Real time solving of online discrete optimization problems

Yair Nof
Ofer Strichman
IE&M, Technion

Problem properties

A discrete optimization problem with the following characteristics:

- **Incremental**
- **Online** – Future input is unknown
- **Temporal constraints** – Solution values are associated with durations
- **Real time** – Find solution within, e.g., 1 sec.

NP-Hard, Should be solved fast ➔ Approximation

Problem's Sources

1. **A group of robots which explore Mars**
   - Initial information about scientific value, which updates when some robot discovers an interesting area, or a non-relevant one
   - When a robot moves, it imposes constraints on the other robots' movements for several following time steps
   - A decision problem which has to be solved fast – no extra batteries
2. **Stocks investment system**
   - News events influence investment decisions
   - An investment allocates part of the money aside, for a time period
   - Time is money
3. …

Problem Formulation

The Constrained Optimization Problem

Defined by a tuple $\boldsymbol{P} = \langle \boldsymbol{V}, \boldsymbol{D}, \boldsymbol{C}, \boldsymbol{F} \rangle$ where

Constraint Satisfaction Problem

$\boldsymbol{V} = \{\boldsymbol{V_1}, \dots, \boldsymbol{V_n}\}$ is the set of **variables**

$\boldsymbol{D} = \{\boldsymbol{D_1}, \dots, \boldsymbol{D_n}\}$ is the set of **domains**, defining a domain for each of the variables in V

$\boldsymbol{C} = \{\boldsymbol{C_1}, \dots, \boldsymbol{C_m}\}$ is the set of **constraints** over V

Objective function

$\boldsymbol{F}: \boldsymbol{D_1} \times \cdots \times \boldsymbol{D_n} \rightarrow \boldsymbol{R}$ is the **objective function**

An optimal solution to the COP is an assignment S^* of values to variables such that no constraint is violated, and every other solution S, implies $F(S^*) \geq F(S)$.

Problem Formulation

The Dynamic Constraint Optimization Problem

The DCOP is an **online series of COPs**: $P1, P2, \ldots$ where

$\boldsymbol{FinalD^i}$ is a partial assignment added by the solution of P_i

Each domain value d is associated with a time interval,

$t_d = [start_d; end_d]$

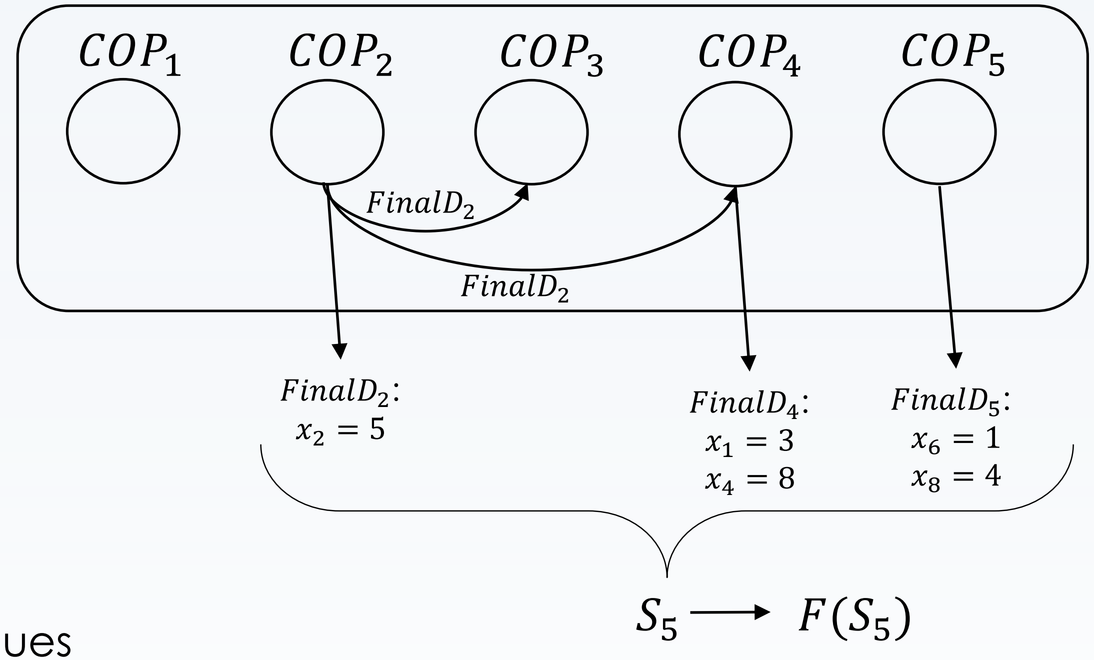
The solver of P_i might choose some variables set to domain values

with $t = [i + 1; i + k]$ for some $k \geq 1$.

The chosen domain values are added to the set $FinalD^i$.

An optimal solution to the DCOP is a series of solutions to its COPs

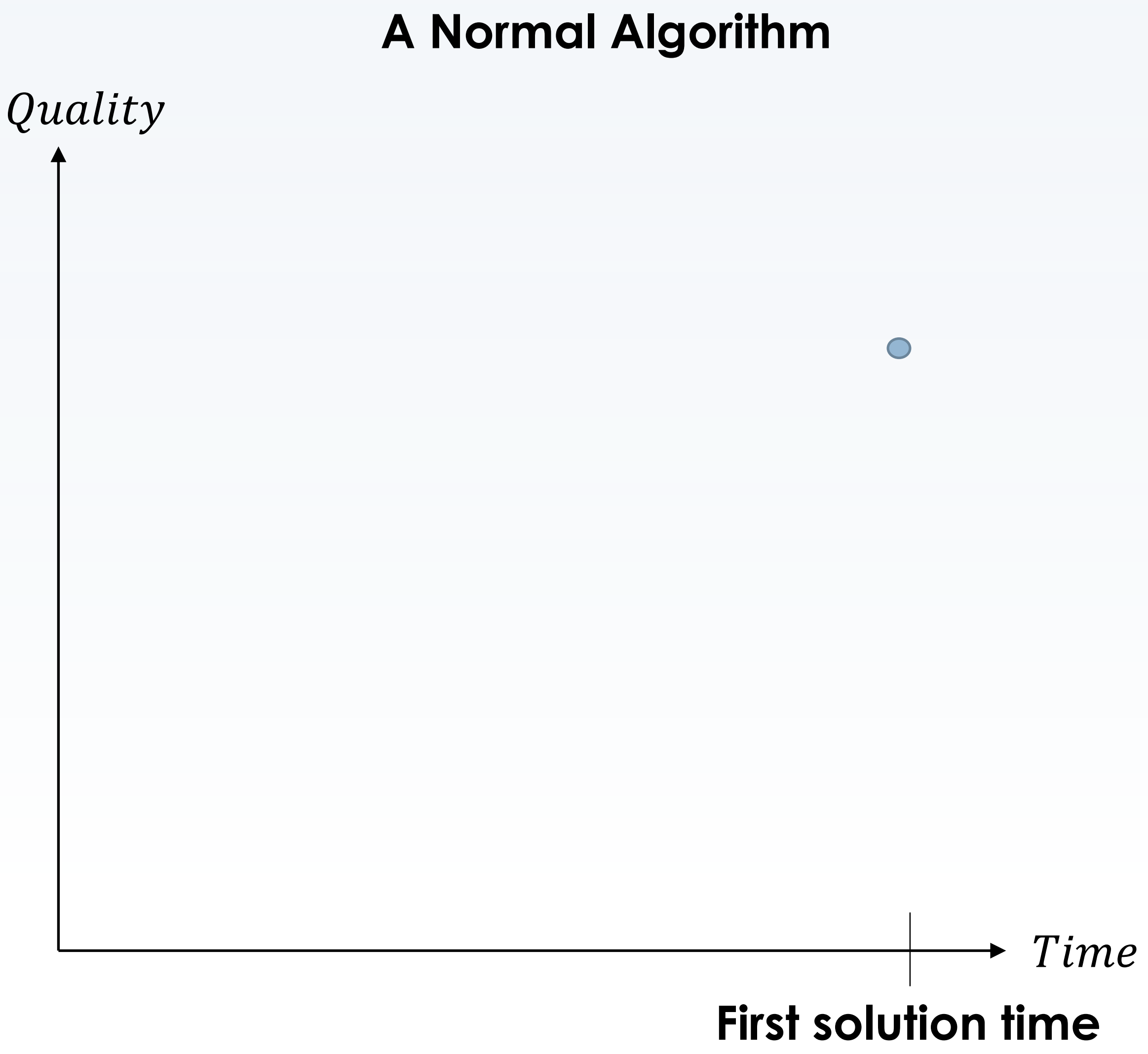
with the best objective function over $\left(\bigcup_{i=1}^{g-1} FinalD^i\right)$.

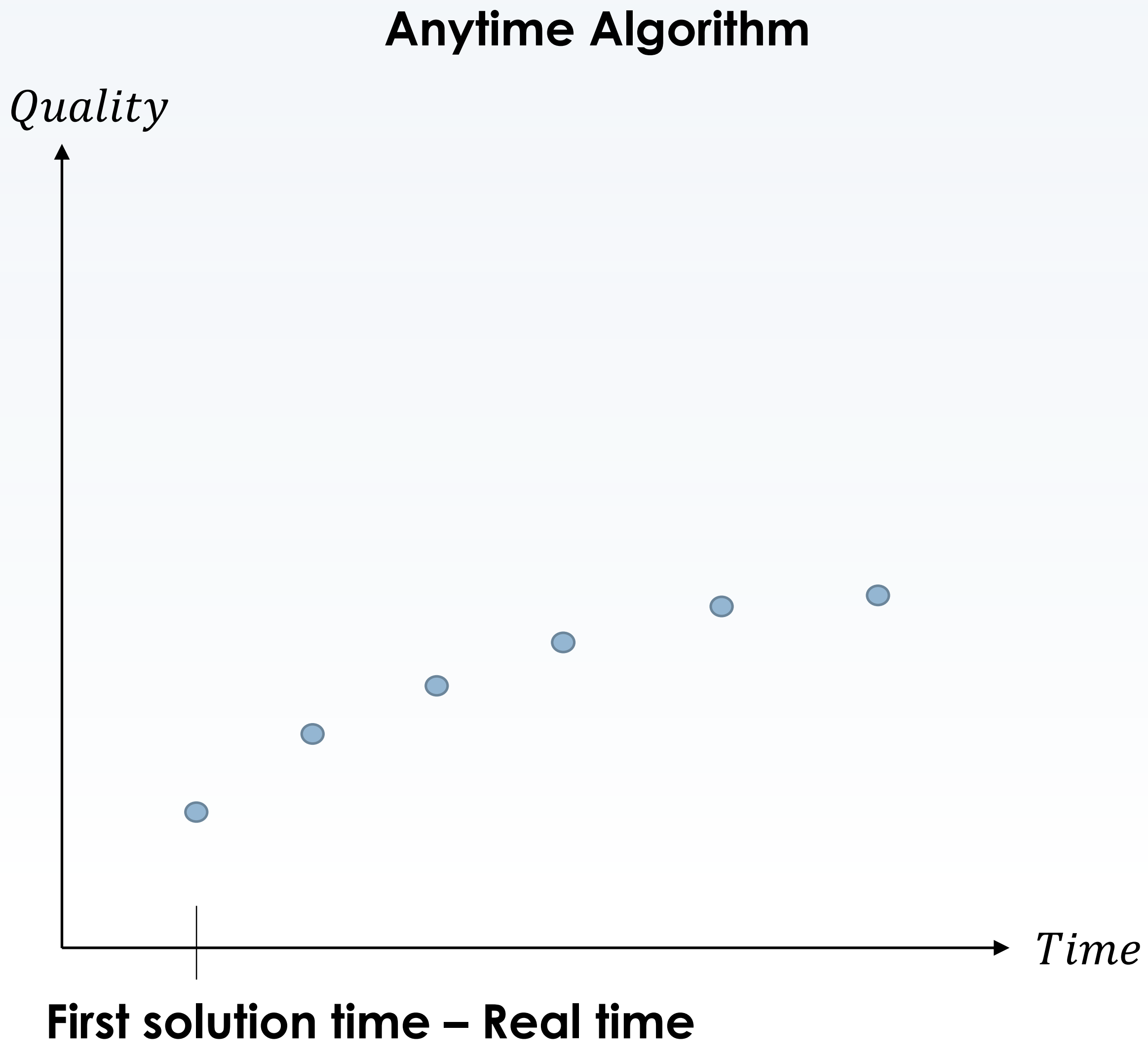


Solution approaches

- **Hardware**: Multiple cores
- **Software**: Anytime algorithms

Normal vs. Fast Anytime algorithm





Candidate Algorithms

- **Local Search – Full Solutions, only partially feasible**
  - The Cross-Entropy Method
  - Simulated Annealing
  - Tabu Search
  - Stochastic Hill Climbing
- **Complete Algorithms – Partial solutions, feasible**
  - Depth-First Branch And Bound
  - Anytime variants of A*
- **Online Search**
  - Learning Real Time A*
  - Monte Carlo Tree Search

For the COP

For the DCOP

Candidate Algorithms

CDF_i –probability distribution over D_i (initially uniform)
N – sample size
ρ – elite ratio
α – smoothing factor

The Cross-Entropy Method for COP

$$\text{while } (!\,\text{converged} \;\&\&\; !\,\text{timeout})\ \{$$

$$\quad for\ (j = 1..N)\ \{$$

$$\quad\quad for\ all\ i, choose\ d_i\ randomly\ according\ to\ CDF_i$$

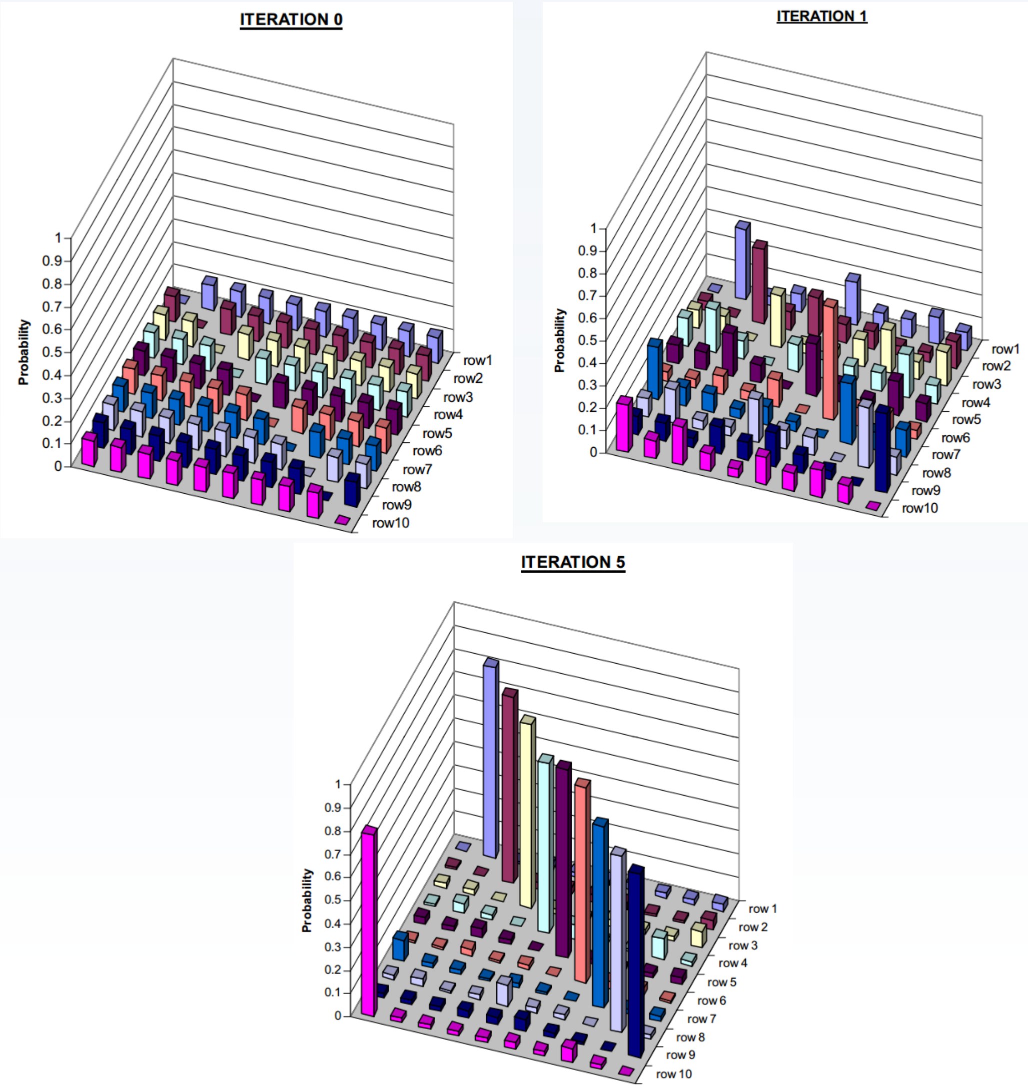
$$\quad\quad\quad EvaluateSolution$$

$$\quad \}$$

$$\quad Choose\ \rho \cdot N\ best\ solutions\ to\ form\ distribution\ CDF_i'$$

$$\quad CDF_i = \alpha CDF_i' + (1-\alpha) CDF_i$$

$$\}$$



de Boer et al (2005)

Candidate Algorithms

Monte Carlo Tree Search for DCOP

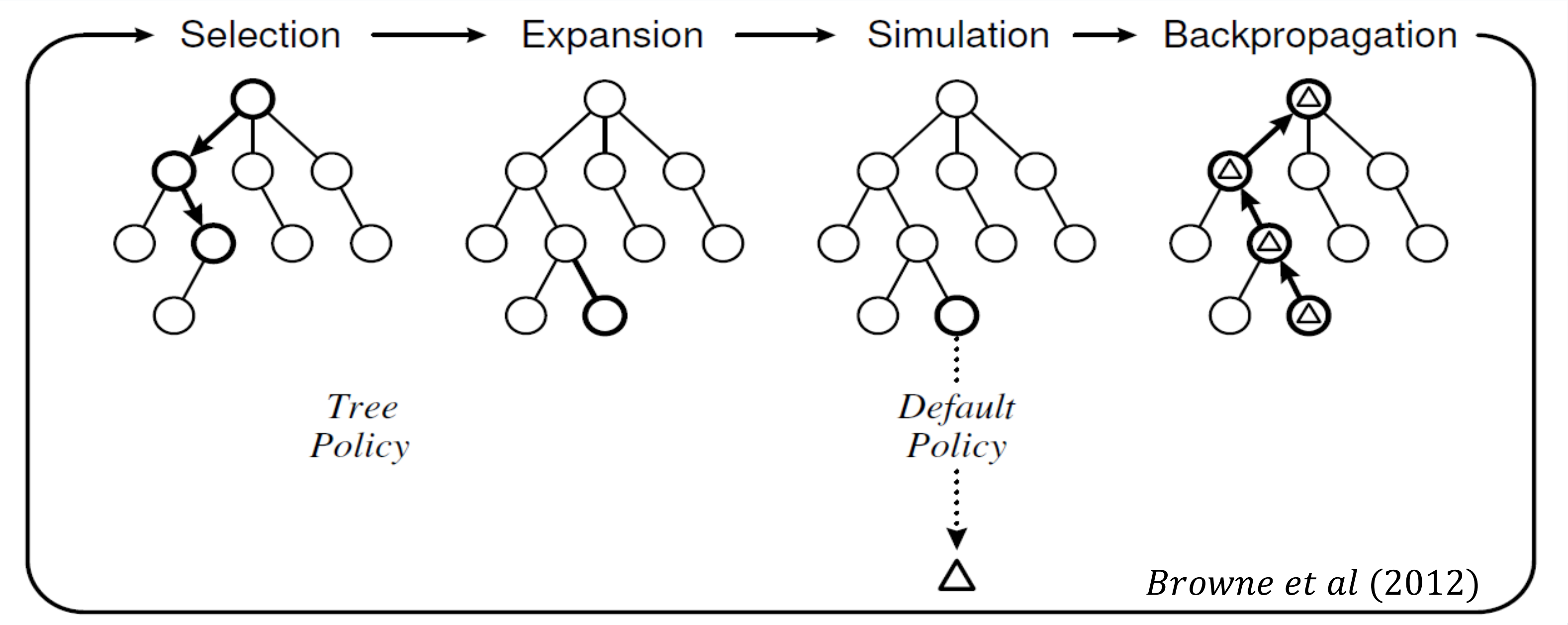
$A_0 = EmptyAssignment$
$while\ (!\,\text{timeout})\ \{$
$\quad A_l \leftarrow \epsilon - \text{greedyTreePolicy}(A_0);$
$\quad \Delta = DefalutRandomPolicy(A_l);$
$\quad BackPropagate(A_l, \Delta)$
$\}$
$return\ best(A_1)$

$\boldsymbol{\epsilon - \text{greedyTreePolicy}(\text{A})}$**:**

$while\ (!\,OnTree)\ \{$
$\quad TreeAssignment(CurrentTimeStep) \leftarrow$
$\qquad \text{best}\ assignment\ with\ prob.\ 1-\epsilon$
$\qquad random\ assignment\ with\ prob.\ \epsilon$
$\quad CurrentTimeStep++;$
$\}$
$return\ TreeAssignment;$

$\boldsymbol{DefaultRandomPolicy}(\text{A})$**:**

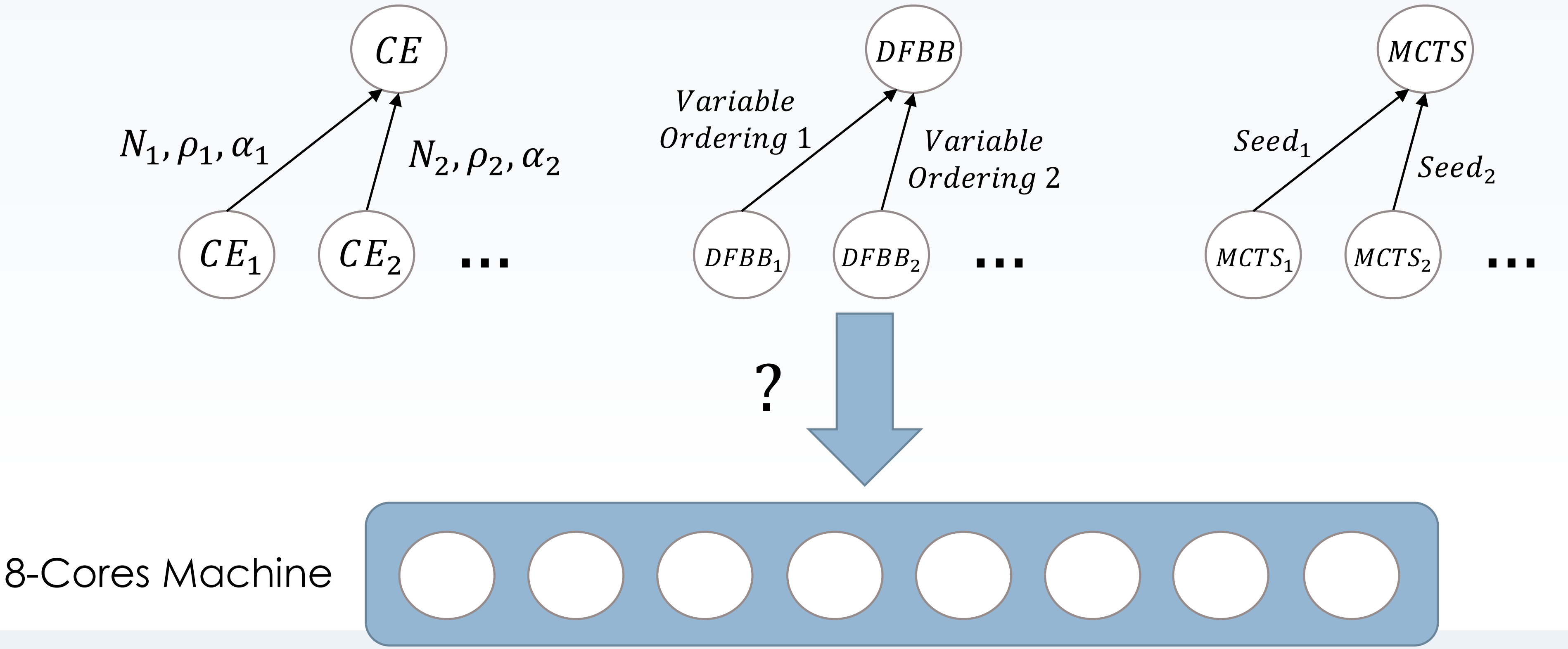
$while\ (!\,AllVariablesAssigned)\ \{$
$\quad DefaultAssignment(CurrentTimeStep)$
$\qquad \leftarrow Random;$
$\quad CurrentTimeStep++;$
$\}$
$return\ DefaultAssignment;$



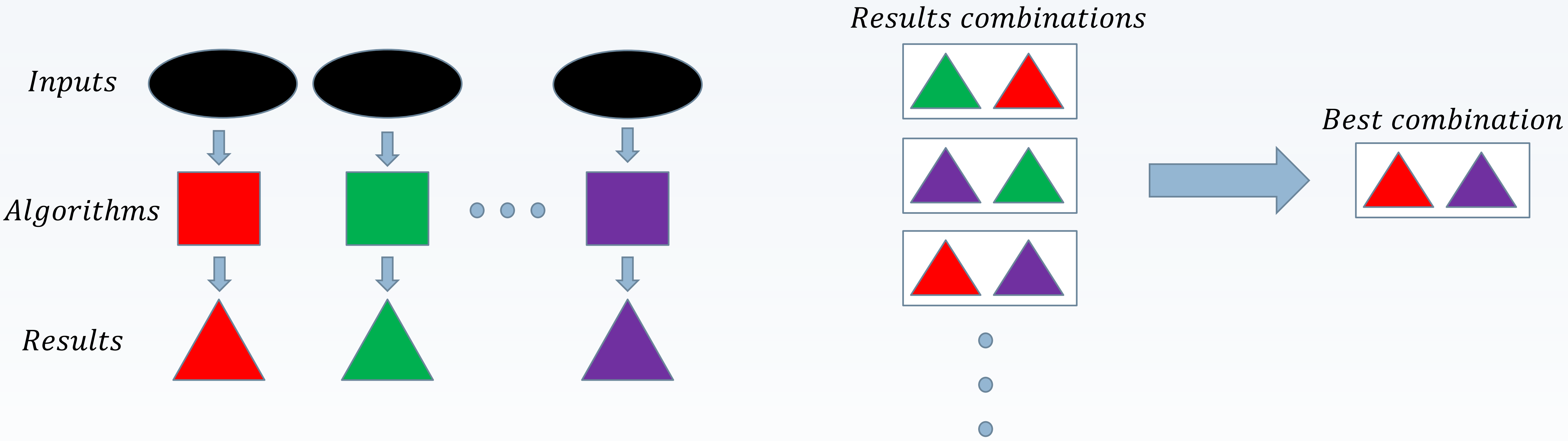
Possible assignments **sorted by time steps**

Parallel Portfolio – Diversify & Choose algorithms

- Different algorithms
- Different parameters to the same algorithm
- Adding randomness to deterministic algorithms



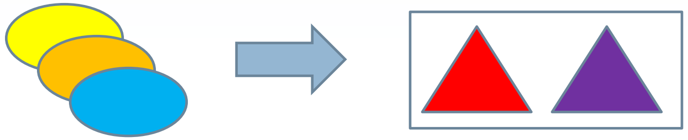
Parallel Portfolio – Static, Non-cooperating



Offline

1. Each algorithm is run alone

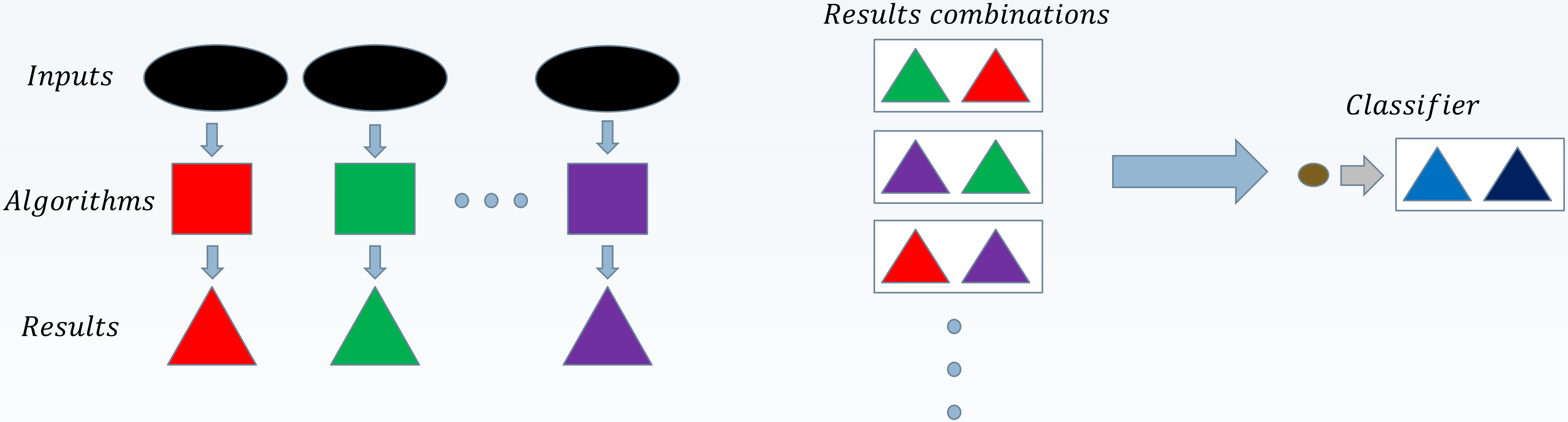
2. Search results



Online

Use the single portfolio

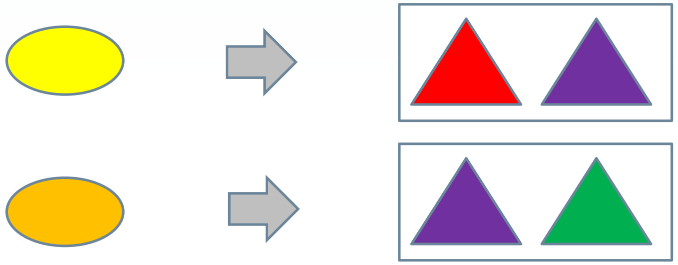
Parallel Portfolio – Dynamic, Non-cooperating



Offline

1. Each algorithm is run alone

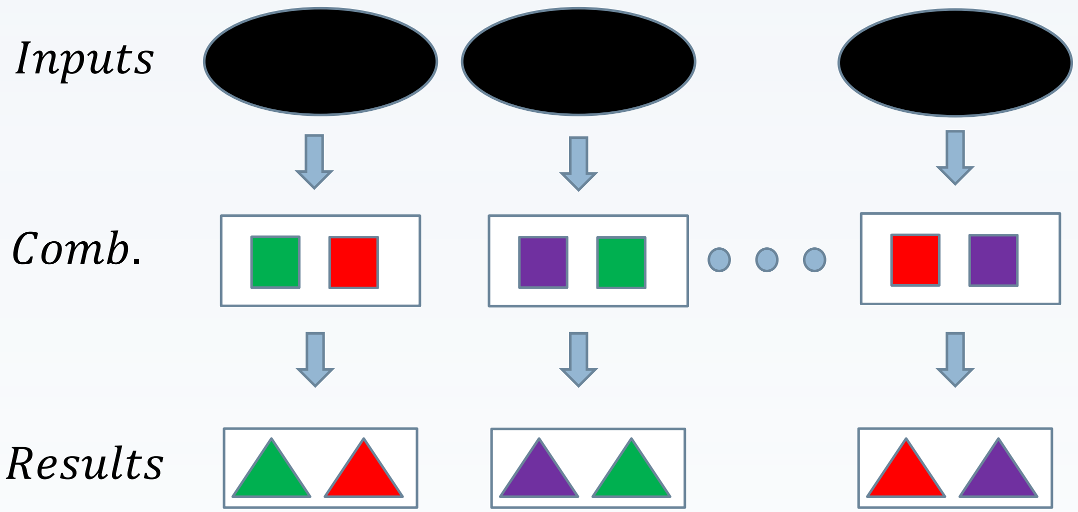
2. Learn a classifier over results

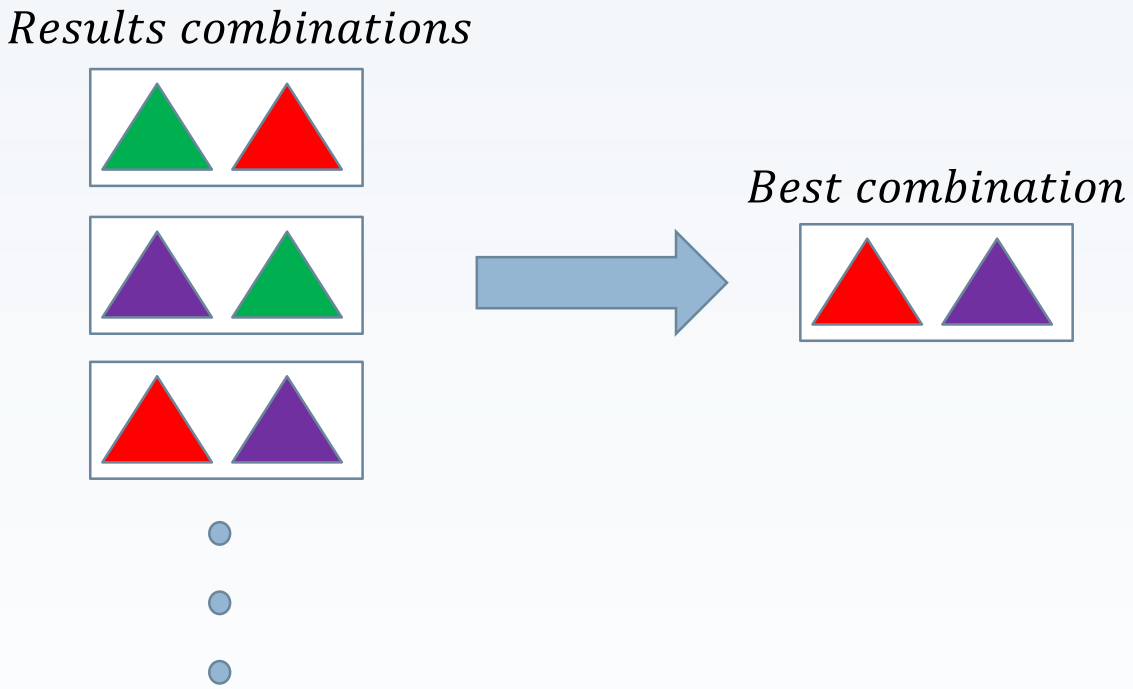


Online

Fit a portfolio

Parallel Portfolio – Static, Cooperating

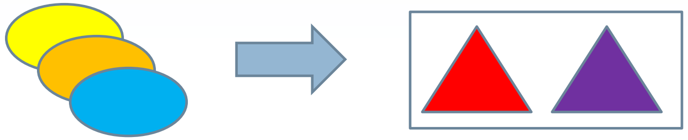




Offline

1. Each algorithm is run alone

2. Search results



Online

Use the single portfolio

Summary

- The Dynamic Constrained Optimization Problem is a realistic problem arises in many systems: An Online COP which should be solved in **real-time**
- The requirement to solve real time suggest the use of **anytime algorithms** and **parallelism**
- Candidate algorithms include **complete anytime methods** (eg. DFBB), **stochastic local search** (eg. CE) and **Online search** (eg. MCTS).
- A good **non-cooperating static parallel portfolio** is a single portfolio. It can be built **systematically** using offline runs of each algorithm alone on a given set of inputs, and **search the space of results** for a good subset of algorithms.
- A good **non-cooperating dynamic portfolio** is a function from problem's features to a subset of algorithms. It can be built by learning a classifier.
- In a **cooperating portfolio**, the offline runs are over portfolios, not single algorithms