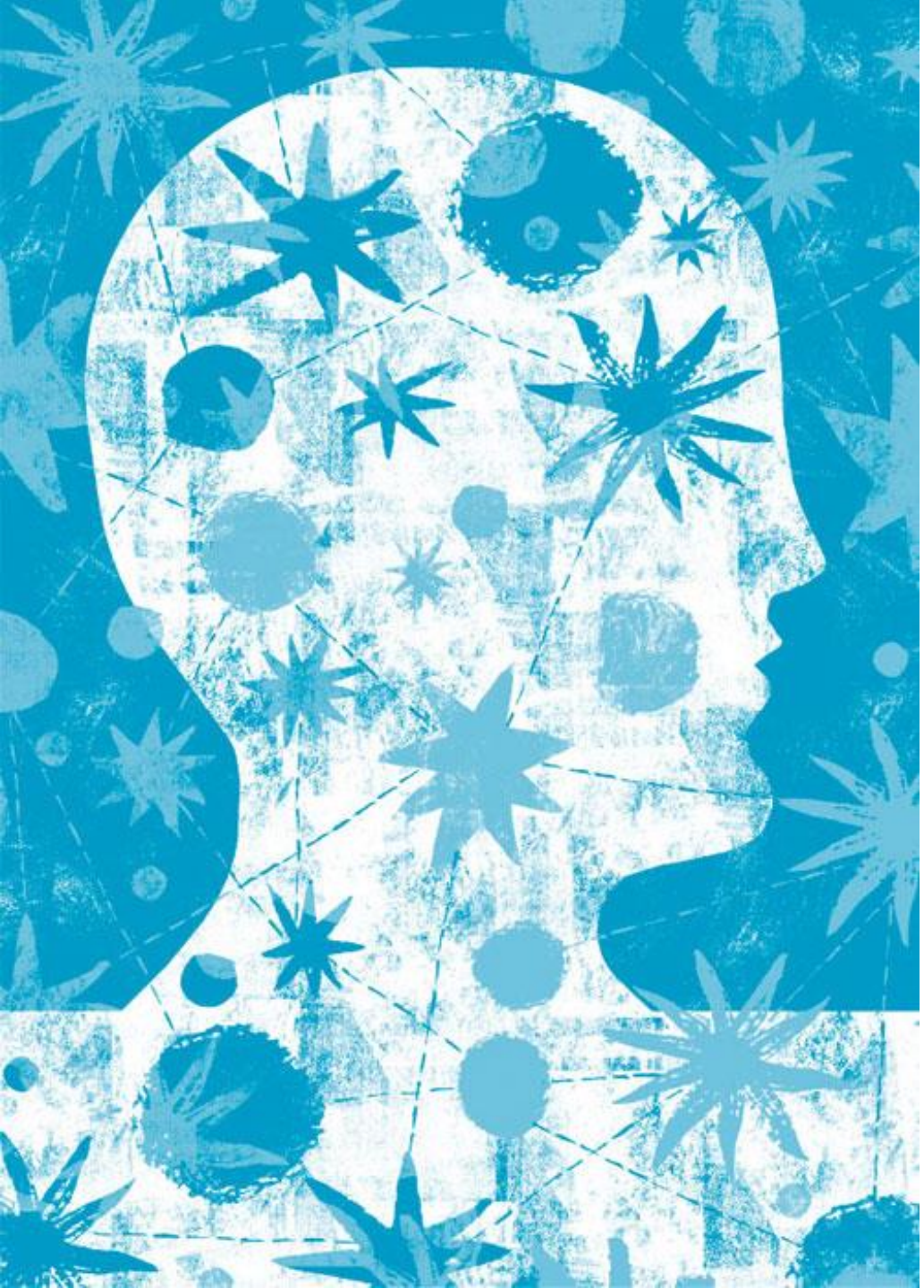


Edinburgh Napier UNIVERSITY

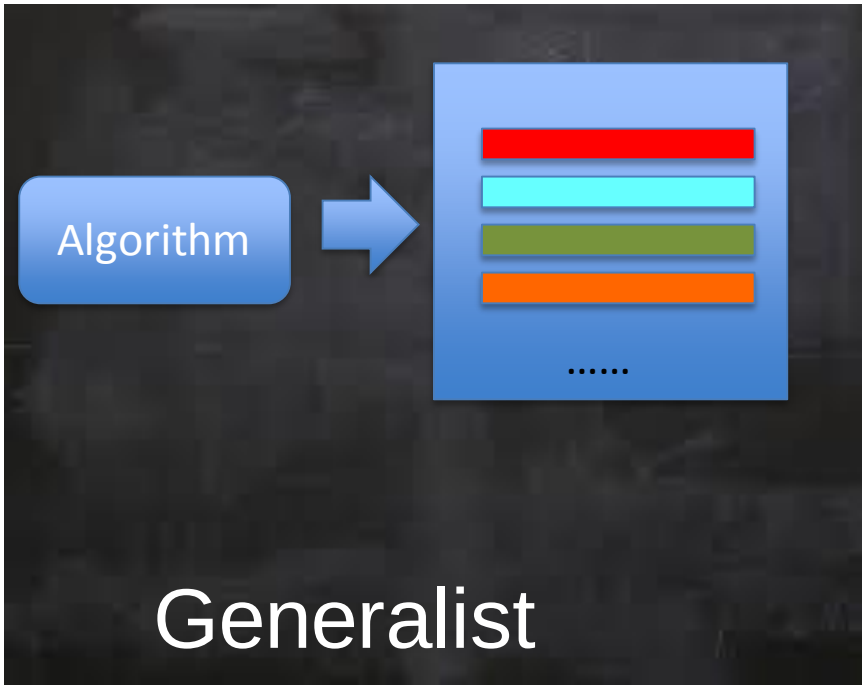
Lifelong Learning in Optimisation

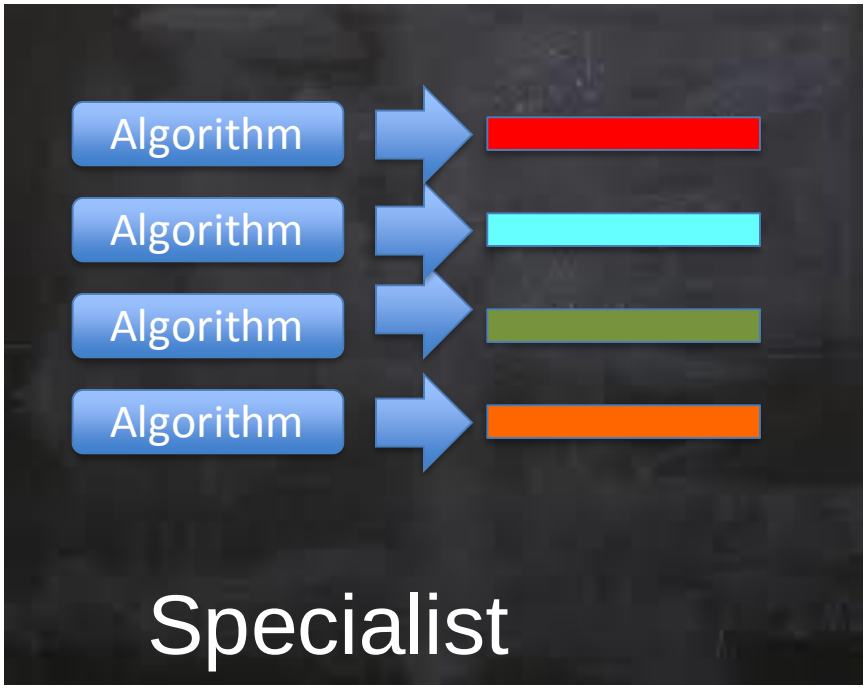
Emma Hart

Edinburgh Napier University

http://jamesobrien.tumblr.com/post/1112777561/lifelong-learning-illustration

Optimisation Algorithms

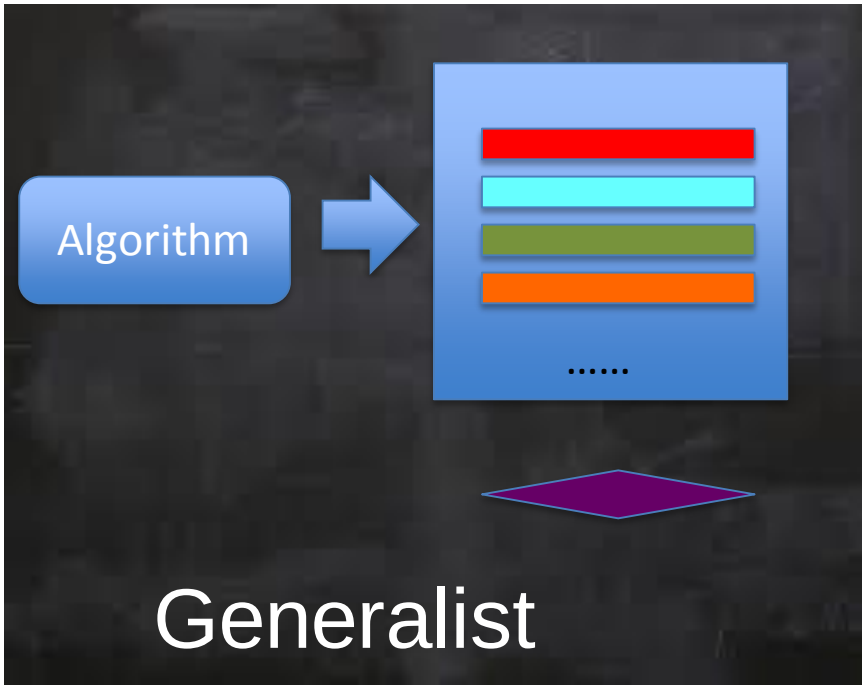




Tuned metaheuristics
Offline hyper heuristics
….

Evolution Strategies
Online hyper heuristics
Metaheuristics…

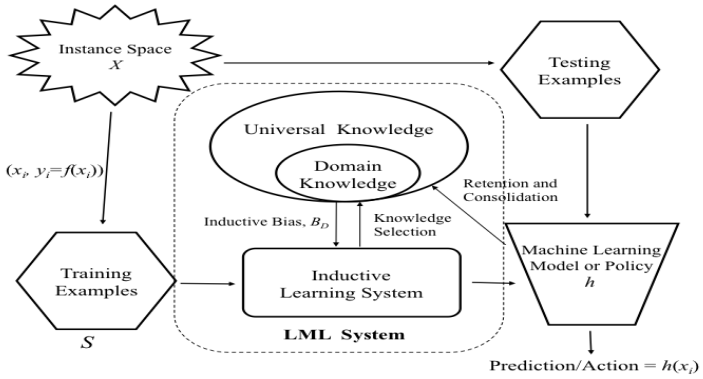
Optimisation Algorithms



- Generalist: incapable of adapting to new problem characteristics
- Specialist: unable to learn from experience or exploit prior knowledge

Machine Learning

- Contemporary ML systems usually exploit prior knowledge if faced with new but similar task



"it is now appropriate for the AI community to move beyond learning algorithms to more seriously consider systems that are capable of learning over a lifetime"

Silver, 2013

Machine Learning

An LML should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Selectively transfer prior knowledge when learning new tasks
3. Adopt a systems approach that ensures effective and efficient interaction of elements of the system

What kind of approach might provide these features ?

Machine Learning

An LML should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Selectively transfer prior knowledge when learning new tasks
3. Adopt a systems approach that ensures effective and efficient interaction of elements of the system

Natural Immune System

Basis of vaccination, can be very long term

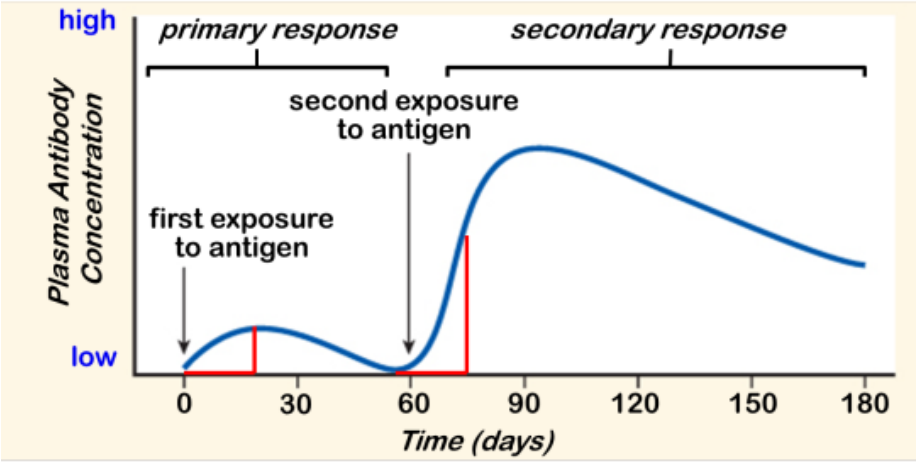
Machine Learning

An LML should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Selectively transfer prior knowledge when learning new tasks
3. Adopt a systems approach that ensures effective and efficient interaction of elements of the system

Natural Immune System

Basis of vaccination, can be very long term



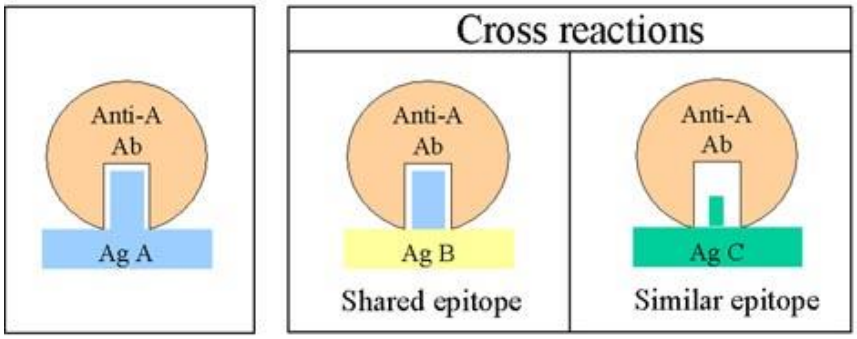
Machine Learning

An LML should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Selectively transfer prior knowledge when learning new tasks
3. Adopt a systems approach that ensures effective and efficient interaction of elements of the system

Natural Immune System

Selectively transfer prior knowledge when learning new tasks



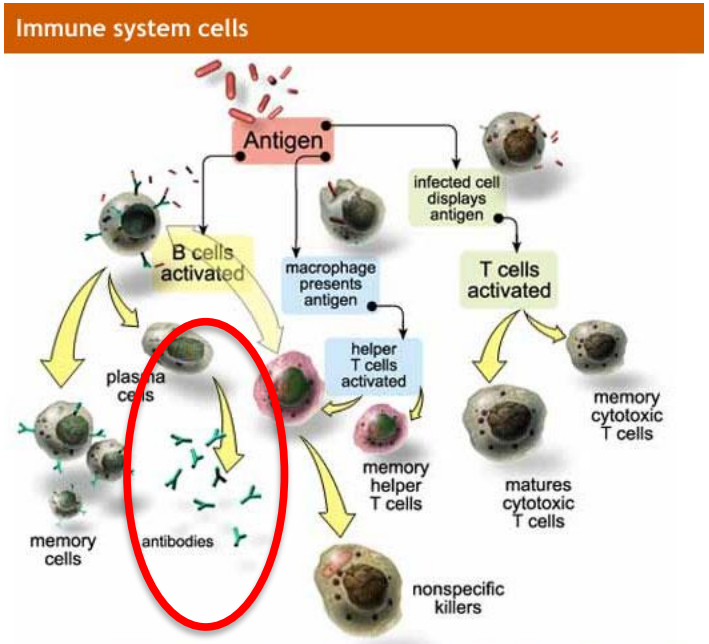
Machine Learning

An LML should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Selectively transfer prior knowledge when learning new tasks
3. Adopt a systems approach that ensures effective and efficient interaction of elements of the system

Natural Immune System

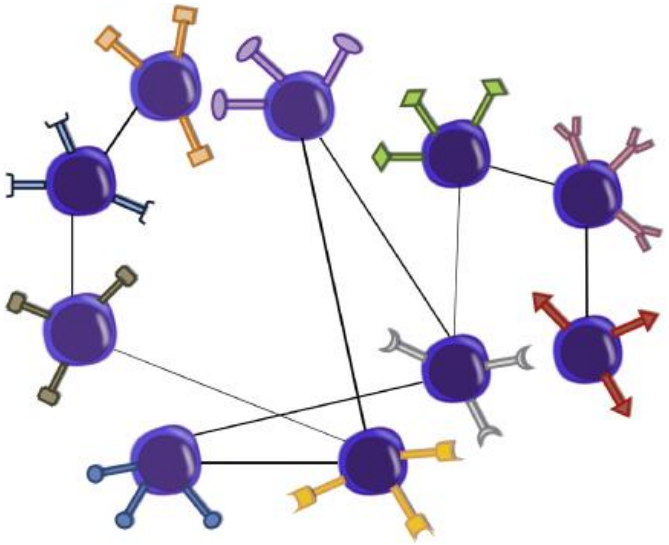
Behaviour is the result of many interacting components



(Source: the Human Immune Response System www.uta.edu/chagas/images/immunSys.jpg)

The Role of the Immune Network

- Immune cells interact with other and with antigen
  - Can be stimulatory or suppressive
- Results in a network with dynamically changing topology
- Useful cells recruited into network
- Redundant ones rejected
- Topology depends on past & current environment



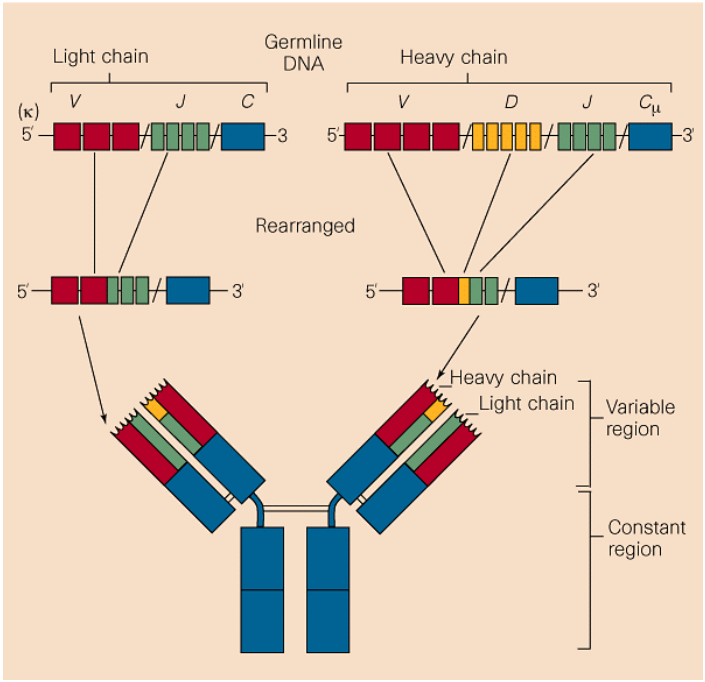
Machine Learning

An LML should:

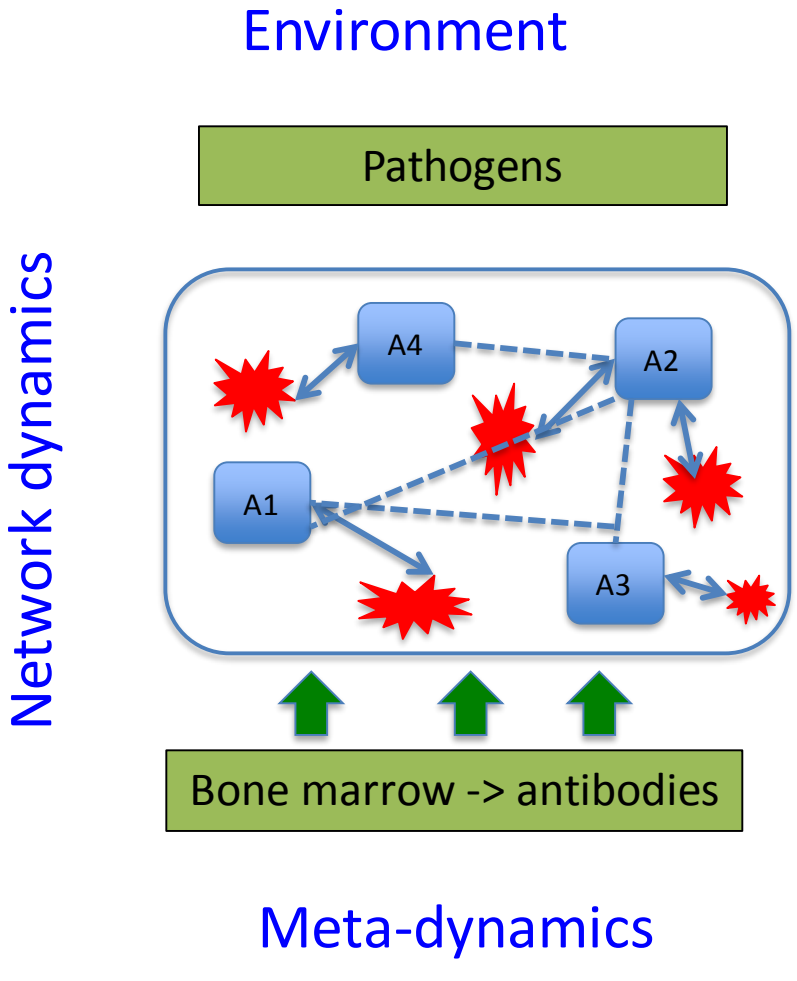
1. Retain and/or consolidate knowledge (long-term memory)
2. Selectively transfer prior knowledge when learning new tasks
3. Adopt a systems approach that ensures effective and efficient interaction of elements of the system
4. *Generate new knowledge*

Natural Immune System

Gene recombination in bone marrow continually trials news cells leading to a useful repertoire of antibodies



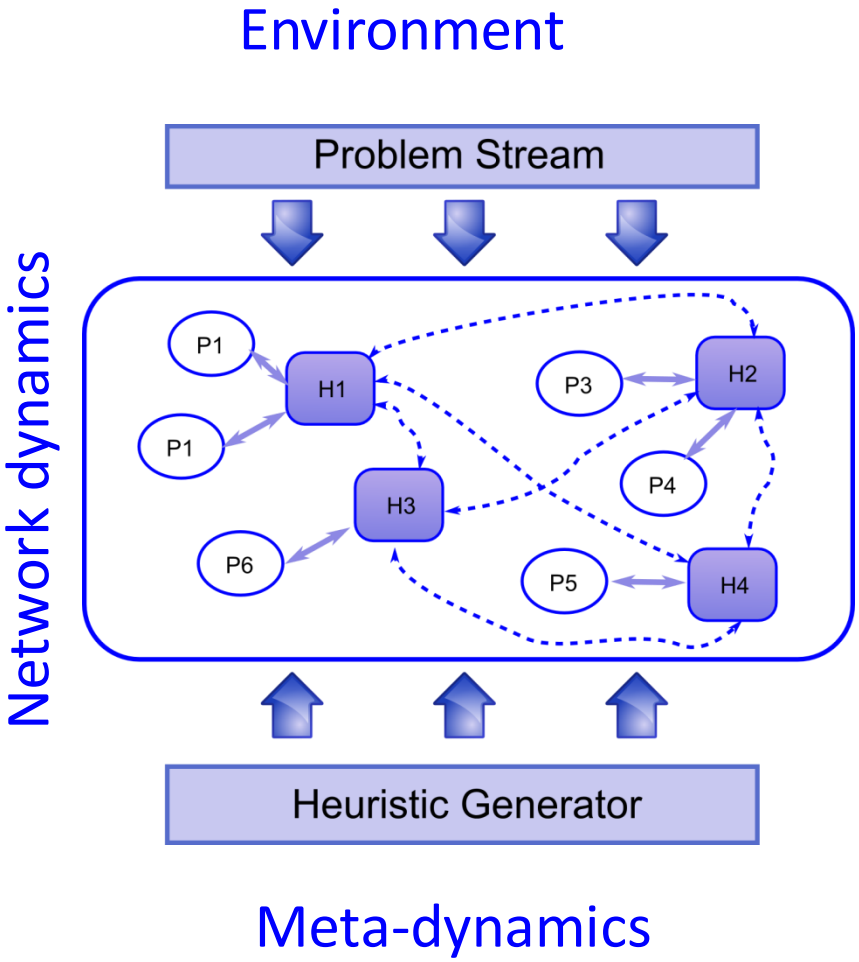
Immune Systems



Computational Properties

- *Exploration*
  - randomly combining components from a library gives rise to many cells
- *Exploitation*
  - focuses search on promising cells
- *Memory*:
  - network provides a 'map' of the antigen space
- *Adaptable*
  - Doubly plastic: parametric & structural
- *Diverse*
  - Finite repertoire of cells has to ensure all pathogens recognised

Optimisation Systems

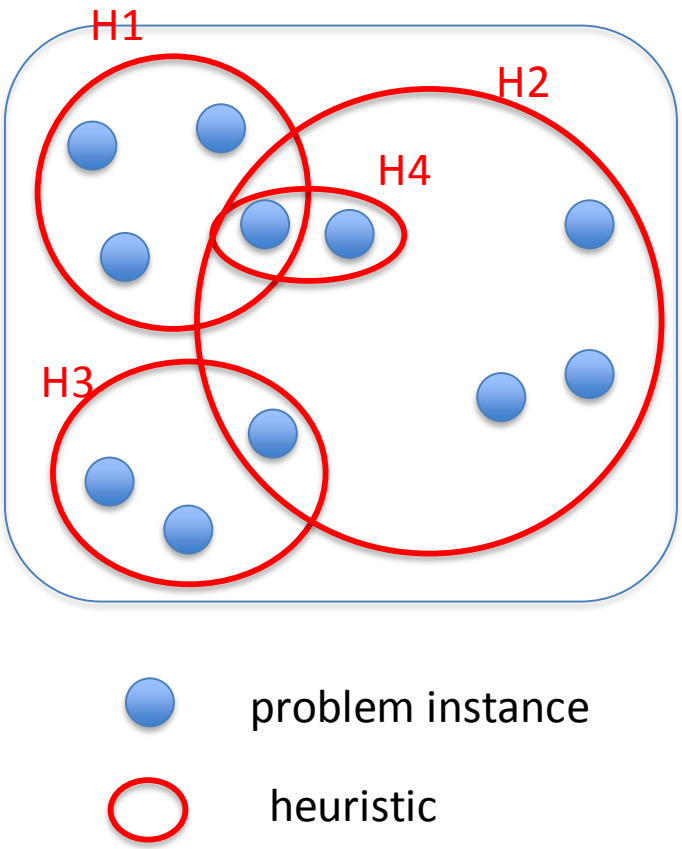


Computational Properties

- *Exploration*
  - randomly combining components from a library gives rise to many heuristics
- *Exploitation*
  - focuses search on promising heuristics
- *Memory*:
  - network provides a 'map' of the problem space
- *Adaptable*
  - Doubly plastic: parametric & structural to deal with changes in problem characteristics
- *Diverse*
  - Finite repertoire of heuristics has to ensure all problems solved

Conceptual Overview

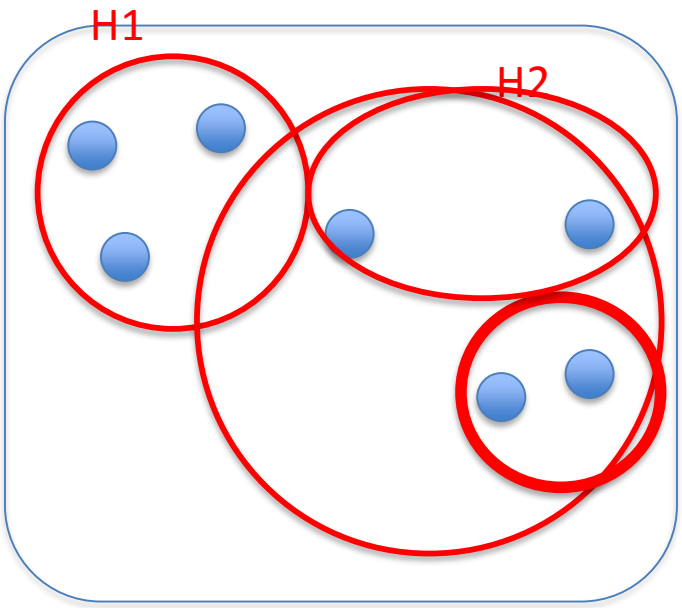
2d Representation of problem space



- The network sustains ***heuristics*** that work best in distinct regions of the instance space (diversity)
  - Need to win to be in!
- The network sustains ***problems*** that are representative of areas of the problem space
  - Problems that are solved by more than one heuristic are not 'interesting'
- Problems & heuristics gain concentration through mutual stimulation
  - Decay mechanisms enable gradual forgetting
  - Lack of stimulation leads to removal
- Topology of network changes over time depending on problems injected and heuristics generated

Conceptual Overview

2d Representation of problem space



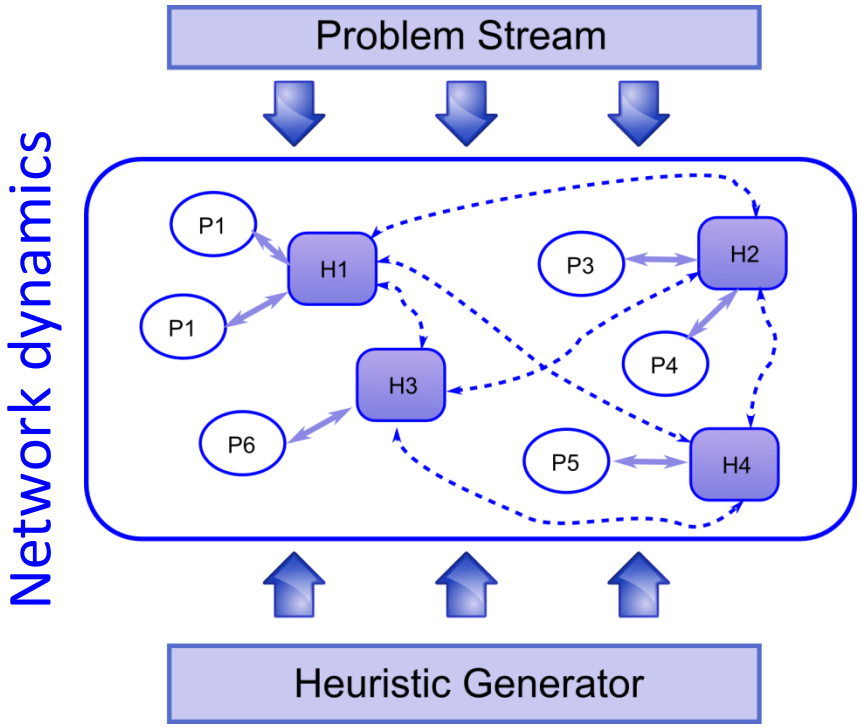
problem instance

heuristic

- The network sustains ***heuristics*** that work best in distinct regions of the instance space (diversity)
  - Need to win to be in!
- The network sustains ***problems*** that are representative of areas of the problem space
  - Problems that are solved by more than one heuristic are not 'interesting'
- Problems & heuristics gain concentration through mutual stimulation
  - Decay mechanisms enable gradual forgetting
  - Lack of stimulation leads to removal
- Topology of network changes over time depending on problems injected and heuristics generated

NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng

Environment



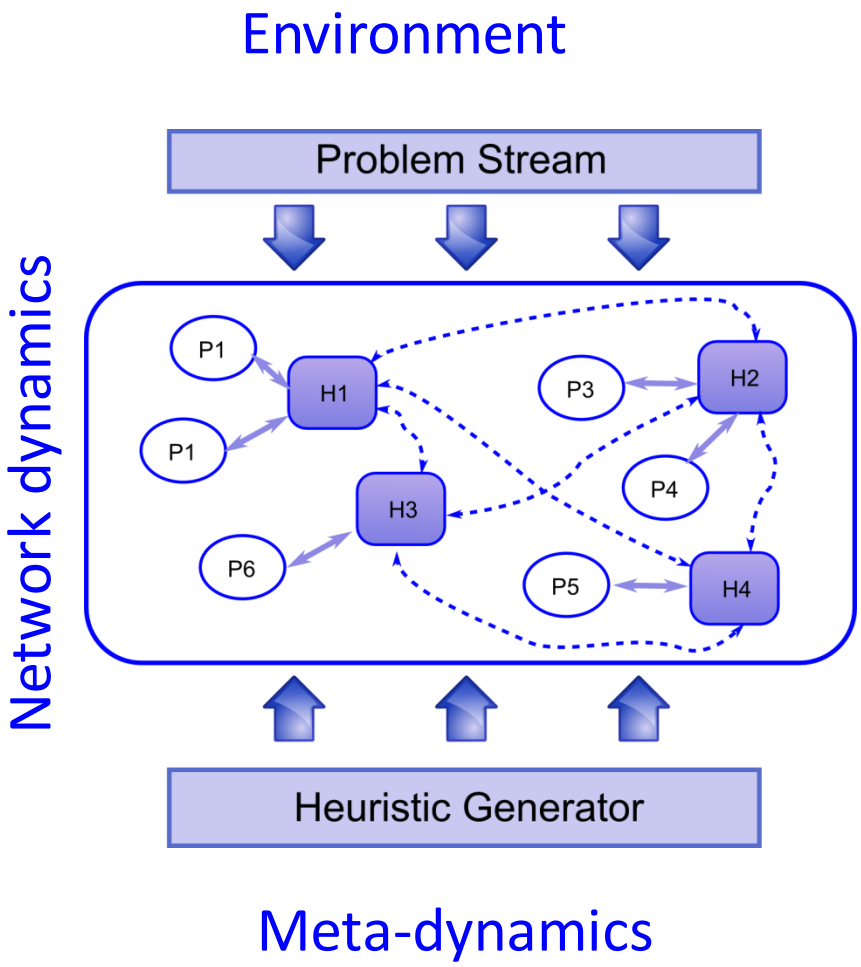
Meta-dynamics

① Problem Stream

② Heuristic Generator

③ Network of heuristics & problems

NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng

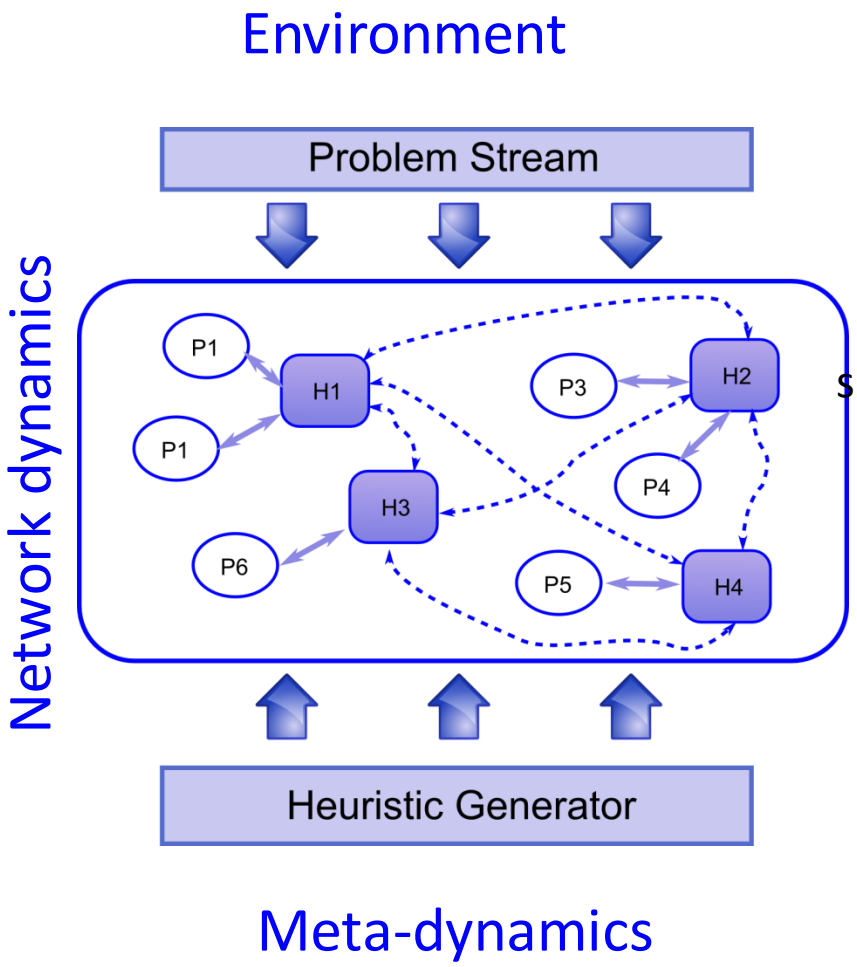


① Problem Stream

At each iteration instances *can* be injected into the system

- Single instance
- Multiple instances
- Frequent/infrequent

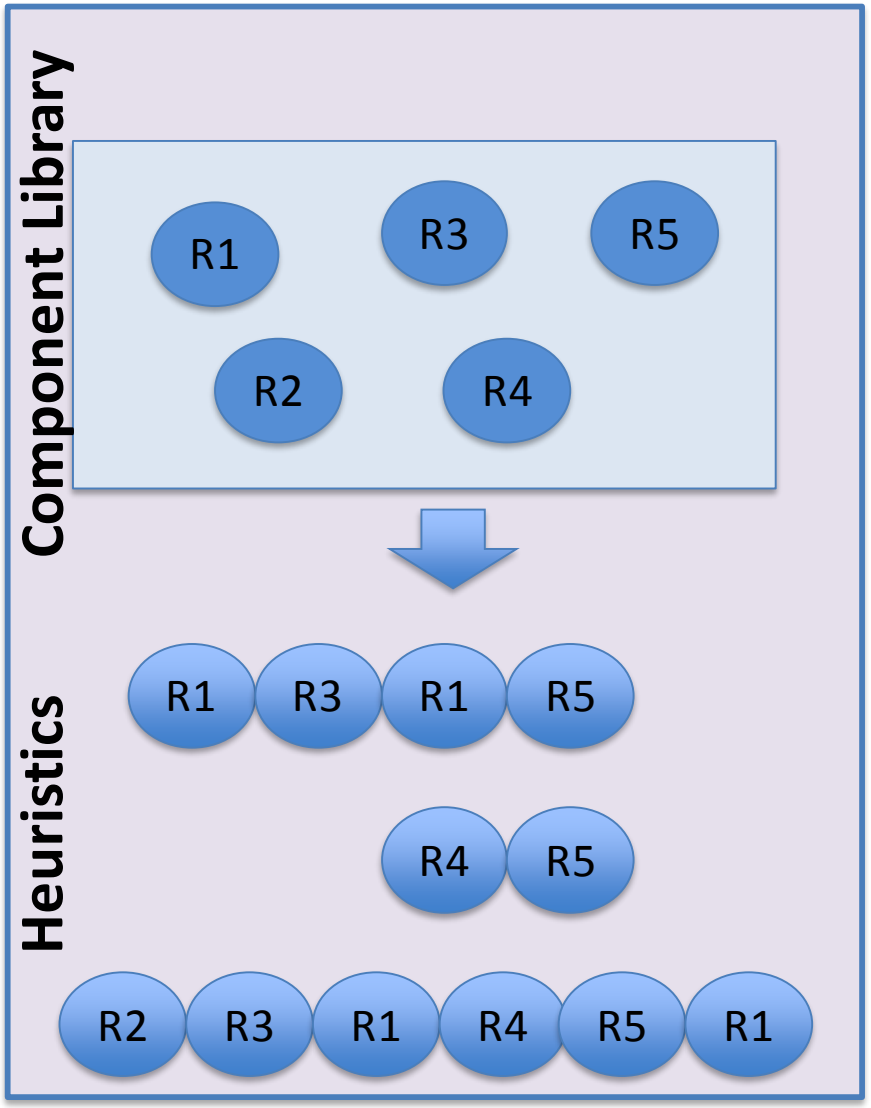
NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

- Library of components
- Components can be ‘pre-defined’ or evolved
- Components are combined into *heuristics*

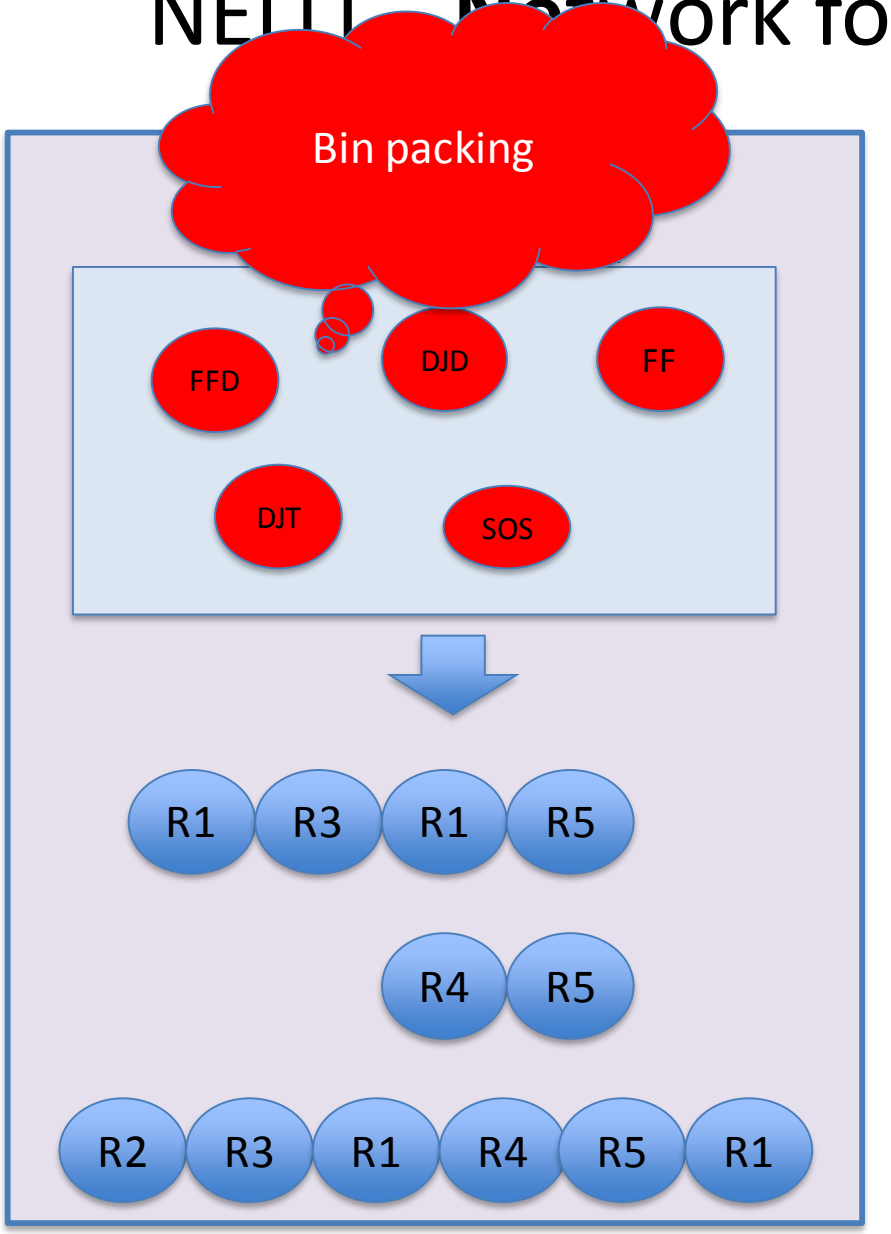
NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

- Library of components
- Components can be ‘pre-defined’ or evolved
- Components are combined into *heuristics*
- (few components -> lots of heuristics)

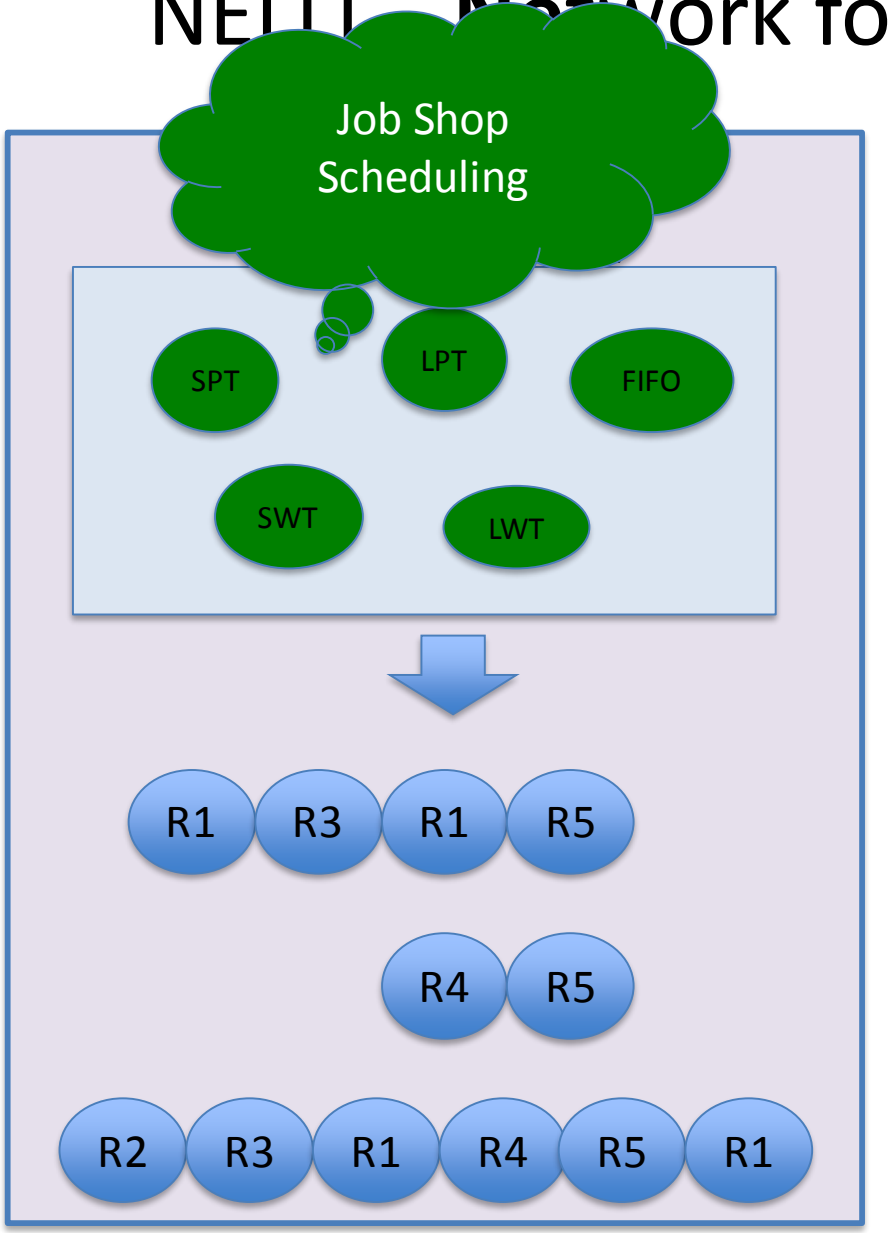
NELLI – **N**etwork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

- Library of components
- Components can be '**pre-defined**' or evolved
- Components are combined into *heuristics*

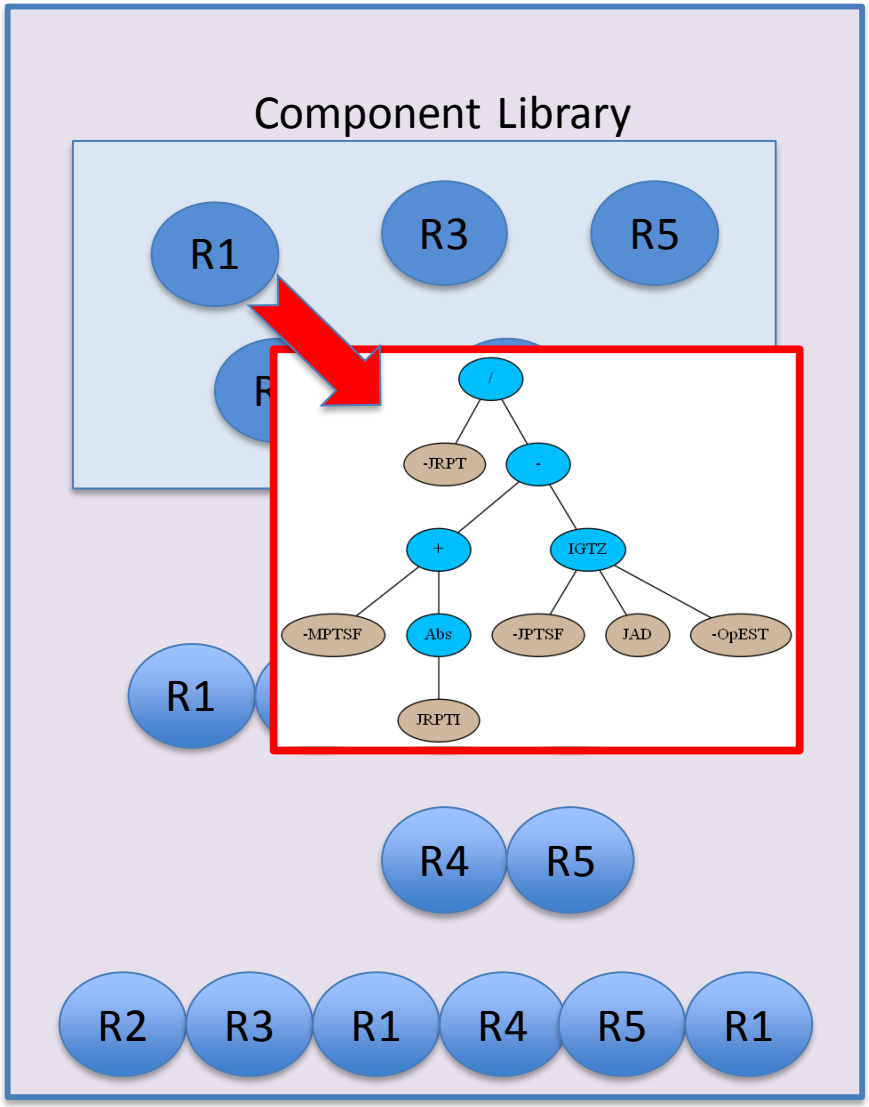
NELLI – **N**etwork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

- Library of components
- Components can be '**pre-defined**' or evolved
- Components are combined into *heuristics*

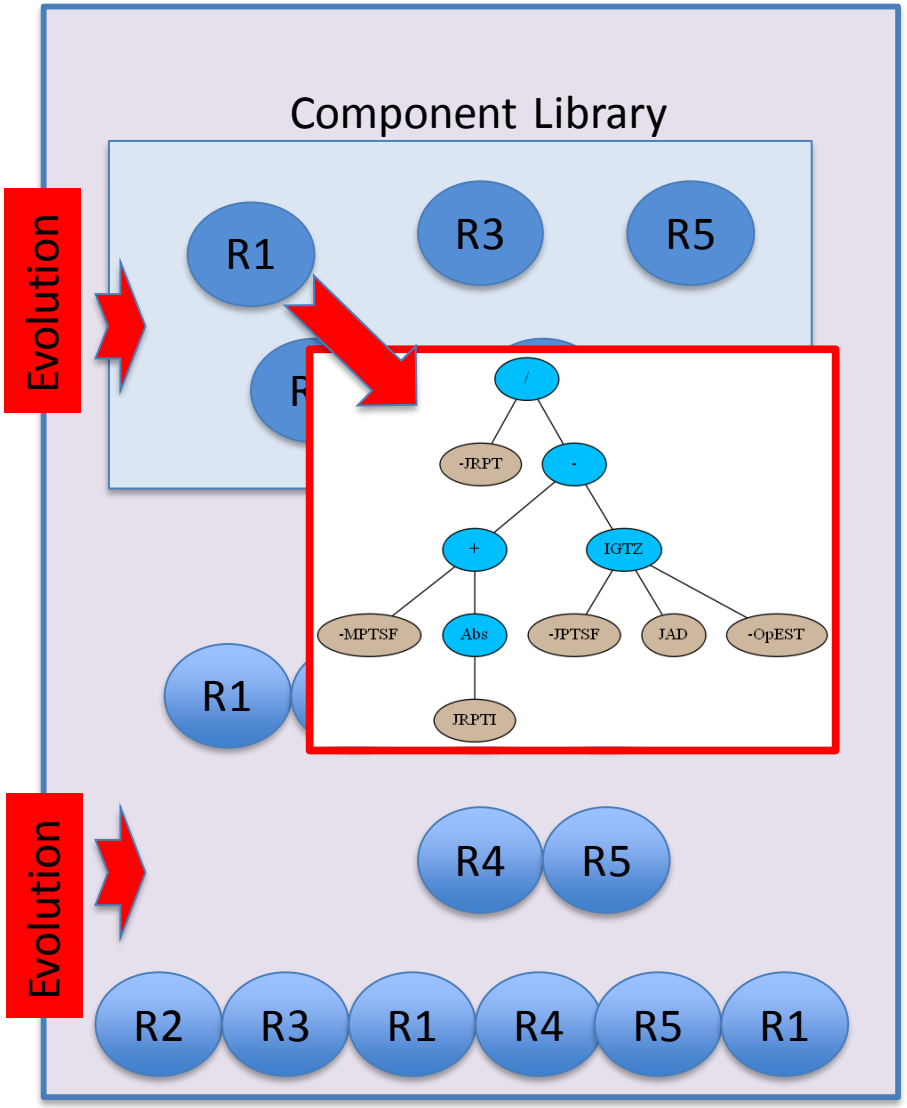
NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

- Library of components
- Components can be ‘pre-defined’ or **evolved**
- Components are combined into *heuristics*

NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng

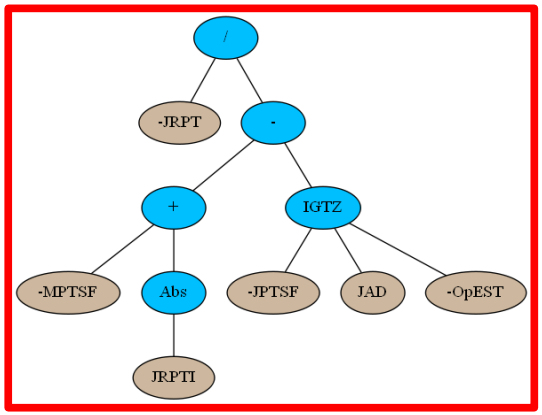


② Heuristic Generator

- Library of components
- Components can be ‘pre-defined’ or **evolved**
- Components are combined into *heuristics*
- Both **components** and **heuristics** can **evolve**

NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng

Evoluion

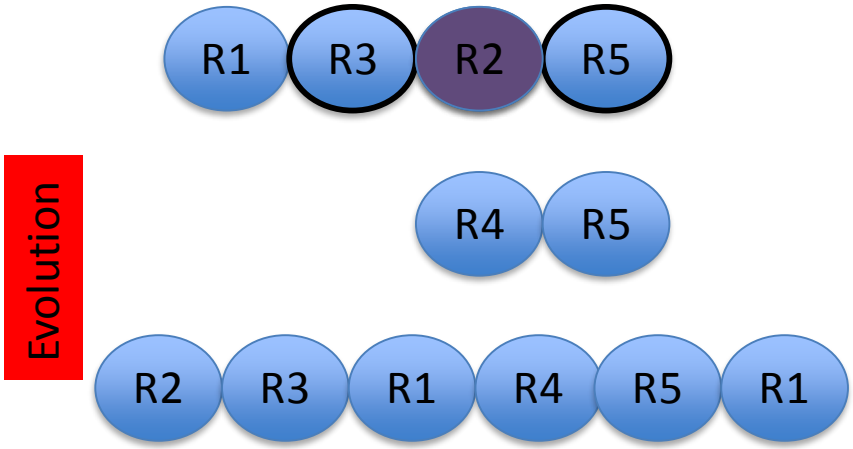


- Mutate terminal nodes
- Mutate function nodes
- Remove subtree
- Swap subtrees

② Heuristic Generator

- Library of components
- Components can be ‘pre-defined’ or **evolved**
- Components are combined into *heuristics*
- Both **components** and **heuristics** can **evolve**

NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng



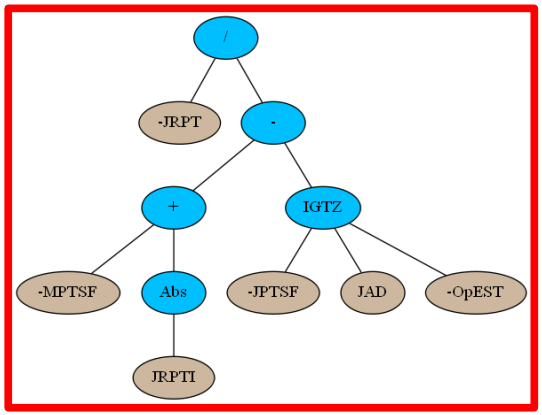
Swap components
Change components
Remove/insert components
Concatenate heuristics

② Heuristic Generator

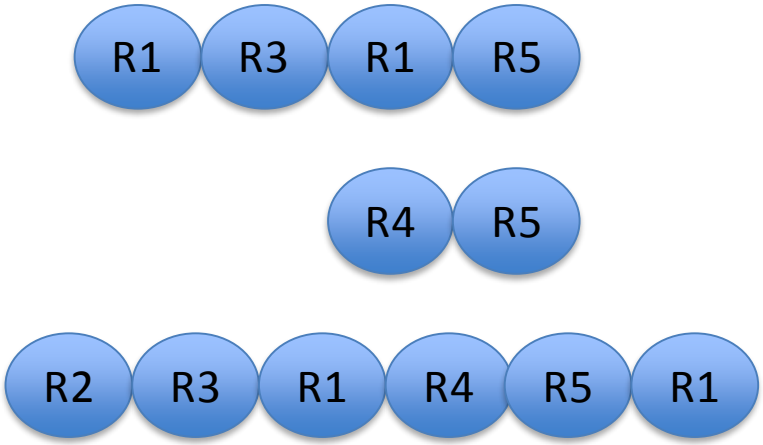
- Library of components
- Components can be ‘pre-defined’ or **evolved**
- Components are combined into *heuristics*
- Both **components** and **heuristics** can **evolve**

NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng

Evoluion



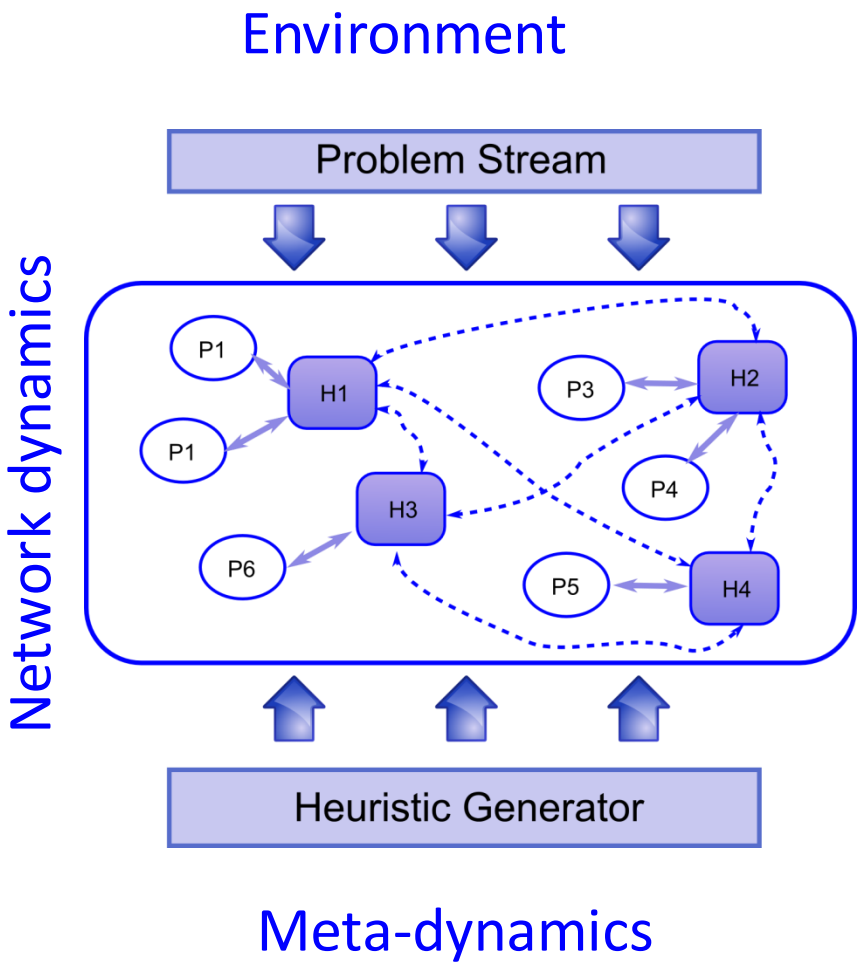
Evolution



② Heuristic Generator

- Library of components
- Components can be ‘pre-defined’ or **evolved**
- Components are combined into *heuristics*
- Both **components** and **heuristics** can **evolve**

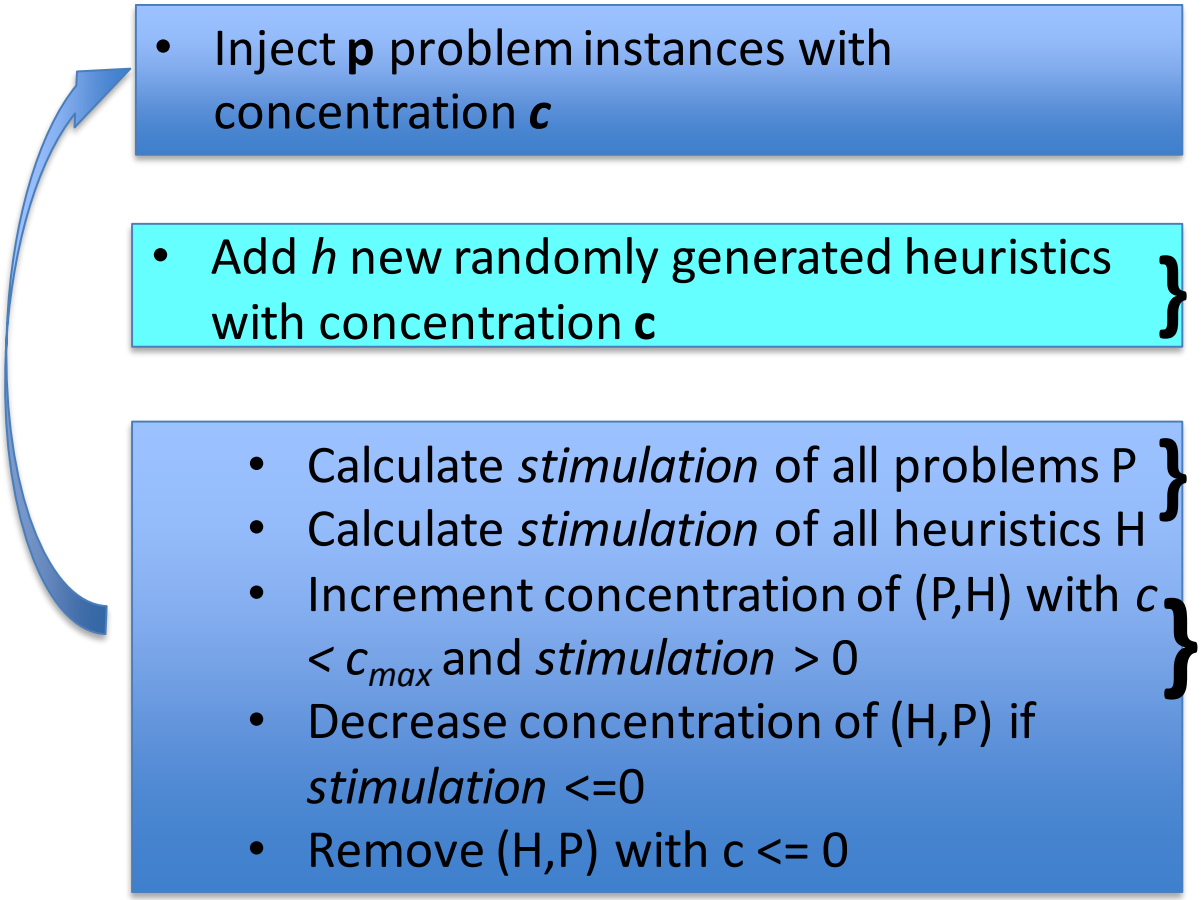
NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng



③ Network

- *Heuristics* are **stimulated** by winning at least one problem
  - The higher the win, the bigger the stimulation
- *Problems* are **stimulated** if they are won by only one heuristic
  - The higher the win, the bigger the stimulation

NELLI pseudo-code



Tune for more exploration

Depends on strength of win

Can be tuned to alter memory span & lifetime

Basic optimiser

- Bin packing:
  - 1370 instances from literature
  - **All presented at start**
  - Run for 500 iterations
- Number of heuristics evolved is an emergent property
- Number of problems retained gives insight into similarity of instances

	Problems solved	Extra bins
FFD	788	2142
DJD	716	2409
DJT	863	881
ADJD	686	1352
NELLI	**1126**	**308**

Heuristics Retained	7
Problems Retained	36

Sim & Hart: A Lifelong Learning Hyper-heuristic Method for Bin Packing, Evolutionary Computation, Spring 2015, Vol. 23, No. 1, Pages 37-67

Generalisation Capabilities

- Train NELLI by injecting p instances and running for t iterations
- Apply evolved heuristic network to an unseen test-set:
  - Bin packing (685 train, 685 test)
  - JSSP (train, test)

Bin packing

	Problems Solved	Extra Bins
Greedy Selection	548	188
AIS Model	559	159
Island Model	557	159
NELLI	**576**	**131**

Hart & Sim:, On the Life-Long Learning Capabilities of a NELLI*: A Hyper-Heuristic Optimisation System, PPSN 2014

Generalisation Capabilities

- Train NELLI by injecting *p* instances and running for *t* iterations
- Apply evolved heuristic ensemble to an unseen test-set:
  - Bin packing (685 train, 685 test)
  - JSSP (train, test)

65 Taillard instances (JSSP)

	T1	T2	T3
EGP-JSS	0.26+/- 0.04	0.26+/- 0.03	0.26+/- 0.010
NELLI	**0.20 +/- 0.09**	**0.18 +/- 0.03**	**0.18 +/- 0.04**

200 new instances (JSSP)

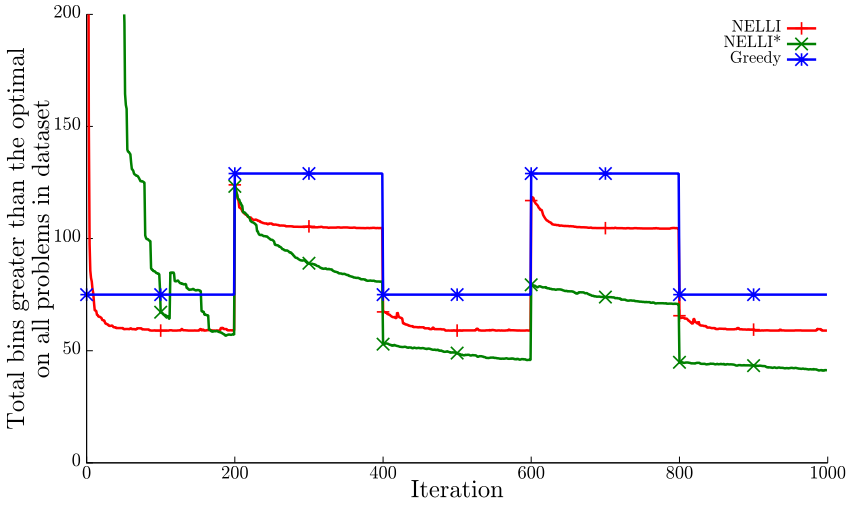
	T1
Greedy Selection	68146
GP(200P)	69795
GP(1P)	69068
NELLI	**68125**

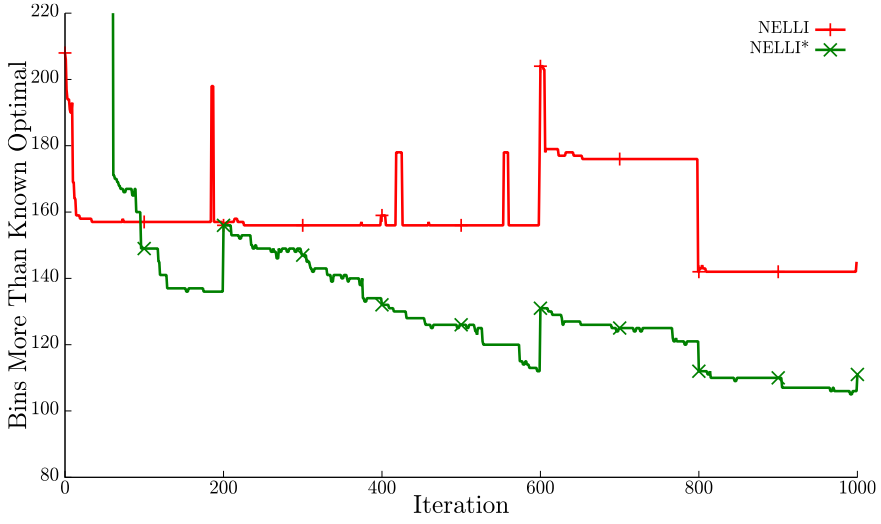
Hart & Sim (in review, J. Evolutionary Computation)

Memory

a) Alternate between two different datasets every 200 iterations

b) Present 685 randomly drawn instances each iteration

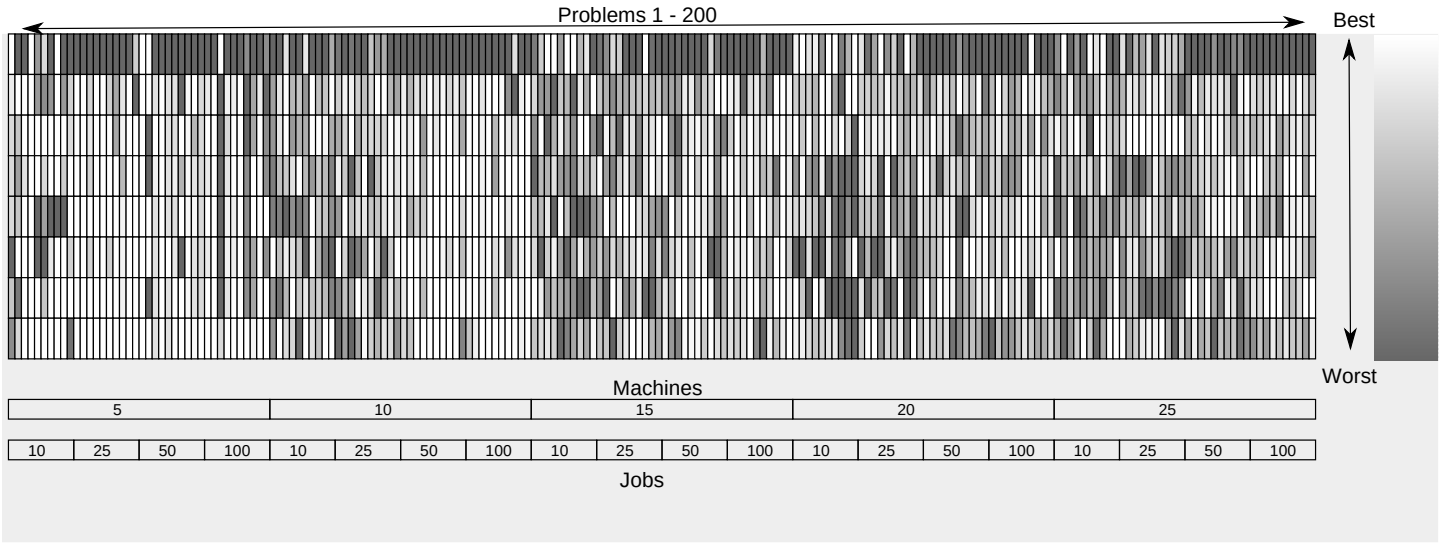




- Memory
- Learning over epoch
- Learning over lifetime

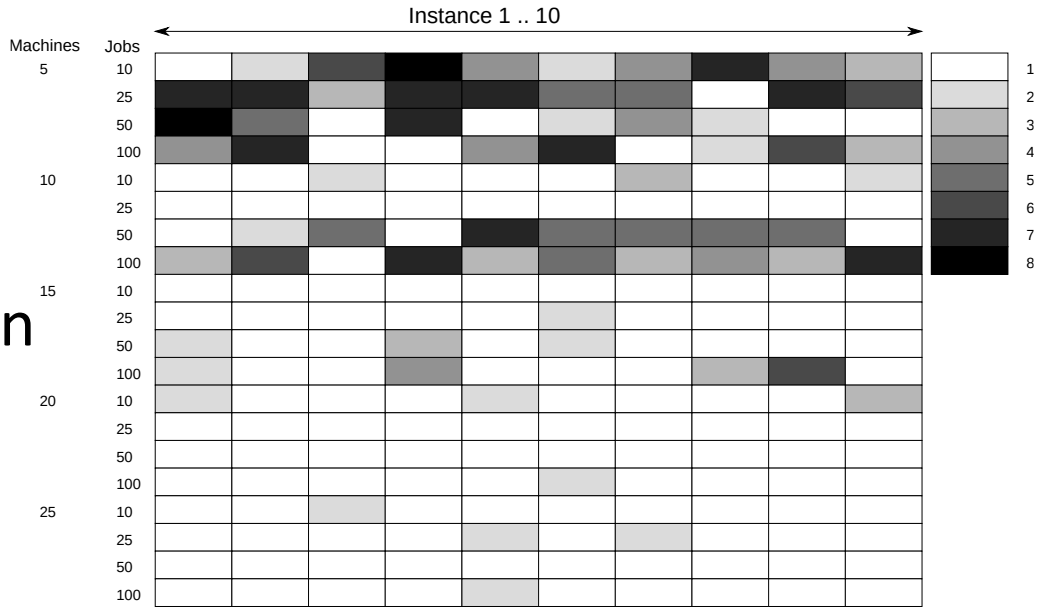
Insights into heuristic performance

- Define a **profile** (*barcode*) for each heuristic based on its relative performance on an instance
  - Results from an evolved ensemble of **8 heuristics** applied to 200 unseen JSSP instances
- Could be used to directly compare diversity of heuristics using a distance metric



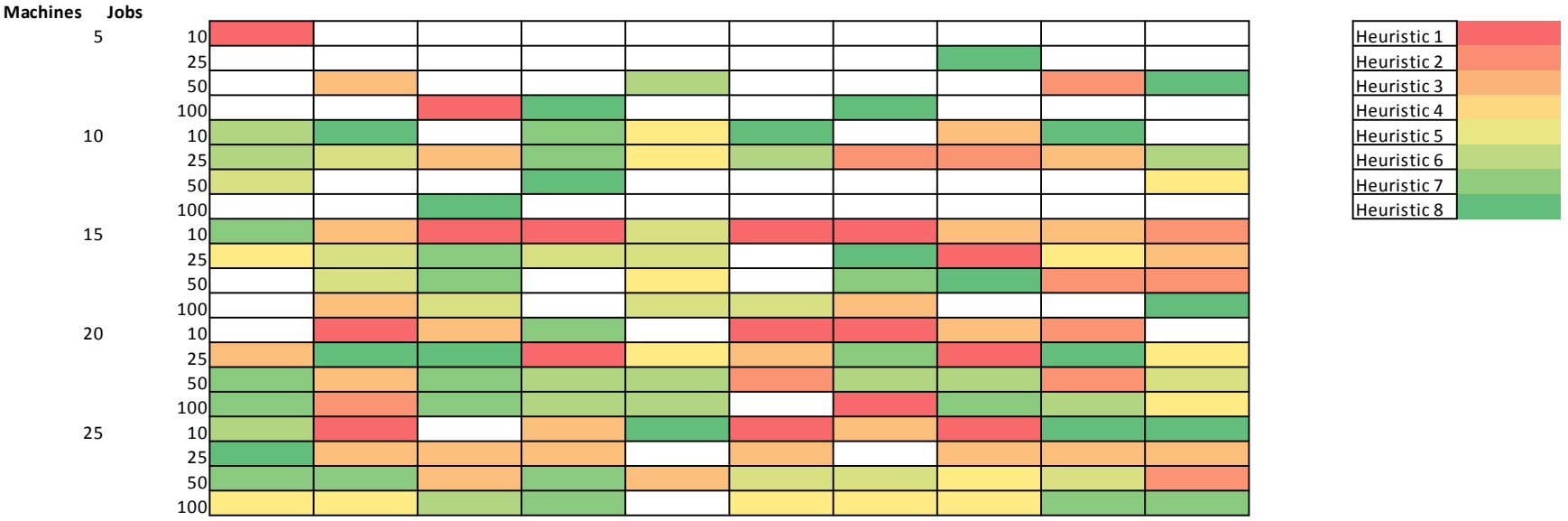
Insights into problem space

- Record how many heuristics ‘win’ each instance
- *Easy* problems won by many instances
- *Hard* problems only won by one heuristic
- Insights into relative difficulty of instances based on parameter combinations



Heuristic Perspective

- Where in the space does a particular heuristic perform well?
  - Can map heuristics to problem characteristics



Summary

- NELLI (Network for lifelong Learning)
  - *Generates novel heuristics that collaborate to cover instance space*
  - *Encapsulates memory*
  - *Generalises to new instances*
  - *Adapts to new instances*
  - *Better solutions than some other heuristic methods*

Conclusions

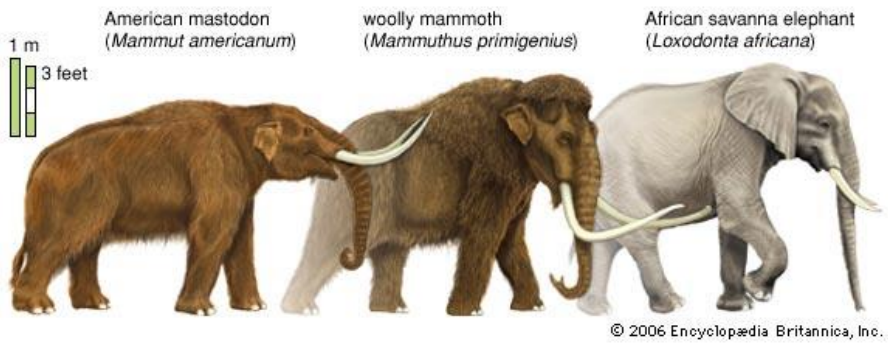
- **Optimisation systems should continuously learn**
  - *Exploit previous knowledge*
  - *Adapt to changing instance characteristics*

Conclusions

- Optimisation systems should continuously learn
  - *Exploit previous knowledge*
  - *Adapt to changing instance characteristics*
- **Optimisation systems that exploit ensembles are likely to be promising**
  - *No Free Lunch*
  - (*some) portfolio/multi-method algorithms*
  - *Machine learning community has plenty to say!*

Conclusions

- **Optimisation systems should continuously learn**
  - *Exploit previous knowledge*
  - *Adapt to changing instance characteristics*
- Optimisation systems that exploit ensembles are likely to be promising
  - *No Free Lunch*
  - (*some) portfolio/multi-method algorithms*
  - *Machine learning community has plenty to say!*
- **Ensemble methods like NELLI offer new insights into heuristic performance and problem difficulty**
  - *Create heuristic profiles that enable objective comparisons between heuristics and means of recognising diversity*
  - *Design better benchmarks*



THANK YOU!

EPSRC
Engineering and Physical Sciences Research Council

Acknowledgements

Dr Kevin Sim

EPSRC **EP/J021628/1**