Hyper-heuristics and Cross-domain Optimization

Gabriela Ochoa
Computing Science and Mathematics,
University of Stirling, Scotland
Gabriela.ochoa@cs.stir.ac.uk

http://www.cs.stir.ac.uk/~goc/
http://www.sigevo.org/gecco-2012/

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

Instructor/Presenter

- **Gabriela Ochoa** is a transitional senior research fellow at the University of Stirling, Scotland UK. She was for six years a researcher at the University of Nottingham, UK. She holds BSc and MRes degrees in Computer Science from the University Simon Bolivar, Venezuela; and a PhD in Artificial intelligence from the University of Sussex, UK. Her research interests lie in the foundations and application of evolutionary algorithms and heuristic search methods with emphasis in automated heuristic design, self-* search heuristics, hyper-heuristics and fitness landscape analysis. Among her contributions are the use of L-systems as a representation, the study of error thresholds and the role of mate selection in evolutionary algorithms; the conception of the local optima network model of combinatorial landscapes; the definition and classification of hyper-heuristics and the conception of the HyFlex hyper-heuristic framework. She is an associate editor of the Journal of Evolutionary Computation (MIT Press) and proposed and co-organised the first "Cross-domain Heuristic Search Challenge" (CHeSC 2011), a an international research competition in hyper-heuristics and adaptive heuristic search.

Content

- Part I
  - Introduction and background
  - Hyper-heuristics
- Part II
  - The HyFlex (Hyper-heuristic Flexible) framework
  - The first *Cross-Domain Heuristic Search Challenge*

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

Part I

- Search and optimisation in practice
  - Increase in complexity in problems and algorithms
  - Algorithm design and tuning
  - Learning and optimisation
- Hyper-heuristics
  - Definition
  - Origins and early approaches
  - Classification of approaches
  - Selection hyper-heuristics
  - Summary and future work

Search and optimisation in practice

Many challenging applications in science and industry can be formulated as optimisation problems!

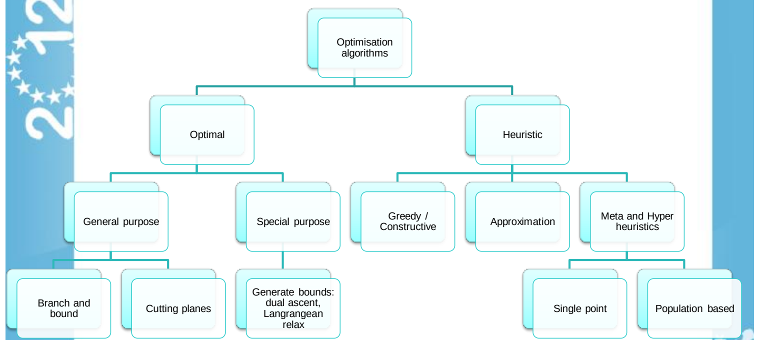
Real-world problem → Formulation → Model → Algorithm → Numerical solution

Mathematical Model
- Decision variables
- Constraints
- An objective function

Optimisation Algorithm
- Mathematical programming
- Heuristic search methods

Solution to the Model
- Set of variable values which are feasible
- Lead to the optimal (or good enough) value of the objective function

Optimisation algorithms



Increase in complexity

- Real world problems are complex
- Heuristic search algorithms are powerful, but they're getting increasingly complex
  - Many parameters
  - Many heuristics or components
- Advantage
  - More flexible algorithms
  - Fit to different problems
- Disadvantage
  - Need to set the parameters, or
  - Select the heuristics, search operators or other components

Algorithm design and tuning

Questions:
- How to set the values of the numerical parameters?
- How to choose the suitable operator at each iteration?

Currently, must of the work is done by the human designer (trial and error, experience)

Can we automate this process?

Learning and optimisation

- ❖ ***Online* learning 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, hyper-heuristics
- ❖ ***Offline* learning 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, Sequential Parameter Optimization (SPO), hyper-heuristics

What is a hyper-heuristic?

'standard' search heuristic

Operates upon

potential Solutions

Hyper-heuristics:

"Operate on a search space of heuristics"

'standard' search heuristic

Operates upon

potential Solutions

hyper-heuristic

Operates upon

heuristics

Operates upon

potential Solutions

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 E, K, Kendall G, Soubeiga E (2003) A tabu-search hyperheuristic for timetabling, and rostering. *Journal of Heuristics*,9(6):451-470

The term hyper-heuristics

- 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

- Automated heuristic sequencing (early 60s and 90s)
  - Fisher H, Thompson GL (1963) Probabilistic learning combinations of local job-shop scheduling rules, *Industrial Scheduling*, Prentice-Hall, Inc, New Jersey, pp 225-251.
  - Storer, R.H., Wu, S.D and Vaccari, R (1992) New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling, Management Science, Vol 38 No 10, 1495-1509.
  - H-L Fang, P.M.Ross and D.Corne (1994) A Promising Hybrid GA/Heuristic Approach for Open-Shop Scheduling Problems", in Proceedings of *ECAI 94: 11th European Conference on Artificial Intelligence*, pp 590-594.
  - Hart E, Ross P. and Nelson J.A.D. (1998) Solving a Real World Problem using an Evolving Heuristically Driven Schedule Builder. *Evolutionary Computing* 6(1):61-80, 1998

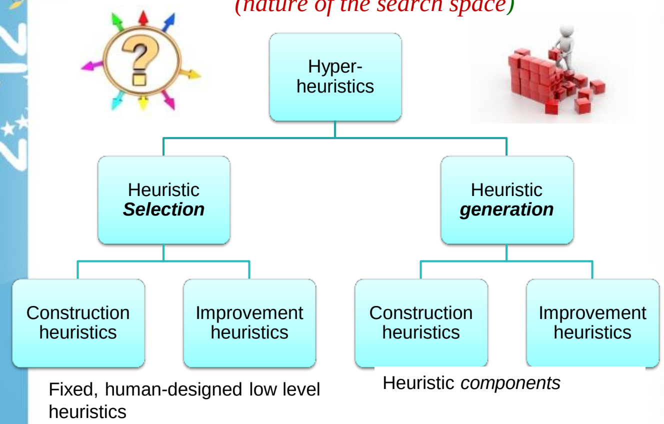
Origins and early approaches

Other early approaches and related themes

- 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 configuring constraint satisfaction problems: a case study. *Constraints* 1(1):7-43

Classification of hyper-heuristics

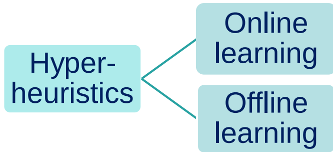
(nature of the search space)



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

Classification of hyper-heuristics

There are 2 Types of Heuristics

Construction

- Search space: partial candidate solutions
- Search step: extension with one or more solution components
- Example in TSP: nearest neighbour

Improvement

- Search space: complete candidate solutions
- Search step: modification of one or more solution components
- Example in TSP: 2-opt exchanges

Complete vs. partial solutions

- Constructive hyper-Heuristics
  - Build the solution incrementally, w.o. backtracking
  - Start with an empty solution and use construction heuristics to build a complete solution
- Improvement or local search hyper-heuristics
  - Find a reasonable initial solution, then use heuristics (neighbourhood structures, or hill-climbers), to find improved solutions
  - Start from a complete solution, then search for improvements by heuristically-guided local search methods

HHs based on construction heuristics vs. HHs based on improvement heuristics

	Improvement	Construction
Initial solution	Complete	Empty
Training phase	No (Online)	Yes (Offline) and No
Objective function	Yes	Other measures may be needed
Low-level heuristics	Operate in solution space	Operate in state space
Stopping condition	User-defined	(automatic) final state
Re-usability	Easy	Less (training required for each problem)

Selection hyper-heuristic based on improvement heuristics

- Example problem: nurse rostering (personnel scheduling)
- The domain barrier hyper-heuristic framework
- Choice function hyper-heuristics
- Tabu-search hyper-heuristic

Nurse rostering: motivation

- Nurse rostering is a complex scheduling problem that affects hospital personnel on a daily basis all over the world
- It is important to:
  - efficiently utilise time and effort
  - evenly balance the workload among people
  - attempt to satisfy personnel preferences
- A high quality roster can lead to a more contented and thus more effective workforce

Nurse rostering: description

- Involves deciding at which times and on which days each employee such work over a specific planning period
- Problems differ in their constraints and objectives
- Basic terminology:
  - **Planning period**: time interval over which the staff have to be scheduled (e.g. 4 weeks)
  - **Skill Category**: class of staff who have a particular level of qualification, skill or responsibility.
  - **Shift type**: hospital duty with a well-defined start and end time. Typically: Early (7:00-15:00), Late (15:00-22:00), and Night (22:00-7:00)
  - **Coverage constraints** (personnel requirements): express the number of personnel needed for every skill category and for every shift or time interval during the entire planning period

Nurse rostering: two types of objectives

- Coverage objectives: aim to ensure that the preferred number of employees (possibly with skills) are working during each shift.
- Employee working objectives: relates to the individual work patterns (schedules) for each employee. They aim to maximise the employees' satisfaction with their work schedules. Example objectives within this group include:
  - Minimum/maximum number of hours worked.
  - Minimum/maximum number of days on or off.
  - Minimum/maximum number of consecutive working days.
  - Minimum/maximum number of consecutive days off.
  - Minimum/maximum number of consecutive working weekends
  - Minimum/maximum number of consecutive weekends off

Nurse rostering: Visualising a solution

	1							2							3							4								
December	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S		
1A	D	E	E	E	L			E	E	E	E		D	D	D	N	N	N				L	L	L	L				51	11
A	DH	DH	DH	DH	DH			DH	DH	DH		DH	DH	DH	DH		DH	DH				DH	DH	DH	DH	DH			20	3
B	N	N	N	N				D	D	L	L	L				L	L	L				E	E	E	D	D			0	0
C	D	D	D	D	D				N	N	N		L	L	L				L	L	L		E	E	E	L			25	4
D				L	N	N	N	N			DH	D				E	E	E	DH	E	E		N	N			E	E	13	7
E					D	DH	DH	D					E	E		DH	E	E	E	DH	DH		D	D	E	E	DH	DH	21	10
F	L	L	L			L	L	L	L			N	N	N	N			D	D			D				D	D	D	10	3
G				E	E	E	E			D	D	D			E	E			D	D	D	D			N	N	N	N	10	2
H	E	E	E			D	D		E	E	E	E			D	D	D		N	N	N	N					L	L	26	6

Total Penalty 176

Unassigned Shifts 0

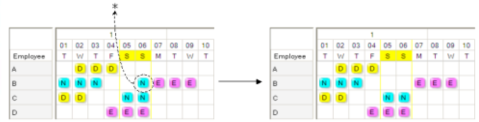
Minimum Cover

E	1	2	2	2	1	1	1	1	2	2	2	1	1	1	1	2	2	2	1	1	1	1	2	2	2	1	1	1
D	2	1	1	1	2	1	1	2	1	1	1	2	1	1	2	1	1	1	2	1	1	2	1	1	1	2	1	1
DH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
L	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
N	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

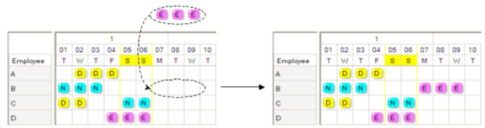
Screenshot from the Nurse Rostering web site at http://www.cs.nott.ac.uk/~tec/NRP/, by Tim Curtois

Move operators: new swaps

The *new* swaps are so called because they introduce new shifts into the roster (or oppositely delete shifts).



Swap single shift

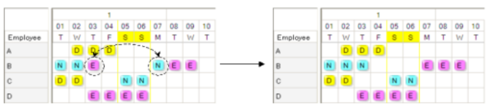


Swap block of shifts

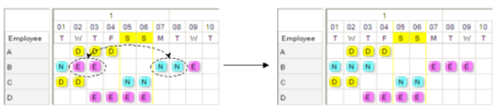
T. Curtois, G. Ochoa, M. Hyde, J. A. Vazquez-Rodriguez (2011) A HyFlex Module for the Personnel Scheduling Problem, University of Nottingham, Tech. Rep.

Move operators: horizontal swaps

Horizontal swaps move shifts in single employee's work pattern hence the shifts move horizontally in the roster.



Swap single shift

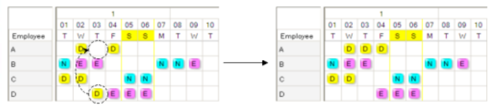


Swap *block* of shifts (adjacent days)

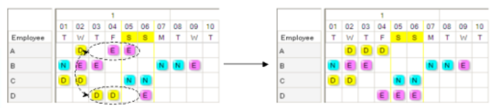
T. Curtois, G. Ochoa, M. Hyde, J. A. Vazquez-Rodriguez (2011) A HyFlex Module for the Personnel Scheduling Problem, University of Nottingham, Tech. Rep.

Move operators: vertical swaps

Vertical swaps move shifts between two employees hence the shifts move vertically in the roster



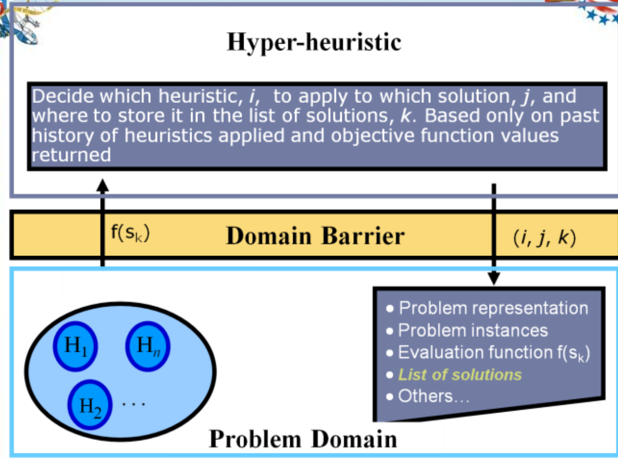
Swap single shift



Swap block of shifts

Instance	Best known	Staff	Shift types	Length (days)	Ref.
BCV-8.13.2	148	13	5	28	[2, 7]
BCV-A.12.1	1294	12	5	31	[2, 7]
ORTEC01	**270**	16	4	31	[4]
GPost	**5**	8	2	28	
QMC-1	**13**	19	3	28	
QMC-2	**29**	19	3	28	
Ikegami-2Shift-DATA1	**0**	28	2	30	[9]
Ikegami-3Shift-DATA1	**2**	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]
WHPP	**5**	30	3	14	[14]
LLR	**301**	27	3	7	[10]
Musa	**175**	11	1	14	[11]
Ozkarahan	**0**	14	2	7	[12]
Azaiez	**0**	13	2	28	[1]
SINTEF	**0**	24	5	21	
CHILD-A2	1111	41	5	42	
MER-A	9915	54	12	48	

Subset of instances from: http://www.cs.nott.ac.uk/~tec/NRP/ (Tim Curtois)



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

Choice function hyper-heuristic

- Several improvement heuristics available. They are ranked according to learned utilities that reflect their past performance
- The overall effectiveness of a heuristic, H_k, is expressed by: $f(H_k) = \alpha f_1(H_k) + \beta f_2(H_j, H_k) + \delta f_3(H_k)$
  - $f_1(H_k)$: recent performance of heuristic H_k
  - $f_2(H_j,H_k)$:recent performance of heuristic pair H_j,H_k
  - $f_3(H_k)$: amount of time since heuristic H_k was called
  - α, β, δ: weights which reflect the importance of each term. Adjusted adaptively
  - f_1 ,f_2 control intensification, f_3 *controls diversification*

Choice function hyper-heuristic

Hyper-heuristic procedure:

Do

-Select low-level heuristic that maximises choice function f and apply it

-Update choice function parameters

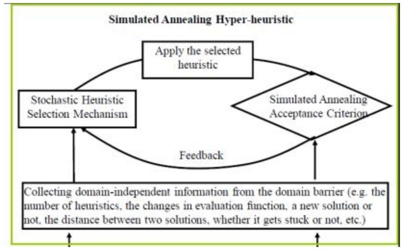
Until Stopping Condition is met

(α, β, δ) *parameters* , adjusted adaptively

- Increase value of intensification *(α, β)* parameters when low-level heuristic produced a better solution (**reward**)
- Decrease values otherwise (**penalty**)
- Increase value of diversification parameter (δ) when there has been no improvement after a certain number of iterations

Tabu-search hyper-heuristic

- Heuristics selected according to learned ranks (using reinforcement learning)
- Dynamic tabu list of heuristics that are temporarily excluded from the selection pool
- Applied to: nurse rostering and course timetabling



Later combined with SA and other acceptance criteria

Tabu search hyper-heuristic

Each heuristic *k* is assigned a rank r_k initialised to *0* and allowed to increase and decrease within interval $[r_{min}, r_{max}]$

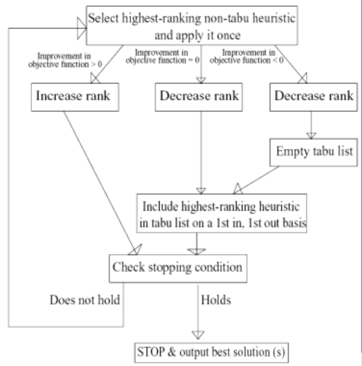
Do:

1- Select heuristic k with highest rank r_k and apply it once

2 - If $\Delta > 0$ then $r_k = r_k + \alpha$

- Otherwise $r_k = r_k - \alpha$, Include heuristic k in TABULIST

Until Stop = true.



Summary of Part I

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

- Main feature: search in a space of heuristics
- Term used for '*heuristics to choose heuristics*' in 2000
- Ideas can be traced back to the 60s and 70s
- Two main type of approaches
  - Heuristic selection
  - Heuristic 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

References: Hyper-heuristics

Introductory tutorials and survey papers

- E. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, R. Qu, Hyper-heuristics: A Survey of the State of the Art, *Journal of the Operational Research Society*, Palgrave Macmillan, (*to appear*).
- E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and J. Woodward (2010). 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.
- E. Burke, M. Hyde, G. Kendall, G. Ochoa, J. Woodward (2009) Exploring Hyper-heuristic Methodologies with Genetic Programming, *Collaborative Computational Intelligence*, Intelligent Systems Reference Library, vol.1.
- E.K.Burke, G. Kendall, J.Newall, E.Hart, P.Ross & S.Schulenburg, Hyper-Heuristics: An Emerging Direction in Modern Search Technology, *Handbook of Metaheuristics* (eds. F.Glover & G.Kochenberger), pp 457 – 474, Kluwer, 2003.
- P. Ross, P. (2005) Hyper-heuristics, Chapter 17 in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Methodologies* (Eds. E.K.Burke and G.Kendall), Springer, 529–556.

References: Hyper-heuristics

Online resources and bibliographies

- CHeSC 2011: http://www.asap.cs.nott.ac.uk/external/chesc2011/resources.html
- HH Bibliography: http://allserv.kahosl.be/~mustafa.misir/hh.html

Papers discussed: selection hyper-heuristics with improvement heuristics

- 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
- Edmund Burke, Graham Kendall, Eric Soubeiga, A Tabu-Search Hyper-Heuristic for Timetabling and Rostering, *Journal of Heuristics*, 9(3), Springer, 2003.

References: related areas

Books

- R. Battiti, M. Brunato, F. Mascia (2008) *Reactive Search and Intelligent Optimization*, Operations Research/Computer Science Interfaces Series, Vol. 45, Springer.
- T. Bartz-Beielstein, M. Chiarandini, L. Paquete, M. Preuss (Eds.) (2010) *Experimental Methods for the Analysis of Optimization Algorithms*, Springer Berlin.
- M. Birattari (2009). *Tuning Metaheuristics: A machine learning perspective*. Studies in Computational Intelligence,197. Springer, Berlin.
- F.G. Lobo, C.F. Lima, and Z. Michalewicz (Eds.), (2007) *Parameter Setting in Evolutionary Algorithms*, Studies in Computational Intelligence, Springer.

Journal Papers

- A. Fialho, L. Da Costa, M. Schoenauer and M. (2010) Analyzing Bandit-based Adaptive Operator Selection Mechanisms. *Annals of Mathematics and Artificial Intelligence*, Springer,
- F. Hutter, h. Hoos H, Leyton-Brown K, Stutzle T (2009) Paramils: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research (JAIR),* 36:267-306.
- Y.S. Ong, M.H Lim, N. Zhu, K.W. Wong (2006) Classification of adaptive memetic algo rithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B 36(1):141-152
- Pisinger, D. amd S. Ropke (2007) A general heuristic for vehicle routing problems, *Computers & Operations Research*, Vol. 34, Issue 8, 2403–2435.
- Smith, J. E. (2007) Co-evolving Memetic Algorithms: A review and progress report. *IEEE Transactions in Systems, Man and Cybernetics, part B*. Vol 37:1 pp 6-17.

Part II

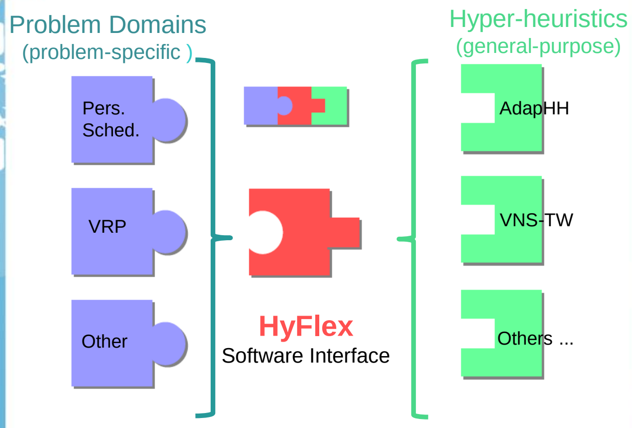
- HyFlex: (*Hyper-heuristic Flexible framework*)
  - Motivation
  - Main features
  - Example problem domains

- The Cross-domain Challenge
  - Main features
  - Results
  - Design principles of the best algorithms
  - Summary and future work

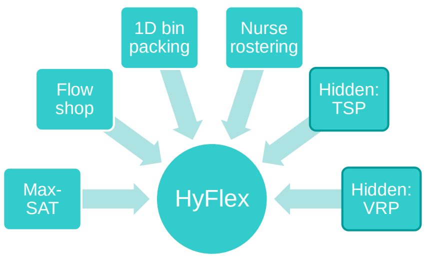
HyFlex : Motivation

- Researchers are often constrained on the number of problem domains on which to test their adaptive methods
- **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

The concept of HyFlex



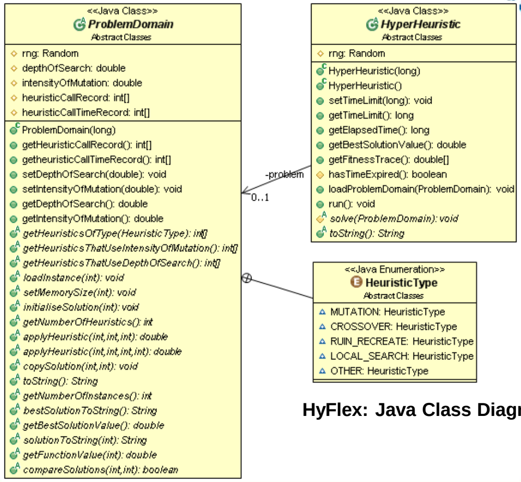
HyFlex: currently 6 problem domains



- **HyFlex**: (1) a Java Interface (2) a library of interesting problems
- **Antecedents**
  - The domain barrier framework of hyper-heuristics (Cowling P., Kendall G. and Soubeiga, 2000, 2001)
  - The PISA Framework for Multi-objective Optimization (S. Bleuler, M. Laumanns, L. Thiele, E. Zitzler, 2003), ETH, Zurich

The problem domain modules

1. Initialisation of solutions
2. Population or memory of solutions
3. Problem instances
4. Fitness function
5. Low-level heuristics (search operators)
   1. Mutation
   2. Ruin-recreate
   3. Crossover
   4. Hill-climbers



HyFlex: Java Class Diagram

Java code for running a hyper-heuristic on a problem domain

```
ProblemDomain problem = new SAT(seed1);
HyperHeuristic HHObject = new ExampleHyperHeuristic1(seed2);
problem.loadInstance(0);
HHObject.setTimeLimit(60000);
HHObject.loadProblemDomain(problem);
HHObject.run();
System.out.println(HHObject.getBestSolutionValue())
```

Algorithm 1

Algorithm 2 Pseudocode for the solve method of `ExampleHyperHeuristic1`. This is called when the `run()` method of the hyper-heuristic is called (see Algorithm 1)

```
Require: A ProblemDomain object, problem
  int numberOfHeuristics = problem.getNumberOfHeuristics
  double currentObjValue = Double.POSITIVE-INFINITY
  problem.initialiseSolution(0)
  while hasTimeExpired = FALSE do
    int h = rng.nextInt(numberOfHeuristics)
    double newObjValue = problem.applyHeuristic(h, 0, 1)
    double delta = currentObjValue - newObjValue
    if delta > 0 then
      problem.copySolution(1, 0)
      currentObjValue = newObjValue;
    else
      if rng.nextBoolean = TRUE then
        problem.copySolution(1, 0)
        currentObjValue = newObjValue;
      end if
    end if
  end while
```

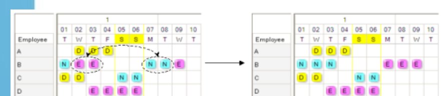
Personnel scheduling

Tim Curtois

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

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

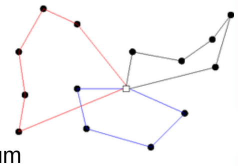
	BCV-A.12.1	1294	12	5	31	[2, 7]
	BCV-A.12.2	1953	12	5	31	[2, 7]
	ORTEC01	**270**	16	4	31	[4]
	ORTEC02	290	16	4	31	[4]
	GPost	**5**	8	2	28	
	GPost-B	**3**	8	2	28	
	QMC-1	16	19	3	28	
	QMC-2	29	19	3	28	
	Ikegami-2Shift-DATA1	**0**	28	2	30	[9]
	Ikegami-3Shift-DATA1	6	25	3	30	[9]
	Ikegami-3Shift-DATA1.1	13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



Example heuristic *horizontal* swap: move shifts in single employee's work pattern

Vehicle Routing Problem

- A set of customers and a central depot
- A set of vehicles, located at the depot
- Goal: Design minimum cost routes visiting all customers
- Additional constraints
  - Capacity
  - Time windows
- Objective function: weighted sum number of vehicles and distance travelled



Vehicle routing domain

James Walker, Gabriela Ochoa, Prof. Michel Gendreau

Mutational	Local Search	Ruin & Recreate	Crossover
Two-opt [4] Or-opt [5] Two-opt* [2] Shift [1] Interchange [1]	Simple hill-climbers based on mutational heuristics GENI [3]	Time-based radial ruin[6] Location-based radial ruin[6]	Combine routes Longest Combine: orders routes according to length

[1] M. W. P. Savelsbergh. The vehicle routing problem with time windows: Minimizing route duration. *INFORMS Journal on Computing, 4(2):146-154, 1992.*
[2] J-Y. Potvin and J-M. Rousseau. An exchange heuristic for routing problems with time windows. *The Journal of the Operational Research Society, 1995.*
[3] M. Gendreau, A. Hertz, and G. Laporte. A new insertion and postoptimization procedures for the traveling salesman problem. *Operations Research, 1992.*
[4] O. Braysy and M. Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science, 2005.*
[5] I. Or. *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. PhD thesis, Northwestern*
[6] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics, 2000.*

Adaptive iterated local search

s_0 = GenerateInitialSolution
s^* = ImprovementStage(s_0)
Repeat
  s' = PerturbationStage(s^*)
  s'^* = ImprovementStage(s')
  if $f(s'^*) < f(s^*)$
   $s^* = s'^*$
Until time-limit reached

Baseline ILS

- Perturbation stage, AOS:
  - *extreme value* credit assignment
  - *adaptive pursuit* selection
- Improvement stage:
  - order LS according to score
  - Score: mean improvement in obj. function
  - Apply all LS in this order

HyFlex enables connecting hyper-heuristic research with adaptive operator selection and adaptive meta-heuristics

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

CHeSC 2011

Experiments

Instances:

- MAX-SAT
- Flow Shop
- Personnel Scheduling
- Bin Packing
- Hidden Domain TSP
- Hidden Domain VRP

SAT Instance 1:
HH1 – 34
HH2 – 23
HH3 – 27
HH4 – 10
HH5 – 30
...

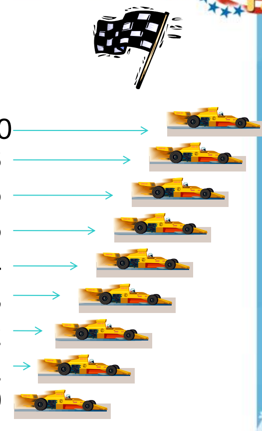
- 30 Instances, 5 per domain
- 10 minutes per run
- 31 runs per instance,
- Median of objective function values used for ranking the algorithms

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

10
8
6
5
4
3
2
1
0



Results vs. Leaderboard

RESULTS

#	Algorithm Description	Score	Author/Team	Affiliation
1	AdapHH	181	Mustafa Misir	University KaHo Sint-Lieven, Belgium
2	VNS-TW	134	Ping-Che Hsiao	National Taiwan University, Taiwan
3	ML	131.5	Mathieu Larose	Université de Montréa,Canada
4	PHUNTER	93.25	Fan Xue	Hong Kong Polytechnic U., Hong Kong
5	EPH	89.75	David Meignan	Polytechnique Montréal, Canada
6	HAHA	75.75	Andreas Lehrbaum	Vienna University of Technology, Austria
7	NAHH	75	Franco Mascia	Université Libre de Bruxelles, Belgium
8	ISEA	71	Jiri Kubalik	Czech Technical University, Czech Rep.

LEADERBOARD

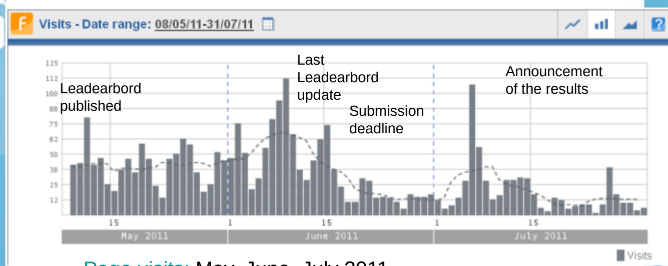
	Algorithm	Author/Team	Score	Date	Affiliation
1	PHunter4	Fan Xue	204.28	07/06/11	Hong Kong Polytechnic U., Hong Kong
2	ISEA2	Jiri Kubalik	177.28	25/05/11	Czech Technical University, Czech Rep.
3	HAHA1	Andreas Lehrbaum	166.78	28/04/11	Vienna University of Technology, Austria
4	TW4	Hsiao Ping-Che	130.20	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

Leaderboard

Leaderboard Submission

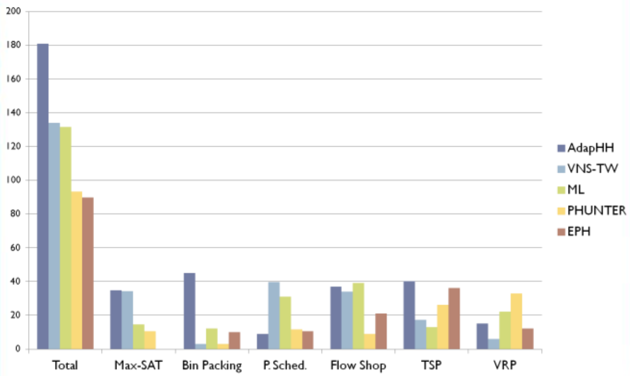
MaxSAT
Bin Packing
Pers. Sched
Flow Shop

10, 33, 22, 7, 10, 14.0, 8.0, 7.0, 7.0, 213.0
0.0108, 0.0077, 0.0210, 0.0226, 0.0045, 0.0032, 0.0160, 0.0268, 0.05495, 0.0231
3338, 2285, 390, 20, 20, 21, 1126, 2277, 3245, 10452.
6303, 6279, 6335, 6350, 6398, 10501, 10922, 26284, 26867, 26658

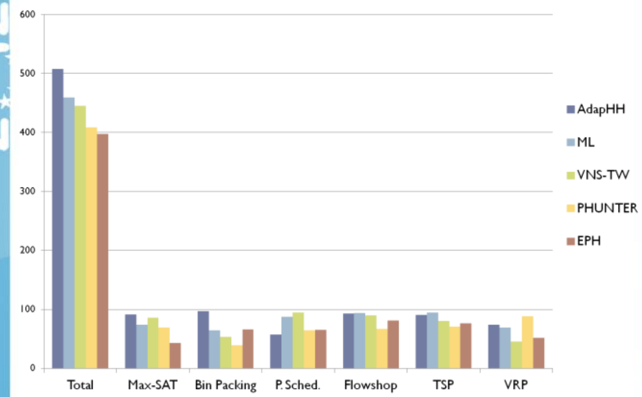


Page visits: May, June, July 2011

Results – Top 5: Formula 1 score

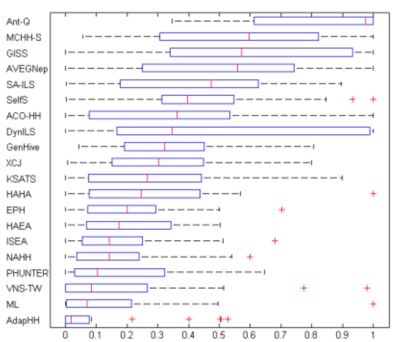


Results – Top 5: Borda count



Normalised objective function values

The median objective function values are then normalised by comparing them to the highest and lowest medians for each instance



Suggestion by Tomasso Urli, one of the CHeSC competitors

Rankings: different metrics

Rank	F1-Median	Borda-Median	F1-Best	Borda-Best	Norm-Median
1	AdapHH	AdapHH	AdapHH	AdapHH	AdapHH
2	VNS-TW	ML	VNS-TW	VNS-TW	ML
3	ML	VNS-TW	PHUNTER	ML	VNS-TW
4	PHUNTER	PHUNTER	ML	PHUNTER	PHUNTER
5	EPH	EPH	ISEA	ISEA	NAHH
6	HAHA	ISEA	EPH	EPH	ISEA
7	NAHH	NAHH	NAHH	NAHH	HAEA
8	ISEA	HAEA	HAHA	HAEA	EPH
9	KSATS-HH	HAHA	KSATS-HH	HAHA	HAHA
10	HAEA	KSATS-HH	HAEA	KSATS-HH	KSATS-HH

Winner: AdapHH
Top 4: AdapHH, VNS-TW, ML, PHUNTER
Top 8: AdapHH, VNS-TW, ML, PHUNTER, EPH, ISEA , NAHH, HAEA

Rankings excluding one domain

Formula 1:

Rank	All	- Max-SAT	- Bin P.	- Pers. S.	- Flow S.	- TSP	- VRP
1	AdapHH	AdapHH	VNS-TW	AdapHH	AdapHH	AdapHH	AdapHH
2	VNS-TW	ML	AdapHH	ML	VNS-TW	ML	VNS-TW
3	ML	VNS-TW	ML	VNS-TW	ML	VNS-TW	ML
4	PHUNTER	EPH	EPH	PHUNTER	PHUNTER	HAHA	EPH
5	EPH	ISEA	HAHA	EPH	HAHA	PHUNTER	NAHH

Borda:

Rank	All	- Max-SAT	- Bin P.	- Pers. S	- Flow S.	- TSP	- VRP
1	AdapHH	AdapHH	VNS-TW	AdapHH	AdapHH	AdapHH	AdapHH
2	ML	ML	AdapHH	ML	ML	ML	VNS-TW
3	VNS-TW	VNS-TW	ML	VNS-TW	VNS-TW	VNS-TW	ML
4	PHUNTER	EPH	PHUNTER	PHUNTER	PHUNTER	HAHA	EPH
5	EPH	ISEA	EPH	NAHH	ISEA	PHUNTER	NAHH

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 competition winner: AdapHH

Feedback from operators

Counter based | Value based

Improving moves | Worsening moves | Equal moves | Amount of improvement | Amount of worsening | Speed

The number of *new* improvement moves and the amount of *new* improvements are also considered

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 heuristics
- Simple adaptive acceptance criteria

The 4th and 5th Places

PHUNTER

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)

Design principles

- ❖ Previous principles confirmed and improved
  - Use of reinforcement learning for heuristic selection
  - Excluding (dynamically) some heuristics (Tabu HH)
  - Feedback to guide heuristic choice: fitness improvement, speed, number of new solutions
- ❖ New(er) principles enhanced by HyFlex
  - Use of diversification and intensification phases
  - Adaptation of the heuristic parameters
  - Use of adaptive acceptance criteria
  - Local and global learning of heuristic performances
  - Evolution and co-evolution of heuristic sequences
  - Use of a population (with or without crossover)

HyFlex achievements

HyFlex Papers in

Nurse rostering best-known solutions obtained by the PHUNTER HyFlex HH

Instance	HyFlex Best	Previous Best	Staff	Shift Types	Days
CHILD-A2	1095	1111	41	5	42
ERRVH-A	2135	2197	51	8	42
ERRVH-B	3105	6659	51	8	42
ERMGH-B	1355	1459	41	4	42
BCV-A.12.2	1875	1953	12	5	31
MER-A	8814	9915	54	12	42

Conclusions

- ❖ HyFlex: A new benchmark for adaptive algorithms
  1. A software interface
  2. A library of interesting problem domains
  3. A library of interesting adaptive algorithms

- ❖ Not only hyper-heuristics! but adaptive ILS, MA, VNS, EAs with AOS, autonomous search, etc.
- ❖ Future
  - Improvements and extensions to the HyFlex interface
  - New and exciting domains
  - Running a more challenging competition!

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

References: HyFlex (ASAP group)

- G. Ochoa, M. Hyde, T. Curtois, J. A. Vazquez-Rodriguez, J. Walker, M. Gendreau, G. Kendall, B. McCollum, A. J. Parkes, S. Petrovic, E. K. Burke (2012). HyFlex: A Benchmark Framework for Cross-domain Heuristic Search. *European Conference on Evolutionary Computation in Combinatorial Optimisation* (EvoCOP 2012), J.-K. Hao and M. Middendorf (Eds.), LNCS 7245, pp. 136-147. Springer, Heidelberg
- J. Walker, G. Ochoa, M. Gendreau, E. Burke (2012) A Vehicle Routing Domain for the HyFlex Hyper-heuristics Framework, *the 6th Learning and Intelligent OptimizatioN Conference* (LION12), Paris, France.
- E. K. Burke, M. Gendreau, G. Ochoa, J. Walker (2011) Adaptive Iterated Local Search for Cross-domain Optimisation. Proceedings of *Genetic and Evolutionary Computation Conference* (GECCO-2011), ACM, pp. 1987-1994.
- E. K. Burke, T. Curtois, M. Hyde, G. Kendall, G. Ochoa, S. Petrovic, J. A. Vazquez-Rodriguez and M. Gendreau (2010) Iterated Local Search vs. Hyper-heuristics: Towards General-purpose Search Algorithms. *IEEE Congress on Evolutionary Computation* (CEC 2010), pp. 3073-3080.

References: HyFlex (others)

- C.Y. Chan, Fan Xue, W.H. Ip, C.F. Cheung (2010) A Hyper-heuristic Inspired by Pearl Hunting, the *6th Learning and Intelligent OptimizatioN Conference* (LION12), Paris, France.
- L. Di Gaspero, T. Urli (2012) Evaluation of a Family of Reinforcement Learning Cross-domain Optimization Heuristics, the *6th Learning and Intelligent OptimizatioN Conference* (LION12), Paris, France
- A. Lehrbaum, N. Musliu (2012) A New Hyperheuristic Algorithm for Cross Domain Search Problems, the 6th Learning and Intelligent OptimizatioN Conference (LION12), Paris, France.
- J. Kubalik (2012) Hyper-Heuristic Based on Iterated Local Search Driven by Evolutionary Algorithm, the *12th European Conference on Evolutionary Computation in Combinatorial Optimisation* (EvoCOP12), Malaga, Spain, LNCS series, Vol. 7245, Springer, pp. 148-159.
- F. Mascia, T. Stützle (2012) A Non-Adaptive Stochastic Local Search Algorithm for the CHeSC 2011 Competition, *the 6th Learning and Intelligent OptimizatioN Conference* (LION12), Paris, France.
- M. Misir, K. Verbeeck, P. De Causmaecker, G. Vanden Berghe (2012) An Intelligent Hyper-heuristic Framework for CHeSC 2011, *the 6th Learning and Intelligent OptimizatioN Conference* (LION12), Paris, France.