Selection Hyper-Heuristics for Automated Design, Configuration and Selection

Dr. Rong Qu
School of Computer Science, University of Nottingham, UK

This Webinar is provided to you by
IEEE Computational Intelligence Society
https://cis.ieee.org

IEEE Computational Intelligence Society

IEEE Advancing Technology for Humanity

Generation Hyper-Heuristics for Automated Design, Configuration and Selection

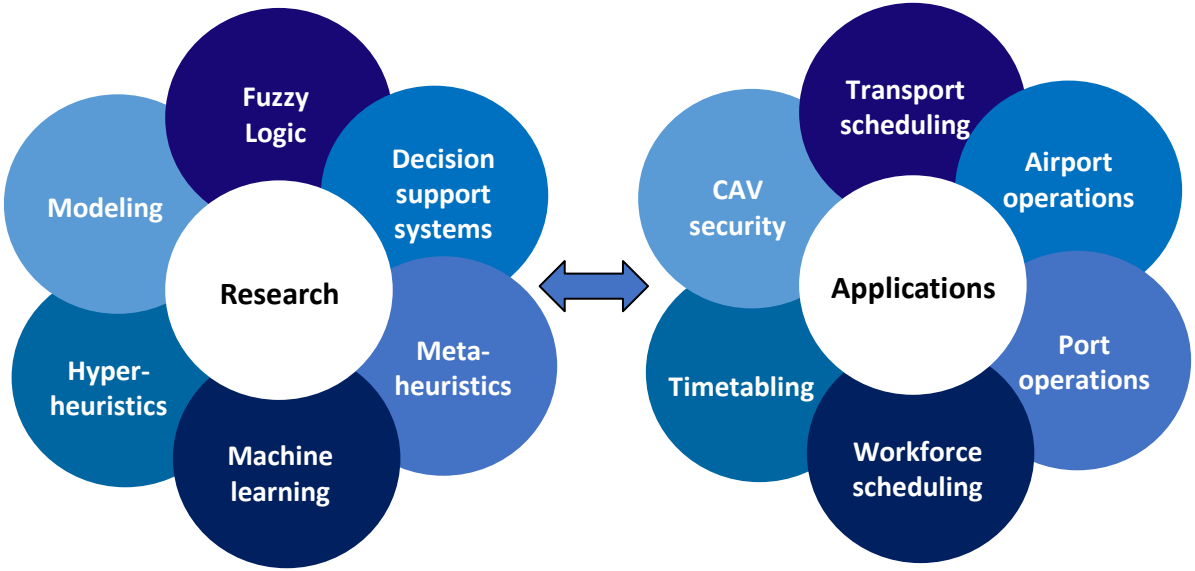
Prof. Nelishia Pillay
Department of Computer Science, University of Pretoria, South Africa

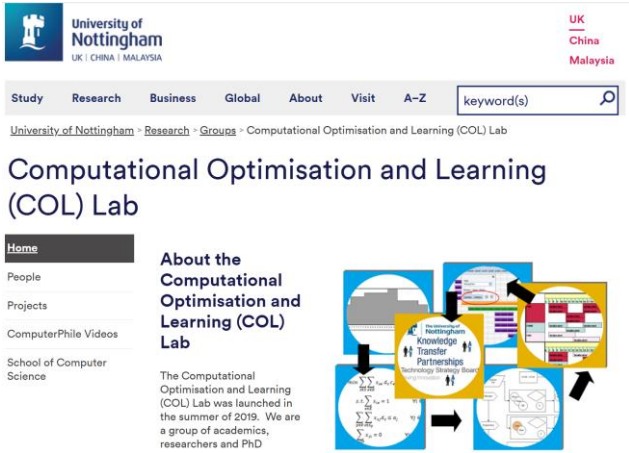
This Webinar is provided to you by
IEEE Computational Intelligence Society
https://cis.ieee.org

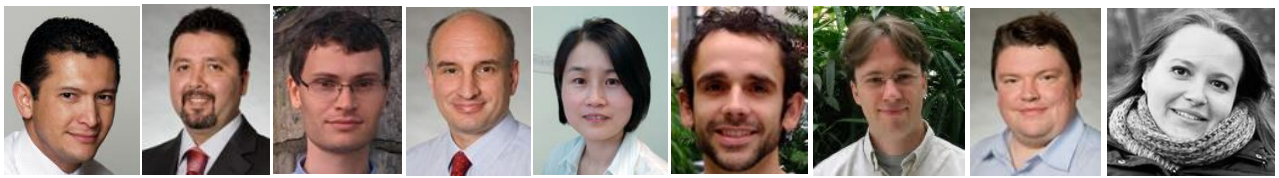
IEEE Computational Intelligence Society

IEEE Advancing Technology for Humanity

COL Lab, University of Nottingham

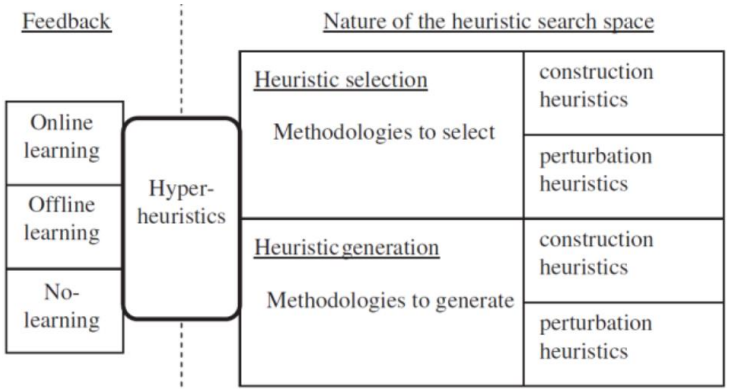






IEEE

Automated Algorithm Design (AutoDes) with Hyper-heuristics



Automated Algorithm Design (AutoDes)

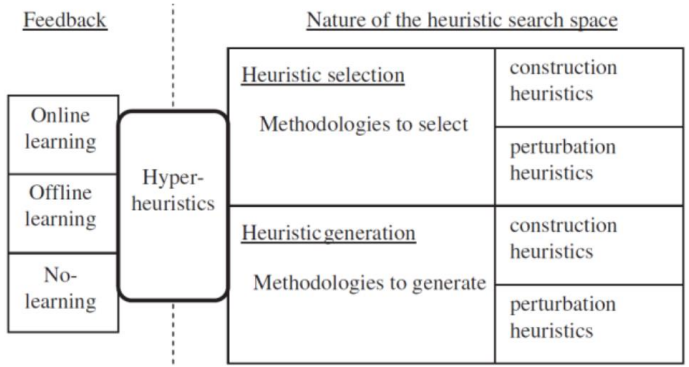
- Decisions to make when designing algorithms
  - Algorithm specific decisions
    - Simulated annealing; Tabu search; Variable neighbourhood search
    - Genetic algorithms; Estimation of distribution algorithm
    - Swarm Intelligence
    - Heuristics / operators
    - And some more …

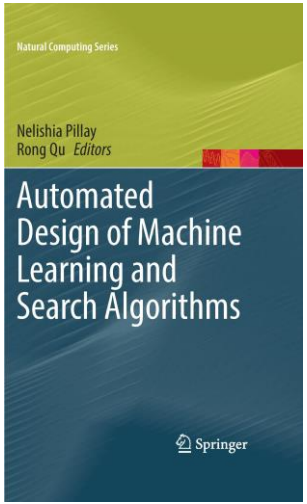
Automated Algorithm Design (AutoDes)

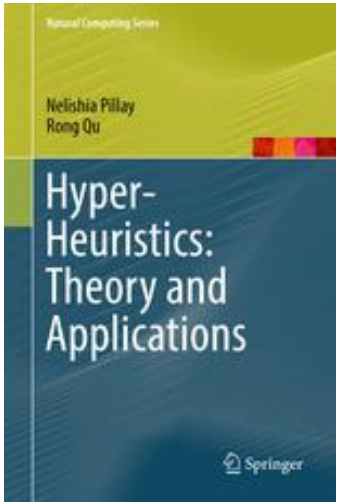
- Decisions to make when designing algorithms
  - Problem specific decisions
    - Operators
    - Solution representation
    - Evaluation function
  - General decisions
    - Initialisation
    - Stopping condition
    - Acceptance criteria

Automated Algorithm Design (AutoDes)

- Recent / advanced research developments
  - Integration of other computational intelligence techniques
    - Hybridisation of evolutionary and local search algorithms
    - Machine learning and optimisation
    - Data-driven optimisation
    - Hyper-heuristics
    - And many more …

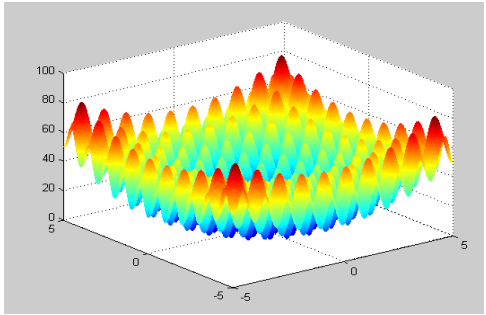


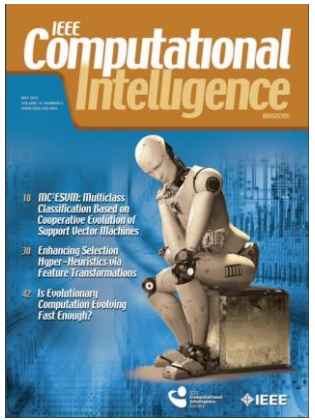




Automated Algorithm Design (AutoDes)

- Recent / advanced research developments
  - Automated algorithm design, w.r.t. decision space (of algorithm design) [Qu20]
    - Automated composition: components of algorithms
    - Automated configuration: parameter selection/setting
    - Automated selection: given algorithms





IEEE Computational Intelligence Society

Automated Algorithm Design (AutoDes)

- Search space: parameter configurations of target algorithms
- Objective: To automatically configure parameters of pre-defined 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 problems
- Platforms: automatically search for the configuration of parameter space 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/

Automated Algorithm Design (AutoDes)

- Search space: a family/portfolio of algorithms/solvers
- Objective: according to the grouping/clustering of a set of 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]

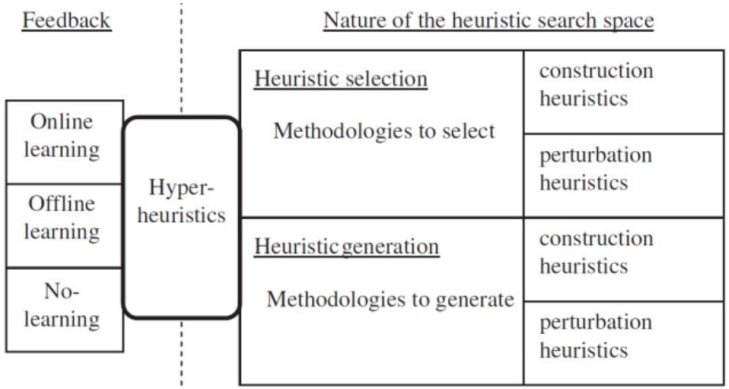
Automated Algorithm Design (AutoDes)

- 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
  - Target algorithms: evolutionary algorithms [Bez14], general new algorithms, i.e. hyper-heuristics [Bur13,Pil18]
  - COPs: timetabling, NRP, TSP, job shop scheduling, VRP

- Platforms:
  - HyFlex: [Bur11]
  - EvoHyp: timetabling, NRP, TSP, VRP, etc. [Pil17]

AutoDes – Hyper-heuristics

- "A high-level approach that, given a particular problem instance and a number of low-level heuristics, can select and apply an appropriate low-level heuristic at each decision point" [Bur13]
- Objective: to find
  - the right high-level method or sequence of easy-to-implement low-level heuristics in a given situation, rather than trying to solve the problem directly
  - an adequate combination of the provided components to effectively solve the given problem(s)
- Platforms:
  - HyFlex: [Bur11]
  - EvoHyp: timetabling, NRP, TSP, VRP, etc. [Pil17]

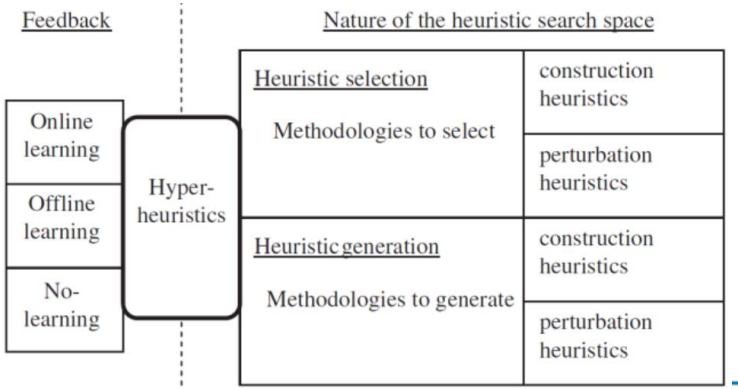


IEEE Computational Intelligence Society

AutoDes – Selection Hyper-heuristics

- Low level heuristics: Constructive
  - Build solutions incrementally
  - Education timetabling (graph coloring), production scheduling (dispatching rules)
  - Bin packing (heuristic rules), workforce scheduling (resource selection)
  - Constraint satisfaction (variable ordering), VRP (both constructive and perturbative)

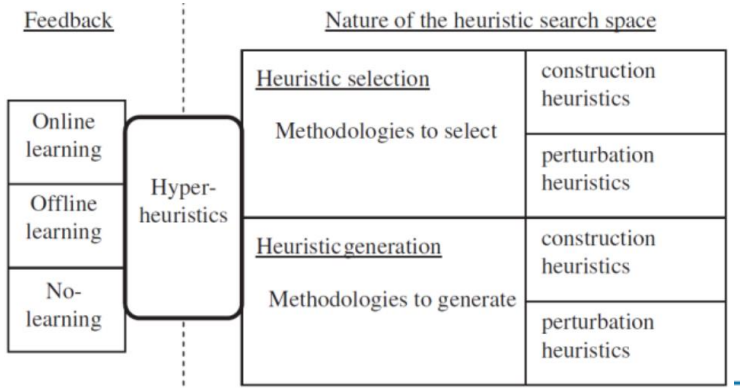
  - Research issues
    - Two search spaces
    - Landscape analysis on heuristic space



IEEE Computational Intelligence Society

AutoDes – Selection Hyper-heuristics

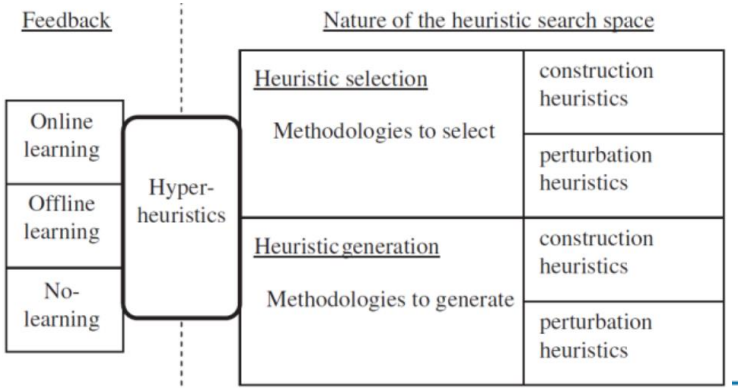
- Low level heuristics: Perturbative
  - Improves candidate solutions
  - Heuristic selection, acceptance criteria
- Research issues
  - Online learning
  - Reinforcement learning
- Cross Domain Heuristic Challenge (CHeSc)
- HyFlex



IEEE Computational Intelligence Society

AutoDes – Selection Hyper-heuristics

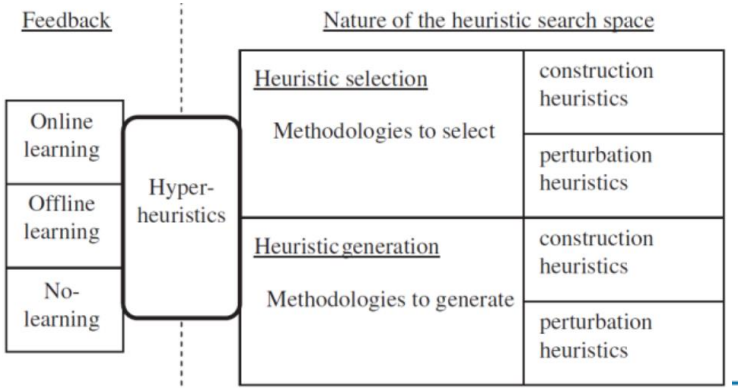
- Learning to select an appropriate / elite set of low-level heuristics / components
  - Online learning
    - Different low level heuristics effective at different stages
    - Step-by-step reduction during the search, snapshot performance
  - Offline learning
    - Evaluation of collective / accumulative performance
    - Statistical analysis, landscape probing



IEEE Computational Intelligence Society

AutoDes – Selection Hyper-heuristics

- Learning to select / compose low-level heuristics / components
  - Online learning
    - Select / predict the most suitable low level heuristics based on their performance during the search
    - Reinforcement learning: Markov chain / models, choice function
    - States: problem-specific features, general / problem independent features
  - Offline learning
    - Choose low-level heuristics or acceptance criteria based on offline training
    - Classification models, logistic regression, neural networks, apprenticeship learning, etc.

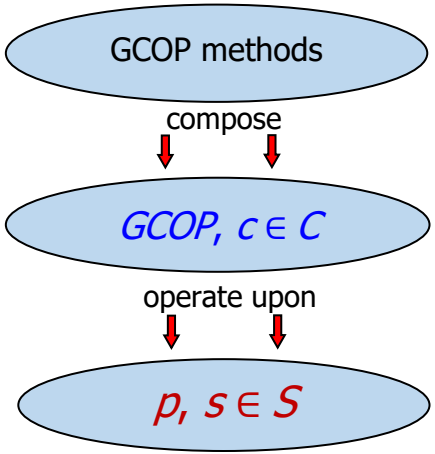


IEEE Computational Intelligence Society

Modelling and Learning in Automated Algorithm Composition

AutoDes – The GCOP Model

- General Combinatorial Optimisation Problem
  - Decision variables: algorithmic components ***a***
- GCOP methods
  - Search for algorithmic components ***a***
    to find algorithmic compositions ***c*** in an algorithm space ***C***
    c match direct solutions ***s*** in the solution space *S* for ***p***
  - Automated algorithm composition

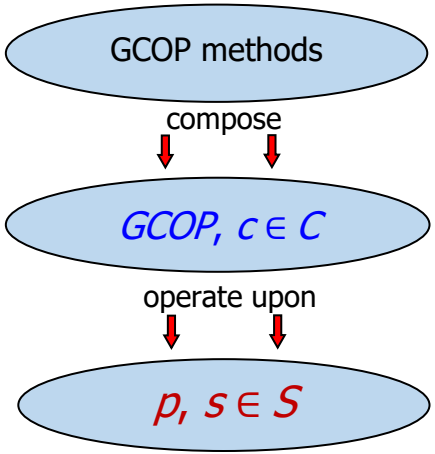


	GCOP Space *C*	Solution space *S*
Encoding	Compositions c upon $a \in A$	Direct solutions $s \in S$ on p
Upper Bound	Depends on |A| and parameters of $a \in A$	Depends on the number of variables in s for p
Operator	Any methods composing a into c	Search operators on $s \in S$
Objective Function	Performance of c that produces s	Solution quality of $s \in S$ for p

AutoDes – The GCOP Model

- The algorithmic 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 matching function ***M***: *f(s)* → *F(c)*
- The objective of GCOP: to find optimal *c**

$$F(c^*|c^* \rightarrow s^*) \leftarrow f(s^*) = \mathbf{min}(f(s))$$



AutoDes – The GCOP Model

- Modelling of VRP and NRP algorithms

$a \in A_{1.0}$	a **in GCOP for solving NRP** $h1_w$: selection criteria such as the cost of constraint violations, shift type balance, etc.
$o_{chg}(k,h1_w,h1_b)$	*change shift*: use $h1_b$ to change the shift type of k nurses chosen by $h1_w$.
$o^{bw}_{xchg}(k,k,h1_w)$	*swap shifts*: swap k shifts between two nurses chosen by $h1_w$.
$o_{rr}(k,h1_w,h1_b)$	*ruin and recreate*: use $h1_b$ to reassign all k shifts of a set of nurses chosen by $h1_w$.

	$a \in A_{1.0}$ **in GCOP for VRP** $h1_w$, $h1_b$: selection criteria/heuristics
$o_{ins}(k,h1_w,h1_b)$	*greedy*, *insertion* [30]: insert k nodes chosen by $h1_w$ to a route chosen by $h1_b$.
$o_{chg}(k,h1_w,h1_b)$	*shift* [31]: use $h1_b$ to change k nodes selected by $h1_w$.
$o_{xchg}(k,m,h1_w)$	*k-opt* [31], *interchange*, *Van Breedam* [32]: swap k and m nodes selected by $h1_w$.
$o_{xo}(k,m,h2_b)$	*crossover*: exchange sub-routes of k and m nodes between two solutions chosen by $h2_b$.
$o_{rr}(k,h1_w,h1_b)$	*destroy and repair*: remove k nodes chosen by $h1_w$, and re-assign them using $h1_b$.

IEEE Computational Intelligence Society

IEEE

AutoDes – The Framework

- General Search Framework [Yi22]
  - Automated Algorithm Composition

TABLE I
COMPONENTS WITHIN THE GENERAL SEARCH FRAMEWORK

Component	Criteria
Initialization	random, problem-specific heuristics
Selection for evolution	probability-based operators, deterministic operators
Evolution	mutation, crossover
Selection for replacement	comma-selection, plus-selection
Termination	time, convergence

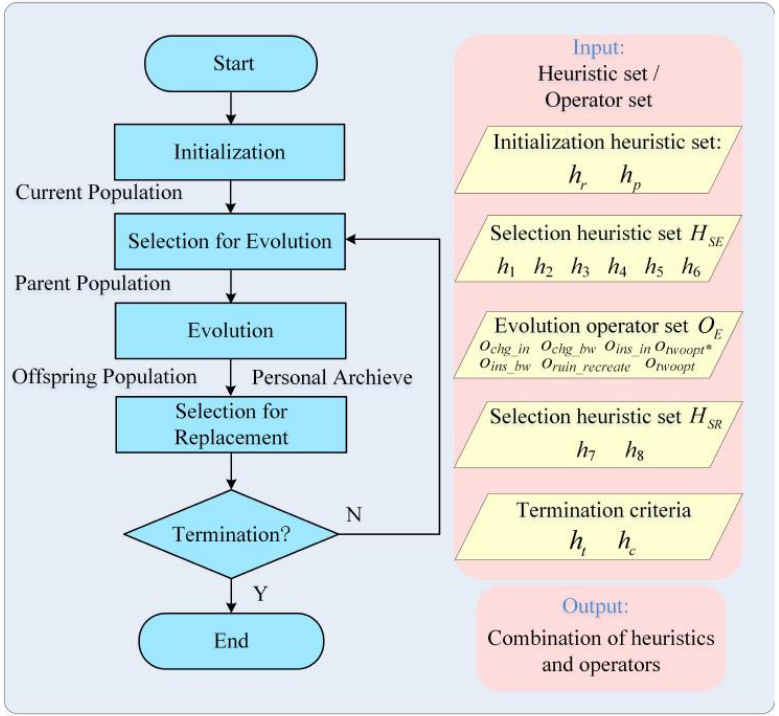


Fig. 1. General search framework

AutoDes – The Framework

- Learning on automated algorithm composition [Men22]

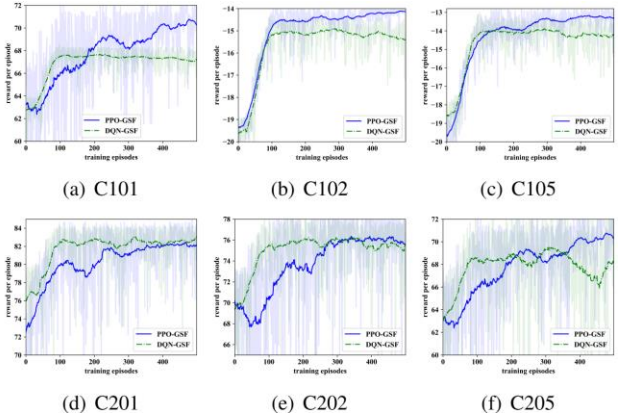


Figure 1: The M_{IP} updated during six iterations

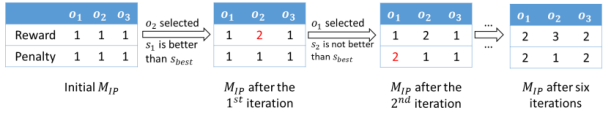
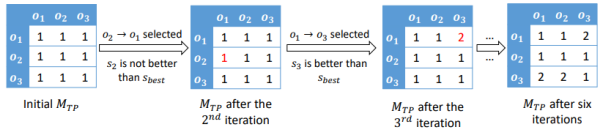
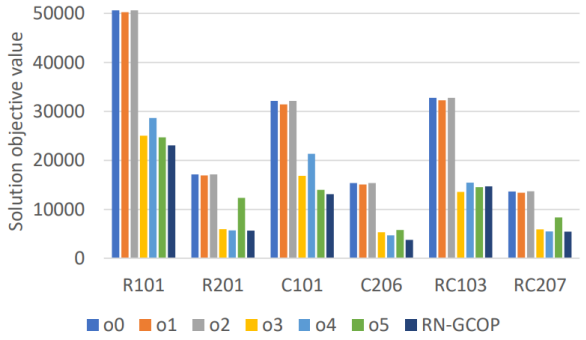


Figure 2: The M_{TP} updated during six iterations



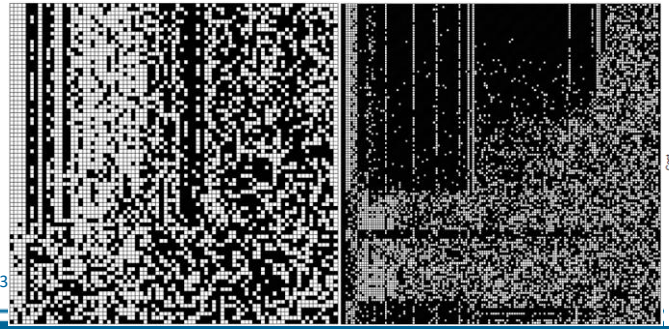


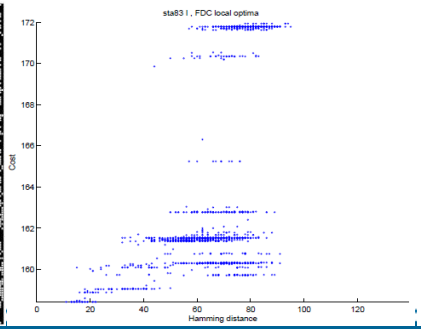
IEEE Computational Intelligence Society

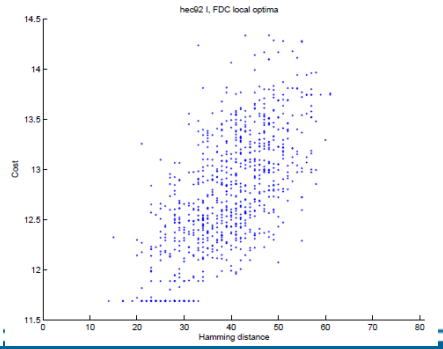
IEEE

AutoDes – Fundamental Issues

- Within unified algorithm design framework
  - Learning on heuristic / components compositions
  - Search space and landscape analysis of high level heuristic compositions *c*
    - High level heuristic compositions *c:* one-dimensional string
    - Easy to measure distances / differences: simpler solution encoding
    - Distribution of costs for local optimal *c*
    - Fitness distance correlation (*fdc*) of local to global optimum







AutoDes – Future Research

- Theory
  - Modelling and standardisation of algorithm design
  - General framework / platforms
  - Search space / landscape analysis
  - Common problem representation / encoding
- Machine learning + optimisation
  - Hidden patterns / new knowledge
  - Reusability and interpretability

References

- [Aka17] R. Akay, A. Basturk, A. Kalini, X. Yao. Parallel population-based algorithm portfolios: An empirical study. Neurocomputing, 247: 115-125, 2017
- [Hut09] F. Hutter, H.H. Hoos, K. Leyton-Brown, T. Stützle. Paramils: an automatic algorithm configuration framework. Journal of Artificial Intelligence Research, 36:267–306, 2009
- [Bir10] M. Birattari, Z. Yuan, P. Balaprakash, T. Stützle. F-race and iterated f-race: An overview. In Experimental methods for the analysis of optimization algorithms, 311–336. Springer, 2010
- [Bur13] E.K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, R. Qu. Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society, 64(12):1695–1724, 2013
- [Liu19] S. Liu, K. Tang, X. Yao. Automatic Construction of Parallel Portfolios via Explicit Instance Grouping. The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)
- [Lop12] M. Lopez-Ibanez, T. Stutzle. The automatic design of multiobjective ant colony optimization algorithms. IEEE Transactions on Evolutionary Computation, 16(6):861–875, 2012
- [Men22] W. Meng, R. Qu. Automated Design of Search Algorithms: Learning on algorithmic components. Expert Systems w. Applications, 2022
- [Pag19] F. Pagnozzi, T. Stützle. Automatic design of hybrid stochastic local search algorithms for permutation flowshop problems. European Journal of Operational Research, 276(2): 409-421, 2019
- [Pil17] N. Pillay, D. Beckedahl. Evohyp - a java toolkit for evolutionary algorithm hyper-heuristics. In 2017 IEEE Congress on Evolutionary Computation (CEC), 2706–2713. IEEE, 2017.
- [Pil18] N. Pillay, R. Qu, D. Srinivasan, B. Hammer, K. Sorensen. Automated design of machine learning and search algorithms [guest editorial]. IEEE Computational intelligence magazine, 13(2):16–17, 2018
- [Qu20] R. Qu, G. Kendall, N. Pillay. The General Combinatorial Optimisation Problem - Towards Automated Algorithm Design. IEEE Computational Intelligence Magazine, 15(2): 14-23, May, 2020
- [Tan14] K. Tang, F. Peng, G. Chen, X. Yao. Population-based algorithm portfolios with automated constituent algorithms selection. Information Sciences, 279:94–104, 2014
- [Xu10] L. Xu, H. Hoos, K. Leyton-Brown. Hydra: Automatically configuring algorithms for portfolio-based selection. In Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010
- [Yi22] W. Yi, R. Qu, L. Jiao, B. Niu. Automated Design of Metaheuristics Using Reinforcement Learning within a Novel General Search Framework. Under revision at IEEE TEVC, 2022