Special Meeting #2

"Research Journey Update"

Hedieh.haddad@uni.lu

20th June 2025

The Challenge

The Tuning Bottleneck

- Constraint Programming (CP) solver performance is highly sensitive to its internal **hyperparameters**.
- Finding the best configuration manually is difficult, time-consuming, and a major barrier to using solvers effectively.
- A lot of possible parameters, but a set of parameters not always good on each problem.
- It is left to the user to manually pick the best set of parameters to obtain the best efficiency.

Research Project: Problem Statement

Problem

Large Parameter Space

Inefficient Methods

Lack of reusable framework

Research Project: Problem Statement

Problem

Large Parameter Space

Inefficient Methods

Lack of reusable framework

Solution

Hyperparameter optimisation (HPO)

Hyperparameter optimisation (HPO)

- HPO is the process of selecting the optimal values for the algorithm’s hyperparameters.
- HPO is very successful in the other fields like ML.
- HPO can improve tremendously the efficiency of the algorithms in ML.

- There are several strategies for hyper-parameter optimisation, including:
    - Grid search
    - Random search
    - Hyper-band optimisation
    - **Hamming distance**
    - **Bayesian optimization**
    - …

Research Project: Problem Statement

Problem

Large Parameter Space

Inefficient Methods

Lack of reusable framework

Solution

Hyperparameter optimisation (HPO)

Specialized Algorithm

Probe and solve algorithm

Two-Phase Approach for Optimizing hyper parameters

Probing Phase

Explores various configurations using HPO methods, ranking them based on performance within a (K percent) limited time.

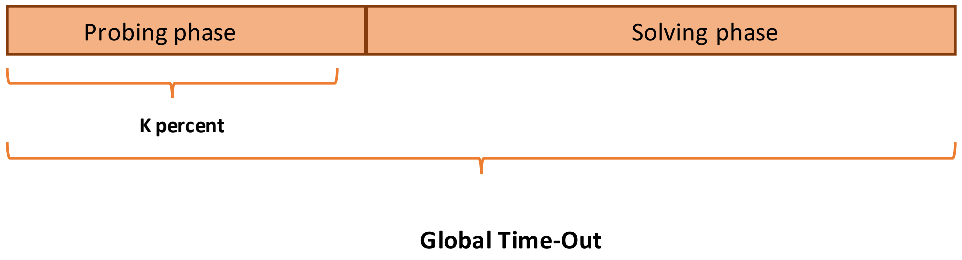
Solving Phase

Utilizes the top-ranked configuration from the probing phase to solve the constraint problem.

Flexibility

Algorithm adapts dynamically based on problem complexity and solver performance, enhancing efficiency.

Adaptive PSA



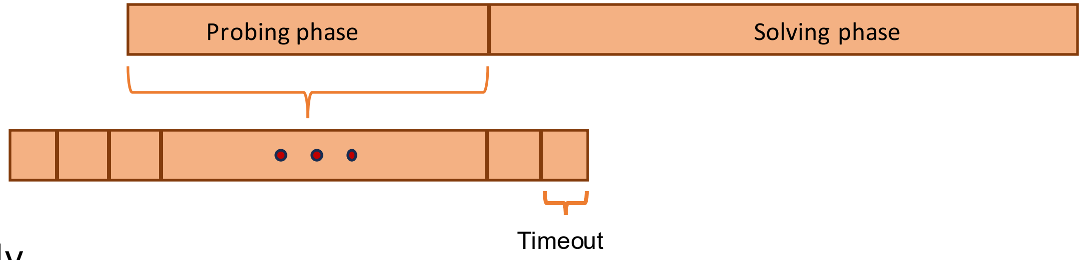
✓ Probing Timeout Allocation

- ***Static Allocation:*** Reserve fixed % of total timeout (e.g., 20%)
- ***Maximum Iterations:*** Limit rounds so total time isn't exceeded
- ***Dynamic Stopping:*** Stop if no improvement after N rounds

Adaptive PSA

✓ Probing Timeout Allocation

- ***Static Allocation:*** Reserve fixed % of total timeout (e.g., 20%)
- ***Maximum Iterations:*** Limit rounds so total time isn't exceeded
- ***Dynamic Stopping:*** Stop if no improvement after N rounds



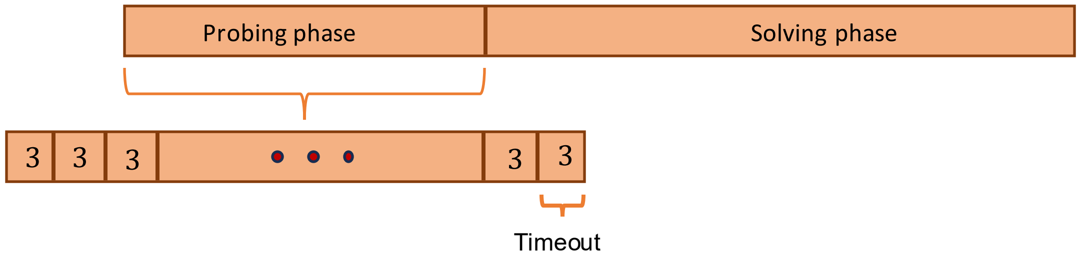
✓ Timeout Initialization Strategies

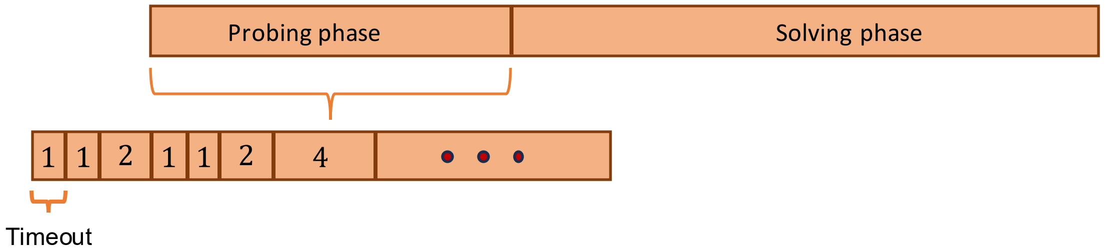
- ***Fixed-Time Start:*** Use fixed timeout for every round (e.g., 5s)
- ***Baseline-Based Start:*** Estimate baseline → adjust timeout adaptively
- ***No Timeout:*** Run until solution found or stopped manually

Adaptive PSA

✓ Timeout Evolution Patterns (Static vs Dynamic Timeout Patterns)

- ***Flat Timeout:*** Same timeout for all rounds (3–5s)
- ***Progressive Increase:*** Timeout increases after failure
  - ***Luby Sequence:*** Structured increase (1,1,2,1,1,2,4...)
  - ***Geometric Multiplication:*** Multiply timeout (e.g., ×1.5 each failure)





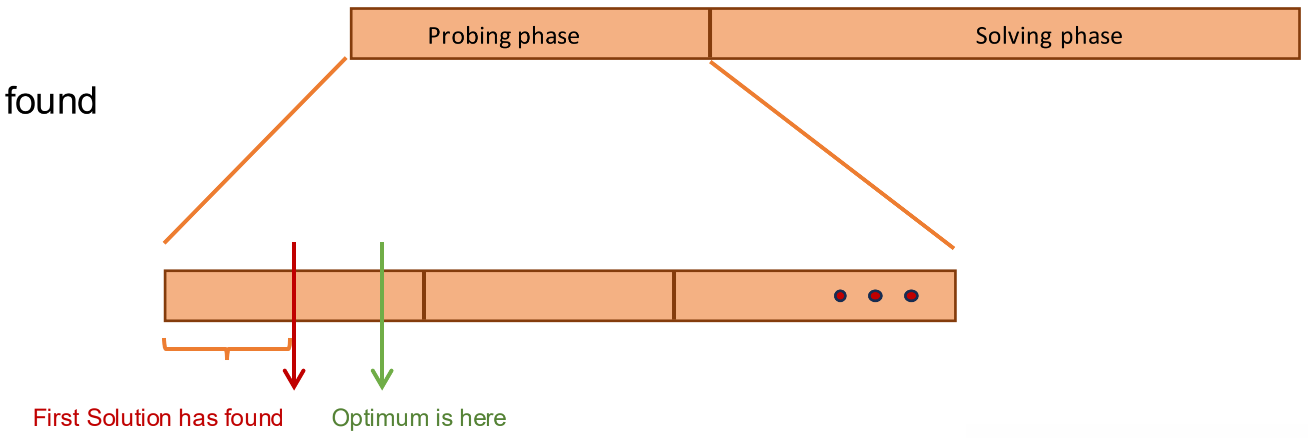
Adaptive PSA

✓ Timeout Evolution Patterns (Static vs Dynamic Timeout Patterns)

- ***Flat Timeout:*** Same timeout for all rounds (3–5s)
- ***Progressive Increase:*** Timeout increases after failure
  - ***Luby Sequence:*** Structured increase (1,1,2,1,1,2,4...)
  - ***Geometric Multiplication:*** Multiply timeout (e.g., ×1.5 each failure)

✓ Solver Stop Conditions

- ***First Solution Found:*** Stop as soon as one solution is found
- ***Timeout Reached:*** Stop when time limit exceeded



Research Project: Problem Statement

Problem	Solution
Large Parameter Space	Hyperparameter optimisation (HPO)
Inefficient Methods	Specialized Algorithm
Lack of reusable framework	Integration via Generic Library and Validation with Widely used Solvers

Adaptive PSA

✓ Timeout Ev

- *Fl*
- *Pr*

✓ Solver S

- *Fi*
- *Tin*

❖ Integrating PSA into Practice

- To apply PSA effectively, we needed a flexible, solver-independent platform
- We chose **CPMPY**: a modelling library in Python that connects to multiple solvers
- This allows PSA to work across solvers like OR-Tools and ACE with no extra setup

Result

- **Approaches Compared:**
  - PSA enhanced by Bayesian Optimization (HPO method).
  - PSA enhanced by Hamming Distance (HPO method).

- **Implementation Framework:**
  - All experiments conducted using CPMPY.
  - Utilized ACE solver .

- **Key Evaluation Metrics:**
  - **Objective Value:** The best objective function value achieved across all compared methods.
  - **Time to Best:** Time (in seconds) taken by each method to reach the overall best objective value.
  - **Consistency:** Number of problem instances where each specific method successfully found the best objective value.

Result

	Bayesian (PSA)		Hamming (PSA)		Result
	Objective	Runtime (s)	Objective	Runtime (s)	Winner
Instance					
Benzenoide-06	**64**	**1.33**	64	1.55	BO (Tiebreak)
Hsp-10405	**198**	**0.63**	198	0.89	BO (Tiebreak)
Hsp-aux-10405	**198**	**0.64**	198	0.68	BO (Tiebreak)
KidneyExchange-4-041	**1**	**1.07**	1	2.15	BO (Tiebreak)
KidneyExchange-4-051	**1**	**1.69**	1	2.54	BO (Tiebreak)
KidneyExchange-4-061	**11**	**3.03**	11	480.55	BO (Tiebreak)
ProgressiveParty-rally-red12-05	**5**	**0.75**	5	1.12	BO (Tiebreak)
ProgressiveParty-rally-red12-07	**7**	**1.10**	7	1.23	BO (Tiebreak)
ProgressiveParty-rally-red12-09	**9**	**0.73**	9	0.99	BO (Tiebreak)
RIP-25-0-j060-01-01	**187**	**0.97**	187	1.98	BO (Tiebreak)
RIP-25-2-j060-20-01	**118**	**0.96**	118	480.54	BO (Tiebreak)
SREFLP-Cl07	**1,590**	**1.10**	1,590	1.65	BO (Tiebreak)
Sonet-s2ring02	**14**	**0.55**	14	4.57	BO (Tiebreak)
Sonet-s3ring09	22	480.45	**22**	**480.36**	HD+PSA (Tiebreak)
Sonet-s3ring13	21	480.38	**21**	**480.36**	HD+PSA (Tiebreak)
Sonet-s3ring14	23	480.39	**23**	**480.35**	HD+PSA (Tiebreak)
TSPTW2-n040w020-1	FAIL	480.41	FAIL	480.40	All Failed

Goals for 2025: Research Goals

- **Publish 2 papers:**
    1. **Constraints Journal**: “Improving Efficiency of Constraint Programming Solver by Focusing on Hyper-parameters”
    2. **A good Conference**: “Integrating Construction Standards into Constraint Programming: A Case Study on Energy Efficiency”
- **Managing the 4th CET.**
- **Synthesize PhD thesis for completion and defense.**

Thank you

Hedieh.Haddad@uni.lu