Metrics for evaluating the performance and complexity of computer programs

Rares Folea

advisor: Prof. Emil-Ioan Slusanschi

Faculty of Automatic Control and Computers
University Politehnica of Bucharest

Contents

“

Complexity of an algorithm is a measure of the amount of time and/or space required by an algorithm for an input of a given size.”

Patrick Prosser
Glasgow University

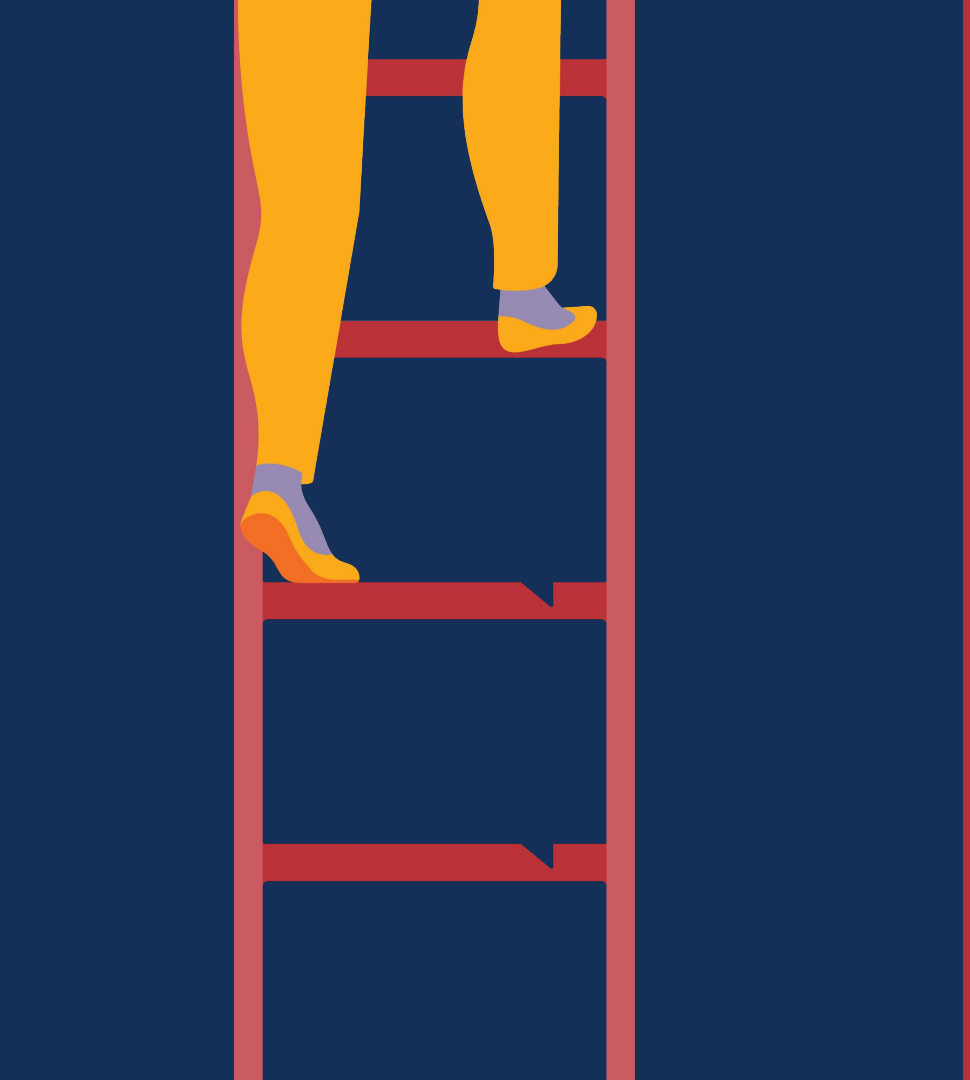
Shortcomings of the classical model

Lack of sensitivity wrt. the total execution time for the base-case

Architecture - insensitive metric

Relatively few complexity classes wrt. the number of algorithms clustered

Insensitive metric to “galactic” algorithms



A refined complexity calculus model:

r-Complexity

A new asymptotic notations that offer better complexity feedback for similar programs, providing subtle insights even for algorithms that are part of the same *conventional* complexity class

Adjusting the *Bachmann–Landau* notations for r-Complexity Calculus (1)

Big r-Theta notation

This set defines the group of mathematical functions similar in magnitude with *g* in the study of asymptotic behavior. *A set-based description of this group can be expressed as:*

$$\Theta_r(g(n)) = \{f \in \mathcal{F} \mid \forall c_1, c_2 \in \mathbb{R}_+^* \ s.t. c_1 < r < c_2, \exists n_0 \in \mathbb{N}^* \\ s.t. \ \ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \ , \ \forall n \geq n_0\}$$

Adjusting the *Bachmann–Landau* notations for r-Complexity Calculus (2)

Big r-O notation

$$\mathcal{O}_r(g(n)) = \{f \in \mathcal{F} \mid \forall c \in \mathbb{R}_+^* \ s.t. \ r < c, \exists n_0 \in \mathbb{N}^* \ s.t. \ f(n) \leq c \cdot g(n), \ \forall n \geq n_0\}$$

Big r-Omega notation

$$\Omega_r(g(n)) = \{f \in \mathcal{F} \mid \forall c \in \mathbb{R}_+^* \ s.t. \ c < r, \exists n_0 \in \mathbb{N}^* \ s.t. \ f(n) \geq c \cdot g(n), \ \forall n \geq n_0\}$$

New metrics

RM1, RM2, ERM1, ERM2

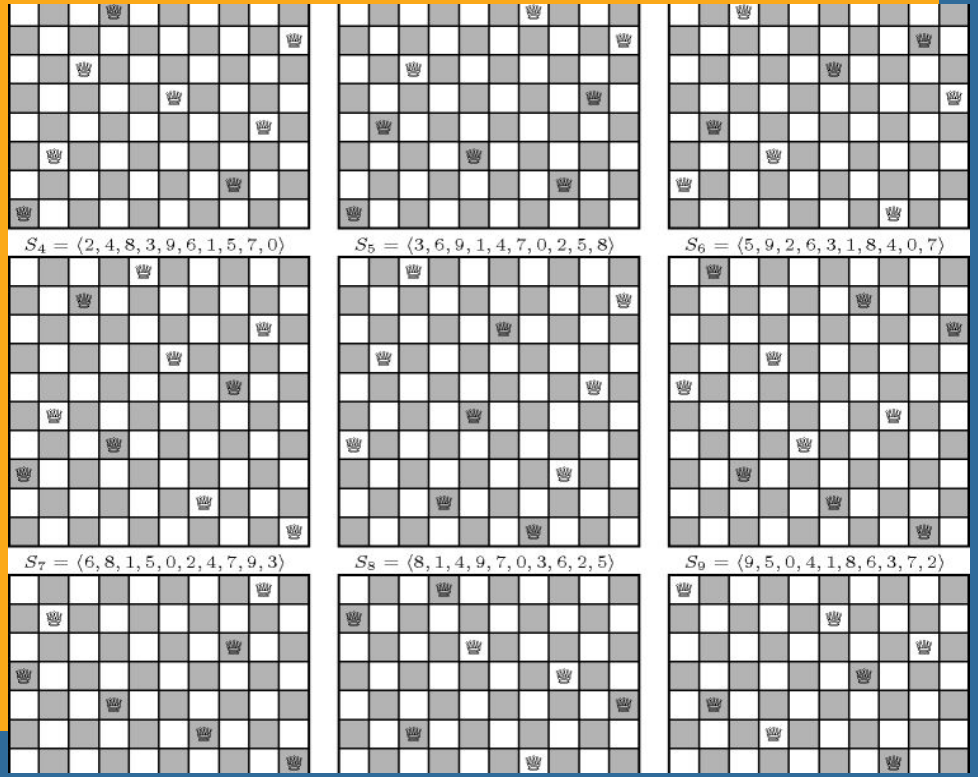
An enhanced metric (ERM2) for time estimation based on *Mean-Value Theorem (Lagrange)* using complexity function in Normalized r-Complexity model:

$$T(n_{min}, n_{max}) = \sum_{k=0}^{f-1} p_k \cdot \int_{n_k}^{n_{k+1}} g_1(n) dn$$

p_i is the weight associated with the probability of the input to be bounded in the interval $[n_i, n_{i+1}]$

r-Complexity: in practice

- **Human-driven** calculus of r-Complexity
- Estimation for algorithms with **known** *Bachmann–Landau Complexity*
- Estimation for algorithms with **unknown** *Bachmann–Landau Complexity*



N-Queens Problem

***Human-driven** calculus of r-Complexity*

Architecture dependent calculus:

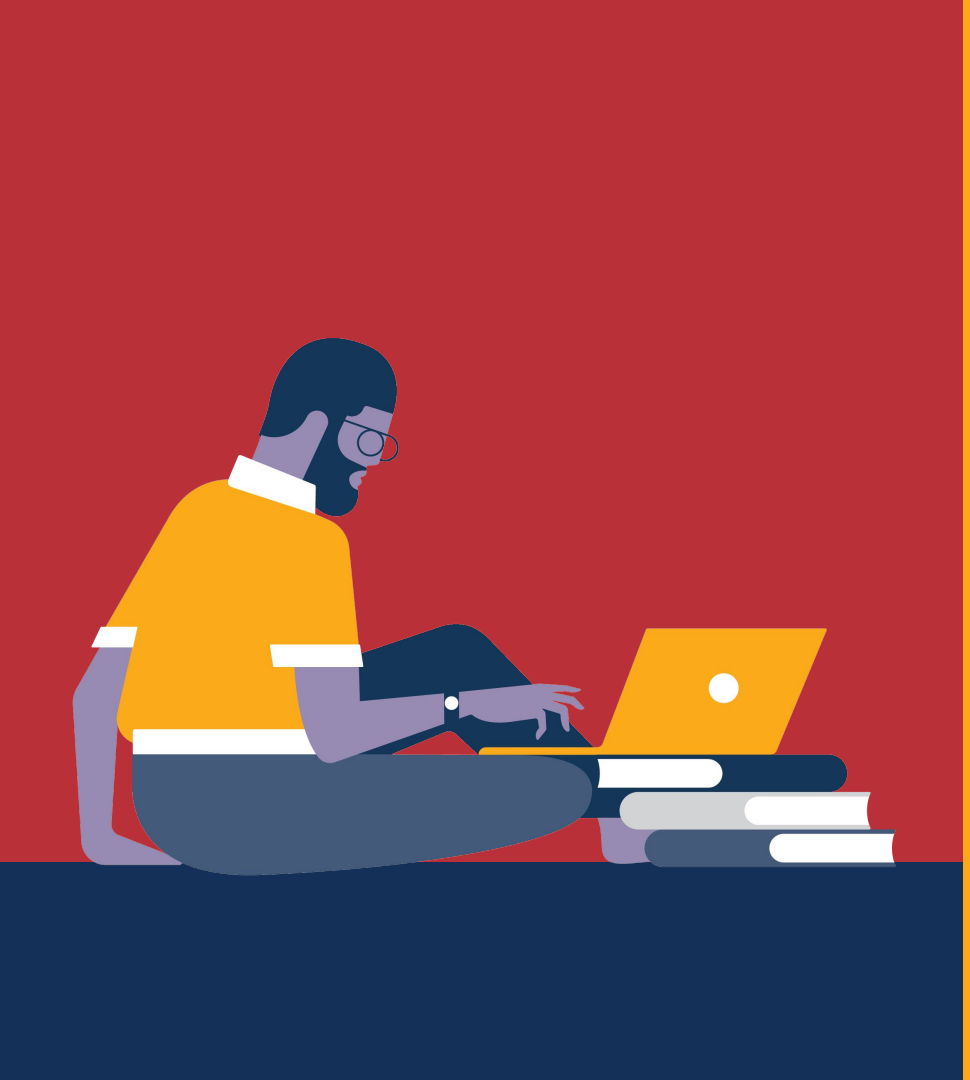
(Core i7, x86-64 instruction set)

$$QueenProblem(n) = O_1(408 \cdot n^3 \cdot \int_0^\infty s^{n-1} e^{-s} ds)$$

N-Queens Problem time estimation

based on the associated r-complexity Class

0.013s	Total time for **n=8**
1 day	Total time for **n=16**
10M centuries	Total time for **n=32**
50x	**Speedup C** vs **Python**

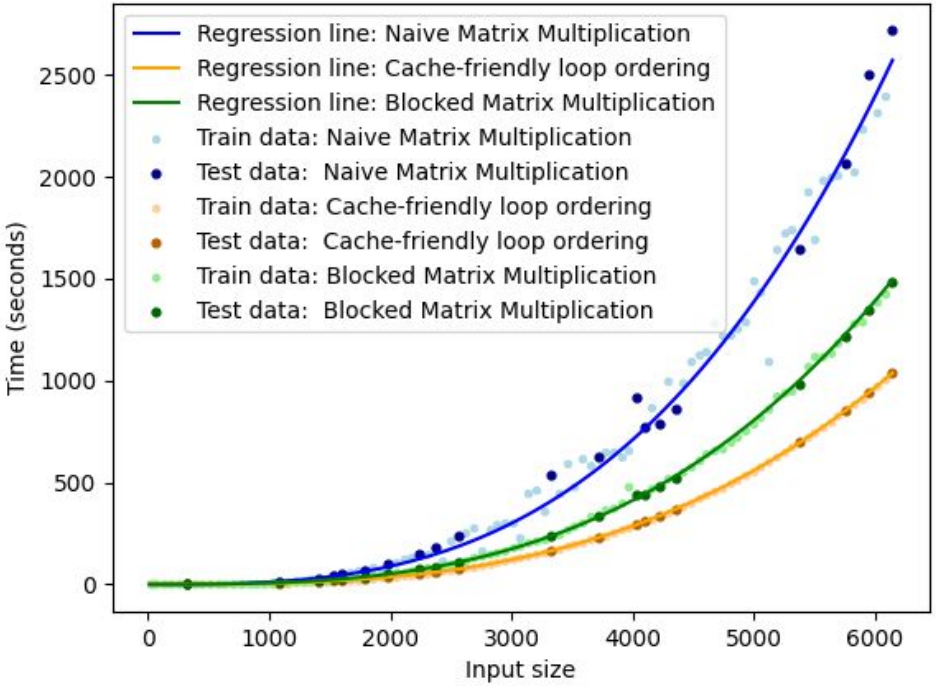


Benchmark:

Matrix multiplication

Automatic ML-based estimator for algorithms with known *Bachmann–Landau Complexity* : TRAINING

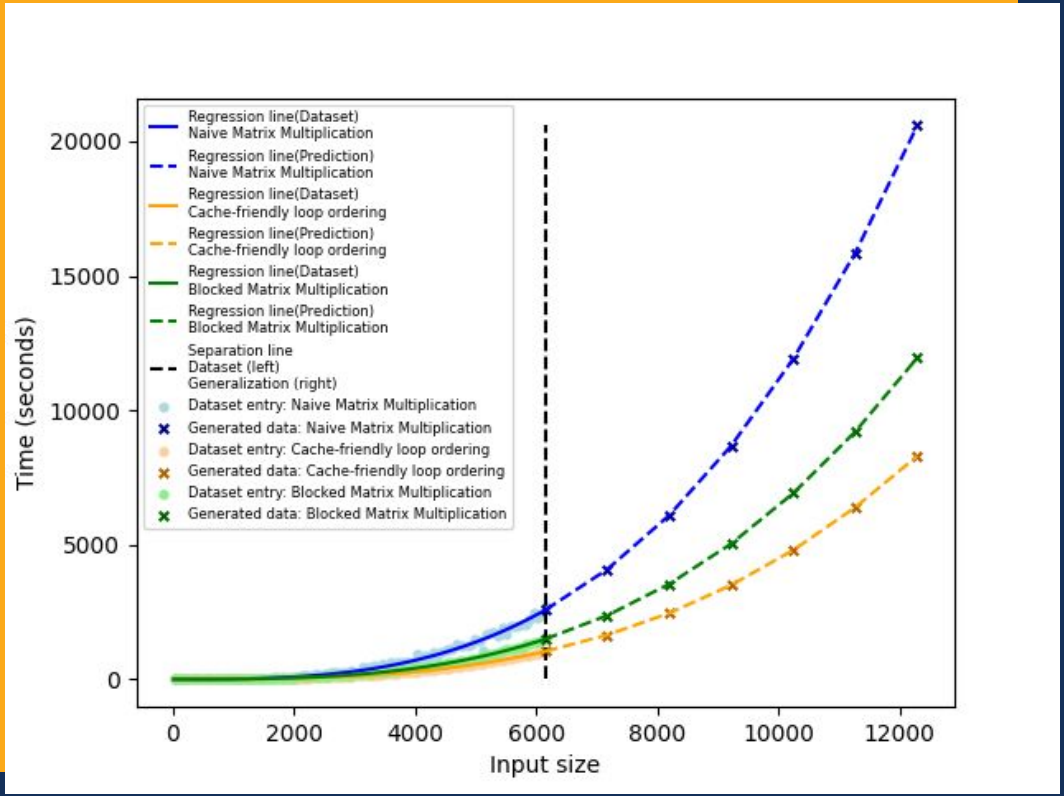
Step 1:
TRAINING

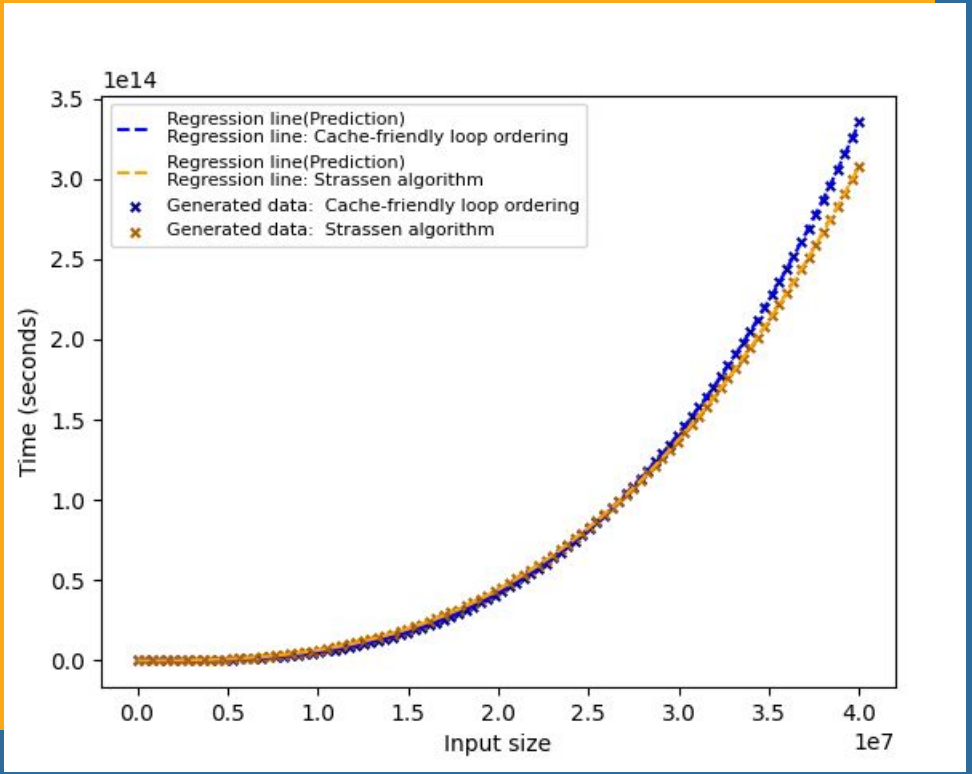


Automatic ML-based estimator for algorithms with known *Bachmann–Landau Complexity* : PREDICTING

Step 2: PREDICTING

(based on r-Complexity model)





$$O_{(n^{2.7})} \ v.s. \ O_{(n^3)}$$

Strassen Algorithm

against Naive Multiplication

Architecture dependent

calculus:

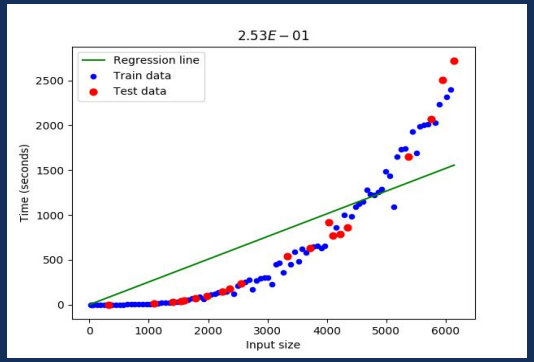
(Core i7, x86-64 instruction set)

Strassen Algorithm - Statistics

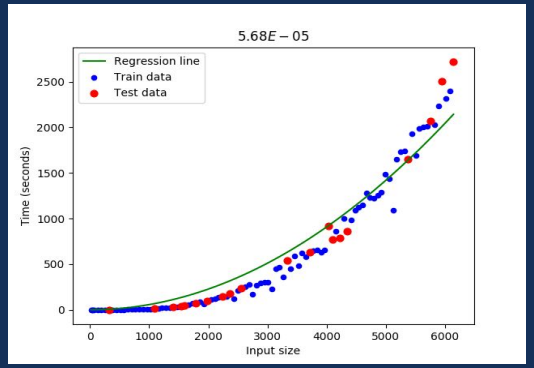
against Naive Multiplication

25 million	25 762 centuries	7.43 ZB	5M centuries
Critical input size to make Strassen algorithm perform better than Naive	Total execution time (for 25Mx25M matrices)	Peak RAM usage of the Strassen algorithm (for 25Mx25M matrices)	Equivalent total time of 1080p digital video content (for 25Mx25M matrices)

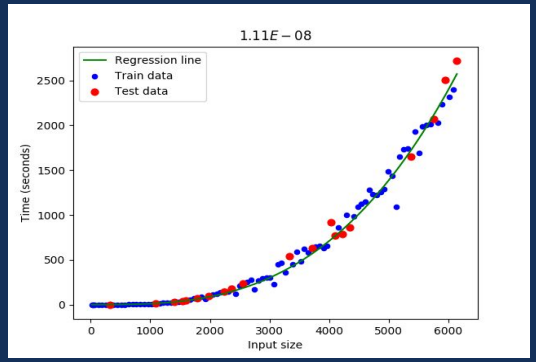
Automatic ML-based estimator for algorithms with unknown *Bachmann–Landau Complexity*



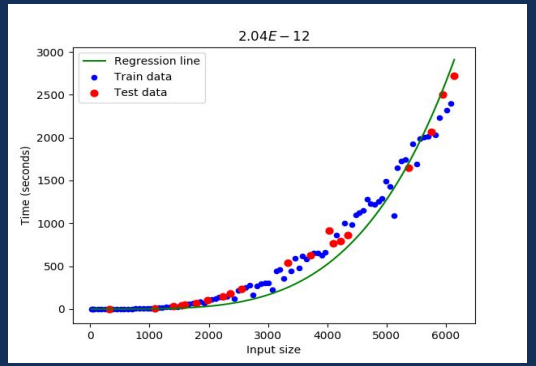
Linear



Quadratic



Cubic

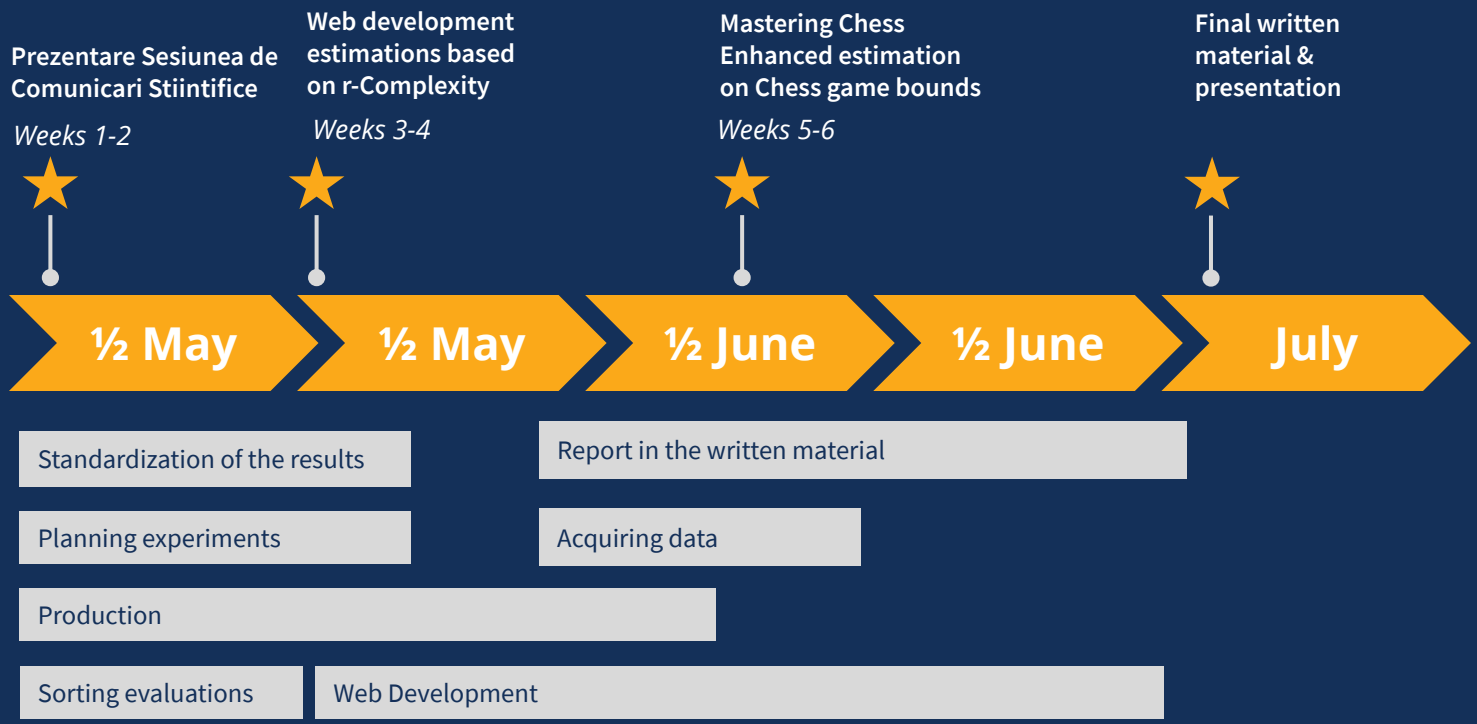


Quartic

Takeaways

1. r-Complexity represents an enhancement that provides better sensitivity wrt. discrete analysis
2. Architecture-sensitive metric
3. r-Complexity is also susceptible to automatic discovery

What’s next


Prezentare Sesiunea de Comunicari Stiintifice
Weeks 1-2
Web development estimations based on r-Complexity
Weeks 3-4
Mastering Chess Enhanced estimation on Chess game bounds
Weeks 5-6
Final written material & presentation
½ May
½ May
½ June
½ June
July
Standardization of the results
Planning experiments
Production
Sorting evaluations
Web Development
Report in the written material
Acquiring data

Thank you!

Follow r-Complexity development:
https://github.com/raresraf/rafMetrics/