Hybridising ConstructiveHeuristics within Hyper-heuristic Frameworks

Rong Qu

Automated Scheduling, Optimisation and Planning Group
School of Computer Science
University of Nottingham

Background

- Constructive heuristics in scheduling
  - Job shop scheduling: dispatching rules
  - Timetabling: graph heuristics
  - Bin packing: 2D/3D packing heuristics

  - Simple and fast

- In complex scheduling problems, using only the basic constructive heuristics often produce unacceptable solutions
- Automated hybridisation / combination of simple heuristics

Background

- Timetabling problems
  - To assign a set of events, which requires different resources, into limited number of time slots, minimising violations of soft constraints

  - Hard constraints
    - Conflicted events in different time slots
    - Room capacity to hold the events
    - etc
  - Soft constraints
    - Spread out events over time slots / at least n events or no event on a day
    - No event scheduled on specific time slots
    - etc

Background

- Timetabling problems
  - Exact methods
    - IP/MILP
  - Constructive heuristics
    - Graph heuristics
    - Constraint satisfaction
    - etc
  - Meta-heuristics
    - Local search based algorithms
    - Population based algorithms
    - Hybridisations
    - etc

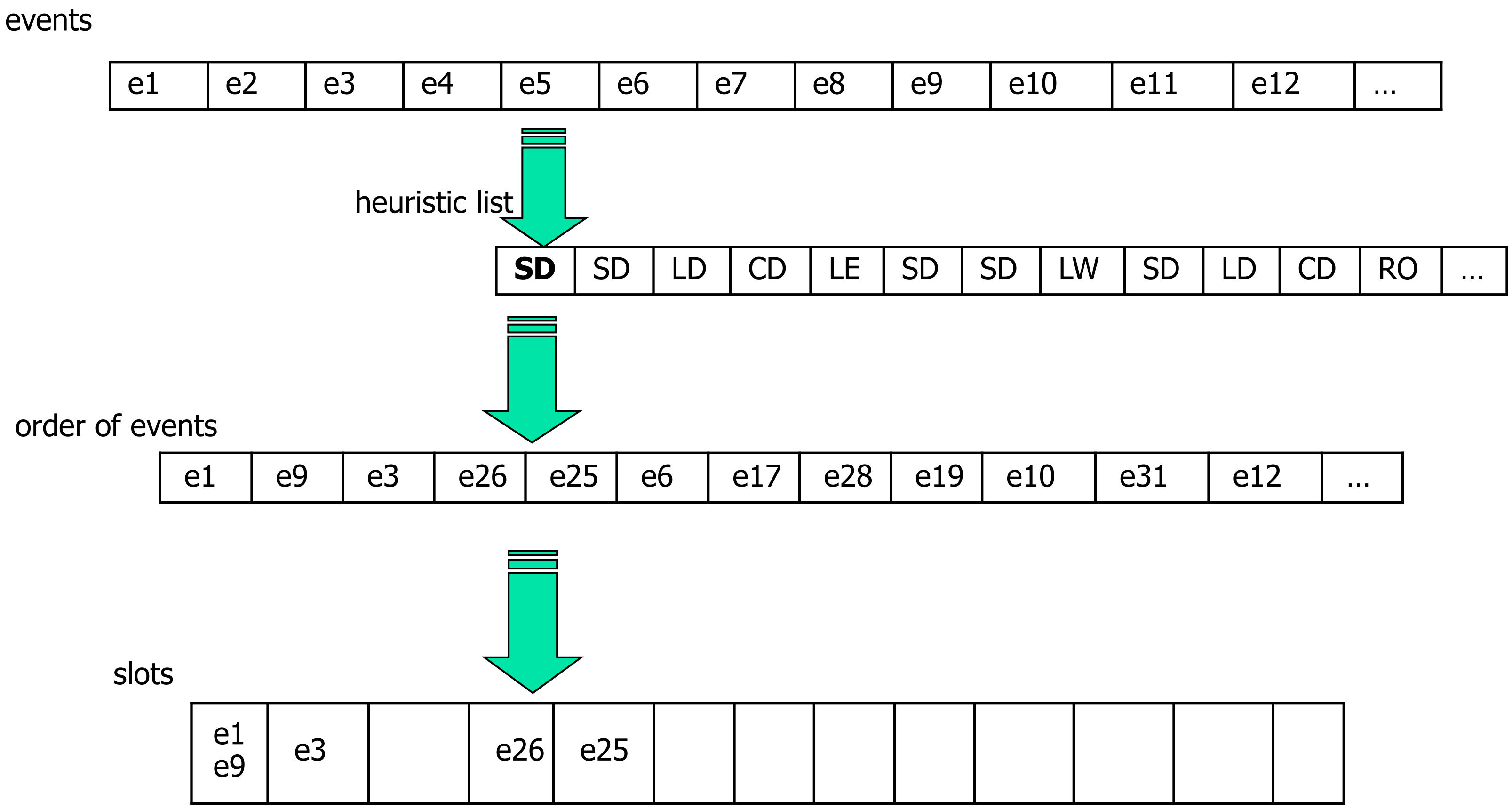
The Framework

- The high level framework
  - Any meta-heuristics or learning/search methodology

- The low level graph 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 involved
  - Largest Enrolment: students enrolled

- Hyper-heuristics
  - Heuristics to choose heuristics

The Framework



events

e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12	…

heuristic list

SD	SD	LD	CD	LE	SD	SD	LW	SD	LD	CD	RO	…

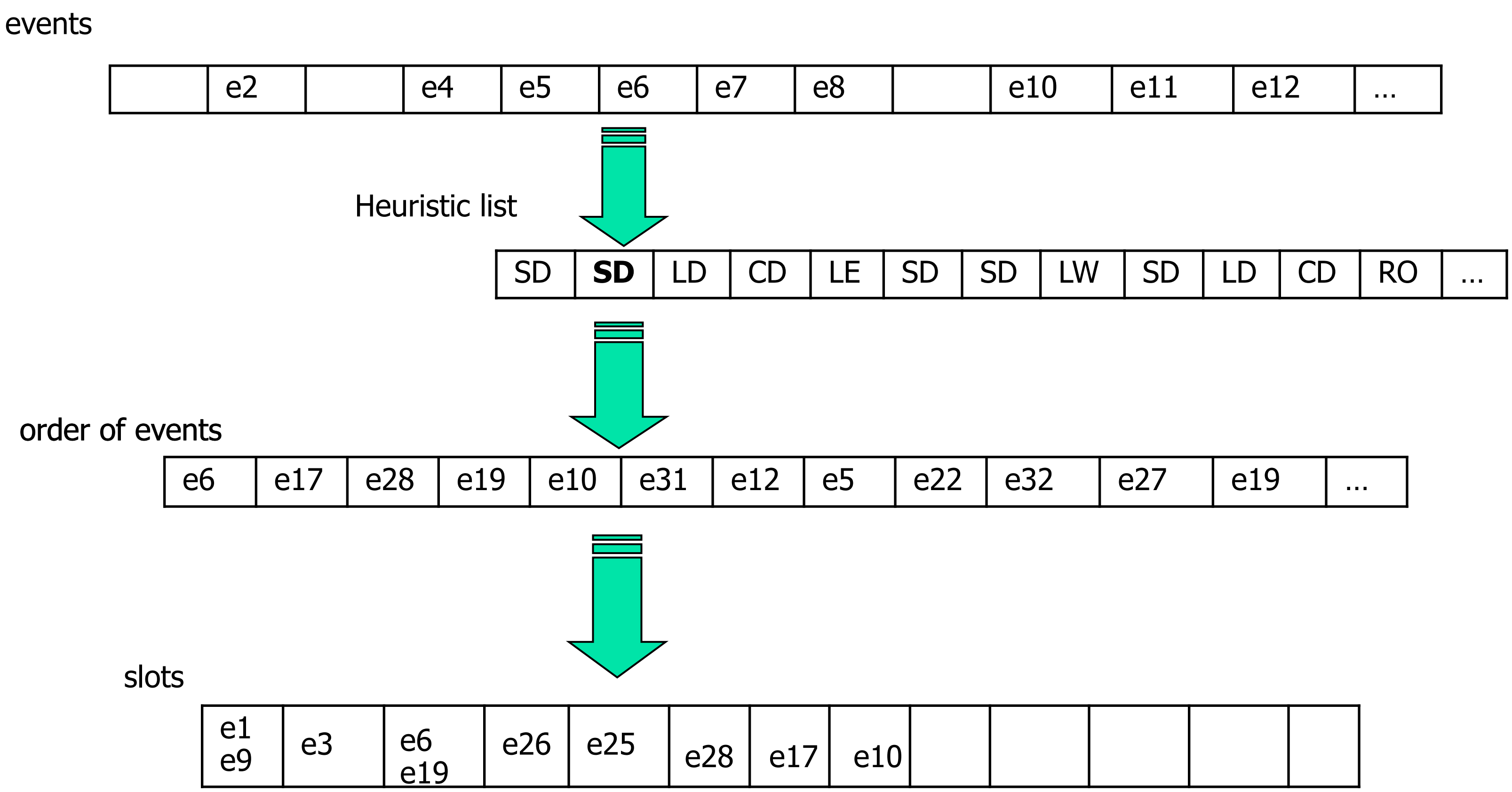
order of events

e1	e9	e3	e26	e25	e6	e17	e28	e19	e10	e31	e12	…

slots

e1 e9	e3		e26	e25								

The Framework



events

	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 events

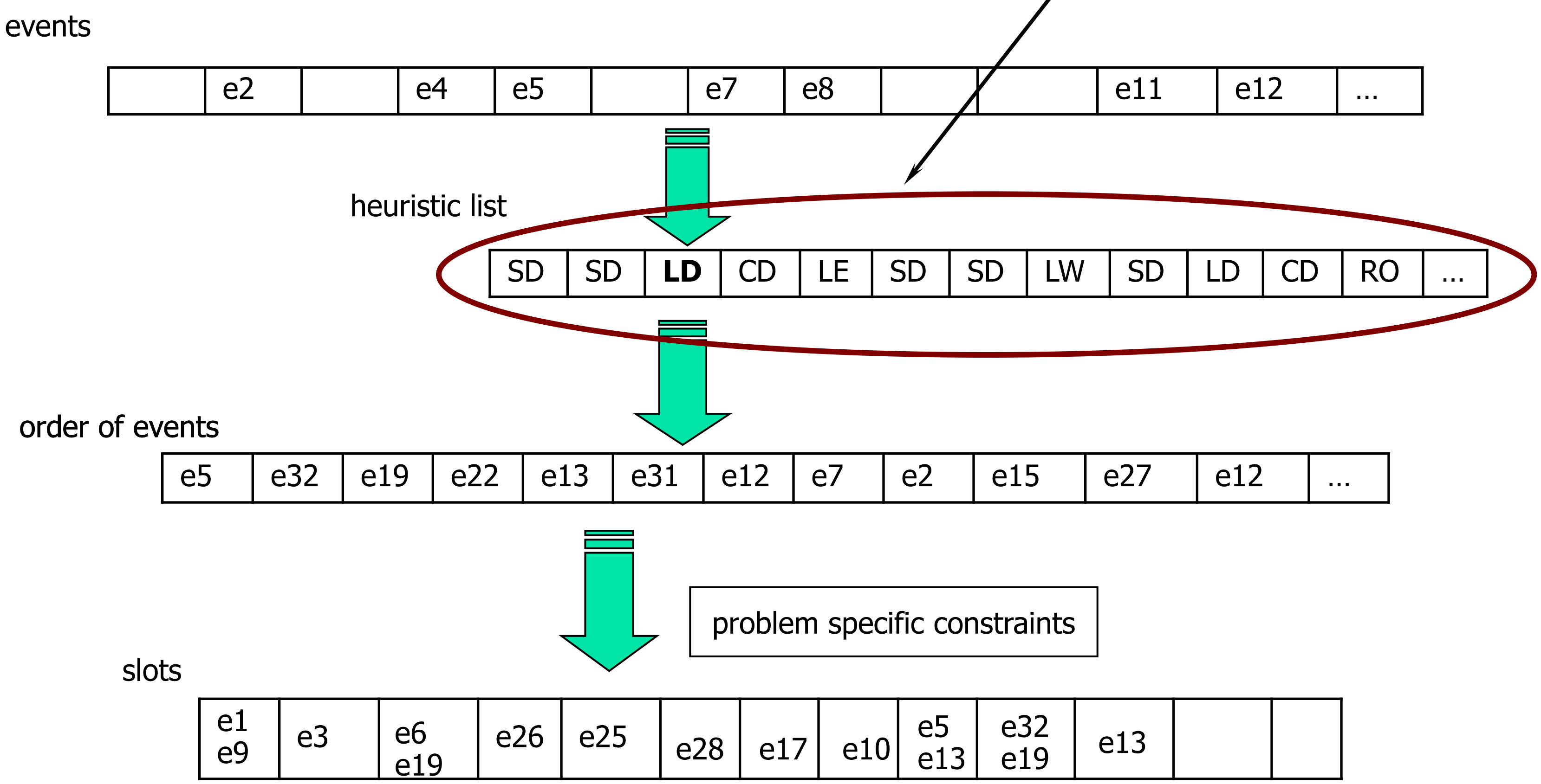
e6	e17	e28	e19	e10	e31	e12	e5	e22	e32	e27	e19	…

slots

e1 e9	e3	e6 e19	e26	e25	e28	e17	e10					

The Framework

High level search



events

	e2		e4	e5		e7	e8			e11	e12	…

heuristic list

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

order of events

e5	e32	e19	e22	e13	e31	e12	e7	e2	e15	e27	e12	…

problem specific constraints

slots

e1 e9	e3	e6 e19	e26	e25	e28	e17	e10	e5 e13	e32 e19	e13		

Research Questions/Issues

- Which high/low level search heuristics?
- Search in two search spaces
- Heuristic hybridisations
- Landscape analysis on heuristic spaces
- Extensions on the framework and other problems

Which high level search method?

- 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

Which high level search method?

- High level search methods

	car91	car92	ear83	hec92	kfu93	lse91	sta83	tre92	ute92	uta92	yor83
SDM avg	6.18	5.3	***36.8***	12.74	15.63	13.51	163.7	9.37	32.6	4.5	43.6
ILS avg	***6.01***	5.18	39.58	13.01	15.35	13.1	***161.6***	8.92	***31.3***	***4.01***	***43.15***
TS avg	6.3	5.34	45.56	14.6	19.55	14.29	169.1	9.67	37.02	4.38	47.97
VNS avg	6.1	***5.1***	38.63	***12.72***	***15.24***	***13.06***	163.3	***8.88***	31.7	4.05	43.93

Which high level search method?

- High level search methods

	s1	s2	s3	s4	s5	m1	M2	m3	m4	m5	l
SDM avg	10.8	15.6	***5***	11.8	12.2	382.5	100%	383	***374.5***	194.5	100%
ILS avg	***8.8***	***13.2***	5.4	***7.6***	12	375	480.5	377.5	380.5	179.7	***1144*** **60%**
TS avg	12.2	16.4	9.2	12.2	18.2	511.5	533 80%	468	539	236	1164 80%
VNS avg	10	14.8	5.2	8	***10.6***	***365***	***443*** **40%**	***369.5***	377.5	***165.5***	1148 80%

Which high level search method?

- High level search methods
  - Similar performance within the same framework (same total number 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

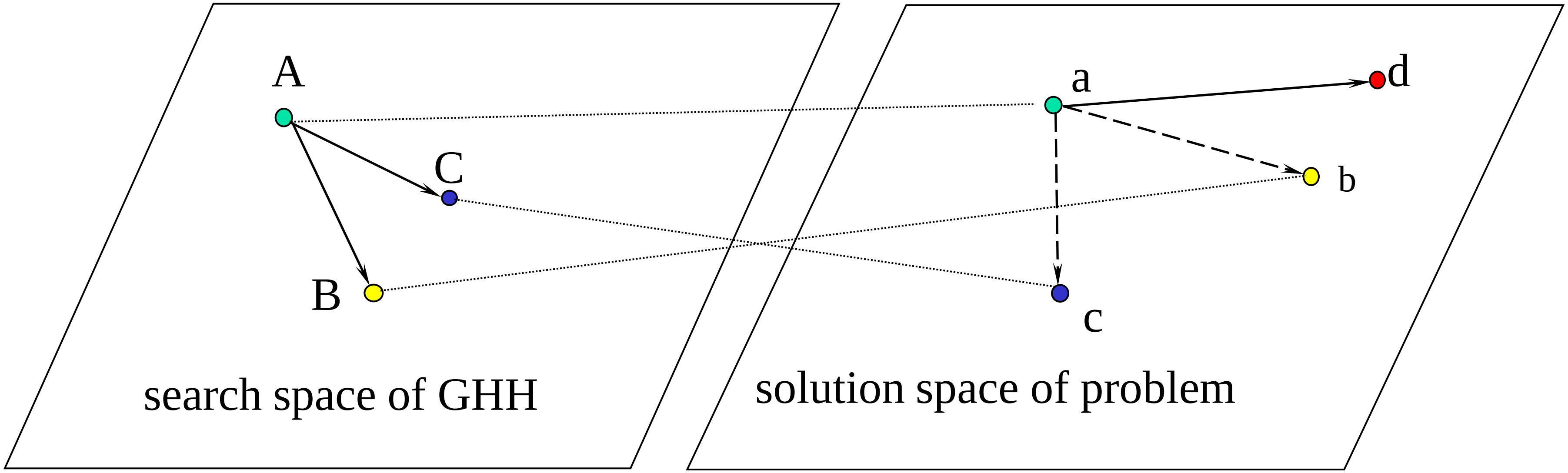
Which low level heuristics?

- Within the 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
  - Random ordering also contributes the performance

Two Search Spaces

Heuristic space　　　　Solution space



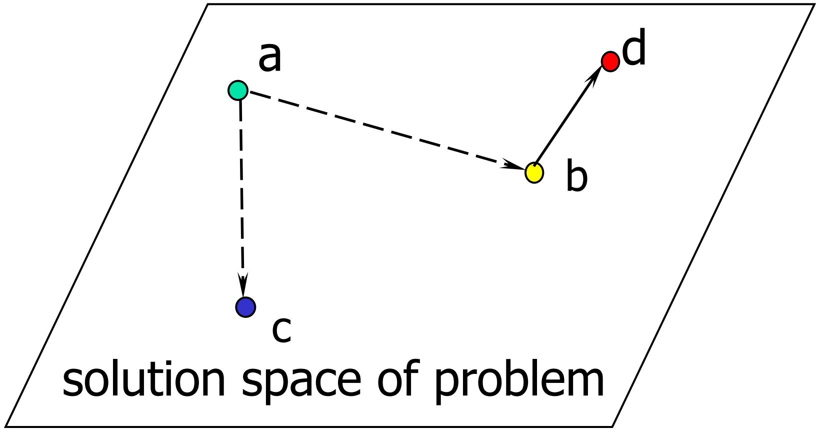
GHH: search is upon heuristics, not solutions
– not all the solutions in solution space are reachable?

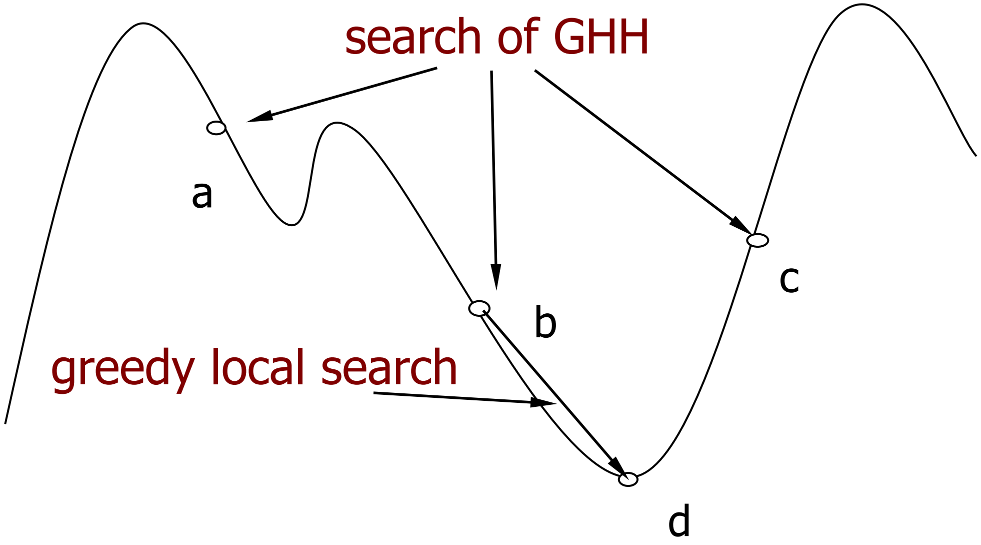
Two Search Spaces

	Heuristic space	Solution space
Representation		
Size (Upper Bound)		
Neighborhood Operator		
Objective Function		

Two Search Spaces

- Hybridisation in the framework with simple greedy search
  - High level search in heuristic space: a, b, c, …
  - Greedy search in solution space: b -> d, …
    - Coverage of the solution space





Two Search Spaces

- Hybridisation in the framework with simple greedy search
  - Results greatly improved!

	car91	car92	ear83	hec92	kfu93	lse91	sta83	tre92	ute92	uta92	yor83
GHH2 best	5.16	4.16	35.86	11.94	14.79	11.15	159	8.6	28.3	3.59	41.81
GHH2 avg	**5.21**	**4.20**	**36.2**	**12.1**	**15.01**	**11.24**	**160.81**	**8.65**	**28.64**	**3.62**	**41.96**
GHH2 time	26001	11666	740	105	3417	2015	128	2293	131	10045	641
GHH1 best	5.3	4.77	38.39	12.01	15.09	12.72	159.2	8.74	30.32	3.42	40.24
GHH1 avg	6.01	5.18	39.58	12.33	15.35	13.1	161.6	9.0	31.3	4.01	43.15
GHH1 time	13684	6553	462	70	1887	1125	72	1433	101	5429	340

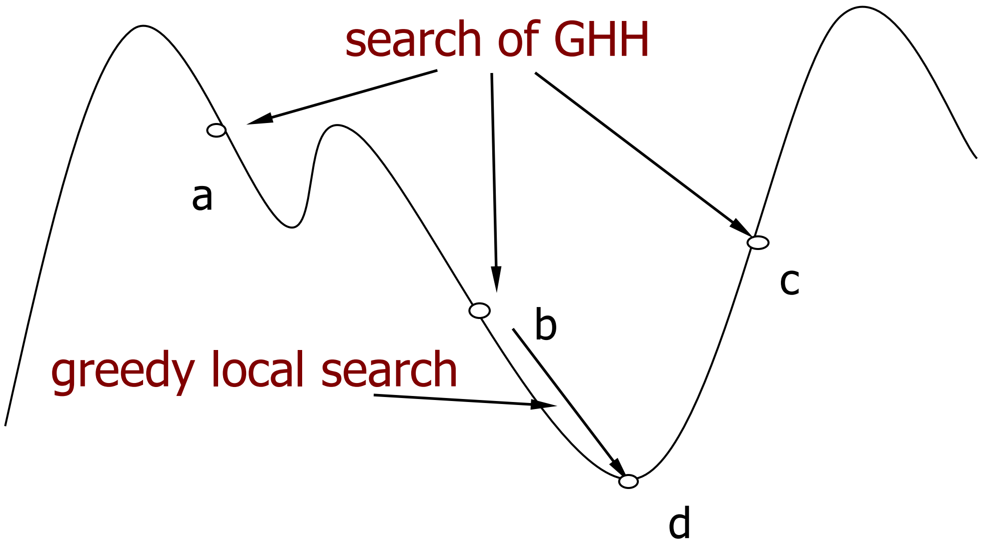
Two Search Spaces

- Hybridisation in the framework with simple greedy search
  - Results greatly improved!

	s1	s2	s3	s4	s5	m1	m2	m3	m4	m5	l
GHH2 best	0	0	0	0	0	257	259	192	235	112	0.8/1132
GHH2 avg	***0.2***	***0.6***	***0***	***0.4***	***0.1***	***261***	***273***	***214.5***	***242***	***116***	***1135***
GHH2 time	50	54	48	45	65	19411	15750	18512	18782	9725	20328
GHH1 best	2	2	1	1	0	310	419	332	324	162	0.8/1162
GHH1 avg	2.6	2.8	1	3	2.6	323	428	345	335	182	1162
GHH1 time	155	218	240	171	260	62115	50403	57387	65821	36955	81148

Two Search Spaces

- Hybridisation in the framework with simple greedy search
  - Hybrid GHH vs. Memetic Algorithms
    - Diversification vs. intensification



Two Search Spaces

- Search in two search spaces
  - Diversification of the high level search in the framework in the heuristic space
  - Intensification by the local search in the solution space

- Role of high level search methods

  To explore diversified solutions in the solution space by searching in the high level heuristic space

Heuristic Hybridisations

- How to (adaptively) hybridise heuristics?

  Knowledge / lesson learnt from the offline heuristic hybridisations?

  I – Random (SD+LWD, SD+LE, SD+LD)

  A large collection of different heuristic sequences

  Systematically produce heuristic sequences

  Full coverage of different amount of hybridisations

Heuristic Hybridisations

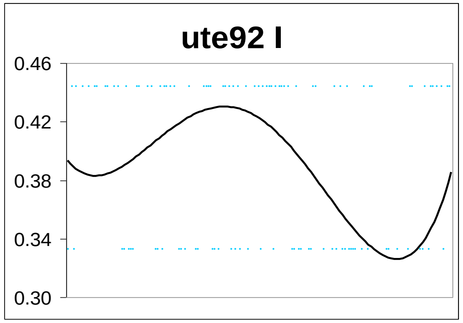
- How to (adaptively) hybridise heuristics?

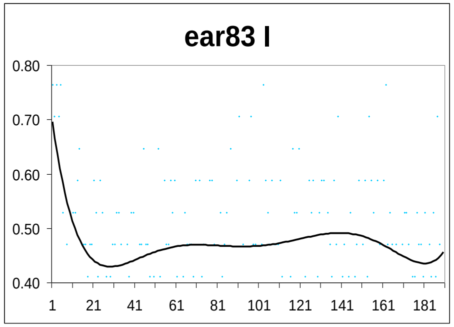
  Knowledge / lesson learnt from the offline heuristic hybridisations?

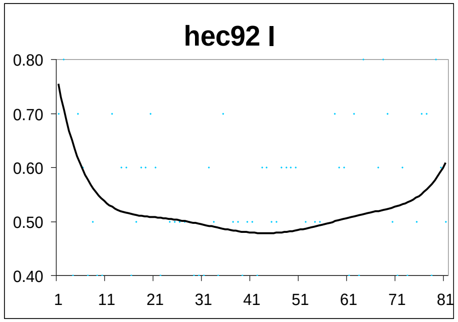
  II – Analyze the best/worst 5% heuristic sequences

  Rates of hybridisation at different positions of heuristic sequences

  Trends of hybridizations in the best sequences





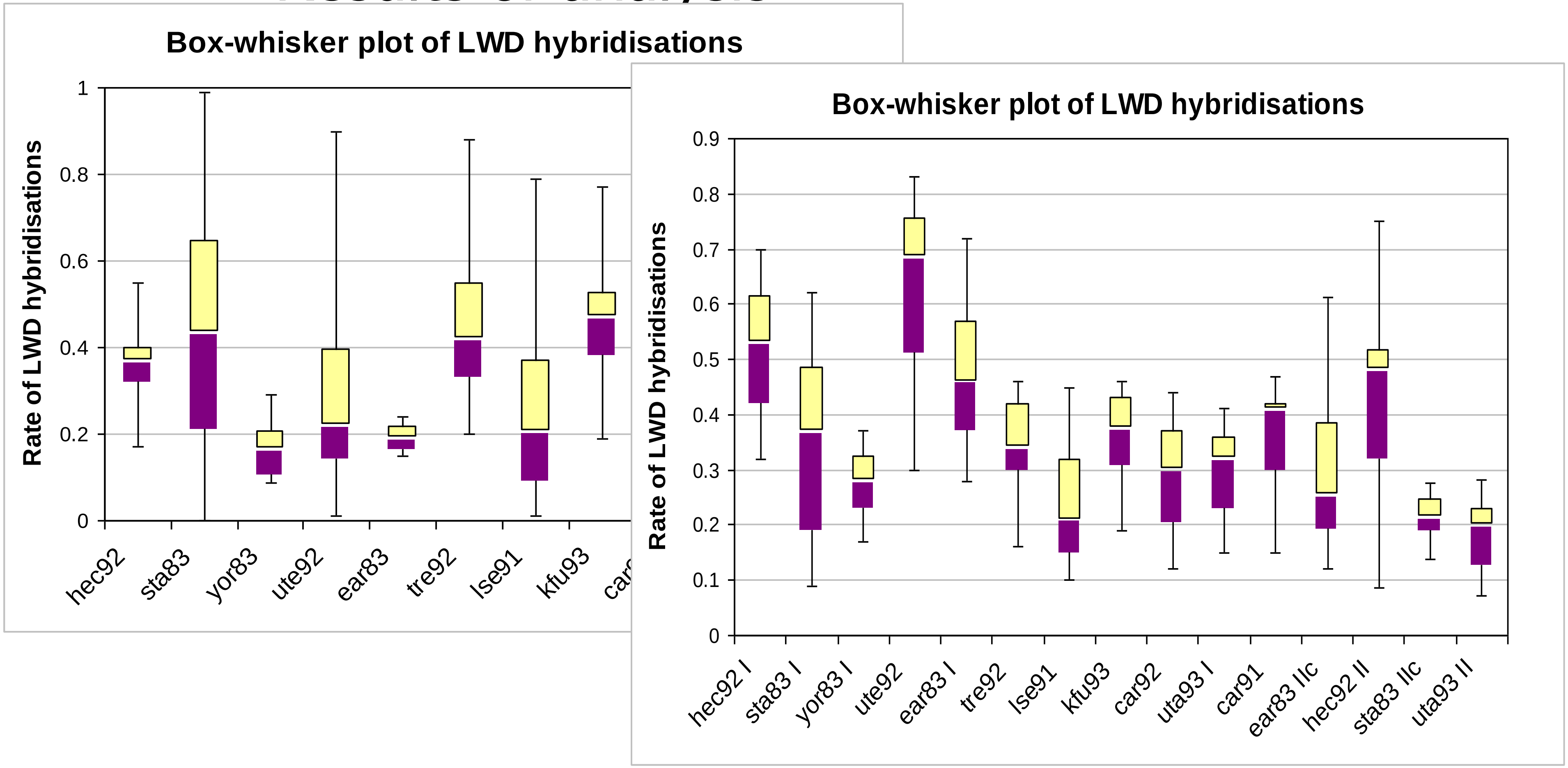


Heuristic Hybridisations

- Results of analysis
  - Hybridising SD with LWD obtained better results compared with LE or LD
  - In the best 5% sequences
    - Higher percentage at early stage
    - High level of vibrancy at early stage
  - No obvious trends in the worst 5% sequences

Heuristic Hybridisations

- Results of analysis



Heuristic Hybridisations

- Adaptive online heuristic hybridization
  - Focus on early stage of heuristic hybridization
  - Rate of LWD hybridisation adaptively adjusted

ETTP	car91	car92	ear83	hec92	kfu93	lse91	sta83	tre92	ute92	uta92	yor83
AGH	**5.17**	**4.32**	**35.7**	**11.93**	15.34	11.45	159.05	**8.68**	28.88	**3.3**	**40.79**
%LWD	27	33	50	40	30	16	18	25	26	29	58
GHH	5.36	4.93	37.92	12.25	**15.2**	**11.33**	**158.19**	8.92	**28.01**	3.88	41.37
RGH	5.37	4.5	36.51	12.08	15.95	11.48	159.58	8.77	28.69	3.4	41.79
RGH20%	5.38	4.5	39.02	12.19	15.6	11.5	159.08	8.73	29.16	3.63	41.52

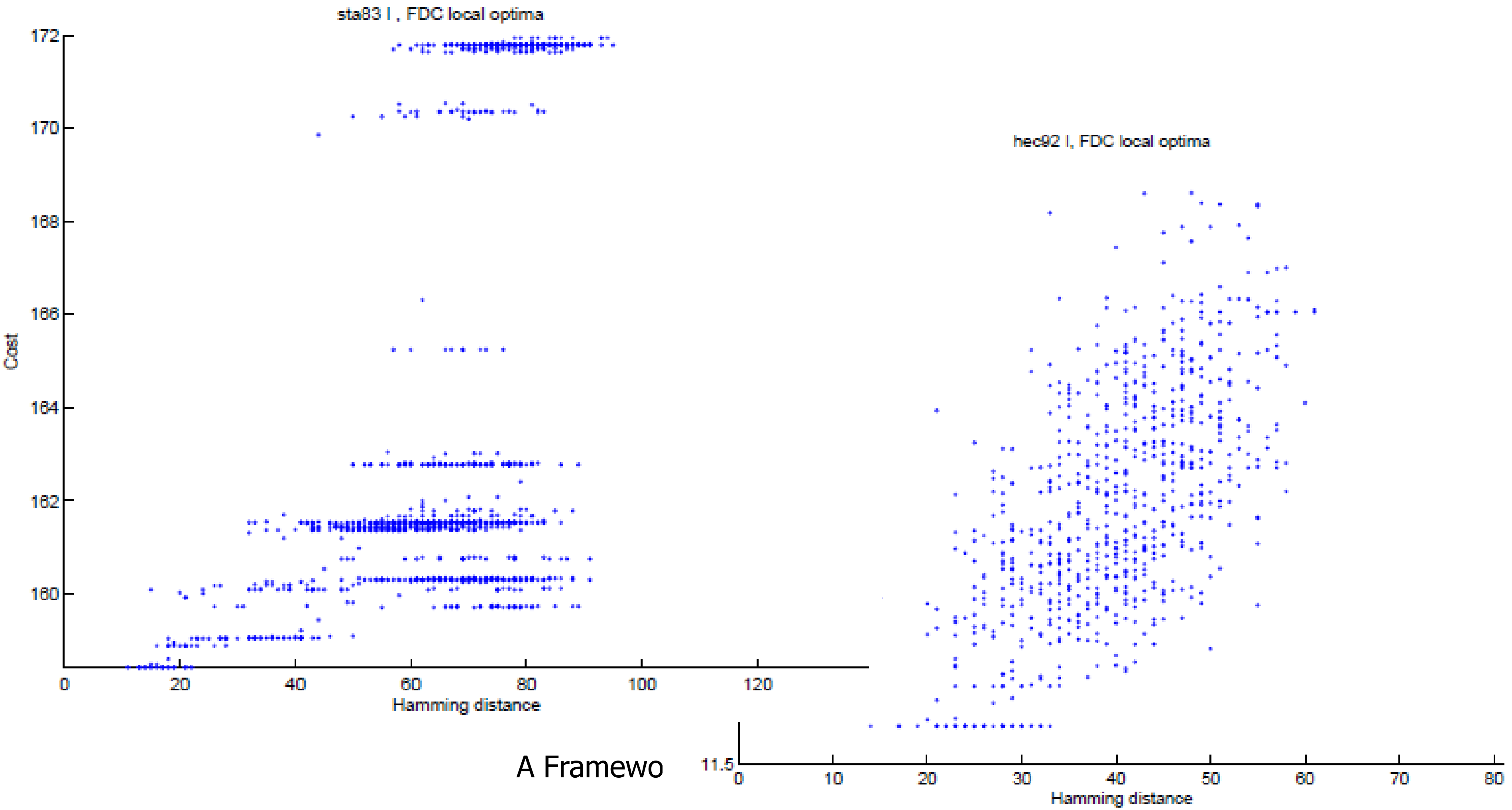
GCP	car91	car92	ear83	hec92	kfu93	lse91	sta83	tre92	ute92	uta92	yor83
AGH	30	29	**22**	**17**	**19**	**17**	**13**	**20**	**10**	31	**19**
%LWD	15	16	15	16	30	30	25	15	30	30	17
RGH	30	29	**22**	18	**19**	**17**	**13**	**20**	**10**	31	20
%LWD	36	13	19	37	46	28	49	45	34	24	17
best	**28**	**28**	**22**	**17**	**19**	**17**	**13**	**20**	**10**	**30**	**19**

Landscape of Heuristic Space

- Understanding the structure of heuristic search spaces, i.e. heuristic sequences vs. solutions
- Fitness landscape analysis on constructive hyper-heuristics
  - Fitness distance correlation (*fdc*) of local optima to the global optimum
  - One-flip of global optimum
  - Correlation length

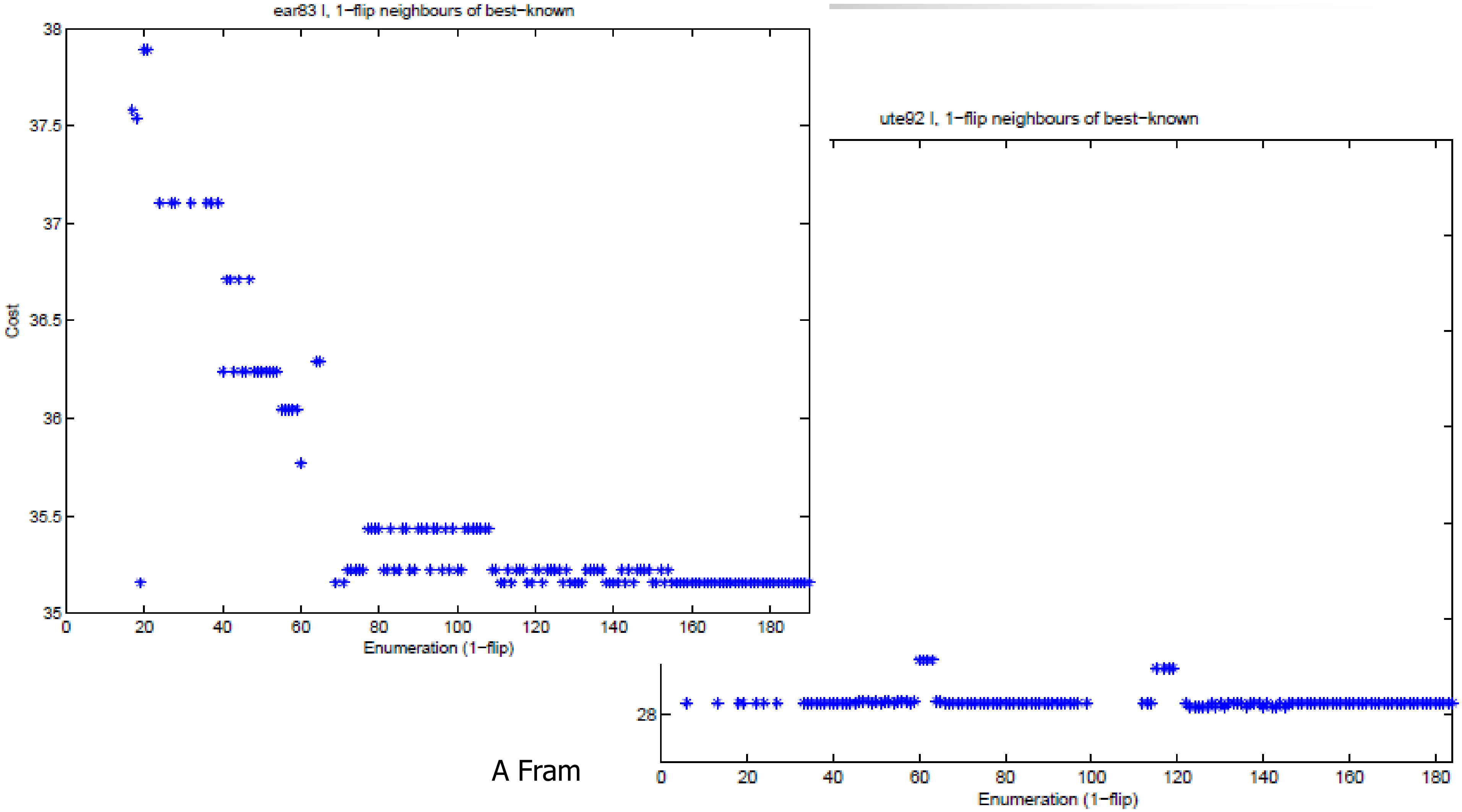
- Although rugged, the encouraging feature of a globally big valley structure
- A high level of neutrality and positional bias

Landscape of Heuristic Space



A Framewo

Landscape of Heuristic Space



A Fram

Extensions & Future work

- 3D strip packing, EDA learning at the high level
- Synchronise the search in two search spaces
  - Co-ordinate search in heuristic/solution space
  - Need to improve every solution?
- Other recent extensions in the literature
  - Hierarchical hybridisation of graph heuristics
  - Tie-breaking mechanism
- …

References

- E. Burke, A. Meisels, S. Petrovic and R. Qu A Graph-based Hyper-Heuristic for Exam Timetabling Problems. *EJOR*, 176: 177-192, 2007
- R. Qu and E. K. Burke. Hybridisations within a Graph Based Hyper-heuristic Framework for University Timetabling Problems. *JORS*, 60, 1273-1285, 2009
- R. Qu, E. Burke and B. McCollum. Adaptive Automated Construction of Hybrid Heuristics for Exam Timetabling and Graph Colouring Problems. *EJOR*, 198(2): 392-404, 2009
- E. Burke, S. Petrovic, R. Qu, Case Based Heuristic Selection for Timetabling Problems. accepted by *Journal of Scheduling*, 9: 115-132, 2006
- G. Ochoa, R. Qu and E.K. Burke, Analyzing the Landscape of a Graph Based Hyper-heuristic for Timetabling Problems, GECCO'09, pp. 341-348, 8-12 July, Montreal, Canada

- More details at: http://www.cs.nott.ac.uk/~rxq/publications.htm

Appendix – Random GHH

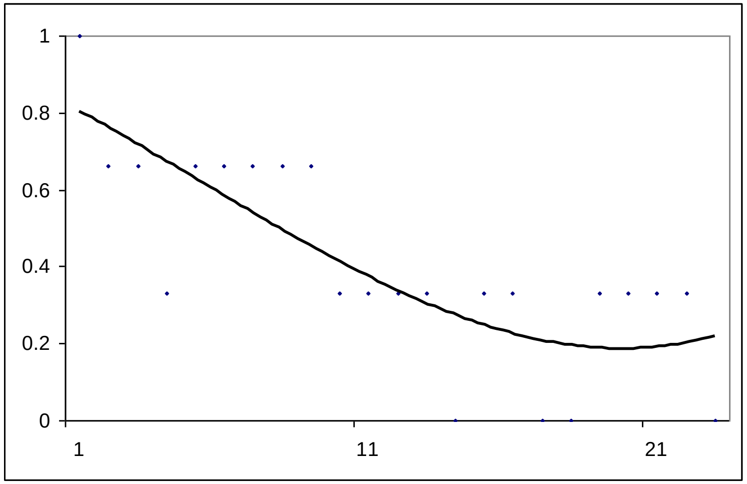
- Systematically produce heuristic sequences
- Full coverage of different amount of hybridisations
- Results: A collection of heuristic sequences

```
For (i = 0; i < length of heuristic sequence; i++)
    Initial heuristic sequences H = SD SD … SD
    Change i SDs in H into LWD, LD or LE
    save H and its quality
```

Appendix – Analyzing Heuristic Sequences

- Under MS Excel
  - Rank the heuristic sequences into *best*, *worst* and *medium* by their quality
  - Get the percentages of LWD, LE or LD at each position of the sequences
  - Draw the trends of hybridization upon the percentage

LWD	LWD	SD	…	…	…	…	SD	12.3
LWD	SD	LWD	…	…	…	…	SD	12.35
LWD	LWD	LWD	…	…	…	…	LWD	12.35
1	0.67	0.67	…	…	…	…	0.33	



Appendix – Adaptive Hybridizations of Heuristics

(1) Based on heuristic sequence of SDs

- Adaptively adjust the amount of LWD hybridization at the early part of sequences
- Iterative process adjusting the amount of LWD hybridization by the quality of previous hybridization

(2) Randomly hybridize LWD upon the whole sequences

- Further improve the sequences

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