MULTI-OBJECTIVE OPTIMIZATION AND MULTI-ARMED BANDITS

Madalina M. Drugan
Artificial Intelligence Lab,
Vrije Universiteit Brussel

Overview

- Background
  - Multi-armed bandits
  - Multi-objective optimization
  - Adaptive operator selection

- Multi-objective optimisation in multi-armed bandits
  - Multi-objective Multi-armed bandits (MO-MAB)

- Multi-armed bandits in multi-objective optimisation
  - Adaptive operator selection using multi-armed bandits

- Related fields: multi-objective optimization under uncertainty

- Conclusions

- References

Multi-armed bandits (MAB)

- Popular mathematical formalism used to study the convergence properties of ***Reinforcement Learning*** with a single state
- A machine learning paradigm used to study and analyse resource allocation in stochastic and noisy environments.
- An example: a gambler faces a row of slot machines and decides
  - which machines to play,
  - how many times to play each machine
  - in which order to play them

- When played, each machine provides a reward generated from an unknown distribution specific to a machine.
- The goal of the gambler is to *maximise the sum of rewards* earned through a sequence of lever pulls.

Multi-armed bandits (MAB) algorithms

- Intuition on the MAB algorithms
  - An agent must choose between *N*-arms (= actions) such that the expected reward over time is maximised.
  - The algorithm starts by fairly exploring the *N*-arms, gradually focusing on the arm with the best performance.
  - The distribution of the stochastic payoff of the different arms is assumed to be unknown to the agent.

- ***Exploration / exploitation trade-off***
  - Explore the sub-optimal arms that might have been unlucky
  - Exploit the optimal arm as much as possible

- **Performance measures**
  - Cumulative regret is a measure of how much reward a strategy loses by playing the suboptimal arms

Multi-armed bandits: type of algorithms

- Continuous or discrete sets of arms
- Adversarial sets of arms
- Stochastic multi-armed bandits
  - Online selection of the arm with the maximum expected mean (i.e., the arm with higher expected reward)
    - The best arm can change over time
    - UCB1 [Auer et al, 2002]
  - Best arm identification algorithms
    - Fixed confidence vs fixed budget
    - Multiple best arm identification
- Contextual multi-armed bandits
  - uses the context to adapt the multi-armed bandit long term behaviour, or regret

Multi-objective optimization problem

- Simultaneous optimization of two or more objectives
- Pareto front —> a set of Pareto optimal solutions

- Dominance relations
  - Pareto dominance is a partial order relation where one solution can be better in one objective and worse in another objective compared to a second solution
  - Scalarizarion dominance transforms the value vector into a scalar value using a scalarization function

- Related with the field of multiple-criteria decision making where a user expresses his / her preference for an objective or a search region

- Real world applications: economics, optimal control, resource allocation, etc.

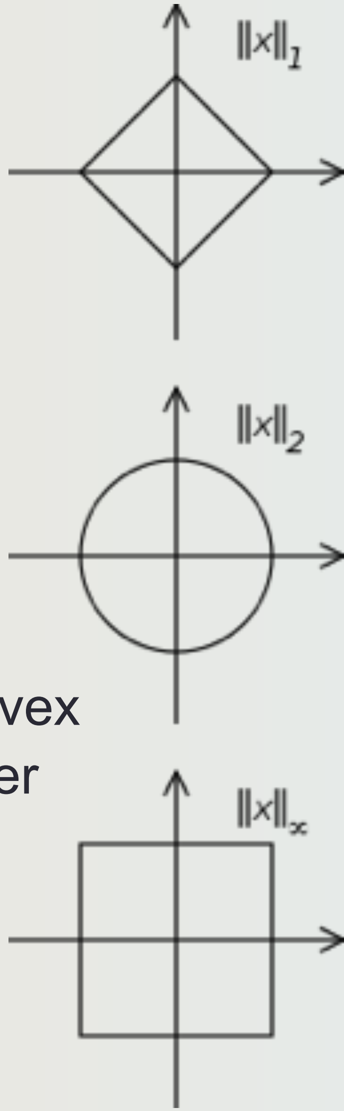
Pareto dominance relation

- *A reward vector can be better than another reward vector in one objective and worse in another objective*
- The natural order relationship for multi-objective search spaces
- Examples of relationships between reward vectors

relationship	notation	relationships
μ_1 dominates μ_2	$\mu_2 \prec \mu_1$	$\exists j, \mu_2^j < \mu_1^j$ and $\forall o, j \neq o, \mu_2^o \leq \mu_1^o$
μ_1 weakly domin μ_2	$\mu_2 \preceq \mu_1$	$\forall j, \mu_2^j \leq \mu_1^j$
μ_1 is incomp with μ_2	$\mu_2 \| \mu_1$	$\mu_2 \nprec \mu_1$ and $\mu_1 \nprec \mu_2$
μ_1 is non-domin by μ_2	$\mu_2 \nprec \mu_1$	$\mu_2 \prec \mu_1$ or $\mu_2 \| \mu_1$

- *The Pareto front is* the set of expected reward vectors that are *non-dominated* by the other expected reward vectors
- All the solutions in the Pareto front are considered equally important

Lp scalarization function



- *Goal*: Lp transforms the multi-objective search space into a single objective space using a scalarization function

- Weighted power p sums of reward values, where a set of predefined weights is considered

$$f_p(\mu_i) = \sqrt[p]{\sum_{j=1}^{D} \omega^j \cdot (\mu_i^j - z^j)^p}$$

  - L_p function can find all solutions of any shape, i.e. non-convex
  - The reference point $\mathbf{z} = (z^1, \ldots, z^D)$ is an extra parameter
    - L_1 function is *a linear scalarization* function
    - L_∞ function is *a Chebyshev scalarization* function

Multi-objective multi-armed bandits (MOMABs)

- Multi-armed bandits use reward vectors
- Evolutionary Computation (EC) techniques are used to design computationally efficient MOMABs

- The exploration / exploitation trade-off is common for both multi-armed bandits (MABs) and EC for multi-objective optimisation
  - In EC, exploration means evaluation of new solutions in a very large search space where states cannot be enumerated
  - In MAB, exploration means to pull arms that have suboptimal mean reward values
  - In EC, exploitation means to focus the search in promising regions where the global optimum could be located
  - In MAB, exploitation means to pull the currently identified best arm(s)

- MOMABs with a finite set of arms and reward vectors generated from stochastic distributions

Multi-objective multi-armed bandits (MOMABs)

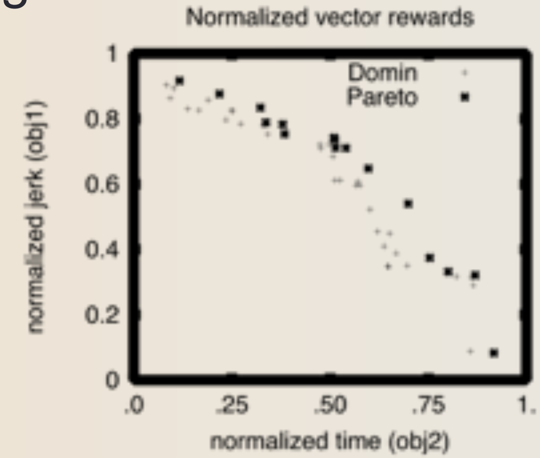
- The goal of MOMABs is either
  - to maximise the returned reward; or to minimise the regret of pulling suboptimal arms
  - identify the set of Pareto optimal arms
- We assume that all Pareto optimal arms are equally important and need to be identified
- Performance measures
  - Pareto regret → sum of the distances between each suboptimal arm and the Pareto front
  - Variance regret → variance in using the Pareto optimal arms
- Theoretical analysis
  - Upper and lower bounds on expected cumulative regret
- ***Challenges***
  - Large and complex *stochastic* multi-objective search spaces
  - Non-convex Pareto fronts
  - Non-contiguous mapping of attractors from the solution to the objective space

The bi-objective transmission problem of wet clutch

- An application from control theory
- *Goal:* optimise the functionality of the clutch:
  - the optimal *current profile* of the electro-hydraulic valve that controls the pressure of the oil to the clutch
  - the *engagement time*.

- *Stochastic output data* —> some external factors, such as the surrounding temperature, cannot be exactly controlled.

- *Goal: optimise the parameters* —> that minimise the clutch's profile and the engagement time in varying environmental conditions.



Stochastic discrete MOMAB problems

- *K*-armed bandit, $K \geq 2$, with independent arms
- The reward vectors have *D* –objectives, where *D* fixed

- An arm *i* is played at time steps $t_{1,i}, t_{2,i}, \ldots$
- The corresponding *reward vectors* $X_{i,t_1}, X_{i,t_2}, \ldots$ are independently and identically distributed according to an unknown law with unknown expectation vectors

- *The goal of MOMAB:*
  - Identify the set of best arms by simultaneously maximising rewards in all objectives
  - The arms in the Pareto front are considered equally important and should be pulled the same number of times.
  - Minimise the regret (or the loss) of not selecting the arms in the Pareto front

Pareto MAB algorithms

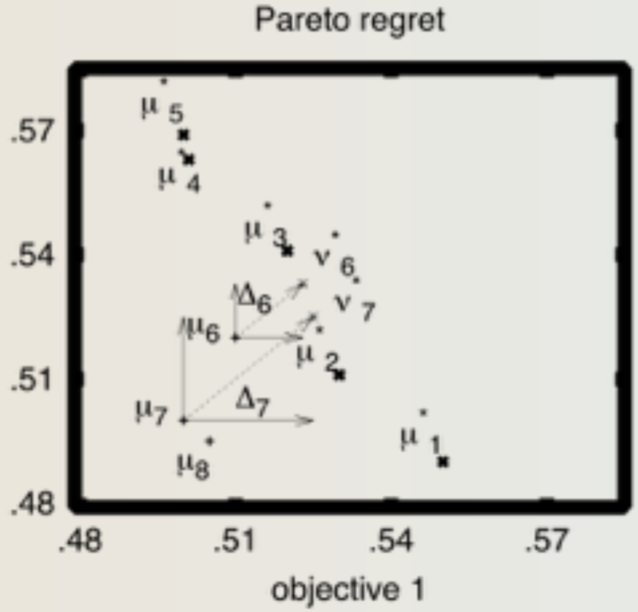
- ***Definition:*** a multi-objective MAB algorithm that uses the *Pareto partial order relationship*
- The Pareto regret metric is used to upper bound the performance of the designed Pareto MAB algorithms

- *Challenges* in designing Pareto MAB algorithms:
  1. Pareto front identification
     1. Identification of a representative Pareto set of arms
  - The exploitation/exploration trade-off:
    - Exploration: pull suboptimal arms that might be unlucky
    - Exploitation: pull as much as possible the optimal arms
  - Optimising the performance of Pareto MABs in terms of upper and lower bounds on expected and/or immediate regret
  - Ameliorate the performance of Pareto MABs for large sets of arms

Performance metric: Pareto regret

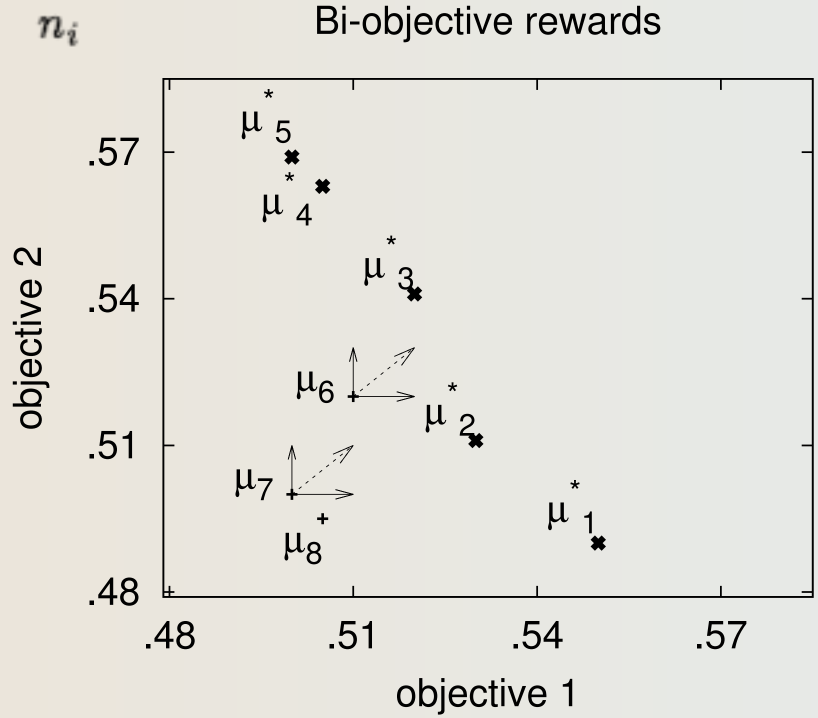
- We denote with $\Delta_i = \|\nu_i^* - \mu_i\|_2$ the empirical distance between an arm i and the Pareto front
- Let ν_i^* be the *virtual reward vector* of the arm i such that μ_i has the minimum distance to ν_i^*,
    - $\nu_i^* = \|\mu_i - \epsilon_i\|_2$ is incomparable with all reward vectors in the Pareto front and $\epsilon_i = (\epsilon_i, \ldots, \epsilon_i)$
- The *expected Pareto regret* for a learning algorithm after n arm pulls is

$$\mathbb{E}[R_n] = \sum_{i=1}^{K} \Delta_i \cdot \mathbb{E}[T_i(n)]$$



Pareto Upper Confidence Bound (PUCB1) [Drugan & Nowe, 2013]

- Straightforward generalisation of UCB1
  - operator selection [Fialho et al, 2009]
  - learning the utility of swap operations in combinatorial optimisation [Puglierin et al, 2013]
- Maximises the reward index $\hat{\boldsymbol{\mu}}_i + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_i}}$



Pareto Upper Confidence Bound (PUCB1)

- Each iteration, a Pareto front is calculated using

$$\hat{\mu}_h + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_h}} \succ \hat{\mu}_i + \sqrt{\frac{2\ln(n\sqrt[4]{D|\mathcal{A}^*|})}{n_i}}$$

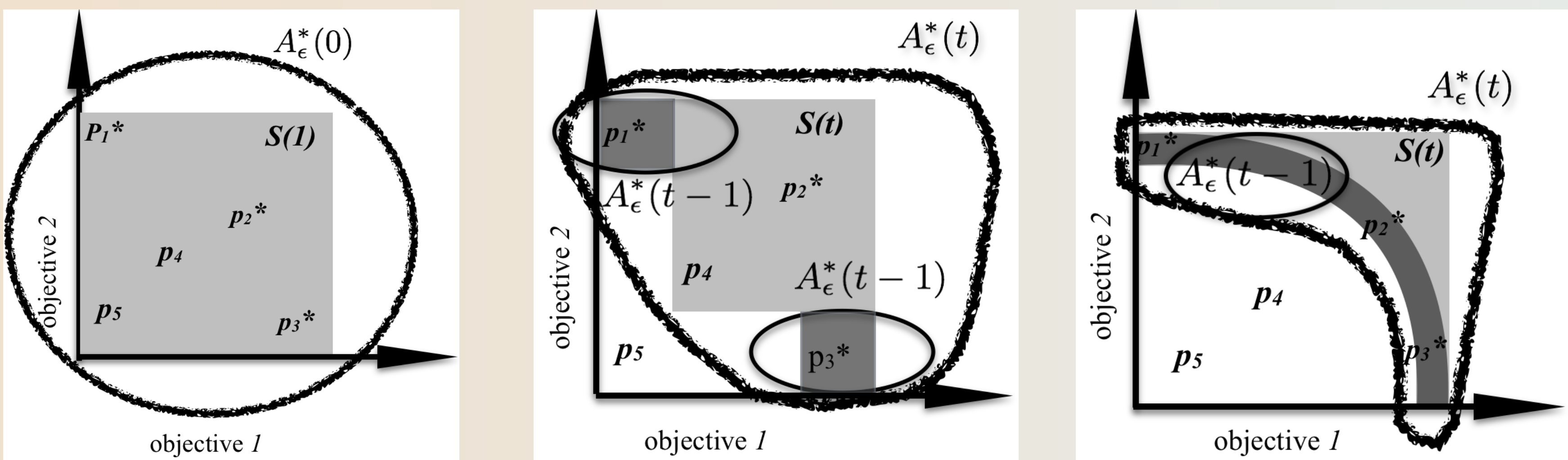
- One of the arms from the Pareto front is selected

- The upper bound is $\sum_{i \notin \mathcal{A}^*} \frac{8 \cdot \log(n\sqrt[4]{D|\mathcal{A}^*|})}{\Delta_i} + (1 + \frac{\pi^2}{3}) \cdot \sum_{i \notin \mathcal{A}^*} \Delta_i$

- The worst-case performance of this algorithm is when the number of arms *K* equals the number of optimal arms
- The algorithm reduces to the standard UCB1 for *D = 1*.
- Pareto UCB1 performs *similarly* with the standard UCB1 for a small number of objectives and small Pareto optimal sets

Annealing Pareto Knowledge gradient [Yahyaa et al, 2014]

- Knowledge gradient policy is a reinforcement learning algorithm where the reward vectors are updated using Bayesian rules
- Annealing like functions that decrease uncertainty around the arms
- The algorithm
  - At initialisation, all arms are considered
  - Iteratively, extreme arms are identified as either Pareto optimal or deleted as suboptimal arms
  - The iteration stops when there are no more arms to classify



Pareto front identification

- This policy is an extension of the *best arm identification algorithm* [Audibert et al.,2010] for a set of arms of equal quality.
- The *m*-best arm identification algorithm [Bubeck et al, 2013] assumes that the *m*-best arms can be totally ordered.

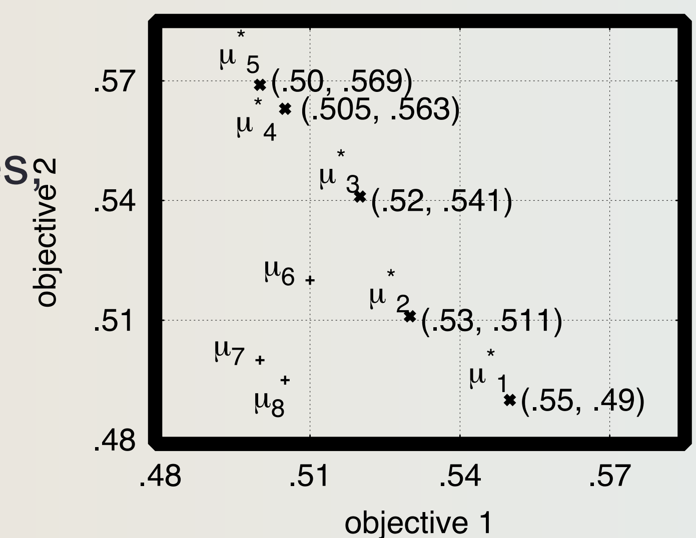
- ***The algorithm***
- Let $A_1 = \{1,\ldots,K\}$, $\overline{\log}(K) = \frac{1}{2} + \sum_{i=2}^{K} \frac{1}{i}$, $n_0 = 0$ and for $k \in \{1,\ldots,K-1\}$
- $$n_k = \left\lceil \frac{\log(D|\mathcal{A}^*|)}{\overline{\log}(K) + \log(D|\mathcal{A}^*|)} \cdot \frac{n-K}{K+1-k} \right\rceil$$
- For all rounds $k = 1,2,\ldots,K-1$
- (1) For each arm $i \in A_k$, select it for $n_k - n_{k-1}$ rounds
- (2) Let $A_{k+1} = A_k \setminus \text{argmin}_{i \in A_k} \widehat{\mathbf{x}}_{i,n_k}$ the arm to dismiss in this round
- Let the remaining set of arms be the Pareto optimal set of arms $\mathcal{A}^*$

ε- Pareto front identification [Drugan & Nowe, 2014a]

- ***Epsilon dominance relation*** assumes there exists a set of representative vectors that is a good approximation of a large Pareto front.
- The reward vector μ ε-dominates another reward vector ν, $\mu \succ_\varepsilon \nu$ iff for all the objectives *j*, we have $\mu^j + \varepsilon^j \geq \nu^j$ and $\exists o$ for which $\mu^o + \varepsilon^o > \nu^o$.
- ε^j positive constants defined for each dimension j, $\varepsilon^j > 0$
- If $\forall j$, $\varepsilon^j = 0$, we have the classical definition of dominance
- A set of reward vectors $\mathcal{O}_\varepsilon$ is an ε-approximate Pareto reward set $\mathcal{O}$, if any reward vector $\nu \in \mathcal{O}$ is ε-dominated by at least one reward vector $\mu \in O_\varepsilon$
- $$\forall\ \nu \in \mathcal{O}\ :\ \exists \mu \in \mathcal{O}_\varepsilon \text{ such that } \mu \succ_\varepsilon \nu$$

- The algorithm
  - A*ssign* arms to the hyper-grid boxes,
  - D*elete* arms that belong to the dominated boxes,
  - S*elect* a single representative arm in each non-dominated box,
  - Return the approximative front

Grid for ε = 0.03



Scalarized multi-objective multi-armed bandits

- Pareto front identification using a set of pre-definited or adaptive scalarization functions

- ***Convex Pareto fronts***
  - Generate the entire Pareto optimal set of arms with a minimum set of weights
    - No assumption on the distribution of the Pareto front
    - No guarantee that all Pareto optimal arms were identified for any set of scalarization functions
- ***Non-convex Pareto fronts***
  - ***Linear scalarization***
    - Easy to understand and to use
    - Not all the Pareto optimal reward vectors are reachable

  - ***Chebyshev scalarization***
    - There is no reference how to search for the set of optimal reference points that will generate the entire Pareto optimal set of arms
    - The reference point is an extra parameter to optimize

Performance of scalarized MOMABs

- *The scalarized regret metric*

$$\Delta_i^j =^{def} \max_{k \in \mathcal{A}} f^j(\mu_k) - f^j(\mu_i), \quad \forall j$$

  - where the optimum reward value $\boldsymbol{\mu}^*$ is the reward for which the function f_p^j attains its maximum value

$$f_p^j(\mu^*) = argmax_{k \in \mathcal{I}} f_p^j(\mu_k)$$

  - the maximum value for any set of weights is a Pareto optimal arm
  - this regret alone is improper for the MOMAB algorithms because it gathers a collection of independent regrets instead of minimizing the regret in all objectives

- *The Pareto variance regret metric*

$$R_v(n) = \frac{1}{|I^*|} \cdot \sum_{i \in I^*} \left(\frac{T_i^*(n)}{n} - \mathbb{E}\left[\frac{T^*(n)}{n} \right] \right)^2$$

  - *Measures the variance* in pulling the Pareto optimal arms
  - $T_i^*(n)$ the number of times that arm i is pulled during n number of pulls

Scalarized multi-objective UCB1 [Drugan & Nowe, 2013]

- A fixed set of weight vectors $S = (f^1, \ldots, f^s)$
- Independent scalarized UCB1 algorithms
- Regret is independently measured for each scalarized UCB1

- The scalarized multi-objective UCB1 algorithm
  - Initialize the scalarized UCB1 for all the scalarized functions f^j
  - $n \leftarrow S \cdot K\ ;\ n_i \leftarrow S$
  - Until some stopping criteria is met do
    - Choose uniform randomly a function f^j
    - Play one time scalarized UCB1 for $f^j = (w_1^j, \ldots, w_m^j)$
    - Play each arm once
    - $\forall j, i,\ n^j \leftarrow K\ ;\ n_i^j \leftarrow 1$

    Update mean vectors and counters

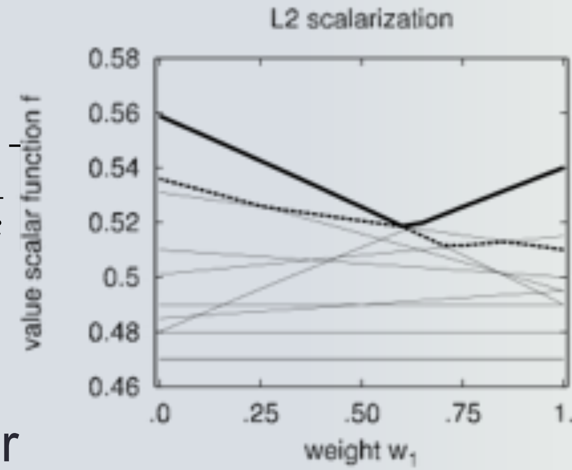
Scalarized multiple arm successive accepts and rejects [Drugan & Nowe, 2014b]

- Successively deletes suboptimal arms in K-1 rounds
- The length of the rounds increases with the number of arms' pulls
- We consider a *fixed* set of scalarization functions $S = \{f^1, \ldots, f^{|S|}\}$
- Each scalarization function is associated with a set of *active arms*

- sSAR assumes there are ***p*** Pareto optimal arms identifiable with each scalarization function

- To each scalarization function is assigned
  - A set of active arms that is initialized to the set of arms I
  - A set of accepted arms that is initialized to the empty set
- An arm i is deleted in the k-th epoch from the active set if it maximizes the reward gap to the p(k)+1-st arm
- The deleted arm i is accepted if better than the p(k) best arm
- The algorithm stops when there are p arms identified as the best arms

Scalarized multiple arm successive accepts and rejects



- Initialization:
  - for each scalarization function f^j, initialize the set of active arms $A_1^j \leftarrow \mathcal{A}$
  - The length of the k-th round is $n_k \leftarrow \left\lceil \frac{1}{\overline{\log}(K)} \cdot \frac{n-K}{K+1-k} \right\rceil$
  - The Pareto front $A^* \leftarrow \emptyset$,
  - the set of accepted arms $J_p^j \leftarrow \emptyset$
- For all rounds $k = 1, 2, \ldots, K-1$
  - For all the arms i for which $\exists j,\ i \in A_k^j$ play the arm for
  - For all the scalarization functions $f^j \in S$ do
    - Let $i \leftarrow argmax_{i \in A_k^j} \widehat{\Delta}_{(i)j}^{<p^j(k)>}$ be the arm to dismiss next $n_k - n_{k-1}$
    - Update the set of active arms $A_{k+1}^j \leftarrow A_k^j \setminus \{i\}$
    - If the arm i among the best $p^j(k)$ arms, $f^j(\widehat{\mu}_i) > f^j(\widehat{\mu}_{p^j(k)+1})$
      - Accept the arm i,
      - Update the set of accepted arms $J_{p-p^j(k)}^j \leftarrow i$
      - Set the remaining number of arms to be accepted to $p^j(k+1) \leftarrow p^j(k) - 1$
- Return the Pareto front as the reunion of accepted arms $\mathcal{I}_S^* \leftarrow \cup_{1 \le i \le p} \cup_{j \in S} J_i^j$

Shape driven Pareto front identification algorithms [Drugan, 2015a]

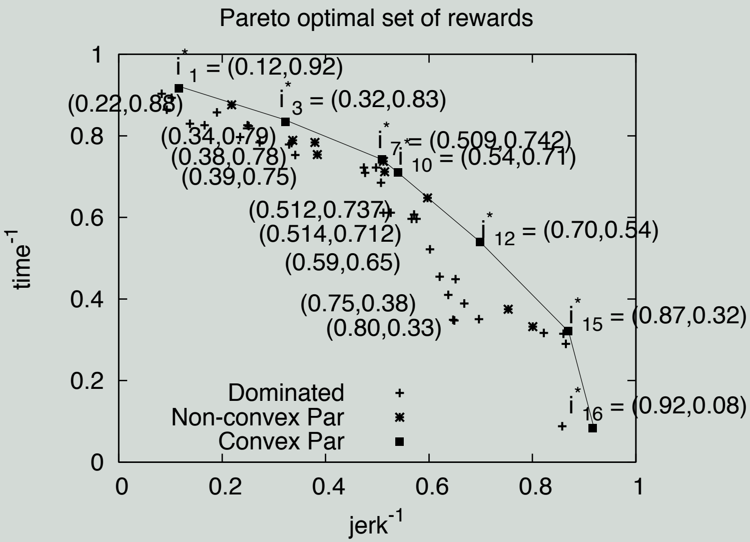
- ϵ-optimal arm given a scalarization f_ω is $f_\omega(\widehat{\boldsymbol{\mu}}_i) > \max_{\ell \in \mathcal{I}} f_\omega(\widehat{\boldsymbol{\mu}}_\ell) - \epsilon$
- ϵ is the accuracy probability and δ is the error probability
- A policy is (ϵ, δ) correct if $P\left(f_\omega(\widehat{\boldsymbol{\mu}}_i) > \max_{\ell \in \mathcal{I}} f_\omega(\widehat{\boldsymbol{\mu}}_\ell) - \epsilon\right) \geq 1 - \delta$
- Each arm is sampled an equal and fixed number of times

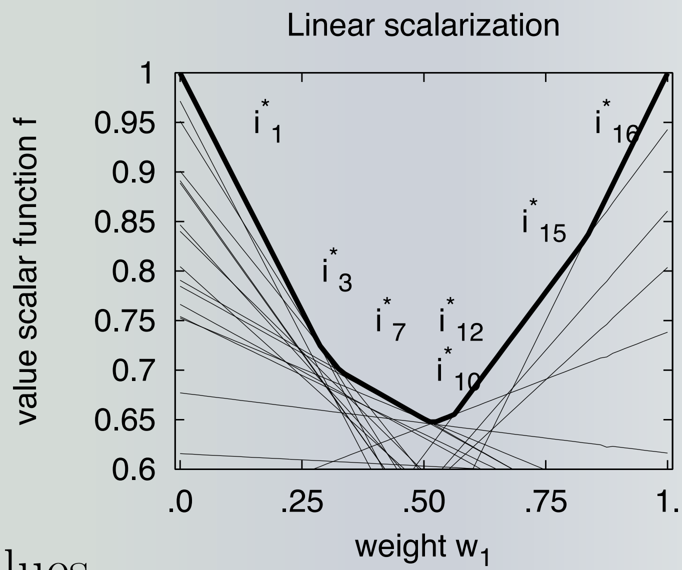
$$\frac{1}{(\epsilon/2)^2} \log\left(\frac{2K}{\delta}\right)$$

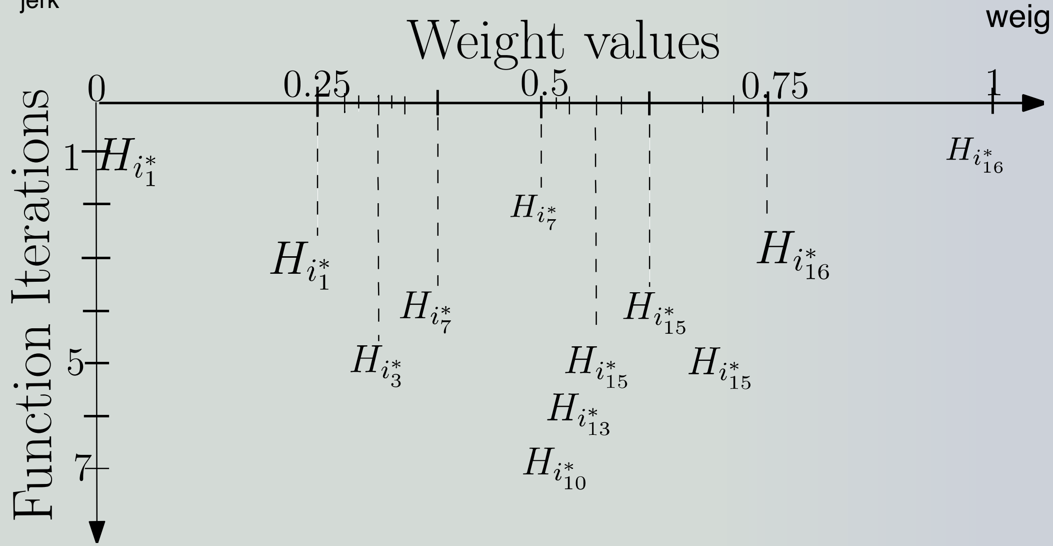
- For each weight vector, several ϵ- optimal arms are identified

- ***Weight D - rectangles***
  - Two weight vectors belong to the same D-rectangle if they have the same optimal arm.
    - Convex Pareto front → contiguous D-rectangles
    - We do not need to search further between two weight vectors belonging to the same D-rectangle
    - Update the list of D-rectangles with a new weight vector generated between two D - rectangles
    - Stop when the distance between two D-rectangles is less than accuracy

Weight hyper-rectangle decomposition on the wet clutch example







Challenges in designing scalarized MOMABs

- Identify the entire Pareto front
    - Large Pareto fronts
    - Non-convex Pareto fronts
    - Non-uniform distributions of arms on the Pareto front

- Optimising the performance of scalarized MOMABs in terms of upper and lower regret bounds
    - The *scalarized* / Pareto regret metric
    - The Kullback-Leibler divergence regret metric

- Exploitation/exploration trade-off:
    - Exploration: *sample scalarization* functions, and pull arms that might be unluckily identified as suboptimal
    - Exploitation: pull as much as possible the Pareto optimal arms of *relevant scalarization* functions

Multi-armed bandits for multi-objective optimisation

- Adaptive operator selection for evolutionary multi-objective algorithms
  - UCB1 is used in continuous multi-objective optimization [Ke Li et al, 2014]
  - adaptive pursuit is used for selecting scalarization functions for many-objective combinatorial optimization [Drugan, 2015b]

- Adaptive multi-operator selection
  - multi-objective multi-armed bandits (a multi-objective version of adaptive pursuit) is used to select multiple parameters for Pareto local search [Drugan & Talbi, 2014]

- Monte Carlo Tree search
  - splits the solution space into areas in order to focus the search in the most promising (high fitness) area

Adaptive operator selection

- Motivation:
  - the performance of EAs depends on the used parameters
  - the performance of a genetic operator depends on the landscape
  - an operator can have different performance in different regions of the landscape

- Tuning genetic operators
  - Selection of parameters
  - Mutation rates / Recombination exchange rates
  - Population size
  - Variable neighbourhood size (local search)

- Online learning strategy
  - The algorithms should learn relatively fast the best operator
  - There are several operators that perform similarly

UCB1 for online operator selection [Fialho at al, 2010]

- Each operator is considered an arm with unknown probability of getting a reward $\widehat{\mu}_i$
- The reward function for operator i contains
  - the estimated value for the operator
  - the exploitation coefficient

$$C\sqrt{\frac{2\log\sum_j n_j}{n_i}}$$

  - where n_i is the number of times the operator *i* was selected and *C* the exploration constant
- Remarks
  - Originally, UCB1 has positive sub-unitary values
  - Tuning *C* is important for any fitness landscape
  - UCB1 detects changes in the environment but will react quite slow to them
  - UCB1 is combined with other optimisation techniques to improve the performance of the online operator selection algorithm

UCB1 for operator selection in multi-objective optimization

- Performance of operator selection depends on the improvement measure considered like difference in fitness value and / or diversity

- Techniques to improve the performance of UCB1
  - Detect a change in the distribution with Page-Hinkley statistical tests
  - Weigh the operators using their frequency in applying it
  - Area under curve is also used as a measure of improvement in UCB1
  - Extreme values operator selection focuses on extremes to encourage exploration

- Hyper-parameter tuning, or tuning the tuner
- Off-line parameter tuning with F-race

- UCB1 is used to
  - select solutions that adapt the CMA-ES matrix in continuous MO-CMA-ES [Loshchilov et al, 2011]
  - select operators to improve the performance of MOEA/D algorithms [Ke Li et al, 2014]

Adaptive pursuit strategy (AP) [Thierens, 2005]

- Each operator i has associated a probability value $P_i^{(t)}$ of selection and an estimated reward value $Q_i^{(t)}$
- Online operator selection algorithm with fixed target probabilities is a step like distribution $D = \{p_M, p_m, \ldots, p_m\}$
  - p_M has a large probability value to select often the best operator
  - p_m has a small non zero probability to select any suboptimal operator
- The iterative algorithm
  - Pursuit with probability $P_v^{(t)}$ the operator *v* with the maximal estimated reward $Q_v^{(t)}$
  - Get reward vector $R_v^{(t)}$ for the operator *v*
  - Update reward value $Q_v^{(t)}$ using the immediate reward $R_v^{(t)}$
  - High rank the estimated reward distribution $Q_v^{(t)}$ and set the values in vector r
  - For each operator i, update the selection probabilities

$$P_i^{(t+1)} \leftarrow P_i^{(t)} + \beta \cdot (D_{r[i]} - P_i^{(t)})$$

Online multi-operator selection [Drugan & Talbi, 2014]

- Optimise the usage of two or more operators simultaneously
- Motivated by the quadratic assignment problem:
    - Exploring large variable neighbourhoods is expensive
    - Iterated local search is efficient for QAPs
- ***Probability distribution*** of the mutation and the neighbourhood operators

$$\mathbf{P}_{\mathcal{N}}^{(t)} = \{\mathbf{P}_{11}^{(t)}, \ldots, \mathbf{P}_{1K}^{(t)}\}, \quad \mathbf{P}_{\mathcal{M}}^{(t)} = \{\mathbf{P}_{21}^{(t)}, \ldots, \mathbf{P}_{2P}^{(t)}\}$$
$$\mathbf{Q}_{\mathcal{N}}^{(t)} = \{\mathbf{Q}_{11}^{(t)}, \ldots, \mathbf{Q}_{1K}^{(t)}\}, \quad \mathbf{Q}_{\mathcal{M}}^{(t)} = \{\mathbf{Q}_{21}^{(t)}, \ldots, \mathbf{Q}_{2P}^{(t)}\}$$

- ***Quality distribution*** of the mutation and the neighbourhood operators

$$\mathcal{Q}_{\mathcal{N}}^{(t)} = \frac{\#\ improv\ of\ v_{\mathcal{N}}}{\#\ trials\ of\ v_{\mathcal{N}}}, \quad \mathcal{Q}_{\mathcal{M}}^{(t)} = \frac{\#\ improv\ of\ v_{\mathcal{M}}}{\#\ trials\ of\ v_{\mathcal{M}}}$$

- ***Update reward vectors***: an improvement in the cost of the candidate solution when compared with the current solution

$$\mathcal{P}_{\mathcal{N}i}^{(t+1)} \leftarrow \mathcal{P}_{\mathcal{N}i}^{(t)} + \beta \cdot (\mathcal{D}_{r_{\mathcal{N}i}} - \mathcal{P}_{\mathcal{N}i}^{(t)}), \quad \forall 1 \leq i \leq P$$
$$\mathcal{P}_{\mathcal{M}j}^{(t+1)} \leftarrow \mathcal{P}_{\mathcal{M}j}^{(t)} + \beta \cdot (\mathcal{D}_{r_{\mathcal{M}j}} - \mathcal{P}_{\mathcal{M}j}^{(t)}), \quad \forall 1 \leq j \leq K$$

- ***Update probabilities***: the probability distributions are independently updated

Bandits trees for continuous multi objective optimisation

- Monte Carlo Tree Search (MCTS) is a heuristic used to solve intractable problems, i.e. huge search spaces, like playing computer Go
- MCTS builds a search tree using a search policy selecting the most probable nodes to expand
- A top down approach, i.e. root to leaves, with the following steps
  - Selection of the most promising children
  - Expansion creates new nodes using a tree policy
  - Simulation plays at random from the current node to the end of the game
  - Back-propagation updates the information on the explored path
- MCTS variants are used in optimisation of real-coded multi-dimensional functions by partitioning the search space in subdomains
- The search focuses on the most promising partitions, i.e. that contain the best solutions
- Simultaneous optimistic optimisation (SOO) [Preux et al, 2014] is successfully applied on many dimensional test problems from the CEC'2014 competition on single objective real-parameter numerical optimisation.
- SOO is straightforwardly extended to multi-objective optimisation in [van Moffaert et al, 2014]

Multi-objective optimization under uncertainty

- ***Stochastic multi-objective optimization*** [Gutjahr, 2011]
  - stochastic optimization and multi-objective optimization evolved separately even though their intersection is multi-criteria decision making (MCDM)
  - operational research —> risk analysis, finances, facilities allocations
    - combinatorial multi-objective optimization problems that use Pareto dominance
  - Risk neutral decision making
    - only expectations of reward vectors are optimised
    - linear utility functions are considered by the decision maker
  - Risk adversarial decision making
    - non-linear utility functions
    - both expectations are optimised and variations are minimised

Concluding remarks on multi-objective multi-armed bandits algorithms

- Multi-objective multi-armed bandits
  - Follows closely the latest developments in MABs and MOO
  - New theoretical tools needed to study the performance of MOMAB algorithms
- Multi-criteria reinforcement learning
  - Reinforcement learning is a generalisation of multi-armed bandits to multi-states that associate state and action pairs with transition probabilities
  - Hybrid algorithms between reinforcement learning and evolutionary computation
- Open research questions
  - Computationally efficient exploitation / exploration trade-off
  - Adequate performance measures for MOMABs
  - Advanced MOO techniques to improve the performance of MOMAB algorithms
  - Challenging real world problems to motivate MOMABs paradigms

Concluding remarks on multi-armed bandits and multi-objective optimization

- New emerging paradigms between multi-objective optimisation and multi-armed bandits problem
  - to solve challenging realistic problems for example finances and engineering
    - incomplete observations and / or large stochastic and changing environments
  - potential to develop new algorithms for automatic parameter tuning

- Focus on integrating techniques from one problem to another depending on the goal of the designed algorithm
  - deterministic or stochastic optimization
  - finite, large or continuous search spaces (or environments)

References

[Auer et al, 2002] Peter Auer, Nicolò Cesa-Bianchi, Paul Fischer: Finite-time Analysis of the Multiarmed Bandit Problem. Machine Learning 47(2-3): 235-256 (2002)

[Fialho et al, 2009] Álvaro Fialho, Luís Da Costa, Marc Schoenauer, Michèle Sebag: Dynamic Multi-Armed Bandits and Extreme Value-Based Rewards for Adaptive Operator Selection in Evolutionary Algorithms. LION 2009: 176-190

[Puglierin et al, 2013] Francesco Puglierin, Madalina M. Drugan, Marco Wiering: Bandit-Inspired Memetic Algorithms for solving Quadratic Assignment Problems. IEEE Congress on Evolutionary Computation 2013: 2078-2085

[Audibert et al, 2010] Jean-Yves Audibert, Sébastien Bubeck, Rémi Munos: Best Arm Identification in Multi-Armed Bandits. COLT 2010: 41-53

[Drugan & Nowe, 2014a] Madalina M. Drugan, Ann Nowé: Scalarization based Pareto optimal set of arms identification algorithms. IJCNN 2014: 2690-2697

[Drugan, 2015a] Madalina M. Drugan: Linear Scalarization for Pareto Front Identification in Stochastic Environments. EMO (2) 2015: 156-171

[Drugan, 2015b] Madalina M. Drugan: Stochastic Pareto local search for many objective quadratic assignment problem instances, CEC 2015

[Drugan, 2015c] Madalina M. Drugan: Multi-objective optimization perspectives on reinforcement learning algorithms using reward vectors, ESANN 2015

[Yahyaa et al, 2014] Saba Q. Yahyaa, Madalina M. Drugan, Bernard Manderick: Annealing Pareto multi-objective multi-armed bandit algorithm. ADPRL 2014: 1-8

[Thierens, 2005] Dirk Thierens: An adaptive pursuit strategy for allocating operator probabilities. GECCO 2005: 1539-1546

[Fialho et al, 2010] Álvaro Fialho, Luís Da Costa, Marc Schoenauer, Michèle Sebag: Analyzing bandit-based adaptive operator selection mechanisms. Ann. Math. Artif. Intell. 60(1-2): 25-64 (2010)

[Drugan & Thierens, 2011] Madalina M. Drugan, Dirk Thierens: Generalized adaptive pursuit algorithm for genetic pareto local search algorithms. GECCO 2011: 1963-1970

[Loshchilov et al, 2011] Ilya Loshchilov, Marc Schoenauer, Michèle Sebag: Not All Parents Are Equal for MO-CMA-ES. EMO 2011: 31-45

[Drugan & Talbi, 2014] Madalina M Drugan, Talbi El-Ghazali: Adaptive Multi-operator MetaHeuristics for quadratic assignment problems. EVOLVE 2014, Springer

[Kocsis & Szepesvári, 2006] Levente Kocsis, Csaba Szepesvári: Bandit Based Monte-Carlo Planning. ECML 2006: 282-293

[Preux et al, 2014] Philippe Preux, Rémi Munos, Michal Valko: Bandits attack function optimization. IEEE Congress on Evolutionary Computation 2014: 2245-2252

[van Moffaert et al, 2014] Kristof Van Moffaert, Kevin Van Vaerenbergh, Peter Vrancx, Ann Nowé: Multi-objective χ-Armed bandits. IJCNN 2014: 2331-2338

[Gutjahr, 2011] Walter J. Gutjahr: Recent trends in metaheuristics for stochastic combinatorial optimization. Central Europ. J. Computer Science 1(1): 58-66 (2011)

[Ke Li et al, 2014] Ke Li, Álvaro Fialho, Sam Kwong, Qingfu Zhang: Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Trans. Evolutionary Computation 18(1): 114-130 (2014)

[Drugan & Nowe, 2014b] Drugan, M. M., & Nowe, A.. (2014). Epsilon-approximate Pareto optimal set of arms identification in multi-objective multi-armed bandits. In *BENELEARN 2014 - 23th annual Belgian-Dutch conference on Artificial Intelligence*. presented at the 06/2014

[Wiering & van Otterlo, 2012] Marco A. Wiering, Martijn van Otterlo: Reinforcement Learning: State-of-the-Art, Springer, 2012