

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

**FACULTATEA DE INFORMATICĂ**



LUCRARE DE LICENȚĂ

**Tranzacționare evolutivă: Optimizarea și  
meta-optimizarea strategiilor cu medii mobile**

propusă de

**Șerban-Constantin Chișca**

**Sesiunea: iulie, 2025**

Coordonator științific

**Lect. Dr. Croitoru Eugen**

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

**FACULTATEA DE INFORMATICĂ**

**Tranzacționare evolutivă: Optimizarea  
și meta-optimizarea strategiilor cu  
medii mobile**

**Șerban-Constantin Chișca**

**Sesiunea: iulie, 2025**

Coordonator științific

**Lect. Dr. Croitoru Eugen**

Avizat,  
Îndrumător lucrare de licență,  
Lect. Dr. Croitoru Eugen.

Data: ..... Semnătura: .....

### **Declarație privind originalitatea conținutului lucrării de licență**

Subsemnatul **Chișca Șerban-Constantin** domiciliat în **România, jud. Iași, mun. Iași, Strada Profesor Neculai Zamfirescu nr 13**, născut la data de **13 Iunie 2000**, identificat prin CNP **5000613226755**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2025, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Tranzacționare evolutivă: Optimizarea și meta-optimizarea strategiilor cu medii mobile** elaborată sub îndrumarea domnului **Lect. Dr. Croitoru Eugen**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: .....

Semnătura: .....

## Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Tranzacționare evolutivă: Optimizarea și meta-optimizarea strategiilor cu medii mobile**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea "Alexandru-Ioan Cuza" din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Șerban-Constantin Chișca**

Data: .....

Semnătura: .....

# Cuprins

<b>Abstract</b>	<b>2</b>
<b>Glosar de termeni</b>	<b>3</b>
<b>1 Introducere</b>	<b>5</b>
1.1 Motivație . . . . .	6
1.2 Contribuții . . . . .	6
<b>2 Metode de ultima generație</b>	<b>7</b>
2.1 Introducere . . . . .	7
2.2 Analiza metodelor evolutive în optimizare . . . . .	7
2.2.1 Optimizarea indicatorilor tehnici . . . . .	8
2.2.2 Modele de predicție . . . . .	8
2.2.3 Rețele evolutive . . . . .	9
2.2.4 Optimizarea bayesiană . . . . .	9
2.2.5 Reinforcement Learning . . . . .	10
2.3 Provocări . . . . .	11
2.4 Concluzie . . . . .	12
<b>3 Fundamente Teoretice</b>	<b>13</b>
3.1 Strategii de tranzacționare folosind indicatori tehnici . . . . .	13
3.1.1 Categoriile de strategii bazate pe indicatori tehnici . . . . .	13
3.2 Medii mobile - definire, tipuri și utilizări . . . . .	15
3.3 Algoritmi genetici - concepte de bază și meta optimizarea . . . . .	17
<b>4 Strategii de bază - încrucișarea de medii mobile</b>	<b>21</b>
4.1 Descrierea strategiei . . . . .	21
4.2 Performanța inițială fără optimizare . . . . .	24

4.2.1	Cazul ideal . . . . .	24
4.2.2	Dinamica pieței . . . . .	26
<b>5</b>	<b>Optimizarea cu algoritmi genetici</b>	<b>28</b>
5.1	Reprezentarea individului și a populației . . . . .	28
5.2	Funcția de fitness și evaluarea populației . . . . .	29
5.3	Operatorii genetici . . . . .	30
5.4	Metoda de testare . . . . .	32
5.4.1	Implementarea Strategiei . . . . .	32
5.4.2	Structura generală a funcției de backtest . . . . .	32
5.4.3	Strategia Long Only cu Crossover . . . . .	33
5.4.4	Strategia Short Only cu Crossover . . . . .	33
5.4.5	Avantajele abordării . . . . .	33
5.4.6	Analiza Strategiei . . . . .	34
5.5	Rezultate experimentale . . . . .	36
5.5.1	Cazul Ideal . . . . .	36
5.5.2	Rezultate pentru BTC . . . . .	37
<b>6</b>	<b>Meta optimizare și evoluția adaptivă</b>	<b>46</b>
6.1	Justificare: de ce meta optimizare? . . . . .	46
6.2	Structura meta algoritmului . . . . .	46
6.3	Evoluția parametrilor de optimizare . . . . .	47
6.4	Rezultate experimentale pentru BTC . . . . .	49
	<b>Concluzii</b>	<b>51</b>
	<b>Bibliografie</b>	<b>53</b>

# Abstract

Această lucrare evaluează performanța unei strategii de tranzacționare pe bursă, bazată pe încrucișarea mediilor mobile și optimizată folosind algoritmi genetici. Strategia este aplicată pe evoluția prețului Bitcoin, și o compară cu strategia clasică de Buy & Hold. Au fost analizate patru funcții de fitness care maximizează: randamentul, Sharpe Ratio, drawdown-ul și o funcție personalizată de forma  $\frac{\text{Return}}{\text{Drawdown}^{0,8}}$ . Testele efectuate au arătat un randament de 87.8% atunci când optimizarea s-a făcut folosind funcția de fitness drawdown, și 35,5% în cazul funcției personalizate, cu o scădere maximă pe întreaga perioadă de 16,17% și respectiv 12,48%. Strategia a fost antrenată și testată pe date istorice din perioada 2018-2025, folosind aproximativ 44 000 de puncte de date.

# Glosar de termeni

**Moving Average** - Indicator tehnic care calculează media prețurilor pe o anumită perioadă, fiind folosit pentru a netezi variațiile prețului.

**MACD (Moving Average Convergence Divergence)** - Indicator bazat pe medii mobile care arată când două dintre ele se apropie sau se depărtează, semnalând schimbări de trend.

**ATR (Average True Range)** - Indicator de volatilitate ce arată cât de mult variază prețul unui activ, fiind util în ajustarea pozițiilor și a riscului.

**RSI (Relative Strength Index)** - Oscilator care măsoară forța recentă a mișcărilor de preț și indică dacă un activ este supracumpărat sau supravândut.

**Money Management** - Set de reguli pentru gestionarea capitalului în tranzacționare, cu scopul de a controla riscurile și de a crește profitul.

**Sharpe Ratio** - Indicator care arată cât câștigă o strategie în raport cu riscul asumat, calculat prin raportul dintre randamentul în exces și riscul său.

**Return** - Reprezintă câștigul sau pierderea unei investiții, exprimat ca procent din suma investită.

**Drawdown** - Scăderea maximă a capitalului față de un vârf anterior, fiind o măsură a riscului unei strategii.

**Alpha** - Arată cât de bine performează o strategie față de piață. Un alpha pozitiv înseamnă o performanță peste media pieței.

**Taker Fees** - Comisioane plătite când un ordin este executat imediat, luând o ofertă deja existentă.

**Maker Fees** - Comisioane plătite pentru ordine care rămân active în piață și contribuie la lichiditate.

**Short Position** - Tranzacție care mizează pe scăderea prețului, vânzând un activ împrumutat pentru a-l răscumpăra mai ieftin.

**Long Position** - Tranzacție prin care se cumpără un activ în așteptarea unei creșteri



a prețului.

**Market Order** - Ordin care se execută imediat la cel mai bun preț disponibil.

**Slippage** - Diferența dintre prețul estimat și cel la care se execută efectiv ordinul, apărută din cauza volatilității sau lichidității.

**Walk Forward Optimization** - Tehnică prin care o strategie este optimizată pe un interval și testată pe următorul, pentru a simula condiții reale de piață.

**Backtesting** - Testarea unei strategii pe date istorice pentru a estima cum ar fi funcționat în trecut.

# Capitolul 1

## Introducere

În contextul piețelor financiare moderne, caracterizate printr-un grad ridicat de complexitate, volatilitate și incertitudine, dezvoltarea de strategii robuste și adaptabile de tranzacționare a devenit un subiect de interes major atât pentru cercetători, cât și pentru specialiștii din domeniul financiar. O direcție activă de cercetare vizează integrarea metodelor de optimizare computațională în procesul de construcție și calibrare a strategiilor tehnice, cu scopul de a îmbunătăți performanța acestora.

În acest studiu îmi propun analiza, optimizarea și evaluarea unei strategii tehnice clasice de tip *moving average crossover*<sup>1</sup>, utilizată pe scară largă în tranzacționarea algoritmică. Abordarea propusă constă în aplicarea unui algoritm genetic pentru identificarea configurațiilor optime ale parametrilor strategiei, cum ar fi ferestrele temporale ale mediilor mobile, în corelație cu condițiile istorice de piață. Optimizarea este realizată folosind un cadru riguros de testare cu ajutorul *walk forward testing*<sup>2</sup> pentru a evalua adaptabilitatea și generalizarea parametrilor obținuți dar și a simula un cadru realist de tranzacționare.

Prin această integrare a metodelor evolutive cu principii de validare, urmărim construirea unui model algoritmic capabil să se adapteze automat la schimbările structurale ale pieței, minimizând riscul de supraînvățare și maximizând stabilitatea performanței. Studiul adresează, de asemenea, implicațiile utilizării meta optimizării și selecția automată a parametrilor folosiți în evoluție.

Scopul final al lucrării este dezvoltarea unui cadru conceptual și computațional care permite optimizarea adaptivă a strategiilor tehnice, oferind o contribuție practică la procesul de automatizare inteligentă a deciziilor de tranzacționare.

---

<sup>1</sup>O strategie bazată pe 2 medii mobile care se încrucișează.

<sup>2</sup>Testare iterativă care verifică adaptabilitatea strategiei.

## 1.1 Motivație

Interesul meu pentru piețele financiare și pentru modul în care funcționează economia a dus, în mod firesc, către o preocupare constantă pentru bursă. Combinând această pasiune cu competențele dobândite în dezvoltarea software, am început să explorez și să testez diferite strategii de tranzacționare, de la cele de bază până la unele mai complexe, în încercarea de a obține rezultate cât mai bune.

Pe parcursul acestor experimente, am observat că piețele financiare sunt extrem de dinamice și uneori imprevizibile, reacționând la factori externi care pot fi greu de anticipat. Acest comportament variabil al pieței a stat la baza deciziei de a realiza această lucrare, care își propune să optimizeze strategiile de tranzacționare prin utilizarea celor mai relevante date disponibile și prin aplicarea algoritmilor genetici.

## 1.2 Contribuții

Această lucrare reprezintă o inițiativă personală de a identifica o metodă eficientă pentru obținerea unor randamente favorabile pe piața criptomonedelor. Ideea utilizării algoritmilor genetici pentru optimizarea unei strategii de tranzacționare bazate pe urmărirea trendului îmi aparține și a rezultat din experiența acumulată în ultimii ani, în care am analizat diverse strategii de tranzacționare din această piață.

# Capitolul 2

## Metode de ultima generație

### 2.1 Introducere

În ultimii ani, piețele financiare au devenit tot mai dinamice și impredictibile, în mare parte ca urmare a creșterii accelerate a tranzacționării algoritmice. Acest context a amplificat complexitatea pieței și a redus eficiența strategiilor tradiționale, caracterizate prin reguli fixe și abordări rigide. Ca răspuns, literatura de specialitate s-a concentrat tot mai mult pe dezvoltarea și evaluarea unor metode avansate de optimizare și modelare, menite să ofere o înțelegere mai profundă a mecanismelor pieței și să identifice strategii de tranzacționare cu potențial ridicat de adaptabilitate. În prezent, metode precum algoritmi genetici, meta heuristicele evolutive și rețelele neuronale sunt intens studiate în vederea optimizării strategiilor de tip black box, capabile să funcționeze eficient în domeniul financiar care este volatil.

### 2.2 Analiza metodelor evolutive în optimizare

Una dintre cele mai studiate direcții de cercetare în domeniul tranzacționării algoritmice este reprezentată de optimizarea strategiilor clasice, precum strategiile bazate pe MACD <sup>1</sup> sau cele bazate pe intersectarea mediilor mobile (MA crossover). În încercarea de a spori eficiența acestor strategii și de a reduce riscul asociat, numeroase lucrări din literatura recentă explorează metode variate de optimizare, de la algoritmi genetici clasici la rețele neuronale evolutive și tehnici bayesiene.

---

<sup>1</sup>Moving Average Convergence Divergence

### 2.2.1 Optimizarea indicatorilor tehnici

Chen și Zhu (2025) [1] propun o abordare inovatoare pentru optimizarea indicatorului MACD, combinând tehnici de prelucrare a semnalelor digitale (Transformata Fourier), într-un cadru implementat în framework-ul MindSpore. Parametrii MACD sunt adaptați individual pentru fiecare activ financiar, prin intermediul unui algoritm genetic clasic cu elitism, crossover și mutație. Rezultatele indică o îmbunătățire substanțială a randamentului anual (+272%), a Sharpe Ratio (+86%) și a ratei de succes, alături de o reducere a drawdown-ului maxim.

Totuși, strategia este aplicată în mod retroactiv pe întregul set de date, fără validare pe date neutilizate anterior, ceea ce ridică riscul semnificativ de overfitting. Acest lucru evidențiază necesitatea unor metode de testare și antrenare care să permită adaptarea parametrilor în funcție de evenimentele din piața bursieră.

### 2.2.2 Modele de predicție

Un exemplu relevant al aplicării metodelor evolutive în contextul modelării secvențiale este oferit de Sha (2024) [7], care propune o arhitectură de tip GA LSTM<sup>2</sup> pentru predicția prețului acțiunilor combinând algoritmi genetici cu rețele neuronale de tip Long Short-Term Memory (LSTM). Scopul abordării este optimizarea hiperparametrilor arhitecturii LSTM (inclusiv numărul de neuroni ascunși, rata de învățare sau adâncimea rețelei) prin explorare evolutivă globală. Acest lucru permite creșterea preciziei în predicții.

Modelul GA-LSTM capitalizează pe forța exploratorie a algoritmilor genetici pentru a identifica configurații optime ale modelului LSTM înainte de antrenare. În cadrul studiului, fiecare individ din populație reprezintă o combinație de hiperparametri ai rețelei. Evoluția se realizează prin selecție, încrucișare (single/multipoint crossover) și mutație aleatorie, în funcție de o funcție de fitness care reflectă acuratețea predicțiilor pe un set de validare.

Testele au fost efectuate pe un set de date reale dintr-o companie financiară americană, acoperind perioada 2006-2022. Rezultatele experimentale au arătat o îmbunătățire semnificativă a performanței modelului, cu o reducere a erorii medii absolute (MAE) de la 0.11 la 0.01 în timpul antrenării, și un scor  $R^2$  de 0.87 pe setul de test, indicând o corelație ridicată între predicții și valorile reale.

---

<sup>2</sup>Se utilizează un algoritm genetic pentru a optimiza o rețea LSTM cu scopul de a prezice comportamentul unei serii temporale.

Această lucrare subliniază versatilitatea metodelor evolutive, nu doar în generația de strategii directe de tranzacționare, ci și în optimizarea indirectă a arhitecturilor de învățare, care pot modela în mod eficient dinamica prețurilor. Utilizarea GA ca meta-optimizator în contextul unei rețele LSTM oferă un compromis solid între capacitatea de generalizare și complexitatea structurală.

Totodată, abordarea evidențiază potențialul metodelor hibridizate, în care algoritmi evolutivi sunt utilizați ca instrumente pentru explorarea spațiilor mari de căutare. Această direcție poate fi extinsă către strategii de optimizare multitask sau sisteme adaptative în timp real, capabile să reconfigureze dinamica modelului în funcție de schimbările pieței.

### **2.2.3 Rețele evolutive**

Într-o direcție complementară, Huang (2024) [4] aplică algoritmul NEAT (Neuro-Evolution of Augmenting Topologies) pentru generarea de strategii de tranzacționare pe baza a șapte indicatori tehnici. Deși modelul evoluat a obținut o performanță stabilă și un risc mai redus decât strategia de Buy & Hold, randamentul mediu a fost inferior. De asemenea, au fost semnalate probleme legate de convergența arhitecturii rețelei neuronale, cu numeroase conexiuni inactive și o complexitate greu de interpretat. Astfel, deși atractive teoretic, rețelele evolutive pot introduce un grad ridicat de opacitate și instabilitate computațională.

Menoita și Silva (2025) [2] extind abordările evolutive prin Vectorial Genetic Programming (VGP), folosind operatori vectoriali și o arhitectură strongly typed pentru a genera expresii matematice aplicabile pe date financiare. VGP a produs strategii semnificativ mai profitabile decât cele obținute cu Genetic Programming clasic, însă diferența dintre performanța din perioada de antrenament și cea din perioada de testare a fost semnificativă, semnalând o tendință spre supraajustare și o scădere a capacității de generalizare.

### **2.2.4 Optimizarea bayesiană**

Rahman (2024) [3] explorează utilizarea optimizării bayesiene (Bayesian Optimization - BO) pentru ajustarea automată a parametrilor indicatorului Supertrend. Prin învățare secvențială și funcții de achiziție, BO permite explorarea eficientă a spațiului de parametri, îmbunătățind performanțele în raport cu strategiile configurate manual.

Cu toate acestea, metoda presupune o piață staționară și nu integrează adaptarea în timp real, iar costurile computaționale pot deveni mari în scenarii cu date frecvent actualizate.

## 2.2.5 Reinforcement Learning

O direcție tot mai promițătoare în cercetarea recentă o reprezintă utilizarea algoritmilor de tip reinforcement learning (RL), în special în contextul regimurilor de piață non staționare. Noguera Alonso et al. (2023) [5] aplică algoritmul de Proximal Policy Optimization (PPO), alături de tehnica de reglaj automat Population Based Training (PBT), într-un cadru de tranzacționare intraday pentru acțiuni americane. Modelul, instruit pe date cu frecvență de 5 minute, a demonstrat performanțe out of sample superioare față de piață și capacitate de generalizare pe titluri neincluse în setul de antrenament.

Un aspect remarcabil al studiului este renunțarea la value function din arhitectura clasică PPO, considerată instabilă în condiții financiare reale. Rezultatele validează fezabilitatea metodelor RL în medii financiare reale, dar subliniază și nevoia unor mecanisme robuste de adaptare continuă.

O contribuție interesantă asupra aplicării RL în tranzacționare automată este adusă de Luo și Duan (2023) [6], care explorează performanța unor agenți antrenați cu Deep Q Learning, Deep SARSA și Policy Gradient, într-un cadru care diferențiază explicit între perioade de piață „favorabile” (pre 2021) și „nefavorabile” (post 2021). Studiul oferă o analiză comparativă detaliată asupra comportamentului agenților în fața volatilității crescute și a schimbărilor structurale din datele istorice. Cadrul propus de autori folosește un mediu de tip Markov Decision Process (MDP) cu stări definite prin vectori conținând prețul curent, balanța portofoliului, numărul de acțiuni deținute și mai mulți indicatori tehnici (MACD, RSI, CCI, ADX). Acțiunile agentului sunt discrete și cuantifică volumele de tranzacționare (vânzare, păstrare, cumpărare), iar recompensa reflectă modificarea valorii totale a portofoliului după comisioane. În scenariile „bune” (2013-2020), toți cei trei agenți au obținut randamente anuale medii remarcabile: 70-90%, cu rezultate remarcabile pentru acțiuni precum Tesla (peste 400%). Totuși, în perioada „nefavorabilă” (2021-2022), marcată de incertitudine, performanțele au scăzut semnificativ: unele acțiuni au înregistrat randamente negative, în timp ce altele au menținut profituri modeste (2-7% anual).

Această lucrare scoate în evidență două aspecte esențiale:

- Degradarea performanței în medii non staționare: modelele antrenate pe date stabile nu generalizează eficient în regimuri instabile.
- Importanța diversității algoritmice: chiar și între metode RL clasice (SARSA vs Q Learning), performanțele diferă semnificativ în funcție de dinamica pieței și de structura datelor.

În plus, autorii evidențiază faptul că discretizarea stărilor sau acțiunilor reduce capacitatea de generalizare, iar folosirea rețelelor neuronale pentru aproximarea funcțiilor Q sau politicii directe este esențială în spații continue. De asemenea, folosirea unor metrici adaptate (ex. Huber Loss) și a optimizatorilor precum Adam sau SGD a permis o convergență stabilă în cele mai multe cazuri.

## 2.3 Provocări

Deși metodele evolutive și cele bazate pe reinforcement learning au demonstrat rezultate promițătoare în simulări, ele prezintă limitări semnificative în contexte reale, caracterizate de regimuri de piață dinamice și non staționare. Un prim aspect critic este reprezentat de lipsa unui mecanism de adaptare secvențială, care să permită modelelor să răspundă proactiv la schimbările structurale ale pieței.

Algoritmii genetici, prin natura lor, tind să identifice configurații optime doar pentru seturile de date pe care au fost antrenați, fără a include un sistem de recalibrare automată sau actualizare incrementală. Astfel, strategiile riscă să fie supraajustate la perioade istorice favorabile și să își piardă relevanța în condiții noi de piață.

În mod similar, metodele de tip reinforcement learning, deși capabile să învețe politici decizionale complexe, se confruntă adesea cu instabilitate în antrenare, sensibilitate la hiperparametri și dificultăți de generalizare. Absența unor arhitecturi robuste de tip online learning sau meta-învățare reduce capacitatea agenților de a menține performanța în regimuri volatile, mai ales în lipsa unei funcții de recompensă bine definite sau în fața datelor zgomotoase.

În acest context, se conturează tot mai clar necesitatea unui cadru hibrid adaptiv, care să integreze:

- **Robustetea globală a metodelor evolutive** (ex. GA, VGP) pentru explorarea configurațiilor optime;



- **Flexibilitatea decizională a algoritmilor de reinforcement learning** (ex. PPO, DQN) pentru adaptarea la context în timp real;
- **Mecanisme de recalibrare secvențială** (ex. rolling window, reînvățare periodică) pentru menținerea performanței în fața schimbărilor de regim ale pieței.

## 2.4 Concluzie

Studiul metodelor avansate de optimizare și învățare automată aplicate în tranzacționarea algoritmică relevă un peisaj tehnologic în plină expansiune, în care algoritmi genetici, rețelele neuronale și modelele de tip reinforcement learning contribuie semnificativ la îmbunătățirea performanțelor în piețele financiare. Fiecare clasă de metode aduce avantaje specifice, de la explorarea eficientă a spațiului de soluții (GA), la capacitatea de învățare secvențială și adaptivitate (PPO, dar și limitări care necesită atenție critică.

Literatura actuală indică o direcție clară spre abordări hibride și sisteme adaptive, capabile să încorporeze atât optimizarea globală, cât și învățarea contextuală în timp real. Totodată, provocările legate de supraajustare, complexitate computațională și lipsa explicabilității continuă să reprezinte obstacole majore în implementarea practică a acestor modele.

În concluzie, avansul real în tranzacționarea algoritmică nu va proveni dintr-o metodă singulară, ci din combinarea inteligentă a mai multor paradigme evolutive, bayesiene și bazate pe învățare într-un cadru robust, adaptiv și transparent. Această direcție oferă nu doar perspective teoretice promițătoare, ci și potențialul de a construi sisteme algoritmice sustenabile, cu aplicabilitate concretă în piețele financiare moderne.

# Capitolul 3

## Fundamente Teoretice

### 3.1 Strategii de tranzacționare folosind indicatori tehnici

Indicatorii tehnici reprezintă instrumente matematice aplicate pe datele istorice ale prețurilor și volumelor de tranzacționare, utilizați pentru a sprijini deciziile de cumpărare sau vânzare în piețele financiare. Aceștia au apărut în prima jumătate a secolului XX, odată cu dezvoltarea analizei tehnice ca metodă alternativă la analiza fundamentală. Așa cum zicea și John Murphy, „Analiza tehnică este studiul acțiunii pieței, realizat în principal prin utilizarea graficelor, cu scopul de a anticipa tendințele viitoare ale prețurilor.” [8] Printre primii practicanți ai analizei tehnice se numără Charles Dow, redactor la Wall Street Journal și co-fondator al Dow Jones & Company, considerat părintele analizei tehnice moderne

Indicatorii tehnici sunt utilizați pentru a identifica modele recurente în evoluția prețurilor, a evalua momentumul pieței, volatilitatea și posibilele puncte de inflexiune ale trendurilor. În mod ideal, aceștia pot ajuta la obținerea unui avantaj statistic, cunoscut ca  $\alpha$ , prin identificarea unor patternuri predictibile în comportamentul pieței. Analiza tehnică se bazează pe premisa că toată informația relevantă este deja reflectată în preț, iar mișcările prețurilor tind să se repete în timp, permițând astfel anticiparea direcției viitoare a pieței.

#### 3.1.1 Categoriile de strategii bazate pe indicatori tehnici

Strategiile de tranzacționare care utilizează indicatori tehnici pot fi clasificate, în general, în două categorii fundamentale: strategii de urmărire a trendului (trend following) și strategii de revenire la medie (mean reversion). Potrivit lui Clenow (2013),

majoritatea strategiilor de tranzacționare se încadrează într-una dintre aceste două abordări, fiecare bazându-se pe ipoteze diferite cu privire la comportamentul pieței [9].

**Strategii de tip mean reversion.** Aceste strategii presupun că prețurile tind să revină către o valoare medie sau echilibrată în timp. Indicatorii tehnici folosiți în acest context detectează condiții de supracumpărare sau supravânzare. Un exemplu clasic este strategia bazată pe RSI (Relative Strength Index)[10]. Atunci când RSI depășește valoarea de 70, se consideră că activul este supracumpărat și este probabil să urmeze o corecție. Invers, valori sub 30 indică o posibilă revenire ascendentă. Un semnal tipic de cumpărare este RSI crossover, când RSI urcă peste 30, indicând o inversare de trend spre creștere.

**Strategii de tip trend following.** Aceste strategii urmăresc capturarea mișcărilor pieței de capital, presupunând că un trend existent are probabilitatea de a continua. Un exemplu reprezentativ este strategia care folosește indicatorul MACD (Moving Average Convergence Divergence) pentru a identifica discrepanțele între direcția prețului și MACD. O divergență pozitivă apare atunci când prețul atinge un minim inferior, dar MACD indică un minim superior, sugerând o potențială inversare în sus a trendului.

Un alt exemplu este utilizarea benzilor Bollinger. [11]. Acestea constau dintr-o medie mobilă centrală și două benzi plasate la un anumit număr de deviații standard. Într-o strategie de trend following, pătrunderea prețului peste banda superioară poate fi interpretată ca un semnal de continuare a trendului ascendent.

## Sharpe Ratio - definire și utilizare

*Sharpe Ratio* este unul dintre cei mai utilizați indicatori de performanță în analiza strategiilor de investiții, fiind introdus de William F. Sharpe în 1966.[12] Acesta măsoară randamentul unei strategii raportat la riscul asumat, oferind astfel o metrică standardizată pentru compararea performanțelor diferitelor portofolii sau strategii.

Sharpe Ratio este definit prin relația:

$$S = \frac{R_p - R_f}{\sigma_p}$$

unde:

- $R_p$  — randamentul mediu al portofoliului (sau al strategiei),
- $R_f$  — rata fără risc (ex: randamentul obligațiunilor de stat),
- $\sigma_p$  — deviația standard a randamentului portofoliului (ca măsură a riscului).

Valoarea Sharpe Ratio exprimă cât de mult profit se obține pe unitate de risc. Un Sharpe Ratio pozitiv și ridicat indică o strategie eficientă, în timp ce valori negative sau apropiate de zero sugerează performanță slabă sau risc excesiv.

În această lucrare, **Sharpe Ratio este utilizat ca parte integrantă a funcției de fitness** pentru a controla volatilitatea strategiilor selectate de algoritmul genetic. În combinație cu randamentul brut, utilizarea Sharpe Ratio permite o selecție preferențială a indivizilor care nu doar obțin profit, ci o fac într-un mod stabil și controlat. Această abordare are rolul de a echilibra presiunea de selecție între obiectivul de maximizare a câștigurilor și cel de limitare a riscurilor, conducând astfel la strategii mai robuste în contextul piețelor financiare volatile.

Totuși, această metodă introduce un comportament particular: **în unele cazuri, randamentul este excesiv penalizat de componenta de volatilitate**, ceea ce conduce la selecția unor strategii cu profit mediu apropiat de zero. Cu alte cuvinte, algoritmul poate prefera strategii care evită complet riscul, chiar dacă acest lucru înseamnă renunțarea la orice potențial câștig. Acest efect secundar evidențiază limitările folosirii exclusive a Sharpe Ratio în funcția de fitness și necesită o calibrare atentă sau completarea acestuia cu alte metrice pentru a evita astfel de situații de sub-performanță extremă.

## 3.2 Medii mobile - definire, tipuri și utilizări

Mediile mobile constituie un instrument fundamental al analizei tehnice, fiind utilizate în special în cadrul strategiilor de tip trend following. Acestea au rolul de a netezi seriile temporale ale prețurilor, reducând impactul fluctuațiilor de scurtă durată și evidențiind direcția predominantă a pieței. Din punct de vedere matematic, mediile mobile sunt calculate prin aplicarea unei funcții de mediere (aritmetică, exponențială sau ponderată) asupra unui set definit de date istorice [8]

## Tipuri de medii mobile și formulele asociate

**1. Simple Moving Average (SMA)** Media mobilă simplă este calculată ca media aritmetică a ultimelor  $n$  prețuri de închidere:

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i}$$

unde  $P_{t-i}$  reprezintă prețul de închidere în ziua  $t - i$ , iar  $n$  este fereastra de timp.

**2. Exponential Moving Average (EMA)** EMA se calculează recursiv cu formula:

$$EMA_t = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1}$$

unde  $\alpha = \frac{2}{n+1}$  este factorul de netezire, iar  $n$  este lungimea perioadei.

**3. Weighted Moving Average (WMA)** WMA acordă o pondere descrescătoare valorilor mai vechi:

$$WMA_t = \frac{\sum_{i=1}^n w_i \cdot P_{t-n+i}}{\sum_{i=1}^n w_i}$$

unde  $w_i$  este ponderea pentru fiecare punct de date, de obicei  $w_i = i$ .

**4. Double Exponential Moving Average (DEMA)** Double Exponential Moving Average este o medie mobilă adaptivă, proiectată pentru a reduce întârzierea semnalului generat de media exponențială clasică. Formula sa este:

$$DEMA_t = 2 \cdot EMA_t - EMA(EMA_t)$$

unde  $EMA_t$  este media exponențială la momentul  $t$  calculată pe o perioadă  $n$ . Această combinație reduce efectiv lag-ul introdus de netezirea tradițională, oferind astfel un indicator mai reactiv la schimbările de preț, fără a compromite semnificativ caracterul său de filtrare a zgomotului de piață.

## Exemplu de crossover între două medii mobile simple

Presupunem o serie de prețuri de închidere pentru 10 zile:

$$\{10, 11, 12, 13, 14, 15, 14, 13, 12, 11\}$$

Calculăm:

- SMA pe 3 zile (short)
- SMA pe 5 zile (long)

**SMA3 la ziua 7 (index  $t = 7$ )**

$$SMA3_7 = \frac{14 + 15 + 14}{3} = \frac{43}{3} \approx 14.33$$

**SMA5 la ziua 7 (index  $t = 7$ )**

$$SMA5_7 = \frac{13 + 14 + 15 + 14 + 13}{5} = \frac{69}{5} = 13.8$$

**Observație:**  $SMA3 > SMA5 \Rightarrow$  Semnal de **cumpărare** (short MA intersectează de jos în sus long MA).

**La ziua 10:**

$$SMA3_{10} = \frac{13 + 12 + 11}{3} = 12$$
$$SMA5_{10} = \frac{15 + 14 + 13 + 12 + 11}{5} = 13$$

**Observație:**  $SMA3 < SMA5 \Rightarrow$  Semnal de **vânzare** (short MA intersectează de sus în jos long MA).

### Utilizare în tranzacționare

În practică, cele mai frecvente combinații sunt:

- SMA(50) și SMA(200): utilizate în strategii pe termen lung.
- EMA(12) și EMA(26): utilizate în calculul MACD.
- SMA(20) și SMA(50): utilizate în tranzacționarea de trend pe termen mediu.

## 3.3 Algoritmi genetici - concepte de bază și meta optimizarea

Algoritmii genetici (GA) reprezintă o clasă de algoritmi evolutivi inspirați de mecanismele selecției naturale și ale evoluției biologice. Propuși inițial de John Holland în anii 1970, aceștia au fost dezvoltati ca metode de optimizare stocastică, simulând procesele de moștenire genetică și selecție darwiniană [13]. Potrivit lui Goldberg, „Algoritmii genetici sunt algoritmi de căutare bazați pe mecanismele selecției naturale și ale geneticii naturale” [14]. Alături de strategiile evolutive și programarea evolutivă, algoritmii genetici formează clasa mai generală a algoritmilor evolutivi.

## Structura generală

Algoritmii genetici mențin o populație de soluții candidat, numite cromozomi, care evoluează de-a lungul mai multor generații. Evoluția populației este ghidată de o funcție *fitness*, care cuantifică performanța fiecărui individ. Cromozomii sunt reprezentați de șiruri de gene, iar fiecare poziție este un *locus*. Evoluția are loc prin aplicarea operatorilor genetici (mutație, încrucișare), urmată de selecția indivizilor cei mai bine adaptați. [15]

### Pseudocodul general al unui algoritm genetic:

```
t := 0
generate the starting population P(t)
evaluate P(t)
while not StoppingCondition
    t := t + 1
    select P(t) from P(t-1)
    mutate P(t)
    cross-over P(t)
    evaluate P(t)
```

## Reprezentarea soluțiilor și funcția fitness

În optimizarea numerică, cromozomii sunt adesea reprezentați binar. Funcția fitness este derivată din funcția obiectiv, fiind formulată pentru maximizare. Pentru funcții de minimizat, se aplică transformări precum:

$$f(x) = -g(x), \quad \text{sau} \quad f(x) = \frac{1}{g(x)}$$

Dacă funcția obiectivă  $g(x)$  nu este pozitivă, se poate adăuga o constantă:

$$f(x) = g(x) + C$$

Alegerea transformării influențează presiunea de selecție, care trebuie controlată pentru a preveni stagnarea prematură a evoluției.

## Mecanisme de selecție

- **Selecția proporțională** (roata norocului): indivizii sunt selectați cu probabilitate proporțională cu fitnessul lor.
- **Selecția bazată pe rang**: se evită convergența prematură prin atribuirea probabilităților în funcție de rang, nu de valoarea fitnessului.
- **Selecția turneu**: se aleg aleatoriu  $k$  indivizi, dintre care cei mai buni  $j$  sunt selectați.
- **Elitism**: cei mai buni  $k$  indivizi sunt păstrați neafecți în următoarea generație, pentru a evita pierderea soluțiilor bune.

## Operatorii genetici

**Mutatia:** presupune modificarea aleatorie a unei gene într-un cromozom:

001011000  $\rightarrow$  001011010

**Încrucișarea:** presupune combinarea informației genetice a doi cromozomi:

- *Cu un punct de tăiere:*

01001011  $\rightarrow$  01111100

10111100  $\rightarrow$  10001011

- *Cu  $n$  puncte de tăiere (ex.  $n = 3$ ):*

01001011  $\rightarrow$  01101000

10111100  $\rightarrow$  10011111

- *Încrucișare uniformă:* fiecare genă este aleasă aleator din unul dintre părinți.

## Meta optimizarea

Meta optimizarea reprezintă aplicarea unor tehnici de optimizare asupra hiperparametrilor algoritmilor genetici, cum ar fi dimensiunea populației, rata de mutație, rata de încrucișare, procentul de elitism sau schema de selecție. Alegerea acestor valori are un impact direct asupra performanței algoritmului, influențând atât viteza de convergență, cât și capacitatea acestuia de a evita capcanele locale [16].



O direcție importantă în meta optimizare o constituie utilizarea algoritmilor genetici de nivel superior, denumiți *meta algoritmi genetici*, care optimizează automat hiperparametrii unui alt algoritm genetic. Fiecare individ din meta populație reprezintă un set complet de parametri, iar fitnessul acestora este evaluat prin rularea algoritmului de bază și măsurarea calității soluției generate.

Structura unui meta algoritm genetic urmează schema clasică evolutivă:

1. Se generează o populație de seturi de parametri.
2. Pentru fiecare individ, se rulează algoritmul de bază cu parametrizarea respectivă și se evaluează performanța.
3. Se aplică selecția, încrucișarea și mutația pentru a genera o nouă generație de seturi de parametri.
4. Procesul se repetă până la atingerea unei condiții de oprire.

Meta optimizarea poate fi privită ca un proces de nivel superior, în care scopul nu este optimizarea directă a unei funcții obiectiv, ci optimizarea procesului de optimizare în sine. Această idee este bine surprinsă de Eiben și Smith, care afirmă că „În loc să proiectăm un algoritm, lăsăm algoritmul să se proiecteze singur.” [17]. Această abordare este aplicabilă nu doar algoritmilor genetici, ci și altor metode evolutive sau algoritmi euristici.

Avantajul principal al meta optimizării prin algoritmi genetici constă în capacitatea de adaptare automată la specificul problemei fără intervenție manuală. Dezavantajul major este costul computațional ridicat, deoarece fiecare evaluare implică rularea completă a unui algoritm interior. Pentru a limita acest impact, pot fi utilizate tehnici precum evaluarea paralelă, folosirea de modele surrogat sau reducerea setului de date.

Pentru a evita supraînvățarea pe o instanță fixă a problemei, este recomandată validarea pe multiple instanțe sau folosirea validării încrucișate. În practică, meta optimizarea s-a dovedit eficientă în optimizarea hiperparametrilor pentru rețele neuronale evolutive, probleme combinatorii și algoritmi hibridi [18].

# Capitolul 4

## Strategii de bază - încrucișarea de medii mobile

### 4.1 Descrierea strategiei

Strategia de tip *moving average crossover* este una dintre cele mai utilizate metode tehnice în tranzacționarea algoritmică, datorită simplității conceptuale și eficienței în detectarea direcției trendului. Aceasta se bazează pe compararea a două medii mobile aplicate asupra prețului unui activ financiar: una cu perioadă scurtă (numită medie mobilă rapidă), și una cu perioadă mai lungă (medie mobilă lentă). Principiul fundamental constă în generarea semnalelor de cumpărare sau vânzare pe baza punctelor de intersecție (crossover) dintre cele două curbe. [19].

#### Reguli de tranzacționare

- **Semnal de cumpărare (buy):** Se generează în momentul în care media mobilă rapidă intersectează de jos în sus media mobilă lentă. Acest semnal este interpretat ca un indiciu că trendul pieței devine ascendent.
- **Semnal de vânzare (sell):** Se generează atunci când media mobilă rapidă intersectează de sus în jos media mobilă lentă, sugerând o posibilă inversare descendentă a trendului.

#### Parametrii strategiei

Elementele definitorii ale acestei strategii sunt:

- **Lungimea mediei mobile rapide** ( $n_s$ ) — de exemplu, 5, 10 sau 20 de perioade.
- **Lungimea mediei mobile lente** ( $n_l$ ) — de exemplu, 50, 100 sau 200 de perioade.
- **Tipul mediei mobile** — poate fi simplă (SMA), exponențială (EMA), ponderată (WMA) etc.
- **Condițiile suplimentare** — filtre pe volum, volatilitate, time-frame sau confirmări prin alți indicatori (opțional).

## Implementare algoritmică

La fiecare moment  $t$ , se calculează:

$$MA_{\text{scurt}}(t) = \frac{1}{n_s} \sum_{i=0}^{n_s-1} P_{t-i}, \quad MA_{\text{lung}}(t) = \frac{1}{n_l} \sum_{i=0}^{n_l-1} P_{t-i}$$

Apoi se evaluează condițiile:

Dacă  $MA_{\text{scurt}}(t) > MA_{\text{lung}}(t)$  și  $MA_{\text{scurt}}(t-1) \leq MA_{\text{lung}}(t-1) \Rightarrow$  cumpărare

Dacă  $MA_{\text{scurt}}(t) < MA_{\text{lung}}(t)$  și  $MA_{\text{scurt}}(t-1) \geq MA_{\text{lung}}(t-1) \Rightarrow$  vânzare

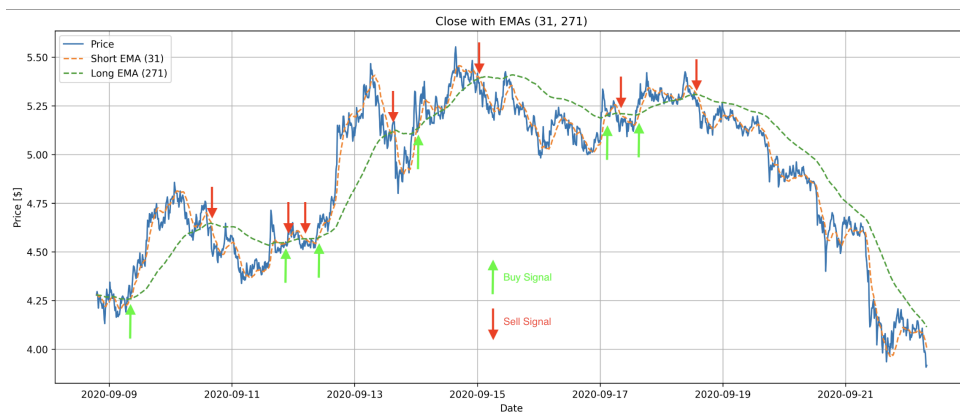


Figura 4.1: Exemplu de semnal generat de strategia Moving Average Crossover

## Completări privind gestionarea pozițiilor

Pentru a îmbunătăți precizia punctelor de intrare și ieșire, strategia este completată prin tehnici de *money management*, menite să controleze riscul și să optimizeze alocarea capitalului. Astfel, nu se recomandă expunerea totală a portofoliului

într-o singură tranzacție. În schimb, se utilizează o fracțiune prestabilită din capital, proporțională cu riscul asumat per tranzacție.

O componentă esențială a acestor tehnici este utilizarea ordinelor de *stop-loss* și *take-profit*, care permit limitarea pierderilor și marcarea profiturilor într-un mod disciplinat. *Stop-loss-ul* definește pierderea maximă acceptabilă pentru o tranzacție, în timp ce *take-profit-ul* fixează un obiectiv de profit prestabilit. Împreună, aceste instrumente contribuie la menținerea unui raport risc-recompensă favorabil și la reducerea impactului emoțional asupra deciziilor de tranzacționare.

De asemenea, tehnica de *position sizing* — adică determinarea dimensiunii optime a fiecărei poziții în funcție de riscul asumat și volatilitatea pieței — este crucială pentru conservarea capitalului pe termen lung. Așa cum subliniază Van K. Tharp, "Gestionarea capitalului este partea sistemului tău de tranzacționare care îți spune «cât de mult»" [20], iar o bună gestionare a riscului poate avea un impact mai mare asupra performanței decât acuratețea semnalelor de tranzacționare în sine.

Un instrument frecvent utilizat în acest context este **ATR-ul** [21] (*Average True Range*), care oferă o estimare a volatilității recente a pieței. Pe baza valorii ATR, se poate stabili un prag adaptiv de *stop-loss*, menit să protejeze capitalul împotriva mișcărilor adverse bruște. De exemplu, un *stop-loss* poate fi plasat la o distanță de  $k \cdot \text{ATR}$  sub nivelul de intrare, unde  $k$  este un factor calibrat empiric.

**Exemplu de calcul:** Presupunem următoarele valori:

- Preț de intrare (buy):  $P_{\text{entry}} = 25,00$
- Valoare ATR pe ultimele 14 zile:  $\text{ATR}_{14} = 1,20$
- Factor de risc aplicat:  $k = 2$

Se calculează nivelul *stop-loss* astfel:

$$\text{Stop-Loss} = P_{\text{entry}} - k \cdot \text{ATR}_{14} = 25,00 - 2 \cdot 1,20 = 22,60$$

Astfel, poziția va fi închisă automat dacă prețul scade sub 22,60, limitând pierderea potențială.

Prin integrarea acestor mecanisme de protecție și gestionare a riscului, strategia devine mai robustă în fața incertitudinii pieței. Totodată, este important de menționat

că atât parametrizarea indicatorilor tehnici, cât și componentele de money management trebuie ajustate în funcție de caracteristicile specifice ale pieței de capital analizate, cum ar fi nivelul de lichiditate, volatilitatea istorică sau orizontul de investiție ales.

## Avantaje și limitări

Strategia oferă o modalitate robustă de urmărire a trendului, dar are și limitări:

- **Avantaje:**

- Ușor de implementat și interpretat.
- Funcționează eficient în piețe direcționale.
- Scalabilă și ușor de automatizat.

- **Limitări:**

- Performanță slabă în condiții de piață laterală (fără trend).
- Risc crescut de semnale false în prezența volatilității ridicate.
- Sensibilitate ridicată la alegerea parametrilor.

Prin urmare, alegerea optimă a parametrilor de intrare devine esențială pentru maximizarea performanței strategiei, motiv pentru care se justifică utilizarea metodelor de optimizare, cum ar fi algoritmi genetici, în cadrul procesului de calibrare.

## 4.2 Performanța inițială fără optimizare

### 4.2.1 Cazul ideal

Într-un cadru teoretic ideal, strategia de tip *moving average crossover* poate genera semnale de tranzacționare eficiente, cu o rată ridicată de succes. Acest lucru se întâmplă atunci când piața se află într-un trend bine definit, iar intersecțiile dintre mediile mobile surprind puncte relevante de inversare sau continuare. În astfel de condiții, perioadele fixe alese pentru mediile mobile (de exemplu, 20 și 50 de zile) pot fi suficiente pentru a furniza semnale robuste, fără a necesita adaptare suplimentară.

Acest scenariu ideal presupune o corelație stabilă între preț și trendul detectat de medii mobile, o frecvență redusă a semnalelor false și un raport risc-recompensă

favorabil. Totuși, aceste condiții sunt rare în practica, iar aplicarea directă a strategiei, fără calibrare, are limitări semnificative.

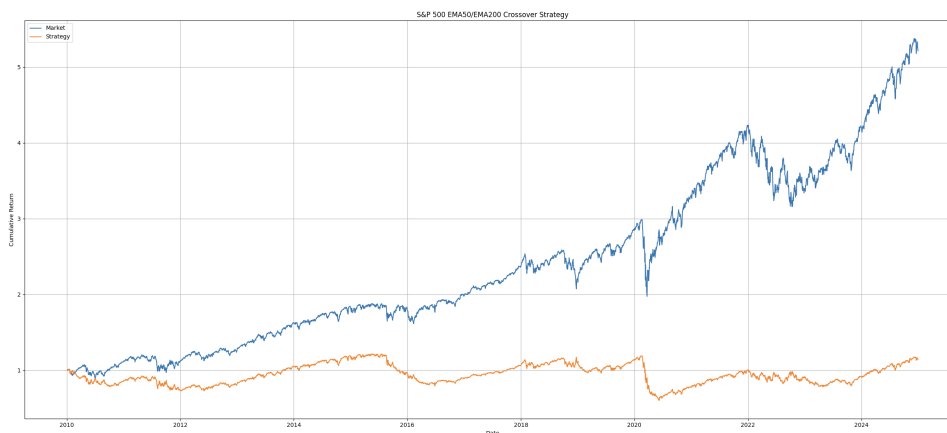


Figura 4.2: Exemplu de performanta intre Buy & Hold si strategia de crossover pentru S&P500. Albastru - B&H, Portocaliu - Crossover

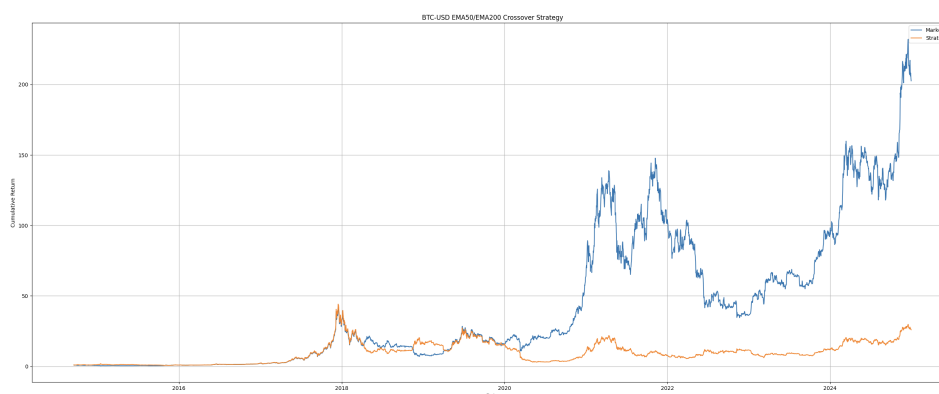


Figura 4.3: Exemplu de performanta intre Buy & Hold si strategia de crossover pentru BTC. Albastru - B&H, Portocaliu - Crossover

Rezultatele evidențiază faptul că, pe termen lung, o strategie de crossover cu parametri constanți nu oferă randamente satisfăcătoare.

### 4.2.2 Dinamica pieței

Piața financiară are un comportament intrinsec dinamic, fiind influențată simultan de factori economici, politici, psihologici și tehnologici. Prețul activelor nu are un caracter staționar, iar proprietățile statistice ale seriilor temporale evoluează în timp. Această variabilitate structurală face ca o strategie cu parametri fixați să își piardă eficiența în anumite contexte.

În mod particular, pot fi identificate două stări dominante ale pieței:

- **Piețe în trend** - caracterizate prin mișcări direcționale prelungite, în care strategiile de tip crossover performează bine.
- **Piețe laterale (range-bound)** - caracterizate prin lipsa unui trend clar și oscilații în jurul unui interval, unde strategiile de urmărire a trendului generează semnale false.

În plus, volatilitatea pieței variază semnificativ în timp. Perioadele de volatilitate ridicată pot genera fluctuații bruște care invalidează semnalele oferite de intersecțiile medii mobile. În contrast, perioadele cu volatilitate scăzută pot duce la absența semnalelor relevante și implicit la o scădere a profitabilității [22].

Prin urmare, performanța unei strategii standard, neoptimizate, este puternic condiționată de regimul de piață curent. În lipsa unui mecanism adaptiv, performanțele pot deveni negative în intervale extinse de timp. Acest aspect evidențiază importanța calibrării dinamice a parametrilor strategiei în funcție de contextul pieței.

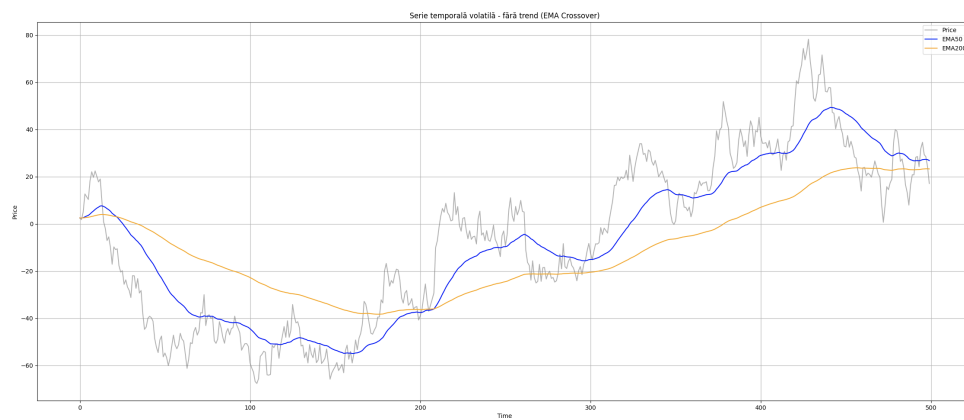


Figura 4.4: Pret simulat cu volatilitate. Se poate observa comportamentul haotic care influențează mediile mobile oferind un effect de lag. Albastru - moving average(50), Portocaliu - moving average (200)

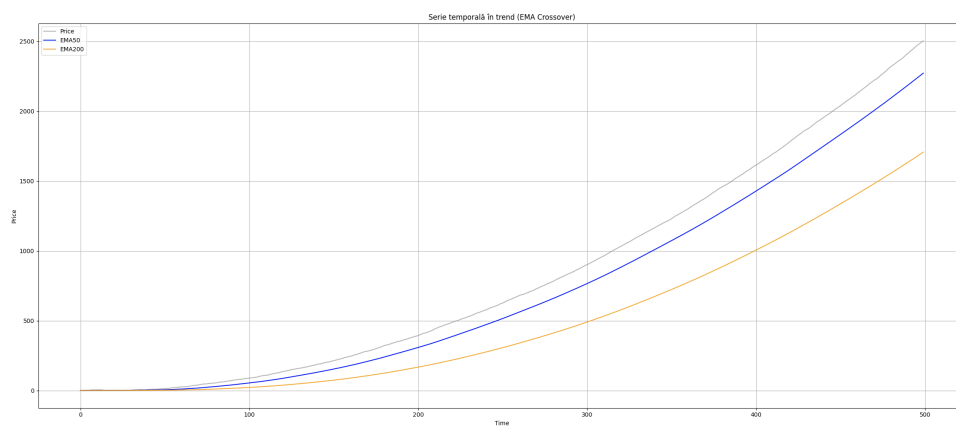


Figura 4.5: Pret simulat fara volatilitate. Albastru - moving average(50), Portocaliu - moving average (200)



# Capitolul 5

## Optimizarea cu algoritmi genetici

### 5.1 Reprezentarea individului și a populației

În cadrul algoritmului genetic, un **individ** reprezintă o soluție candidat, exprimată ca o tuplă de patru gene corespunzătoare parametrilor unei strategii de tranzacționare. Aceste gene sunt generate aleator în cadrul unor intervale bine definite, relevante din punct de vedere financiar:

- `Short_MA`, perioada mediei mobile rapide, între 1 și 300;
- `Long_MA`, perioada mediei mobile lente, tot între 1 și 300;
- `Stop_Loss_Multiplier`, un coeficient ATR, între 0.1 și 10.0;
- `Position_Size`, proporția capitalului investit, între 0.1 și 1.0.

Valorile posibile pentru fiecare parametru sunt stocate într-o structură de tip dicționar, ceea ce permite atât controlul strict al domeniului de valori, cât și ajustări rapide în etapa de testare.

Generarea tuturor valorilor se face cu funcții auxiliare care selectează aleator valori din intervalele stabilite, adaptate la tipul fiecărui parametru (întreg sau real). De exemplu, generarea unei perioade pentru media mobilă rapidă se face astfel:

```
def generate_Short_MA():  
    return random.randint(1, 300)
```

Acest principiu este aplicat similar și celorlalte gene, folosind `random.uniform()` pentru parametri de tip `float`. Funcțiile asigură respectarea limitelor definite în dicționarul global de configurare și oferă flexibilitate pentru experimente.

Pentru eficiență computațională, tipul fiecărui individ este definit cu ajutorul unei structuri de tip `numpy.dtype`, permițând procesare vectorizată și utilizare redusă de memorie.

Fiecare individ este generat aleator și adăugat populației doar dacă este unic, pentru a asigura o diversitate genetică ridicată încă din prima generație. Această verificare se realizează cu ajutorul unui set în care sunt stocate tuplele indivizilor deja generați.

Rezultatul final constă într-un array de indivizi, pregătit pentru a fi utilizat în procesul de selecție, crossover și mutație.

## 5.2 Funcția de fitness și evaluarea populației

În cadrul algoritmului genetic, funcția de fitness joacă un rol esențial în evaluarea calității fiecărui individ, stabilind ordinea de selecție pentru generațiile viitoare. În această lucrare, am testat mai multe abordări distincte pentru evaluare:

- O funcție de fitness simplificată, bazată exclusiv pe **randamentul absolut** (`RETURN`), fără a ține cont de volatilitate sau risc;
- O funcție de fitness extinsă, care integrează **Sharpe Ratio**, un indicator financiar ce măsoară randamentul ajustat la risc, punând accent pe stabilitatea performanței;
- O funcție de fitness neconvențională, care are ca obiectiv maximizarea **Drawdown-ului**, pe baza observațiilor empirice conform cărora strategiile ce acceptă expunere mai mare tind să obțină performanțe superioare în anumite regimuri de piață;
- O funcție de fitness personalizată, definită ca raportul dintre `RETURN` și `DRAWDOWN`<sup>0.8</sup>, oferind un compromis între profitabilitate și controlul riscului extrem.

### Structura funcției de fitness

- **Evaluarea inițială:** se extrage randamentul fiecărui individ (`RETURN`) și se elimină valorile negative:

$$\text{final\_score} = \max(0, \text{RETURN})$$

- **Penalizare logică:** dacă raportul `short_MA / long_MA` este prea apropiat de 1 (ex: peste 0.85), scorul este penalizat, întrucât cele două medii nu vor produce semnale semnificative. Penalizarea este proporțională cu abaterea de la prag:

$$\text{penalty} = \text{clip} \left( \frac{\text{ratio} - \theta}{1 - \theta}, 0, 1 \right)$$

- **Regula de validitate:** indivizii pentru care `short_MA` este mai mare sau egal cu `long_MA` sunt invalidați complet, atribuindu-li-se un scor final nul:

$$\text{final\_score} = 0 \quad \text{dacă} \quad \text{short\_MA} \geq \text{long\_MA}$$

Această constrângere a fost introdusă în urma observațiilor experimentale, care au evidențiat o tendință problematică. În absența acestei reguli, mediile mobile optimizate tind să se alinieze cu direcția trendului dominant din perioada de antrenament. Astfel, în scenariul unei piețe aflate în creștere, strategiile optimizate pot favoriza semnale de tip `long only` (de exemplu, `short_EMA` mai mare fata de `long_EMA`), ceea ce contravine intenției inițiale și poate duce la rezultate proaste în perioada de testare. Această regulă asigură astfel o separare clară a intervalelor de reacție ale mediilor mobile și oferă un plus de stabilitate în luarea deciziilor, chiar și în condiții de piață schimbătoare.

- **Sortare:** după calcularea scorului, populația este sortată descrescător în funcție de scorul obținut.

## Structura de ieșire

Rezultatul funcției este un vector de indivizi de tip `INDIVIDUAL_TYPE_SCORE`, care include toți parametrii originali, împreună cu scorul de fitness. Această structură este utilizată în etapele ulterioare.

Prin utilizarea acestei funcții de fitness, se obține o presiune de selecție echilibrată, care încurajează randamentele pozitive, penalizează strategiile cu structură incorectă și favorizează raporturi solide între profit și risc.

## 5.3 Operatorii genetici

Evoluția populației în cadrul algoritmului genetic este guvernată de trei operatori principali: **selecția**, **încrucișarea** și **mutația**. Aceștia permit explorarea și exploatarea

eficientă a spațiului soluțiilor, favorizând apariția indivizilor cu performanță superioară.

Parametrii ce controlează comportamentul acestor operatori, precum rata de încrucișare sau probabilitatea de mutație, sunt adaptați dinamic de un meta algoritm genetic, pentru a menține un echilibru între diversitate și convergență.

## Operatorul de selecție - Roata Norocului

Pentru selecție se folosește o metodă probabilistică, unde șansele fiecărui individ sunt proporționale cu performanța sa, numita „Roata Norocului”.

```
selection_probabilities = fitness_scores / np.sum(fitness_scores)
selected = np.random.choice(..., p=selection_probabilities)
```

Această metodă favorizează indivizii performanți, dar menține șanse pentru cei mai slabi, prevenind stagnarea prematură. În cazul în care scorurile de fitness sunt nule, se generează o nouă populație aleatoare.

## Operatorul de încrucișare - Crossover

Încrucișarea se aplică perechilor de indivizi selectați, conform unei probabilități adaptate dinamic. Genele sunt împărțite în două grupe logice:

- **Grupa 1**, Parametrii tehnici: perioada mediei mobile rapide (`Short_MA`) și perioada mediei mobile lente (`Long_MA`).
- **Grupa 2**, Parametrii de gestionare a riscului: coeficientul de stop-loss (`Stop_Loss_Multiplier`) și dimensiunea poziției (`Position_Size`).

La fiecare pereche de părinți, se decide aleator dacă se vor schimba genele unei grupe, în funcție de o probabilitate specificată. Astfel, părinții pot genera urmași prin permutarea completă a unei grupe de gene, menținând coerența între componentele funcționale ale individului. De exemplu:

```
if random.random() < CROSSOVER_RATE:
    parent1[gene], parent2[gene] = parent2[gene], parent1[gene]
```

Această strategie de încrucișare pe grupe permite menținerea structurii interne a fiecărei categorii de parametri, încurajând totodată diversitatea și explorarea unor combinații mai eficiente.

## Operatorul de mutație

Mutația este aplicată fiecărei gene cu o probabilitate individuală definită. Dacă mutația este activată, valoarea genei este înlocuită cu una nouă, generată aleator, dar respectând constrângerile impuse în `GENE_RANGES`:

```
if random.random() < MUTATION_PROBABILITY:
    mutated_gene = generate_new_value(var_type)
```

Această tehnică contribuie la menținerea diversității genetice și previne convergența către un optim local, fiind esențială în explorarea spațiului de soluții.

## 5.4 Metoda de testare

### 5.4.1 Implementarea Strategiei

Pentru simularea și evaluarea performanței strategiilor de tranzacționare propuse, a fost utilizată biblioteca `backtesting.py` [23], care oferă o interfață simplificată pentru definirea strategiilor algoritmice și rularea backtesting-ului.

### 5.4.2 Structura generală a funcției de backtest

Funcția `run_backtest` parcurge fiecare individ din populația evoluată de algoritmul genetic, extrăgând parametrii caracteristici unei strategii: perioada mediei mobile scurte (`short_ma`), perioada mediei mobile lungi (`long_ma`), multiplicatorul pentru trailing stoploss calculat pe baza ATR-ului (`stop_loss_multiplier`) și proporția capitalului alocată fiecărei tranzacții (`position_size`). În funcție de direcția testului (`isShortOnly`), se creează o strategie long only sau short only, iar apoi se rulează backtest-ul.

Datele EMA pre calculate (argumentul opțional `emas_data_set`) sunt combinate cu datele recente pentru a permite o inițializare corectă a indicatorilor tehnici.

Rezultatul fiecărui test este salvat ca un tuplu ce conține parametrii strategiei și o serie de metrici de performanță (randament total, drawdown maxim, randamentul strategiei *buy & hold*, durata rulării, numărul de tranzacții, timpul de expunere și Sharpe Ratio).

### 5.4.3 Strategia Long Only cu Crossover

Funcția `createLongOnlyCrossOverStrategy` definește o clasă interioară derivată din `Strategy`, unde logica de tranzacționare este implementată în metoda `next()`. Principalele componente sunt:

- **Indicatori tehnici:** Două medii mobile exponențiale (EMA) calculate cu perioadele scurtă și lungă, precum și indicatorul ATR (Average True Range) cu o perioadă de 14 zile.
- **Condiție de intrare în poziție long:** Inițial, dacă EMA short depășește EMA long. După inițializare, se așteaptă un *crossover* pozitiv al celor două medii mobile.
- **Gestionarea trailing stoploss:** După intrarea în poziție, prețul maxim atins este monitorizat. Dacă prețul actual scade sub un prag determinat din acest maxim și ATR, poziția este închisă.
- **Adjustarea dimensiunii poziției:** Dacă ultima tranzacție a fost pierzătoare, dimensiunea poziției este redusă progresiv până la un minim de 25%. În caz contrar, se revine la dimensiunea inițială.

### 5.4.4 Strategia Short Only cu Crossover

Strategia definită de `createShortOnlyCrossOverStrategy` urmează o structură similară, însă aplicată unei logici inverse:

- **Semnal de intrare short:** Inițial, dacă EMA long este peste EMA short; ulterior, se caută un *crossover* negativ.
- **Trailing stoploss adaptat:** Este setat un prag bazat pe minimul atins de la deschiderea poziției și ATR, pentru a închide poziția în caz de creștere a prețului peste acest prag.
- **Diminuarea poziției la pierdere:** Analog cu strategia long, dimensiunea poziției este ajustată în funcție de profitul ultimei tranzacții.

### 5.4.5 Avantajele abordării

Utilizarea unei clase dedicate pentru fiecare strategie permite definirea clară a regulilor de tranzacționare și a mecanismelor de protecție (ex. trailing stoploss di-

namic). Prin parametrizarea strategiilor, este posibilă evaluarea robustă a diferitelor configurații prin algoritmul genetic.

Astfel, această componentă reprezintă nucleul experimental al lucrării, întrucât permite compararea directă a performanței strategiilor optimizate în diverse regimuri de piață.

#### 5.4.6 Analiza Strategiei

Pentru a evalua performanța strategiilor într-un mod realist și relevant pentru condițiile pieței, am utilizat metoda de **walk forward testing**. Această tehnică presupune împărțirea seriei temporale în segmente cronologice, în care modelul este optimizat pe o fereastră de antrenare și apoi testat pe un interval viitor, neutilizat anterior. Această abordare este recunoscută în literatura de specialitate ca fiind una robustă împotriva supraajustării [24].

**Structura procesului de testare:** Datele istorice au fost segmentate în ferestre cronologice egale. Pentru fiecare iterație, strategia a fost:

- **Antrenată** pe o fereastră de învățare formată dintr-un număr constant de segmente (`TRAINING_CHUNKS = 20`).
- **Testată** pe o fereastră de validare separată, imediat ulterioară (`TEST_CHUNKS = 10`).
- **Preprocesată** cu o porțiune de date anterioare (*warm up*) pentru inițializarea indicatorilor tehnici (ex. EMA, ATR), fără a afecta integritatea datelor.

Această tehnică simulează procesul real de implementare în piață, în care deciziile sunt luate fără a cunoaște viitorul.

**Optimizarea strategiilor:** Pentru fiecare fereastră de antrenare, strategiile sunt optimizate utilizând un algoritm genetic configurat cu o populație mixtă (elită, indivizi aleși aleatoriu, mutații și crossover). Optimizarea urmărește maximizarea randamentului și reducerea drawdown-ului, utilizând ca funcție de fitness o combinație de indicatori de performanță.

**Execuție paralelă:** Pentru fiecare iterație, două populații sunt evaluate în paralel:

- Una pentru strategii **long only**, care exploatează trenduri ascendente.
- Alta pentru strategii **short only**, care profită de scăderi ale prețului.

Această abordare reduce semnificativ timpul de rulare și permite compararea directă a comportamentului strategiilor în funcție de direcția pieței.

**Validare și robustețe:** Prin iterarea acestui proces pe întregul set de date, se obține o imagine fidelă a comportamentului strategiilor în diferite regimuri de piață. În plus, evitarea scurgerii de informație între ferestrele de antrenare și test asigură o evaluare corectă a capacității de generalizare a strategiilor rezultate.

**Evaluarea în faza de testare:** După optimizarea populațiilor long only și short only în fereastra de antrenament, cei mai buni indivizi (conform indicatorului de fitness) sunt selectați pentru evaluare pe fereastra de testare. Se reține doar o proporție fixă, `TOP_PICK`, din fiecare populație. Această selecție se face pentru a simula un portofoliu diversificat format din mai multe strategii performante, în locul unei singure soluții dominante.

Pentru a evalua aceste strategii, fiecare individ este testat pe aceleași date din fereastra de test, utilizând datele de *warm up* pentru inițializarea indicatorilor:

(1) *n* indivizi long + (2) *n* indivizi short  $\Rightarrow$  evaluare paralelă pe aceleași date.

După rulare, pentru fiecare direcție se extrag valorile:

- **Randament total** (câștig sau pierdere acumulată pe perioada de testare).
- **Drawdown maxim** (cea mai mare pierdere relativă față de un vârf anterior).

Fiecare strategie primește o pondere egală din capitalul total, iar valorile finale se calculează ca medie aritmetică ponderată:

$$\begin{aligned} \text{Average Return}_{\text{long}} &= \frac{1}{k} \sum_{i=1}^k R_i^{\text{long}} & \text{Average Drawdown}_{\text{long}} &= \frac{1}{k} \sum_{i=1}^k D_i^{\text{long}} \\ \text{Average Return}_{\text{short}} &= \frac{1}{k} \sum_{i=1}^k R_i^{\text{short}} & \text{Average Drawdown}_{\text{short}} &= \frac{1}{k} \sum_{i=1}^k D_i^{\text{short}} \end{aligned}$$



unde  $k = \text{TOP\_PICK}$ , iar  $R_i$ , respectiv  $D_i$  reprezintă randamentul și drawdown-ul individului  $i$  din selecția finală.

Această abordare reflectă un scenariu realist în care un portofoliu este format din mai multe strategii complementare, fiecare contribuind cu o parte egală la performanța totală. În același timp, se reduce riscul asociat alegerii unui singur model, prin diversificare algoritmică.

## 5.5 Rezultate experimentale

### 5.5.1 Cazul Ideal

Rezultatul performant al strategiei propuse este în mare parte atribuit utilizării unei versiuni filtrate a prețului — și anume, media mobilă exponențială (EMA) — pentru generarea semnalelor de intrare și ieșire. În loc să reacționeze la preț, strategia inițiază tranzacții doar atunci când sunt îndeplinite condiții specifice definite prin comportamentul EMA-urilor. Această abordare, deși eficientă în testele inițiale, este în mod fundamental idealistă, deoarece presupune un comportament de piață neted (*smooth*) și o absență a zgomotului tipic al datelor reale.

Totuși, utilizarea EMA ca proxy pentru preț în generarea semnalelor scoate în evidență un aspect esențial al oricărei strategii de tranzacționare: importanța punctelor de **entry** și **exit**. Un sistem poate deveni semnificativ mai profitabil dacă aceste puncte sunt plasate optim în raport cu structura pieței.

**Simulare în condiții ideale:** Pentru a ilustra potențialul maxim al strategiei, am analizat evoluția activului BTC între anii 2018 și 2025, perioadă în care:

- Prețul a crescut de la **8,000 USD** la **98,000 USD**, rezultând un **randament brut de 1125%**.
- Drawdown-ul maxim al activului în acest interval a fost de **86%**, reflectând volatilitatea ridicată și riscul semnificativ.

Prin contrast, strategia ideală implementată în această lucrare a obținut:

- Un randament total de **500%**.
- Un drawdown maxim de doar **5.87%**.

Raportul **Return-to-Drawdown** pentru strategie este:

$$\frac{500\%}{5.87\%} \approx 85.17$$

Raportul **Return-to-Drawdown** pentru buy & hold este:

$$\frac{1125\%}{86\%} \approx 13.08$$

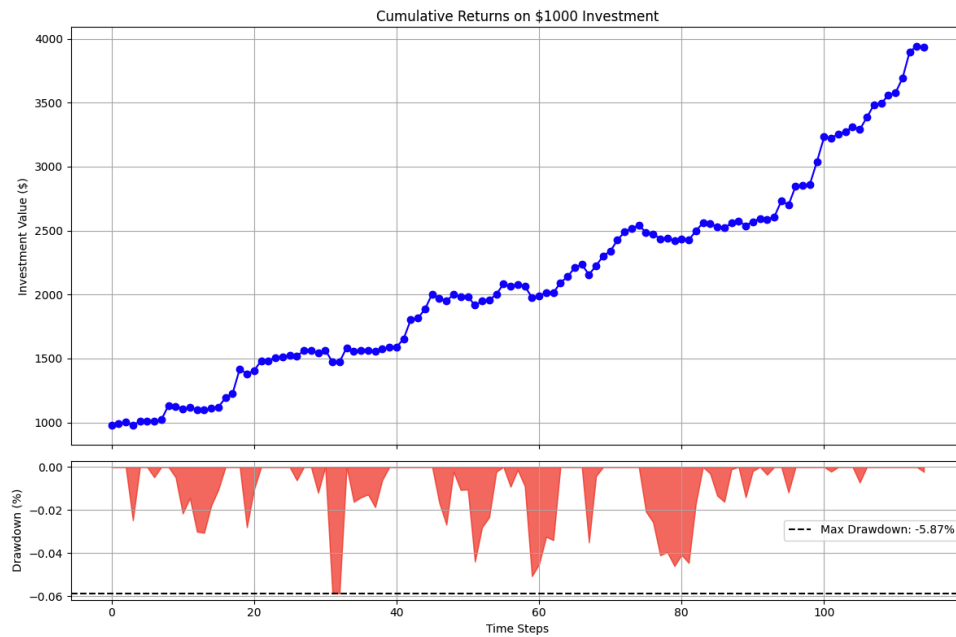


Figura 5.1: Performanța Strategiei în cazul ideal pentru BTC în perioada 2018–2025, timeframe 1h

**Concluzie:** Acest caz ideal relevă potențialul maxim al strategiei în condiții favorabile, trend susținut și volatilitate controlată. Deși asemenea condiții sunt rare în practică, rezultatele subliniază importanța decisivă a punctelor de intrare și ieșire, precum și beneficiile filtrării semnalelor prin indicatori tehnici.

### 5.5.2 Rezultate pentru BTC

Pentru toate experimentele efectuate în faza de testare, am utilizat un capital inițial fix:

$$\text{INITIAL\_CAPITAL} = 200,000 \text{ USD}$$

De asemenea, toate tranzacțiile au fost supuse unui comision standard de:

$$\text{COMMISSION} = 0.0004 \quad (0.04\% \text{ per tranzacție})$$

Acest nivel de comision reflectă *taker fees*-urile practicate în mod obișnuit de către platformele centralizate de tip exchange, precum Binance sau Coinbase Pro, în cazul în care sunt utilizate ordine de tip **market**.

Este important de menționat că în simulările efectuate nu a fost luat în considerare fenomenul de **slippage**, adică diferența dintre prețul așteptat și prețul efectiv de execuție al unei tranzacții în condiții de lichiditate scăzută sau volatilitate extremă. O astfel de omisiune tinde să supraestimeze ușor performanța reală, în special în cazul activelor cu volum redus sau în momentele de mișcări bruște ale pieței.

## Parametrii algoritmului genetic

Pentru antrenarea algoritmului genetic, am utilizat următorii parametri de testare:

- Populație inițială: 100
- Selecție elitistă (top pick): 2
- Selecție prin roata norocului: 88
- Introducerea indivizilor aleatori: 10
- Probabilitate de mutație: 0.035
- Rata de crossover: 0.08
- Numărul maxim de epoci fără îmbunătățiri: 5
- Îmbunătățirea minimă acceptată: 0.001

## Datele utilizate

Pentru antrenarea și testarea modelului, am folosit date istorice cu frecvență orară pentru BTC, încărcate din fișierul `BTC_2018_2025_1h.csv`. Setul de date a fost preluat de pe platforma Kaggle [25], și acoperă perioada 2018-2025. Din întregul set de attribute disponibile, au fost selectate doar coloanele relevante pentru analiza de preț:

Timestamp, Open, High, Low, Close și Volume. Datele sunt procesate în ordine cronologică inversă și sunt împărțite în segmente ("chunks") de câte 100 de valori, fiecare reprezentând o fereastră temporală folosită pentru antrenarea modelului de predicție. Pentru partea vizuală, am folosit *Matplotlib* pentru a afișa randamentul cumulativ pe întreaga perioadă de *walk forward testing* [26].

În următoarele patru grafice, s-a utilizat **Double Exponential Moving Average (DEMA)** ca funcție pentru calculul mediilor mobile, întrucât aceasta a prezentat cea mai bună performanță în toate cele patru funcții de fitness testate. Alegerea DEMA a fost justificată prin capacitatea sa de a reduce întârzierea (*lag-ul*) și de a reacționa mai rapid la schimbările de trend, oferind în același timp o netezire suficientă pentru evitarea semnalelor false.

Pentru început, vom realiza o simulare utilizând următoarea configurație, aplicată pentru mai multe funcții de fitness: un număr de **12** segmente este alocat pentru antrenare, iar **10** segmente sunt utilizate pentru testare.

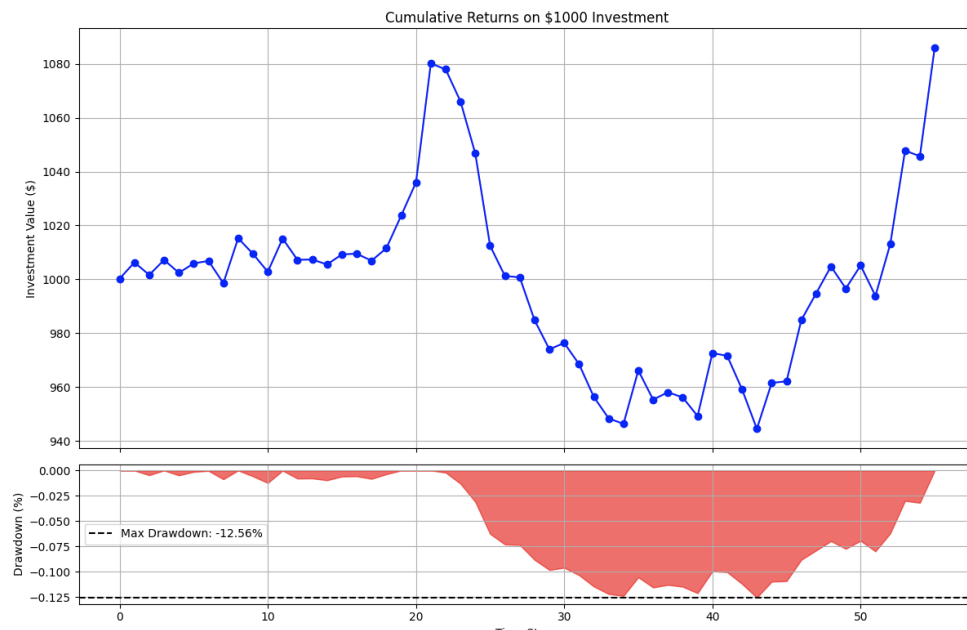


Figura 5.2: Perioada [2018-2025]. Funcția de fitness utilizată: **Sharpe Ratio**.

Se poate observa că strategia oscilează într-un interval relativ restrâns, deoarece componenta de risc este penalizată semnificativ, iar pierderile sunt puternic limitate.

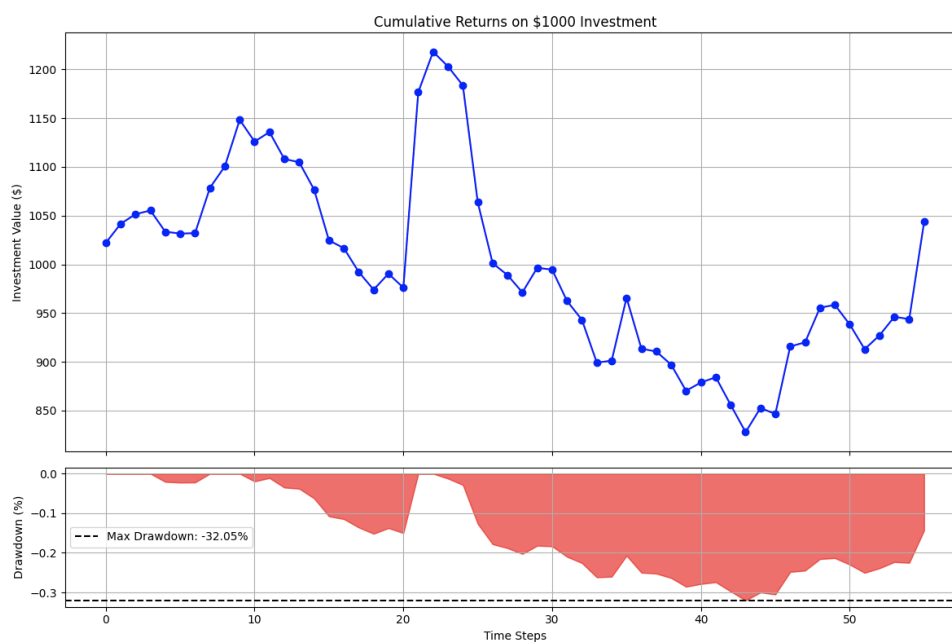


Figura 5.3: Perioada [2018-2025]. Funcția de fitness: Return

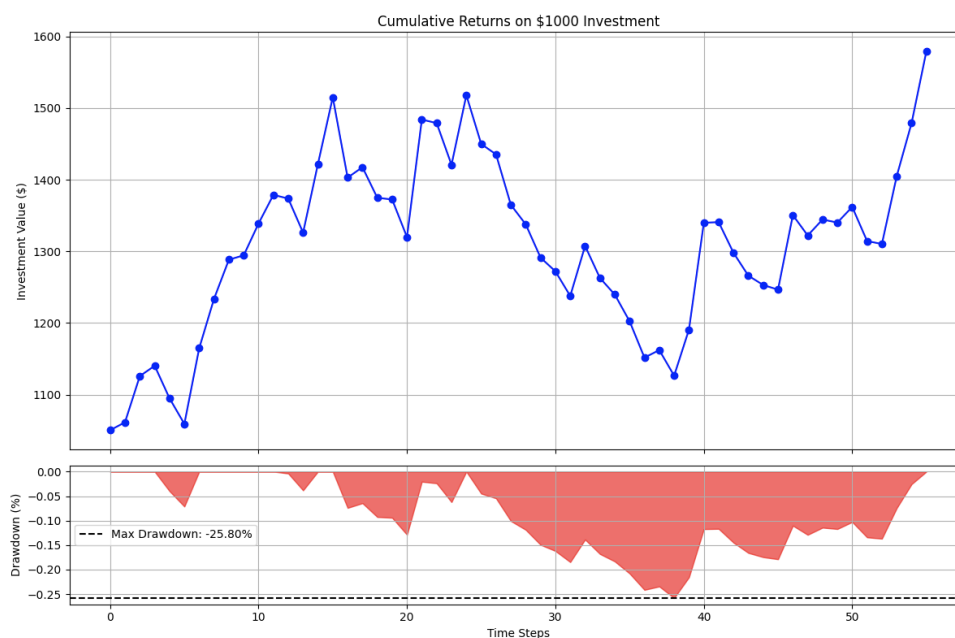


Figura 5.4: Perioada [2018-2025]. Funcția de fitness: Drawdown. Aceasta are ca scop alegerea indivizilor care maximizeaza expunerea la risc.

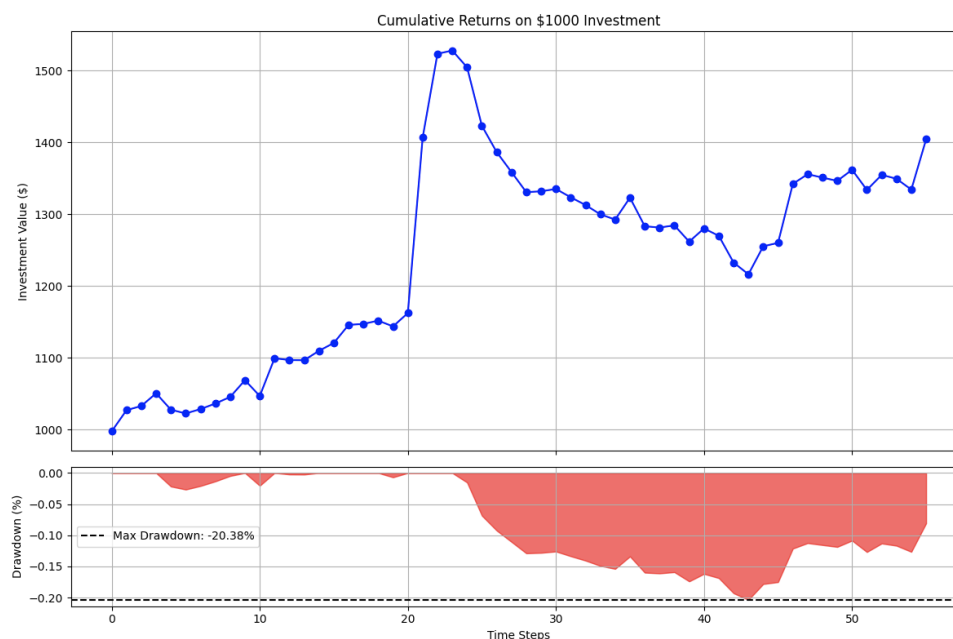


Figura 5.5: Perioada [2018-2025]. Funcția de fitness:  $\frac{\text{Return}}{\text{Drawdown}^{0,8}}$

Se observă că strategia propusă atinge un nivel crescut de stabilitate atunci când funcția de fitness ia în considerare atât randamentul (return), cât și riscul asociat (drawdown). Prin ridicarea drawdown-ului la puterea 0,8, se realizează un compromis controlat între performanță și risc, oferind o flexibilitate sporită în obținerea unor randamente superioare fără a penaliza excesiv variațiile negative. Acest tip de ponderare reflectă o abordare echilibrată în optimizarea strategiilor de tranzacționare.

## Training pe perioade lungi

Analizând cele patru funcții de fitness, se remarcă faptul că funcțiile bazate pe Drawdown și, în special, funcția  $\frac{\text{Return}}{\text{Drawdown}^{0,8}}$  oferă o perspectivă mai favorabilă asupra stabilității randamentelor.

Pentru a evalua performanța strategiei în diferite condiții de piață, a fost realizată o simulare utilizând **30** segmente pentru antrenare și **10** segmente pentru testare.

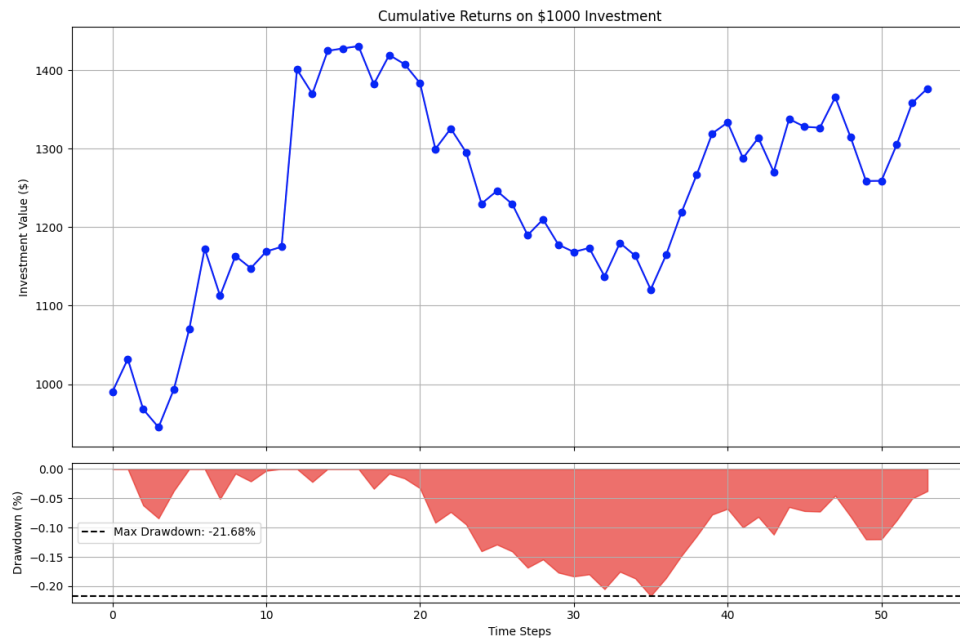


Figura 5.6: Perioada [2018-2025]. Funcția de fitness: Drawdown

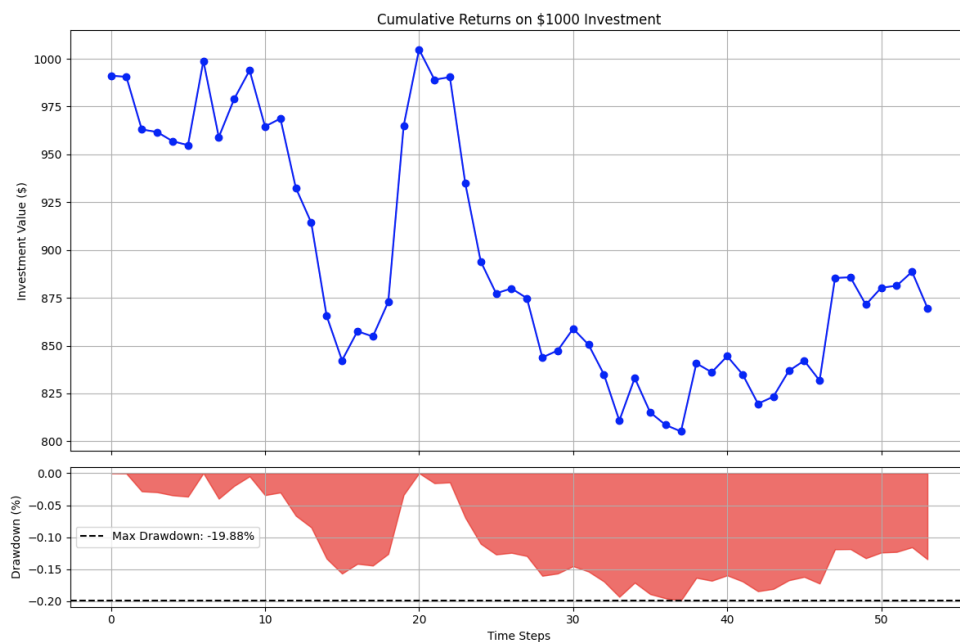


Figura 5.7: Perioada [2018-2025]. Funcția de fitness:  $\frac{\text{Return}}{\text{Drawdown}^{0,8}}$

## Training pe perioade scurte

Analizând performanțele obținute, s-a constatat că, atunci când perioada de antrenament este extinsă, modelul nu reușește să se adapteze eficient la condițiile specifice din perioada de testare. Fiind expus la o varietate largă de regimuri de piață, acesta tinde să se ajusteze simultan la multiple comportamente istorice, ceea ce conduce la o strategie generalistă, dar ineficientă.

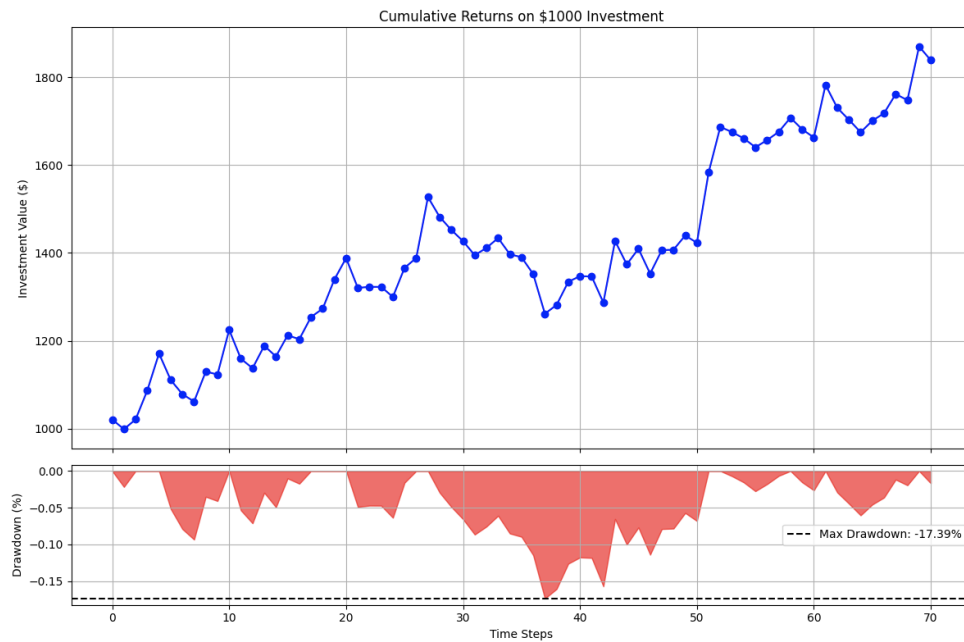


Figura 5.8: Perioada [2018-2025]. Funcția de fitness: Drawdown

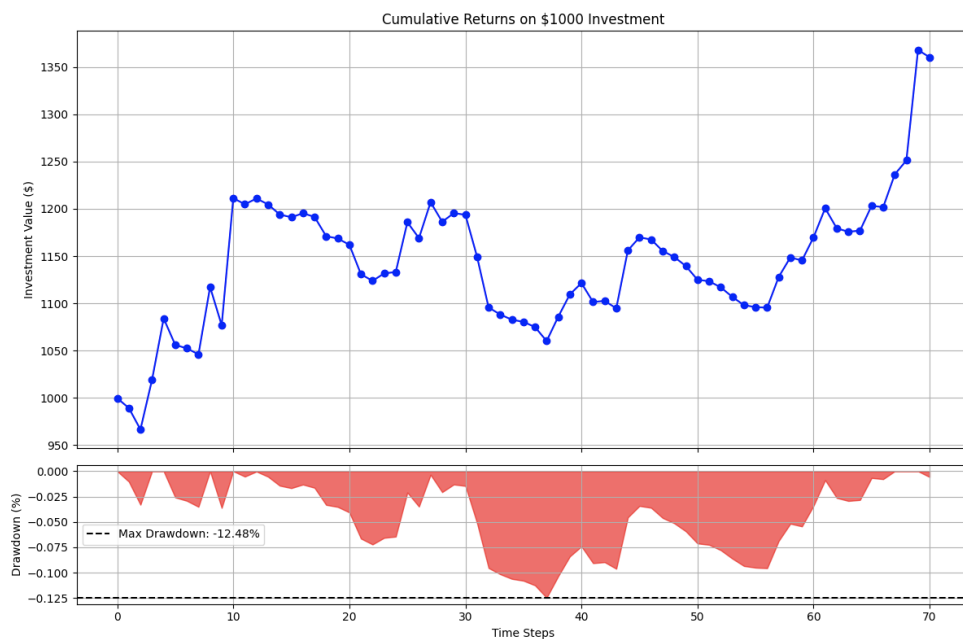


Figura 5.9: Perioada [2018-2025]. Funcția de fitness:  $\frac{\text{Return}}{\text{Drawdown}^{0,8}}$



Această suprageneralizare determină o scădere a performanței în faza de testare, în special atunci când regimul pieței se abate semnificativ de la cel predominant în perioada de antrenament. Prin urmare, adoptând o configurație cu un orizont de antrenare mai scurt, s-a observat o stabilizare progresivă a randamentelor în timp. (10 segmente antrenare, 8 segmente testare)

Așa cum a fost menționat anterior, în simulările inițiale s-a utilizat exclusiv **Double Exponential Moving Average (DEMA)** pentru calculul semnalelor de tranzacționare. Pentru a evalua performanța și în alte configurații, au fost rulate simulări suplimentare folosind aceeași funcție de fitness, **maximizarea drawdown-ului**, dar în combinație cu alte tipuri de medii mobile, și anume **Exponential Moving Average (EMA)** și **Weighted Moving Average (WMA)**. Acest demers a avut ca scop testarea robusteții funcției de fitness în raport cu natura funcției de netezire utilizate.

Se observă că toate cele trei tipuri de medii mobile (DEMA, EMA și WMA), prezintă un comportament similar în perioada analizată, manifestând drawdown-uri în aceleași intervale și evidențiind o evoluție ascendentă relativ apropiată. Această convergență a performanței sugerează că, în anumite condiții de piață, tipul mediei mobile utilizate are un impact redus asupra formei generale a strategiei, atunci când funcția de fitness este centrată pe drawdown.

Tipul de medie mobila	Sharpe Ratio	Return(%)	Max Drawdown(%)
Double Exponential	0.154	82.2	17.39
Exponential	0.168	87.8	16.17
Weighted	0.137	35.4	27.58

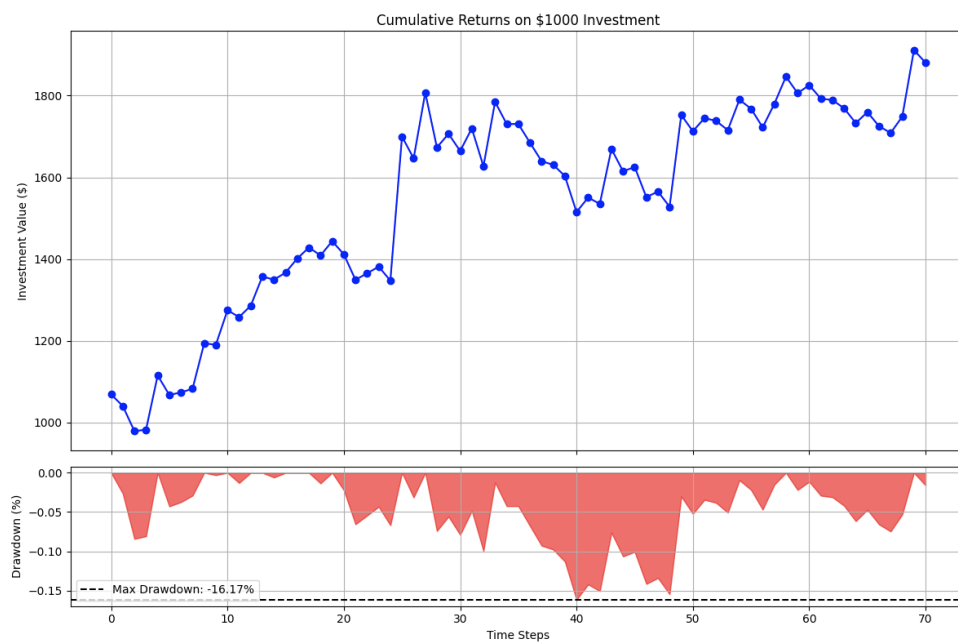


Figura 5.10: Perioada [2018-2025]. Funcția de fitness: Drawdown. Exponential Moving Average.

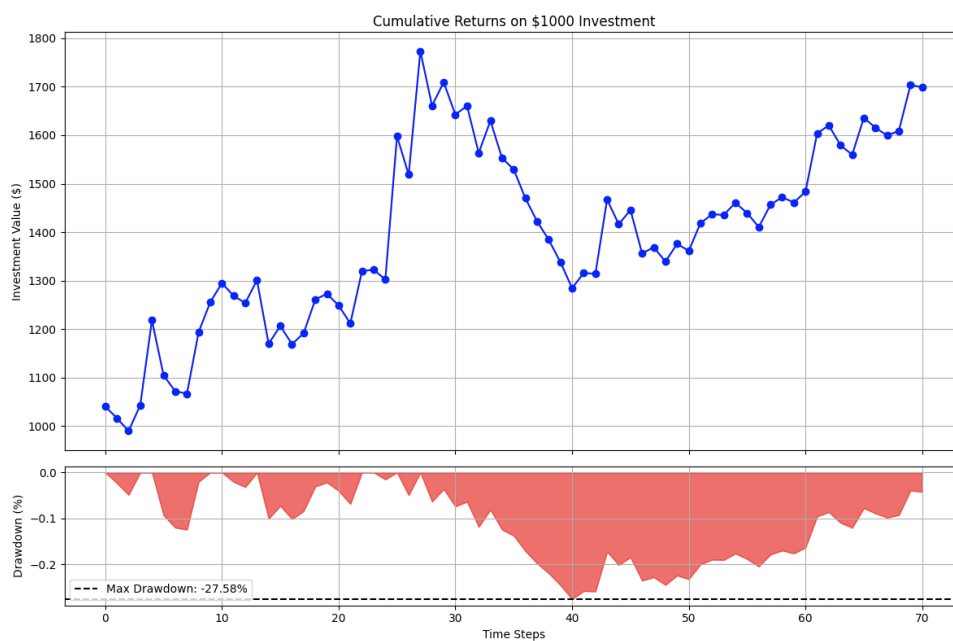


Figura 5.11: Perioada [2018-2025]. Funcția de fitness: Drawdown. Weighted Moving Average

# Capitolul 6

## Meta optimizare și evoluția adaptivă

### 6.1 Justificare: de ce meta optimizare?

În demersul de a identifica un set de parametri optimi pentru strategia de tranzacționare propusă, am extins abordarea prin introducerea unui proces de meta optimizare aplicat asupra algoritmului genetic utilizat. Scopul acestei extinderi este de a analiza direcția în care evoluează parametrii algoritmului și de a înțelege mai bine impactul acestora asupra performanței globale.

Setul inițial de parametri utilizat în teste a fost ales într-o manieră arbitrară, pe baza observațiilor empirice acumulate în timpul dezvoltării. Alegerea acestuia a avut în vedere obținerea unui timp de convergență redus, permițând astfel testarea rapidă pe o gamă variată de active financiare. În același timp, s-a urmărit menținerea unui nivel rezonabil de performanță, care să justifice continuarea experimentelor într-un cadru evolutiv controlat.

### 6.2 Structura meta algoritmului

Meta algoritmul propus funcționează ca un strat evolutiv, responsabil de optimizarea hiperparametrilor utilizați de algoritmul genetic de bază, care la rândul său optimizează strategiile de tranzacționare cu medii mobile. Fiecare individ din populația meta algoritmului reprezintă un set complet de hiperparametri ai algoritmului genetic de baza.

Acești hiperparametri sunt:

- POPULATION, dimensiunea populației algoritmului genetic de bază;

- `TOP_PICK`, numărul de indivizi selectați direct prin elitism;
- `WHEEL_OF_FORTUNE`, numărul de indivizi selectați prin selecție probabilistică;
- `RANDOM_INDIVIDUALS`, numărul de indivizi generați aleator în fiecare generație;
- `MUTATION_PROBABILITY`, probabilitatea aplicării unei mutații pe un offspring;
- `CROSSOVER_RATE`, rata de încrucișare utilizată între părinți;
- `NO_IMPROVEMENT`, numărul maxim de generații fără îmbunătățire;
- `MIN_IMPROVEMENT`, pragul minim de îmbunătățire necesar pentru a continua evoluția.

Fiecare parametru are o limită inferioară și superioară predefinită, conform unui dicționar denumit `GENE_RANGES`. Populația inițială este generată cu indivizi aleatori care respectă aceste constrângeri și condiții de validitate, precum: `WHEEL_OF_FORTUNE + RANDOM_INDIVIDUALS + TOP_PICK = POPULATION`.

Meta algoritmul evoluează această populație de configurații pe parcursul unui număr de generații, evaluând fiecare individ printr-o rulare efectivă a algoritmului genetic de bază cu setarea respectivă, folosind date de antrenament. Fitness-ul este calculat ca suma performanțelor strategiilor long și short.

Procesul evolutiv include selecția celor mai buni indivizi, crossover între configurații și mutații aleatorii, menținând astfel diversitatea și adaptabilitatea. Evaluarea indivizilor se face în paralel pentru a accelera execuția.

Structura codului este modulară, având funcții specializate pentru generarea indivizilor, evaluare, încrucișare și mutație, asigurând un cadru clar pentru meta optimizare adaptivă.

### 6.3 Evoluția parametrilor de optimizare

Pentru a putea obține rezultate într-un timp rezonabil și pentru a permite testarea pe mai multe active financiare (asseturi), meta algoritmul a fost rulat cu următoarea configurație a parametrilor:

- `population_size = 8`

- `generations = 4`
- `top_k = 2`
- `mutation_rate = 0.005`
- `crossover_rate = 0.02`
- `random_injection_ratio = 0.2`

În fiecare generație, cei mai buni `top_k` indivizi sunt păstrați (elitism). Ulterior, se aplică o selecție de tip *wheel of fortune* asupra restului populației, similar algoritmului genetic de bază. Pentru a menține diversitatea și a evita stagnarea prematură, o fracțiune din populație este completată prin inserarea aleatorie de meta indivizi noi, conform procentului definit de `random_injection_ratio`.

Astfel pe anumite perioade cheie (1 luna de date) meta algoritmul a reușit să găsească un set de configurări care să aducă în perioada de testare randamente ridicate și o fluctuație mică.

## Procesul de antrenare și testare pentru meta algoritmul genetic

Întregul proces de antrenare și testare implică o abordare ierarhică, unde un meta algoritmul genetic optimizează parametrii unui algoritmul genetic clasic, acesta din urmă fiind responsabil cu identificarea celor mai performante strategii de tranzacționare pe baza mediilor mobile.

- Considerăm o perioadă de **antrenare** alcătuită din  $T_{train}$  segmente (de exemplu,  $T_{train} = 10$ ) și o perioadă de **testare** cu  $T_{test}$  segmente (ex.  $T_{test} = 8$ ).
- Generăm o populație inițială de  $n = 8$  **indivizi meta**, fiecare reprezentând o configurație distinctă de parametri pentru algoritmul genetic de bază (dimensiunea populației, rata de mutație, rata de crossover etc.).
- Fiecare individ meta este evaluat prin rularea algoritmului genetic corespunzător pe întreaga perioadă de antrenare, scopul fiind găsirea celor mai buni parametri ai strategiei de tranzacționare pe baza mediei mobile.

- După evaluare, fiecărui individ meta i se asociază un **scor de fitness**, reflectând performanța strategiei obținute pe perioada de antrenare.
- Selectăm cei mai buni doi indivizi meta (cu scoruri de fitness maxime) și îi folosim pentru a genera o nouă populație, repetând procesul de selecție și reproducere timp de 4 generații.
- La final, alegem individul meta optim și utilizăm configurația sa pentru a efectua o ultimă optimizare a strategiei de tranzacționare pe întreaga perioadă de antrenare.
- Strategia obținută este ulterior testată pe perioada de testare pentru a evalua generalizarea și robustețea.

## 6.4 Rezultate experimentale pentru BTC

În urma rularii algoritmului descris anterior, s-a constatat că procesul de optimizare converge adesea către un set restrâns de configurații ale meta individului.

Mai jos sunt enumerate câteva dintre cele mai bune configurații identificate:

- (284, 6, 134, 144, 0.0997, 0.219, 9, 0.0097)
- (280, 18, 113, 149, 0.0159, 0.221, 7, 0.0099)
- (220, 12, 154, 54, 0.0780, 0.442, 8, 0.0097)
- (296, 18, 168, 110, 0.0756, 0.345, 6, 0.0041)

Reamintim semnificatia fiecarui element din tuplu prin urmatoarea secvență de cod:

```
constants.Genetic_Algorithm
    .Builder()
    .set_population_size(284)
    .set_top_pick(6)
    .set_wheel_of_fortune(134)
    .set_random_individual(144)
    .set_mutation_probability(0.0997)
```

```

.set_crossover_rate(0.219)
.set_no_improvement(9)
.set_min_improvement(0.0097)
.build();

```

Cu toate acestea, în urma analizelor comparative, nu s-a observat o îmbunătățire semnificativă a metricilor de performanță (randament, drawdown, Sharpe Ratio) comparativ cu strategiile anterioare sau cu parametrii selectați manual.

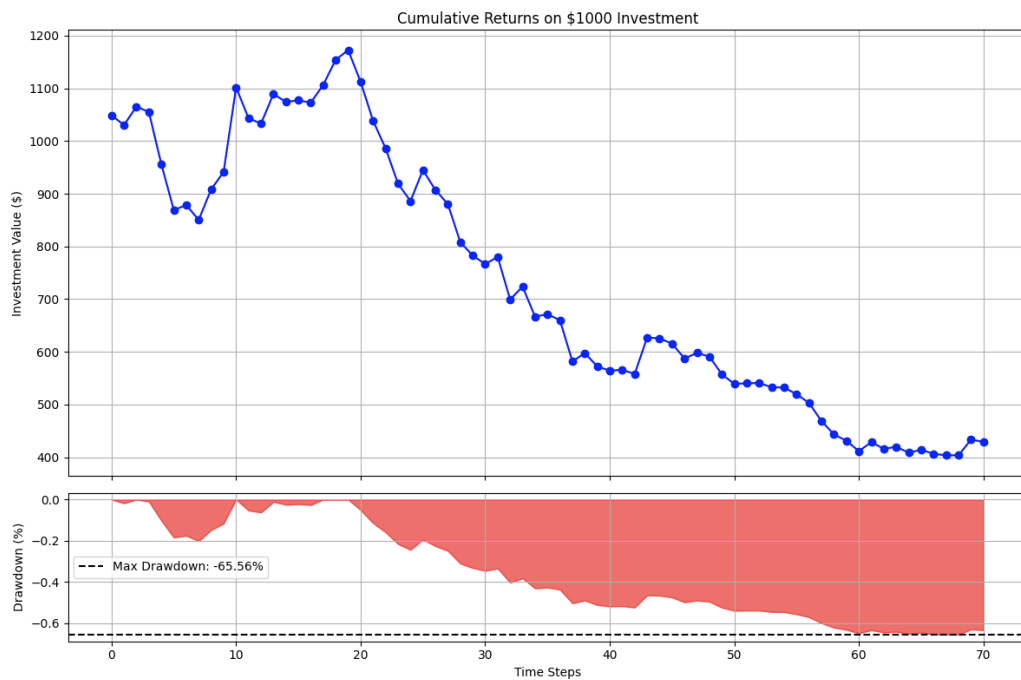


Figura 6.1: Performanța experimentală pentru perioada 2018-2025 pentru BTC, utilizând configurația (284, 6, 134, 144, 0.0997, 0.219, 9, 0.0097)

Această analiză subliniază necesitatea rafinării suplimentare a meta algoritmului, eventual prin integrarea altor factori de selecție sau metrice ce reflectă mai fidel caracteristicile regimurilor de piață.

# Concluzii

În încercarea de a identifica o metodă robustă pentru obținerea unor randamente constante pe piața bursieră, această lucrare a condus la o aprofundare semnificativă a cunoștințelor în domeniul financiar, precum și în cel al optimizării strategiilor de tranzacționare.

Deși nu am reușit să dezvoltăm o metodă care să garanteze randamente constante sau să obținem un Sharpe Ratio superior strategiei clasice de *Buy and Hold*, am reușit să evidențiem comportamente esențiale ale pieței și să identificăm limitările abordării propuse. Aceste constatări pot fi considerate, în parte, drept consecințe ale constrângerilor impuse de utilizarea algoritmilor genetici.

## Observații

Un aspect critic remarcat în urma testării folosind *walk forward testing* constă în sensibilitatea algoritmului genetic la schimbările regimurilor de piață. Mai precis, dacă în perioada de antrenament prețul urmează un trend ascendent cu o volatilitate moderată, iar în perioada de testare acesta devine descendent, strategiile optimizate anterior nu mai performează eficient. Această nepotrivire se traduce într-un *drawback* semnificativ al algoritmilor care nu țin cont de dinamica regimului de piață.

Dacă alegem perioade lungi de antrenament, algoritmul tinde să uniformizeze parametrii strategiilor pentru a acoperi o gamă largă de condiții de piață. Acest lucru duce, inevitabil, la convergența către un set stabil de parametri, dar care nu oferă o adaptabilitate reală. Așa cum s-a discutat în capitolele anterioare, strategiile cu parametri fixați nu pot performa sustenabil pe termen lung.

Pe de altă parte, reducerea perioadei de antrenament la intervale scurte sau împărțirea datelor în secțiuni multiple pare să reducă influența regimurilor de piață variabile. În aceste condiții, strategiile pot demonstra o performanță mai bună în testare, însă acest



comportament este deseori o consecință a unei hiper-optimizări asupra trecutului, fără garanții pentru viitor.

## Direcții de Îmbunătățire

Prețul unui activ financiar are două componente fundamentale: direcția (*trendul*) și volatilitatea. În cadrul lucrării de față, aceste două componente au fost tratate împreună, însă observațiile indică faptul că o separare clară a acestora ar putea duce la o strategie mai eficientă. Volatilitatea, fiind în esență impredictibilă, introduce un nivel ridicat de incertitudine. În schimb, direcția poate fi exploatată mai eficient prin inferență contextuală.

O posibilă direcție viitoare ar fi integrarea lanțurilor Markov ascunse (*Hidden Markov Models*) pentru a estima probabilități asociate anumitor niveluri de preț sau regimuri de piață. Astfel, strategiile de tranzacționare s-ar putea adapta în timp real la condițiile estimate ale pieței, crescând șansele de performanță față de abordările rigide și fixe.

Această direcție ar putea oferi un echilibru între robustețe și adaptabilitate, elemente esențiale în construirea unei strategii algoritmice eficiente în condiții de piață reale și variabile.

# Bibliografie

- [1] W. Chen and Z. Zhu, "Optimizing MACD Trading Strategies: A Dance of Finance, Wavelets, and Genetics," *arXiv preprint arXiv:2501.10808*, 2025.
- [2] R. Menoita and S. Silva, "Evolving Financial Trading Strategies with Vectorial Genetic Programming," *arXiv preprint arXiv:2504.05418*, 2025.
- [3] A. Rahman, "Optimising Supertrend Parameters Using Bayesian Optimization For Maximising Profit And Other Metrics," *arXiv preprint arXiv:2405.14262*, 2024.
- [4] L. Huang, "NEAT Algorithm-based Stock Trading Strategy with Multiple Technical Indicators Resonance," *arXiv preprint arXiv:2501.14736*, 2025.
- [5] M. Noguer i Alonso, H. Agrawal, and D. P. Aznar, "Deep Reinforcement Learning: Policy Gradients for US Equities Trading," *SSRN Electronic Journal*, Elsevier, 2023. [Online]. Available: <https://ssrn.com/abstract=4645453>
- [6] Y. Luo and Z. Duan, "Agent Performing Autonomous Stock Trading Under Good and Bad Situations," *arXiv preprint arXiv:2306.03985*, 2023.
- [7] X. Sha, "Time Series Stock Price Forecasting Based on Genetic Algorithm (GA)-Long Short-Term Memory Network (LSTM) Optimization," *arXiv preprint arXiv:2405.03151*, 2024.
- [8] J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*, New York, NY: New York Institute of Finance, 1999.
- [9] Andreas F. Clenow, *Following the Trend: Diversified Managed Futures Trading*, Hoboken, NJ: John Wiley & Sons, 2013.
- [10] J. W. Wilder Jr., *New Concepts in Technical Trading Systems*, Greensboro, NC: Trend Research, 1978.

- [11] J. Bollinger, *Bollinger on Bollinger Bands*, New York, NY: McGraw-Hill, 2001.
- [12] W. F. Sharpe, "The Sharpe Ratio," *The Journal of Portfolio Management*, vol. 21, no. 1, pp. 49-58, 1994.
- [13] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.
- [14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.
- [15] E. Croitoru, *Noțiuni: Algoritmi Genetici*, Universitatea „Alexandru Ioan Cuza” din Iași. [Online]. Available: <https://profs.info.uaic.ro/eugen.croitoru/teaching/ga/>
- [16] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Berlin, Germany: Springer, 2003.
- [17] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed., Berlin, Germany: Springer, 2015.
- [18] I. Loshchilov and M. Schoenauer, "Adaptive Coordinate Descent with Bandwidth Selection for High-Dimensional Noisy Black-Box Optimization," 2012.
- [19] W. Brock, J. Lakonishok, and B. LeBaron. *Simple Technical Trading Rules and the Stochastic Properties of Stock Returns*. *Journal of Finance*, 47(5):1731-1764, 1992.
- [20] V. K. Tharp, *Trade Your Way to Financial Freedom*, 2nd ed., New York, NY: McGraw-Hill, 2007.
- [21] J. W. Wilder Jr., *New Concepts in Technical Trading Systems*, Greensboro, NC: Trend Research, 1978.
- [22] R. S. Tsay, *Analysis of Financial Time Series*, 3rd ed., Hoboken, NJ: John Wiley & Sons, 2010.
- [23] R. Kukacka, *backtesting.py - Python Backtesting Library*. [Online]. Available: <https://kernc.github.io/backtesting.py/>
- [24] M. López de Prado, *Advances in Financial Machine Learning*, Hoboken, NJ: Wiley, 2018.

- [25] N. Anugrah, *Bitcoin Historical Datasets (2018-2025)*. [Online]. Available: <https://www.kaggle.com/datasets/novandraanugrah/bitcoin-historical-datasets-2018-2024>
- [26] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. [Online]. Available: <https://matplotlib.org/>