

UNIVERSITATEA "ALEXANDRU IOAN CUZA" IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

XYZ

*descriere XYZ*

propusă de

*Denise-Mihaela Goldan*

Sesiunea: Iulie, 2018

Coordonator științific

Conf. dr. Adrian Iftene

UNIVERSITATEA "ALEXANDRU IOAN CUZA" IAȘI  
FACULTATEA DE INFORMATICĂ

XYZ

*Denise-Mihaela Goldan*

**Sesiunea:** Iulie, 2018

**Coordonator științific**  
**Conf. dr. Adrian Iftene**

## **Declarație privind originalitatea și respectarea drepturilor de autor**

Prin prezenta declar că Lucrarea de licență cu titlul "XYZ" este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referință precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imagini etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, XY Iulie 2018

Denise-Mihaela Goldan

## **Declarație de consințământ**

Prin prezența declar că sunt de acord ca Lucrarea de licență cu titlul "XYZ", 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 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.

Iași, XY Iulie 2018

Denise-Mihaela Goldan

# Cuprins

<b>Problema iterata a prizonierului</b>	<b>6</b>
1 Dilema prizonierului . . . . .	6
2 Problema iterată a prizonierului . . . . .	7
3 Strategii pentru problema iterată a prizonierului . . . . .	8
<b>Algoritm genetic</b>	<b>9</b>
1 Apariția noțiunii de algoritm genetic . . . . .	9
2 Terminologie . . . . .	10
3 Pseudocod . . . . .	11
<b>Dezvoltarea unei strategii folosind un algoritm genetic</b>	<b>12</b>

# Problema iterată a prizonierului

## 1 Dilema prizonierului

Dilema prizonierului<sup>1</sup> reprezintă o problemă tratată în teoria jocurilor. A fost formulată de către Merrill Flood and Melvin Dresher, angajați ai companiei RAND Corporation<sup>2</sup>, în 1950. Denumirea a fost dată de Albert W. Tucker, de la Universitatea Princeton, care a formalizat jocul și a introdus noțiunea de răsplată (engl. "payoff").

Enunțul clasic al problemei este următorul: doi suspecți sunt arestați de către poliție. Polițiștii nu au suficiente dovezi pentru a condamna suspectii așa că îi duc în camere separate și le propun aceeași ofertă amândurora. Dacă unul dintre suspecți depune mărturie pentru urmărirea penală împotriva celuilalt suspect și celălalt tănuiește faptele, cel care a trădat este eliberat și cel care a tănuit primește o pedeapsă de 10 ani de închisoare. Dacă ambii suspecți nu mărturisesc, ambii ajung în pușcarie pentru jumătate de an. Dacă se trădează reciproc, fiecare primește o pedeapsă de 5 ani. Suspecții au de ales între a trăda și a tănui faptele.

Putem formaliza această formă prin următoarea matrice a recompenselor<sup>3</sup>:

---

<sup>1</sup>Adaptare după "*Prisoner's Dilemma: Game Theory*", Merrill M. Flood, Melvin Dresher, Albert W. Tucker, Framing Device, Experimental Economics

<sup>2</sup><https://www.rand.org/>

<sup>3</sup>Preluat din <https://plato.stanford.edu/entries/prisoner-dilemma/>

	B tăinuiește	B mărturisește
A tăinuiește	A: "Reward" B: "Reward"	A: "Sucker's payoff" B: "Temptation"
A mărturisește	A: "Temptation" B: "Sucker's payoff"	A: "Punishment" B: "Punishment"

Tabelul 1: Matricea recompenselor pentru dilema prizonierului

Termenii care apar în tabel sunt următorii:

- **Temptation:** recompensa obținută de jucatorul ce mărturisește atunci când celalalt tăinuiește faptele
- **Reward:** recompensa pentru când cei doi "suspecți", A și B, aleg să tăinuiască
- **Punishment:** pedeapsa obținută de cei doi suspecți atunci când se trădează reciproc
- **Sucker's payoff:** pedeapsa pentru cel care a tăinuit atunci când celălalt l-a trădat

Între acești termeni, se respectă următorul lanț de inegalități:

**Temptation > Reward > Punishment > Sucker's payoff**

## 2 Problema iterată a prizonierului

În teoria jocurilor, problema iterată a prizonierului este catalogat drept joc cu suma nenulă <sup>4</sup> (engl. "non-zero-sum game").

– dacă se cunoaște numărul de iterații, o strategie bună e să tradezi la ultimul meci. Repetând raționamentul, se poate trada și la penultimul joc.

.....

---

<sup>4</sup>Numim joc de suma nenulă jocul în care suma câștigurilor este diferită de zero.

### 3 Strategii pentru problema iterată a prizonierului

Considerând acest scenariu drept un joc, folosim termenul de cooperare (engl. "cooperation") pentru a descrie situația când unul dintre suspecți tănuiește faptele. Mărturisirea faptelor de către un suspect va fi numită trădare (engl. "defection").

- **Always cooperate:** Jucătorul cooperează la fiecare rundă a jocului, indiferent de strategia aplicată de celălalt jucător.
- **Always defect:** Jucătorul trădează la fiecare rundă a jocului.
- **Grudger:** Această strategie presupune cooperarea la fiecare rundă, până la prima trădare din partea celuilalt jucător. Așadar, adoptând această strategie, dacă oponentul trădează chiar și o singură dată, următoarele mișcări, până la final de joc, vor fi de trădare.
- **Pavlov:** Se alege cooperarea la prima rundă. Dacă la runda anterioară jucătorul a fost recompensat cu "Temptation"<sup>5</sup> sau "Reward"<sup>6</sup>, acesta repetă ultima mișcare. În celălalt caz, alege mișcarea opusă.
- **Random:** Se alege la întâmplare următoarea acțiune.
- **Tit-For-Tat:** Se alege cooperarea la prima rundă. De la runda a doua, jucătorul ce alege această strategie repetă ultima mișcare a oponentului.
- **Suspicious Tit-For-Tat:** Diferența dintre această strategie și **Tit-For-Tat** este că la prima mișcare se alege trădarea.
- **Tit-For-Two-Tats:** Jucătorul cooperează de fiecare dată, făcând excepție acele cazuri în care jucătorul este trădat de două ori consecutiv.

---

<sup>5</sup>"Temptation" este recompensa obținută de jucătorul ce trădează atunci când oponentul cooperează.

<sup>6</sup>"Reward" reprezintă recompensa primită de ambii jucători atunci când cooperează.



# Algoritm genetic

## 1 Apariția noțiunii de algoritm genetic

Algoritmii genetici<sup>7</sup> au fost introduși de către John Holland în 1960 și dezvoltati, ulterior, alături de colegii de la Universitatea din Michigan, între anii 1960 și 1970. Holland urmărea înțelegerea fenomenului de "adaptare" întâlnit în natură și implementarea unor mecanisme adaptive care să fie utilizate în practică, în contextul programării. Cartea publicată de acesta în 1975, "*Adaptation in Natural and Artificial Systems*" (Holland, 1975/1992) prezintă algoritmii genetici drept abstractizări ale evoluției biologice, și oferă un cadru teoretic pentru dezvoltarea acestora.

Algoritmii genetici ai lui Holland sunt metode de a trece de la o populație de "cromozomi" (e.g., șiruri de "biți" care reprezintă soluții candidate pentru o problemă) la o nouă populație, prin folosirea "selecției", alături de operatorii inspirați din genetică: încrucișare, mutație, inversiune. Cea din urmă este rar folosită în practică.

*"Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand."*

by John H. Holland

Algoritmii genetici au fost creați în încercarea de a imita procese specifice evoluției naturale, cum ar fi lupta pentru supraviețuire și moștenirea materialului genetic. Putem privi evoluția drept strategia abordată de speciile biologice pentru a căuta "soluții" cât mai potrivite, adaptate condițiilor schimbătoare, într-un număr foarte mare de posibilități. Această abordare

---

<sup>7</sup>Adaptare după textul "Genetic Algorithms: An Overview", Melanie Mitchell.

poate fi utilizată în rezolvarea problemelor de optimizare, atunci când metodele clasice exhaustive nu se dovedesc eficiente.

Noțiunea de "algoritm genetic" nu este definită în mod riguros[1], însă toate metodele ce poartă această denumire au în comun următoarele: populația este formată din cromozomi, selecția este făcută pe baza rezultatelor funcției de optimizat, încrucișarea a doi *cromozomi părinți* produce 2 *cromozomi copii*, mutația se aplică *cromozomilor copii*.

## 2 Terminologie

- Soluțiile candidat sunt adesea codificate în forma unor șiruri de biți și se mai numesc **cromozomi** sau **indivizi** ai populației. Fiecare bit este echivalentul unei gene.
- Genele sunt informațiile stocate de către cromozomi.
- **Populația**, care va fi urmărită în procesul său evolutiv, este alcătuită din mai mulți cromozomi.
- Fiecare **generație** marchează câte o etapă din evoluția populației inițiale.
- Pentru a trece de la o generație la alta, apelăm la noțiunea de **reproducere**. În alcătuirea următoarei generații, se pornește de la populația actuală, pe care o supunem unui proces de **selecție**. Pentru a face analogia cu fenomenul de supraviețuire a celor mai adaptați indivizi, măsurăm cromozomii cu ajutorul unei **funcții de optimizat**. O valoare ridicată a acestei funcții este interpretată ca o bună adaptare la mediu a individului.
- Pentru explorarea spațiului de soluții, indivizii selectați suferă modificări. Sunt supuși **încrucișărilor** și **mutațiilor**.
- Încrucișarea combină genele a doi *cromozomi părinți*, rezultând doi **moștenitori**. Există mai multe variante: cu un punct de tăiere, ales aleator (în care un moștenitor este alcătuit dintr-o porțiune de cromozom de la primul părinte și o porțiune de la al doilea), cu mai multe puncte de tăiere și uniformă (unde fiecare genă este selectată probabilist de la unul din cei 2 *cromozomi părinți*).

- Mutația alterează gene alese arbitrar dintr-un cromozom. Numărul de gene afectate poate varia.

### 3 Pseudocod

```
inițializează cu valori aleatorii populația inițială
calculează valoarea funcției de optimizat pentru indivizii populației
cât timp nu s-a îndeplinit condiția de oprire
    aplică o metodă de selecție, pentru a crea populația
    aplică operatorul genetic încrucișare, cu o anumită probabilitate
    aplică operatorul genetic mutație, cu o anumită probabilitate
    calculează valoarea funcției de optimizat pentru indivizii populației
```

Condiția de oprire poate fi atingerea unui număr de iterații stabilit inițial. De asemenea, se poate stabili ca algoritmul să se oprească atunci când nu se mai înregistrează îmbunătățiri în ceea ce privește calitatea soluțiilor furnizate.

Soluția returnată de un algoritm genetic reprezintă cel mai bun individ întâlnit în evoluția populației.

## Dezvoltarea unei strategii folosind un algoritm genetic

# Bibliografie

- [1] Abcdef *An introduction to Genetic Algorithms*  
Melanie Mitchell
- [2] Melanie Mitchell. *Genetic Algorithms: An Overview*. 1995.
- [3] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German)  
[*On the electrodynamics of moving bodies*]. Annalen der Physik,  
322(10):891–921, 1905.
- [4] Knuth: Computers and Typesetting,  
<http://www-cs-faculty.stanford.edu/~uno/abcde.html>