

Documentație Simulare Monte Carlo pentru boți de Catan

Niță Alexandros, Pascaru Alex, Glod Cosmin

January 2025

Contents

1	Prezentare proiect	2
1.1	Strategii Analizate	2
1.2	Metodologie	2
1.3	Rezultate Asteptate	2
2	Explicatie algoritmi	3
2.1	Generarea datelor utilizand Catanatron	3
2.2	Procesarea si extragerea datelor	3
2.3	Simularea Monte Carlo	3
2.4	Vizualizarea rezultatelor	4
2.5	Concluzii	4
3	Justificare teoretică	4
4	Rezultate si grafice	5
4.1	Random Bot vs Random Bot	5
4.2	Weighted Random Bot vs Weighted Random Bot	6
4.3	Victory Point Bot vs Victory Point Bot	7
4.4	Strategiile combinate	7
4.4.1	Random Bot vs Weighted Random Bot	8
4.4.2	Random Bot vs Victory Point Bot	8
4.4.3	Victory Point Bot vs Weighted Random Bot	9
4.5	Bonus: AP vs R vs W vs VP	9
5	Referinte	10

1 Prezentare proiect

Obiectivul acestui proiect este de a simula, utilizand metoda Monte Carlo, cate runde sunt necesare pentru finalizarea unui joc de Catan, in functie de diferite strategii aplicate de jucatori. Vom analiza mai multe abordari strategice si le vom compara pentru a intelege impactul lor asupra lungimii jocului.

1.1 Strategii Analizate

Strategiile analizate includ urmatoarele:

1. **Alegerea aleatorie:** La fiecare pas, deciziile sunt luate in mod randomizat, fara o strategie bine definita. Acest lucru este realizat de *Random Bot (R)*, care selecteaza aleatoriu actiunile disponibile.
2. **Prioritizarea constructiei:** Jucatorii construiesc imediat ce au resursele necesare, favorizand expansiunea rapida. Acest comportament este specific *Weighted Random Bot (W)*, care foloseste ponderi pentru a favoriza actiunile legate de expansiune.
3. **Maximizarea punctelor de victorie:** Deciziile sunt luate pentru a obtine puncte de victorie cat mai rapid, cu accent pe obiective precum cartile de dezvoltare, construirea oraselor sau monopolizarea rutelor. Aceasta strategie este aplicata de *Victory Point Bot (VP)*.
4. **Strategia botului avansat:** *Alpha-Beta Pruning Bot (AP)* utilizeaza un algoritm de tip *alpha-beta pruning*, care ii permite sa anticipeze mutarile adversarilor si sa ia decizii optime pe termen lung. Aceasta strategie este cea mai complexa dintre cele analizate.

1.2 Metodologie

Metoda Monte Carlo va permite rularea unui numar mare de jocuri simulate pentru fiecare strategie, generand statistici precise asupra:

- Duratei medii a jocurilor,
- Variatiei in numarul de runde,
- Distributiei numarului de runde.

Astfel, vom obtine o imagine detaliata asupra eficientei fiecarei abordari.

1.3 Rezultate Asteptate

Rezultatele acestui studiu vor oferi perspective utile asupra dinamicii jocului si vor putea fi folosite pentru a intelege mai bine modul in care strategiile influenteaza durata unui joc de Catan.

2 Explicatie algoritmi

2.1 Generarea datelor utilizand Catanatron

Pentru generarea datelor, am utilizat **Catanatron**, un proiect open-source care implementeaza boti pentru jocul Catan si ofera o mapare completa a jocului. Catanatron permite simularea meciurilor intre boti, folosind strategii variate pentru analiza dinamicii jocului. Datele generate de acest proiect au fost exportate in format JSON pentru procesare ulterioara.

Am simulat doua tipuri principale de meciuri:

- **Meciuri 1v1 intre boti de acelasi tip:** Testam performanta aceleiasi strategii intr-un joc izolat.
- **Meciuri mixte intre tipuri diferite de boti:** Analizam comportamentul diverselor strategii in confruntari directe.

Am folosit un seturi de 10000 de meciuri pentru fiecare categorie de boti.

2.2 Procesarea si extragerea datelor

Datele JSON contin detalii despre fiecare actiune dintr-un meci, inclusiv runde, tipuri de actiuni si decizii. Pentru analiza noastra, ne-am concentrat pe identificarea numarului de runde, reprezentate prin actiuni de tip **ROLL**.

- Functia `get_roll_counts` parcurge fisierele JSON si numara actiunile **ROLL** din fiecare meci. Acestea sunt utilizate pentru a determina durata totala a jocului in termeni de runde.
- Datele procesate sunt salvate intr-un format compact (`.pkl`) pentru eficienta si reutilizare.

Pentru a mentine relevanta statistica, am filtrat jocurile care dureaza mai mult de 400 de runde, observand ca orice era peste acest prag era in realitate o eroare de program, aproximativ 1% din numarul total.

2.3 Simularea Monte Carlo

Datele extrase sunt utilizate pentru a rula simulari Monte Carlo, oferind o estimare a distributiei duratei jocurilor. Simularea presupune:

- Selectarea aleatorie a submultimilor de date din meciurile prelucrate.
- Calculul statisticilor relevante: minim, maxim, medie si deviatie standard pentru durata jocurilor.
- Numarul de simulari este calculat corespunzator formulei Chernoff-Hoeffding pentru fiecare set
- Generarea unei histograme a distributiei rundelor, pentru a vizualiza clar frecventa si variatia duratei.

Simularile sunt efectuate in loturi mari (*batch-uri*) pentru a reduce costurile de procesare. Rezultatele sunt apoi agregate pentru a obtine distributia globala a datelor.

2.4 Vizualizarea rezultatelor

Rezultatele simulării sunt prezentate sub forma de histograme, care evidentiază:

- **Distributia duratei jocurilor:** Intervalul tipic de runde pentru fiecare tip de bot sau meci.
- **Intervalul de incredere ($\pm 2\sigma$):** Estimarea intervalului în care se afla 95% dintre jocuri.
- **Media si dispersia:** Informatii centrale despre durata medie si variabilitatea jocurilor.

2.5 Concluzii

Acest flux de lucru, combinand datele generate de **Catanatron** si analiza prin metoda Monte Carlo, permite o evaluare precisa a duratei jocurilor de Catan pentru diverse strategii si tipuri de boti.

3 Justificare teoretică

Pentru a determina numărul necesar de simulări pentru fiecare tip de joc, va trebui să aplicăm inegalitatea Chernoff-Hoeffding. Enunțul acestei inegalități este:

$$\mathbb{P}(|\overline{S}_n - \mu| \geq t) \leq 2 \exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$

unde $\overline{S}_n = \frac{X_1 + X_2 + \dots + X_n}{n}$, cu X_1, X_2, \dots, X_n variabile aleatoare independente, identic distribuite și mărginite astfel: $a \leq X_i \leq b$, $\mu = \mathbb{E}[X_1]$, iar t este eroarea. Introducem o variabilă ce reprezintă nivelul de încredere pe care vrem să îl avem în model α . Astfel, avem

$$\mathbb{P}(|\overline{S}_n - \mu| < t) \geq 1 - 2 \exp\left(-\frac{2nt^2}{(b-a)^2}\right) \geq \alpha$$

Din această relația, vom obține o valoare minimă pentru n , adică variabila ce reprezintă numărul de simulări. Avem

$$\exp\left(-\frac{2nt^2}{(b-a)^2}\right) \leq \frac{1-\alpha}{2} \implies n \geq -\frac{(b-a)^2}{2t^2} \ln\left(\frac{1-\alpha}{2}\right)$$

Pentru proiectul nostru, am ales $t = 0.01$, $\alpha = 0.99$. Pentru fiecare tip de meci, vom calcula limita inferioară și cea superioară. Valoarea minimă a minimelor este $a = 20$ și valoarea maximă a maximelor $b = 400$. Un mic calcul ne conduce la un număr maxim de $n \geq 3825385139$.

4 Rezultate si grafice

In urmatoarele grafice vom observa si vom analiza rezultatele. Partea verde din imagini reprezinta **regula empirica** care ne spune ca 95% din data se afla la 2 deviatii standard fata de medie.

4.1 Random Bot vs Random Bot

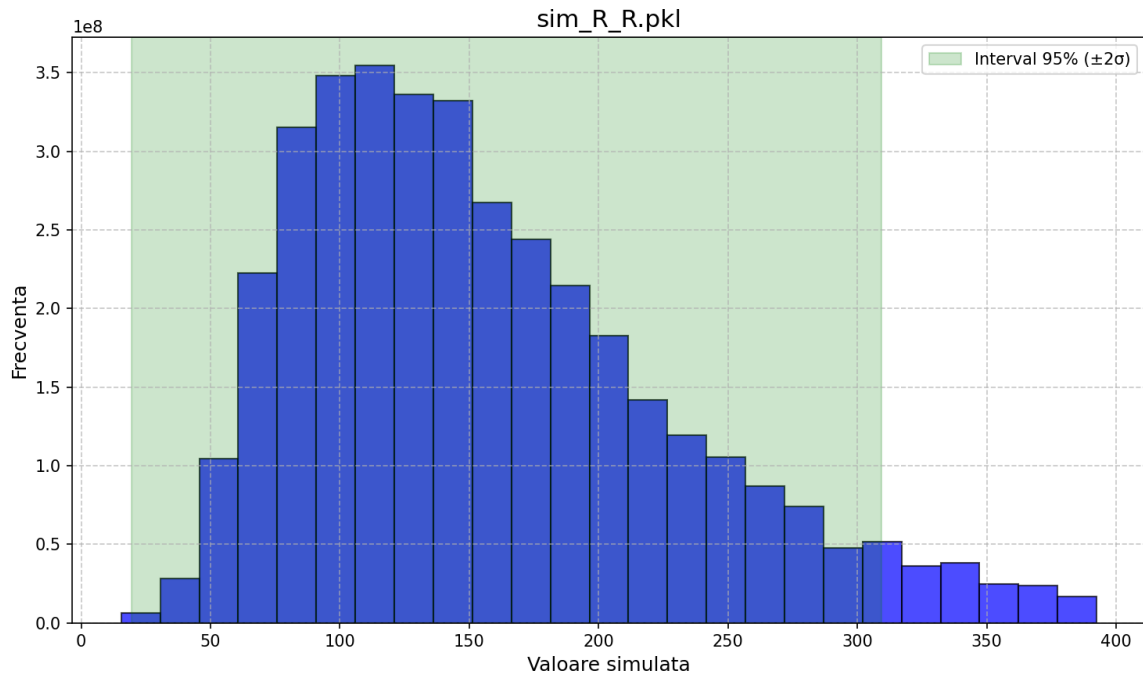


Figure 1: R vs R

Performanta strategiei a Random Bot, adica de alegere aleatoare tinde sa fie haotica si impredictibila din cauza lipsei unei strategii coerente, ceea ce duce la o variabilitate ridicata a numarului de runde. Cu toate acestea, 95% din jocuri se finalizeaza intr-un interval rezonabil.

4.2 Weighted Random Bot vs Weighted Random Bot

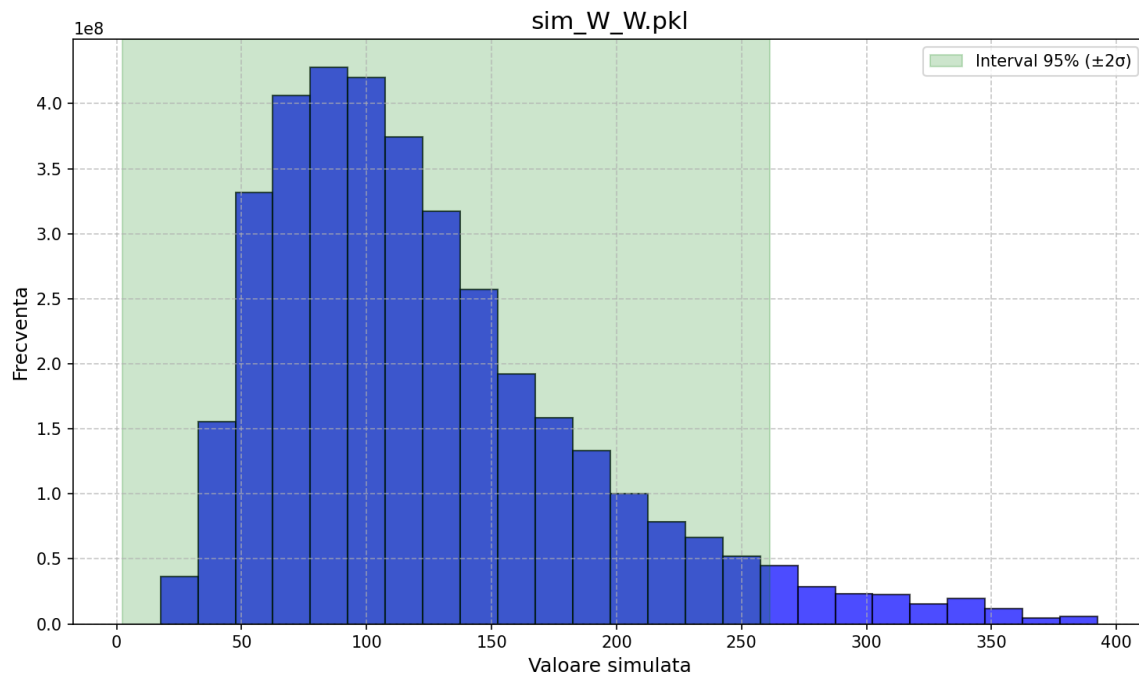


Figure 2: W vs W

Strategia de prioritizare a constructiei utilizata de Weighted Random Bot duce la o performanta mai predictibila si eficienta comparativ cu strategia complet aleatoare. Aceasta abordare favorizeaza o finalizare mai rapida a jocurilor, limitand variabilitatea si reducand numarul de jocuri extrem de lungi. Se pare ca axarea pe constructie are un impact relevant asupra dinamicii jocului, oferind un echilibru intre aleatoriu si decizii directionate.

4.3 Victory Point Bot vs Victory Point Bot

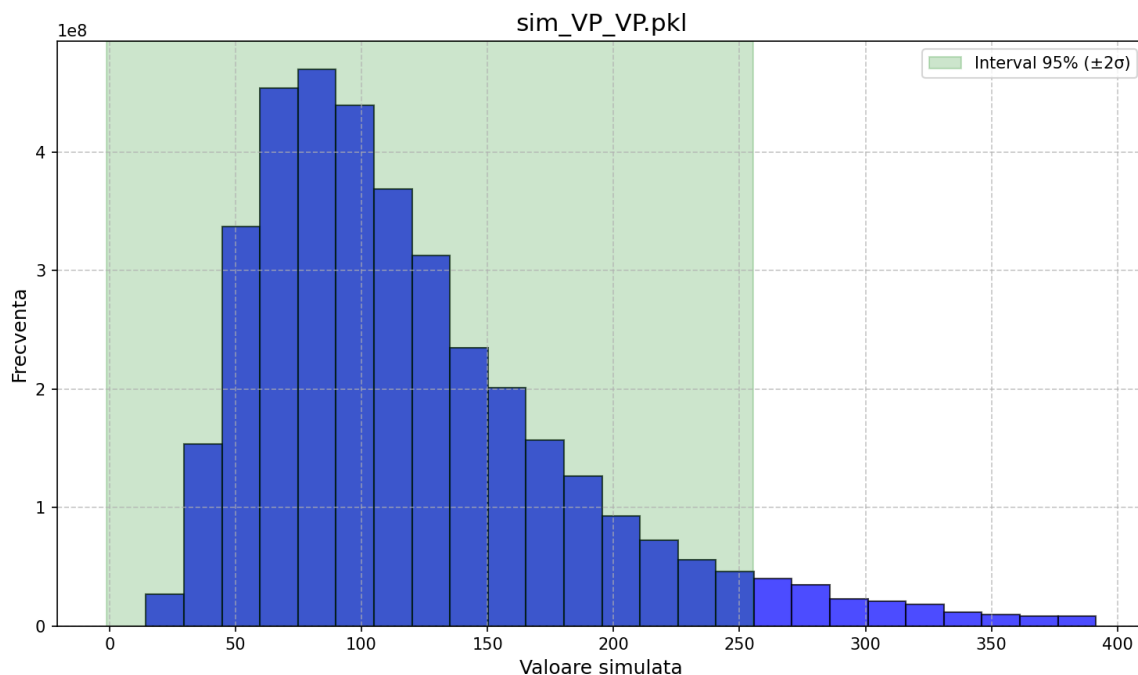


Figure 3: VP vs VP

Strategia de maximizare a punctelor de victorie, aplicata de Victory Point Bot, demonstreaza o eficienta superioara si o consistenta ridicata in finalizarea jocurilor. Focalizarea pe obiective clare si orientate spre puncte reduce variabilitatea si optimizeaza durata jocului.

4.4 Strategiile combinate

Observam ca atunci cand punem strategiile sa joace 1v1, numarul de runde este influentat direct de complexitatea si eficienta strategiilor implicate. Strategiile mai bine definite, cum ar fi Victory Point Bot, duc la un numar semnificativ mai mic de runde, in timp ce strategiile mai aleatorii, cum ar fi Random Bot, genereaza jocuri mai lungi si mai variabile.

4.4.1 Random Bot vs Weighted Random Bot

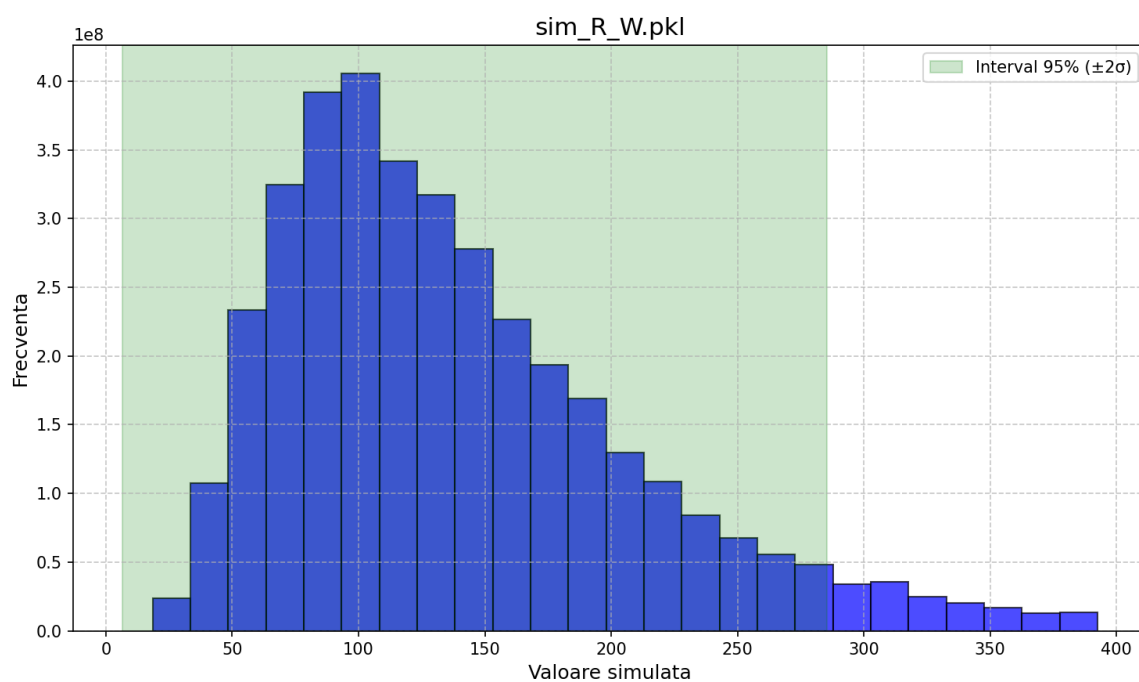


Figure 4: R vs W

4.4.2 Random Bot vs Victory Point Bot

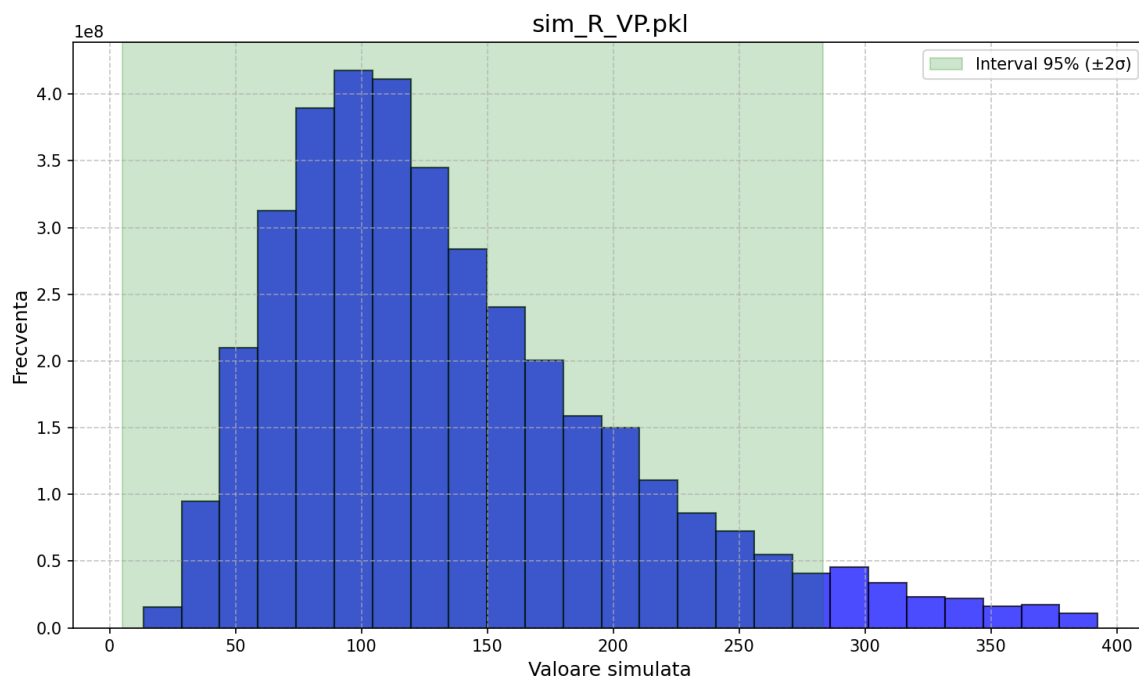


Figure 5: R vs VP

4.4.3 Victory Point Bot vs Weighted Random Bot

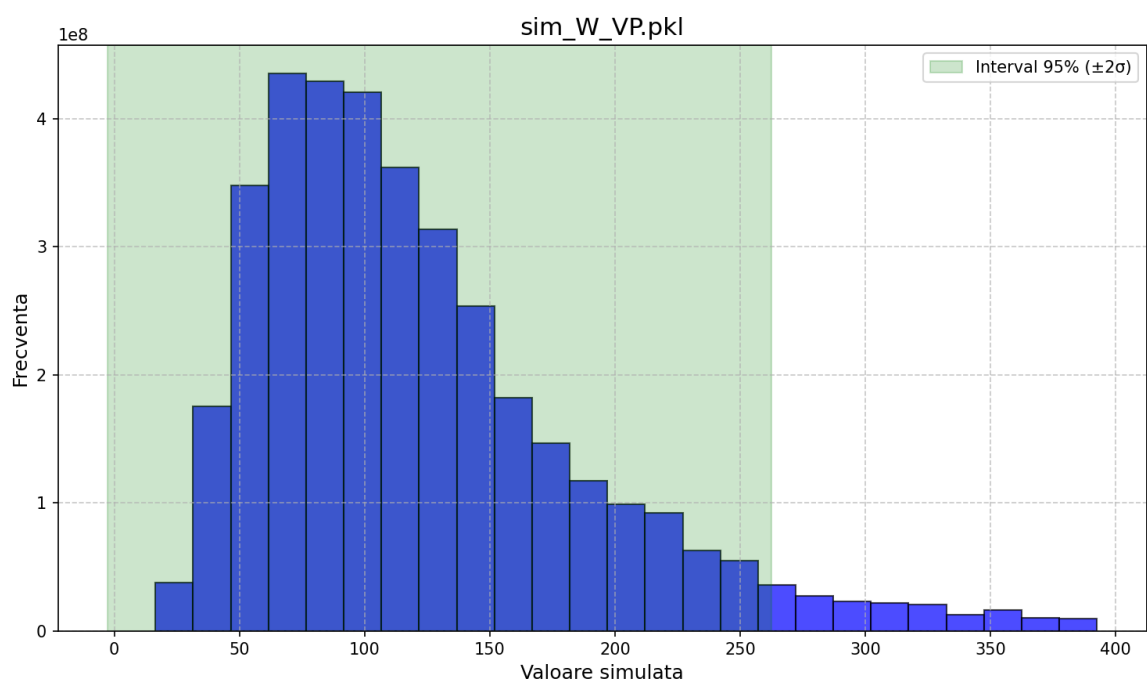


Figure 6: VP vs W

4.5 Bonus: AP vs R vs W vs VP

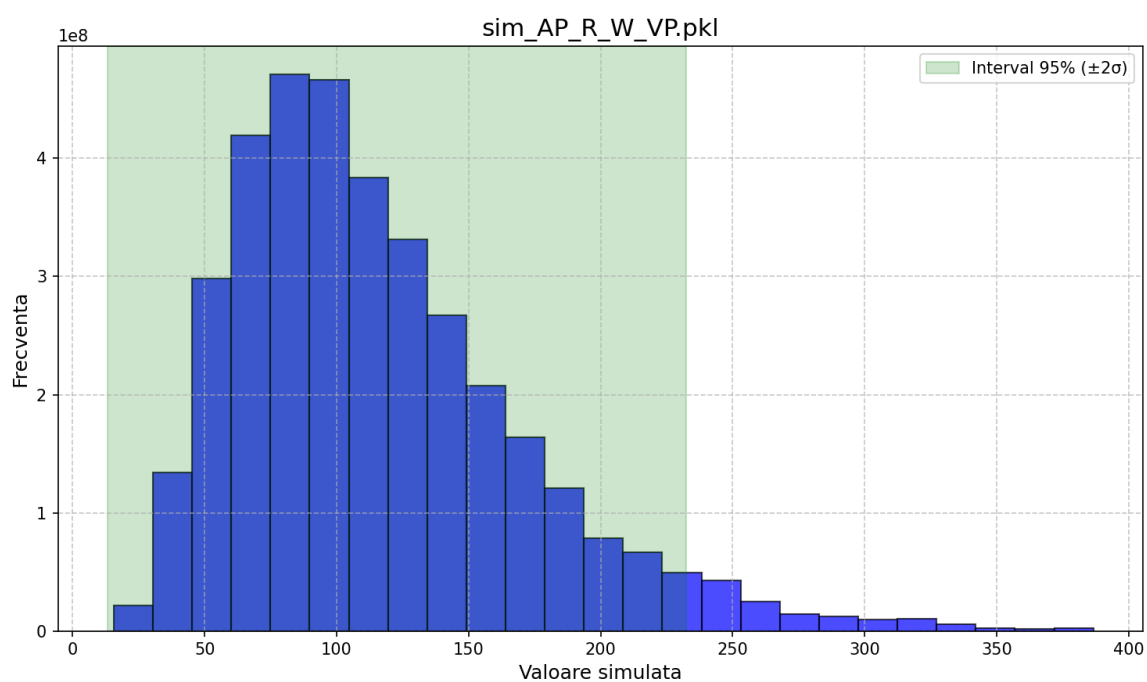


Figure 7: AP dominance

Observam din grafic ca, atunci cand botul Alpha-Beta Pruning (AP) este implicat, acesta domina toate meciurile, iar distributia numarului de runde este influentata de interactiunile dintre AP si celelalte strategii. Desi AP castiga toate meciurile, jocurile prezinta o variabilitate mai mica, fiind mai concentrate in intervalul 70-150 de runde. Aceasta sugereaza ca deciziile optimizate pe termen lung ale botului AP reduc durata jocurilor si limiteaza variabilitatea, facandu-l cea mai eficienta strategie din simulare.

5 Referinte

- [Catanatron](#)
- Cursul de Probabilitati & Statistica 2024-2025, Mihai Bucataru
- [6. Monte Carlo Simulation - MIT OpenCourseWare](#)
- [7. Confidence Intervals - MIT OpenCourseWare](#)
- [But what is the Central Limit Theorem?](#)
- [Why \$\pi\$ is in the normal distribution \(beyond integral tricks\)](#)