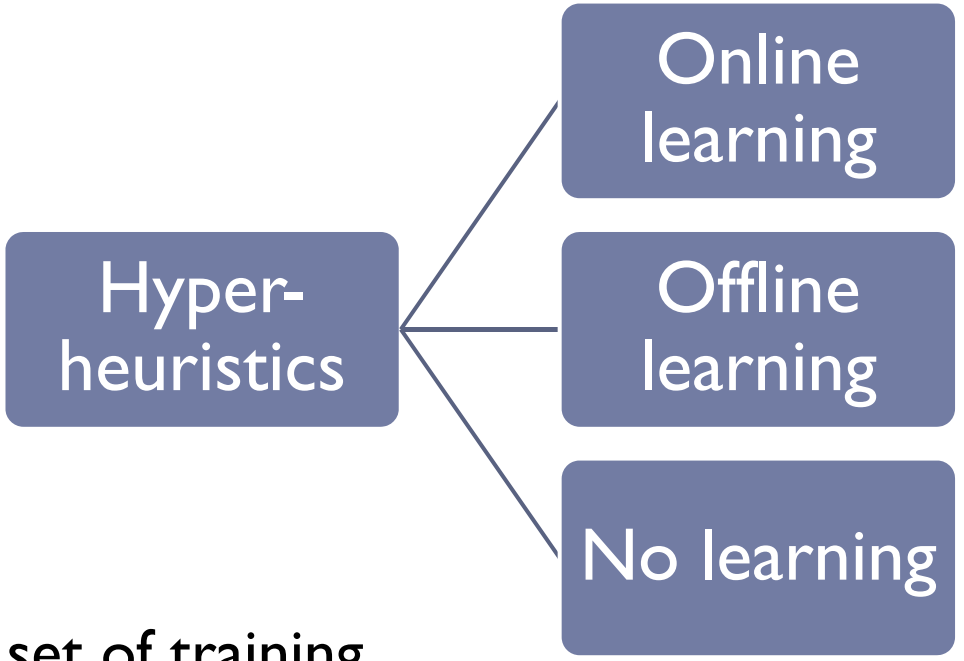
Classification of hyper-heuristics (*source of feedback during learning*)

Online

- Learning while solving a single instance
- Adapt
- Examples: reinforcement learning, meta-heuristics

Offline

- Gather knowledge from a set of training instances
- Generalise
- Examples: classifier systems, case-based, GP



Conclusions (hyper-heuristics)

A hyper-heuristic is an automated methodology for selecting or generating heuristics to solve hard computational search problems

- Main features:
  - Search in a space of heuristics
  - A higher level strategy that manages a *set* or lower-level heuristics
  - Use limited problem-specific information
- Term used for the first time 2000: '*heuristics to choose heuristics*'
- Ideas can be traced back to the 60s and 70s
- Two main type of approaches: selection and generation
- Ideas from online and offline machine learning are relevant, as are ideas of meta-level search

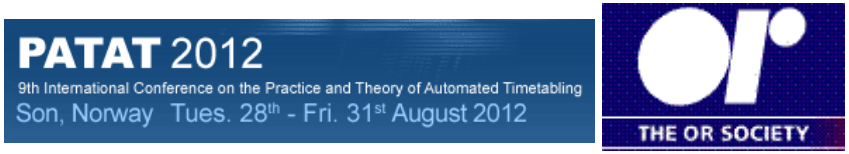
Future work

- Generalisation: By far the biggest challenge is to develop methodologies that work well across several domains
- Foundational studies: Thus far, little progress has been made to enhance our understanding of hyper-heuristic approaches
- Distributed, agent-based and cooperative approaches: Since different low-level heuristics have different strengths and weakness, cooperation can allow synergies between them
- Multi-criteria, multi-objective and dynamic problems: So far, hyper-heuristics have been mainly applied to single objective and static problems

asap research
automated scheduling optimisation & planning

Sponsors:

The University of Nottingham



The First Cross-Domain Heuristic Search Challenge

Motivation

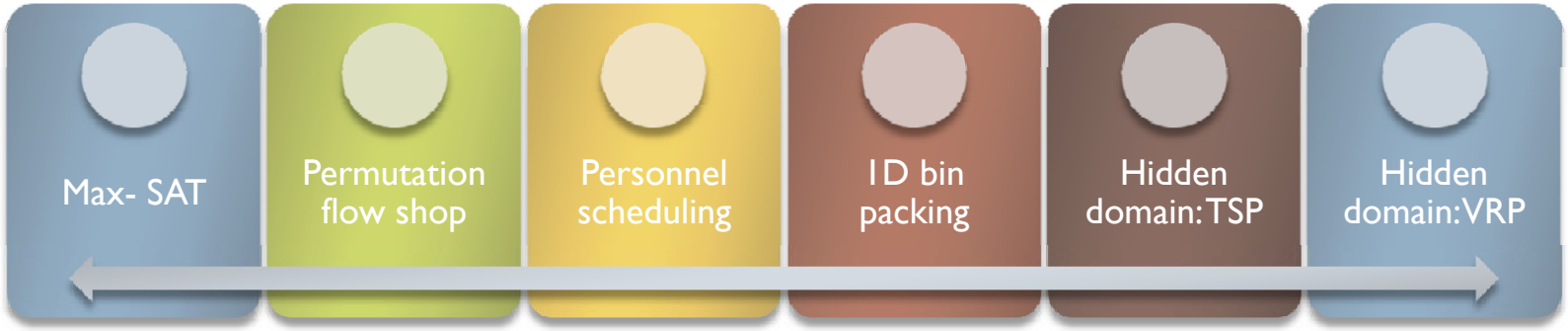
- Automating the design of heuristic search methods: active area of research in CS, AI and OR
- Challenge: raise the level of generality
- Researchers are often constrained on the number of problem domains on which to test their adaptive methods
- This is explained by the inherent difficulty of implementing each of this problem domains

HyFlex (Hyper-heuristics Flexible framework)

- **Question**: Can we produce a benchmark to test the generality of heuristic search algorithms?

- A software framework (problem library) for designing and evaluating general-purpose search algorithms
- Provides the *problem-specific* components
- Efforts focused on designing high-level strategies

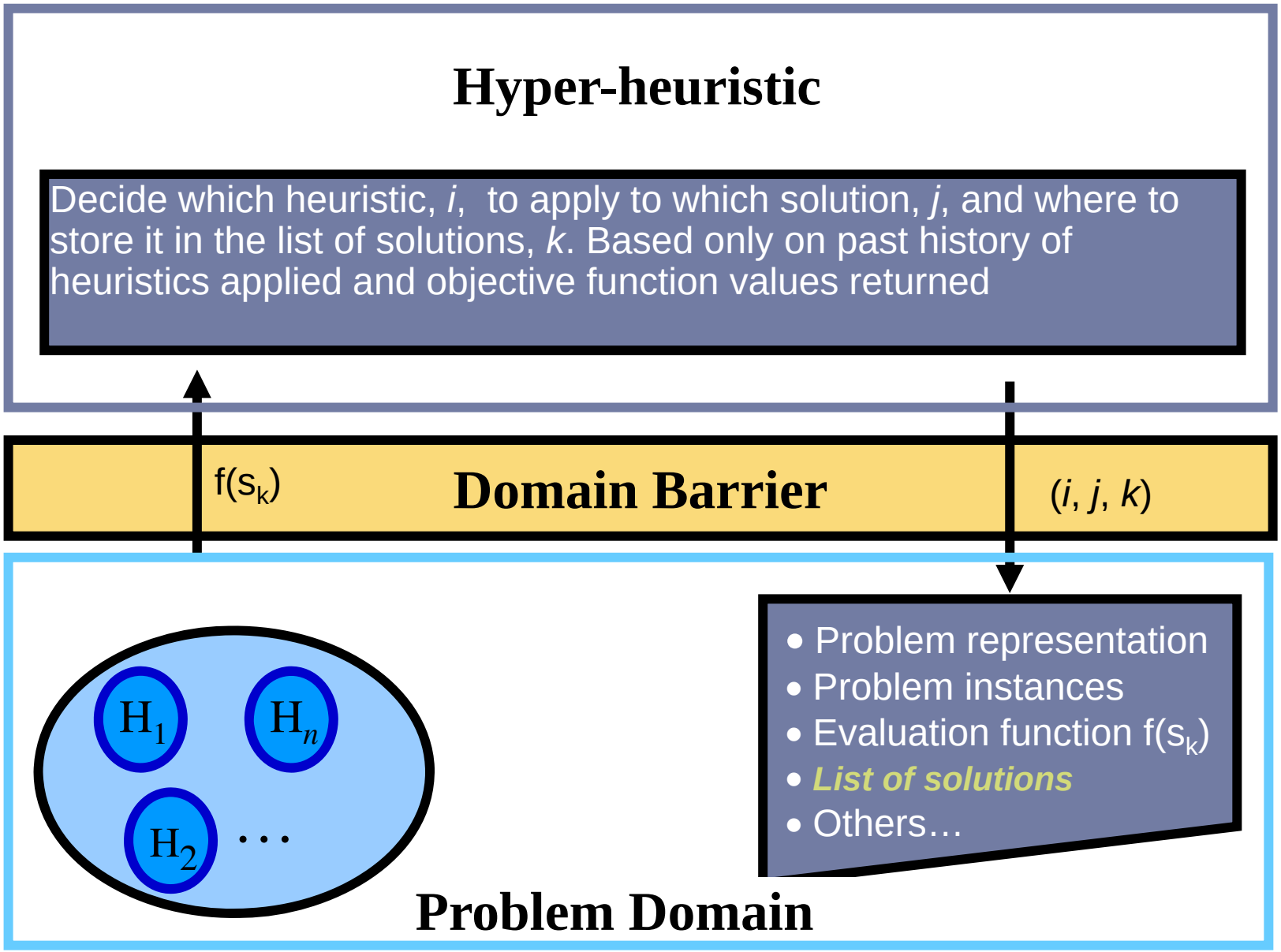
HyFlex: a Benchmark for Cross-domain Heuristic Search



- Six different domains, hard combinatorial problems, interesting and varied set of operators and instances
- Implemented using a common software interface
- A single high-level strategy can operate and solve all the domains

Problem Domains: (problem-specific)
Max-SAT, Flowshop, Personnel sched, ...

Hyper-heuristics (general-purpose)
Competing algorithms

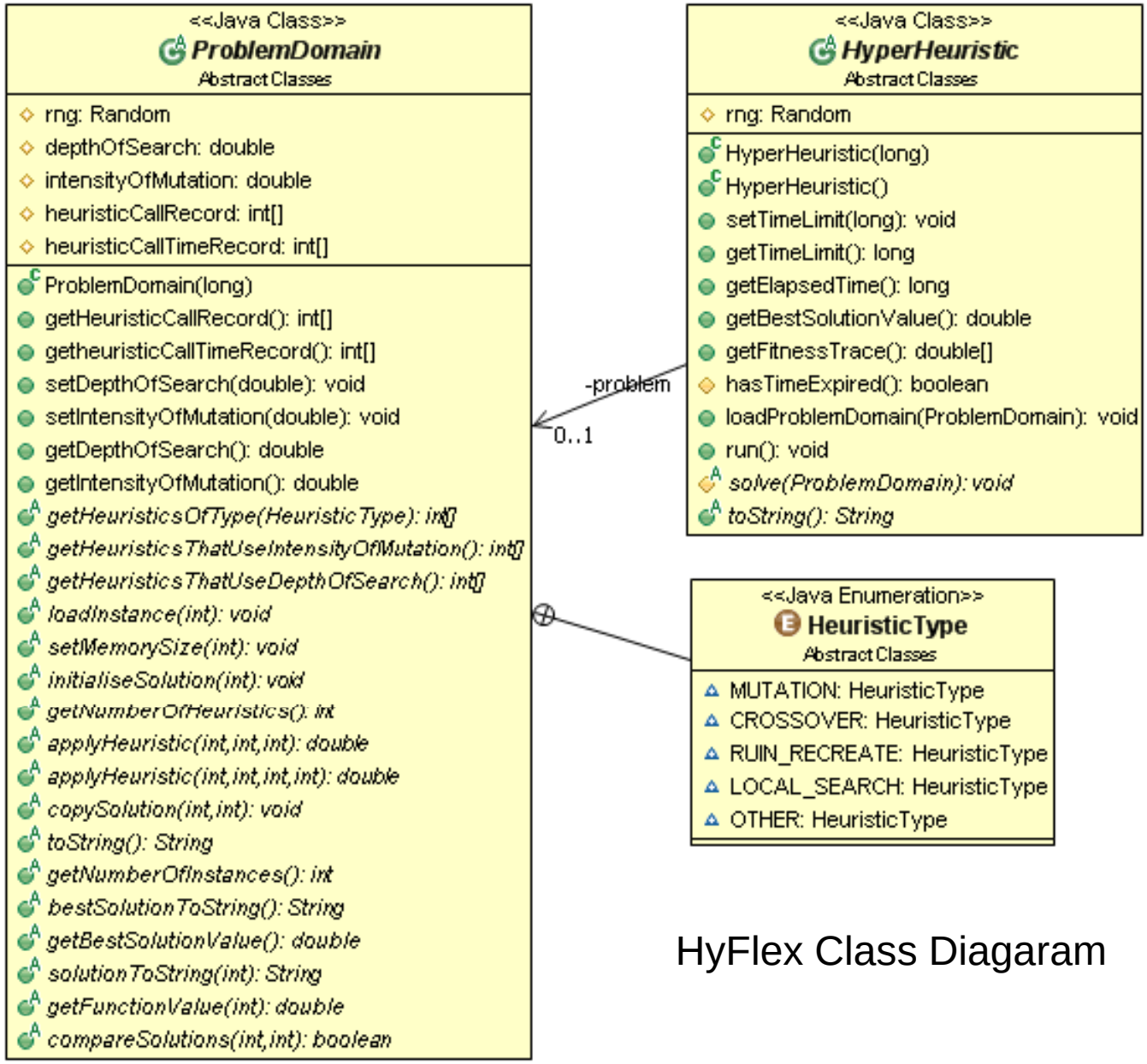


HH fremework:(Cowling P., Kendall G. and Soubeiga, 2000, 2001), (E. K. Burke et al., 2003)

Extension: J. Woodward, A. J. Parkes, G. Ochoa, A Mathematical Framework for Hyper-heuristics. PPSN Hyper-heuristics Workshop. 2008

Overview of the problem domain modules

1. A routine to initialise (randomised) solutions
2. A set of interesting instances, that can be easily loaded
3. A population or list of solutions
4. An objective function
5. A set of heuristics to modify solutions
   a. Mutational: makes a random modification
   b. Ruin-recreate: partially destroy a solution and rebuild it using a constructive procedure
   c. Local-search: iterative procedures searching on the neighbourhood of solutions
   d. Crossover: takes parent solutions and produce offspring solution
6. Two parameters: *IoM*, *DoS* that modify some heuristics



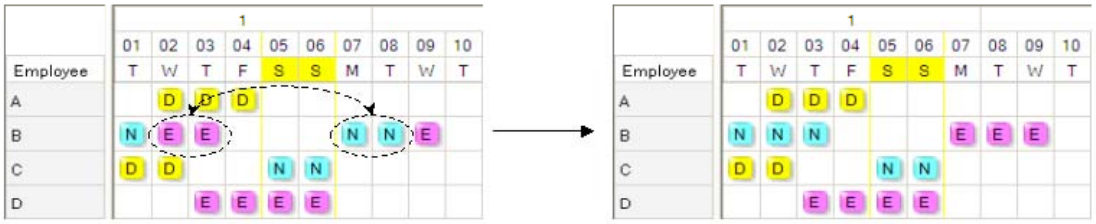
HyFlex Class Diagaram

Personnel scheduling

Instances: Wide range of data sets (Industry, Academia, +10 countries)

BCV-A.12.1	1294	12	5	31	[2, 7]
BCV-A.12.2	1953	12	5	31	[2, 7]
ORTEC01	**270**	16	4	31	[4]
ORTEC02	290	16	4	31	[4]
GPost	**5**	8	2	28	
GPost-B	**3**	8	2	28	
QMC-1	16	19	3	28	
QMC-2	29	19	3	28	
Ikegami-2Shift-DATA1	**0**	28	2	30	[9]
Ikegami-3Shift-DATA1	6	25	3	30	[9]
Ikegami-3Shift-DATA1.1	13	25	3	30	[9]
Ikegami-3Shift-DATA1.2	12	25	3	30	[9]
Millar-2Shift-DATA1	**0**	8	2	14	[9]
Millar-2Shift-DATA1.1	**0**	8	2	14	[9]
Valouxis-1	**20**	16	3	28	[13]

Low level heuristics: 12, different types. LS based on new, horizontal and vertical moves



Horizontal swap: move shifts in single employee's work pattern

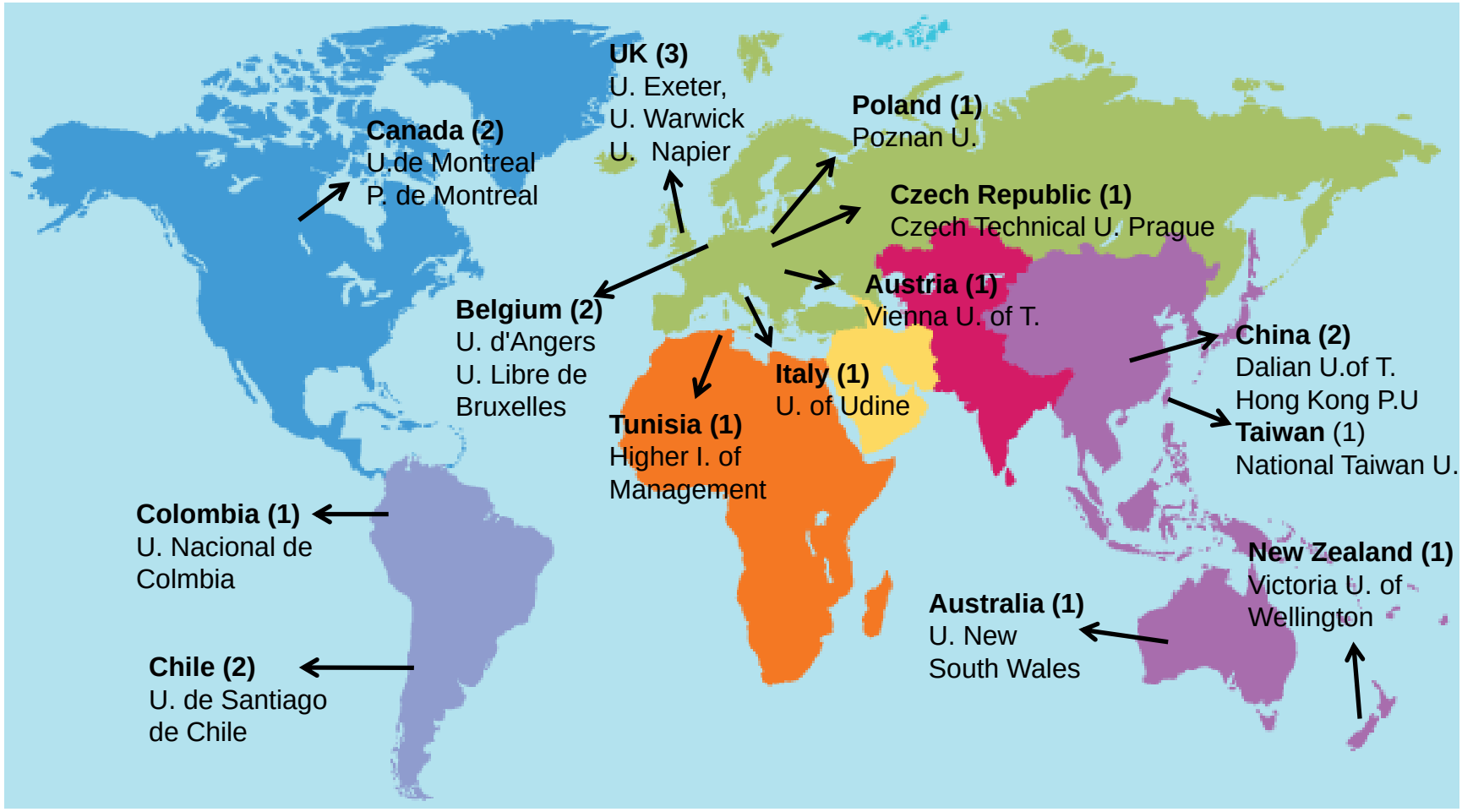
The Cross-Domain Challenge

- Conducted a competition (cross-domain challenge):
  - Using HyFlex
  - **Winners**: algorithms with best overall performance across all of the different domains
  - The ***Decathlon Challenge*** of search heuristics

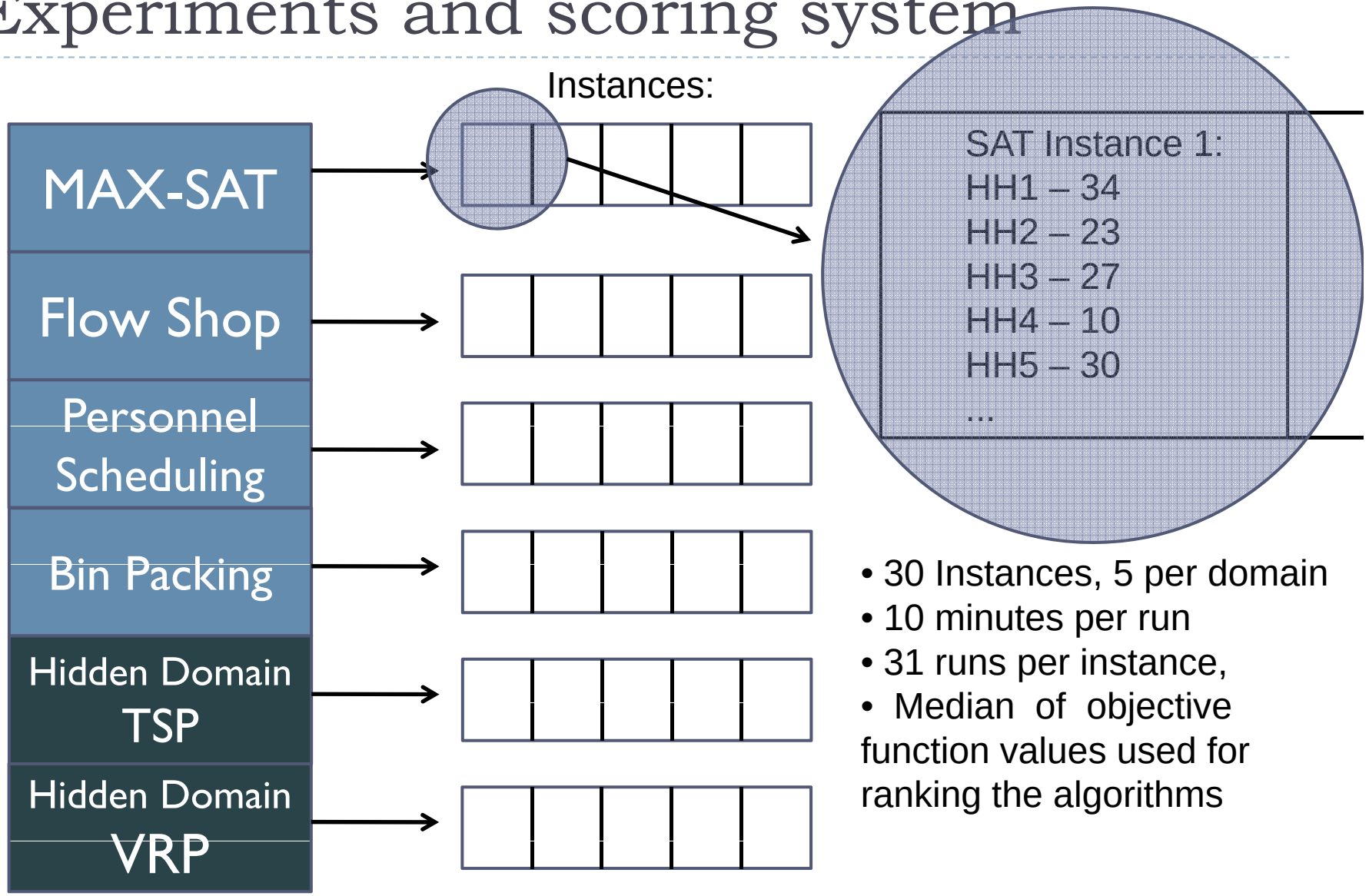
- Why run a competition?
  - Competitions appear to help advance research
  - Successful examples: Timetabling, Nurse Rostering, Planning, SAT, CSP, RoboCop, ...
  - Bring together researchers from sub-fields of CS, AI and OR
  - Achieve a deeper understanding of the design principles of hyper-heuristics across a wide set of problems

Competition entries

Registered participants: 43 (23 countries), Competition entries: 20 (14 countries)



Experiments and scoring system

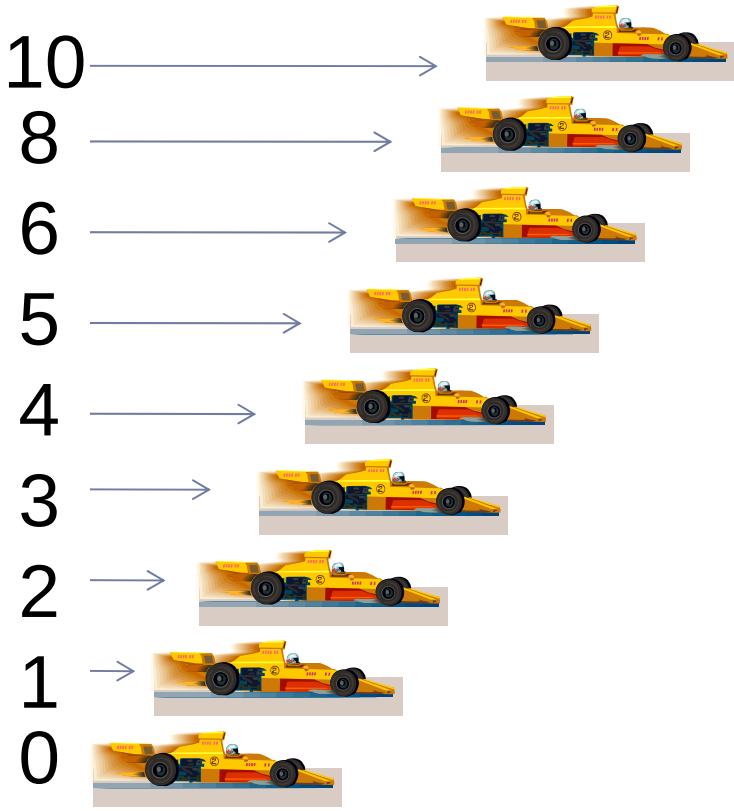


- 30 Instances, 5 per domain
- 10 minutes per run
- 31 runs per instance,
- Median of objective function values used for ranking the algorithms

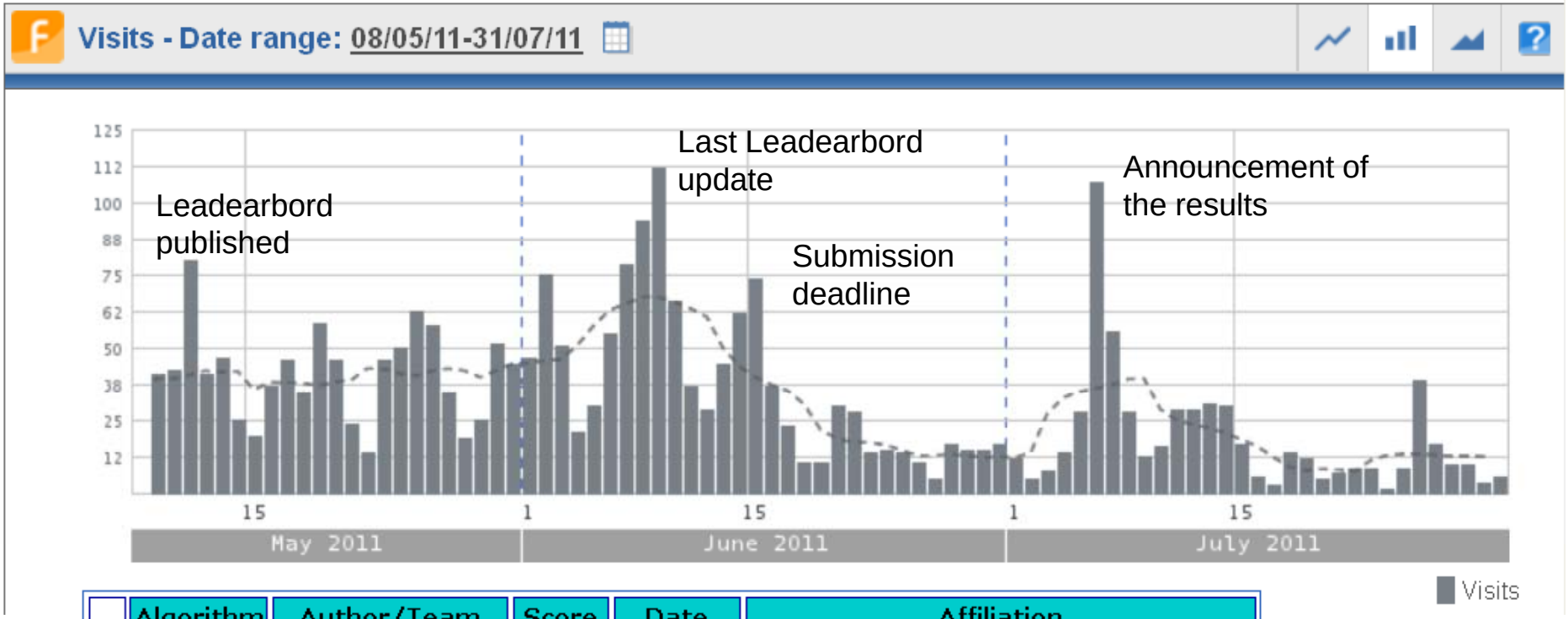
Scoring system

Formula 1

- For each instance (race): algorithms were ranked by their median objective function value (31 runs)
- The top eight 'drivers' score points
- Ties: Points to the relevant positions added and shared equally

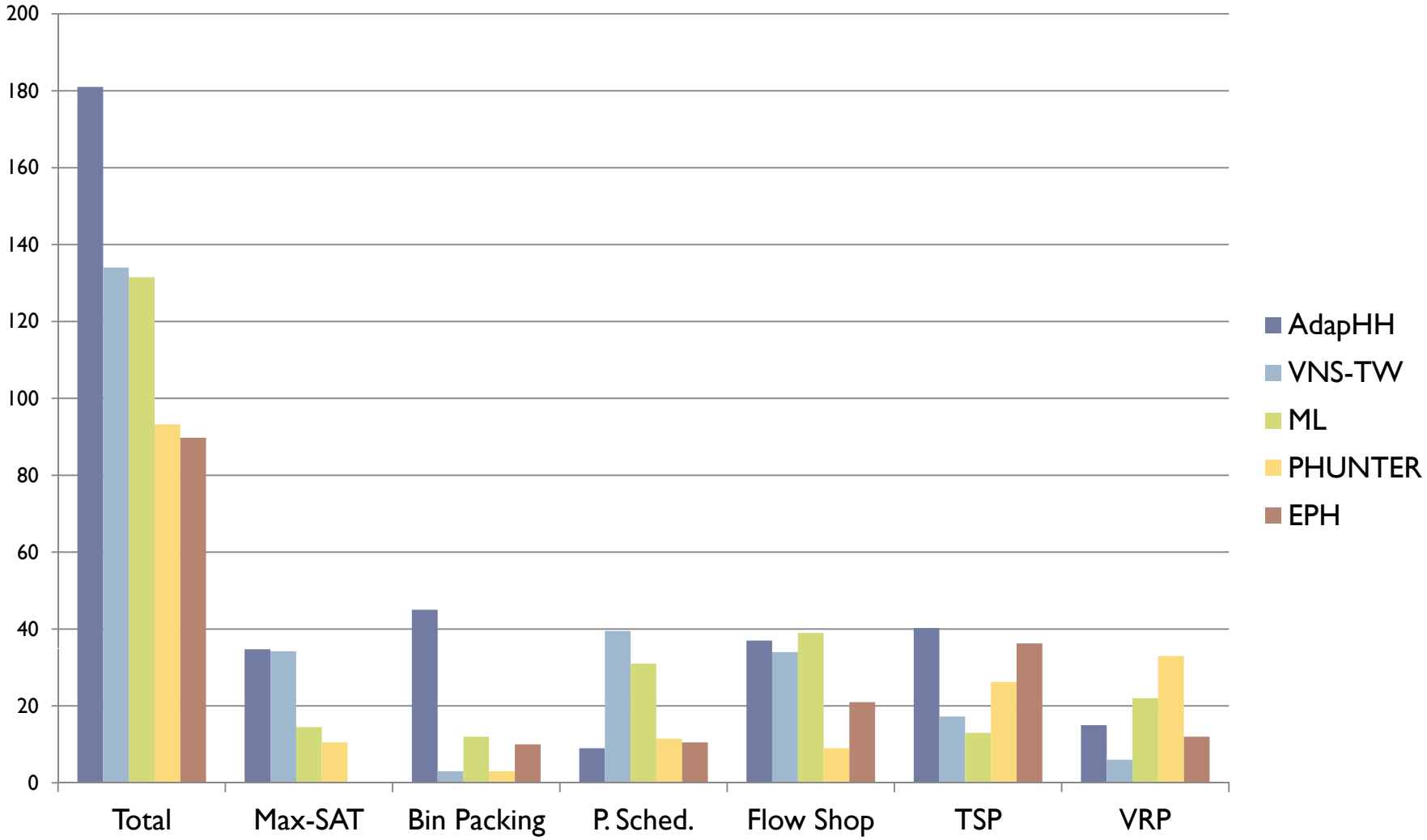


Leaderboard and web statistics

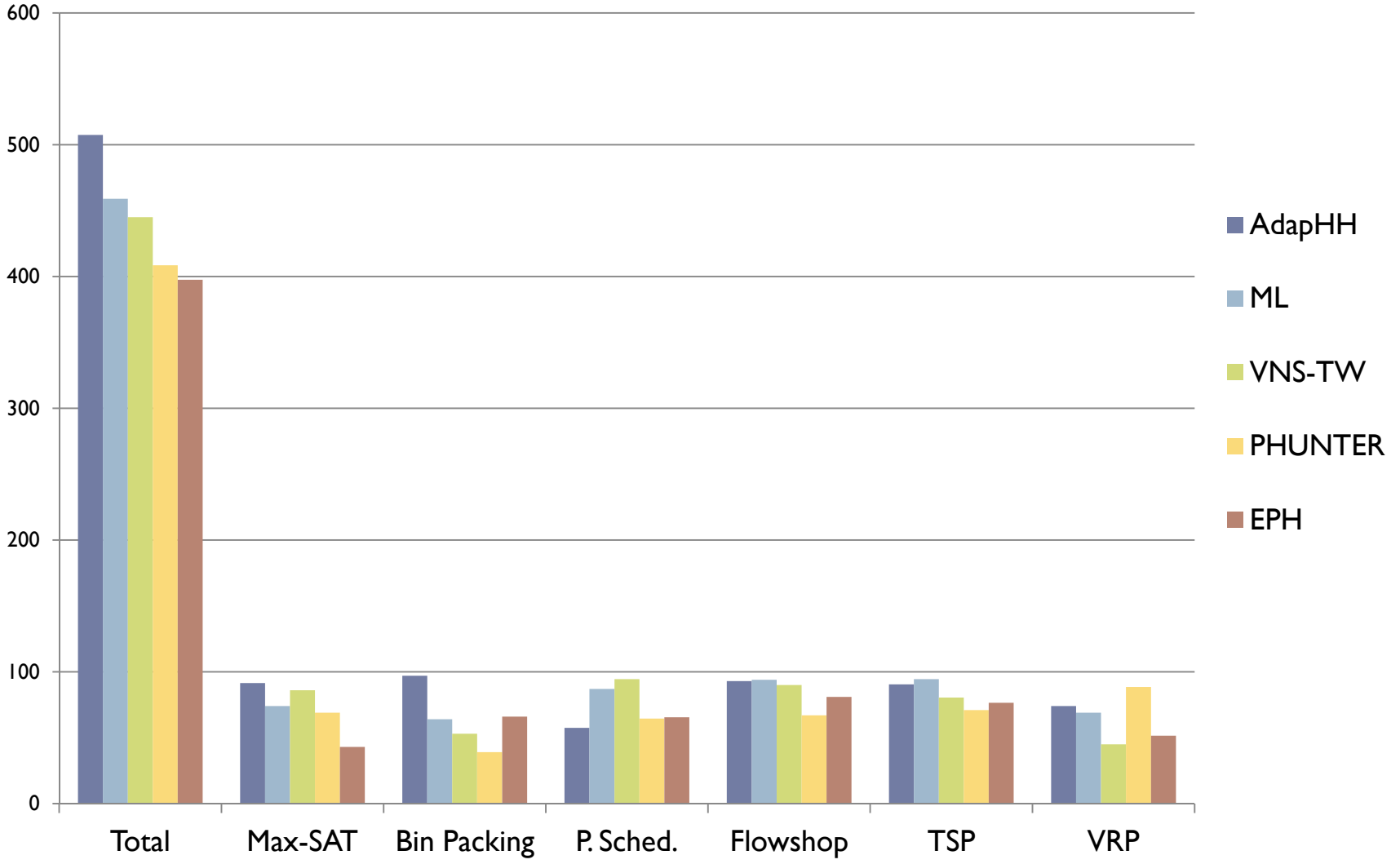


	Algorithm	Author/Team	Score	Date	Affiliation
1	PHunter4	Fan Xue	204.28	07/06/11	Hong Kong Polytechnic U., Hong Kong
2	ISEA2	Jiri Kubalik	177.28	25/05/11	Czech Technical University, Czech Rep.
3	HAHA1	Andreas Lehrbaum	166.78	28/04/11	Vienna University of Technology, Austria
4	TW4	Hsiao Ping-Che	130.20	07/06/11	National Taiwan University, Taiwan
5	basic-test	Franco Mascia	125.67	28/05/11	Université Libre de Bruxelles, Belgium
6	ADHS1	Mustafa Misir	120.08	06/05/11	KaHo Sint-Lieven, Belgium

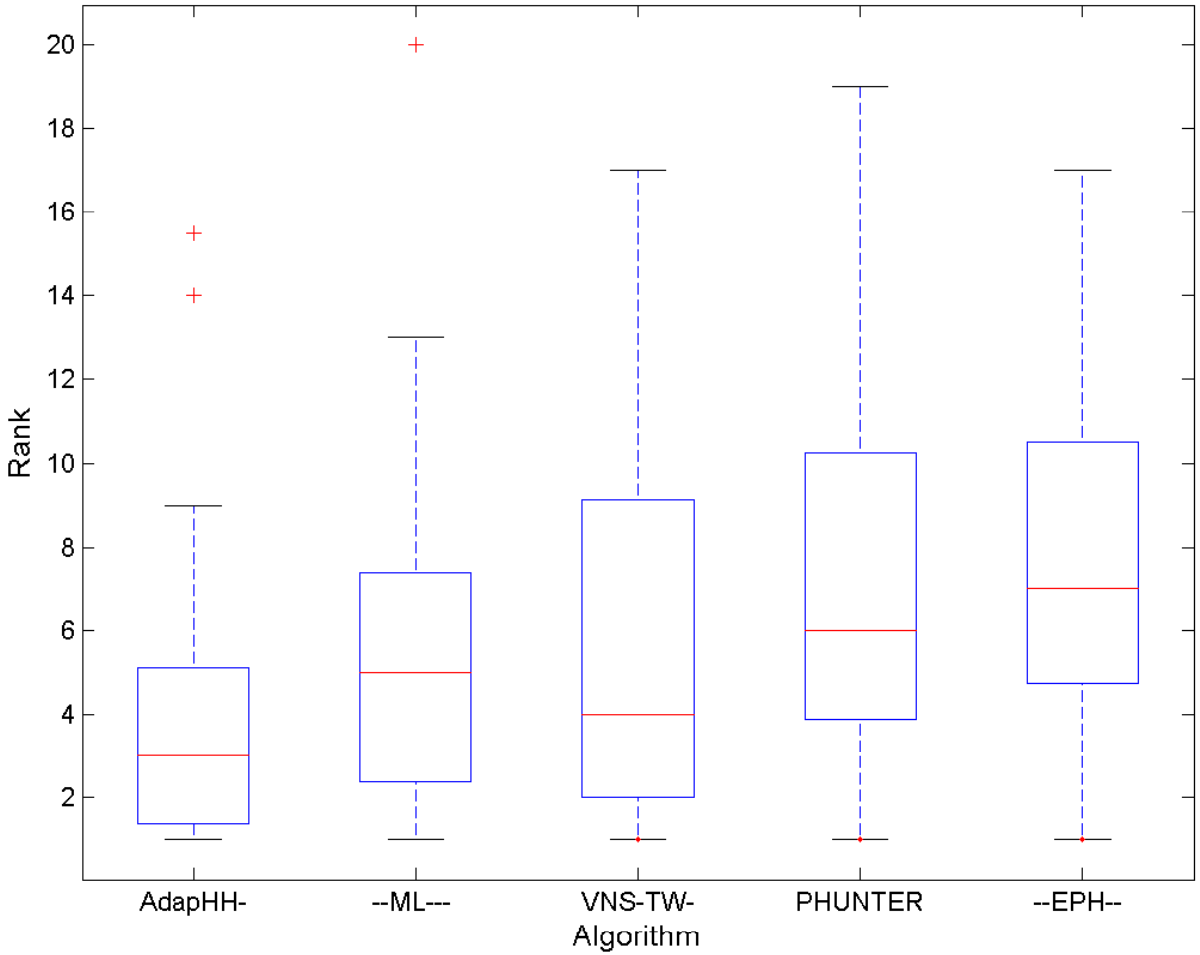
Results – Top 5: Formula 1 score



Results – Top 5: Borda score/ranking



Results – Top 5: distribution of ranks



30 Instances
20 Algorithms

The competition winner: AdapHH

Mustafa Misir, KaHo St.-Lieven, Gent, Belgium

- Adaptive dynamic heuristic set: a performance metric for each heuristic that considers improvement capability and speed. Heuristics not performing well, are dynamically excluded. Memory of performance is kept for long and short term.
- Rely hybridisation: Learning mechanism to determine effective pairs of heuristics that are applied consecutively.
- Adaptation of heuristic parameters: reward-penalty strategy to dynamically adapt *DoS* and *IoM* parameters
- Adaptive iteration limited list-based threshold acceptance: a mechanism determining the threshold in a dynamic manner using the fitness of previous new best solutions

The 2nd and 3rd Places

VNS-TW

Hsiao Ping-Che, National Taiwan University, Taiwan

- VNS: Order the perturbation heuristics according to strength.
- Two stages: shaking (M+RR) and local search
- Adaptive mechanism for adjusting the *DoS* param.
- Use a population

ML

Mathieu Larose, Université de Montréal, Canada

- Adaptive ILS: diversification (M+RR) + intensification (LS)
- Reinforcement learning for selecting diversification and intensification heuristic
- Simple adaptive acceptance criteria

The 4th and 5th Places

HUNTER

Fan Xue, Hong Kong Polyt. U., Hong Kong

- Diversification (surface and change target area – M+RR), intensification (dive and find pearl oysters – LS)
- Two forms of dives: snorkelling and deep dive (low and high *DoS*).
- Offline learning to identify search modes

EPH

David Meignan, Polyt. Montréal, Canada

- Co-evolutionary approach: pop. of heuristic seq. + pop. of solutions.
- Solutions accepted according to obj. value and diversity
- Sequence of heuristics: diversification (M+RR+C), intensification(LS, fixed all)

HyFlex as a research tool

"Civilization advances by extending the number of important operations which we can perform without thinking about them."

Alfred North Whitehead, *Introduction to Mathematics (1911)*

"Nothing is impossible for the man who doesn't have to do it himself."

- A. H. Weiler

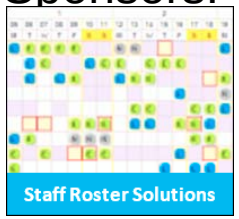
Crowdsourcing: the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call.

Jeff Howe, Wired Magazine, 2006

asap research
automated scheduling optimisation & planning

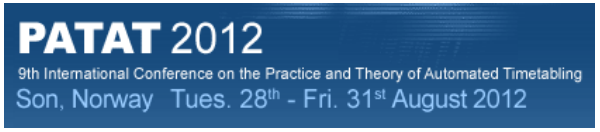
http://www.asap.cs.nott.ac.uk/chesc2011

Sponsors:



WATT

Organising partners:



- Project PI: Edmund K. Burke
- Concept: Gabriela Ochoa, Andrew Parkes
- Organising Committee: Matthew Hyde and Gabriela Ochoa
- HyFlex Design and Implementation: Tim Curtois, Michel Gendreau, Matthew Hyde, Gabriela Ochoa, J. A. Vazquez-Rodriguez, James Walker
- Advisory Board: Edmund K. Burke, Michel Gendreau, Graham Kendall, Barry McCollum, Ender Ozcan, Andrew Parkes and Sanja Petrovic