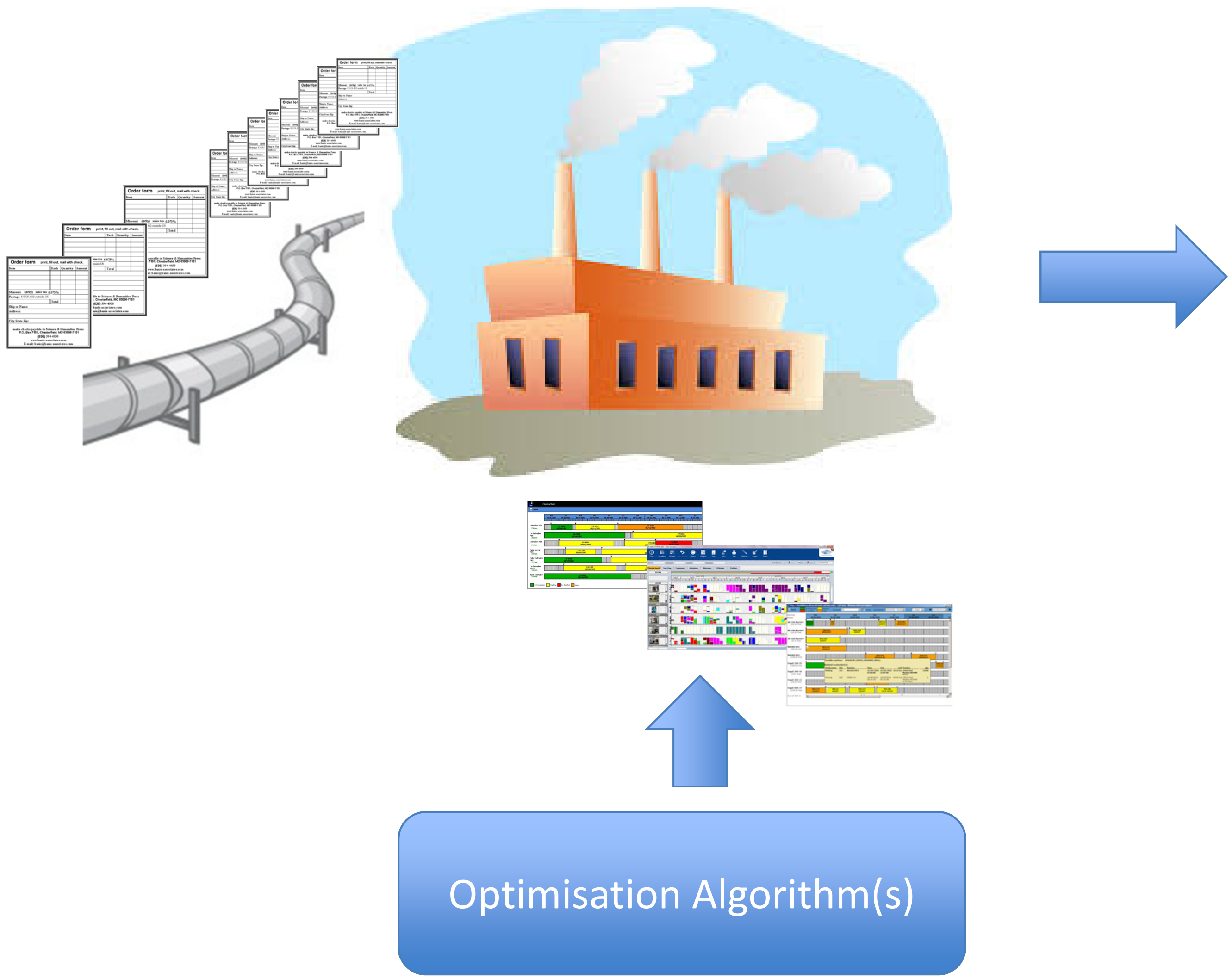
Edinburgh Napier UNIVERSITY

Lifelong Learning in Optimisation

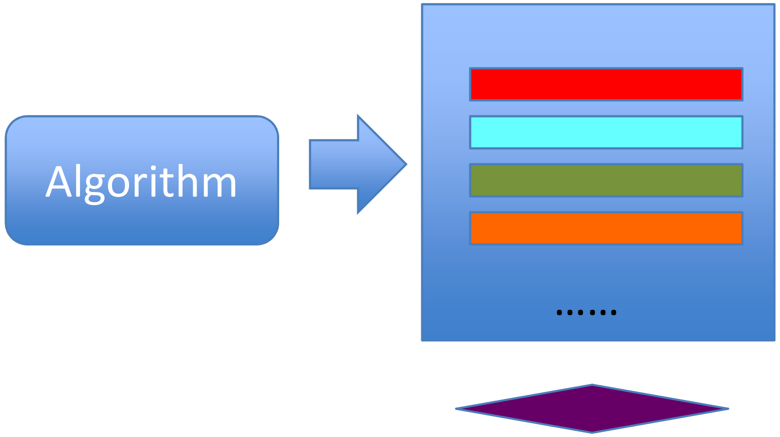
Emma Hart
Edinburgh Napier University

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

Continual Optimisation



Generalist Optimisation Algorithm

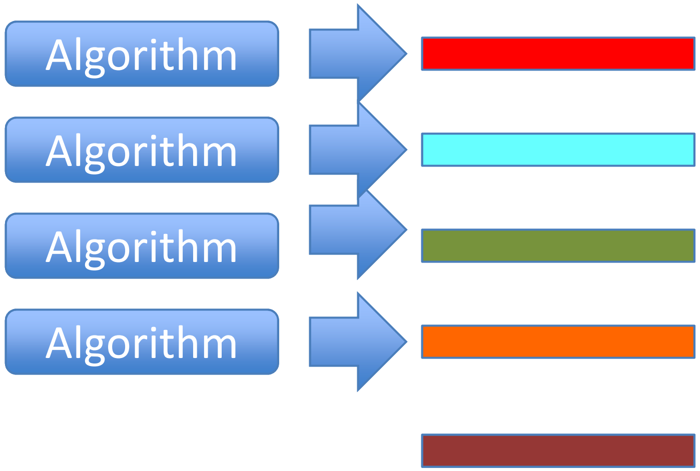


Trained to work well for many instances

Probably some compromise in performance

Tuned algorithms don't adapt well to instances with new characteristics

Specialist Optimisation Algorithms

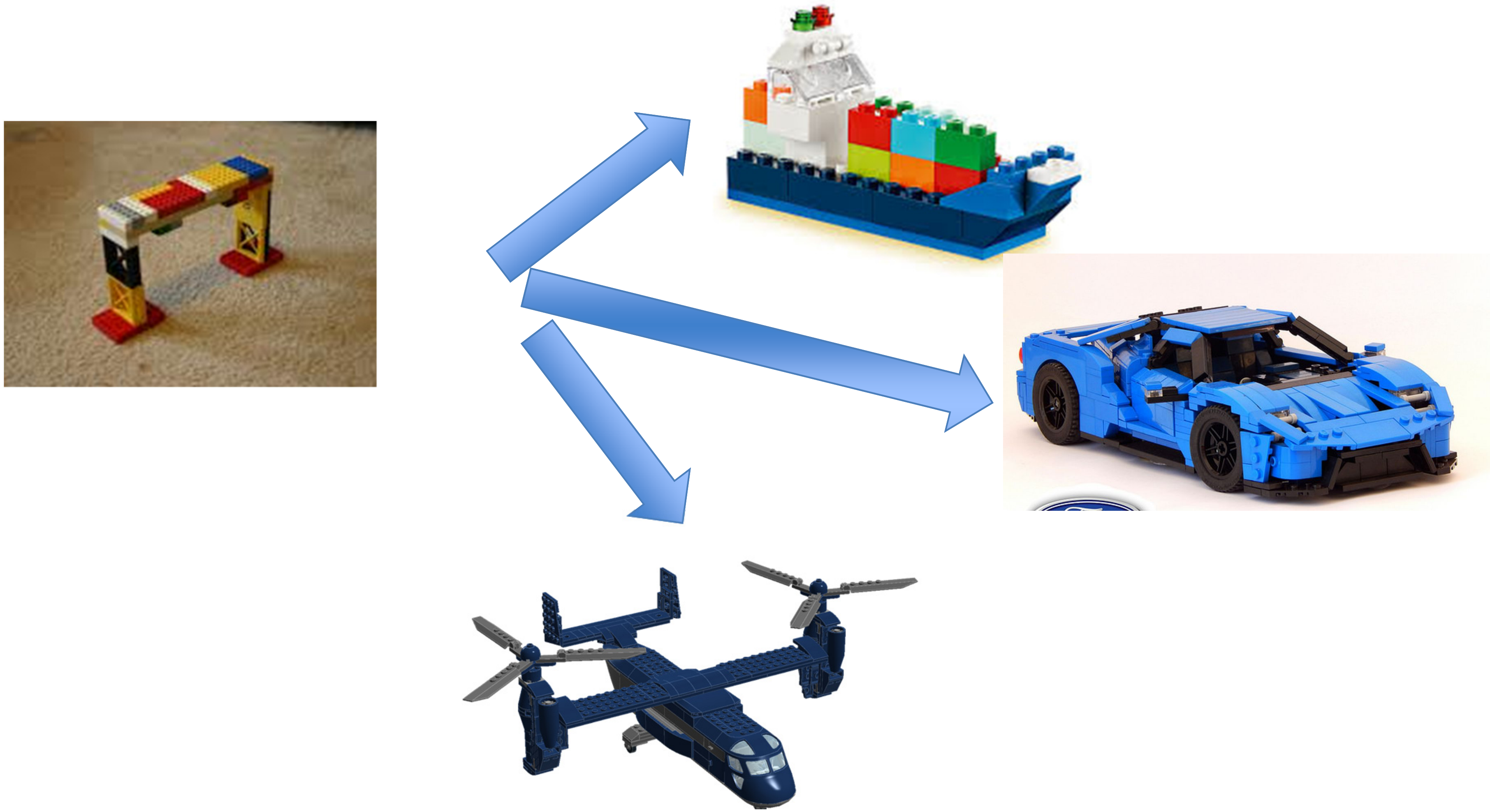


High performance on individual instances

Unable to learn from experience or exploit prior knowledge

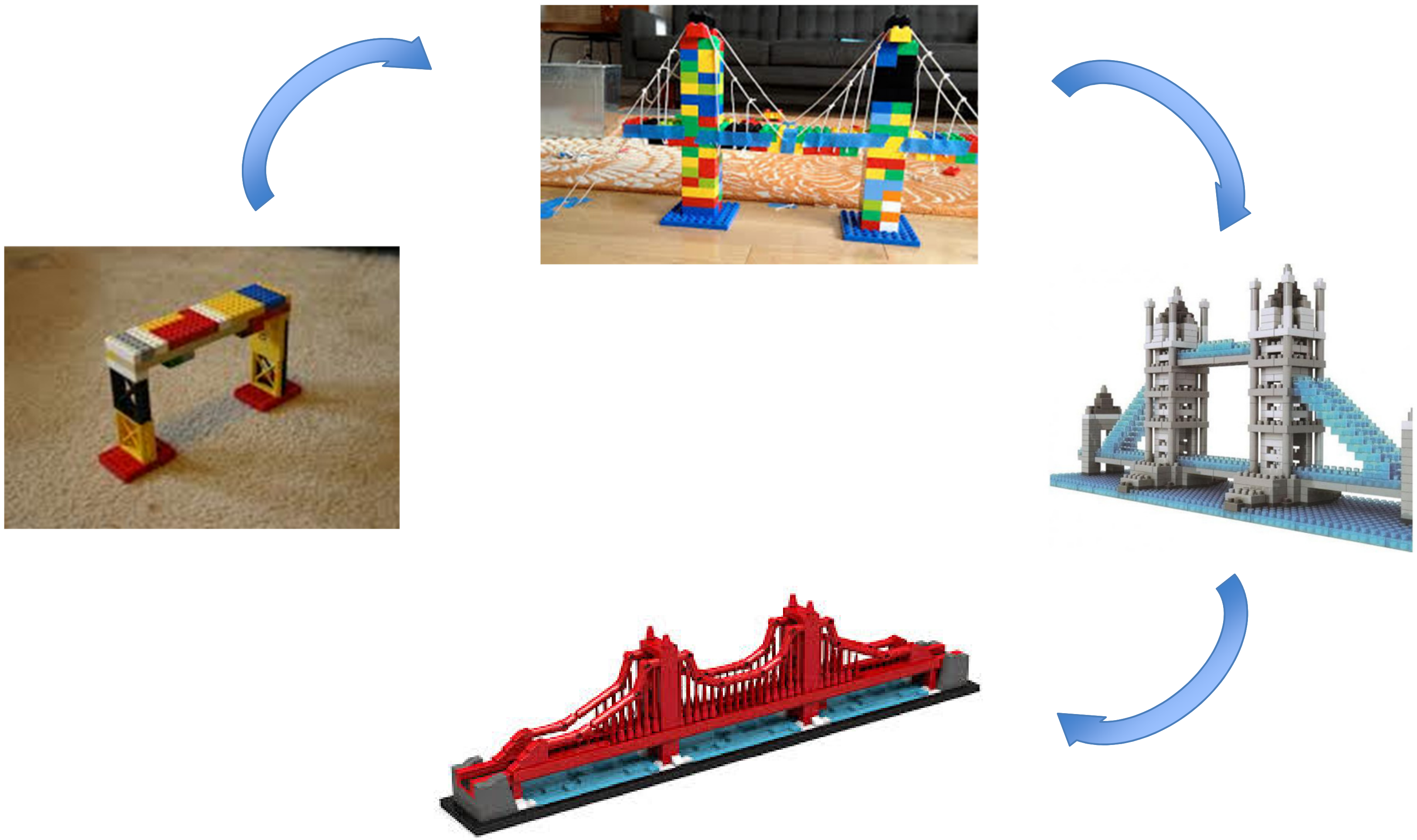
(Human) Lifelong Learning

Adapt knowledge to new situations



(Human) Lifelong Learning:

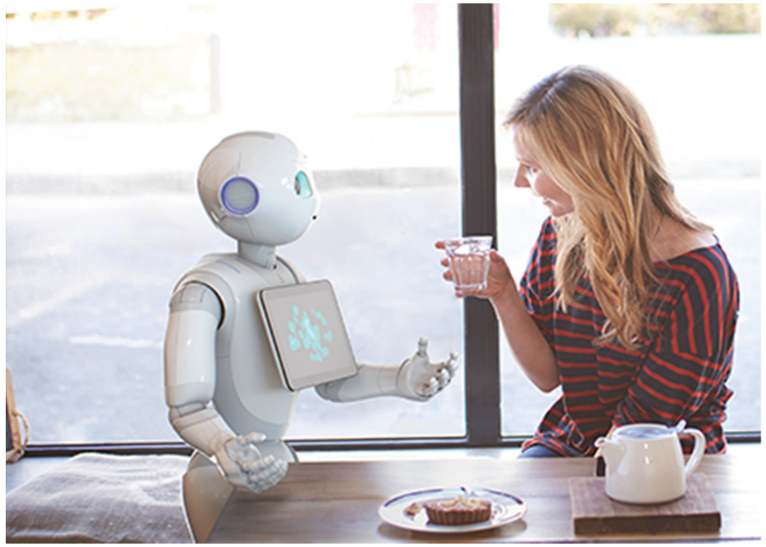
Exploit prior knowledge



(Machine) Life Long Learning

"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"

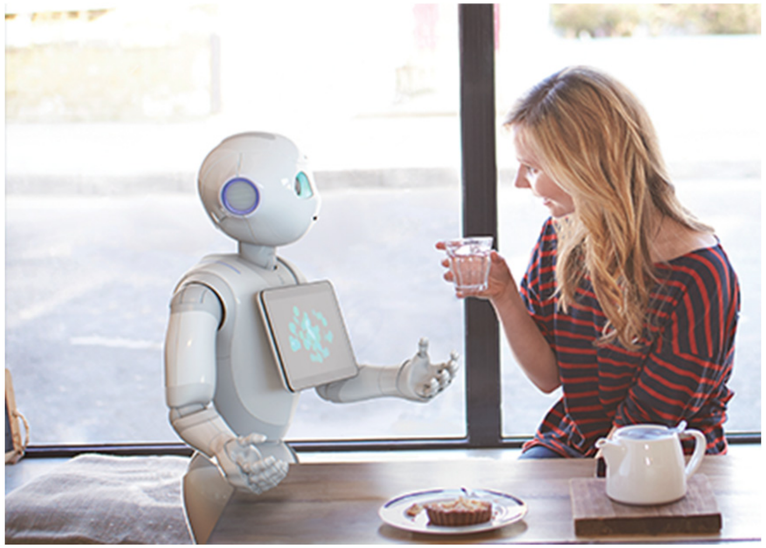
Lifelong Machine Learning Systems: Beyond Learning Algorithms. D. Silver *et al*, 2013



(Machine) Lifelong Learning

A lifelong learning system should:

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



(**Optimisation**) Lifelong Learning

A lifelong learning system should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Improve performance on past tasks over time
3. Selectively transfer prior knowledge when learning new tasks
4. 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

A lifelong learning system should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Improve performance on past tasks over time
3. Selectively transfer prior knowledge when learning new tasks
4. 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

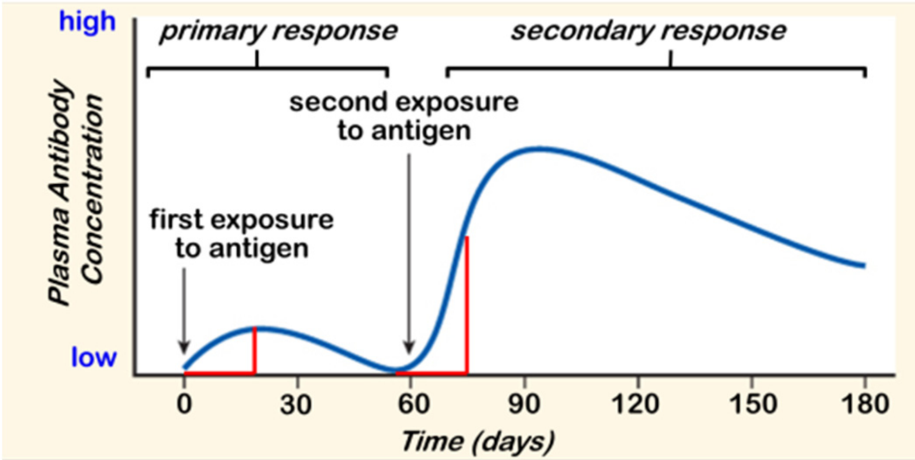
Machine Learning

A lifelong learning system should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Improve performance on past tasks over time
3. Selectively transfer prior knowledge when learning new tasks
4. 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



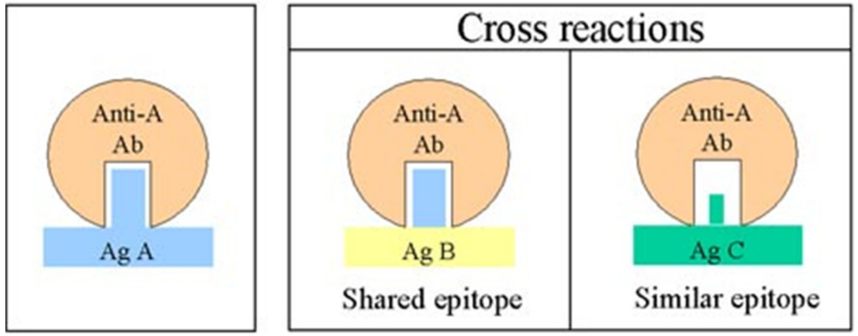
Machine Learning

A lifelong learning system should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Improve performance on past tasks over time
3. Selectively transfer prior knowledge when learning new tasks
4. 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



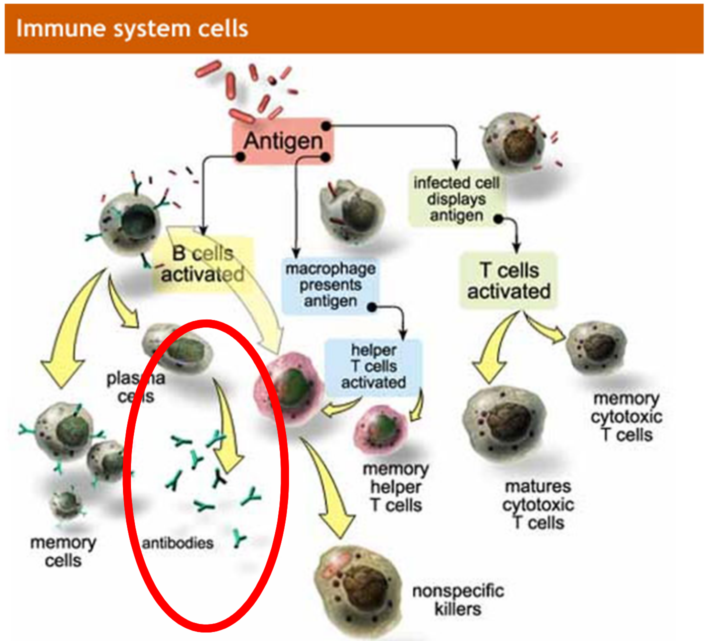
Machine Learning

A lifelong learning system should:

1. Retain and/or consolidate knowledge (long-term memory)
2. Improve performance on past tasks over time
3. Selectively transfer prior knowledge when learning new tasks
4. 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



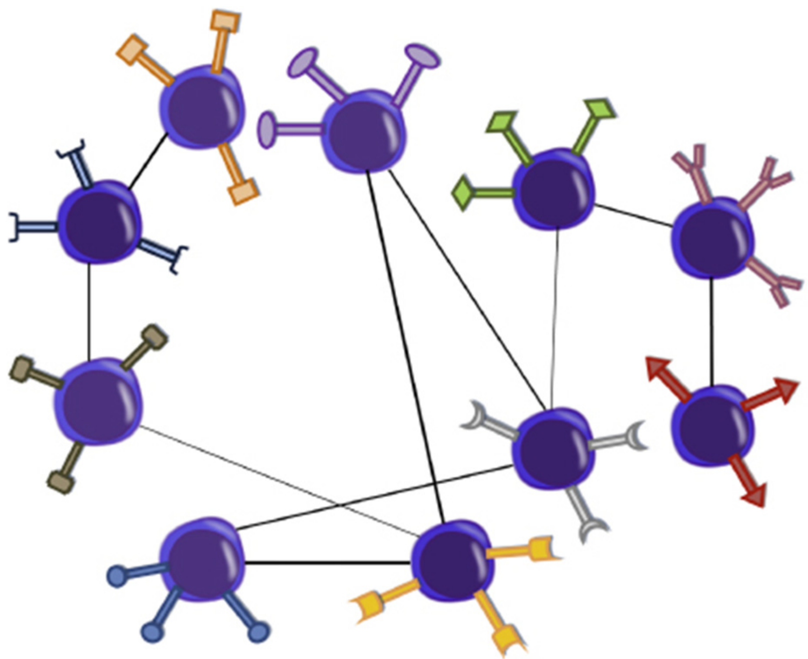
(Source: the Human Immune Response System www.uta.edu/chagas/images/immunSys.jpg)

Machine Learning

A lifelong learning system should:

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

Natural Immune System



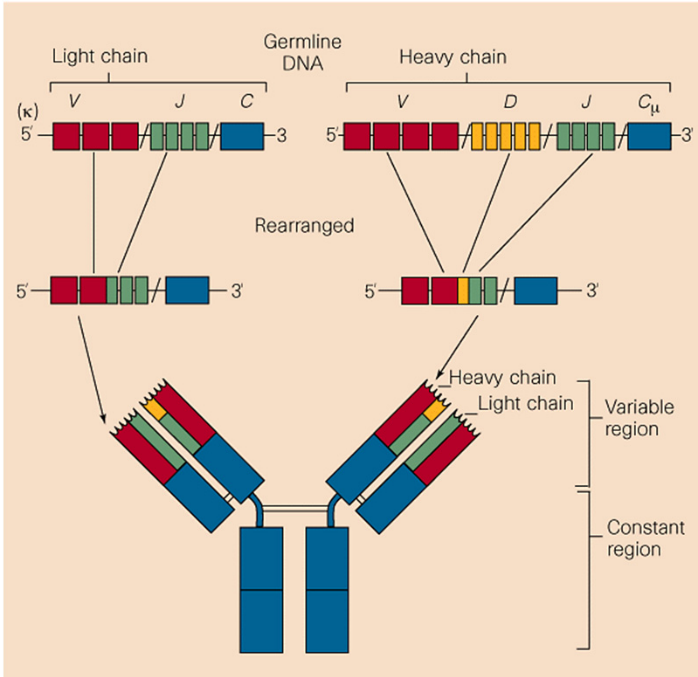
Machine Learning

A lifelong learning system should:

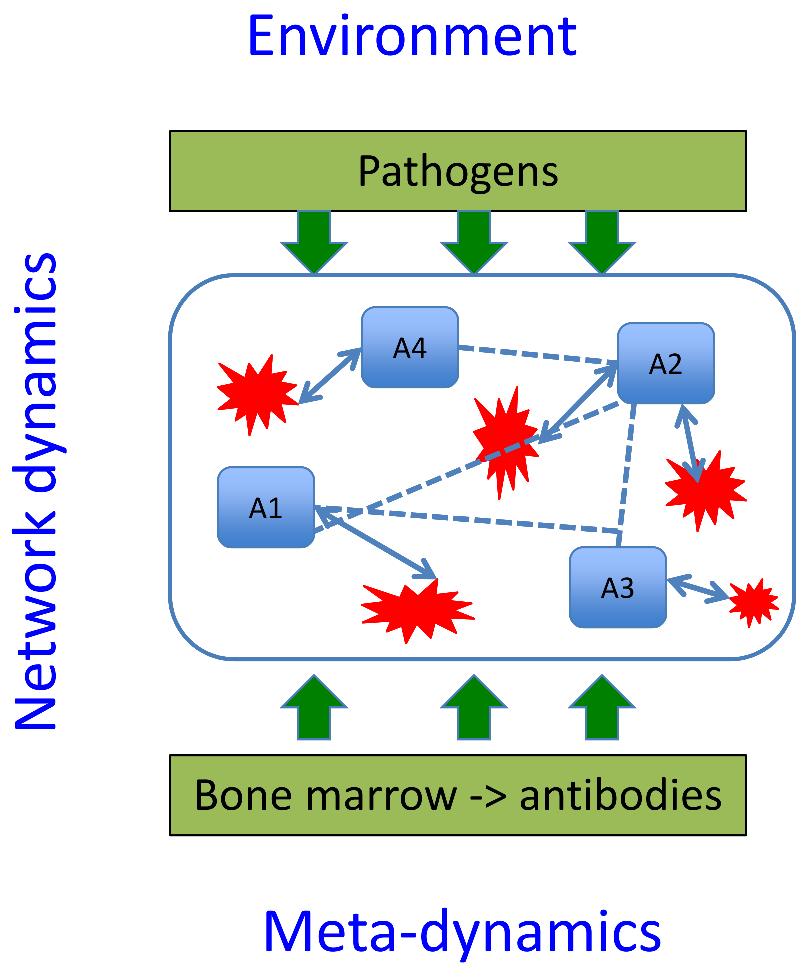
1. Retain and/or consolidate knowledge (long-term memory)
2. Improve performance on past tasks over time
3. Selectively transfer prior knowledge when learning new tasks
4. Adopt a systems approach that ensures effective and efficient interaction of elements of the system
5. Generate new knowledge

Natural Immune System

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



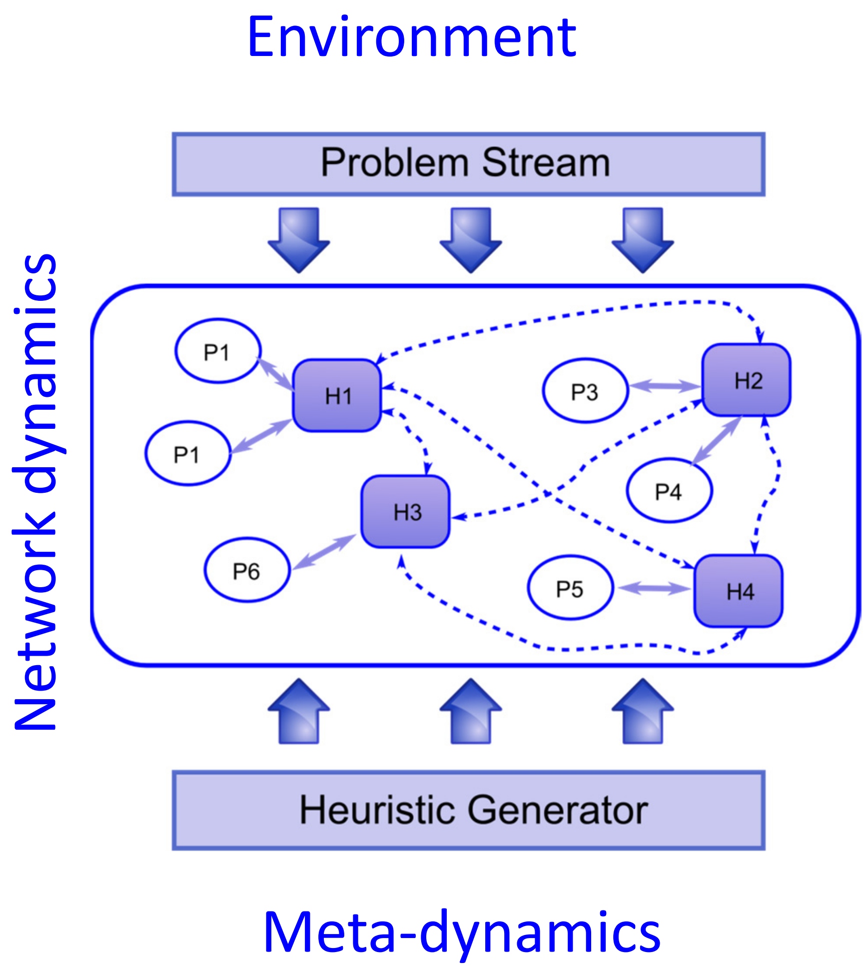
Immune System



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

Optimisation Systems

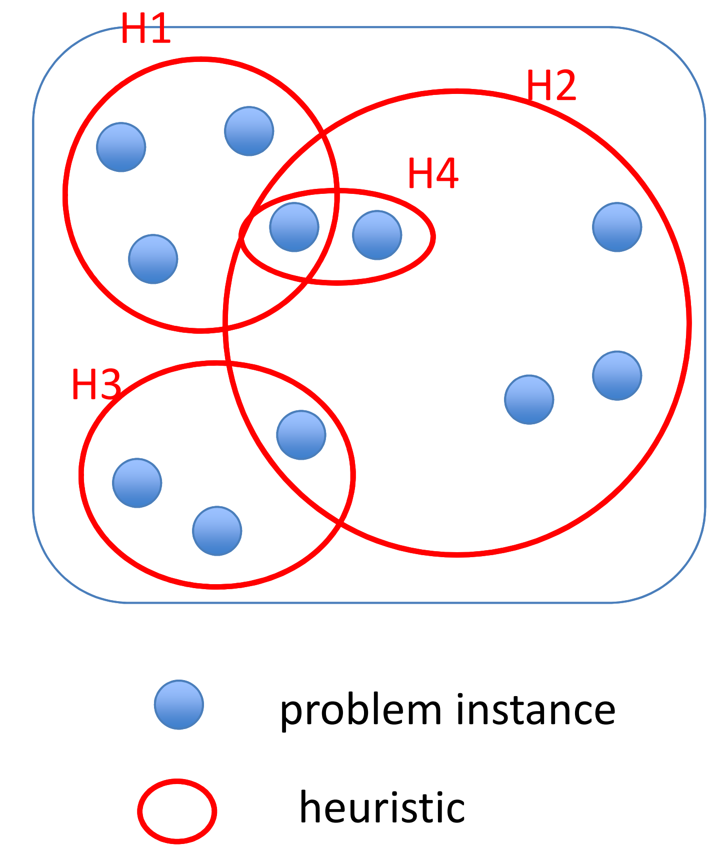


Computational Properties

- *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

Conceptual Overview

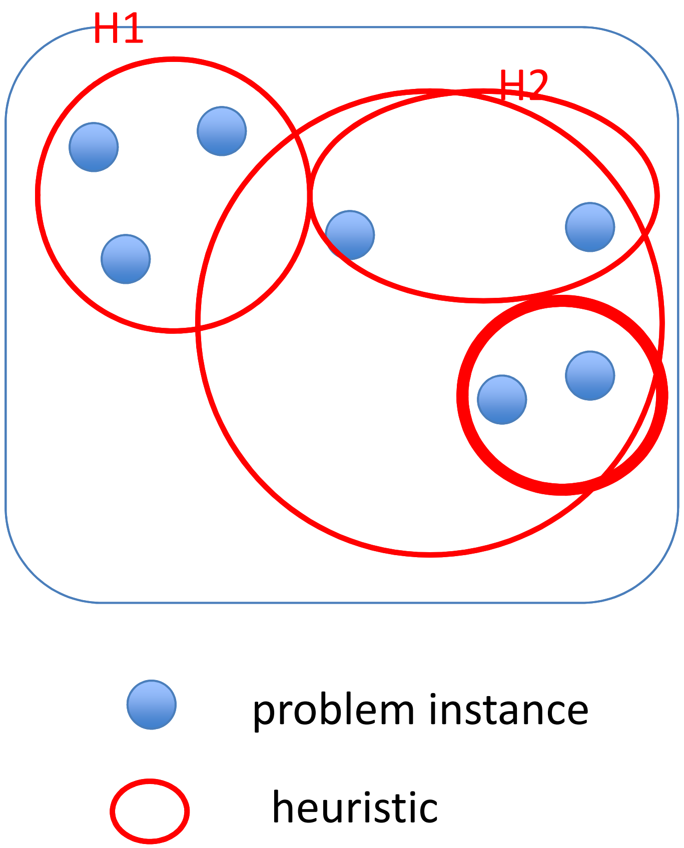
2d Representation of problem space



- 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 'informative'
- 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

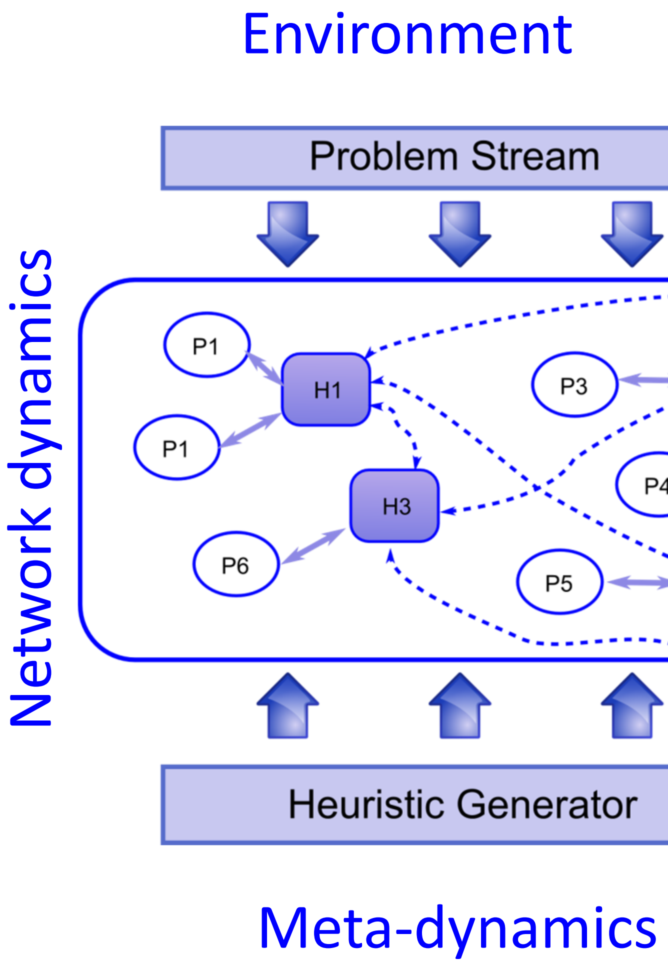
Conceptual Overview

2d Representation of problem space



- 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

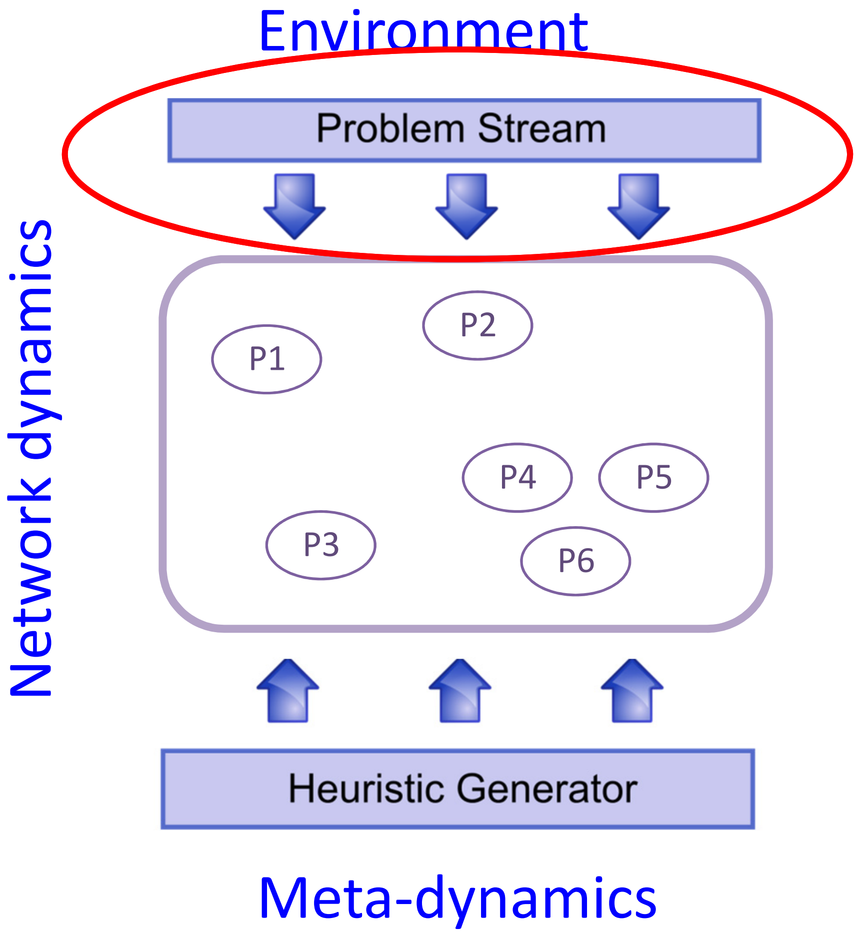


1 Problem Stream

2 Heuristic Generator

3 Network of heuristics & problems

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

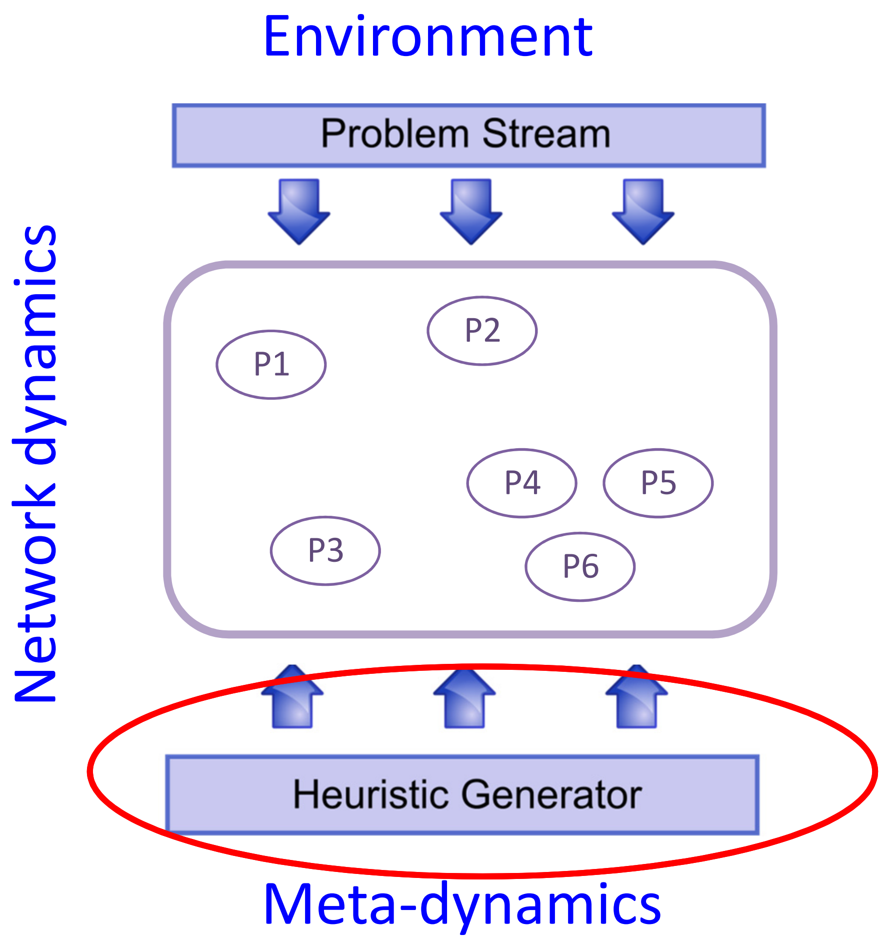


① Problem Stream

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

- Single instance
- Multiple instances
- Frequent/infrequent

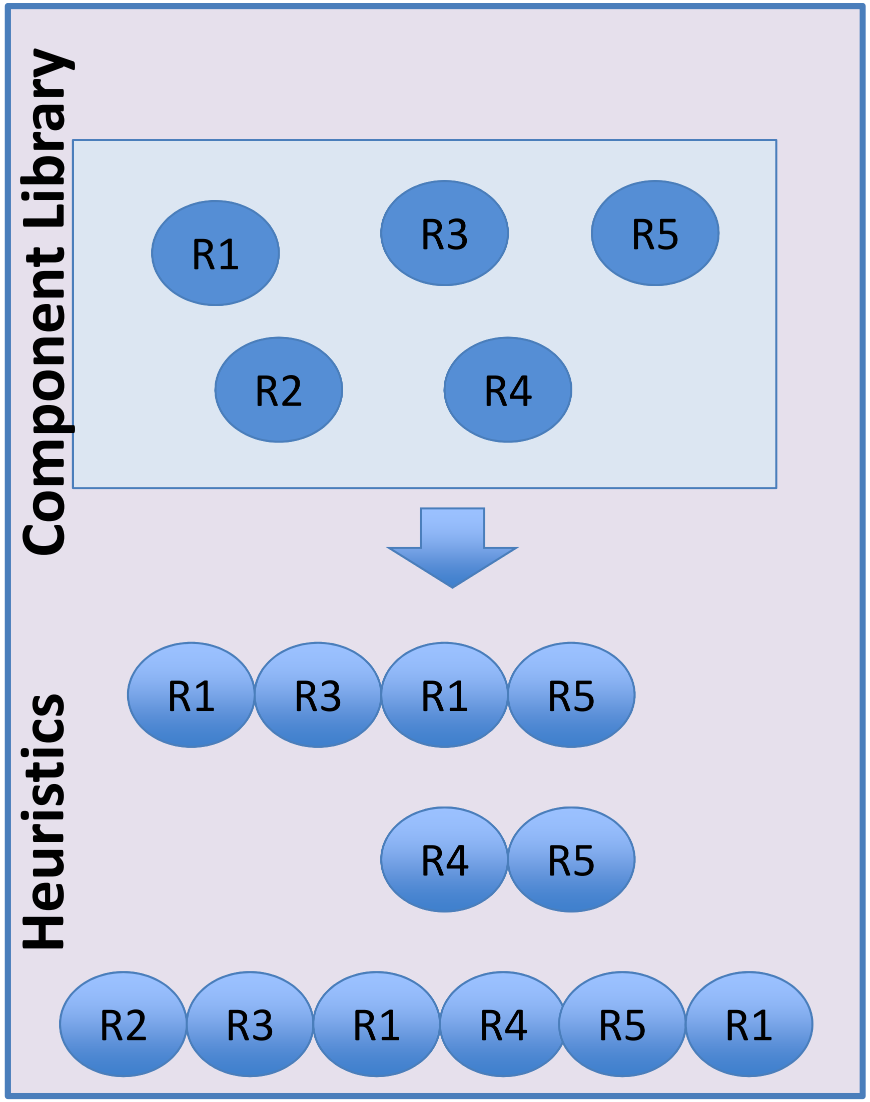
NELLI – **Ne**twork for **L**ife**L**ong **L**ear**ni**ng



② Heuristic Generator

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

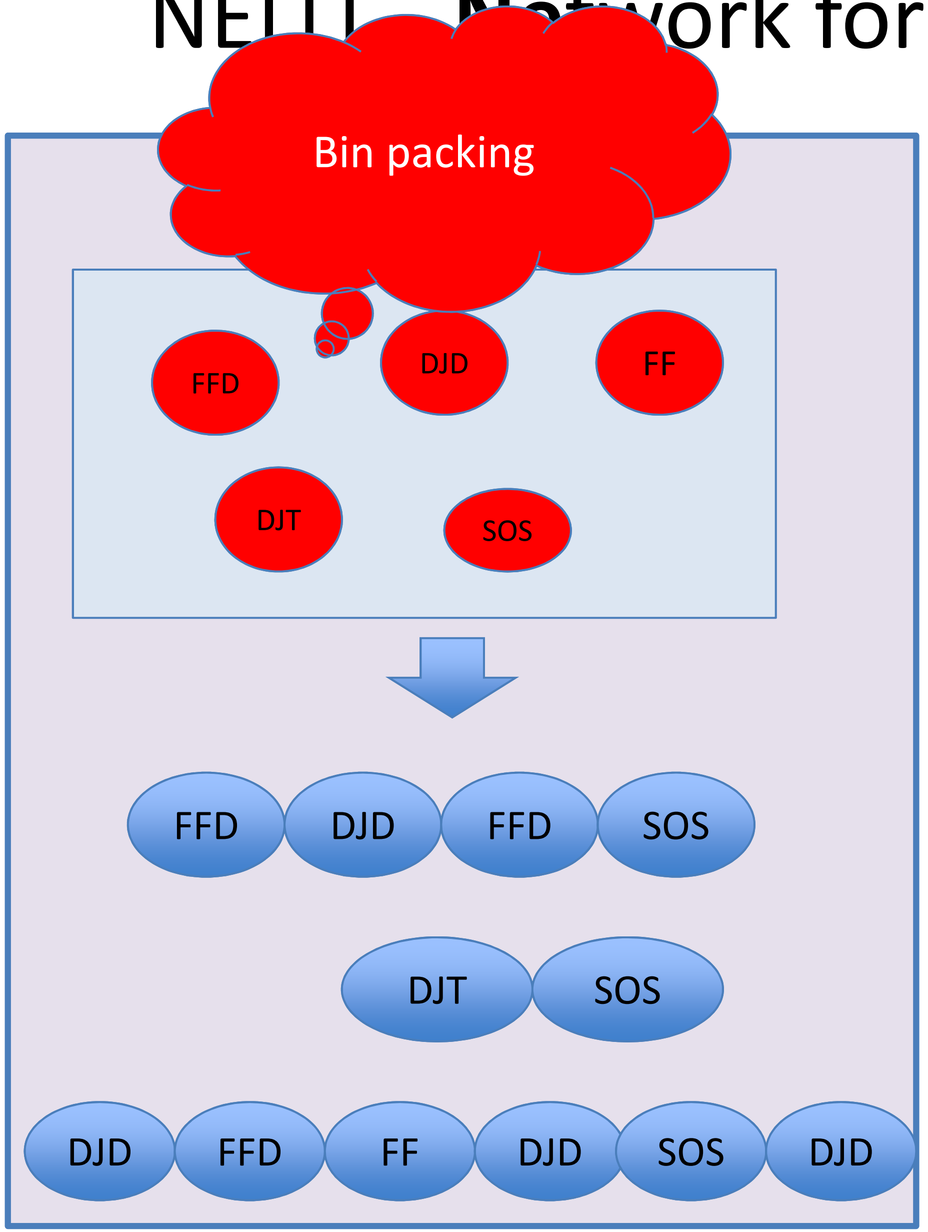
NELLI – **Ne**twork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

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

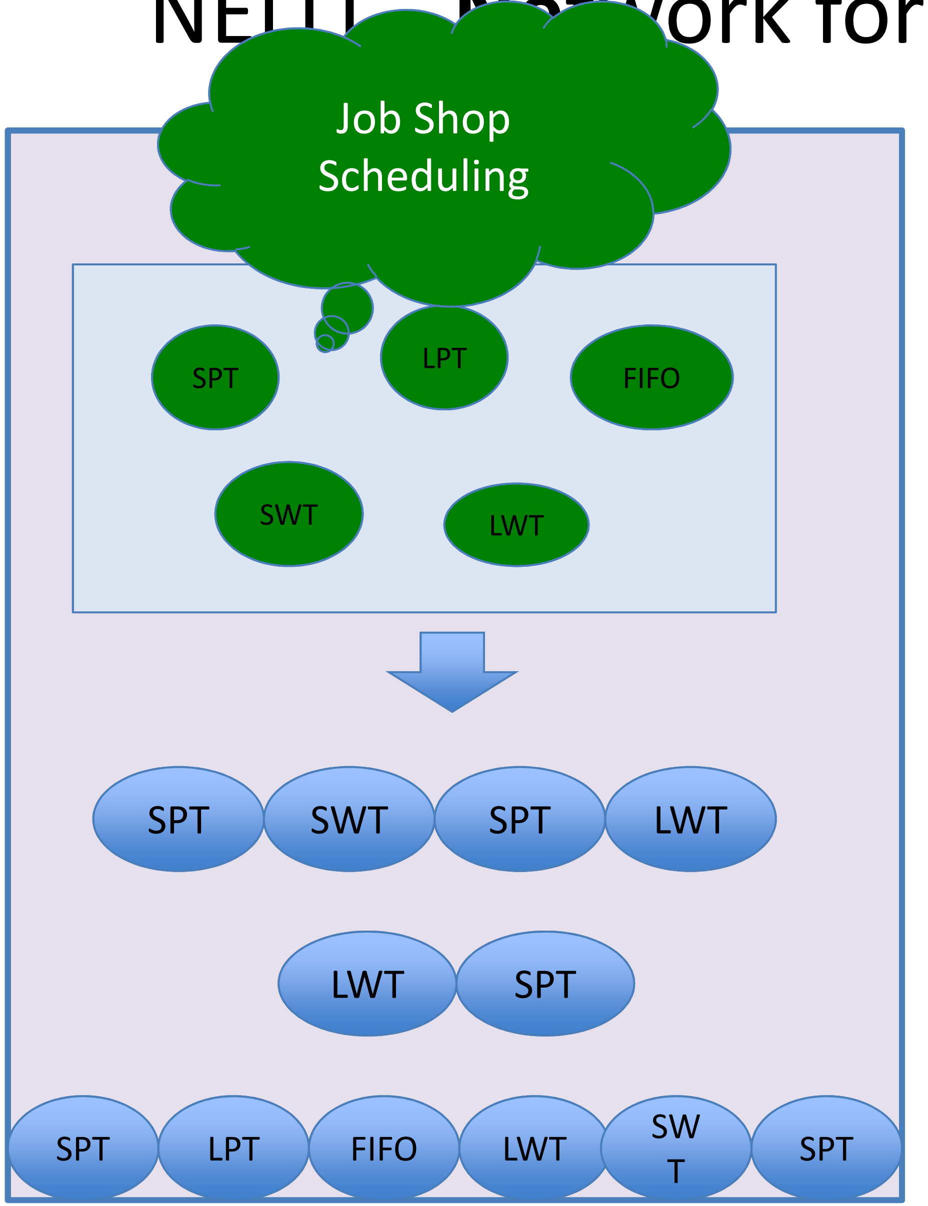
NELLI – **N**etwork for **L**ifeLong **L**earn**i**ng



② Heuristic Generator

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

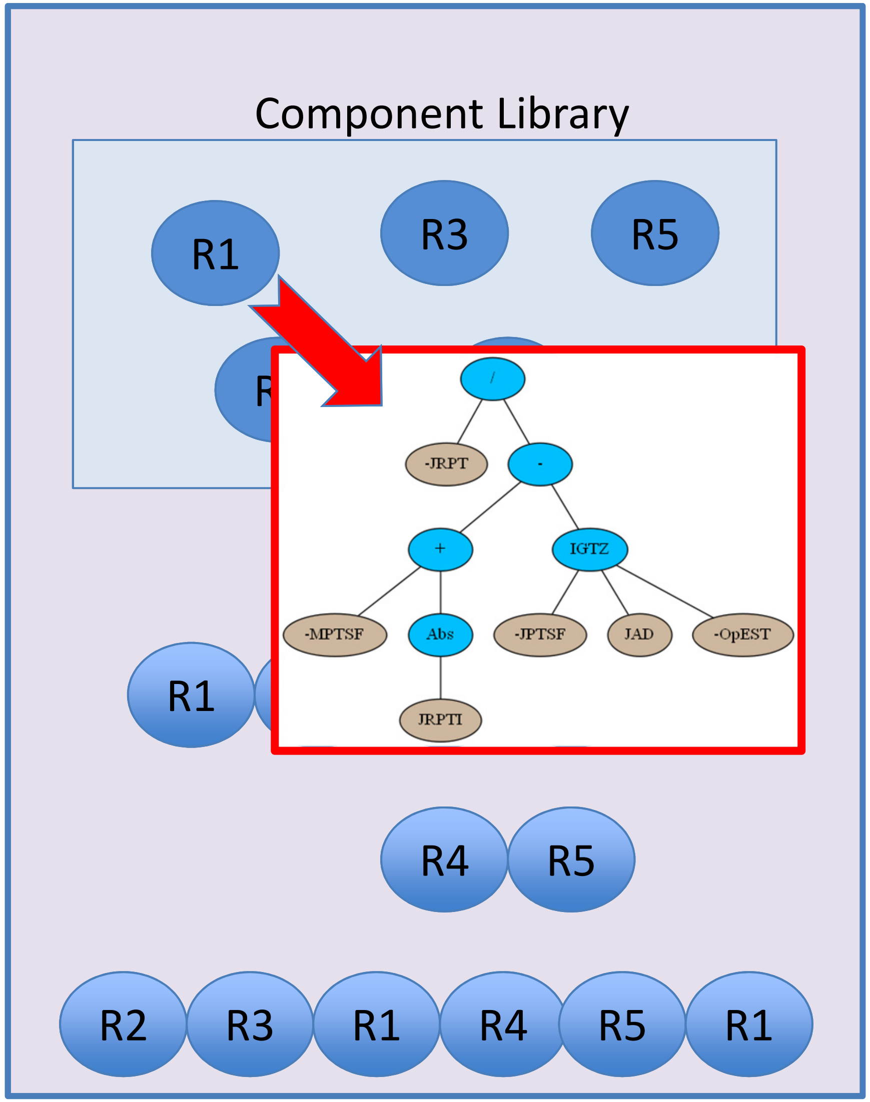
NELLI – **N**etwork for **L**ife**L**ong **L**earn**i**ng



② Heuristic Generator

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

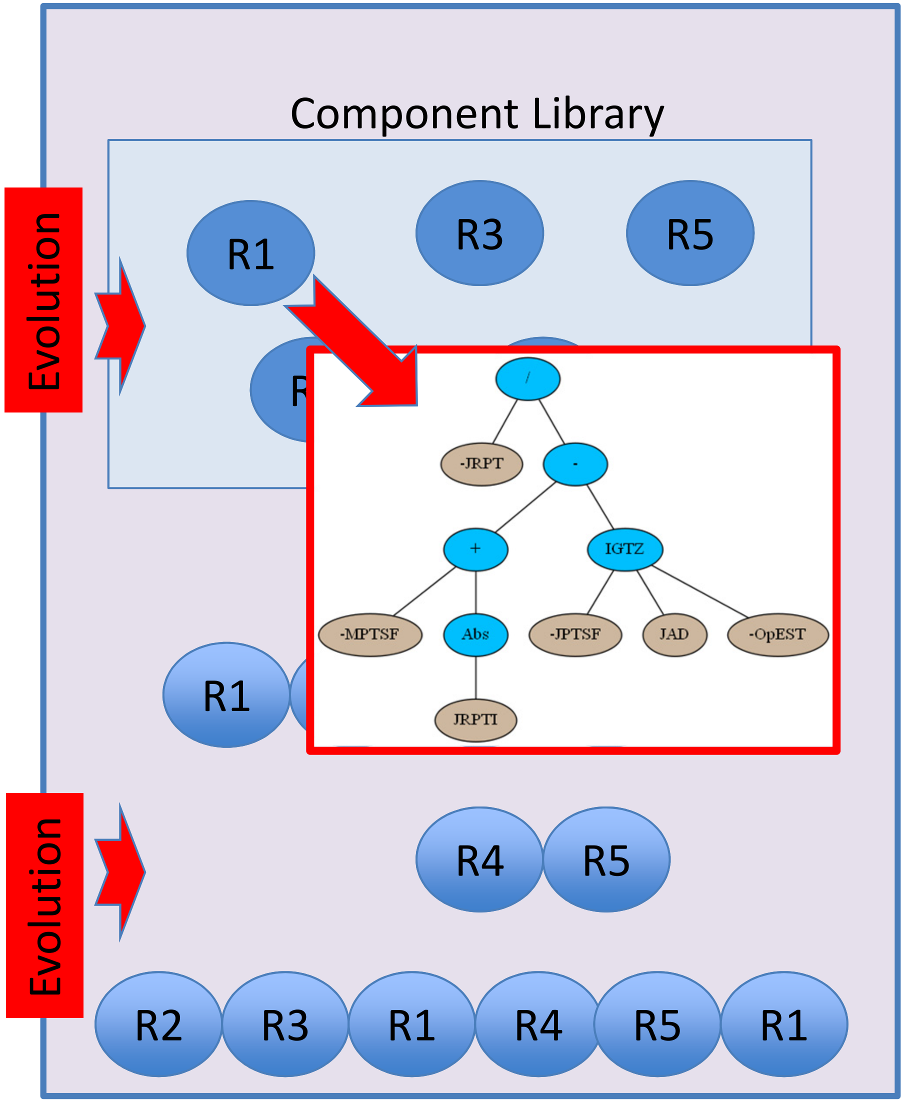
NELLI – **Ne**twork for **L**ife**L**ong **L**earn**i**ng



② Heuristic Generator

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

NELLI – **Ne**twork for **L**ife**L**ong **L**earn**i**ng

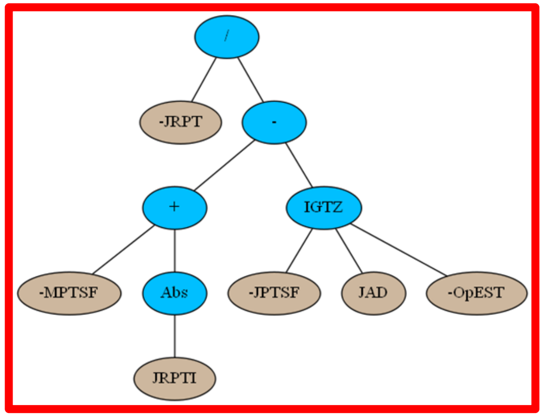


② Heuristic Generator

- 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**ife**L**ong **L**earn**i**ng

Evolution

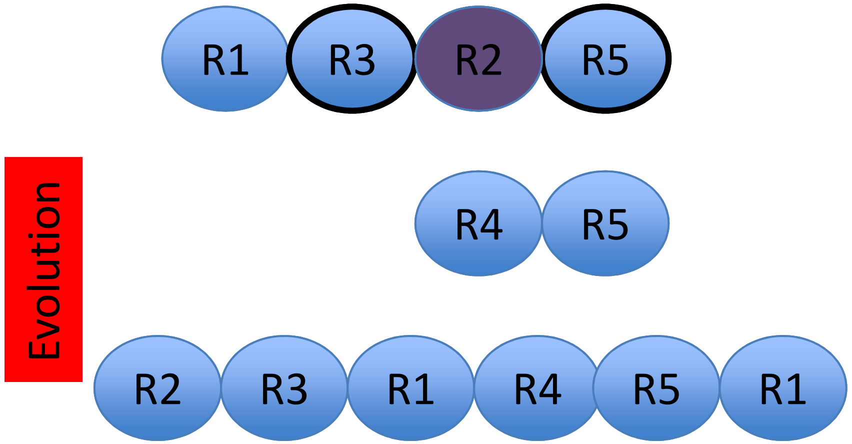


- Mutate terminal nodes
- Mutate function nodes
- Remove subtree
- Swap subtrees

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

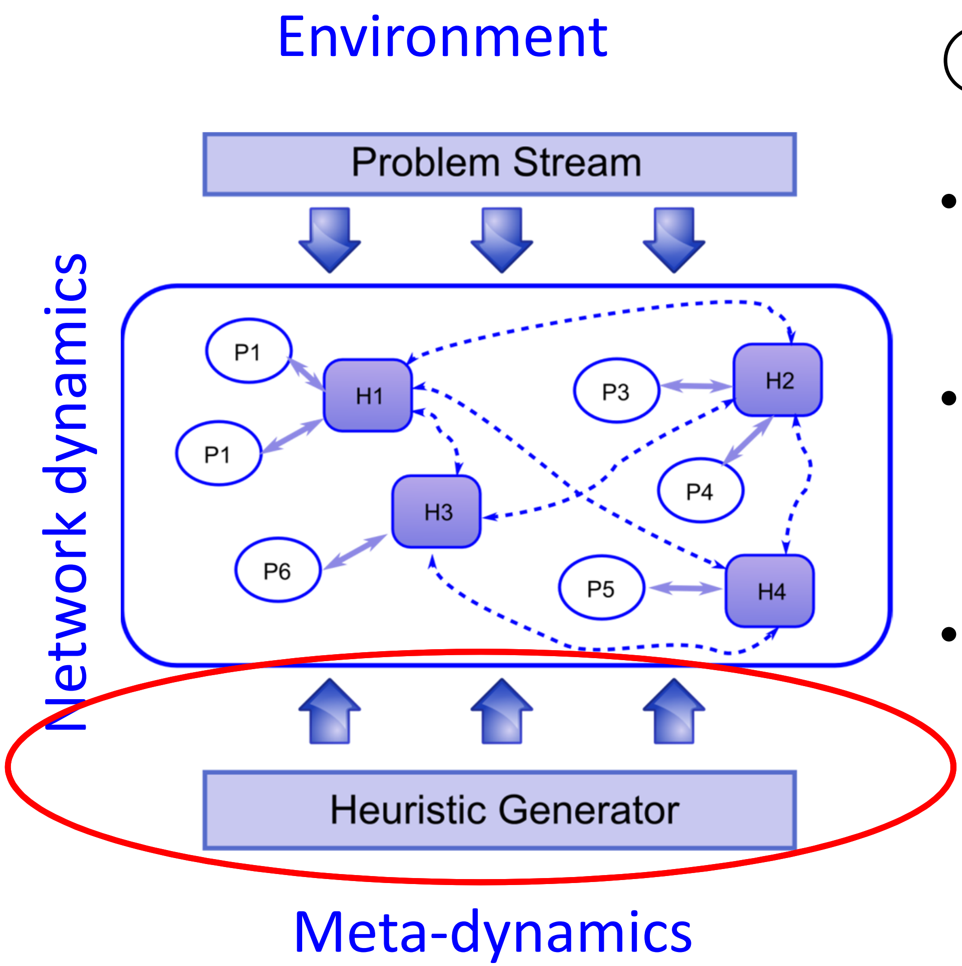


Swap components
Change components
Remove/insert components
Concatenate heuristics

② Heuristic Generator

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

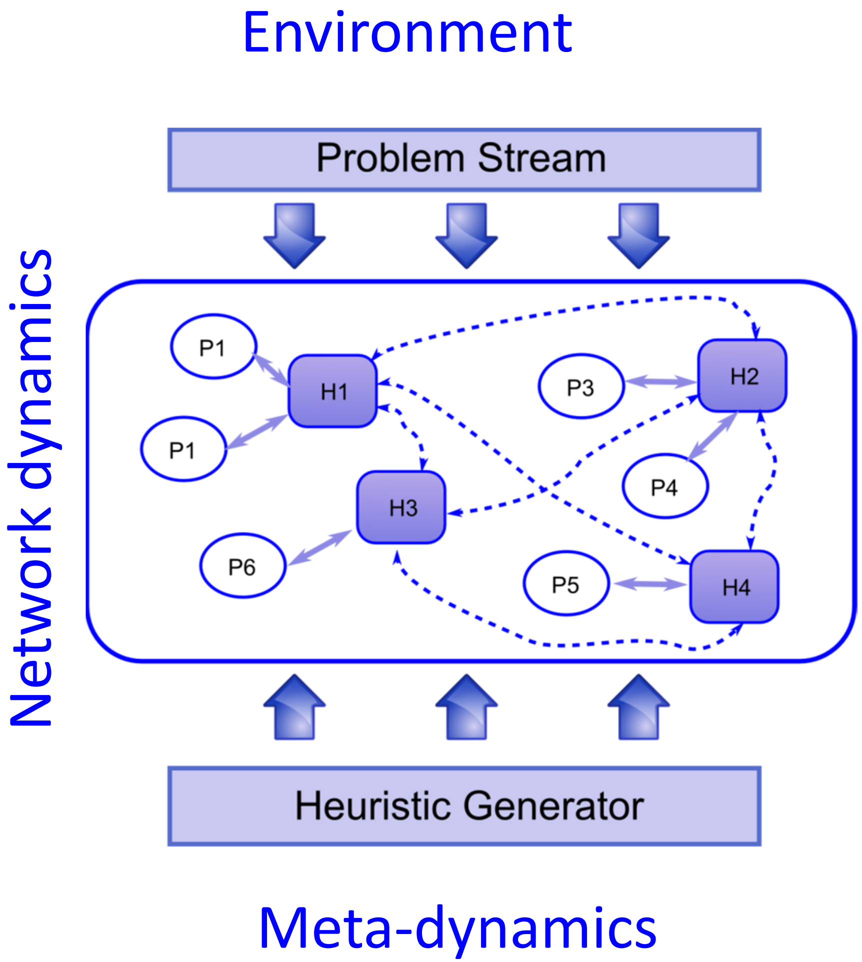
NELLI – **Ne**twork for **L**ife**L**ong **L**earn**i**ng



③ Network

- *Continuous generation of heuristics*
- *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 – **Ne**twork for **L**ifeLong **L**earn**i**ng

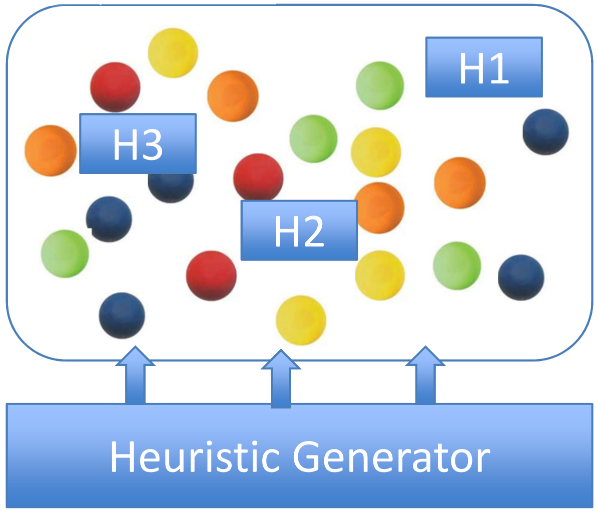


③ Network

- Stimulation affects **survival**
  - retention/removal from network
  - memory
- Stimulation affects the probability of being selected for cloning/ mutation during evolution

Basic optimiser

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



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

Basic optimiser

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

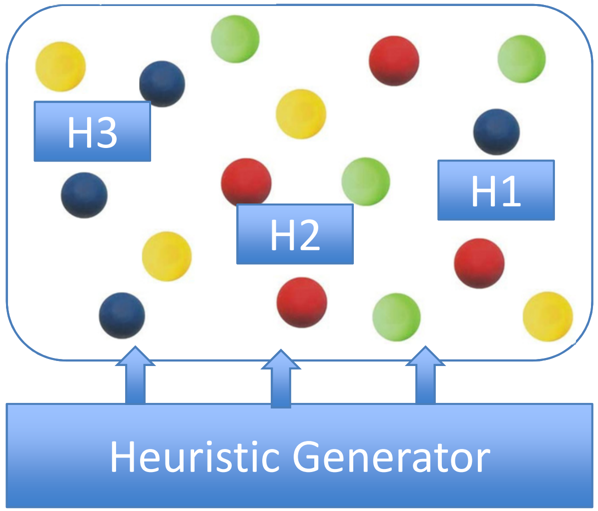
	Problems solved	Extra bins
FFD	788	2142
DJD	716	2409
DJT	863	881
ADJD	686	1352
NELLI	**1126**	**308**

Heuristics Retained	7
Problems Retained	36

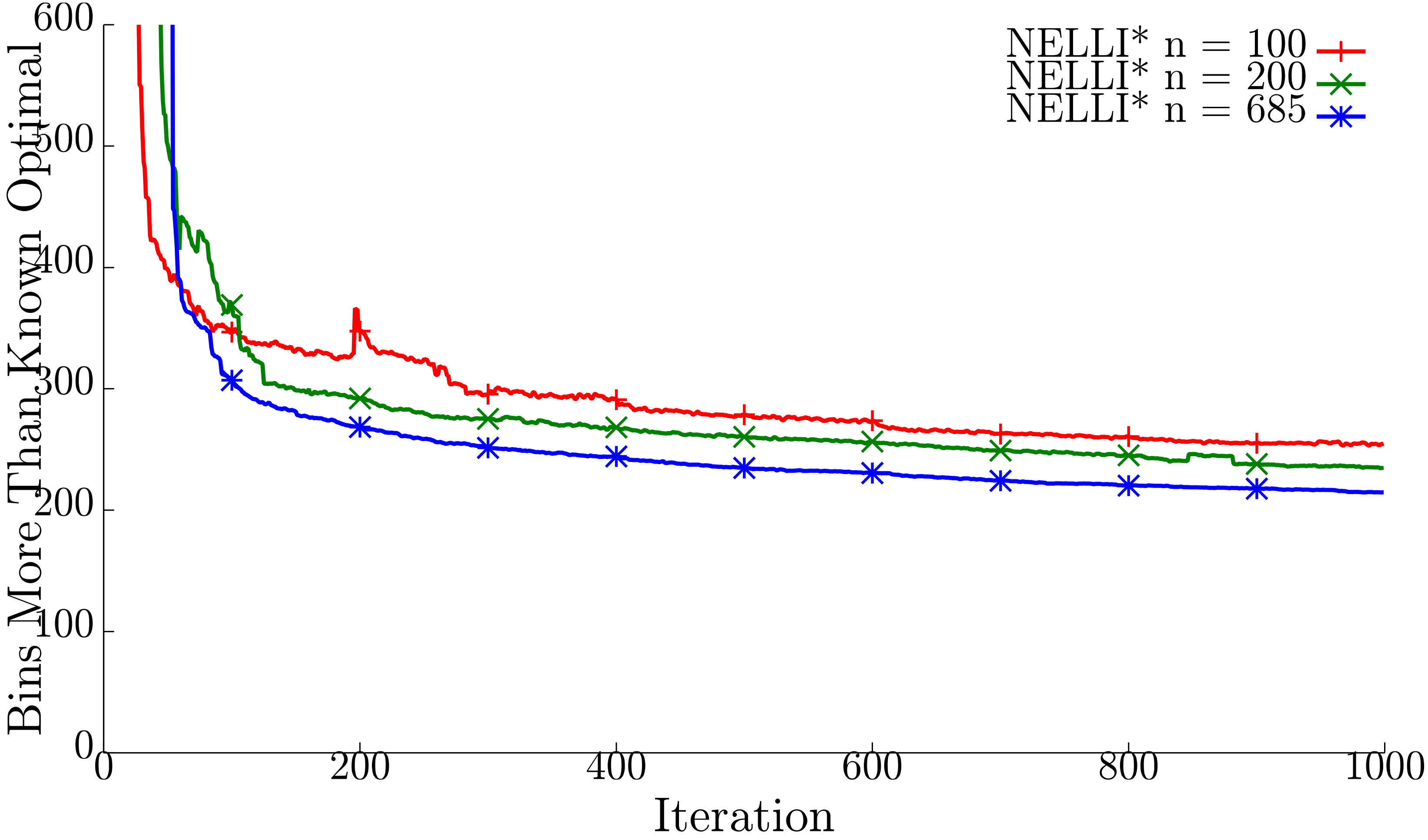
Sim & Hart: A Lifelong Learning Hyper-heuristic Method for Bin Packing, Evolutionary Computation, Spring 2015, Vol. 23, No. 1, Pages 37-67

Evaluating Continual Learning

- "Bag" of 1370 instances
- Inject *n* randomly chosen instances
- Every 200 iterations, replace instances with a new set of *n* instances:
  - Randomly drawn
  - From specific "groups"
- Monitor performance on complete "bag"

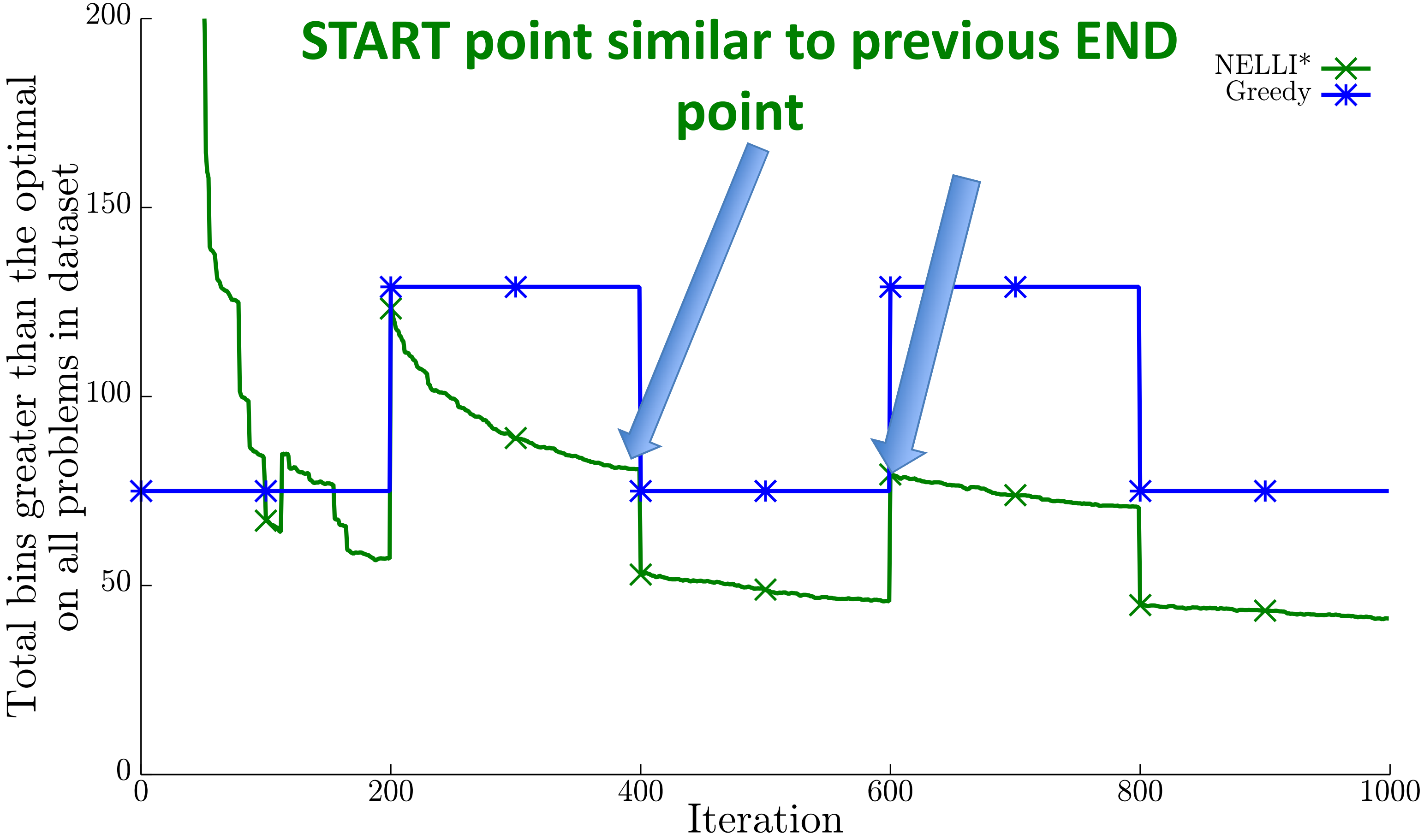


Learning from experience in a changing environment



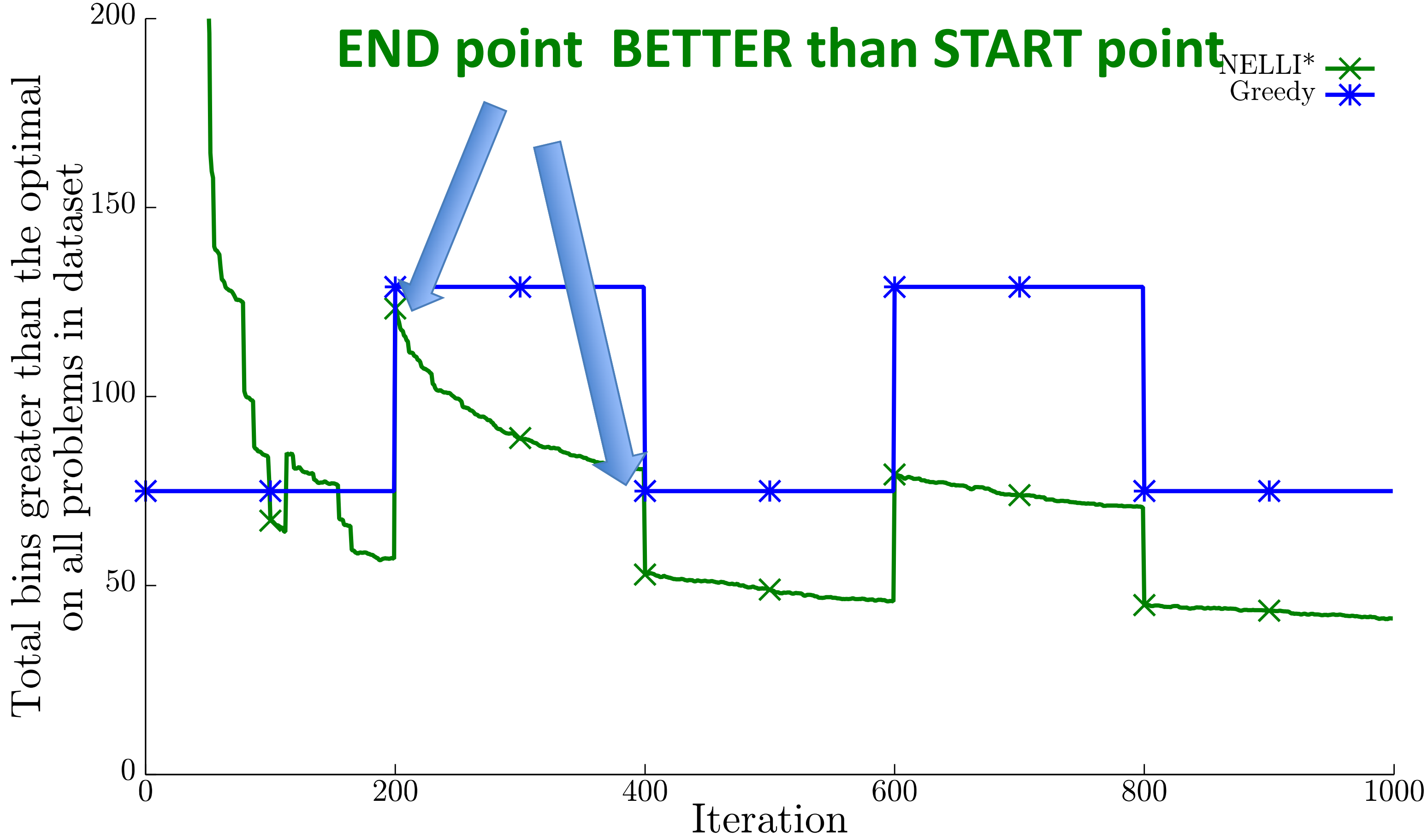
Retaining **Memory**

Alternate between two different datasets every 200 iterations

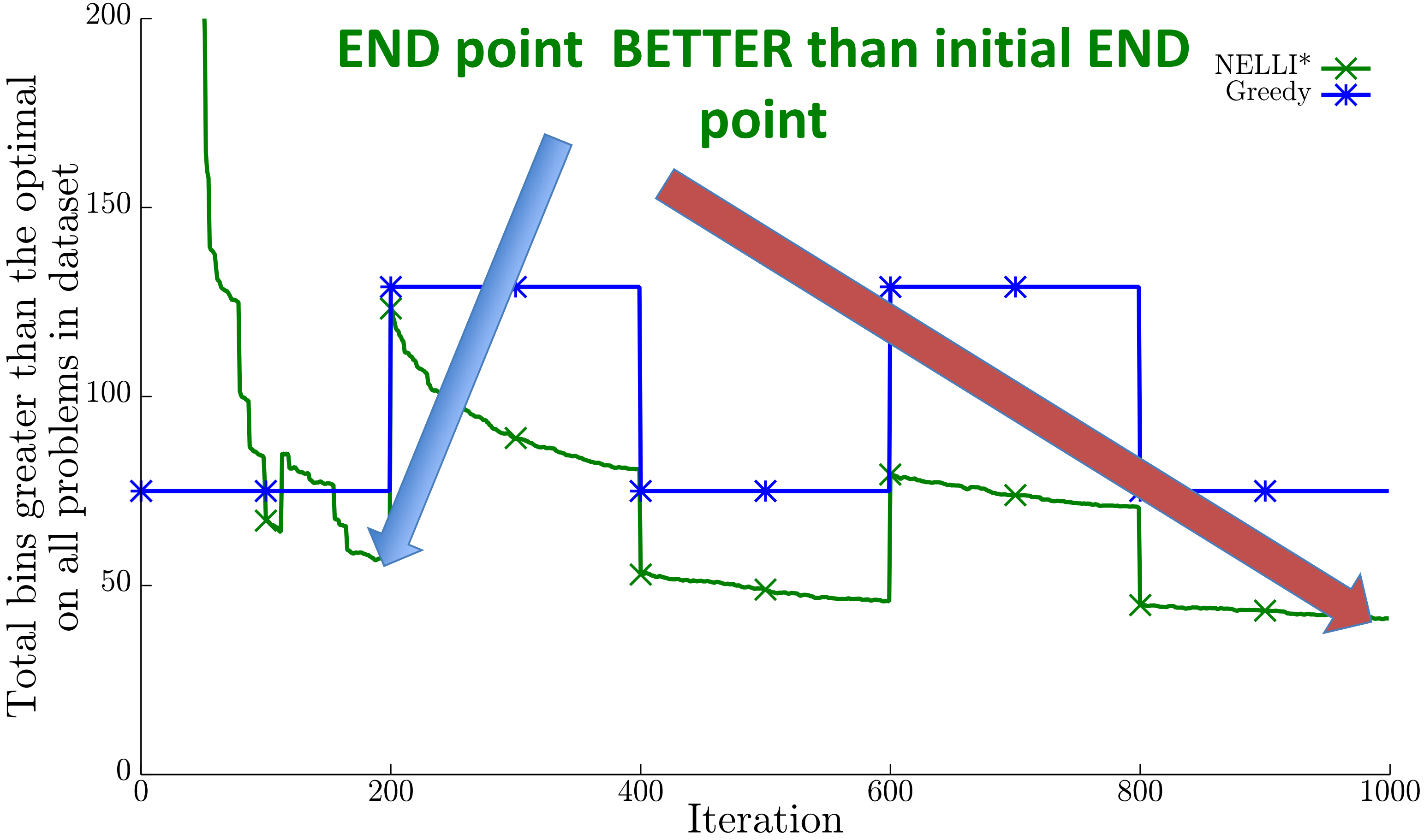


Learning over an **epoch**

Alternate between two different datasets every 200 iterations

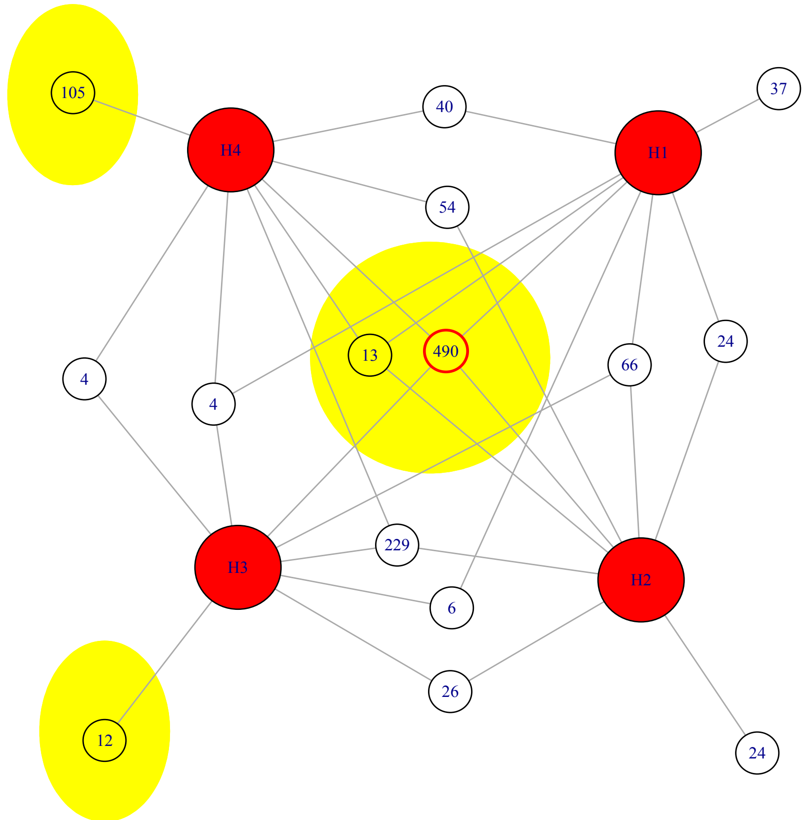


Learning over a **lifetime**



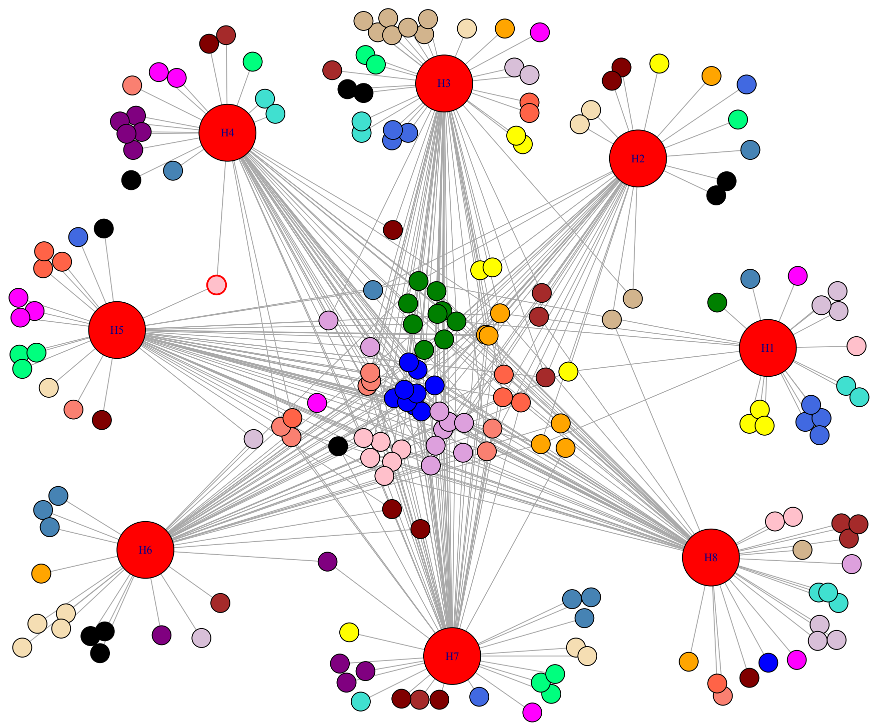
Insights into instance space

- Record how many heuristics 'win' each instance
- "Interesting" regions of the instance space represented by instances won by only one heuristic
  - Define distinct/unique regions
- Insights into relative difficulty of instances
  - Easy: many heuristics solve
  - Hard: few heuristics solve



Heuristic Perspective

- Where in the instance space does a particular heuristic perform well ?
- Algorithm selection:
  - Find a set of common characteristics that define "similar" instances
  - [issue] **Instances that are "similar" from algorithm perspective do not share features which are "human-intuitive"**

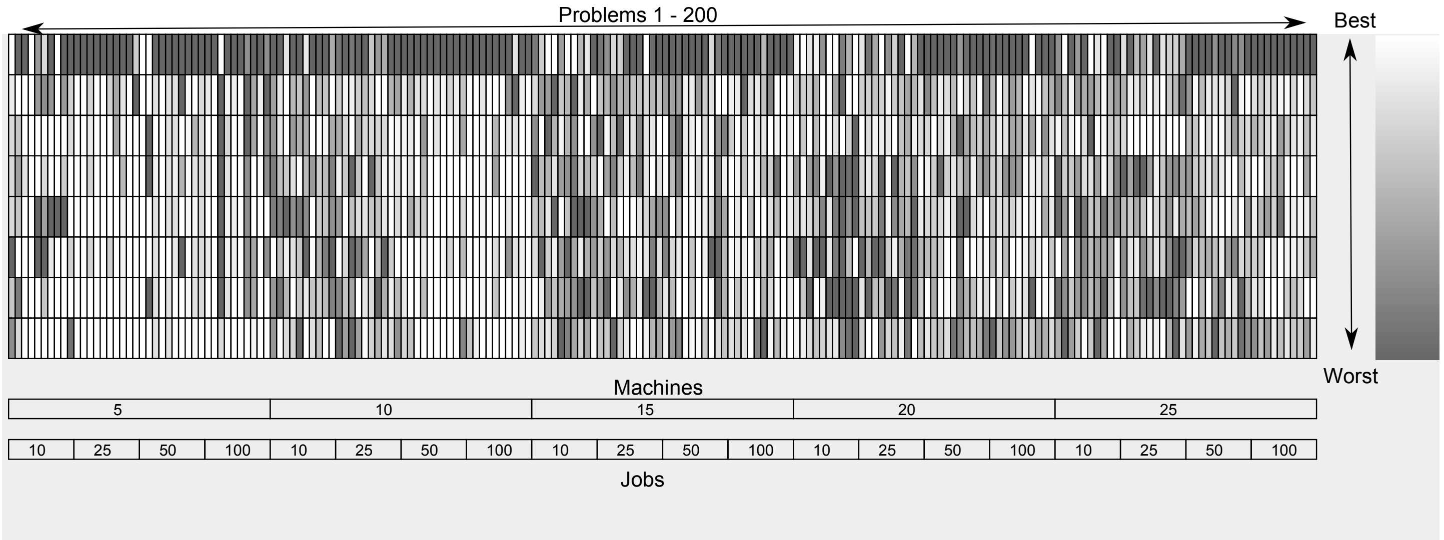


Each colour represents a class of JSSP instances defined by (j,m,r) .. 20 classes

Hart & Sim, A . A Hyper-Heuristic Ensemble Method for Static Job-shop Scheduling. Evolutionary Computation, 2016

Insights into heuristic performance

- Define a **signature** (*barcode*) for each heuristic based on its relative performance on an instance
  - Visual clues to relative quality & diversity of heuristics
- Could be used to directly compare diversity of heuristics using a distance metric



Results from an evolved ensemble of **8 heuristics** applied to 200 unseen JSSP instances

Summary

- A "lifelong" approach to developing optimisation algorithms is required
  - Deal with a continual stream of instances
  - Improve over time
  - Learn from experience
- 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 other heuristic methods*

Illustration by Michael Witte

Key Message

Optimisation systems should continuously learn

- *Exploit previous knowledge*
- *Adapt to changing instance characteristics*

This is a paradigm shift from current optimisation approaches

- Likely to be scope for hybridising with machine-learning approaches
- Don't need a single "killer-optimiser" – switch to ensemble approach

"It's not the strongest of the species that survives, it's the one that's most adaptable to change"……Darwin

On Constructing Ensembles for Optimisation, Hart,E. and Sim,K. ECJ 2017

THANK YOU!

EPSRC
Engineering and Physical Sciences
Research Council

Acknowledgements

Dr Kevin Sim

EPSRC **EP/J021628/1**