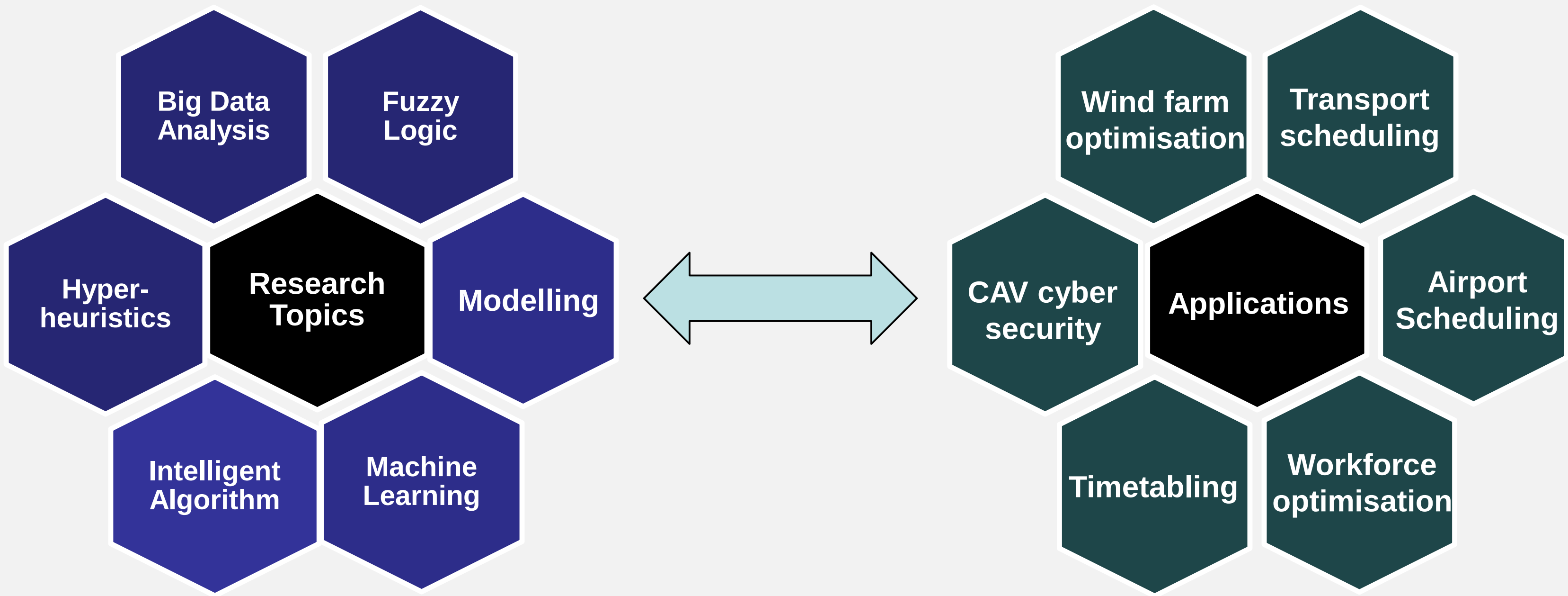
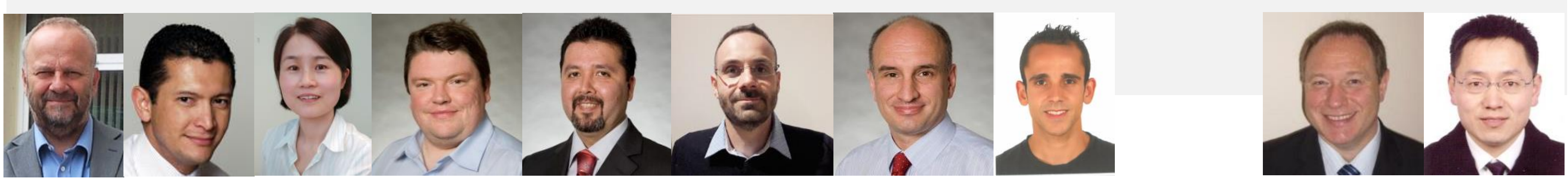
# Towards a Better Understanding of Search Algorithms

**Dr Rong Qu**
**Optimisation and Learning Lab**
School of Computer Science
The University of Nottingham
rong.qu@nottingham.ac.uk

# COL Lab – Research & Applications



The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

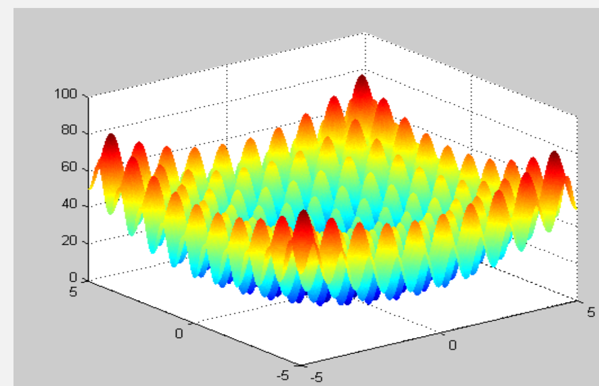
# Algorithm Design

## - An Overview

Computational Optimisation & Learning Lab

# Search Algorithms

- Decisions to make when designing algorithms
  - Algorithm specific decisions
    - Simulated annealing, Tabu search, Variable neighbourhood search, etc.
    - Genetic algorithms, Estimation of distribution algorithm, Swarm Algorithms (i.e. Particle swarm optimisation, Ant colony, etc.)
  - General decisions
    - Solution representation
    - Evaluation function
    - Initialisation
    - Stopping condition
    - Acceptance criteria




Computational Optimisation & Learning Lab

# Search Algorithms

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

- Recent / advanced research developments
  - Integration of computational intelligence techniques
    - Hybridisation of evolutionary and local search algorithms
    - Machine learning and optimisation
    - Data-driven optimisation
    - Hyper-heuristics
  - Automated algorithm design
    - Automated configuration
    - Automated selection
    - Automated composition

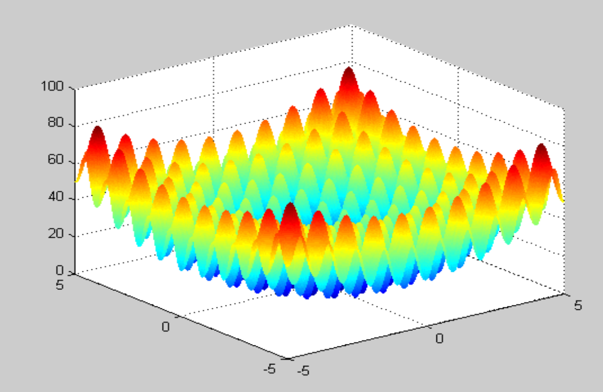
R. Qu, G. Kendall, N. Pillay. (2019) The General Combinatorial Optimisation Problem: Towards Automated Algorithm Design. accepted by **Computational Intelligence Magazine,** 2019

Computational Optimisation & Learning Lab

# Algorithms: automated design

- Automated algorithm design: search space:
  - Automated configuration: parameters of target algorithms
  - Automated selection: target algorithms
  - Automated composition: components of algorithms

- Most current meta-heuristics operate directly on problem solutions
  - Requires extensive experience, time consuming




R. Qu, G. Kendall, N. Pillay. (2019) The General Combinatorial Optimisation Problem: Towards Automated Algorithm Design. accepted by **Computational Intelligence Magazine**, 2019

Computational Optimisation & Learning Lab

# Algorithms: automated configuration

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

- Search space: parameter configurations of target algorithms
- Objective: To automatically configure parameters of target algorithms offline against a given set of training instances
  - Target algorithms: stochastic local search [Pag19], multi-objective evolutionary algorithms [Lop12]
  - Parameters: numerical, categorical
  - COPs: TSP, VRP, flowshop scheduling

- Platforms: automatically search for the configuration of parameters for target algorithms
  - ParamILS[1]: [Hut09]
  - F-Race/I-Race[2]: [Bir10]

1. http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/
2. http://iridia.ulb.ac.be/irace/

Computational Optimisation & Learning Lab

# Algorithms: automated selection

- Search space: a portfolio of algorithms/solvers
- Objective: according to the clustering training instances against certain features, to automatically select from the given target algorithms offline
  - Target algorithms: evolutionary algorithms [Aka17], solvers [Liu19]
  - COPs: TSP, function optimisation

- Platforms
  - Population-based Algorithm Portfolios (PAP): [Tan14]
  - Hydra: [Xu10]

# Algorithms: automated design

- Search space: a set of basic building blocks/components of algorithms
- Objective: to automatically compose new algorithms online by searching for the best composition of components for solving the given problem instances online
  - Algorithms: evolutionary algorithms [Bez14], generic new algorithms, i.e. hyper-heuristics [Bur13,Pil18]
  - COPs: timetabling, NRP, TSP, job shop scheduling, VRP

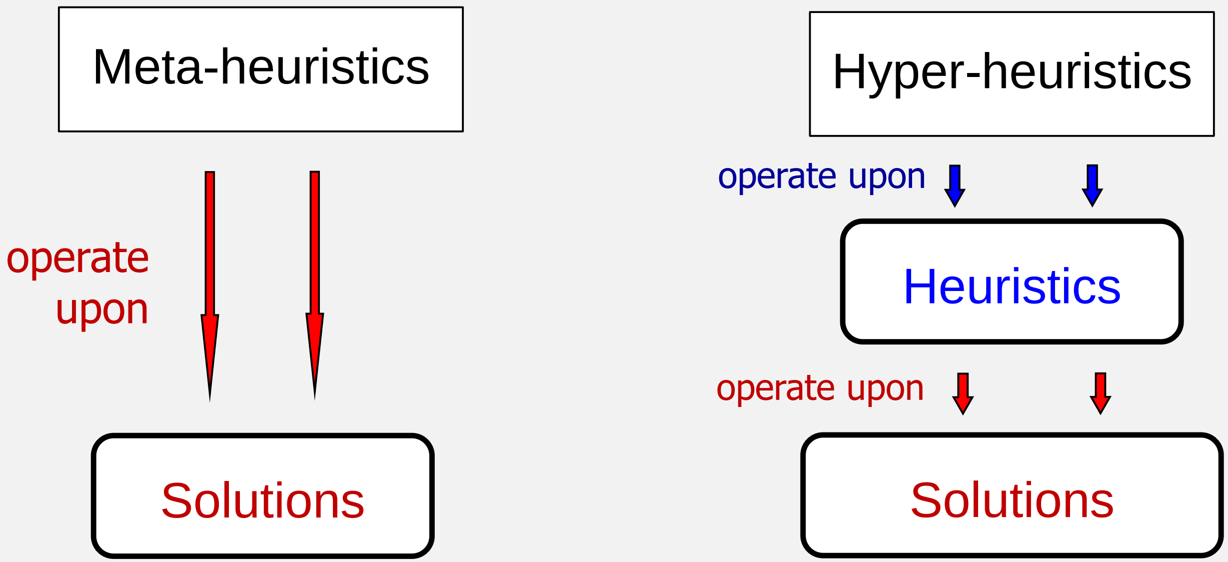
- Platforms:
  - HyFlex[3]: [Bur11]
  - EvoHyp[4]: timetabling, NRP, TSP, job shop scheduling, VRP

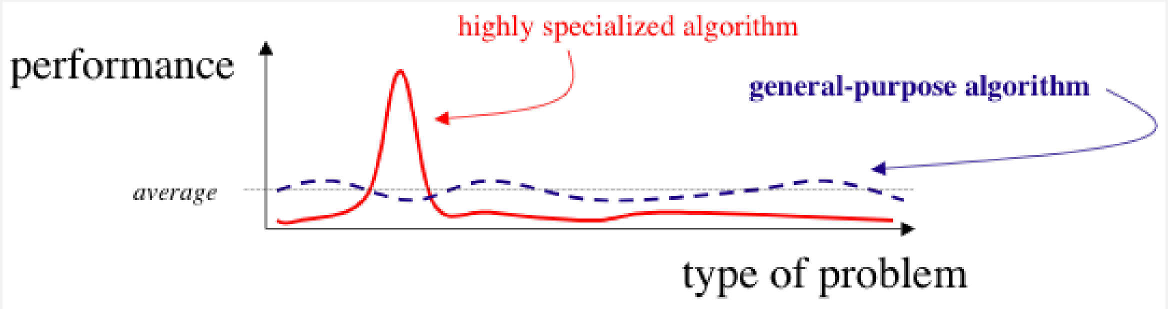
3. http://www.asap.cs.nott.ac.uk/external/chesc2011/
4. http://titancs.ukzn.ac.za/EvoHyp.aspx

# Automated Algorithm Composition
## - A General Graph Based Hyper-heuristic Framework (GHH)

# Hyper-heuristics vs. Meta-heuristics






Wolpert, D.H. and W.G. Macready, No free lunch theorems for search. **IEEE Transactions on Evolutionary Computation** 1: 67, 1997.

# Hyper-heuristics: High Level

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

- Hyper-heuristics operate on a search space of heuristics

- High level heuristics
  - Selection
    - Search methods: evolutionary algorithms, tabu search, simulated annealing, etc.
    - Selection methods: case based reasoning, fuzzy techniques, choice function, etc.
  - Generation
    - Genetic programming, Gene expression programming

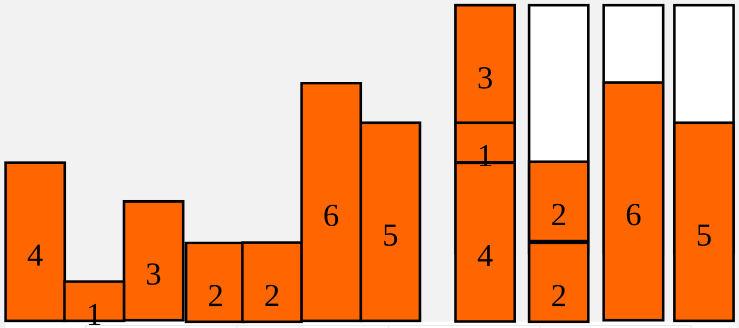
Computational Optimisation & Learning Lab

# Hyper-heuristics: Low Level

- Hyper-heuristics operate on a search space of heuristics

- Low level heuristics
  - Constructive heuristics: construct solutions step by step
    - Graph colouring heuristics, dispatching rules, etc.
  - Improvement heuristics: improve initial solutions by using different strategies iteratively
    - Move operators: swap, insert, destroy and repair

# Hyper-heuristics: Applications



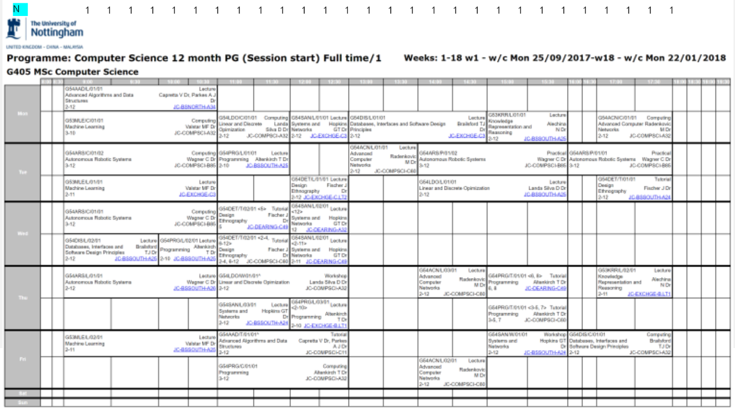

| December | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | | | 2 | | | | | | | 3 | | | | | | | 4 | | | | | | | | |
| | 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 |

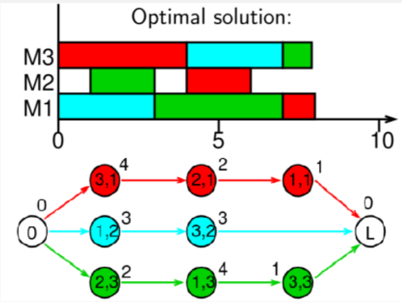
Total Penalty 176

Unassigned Shifts 0

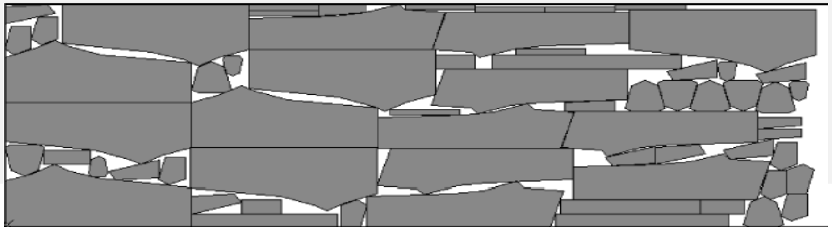
Minimum Cover

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |







# GHH: A General Framework

- Graph-based hyper-heuristic (GHH)
  - High level methods
    - Steepest Descent (hill climbing)
    - Variable Neighbourhood Search
    - Iterated Search
    - Tabu Search

    - Same no. of total iterations
    - “walks” (same quality neighbourhood) allowed
  - Problems: educational timetabling (exam / course); graph colouring

R. Qu and E. K. Burke. Hybridisations within a Graph Based Hyper-heuristic Framework for University Timetabling Problems. **Journal of Operational Research Society**, 60: 1273-1285, 2009

# GHH: A General Framework

- Graph-based hyper-heuristic (GHH)
  - Low level heuristics
    - Five graph colouring heuristics
    - Constructive methods that order events by the difficulties of assigning them (descending order)
    - Random ordering strategy

| Low Level Heuristics | Ordering strategies |
|---|---|
| Largest degree | Number of clashed events |
| Largest weighted degree | LD with number of common students |
| Saturation degree | Number of valid remaining time periods |
| Largest enrolment | Number of students |
| Colour degree | Number of clashed event that are scheduled |
| Random ordering | Randomly |

tional
Optimisation &
Learning Lab

# GHH: Test Problems

- Educational timetabling
  - A number of events $\{e_1, e_2, \ldots, e_e\}$, taken by different students $\{s_1, s_2, \ldots, s_s\}$, need to be scheduled to a limited time period $\{t_1, t_2, \ldots, t_3\}$ and certain rooms $\{r_1, r_2, \ldots, r_3\}$
  - Hard constraints
    - Events with common students can't be assigned to the same time period
    - Room capacity can't be exceeded
  - Soft constraints
    - Spread of exams / no single course on one day
    - Large exams assigned earlier

# GHH: Test Problems

- Educational timetabling
  - Objectives:
    - Minimise timeslots / colours used
    - Minimise violations of soft constraints, i.e. weighted sum
  - Benchmark problems
    - Widely tested in the last decade
    - http://www.cs.nott.ac.uk/~rxq/data.htm
    - 1st International Timetabling Competition - http://www.idsia.ch/Files/ttcomp2002/

Computational Optimisation & Learning Lab

# GHH: Test Problems

- Exam timetabling
  - Hard constraints: conflicts between exams
  - Soft constraints: spread out exams over slots

- Course timetabling
  - Hard constraints: room capacity; conflicts
  - Soft constraints
    - Courses scheduled consecutively (students)
    - More than two classes or no class a day
    - Courses can't be combined into one room
    - Last timeslot of the day
    - Preferred time periods, day-off (lecturers)
  - Courses associated with by lecturers, time periods usually on weekly basis, and rooms with certain features

- Graph-based hyper-heuristic (GHH)

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 | ... |

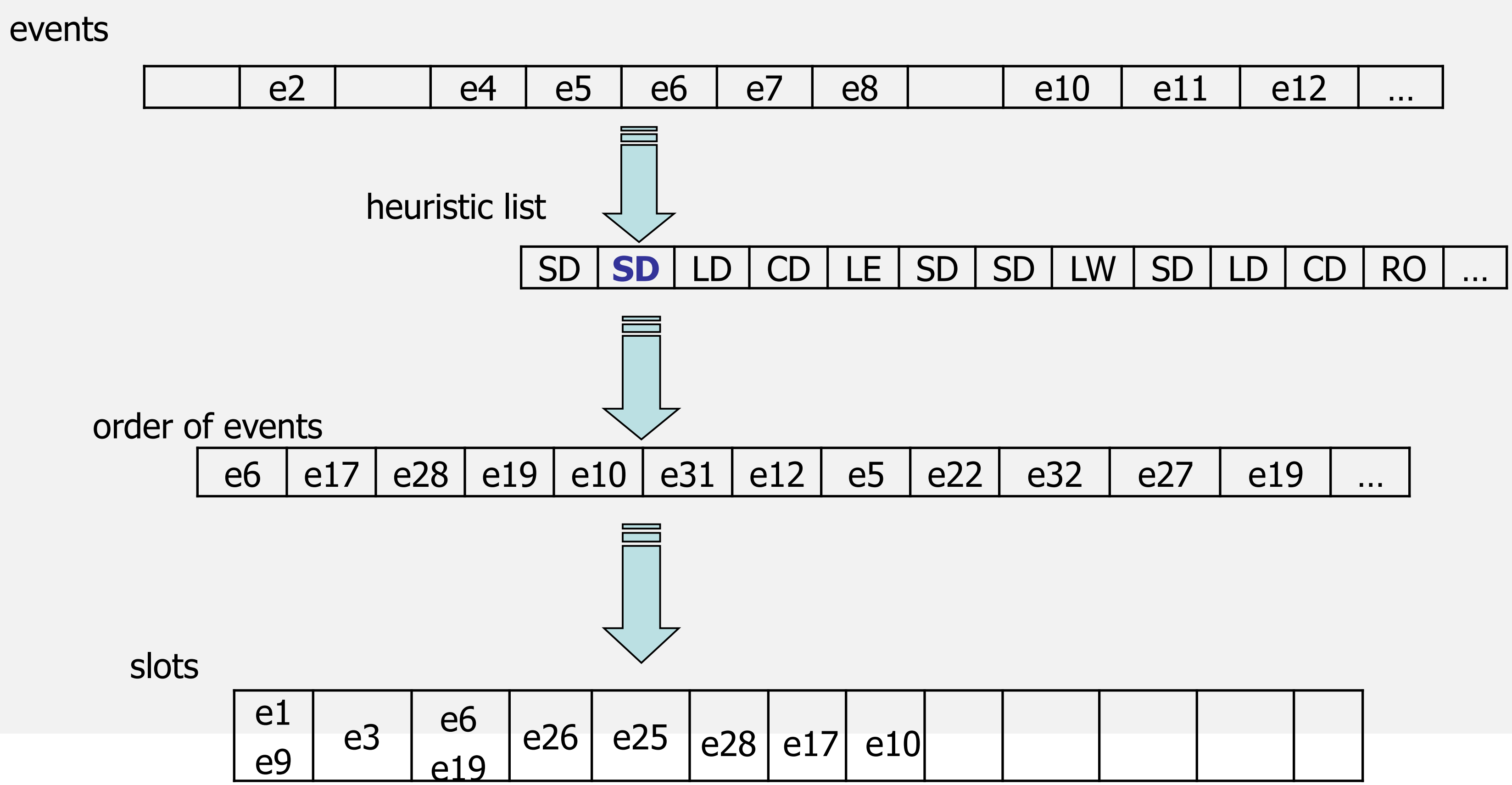
slots

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| e1<br>e9 | e3 | | e26 | e25 | | | | | | | | |

# GHH: A General Framework



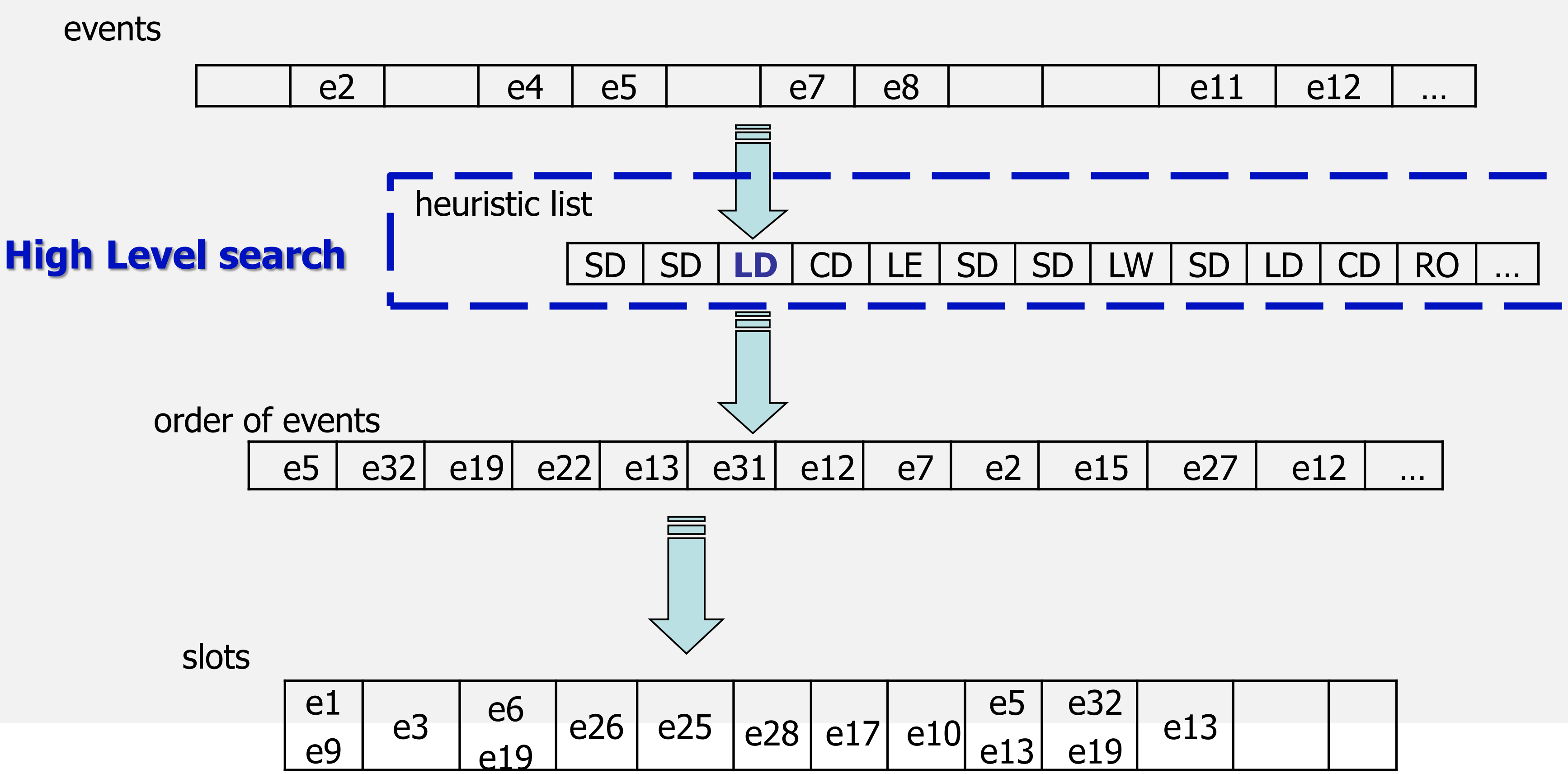
- Graph-based hyper-heuristic (GHH)



# GHH: A General Framework



- Graph-based hyper-heuristic (GHH)



# GHH: Results

- Graph-based hyper-heuristic (GHH)
  - VNS performs better than iterated search, both better than hill climbing and tabu search
  - Iterated search performs generally well

  - Search space of GHH for some instances: large areas of plateau
  - Same quality heuristic lists don't always produce same solutions
  - Greedy local search (almost always) further improve the solutions generated by heuristic lists; however, time increases significantly
  - High level search cannot reach local optima in the solution space

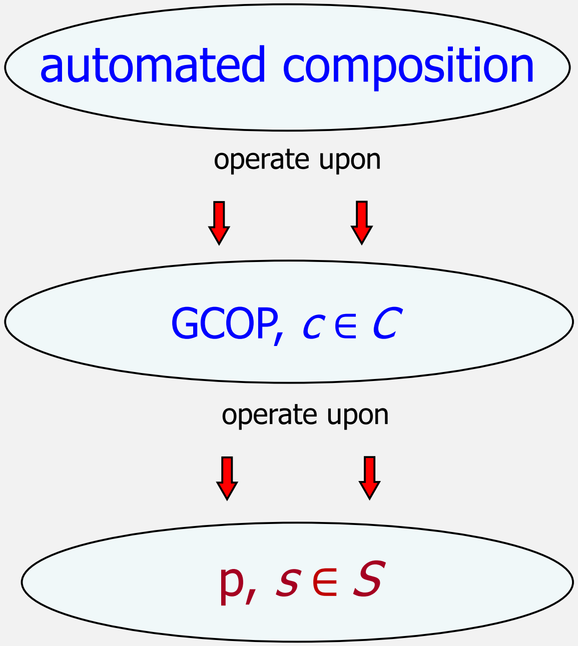
# Automated Algorithm Composition
## - Understanding the Search Algorithms

# GCOP: A New Standard

- A General Combinatorial Optimisation Problem ***GCOP***, whose decision variables are algorithmic components rather than direct solution variables in the optimisation problem ***p***

- To solve ***p***, composition methods explores at a algorithm compositions ***c*** in a component space *C*, which map direct solutions ***s*** in the solution space *S* for ***p***

- **GCOP:** automated algorithm composition





R. Qu, G. Kendall, N. Pillay. (2019) The General Combinatorial Optimisation Problem: Towards Automated Algorithm Design. accepted by **Computational Intelligence Magazine**, 2019

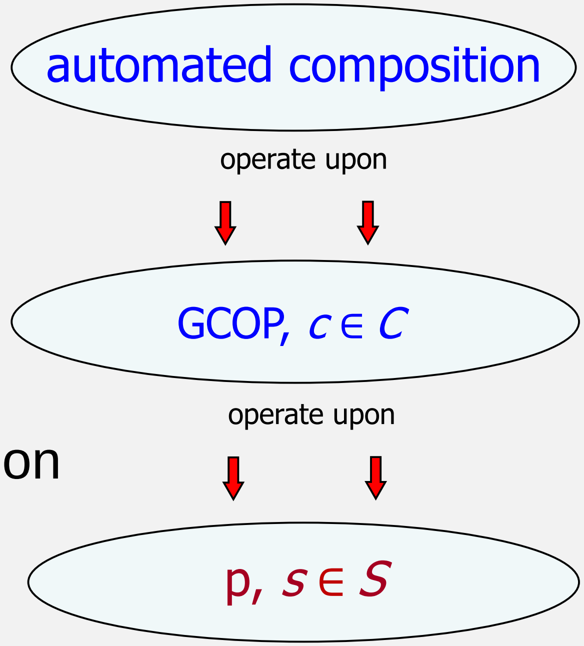
# GCOP: A New Standard

- The algorithm compositions ***c*** ∈ ***C*** are measured by objective function *F(c)* → *R*
  The direct solutions ***s*** ∈ ***S*** are measured by objective function *f(s)* → *R*

- *s* are obtained using *c, i.e. c* → *s*
  Let mapping function ***M***: *f(s)* → *F(c)*

- The objective of GCOP is to find optimal *c**

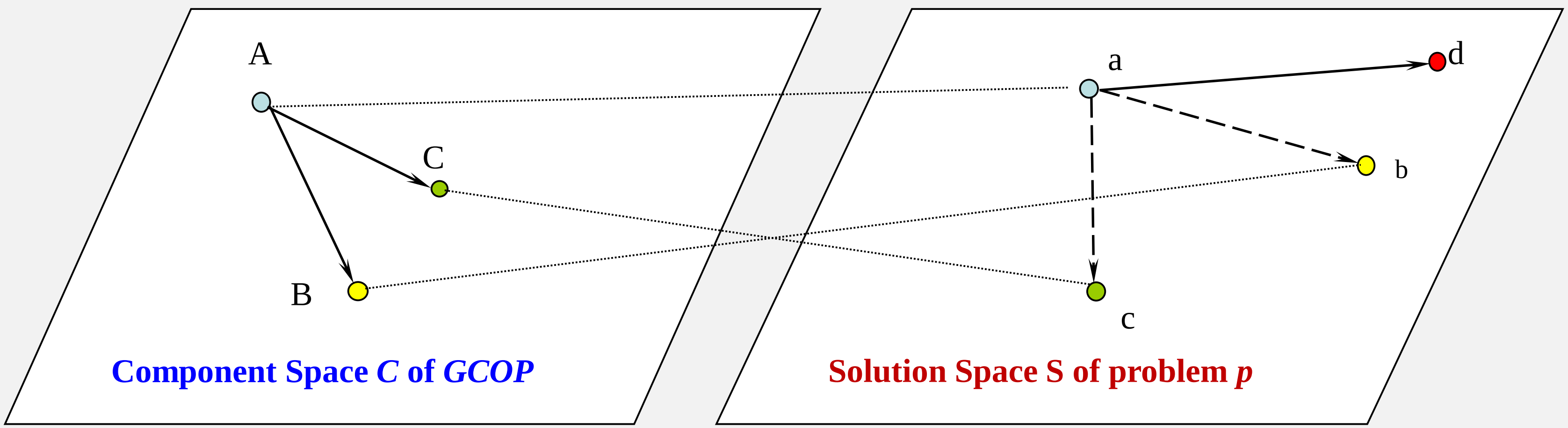
- Therefore, two spaces are under consideration
  - an algorithmic component space ***C*** and
  - a solution space ***S***




Computational Optimisation & Learning Lab

# GCOP: Two Search Spaces

- Graph-based hyper-heuristic (GHH)
  - Two search spaces
    - high level: sequences of low level heuristics (components)
    - low level: actual problem solutions




**Search is upon heuristics (components), not direct solutions: are all the solutions in solution space reachable?**

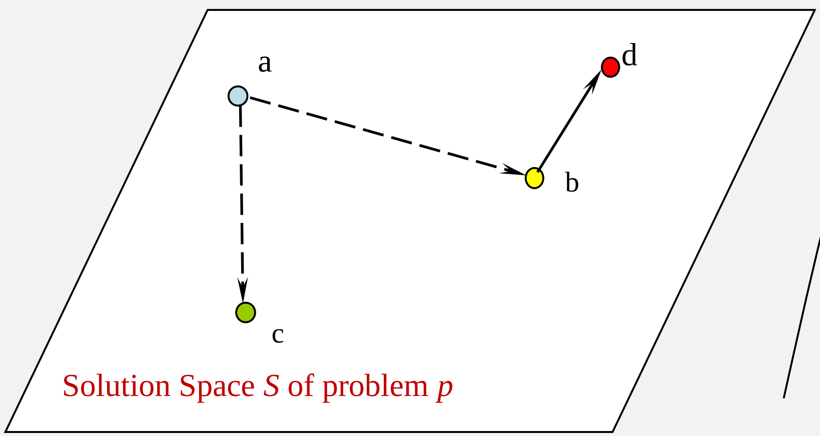
E.K. Burke, B. McCollum, A. Meisels, S. Petrovic, R. Qu*, A Graph-Based Hyper-Heuristic for Educational Timetabling Problems, **European Journal of Operational Research**, 176: 177-192, 2007. ISI Top 0.1% cited; Five Year Top Cited Article EJOR 2007-2011 Award.

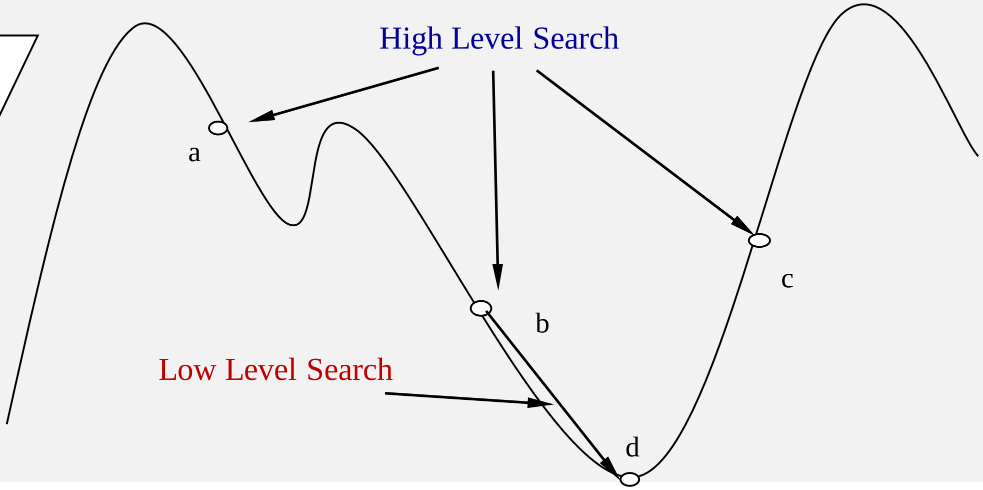
# GCOP: Two Search Spaces

| | Heuristic (component) space *C* | Solution space *S* |
|---|---|---|
| **Representation** | | |
| **Size (Upper Bound)** | | |
| **Neighborhood Operator** | | |
| **Objective Function** | | |

# GCOP: Two Search Spaces

- Search within the two search spaces *C* and *S*
  - Hypothesis: search upon heuristics (components) not solutions, not all solutions are reachable
  - Component space *C*: matched solutions a, b, c, …
  - Solution space *S*: b → d, …
    - Coverage of all solutions in *S*

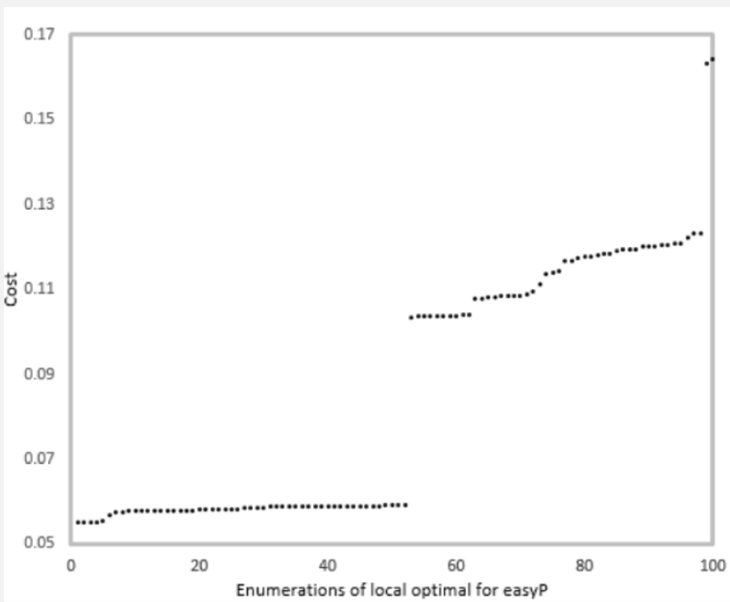


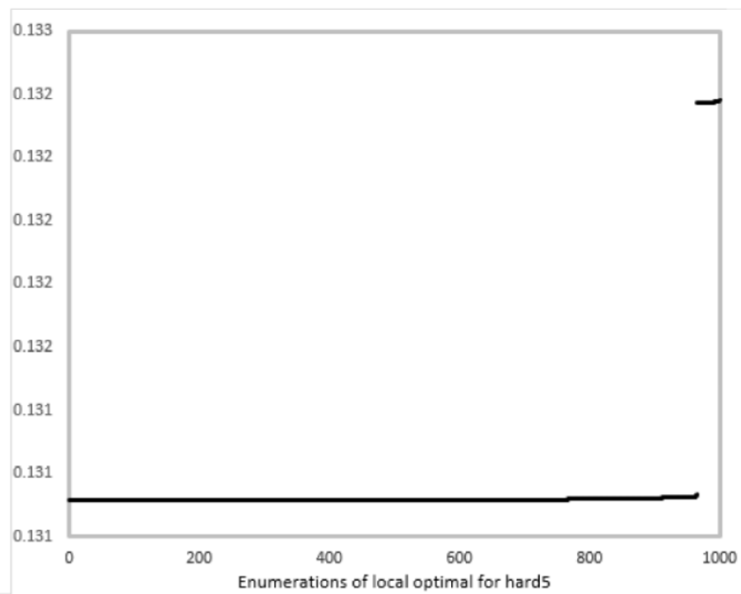



# GCOP: Landscape Analysis



- Landscape Analysis on high level heuristic compositions *c*
  - Distribution of costs for local optimal *c*

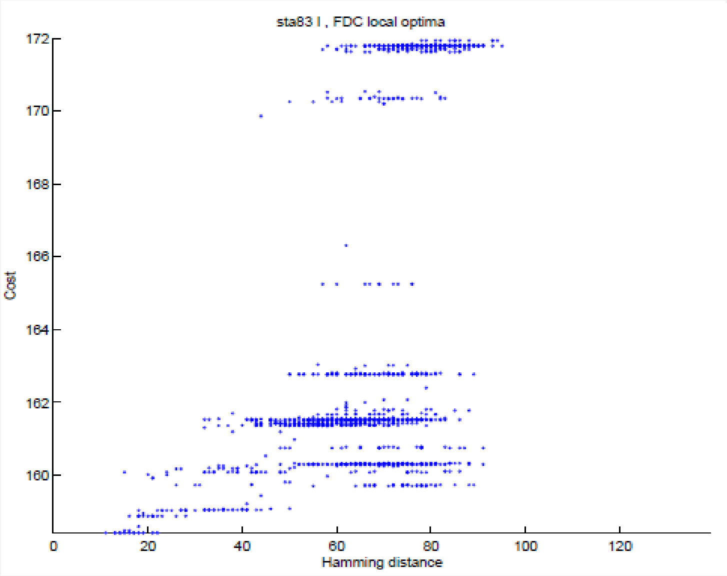


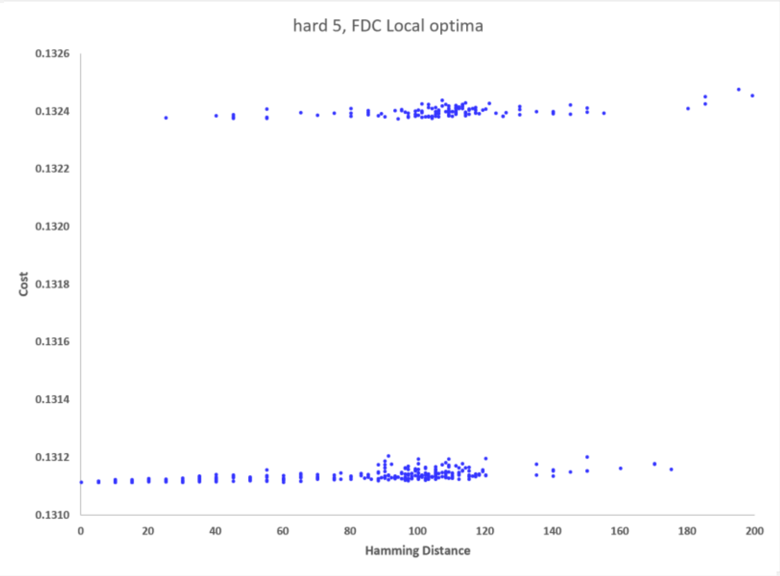




G. Ochoa, R. Qu and E.K. Burke, Analyzing the Landscape of a Graph Based Hyper-heuristic for Timetabling Problems
**The Genetic and Evolutionary Computation Conference (GECCO'09)**, pp. 341-348, 8-12 July, Montreal, Canada

# GCOP: Landscape Analysis

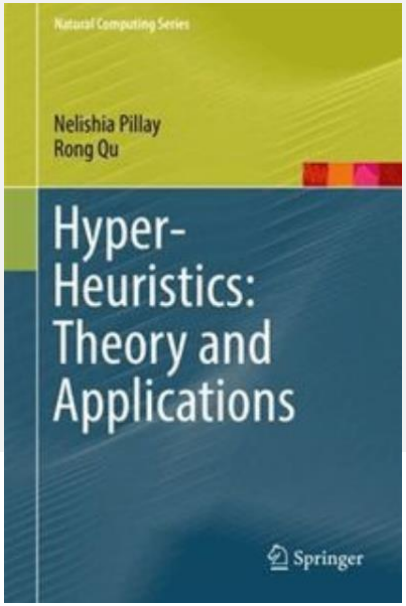
- Landscape Analysis on high level heuristic compositions *c*
  - Fitness distance correlation (*fdc*) of local to global optimum

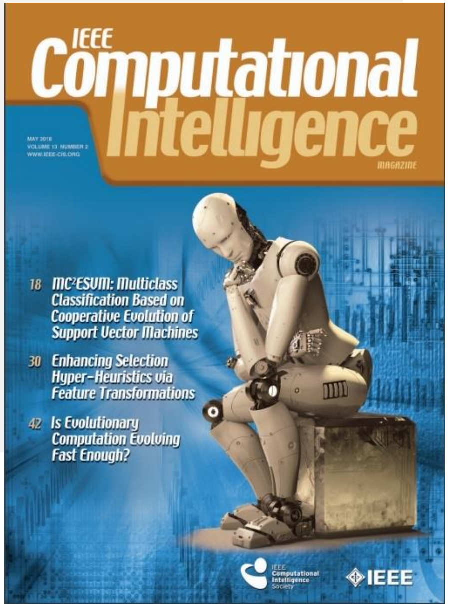


# Related Activities

- N. Pillay, R. Qu, « Hyper-heuristics: Theory and Applications», Springer, ISBN 978-3-319-96514-7, December, 2018
- N. Pillay, R. Qu (ed.), Automated Design of Machine Learning and Search Algorithms, Special Issue at Computational Intelligence Magazine, 13(2), June 2018.
- IEEE Computational Intelligence Society Task Force on Hyper-heuristics
- IEEE Computational Intelligence Society Task Committee on Evolutionary Computation



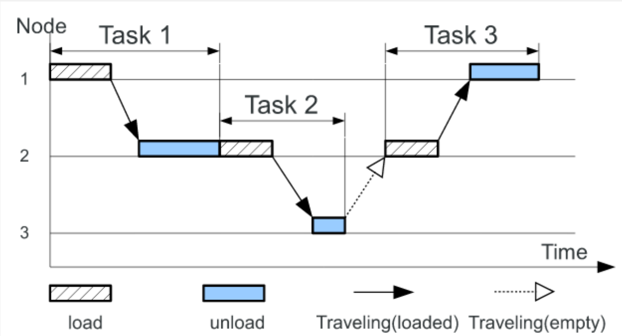



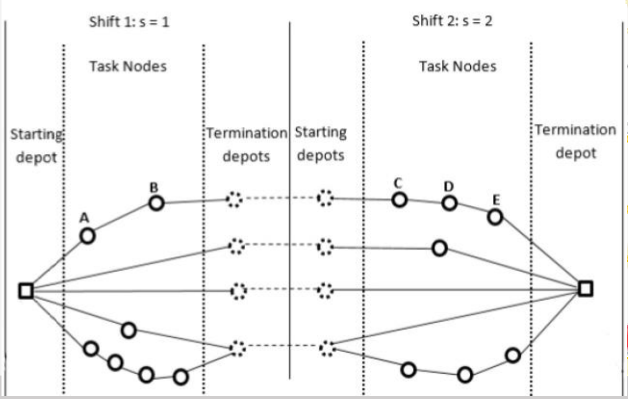
# Modelling and Optimisation of other COPs

Extended new problem model with inland dry ports

Mixed shift types with non-linear driver costs

Bi-objective optimization by hybrid hyper-heuristics





ge international port.


R Qu and R Bai

# Logistic Transport Scheduling @ Ningbo

B. Chen, R. Qu, R. Bai, W. Laesanklang, "A hyper-heuristic with two guidance indicators for bi-objective mixed-shift vehicle routing problem with time windows", in press, *Applied Intelligence*, 2018.

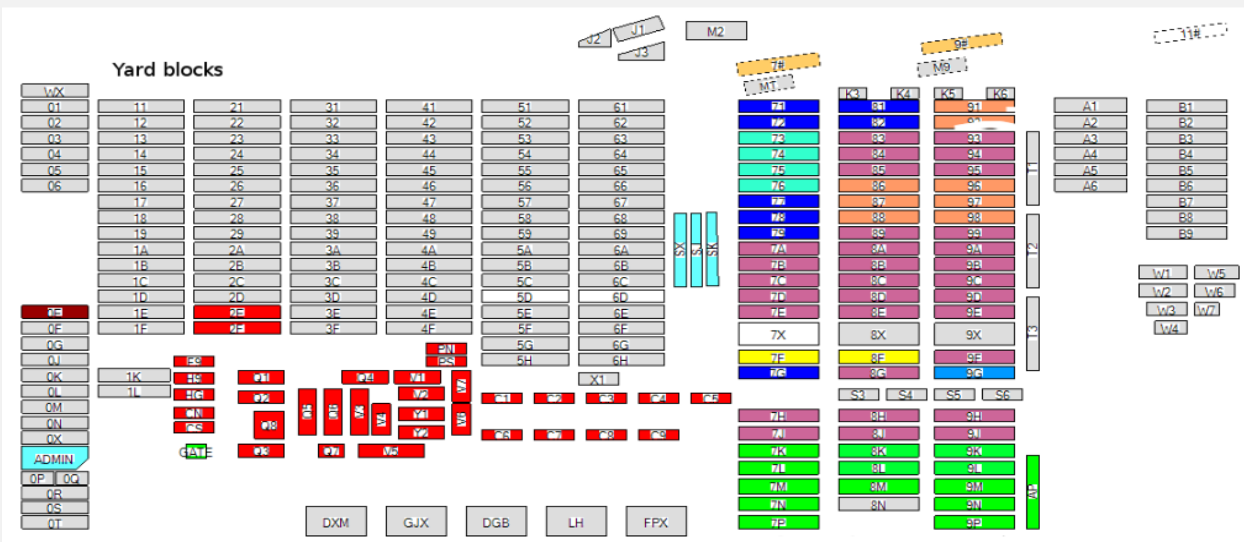
Real time crane scheduling at container terminals

Complex dynamic truck dispatching for containers’ transfer

- Robust hyper-heuristics hybridising three dynamic heuristics


ge international port.

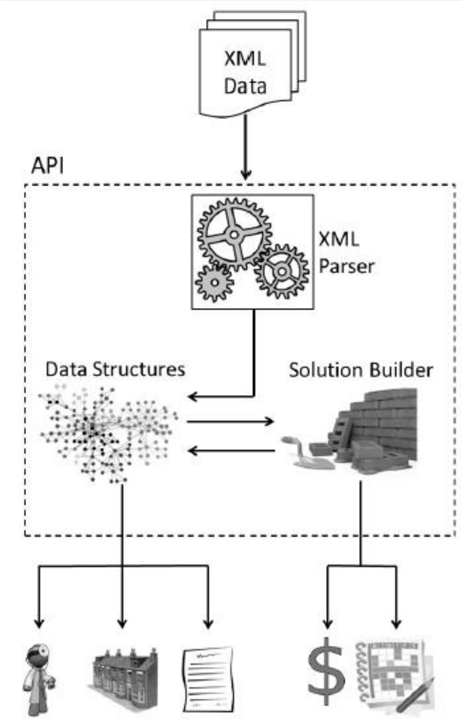
R Qu and R Bai




# Logistic Transport Scheduling @ Ningbo

J Chen, R Bai, H Dong, R Qu and G Kendall, "A Dynamic Truck Dispatching Problem in Marine Container Terminal"
The 2016 Symposium on Computational Intelligence in Scheduling and Network Design (IEEE CISND'16), Dec 6-9, 2016

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA




Innovate UK

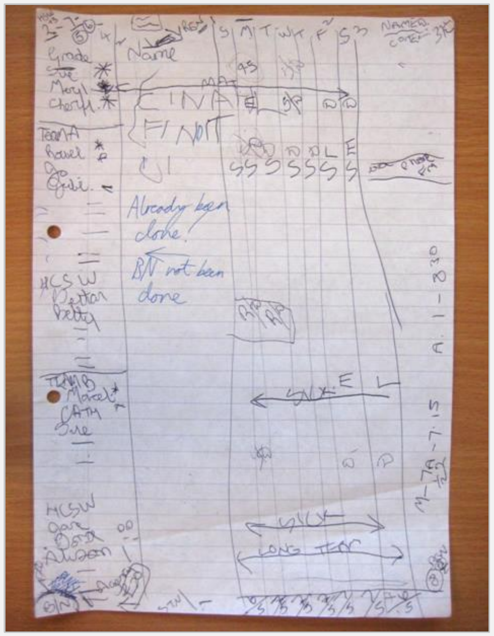
webroster.net

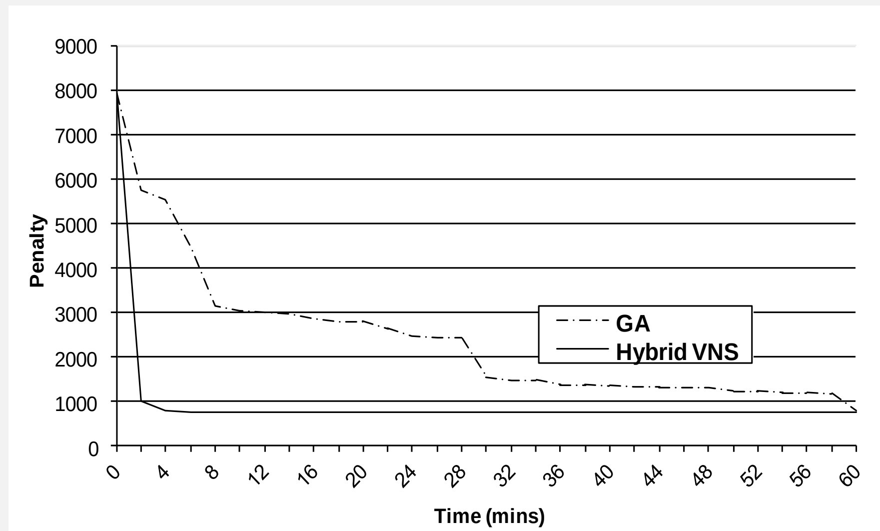
- Routing + Rostering in healthcare
- Optimisation engine demonstrated at roadshows
- “Outstanding” award at KTP, Innovate UK
- Automated heuristic algorithms
- UK SME: Webroster®

# Healthcare Rostering & Routing


J. Arturo Castillo-Salazar, Dario Landa-Silva, Rong Qu, "Computational Study for Workforce Scheduling and Routing Problems"
The 3rd International Conference on Operations Research and Enterprise Systems (ICORES 2014), pp. 434-444, Angers France, March 2014

- Different grades / skill, shift types, coverage
- Constraints
  - Hard: Coverage
  - Soft: work/rest hours, complete weekends, preferences, etc.
- Evolutionary algorithms, hybrid VNS




EPSRC ORTEC

# Nurse Rostering Problems

E.K. Burke, T. Curtois, R. Qu and G. Vanden Berghe. "A Time Predefined Variable Depth Search for Nurse Rostering". INFORMS Journal on Computing, 25: 411-419, 2013

| | 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 |
| A | DH | DH | DH | DH | DH | | | DH | DH | DH | | DH | DH | DH | DH | | DH | DH | | | | DH | DH | DH | DH | DH | | | 20 |
| B | N | N | N | N | | | | D | D | L | L | L | | | | L | L | L | | | | E | E | E | D | D | | | 0 |
| C | D | D | D | D | D | | | | N | N | N | | L | L | L | | | | L | L | L | | E | E | E | L | | | 25 |
| D | | | | L | N | N | N | N | | | DH | D | | | | E | E | E | DH | E | E | | N | N | | | E | E | 13 |
| E | | | | | D | DH | DH | D | | | | | E | E | | DH | E | E | E | DH | DH | | D | D | E | E | DH | DH | 21 |
| F | L | L | L | | | L | L | L | L | | | N | N | N | N | | | D | D | | | D | | | | D | D | D | 10 |
| G | | | | E | E | E | E | | | D | D | D | | | E | E | | | D | D | D | D | | | N | N | N | N | 10 |
| H | E | E | E | | | D | D | | E | E | E | E | | | D | D | D | | N | N | N | N | | | | | L | L | 26 |

Too few resting time (10)

Too few consecutive late shifts (5)

Too few consecutive night shifts (5)

Total Penalty 176

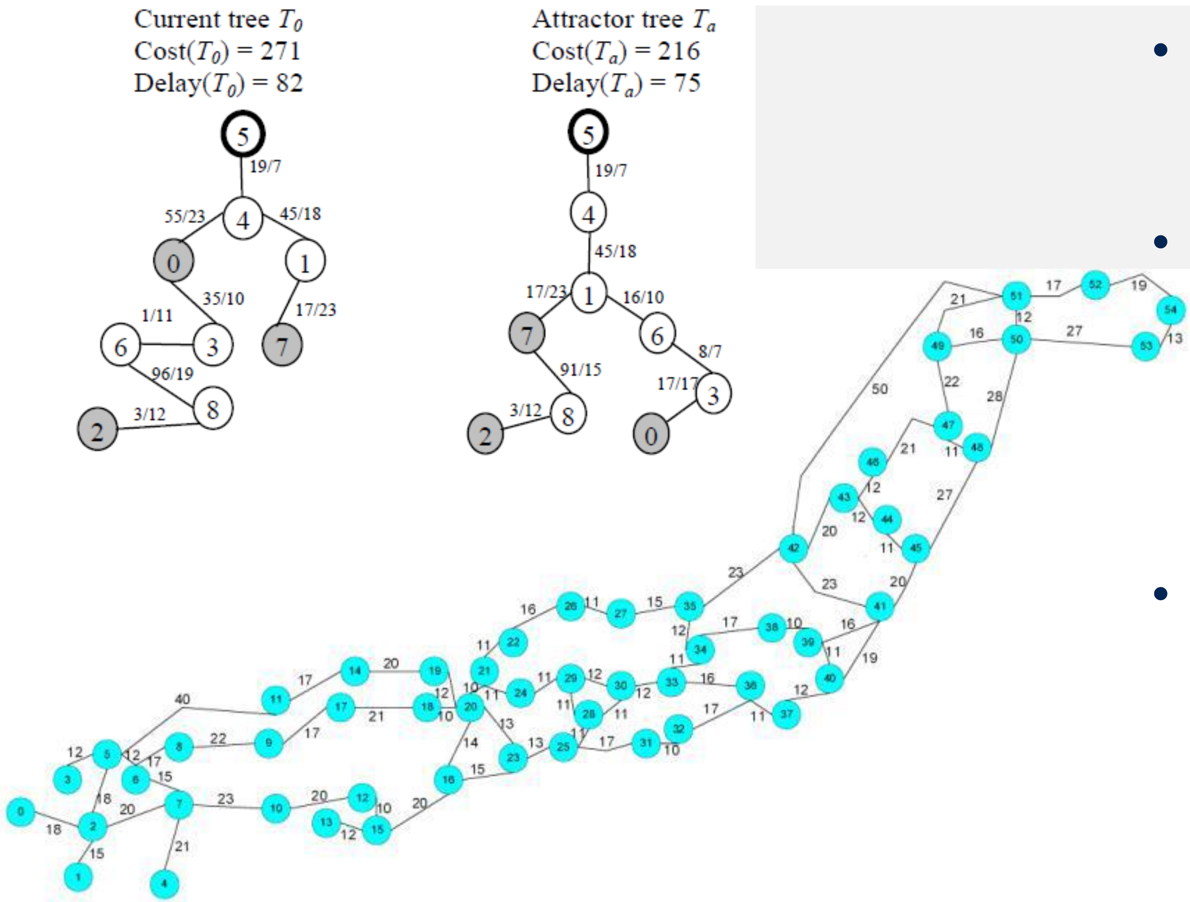
Unassigned Shifts 0

Minimum Cover

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

# Nurse Rostering Problems

NRP benchmark datasets:
http://www.schedulingbenchmarks.org/




- Find the multicast tree serving all terminals, satisfying delay bound
- Multiple objectives
  - Minimal costs
  - Maximal end-to-end delay
  - Maximal link utilisation
  - Average delay
- Applications
  - Video conferencing
  - e-learning, etc.

# Multicast Routing

R. Qu, Y. Xu, J. Castro, D. Landa-Silva. "Particle Swarm Optimization for the Steiner Tree in Graph and Delay-Constrained Multicast Routing Problems.“, Journal of Heuristics, 19(2): 317-342, 2013

The cost of the multicast tree:

$$C(T) = \phi \cdot \sum_{(i,j)\in T} c_{ij}$$

The maximal end-to-end delay of the multicast tree:

$$DM(T) = Max\{d(p_T(s, r_d))\}, r_d \in R$$

The maximal link utilisation:

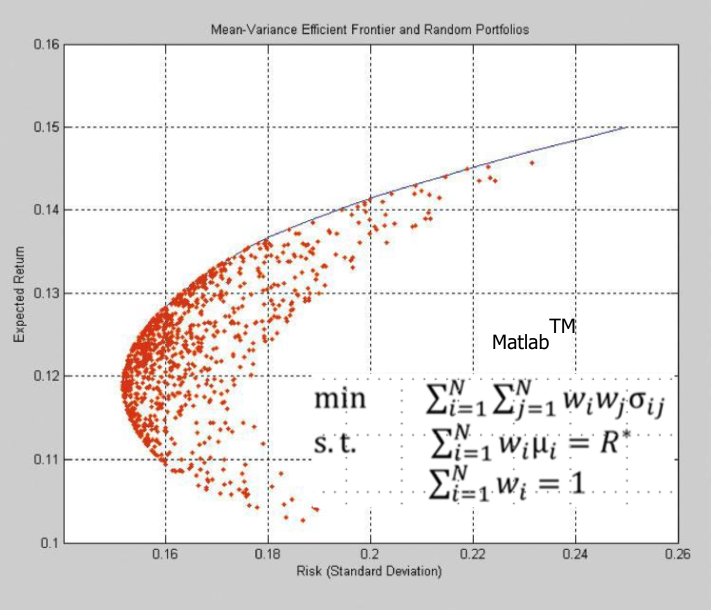
$$\alpha(T) = Max\left\{\frac{\phi + t_{ij}}{z_{ij}}\right\}, (i, j) \in T$$

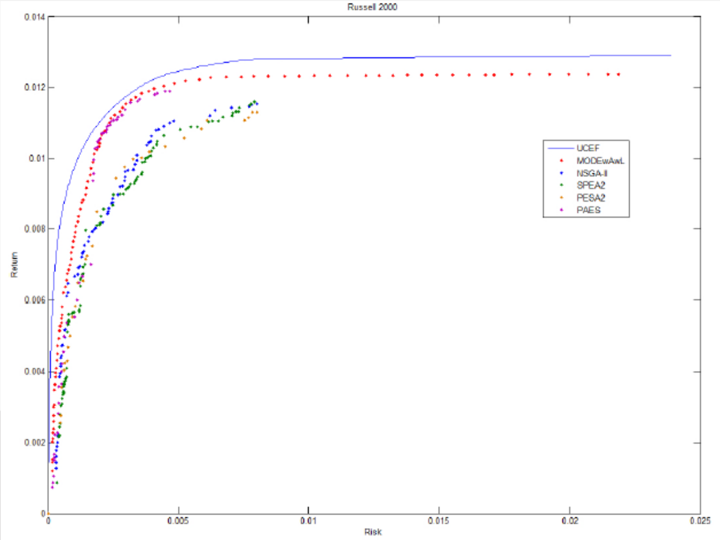
The average delay of the multicast tree:

$$DA(T) = \frac{1}{|R|}\sum_{r_d\in R} d(p_T(s, r_d))$$

- Quality of Service (QoS) constraints
  - Bandwidth, Delay, Cost, Delay variations
- Various multi-objective evolutionary algorithms
  - Scatter search, PSO, iterative local search
- Future extension: problems with uncertainties

# Multicast Routing



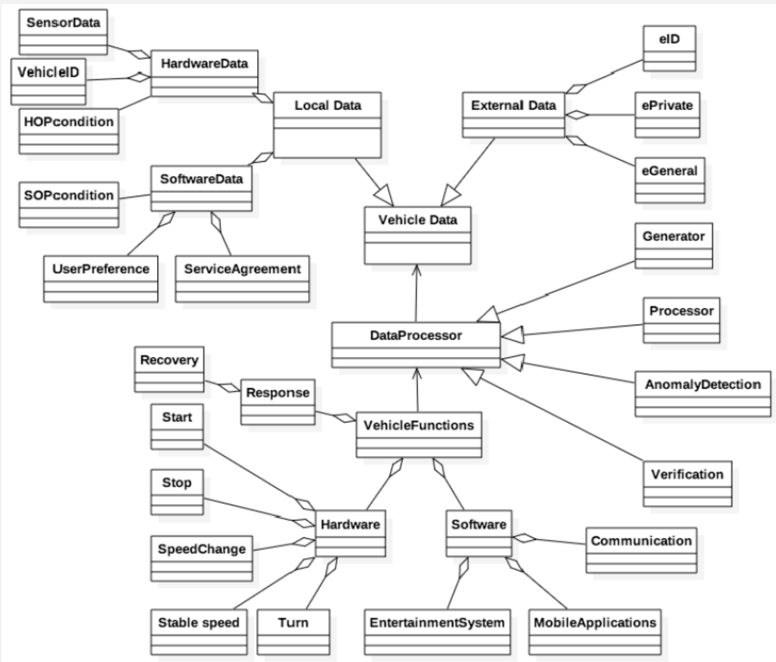



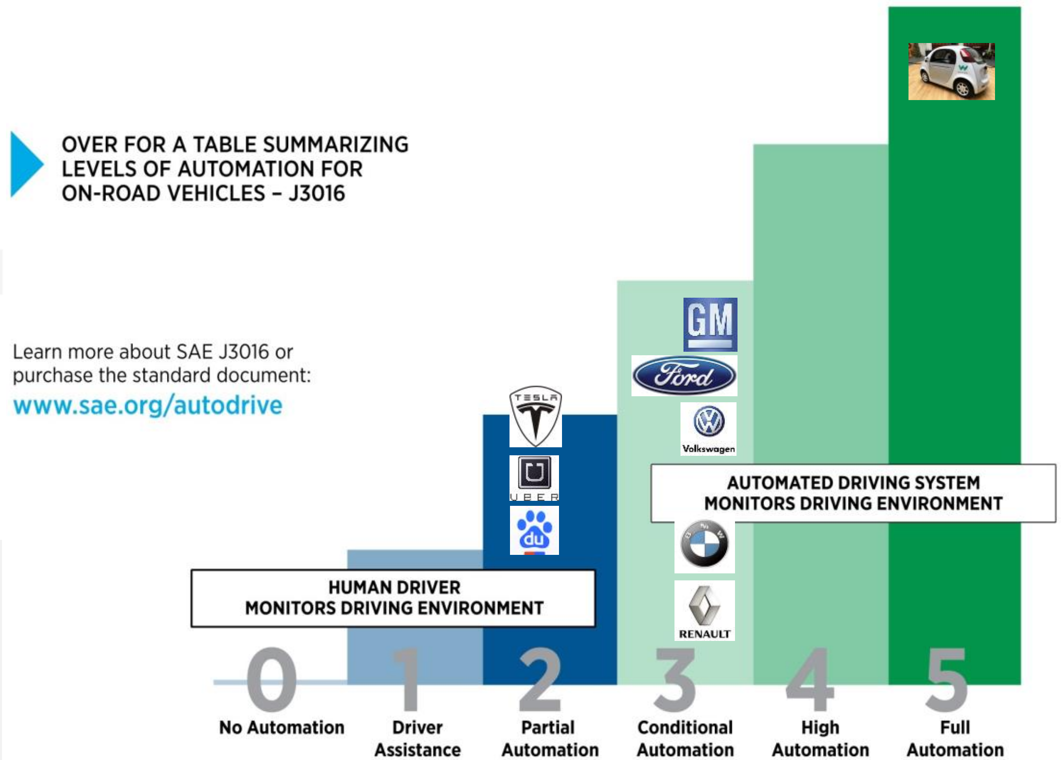

- Allocate capital to selected assets
- Objective: to find the portfolio with
  - the highest expected return or
  - the lowest risk for the expected return

- Markowitz's portfolio theory
  - Mean-Variance model
  - Efficient frontier: risk vs. return
- Real world constraints
  - Cardinality $\sum_{i=1}^{N} s_i = K$
  - Boundary $\epsilon_i s_i \leq w_i \leq \delta_i s_i, i = 1, \ldots, N$
  - Round lot $w_i = y_i . v_i, \quad i = 1, \ldots, N, \quad y_i \in \mathbb{Z}_+$
  - Pre-assignment $s_i \geq z_i, \quad i = 1, \ldots, N$

# Portfolio Optimisation

Khin Lwin, Rong Qu, Bart MacCarthy, "Mean-VaR Portfolio Optimization: A Nonparametric Approach", European Journal of Operational Research, 260(2): 751-766, 2017




# CAV Cyber Security

Q. He, X. Meng, R. Qu, A New Framework for Assessing CAV Cyber Security, under review, 2019.

- KDD99
  - A benchmark dataset: online intrusion / attack detection
  - 5m data records, 42 attributes, labelled with four major types of 39 sub-attacks (only 22 sub-attacks in the training dataset)
- CAV-KDD: new dataset for CAVs cyber security
  - Three types attack points: hardware, software, data
  - KDD99 training and testing datasets processed against characteristics of CAVs, compatible to the new framework
  - Four major types of 14 sub-attacks

Amount of Normal and Attack Data in the Training Datasets

| | 10% KDD99 Data | CAV-KDD Data |
|---|---|---|
| Attacks | 396743 | 13274 |
| Normal | 97278 | 58716 |
| Total | 494021 | 71990 |

Accuracy and Runtime of J48 and Naive Bayes

| | Accuracy on the Testing Dataset | Time to Build Model (s) | Time on the testing Dataset (s) |
|---|---|---|---|
| Naive Bayes | 76.2% | 0.04 | 3.95 |
| J48 | 76.3% | 1.4 | 0.94 |

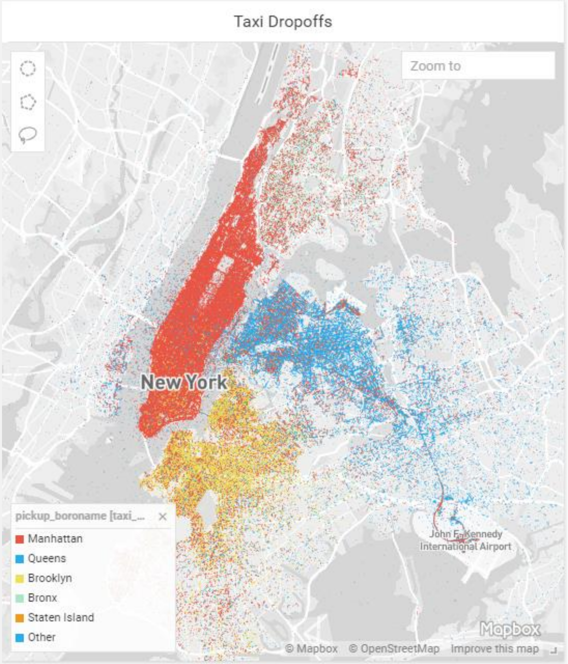
# CAV Cyber Security

Q. He, X. Meng, R. Qu, A New Framework for Assessing CAV Cyber Security, under review, 2019.

# Future Research Collaborations

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

- ITS / Traffic Network Simulation & Optimisation
  - Machine Learning
    - CAV cyber security
    - Image recognition
    - Traffic data analysis
  - Optimisation in Intelligent Transport Systems
  - CAV real time testing with Engineering
- Automated Algorithm Design
  - General search algorithm framework
    - Easy to implement and use
    - Data collection on algorithm compositions
  - Theory: landscape analysis
  - Machine learning: patterns of algorithm design / compositions




Computational Optimisation & Learning Lab