Hybridising Heuristics within a Graph based Hyper-Heuristic Framework

Dr Rong Qu, Associate Professor
ASAP Group, The University of Nottingham
rong.qu@nottingham.ac.uk
http://www.cs.nott.ac.uk/~pszrq

NATCOR – Heuristics and Approximate Algorithms
Nottingham, April, 2016

Educational Timetabling

- Assigning a set of exams into limited timeslots satisfying
  - Hard constraints: cannot be violated
  - Soft constraints: desired
  - Quality of solutions: objective function

- Events
- Timeslots
- Rooms
- Etc.

Programme: Computer Science 12 month PG (Session start) Full time/1

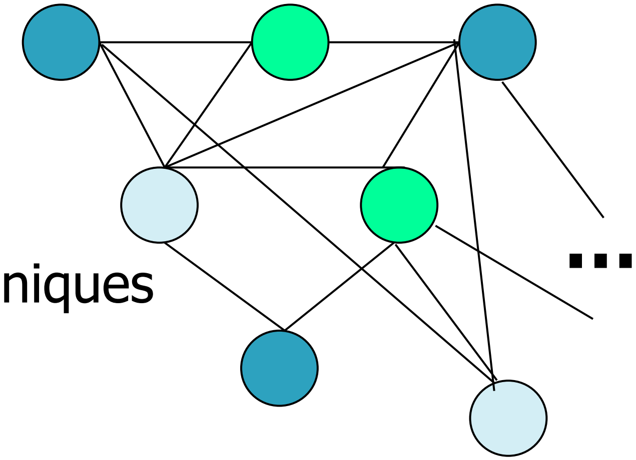
Weeks: 19-26, 31-34 (25 Jan 2016-15 May 2016)

Click here to go to a list view of this timetable

	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30	17:00	17:30
Mon	G54SIM/C1/01 Computing Simulation For Decision Support JC-COMPSCI-A32 19-26, 31-32, 34				G53DIA/T1/01 Tutorial Designing Intelligent Agents JC-COMPSCI-C60 19-26, 31-32		G54SWE/C2/01 Computing Software Engineering JC-COMPSCI-A32 19-26, 31-32, 34		G53FUZ/L1/01 Lecture Fuzzy Sets And Fuzzy Logic Systems JC-AMEN-B18 19-26, 31-32, 34				G53DIA/L1/01 Lecture Designing Intelligent Agents JC-COMPSCI-C60 19-26, 31-32		G53GRA/C1/01 Computing Computer Graphics JC-COMPSCI-A32 19-26, 34			
Mon	G53NMD/S1/01 Seminar New Media Design JC-BSSOUTH-A25 19-26, 31-32, 34						G54GAM/L1/01 Lecture Games JC-EXCHGE-B.LT1 19-26, 31-32, 34		G54PDC/C1/01 Computing Parallel And Distributed Computing JC-COMPSCI-A32 20-26, 31-32, 34		G54DMA/C1/01 Computing Data Modelling And Analysis JC-COMPSCI-A32 20-26, 31-32				G54ACC/L2/01 Lecture Advanced Computer Communications JC-BSSOUTH-A08 19-26, 31-32, 34			
Tue	G54SWE/L1/01 Lecture Software Engineering JC-EXCHGE-B.LT1 19-26, 31-33				G54SWE/C1/01 Computing Software Engineering JC-COMPSCI-A32 19-26, 31-34				G54SIM/L1/01 Lecture Simulation For Decision Support JC-COMPSCI-C60 19-26, 31-33				G53DIA/L2/01 Lecture Designing Intelligent Agents JC-COMPSCI-C60 19-26, 31-32					
Tue					G52PSA/L3/01 Lecture Planning, Search And Artificial Intelligence Programming JC-EXCHGE-B.LT1 19-26, 31-34		G53COM/L1/01 Lecture Computability JC-COMPSCI-C11, JC-BSSOUTH-A06 19-26, 31-33						G54DET/T1/01 Tutorial Design Ethnography JC-COMPSCI-C11 19-26, 31-34		G54DMA/L1/01 Lecture Data Modelling And Analysis JC-EXCHGE-C3 19-26, 31-33			
Wed	G54SPM/L1/01 Lecture Software Project Management JC-EXCHGE-B.LT1 19-26, 31-34				G52PSA/L4/01 Lecture Planning, Search And Artificial Intelligence Programming JC-EXCHGE-B.LT1 20-26, 31-34		G54GAM/L2/01 Lecture Games JC-EXCHGE-B.LT1 19-26, 31-34											
Thu													G54PDC/L1/01 Lecture Parallel And Distributed Computing JC-BSSOUTH-A25 19-26, 31-33				G54SWE/L2/01 Lecture Software Engineering JC-COMPSCI-C60 19-26, 31-33	
Thu	G54ACC/L1/01 Lecture Advanced Computer Communications JC-DEARING-B43 19-26, 31-33		G54ACC/C1/01 Computing Advanced Computer Communications JC-COMPSCI-B52 19-26, 31-34		G54GAM/C1/01 Computing Games JC-COMPSCI-A32 19-26, 31-34				G54DET/L1/01 Lecture Design Ethnography JC-DEARING-B37 19-26, 31-34		G53NMD/C1/01 Computing New Media Design JC-COMPSCI-A32 20-26, 31				G53MLE/L2/01 Lecture Machine Learning JC-EXCHGE-C3 19-26, 31-32			
Fri	G54STA/L1/01 <25-26> Lecture Selected Topics In Artificial Intelligence JC-DEARING-B43 25-26																	
Fri	G54STA/L1/01 <19-24> Lecture Selected Topics In Artificial Intelligence JC-COMPSCI-C60 19-24				G53DIA/T1/02 Tutorial Designing Intelligent Agents JC-COMPSCI-C60 19-26, 31-32				G54PDC/L2/01 Lecture Parallel And Distributed Computing JC-EXCHGE-C33 19-26, 31-33						G53GRA/L1/01 Lecture Computer Graphics JC-BSSOUTH-B52 19-26			
Fri	G53MLE/C1/01 Computing Machine Learning JC-COMPSCI-A32 21-26, 31				G53COM/L2/01 Lecture Computability JC-COMPSCI-C11, JC-BSSOUTH-A24 19-26, 31-33				G53FUZ/P1/01 Practical Fuzzy Sets And Fuzzy Logic Systems JC-COMPSCI-A32 19-26, 31-34		G53MLE/L1/01 Lecture Machine Learning JC-EXCHGE-C33 19-26, 31-32				G52PSA/C2/01 Computing Planning, Search And Artificial Intelligence Programming JC-COMPSCI-A32 19-26, 31-34			

Educational Timetabling

- Important activities in all universities
  - Hard constraints: No events for students at the same time
  - Soft constraints: Spread students' events



- State-of-the-art: different "tailor-made", "fine-tuned" techniques
  - Graph heuristics, constraint based techniques
  - Meta-heuristics, multi-criteria
  - Recent developments:
    - hybrid techniques, hyper-heuristics, VNS, ILS, GRASP, adaptive techniques, etc.

R. Qu, Burke E.K., McCollum B., Merlot L.T.G. and Lee S.Y.: **A Survey of Search Methodologies and Automated Approaches for Examination Timetabling**. Journal of Scheduling, 12(1): 55-89, 2009. **Top 1% cited by ISI**

Educational Timetabling

- Carter, Laporte & Lee (1996): exam timetabling instances
  - Hard constraint: conflicts between exams
  - Objective function: min time slots (graph colouring)
  - Soft constraints: spread out exams over time slots
  - Objective function: $C(t) = (\sum_{s=0}^{4} w_s N_s) / S$
- Meta-heuristic Network (2000): course timetabling instances
  - Hard constraints: exams conflicts, room features
  - Soft constraints: minimise only one class a day, class in the last slot of a day, more than two classes in a row
  - Objective function: min sum of the costs for soft constraints
- New benchmark: international TIMETABLING COMPETITION

A Graph Based Hyper-heuristic

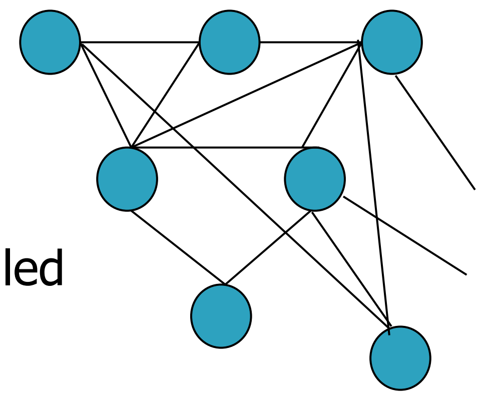
- Hyper-heuristics: Heuristics that choose heuristics
  - High level: Meta-heuristics, Choice function, CBR, etc.
  - Low level: moving strategies, constructive heuristics, etc.
- Aim of **hyper-heuristic**
  - Explore general techniques for wider problems
  - High level search doesn't look into domain knowledge
- Applications
  - bin packing, educational timetabling, personal scheduling, etc.

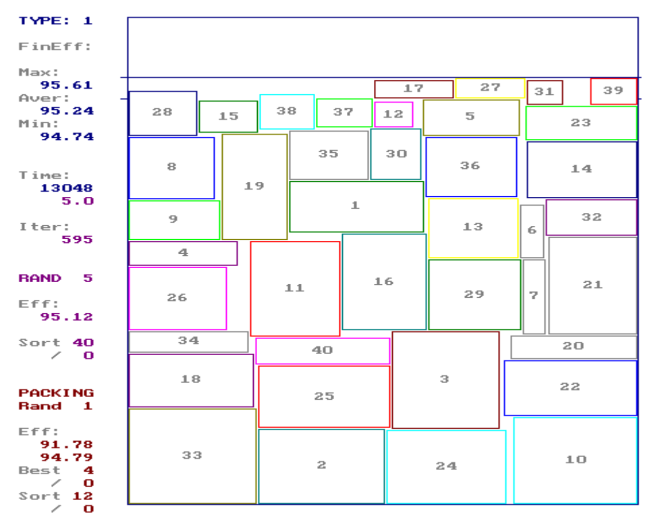
R. Qu, co-authors: E. K. Burke, A. Meisels, S. Petrovic. A Graph-based Hyper-Heuristic for Exam Timetabling Problems. EJOR, 176: 177-192, 2007. Five Year Top Cited Article EJOR 2007-2011 Award

A Graph Based Hyper-heuristic

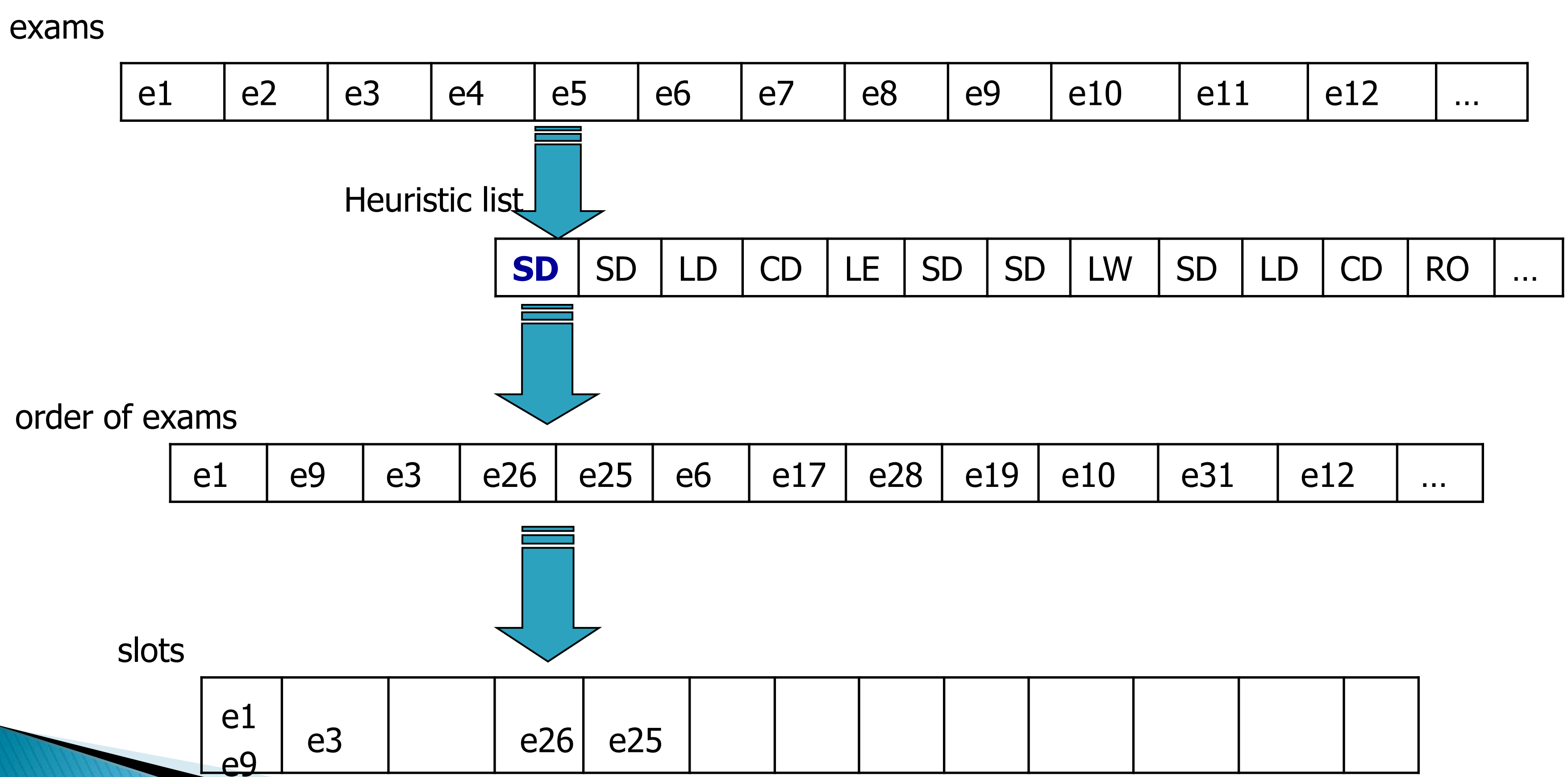
- High level search: Any meta-heuristics
  - Search for lists of low level heuristics to construct solutions
- Low level heuristics: order events by how *difficult* to schedule them
  - Saturation Degree: least available slots
  - Colour Degree: most conflicted with those scheduled
  - Largest Degree: most conflicted with the others
  - Largest Weighted Degree: LD + students
  - Largest Enrolment: students enrolled
  - Random Ordering: brings randomness

  - Bin packing: best fit, first fit





A Graph Based Hyper-heuristic



A Graph Based Hyper-heuristic

exams

	e2		e4	e5	e6	e7	e8		e10	e11	e12	…

Heuristic list

SD	**SD**	LD	CD	LE	SD	SD	LW	SD	LD	CD	RO	…

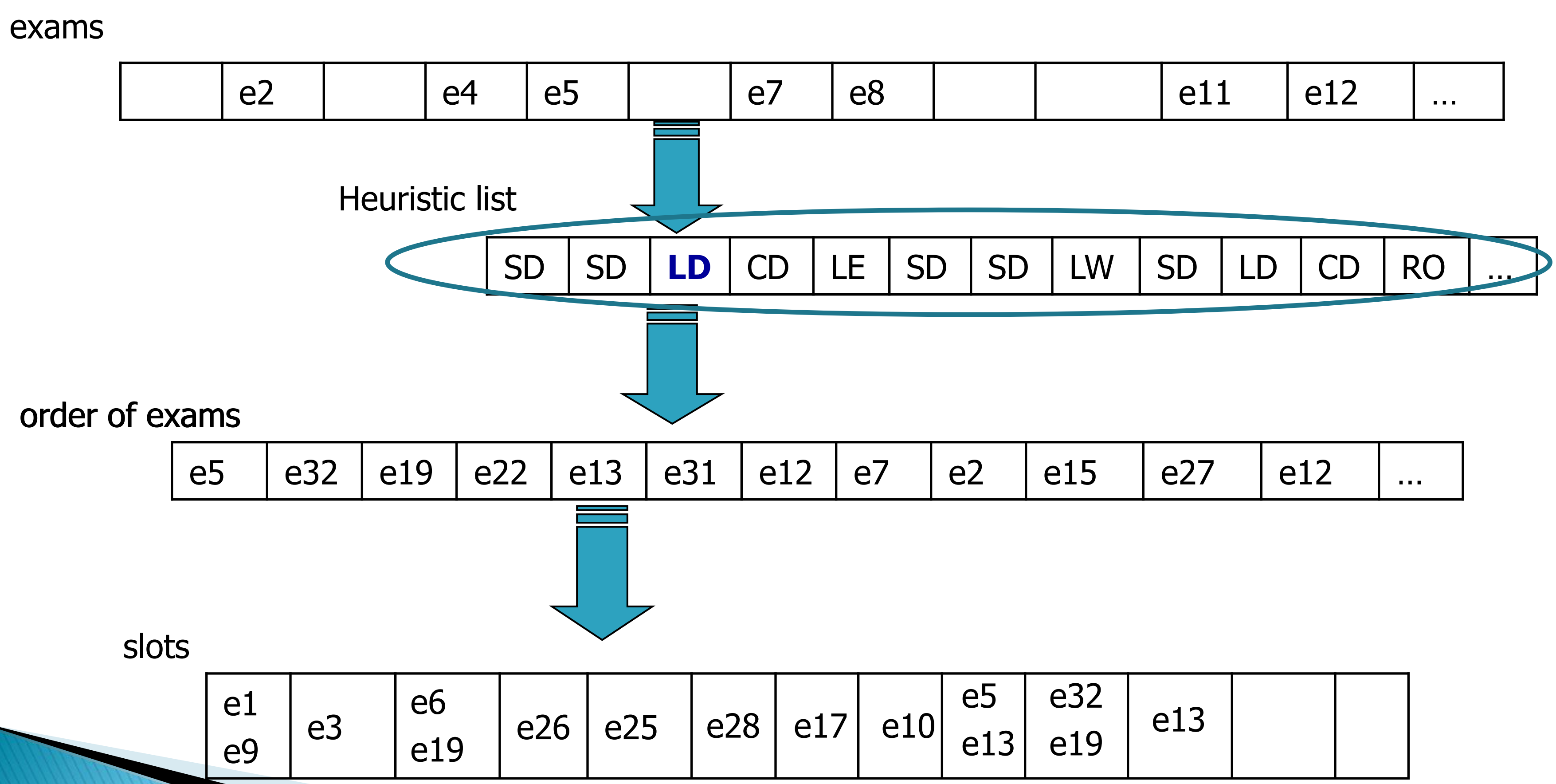
order of exams

e6	e17	e28	e19	e10	e31	e12	e5	e22	e32	e27	e19	…

slots

e1 e9	e3	e6 e19	e26	e25	e28	e17	e10					

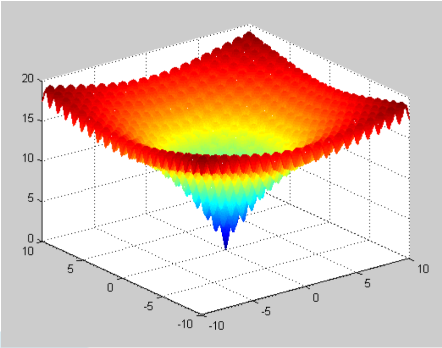
A Graph Based Hyper-heuristic

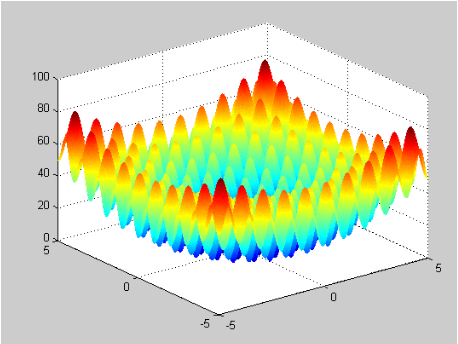


A Graph Based Hyper-heuristic

- Graph based Hyper-heuristics (GHH) Framework
  - Search space: permutations of graph heuristics, rather than actual solutions
  - Moving operator: randomly change two heuristics in the heuristic list within a local search
  - Objective function: maps heuristic lists to penalty of timetables constructed

- Further investigations
  - Role of different high / low level heuristics (ILS, TS, SDM, VNS)
  - Characteristics of *heuristic* search space
  - Search in two search spaces

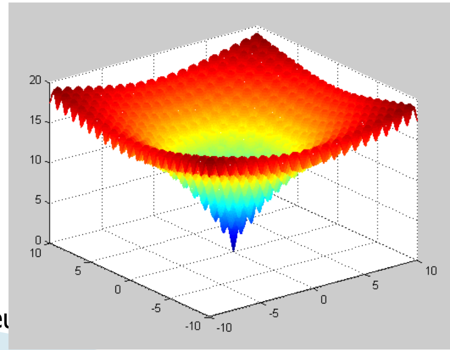


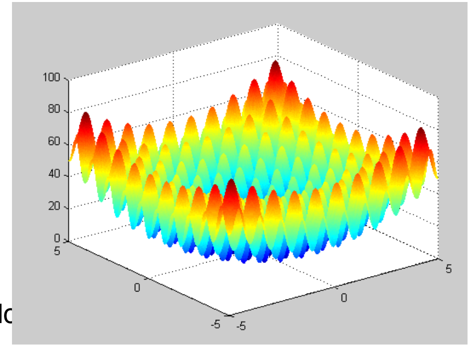


Which High Level Heuristics?

- High level search methods
  - Iterated Local Search
  - Tabu Search
  - Steepest Descent
  - Variable Neighbourhood Search

- Objective function
  - heuristic lists → penalties (costs of timetables constructed)
    - "Walks" are allowed. Why?





Which High Level Heuristics?

		car91	car92	ear83	hec92	kfu93	lse91	sta83	tre92	ute92	uta93	yor83
SDM	best	5.44	4.87	35.54	12.59	15.25	13.01	160.3	9.01	31.77	3.61	42.77
SDM	avg	6.18	5.3	36.8	12.74	15.63	13.51	163.7	9.37	32.6	4.5	43.6
SDM	time	**15367**	**8001**	**584**	**22**	**2502**	*1722*	*69*	**1597**	**87**	**8018**	**426**
ILS	best	**5.3**	*4.77*	38.39	12.72	**15.09**	*12.72*	**159.2**	*8.74*	**30.32**	**3.32**	**40.24**
ILS	avg	6.01	5.18	39.58	13.01	15.35	13.1	161.6	8.92	31.3	4.01	43.15
ILS	time	17334	8200	*617*	*31*	2629	1832	73	*1638*	*100*	10464	527
TS	best	5.43	4.94	38.19	*12.36*	15.97	13.25	165.7	8.87	32.12	*3.52*	*41.3*
TS	avg	6.3	5.34	45.56	14.6	19.55	14.29	169.1	9.67	37.02	4.38	47.97
TS	time	20393	9111	649	32	2768	1970	80	1800	*100*	10464	527
VNS	best	*5.4*	**4.7**	*37.29*	**12.23**	*15.1*	**12.71**	*159.3*	**8.67**	*30.23*	3.56	43
VNS	avg	6.1	5.1	38.63	12.72	15.24	13.06	163.3	8.88	31.7	4.05	43.93
VNS	time	*16321*	*8107*	672	42	*2531*	**1653**	**47**	1721	677	*9210*	*501*

Which High Level Heuristics?

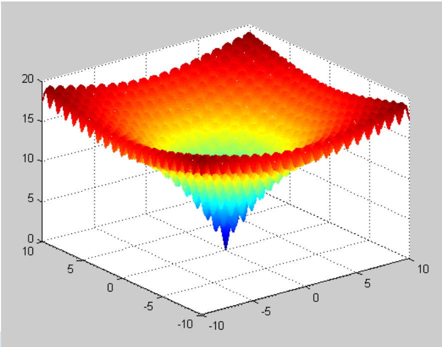
		s1	s2	s3	s4	s5	m1	m2	m3	m4	m5	large
SDM	best	7	8	3	6	10	368	100%	367	356	195	100%
SDM	avg	10.8	15.6	5	11.8	12.2	382.5	100%	383	374.5	194.5	100%
SDM	time	15	38	10	8	30	3823	3672	3752	3637	1989	4013
ILS	best	6	9	4	6	8	373	461	375	374	172	1132
ILS	avg	8.8	13.2	5.4	7.6	12	375	480.5	377.5	380.5	179.7	1144 60%
ILS	time	32	47	15	11	23	3656	3018	3382	3451	1822	3811
TS	best	11	11	5	11	16	496	533	460	529	214	1164
TS	avg	12.2	16.4	9.2	12.2	18.2	511.5	533 80%	468	539	236	1164 80%
TS	time	**12**	**18**	**9**	**7**	**19**	**3326**	**2996**	**3160**	**3280**	**1650**	**3564**
VNS	best	7	12	4	6	6	346	433	359	370	156	1148
VNS	avg	10	14.8	5.2	8	10.6	365	443 40%	369.5	377.5	165.5	1148 80%
VNS	time	32	45	16	10	30	3920	3723	3856	3667	2013	4079

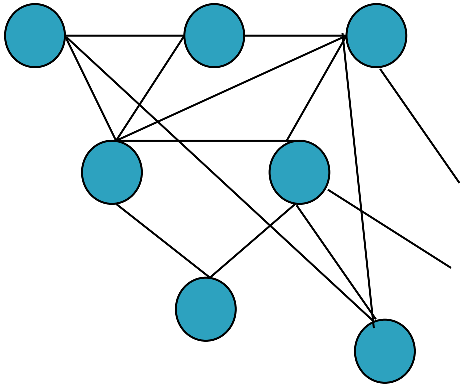
- Similar performance within GHH framework (same total no. of evaluations, same initials, etc.), ILS and VNS are slightly better
- Results are comparable to state-of-the-art approaches on both course and exam benchmark problems

Which Low Level Heuristics?

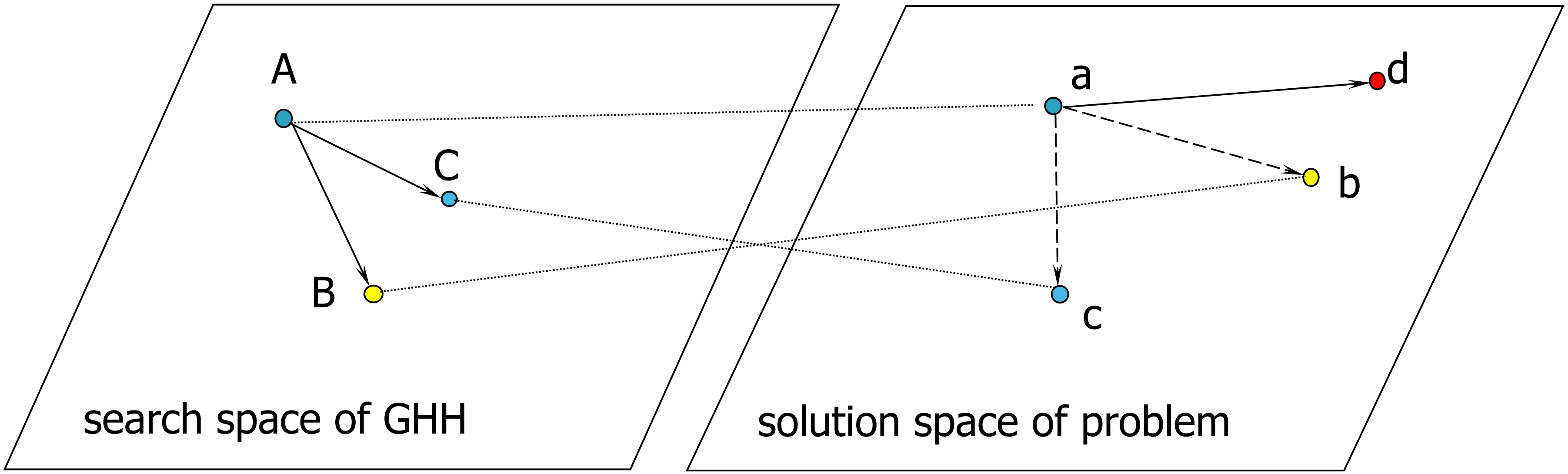
- Within the GHH framework
  - Different subsets of graph heuristics (SD+LD, SD+LWD, SD+LE, SD+LWD+CD, etc.)

  - With a limited computational time: SD + LWD performed the best
  - With more graph heuristics: Longer time given, the better the results
    - h^l (**l**: length of the sequence, **h**: number of graph heuristics)
    - Larger search space, more solutions sampled
  - Random ordering also contributes





Two Search Spaces



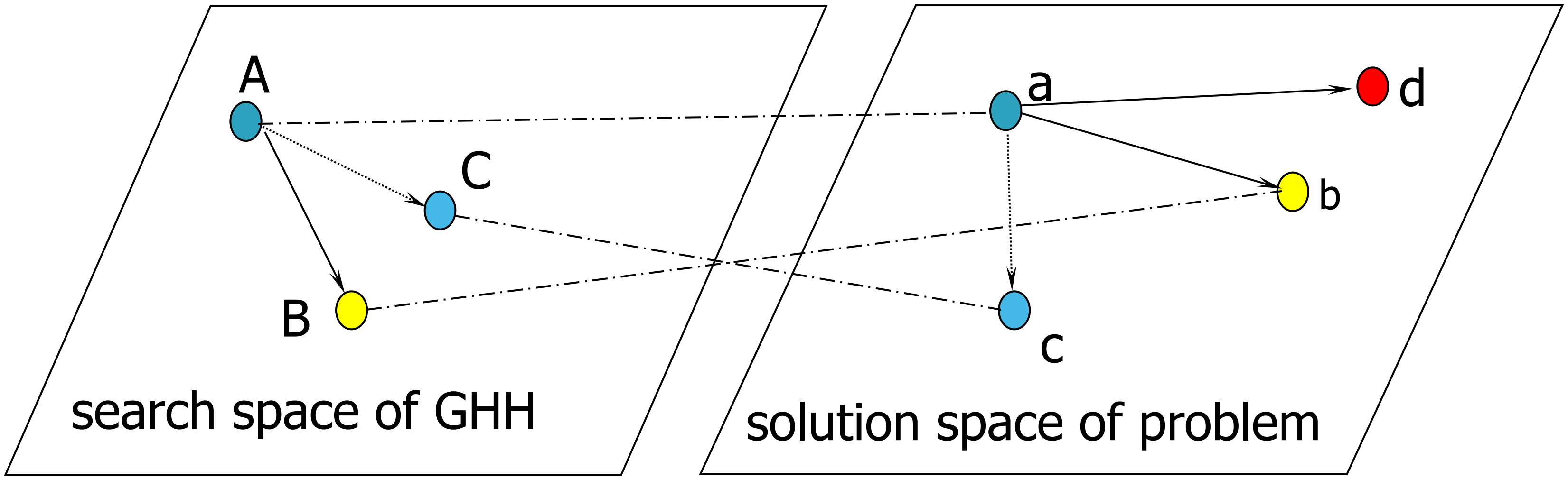
- Search space of high level heuristics: permutations of low level heuristics
- Solution space of problem: actual solutions
- Are all the solutions in solution space reachable?
  - GHH: search is upon heuristics, not solutions

R. Qu and E.K. Burke. Hybridisations within a Graph Based Hyper-heuristic Framework for University Timetabling Problems. JORS, 60: 1273-1285, 2009. Top 5 highly cited paper at JORS 2009-2010

Two Search Spaces

	Heuristic space	Solution space
Representa-tion		
Size (Upper Bound)		
Neighbor-hood Operator		
Objective Function		

Search in Two Spaces



- With one move
  - Local search approaches
  - Graph based hyper-heuristics

One bit different

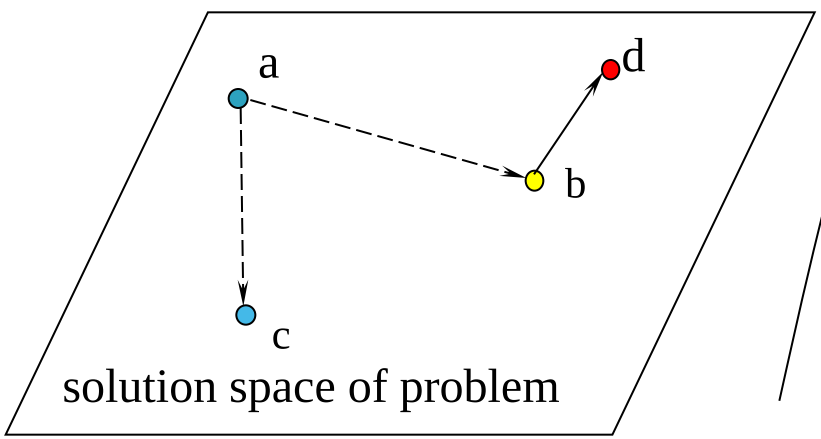
One part different (from different heuristic lists)

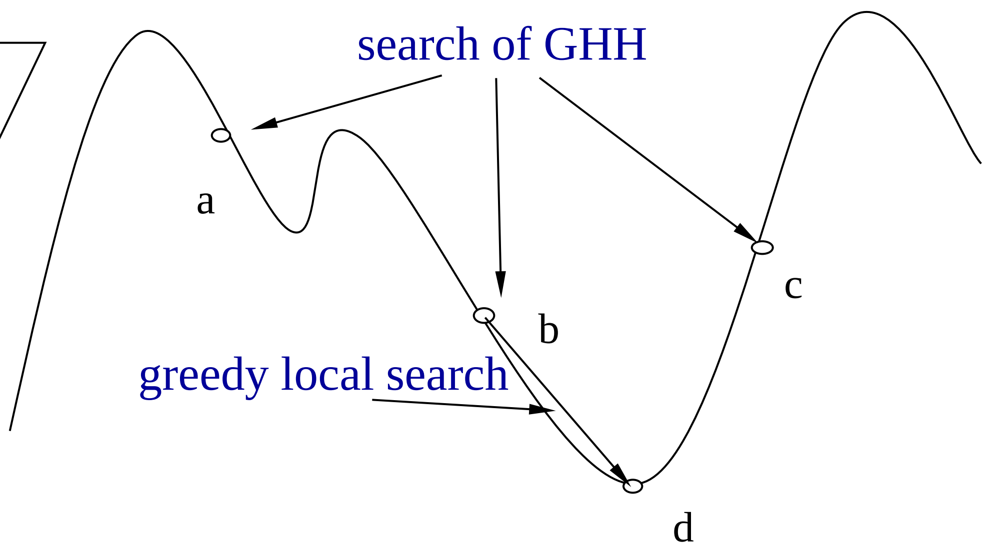
Search in Two Spaces

- Local search based algorithms
  - Move within limited search areas
  - Easily stuck to local optima: different mechanisms developed
  - Search attracted within limited parts of search space
- GHH
  - Change the way of building the solutions at a high level
  - Search space of heuristics -> solutions far from each other in the solution space
  - Key feature: coverage of the solution space
  - GHH vs. VNS?

Search in Two Spaces

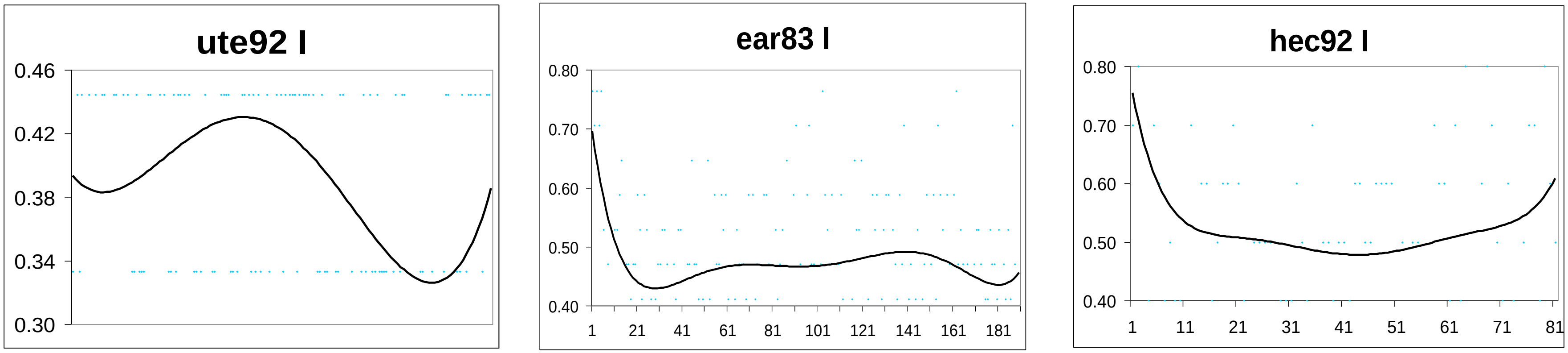
- Hybridisation with greedy local search
  - Coverage of solution space: Results greatly improved!
  - **Diversification** by GHH in the heuristic space, vs.
  - **Intensification** by local search in the solution space
  - Hybrid GHH vs. Memetic Algorithms





Extension I: Adaptive GHH

- Heuristic hybridisations in GHH
  - Knowledge: best solutions → good heuristic hybridisation
  - I - Random GHH (SD+LWD, SD+LE, SD+LD)
    - A large collection of different heuristic sequences
  - II - Analyse the best 5% heuristic sequences
    - Rates of hybridisation at different parts of heuristic sequences
    - Patterns of hybridizations in the best sequences



R. Qu and E. K. Burke. Adaptive Automated Construction of Hybrid Heuristics for Exam Timetabling and Graph Colouring Problems. EJOR, 198(2): 392-404, 2009, Top 10% cited by ISI

Extension I: Adaptive GHH

- Heuristic hybridisations in GHH
  - SD + LWD: better results compared with LE or LD
  - In the best 5% (and 10%) sequences
    - Higher proportion of LWD at early stage
  - No obvious patterns in the worst LWD hybridizations

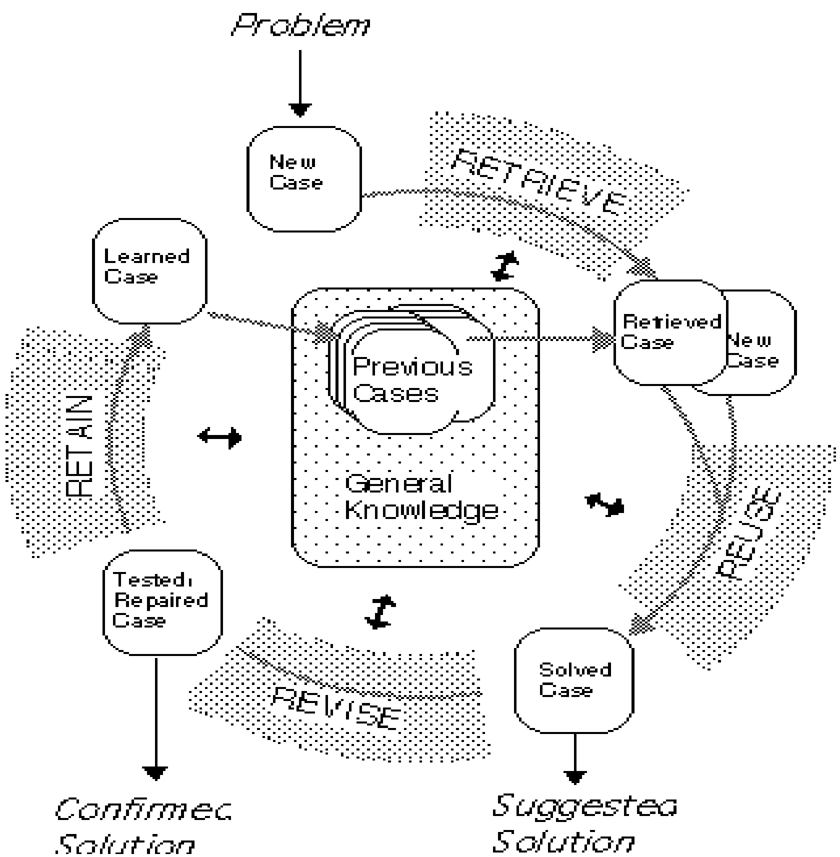
- Adaptive heuristic hybridization
  - GHH: focuses on early sequences
  - Adaptively adjust LWD hybridisation

For *iterations*
    hybridize *a*% of LWD into the first half of ***h***
    produce a solution *s* using ***h***
        If *s* is better or infeasible, increase *a*
        otherwise decrease *a*
        Keep the best ***h*** so far

Extension II: Case Based GHH

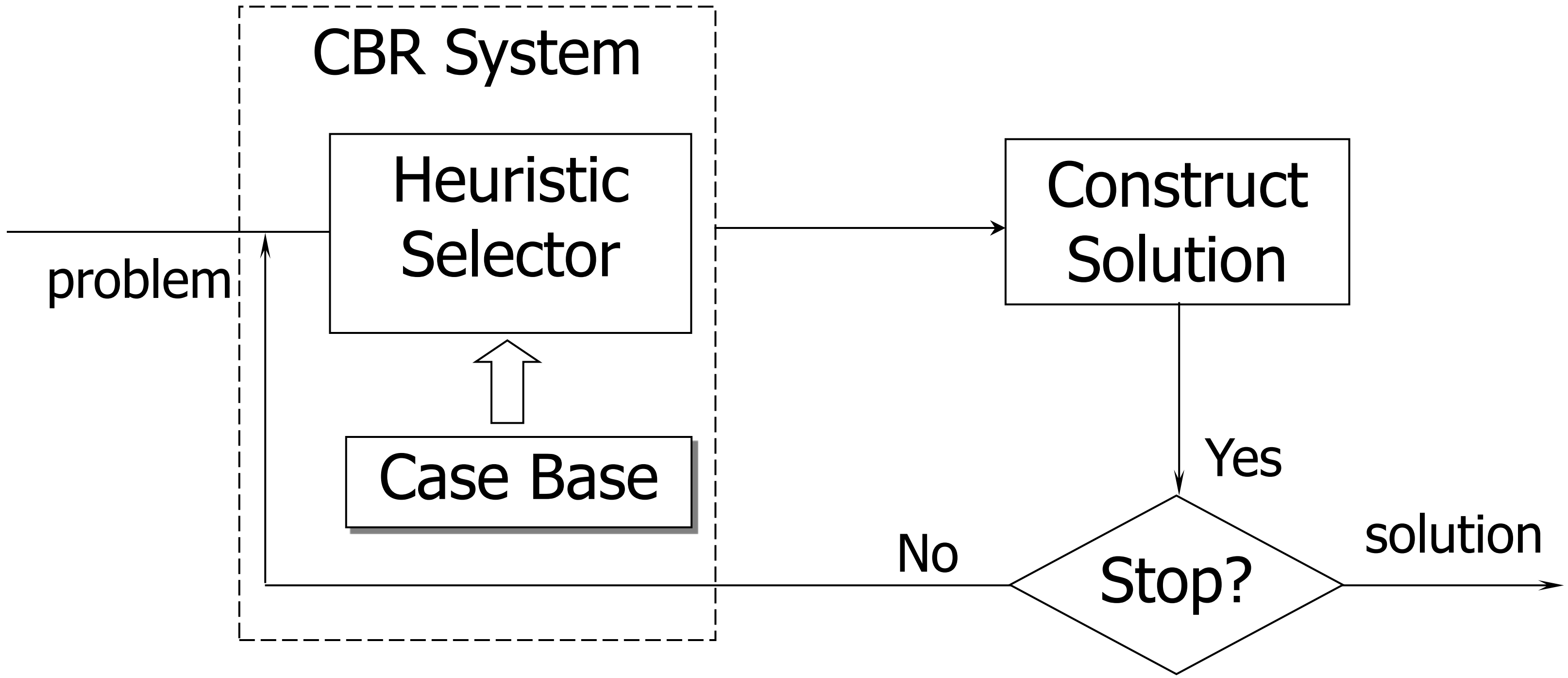
- Extract/record knowledge of heuristic selection during problem solving
- Learn to select and suggest good heuristics for particular situations
- Obtained good results on simulated problems, and test on real-world problems

- Assumption:
similar problems
similar solutions



R. Qu, co-authors: E. Burke, S. Petrovic, Case Based Heuristic Selection for Timetabling Problems. Journal of Scheduling, 9: 115-132, 2006. Top 1% cited by ISI.

Extension II: Case Based GHH



Extension II: Case Based GHH

- CBR: suggests good heuristics that worked well in previous similar situations employing knowledge stored in the system
- Case base
  - problems and their partial solutions during problem solving
  - best heuristics for that situations
- Similarity measure: nearest neighbourhood approach
- Key issue of meaningful comparison between two problem solving situations
  - features describe the characteristics of problem and partial solution (cases)

Extension II: Case Based GHH

- Analysis on all possible features
- Training process on feature list
  - Search for most relevant features by which cases (problems and problem solving situations) can be compared concerning most appropriate heuristics used
  - Tabu search
- Training process on cases in case base
  - Leave-one-out strategy: refine the cases stored in case base for problem solving
  - Only cases that may make contribution to problem solving are retained

Extension II: Case Based GHH

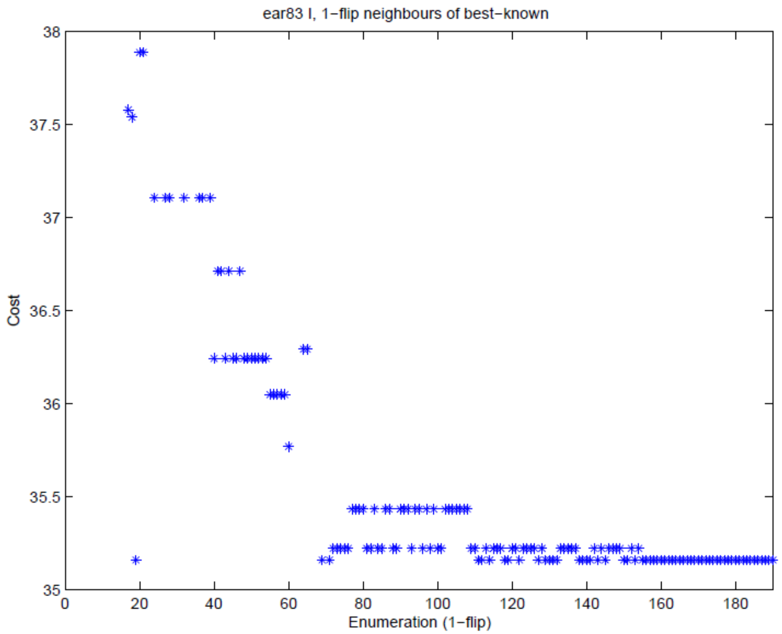
- Observations
  - the more features, the better?
  - features selected are more important than their weights in the similarity measure
  - search methods for the feature list are not crucial

  - vs. graph based hyper-heuristics
  - not an easy task for selecting the best meta-heuristics to solve the whole problem

Extension III: GHH Landscape

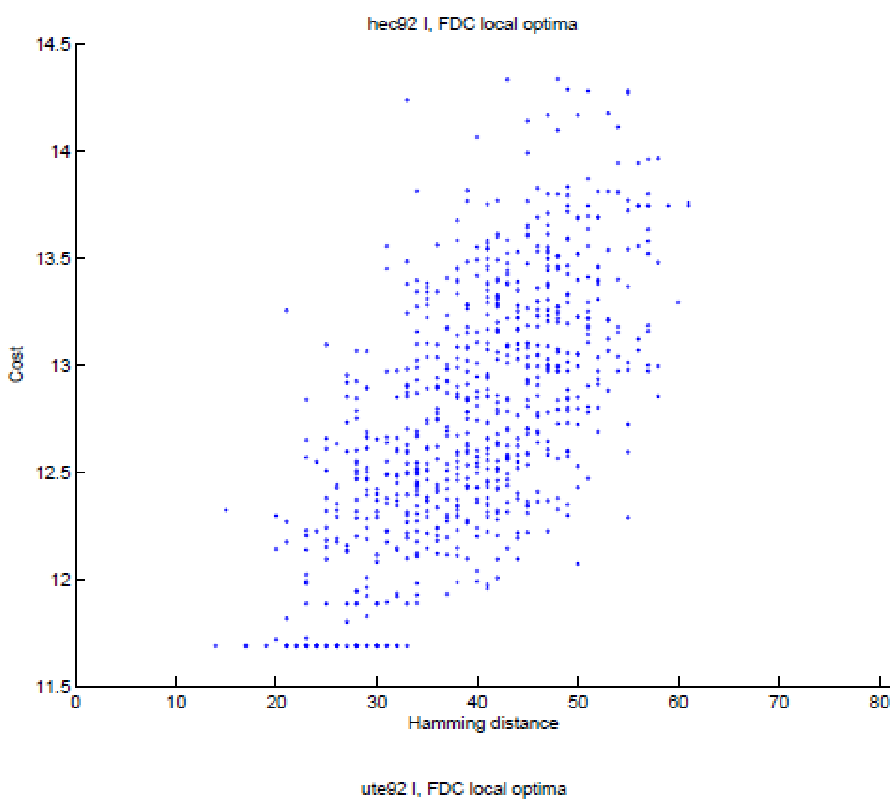
- Landscape of high level heuristic space
  - More likely to have “walks” or plateau
    - Not mapped to all solutions in solution space (hypothesis)
    - Size of neighbourhoods is very large
    - Computational time: limited number of evaluations within a limited time

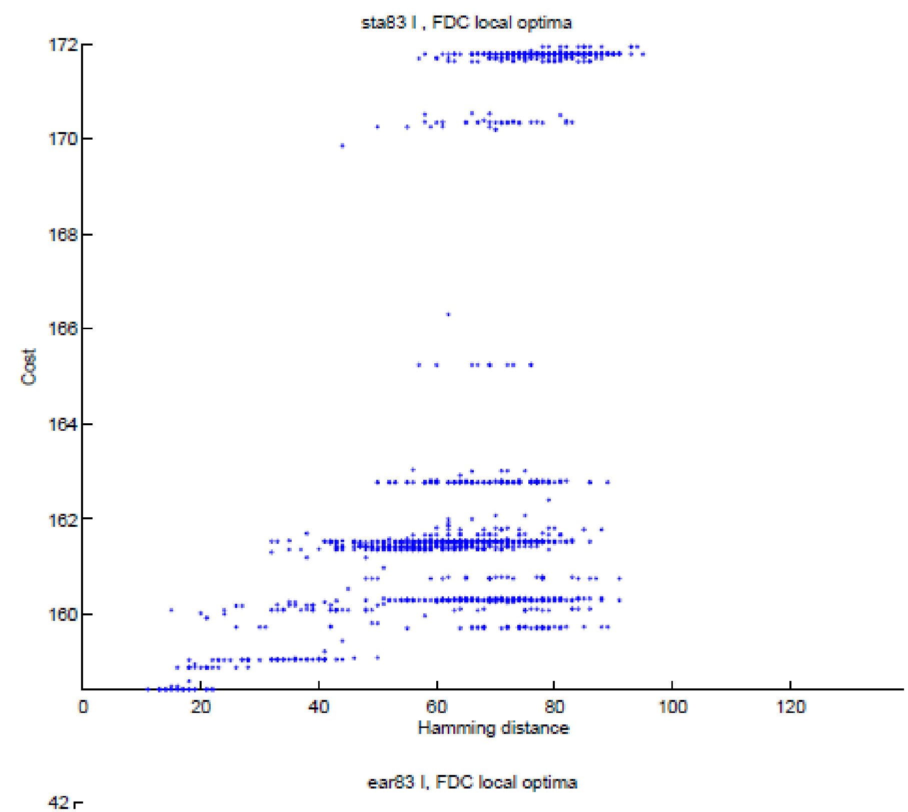
    - 1-flip on a heuristic list
    - Fitness distance correlation (fdc): local optimal vs. best



Extension III: GHH Landscape

- Landscape of high level heuristic space





Other Extensions

- Landscape of high level heuristic space
  - More likely to have plateau (neutral)
- Synchronise the search in two search spaces
  - Difficulty of landscape analysis in solution space
- Other recent extensions in the literature
  - Hierarchical hybridisation of graph heuristics
  - …
- More details available at: http://www.cs.nott.ac.uk/~rxq/publications.htm