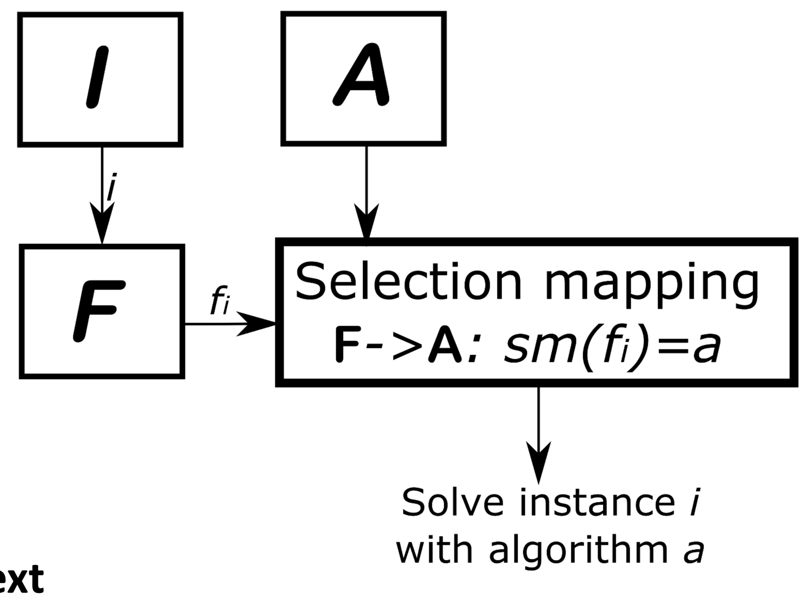
Active Automatic Algorithm Selection Using Multi-armed Bandits

Hans Degroote and Patrick De Causmaecker
KU Leuven Department of Computer Science
CODeS & iMinds-ITEC

What is automatic algorithm selection?



Context

- A problem is described by an **instance distribution *I***
- Instances are characterized by a **set of features *F*** describing problem characteristics (e.g. size)
- A **set of algorithms *A*** exists, none of which dominates all others

Goal

- Solve each instance with its best algorithm

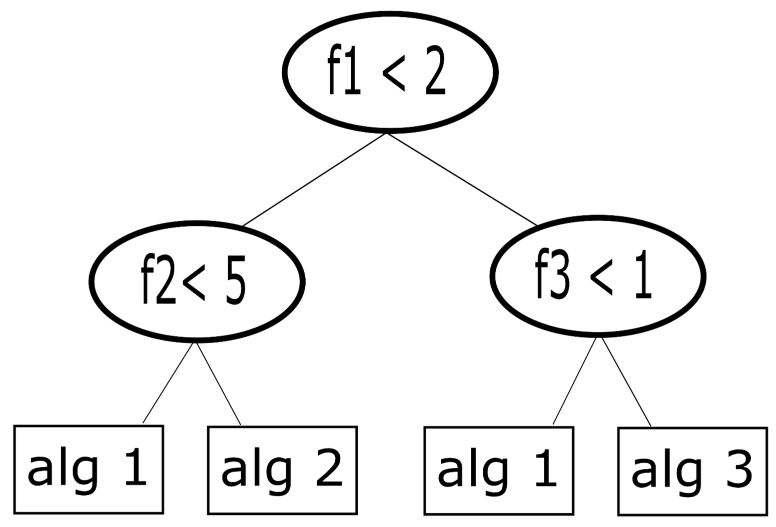
Method

- Create a selection mapping assigning each instance to an algorithm, based on the features

Current offline solution approach

Algorithm selection is modelled as a classification problem:

- Divide the instance space into subsets of similar instances (based on the features)
- Learn for each subset the best algorithm

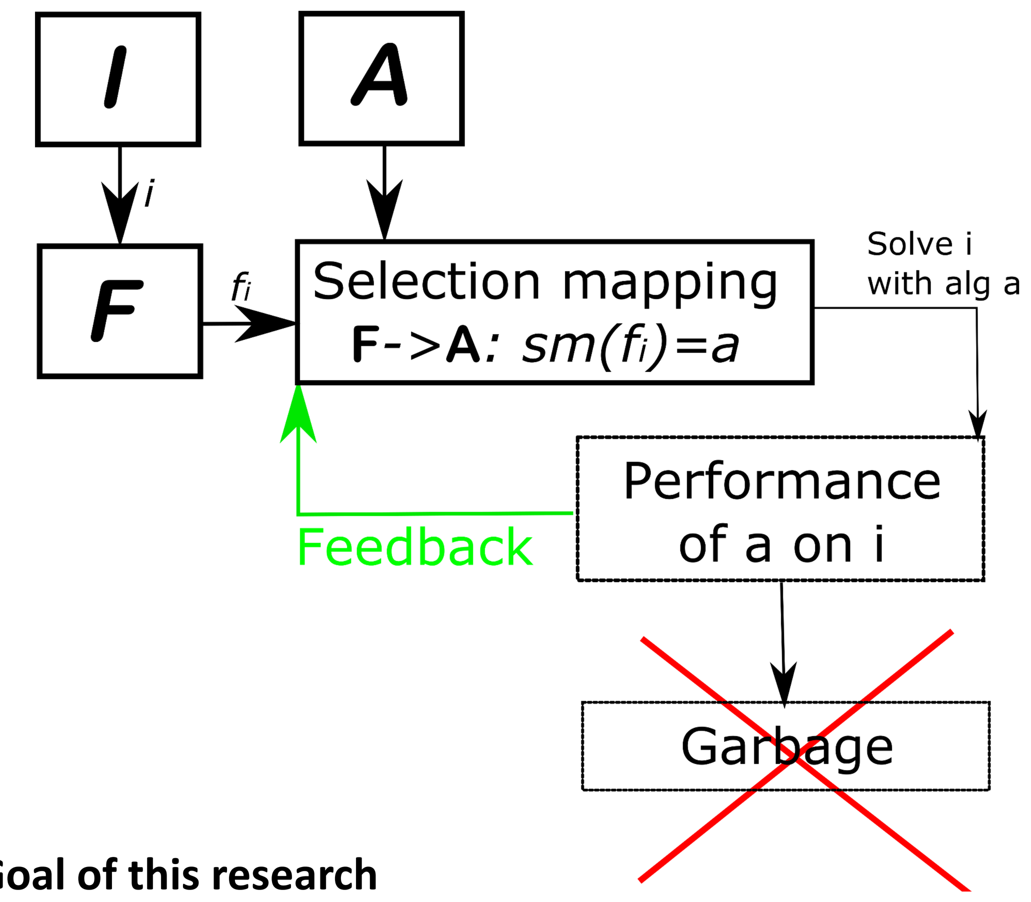


- Training data is fed to a supervised learning technique to create the selection mapping
- Training data = the performance of all algorithms on a set of training instances

Situating this research

Motivating observation

- A continuous stream of performance data is generated when doing algorithm selection
- This free data is not used to improve the selection mapping by current algorithm selection approaches

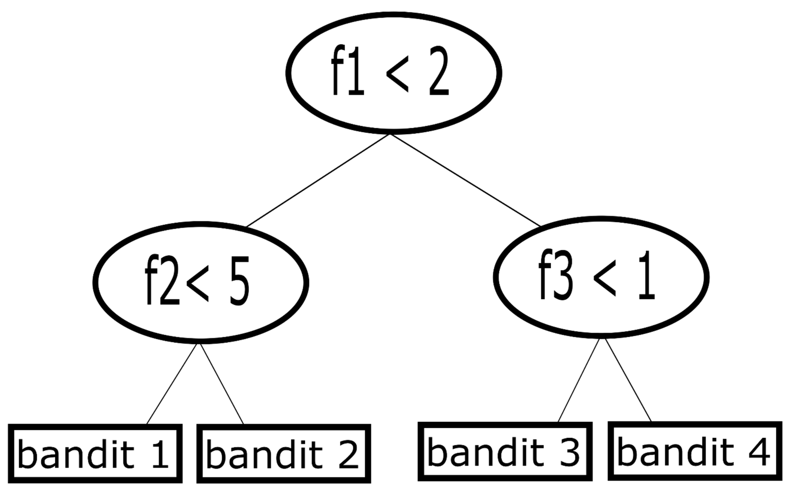


Goal of this research

- Apply active learning to algorithm selection by using the freely generated performance data feedback

Algorithm selection with multi-armed bandits: updating

- **After the training phase** the instance space is divided in subsets and a best algorithm has been learned for each subset, but it might not be the actual best.
- Based on the feedback generated during the online phase, each subset faces its own active learning problem: **"How can the best algorithm for my instance distribution be learned"?**
- More precisely: **"What is the regret-optimal policy for selecting an algorithm to solve sequential random samples from my instance distribution"?**
- This is a multi-armed bandit problem at heart:
  - ✓ arms = algorithms
  - ✓ pulling an arm = selecting an algorithm
  - ✓ reward = algorithm performance



Active learning for automatic algorithm selection

- Identify the subset to which an instance belongs
- Select the algorithm that is best according to a solution to the subset's multi-armed bandit problem
- Feed the performance of the selected algorithm back to the subset's bandit

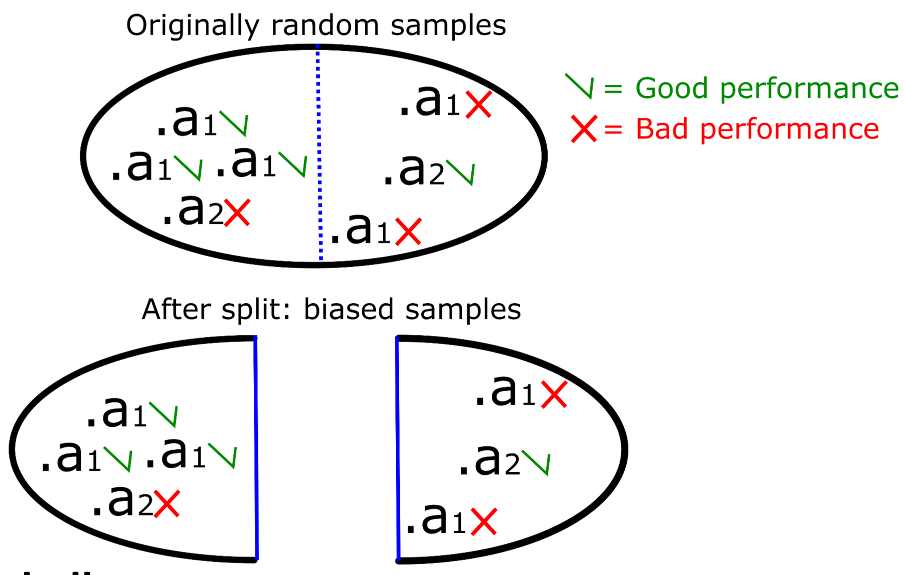
Challenges

- The type of distribution underlying algorithm performance is often unknown
- The distributions underlying algorithm performance might not have a well-defined mean (heavy-tailed distributions)

Algorithm selection with multi-armed bandits: modifying

Can performance feedback also be used to modify the way the subsets are divided?

- It might become clear that a subset has no decisively best algorithm. Introducing an additional split might prove beneficial.

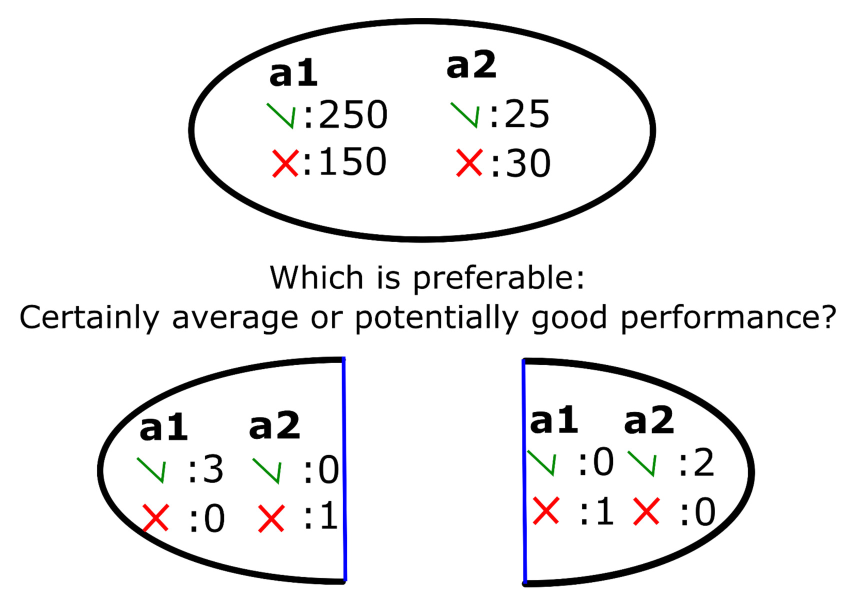


Main challenge

- The performance data points by which a new split is inspired are not random samples for the two newly defined subsets.

Inefficient workaround based on Zadrozny and Elkan's curtailment method for decision tree pruning

- Ignore the new subsets' biased samples
- Keep using the original subset's bandit
- Feed the results (which are unbiased) back to the new bandit processes until making a selection on the level of the new subsets is beneficial.



New challenge

- How to calculate the expected reward of a bandit process?

Is it possible to transfer knowledge between bandits?

- **Biased samples are produced** by
  - Offline data
  - Online data based on which the subset structure has been modified
- The proposed curtailment-inspired approach for modifying the subset structure ignores biased samples: the data used to define a new subset is not used to help learn the best algorithm for the new subset.
- However, the very reason the subset was created was because one algorithm appeared better. This algorithm should be given preference.

Questions to the multi-armed bandit community:

- Is it possible to incorporate the information contained in biased samples into a multi-armed bandit problem?
- How can the expected reward of a bandit process be calculated?
- What Is the link between Bayesian inference/updating and multi-armed bandits?

Conclusions

- An active learning methodology for automatic algorithm selection has been outlined
- At its core it relies on solving multiple multi-armed bandit problems
- An extension splits the reward distribution of a bandit, thereby introducing two new bandits for which biased samples are available that cannot be directly used, yet contain relevant information
- An additional complication is that the reward distributions are often unknown and might even have an ill-defined mean

Contact

Hans Degroote
KU Leuven
Email: Hans.Degroote@kuleuven-kulak.be
Phone: +32 56 24 62 88

comex
combinatorial optimization:
metaheuristics & exact methods