

Universitatea București
Facultatea de Matematică și Informatică

**Modele de Calcul Masiv Paralel
Inspirate din Biologie
-Rezumat-**

doctorand: Ștefan Popescu

îndrumător: Prof. Dr. Victor Mitrana

1 Introducere

Cu douăzeci de ani în urmă, experimentul realizat de către Adleman - rezolvarea cu succes a problemei comis-voiajorului prin folosirea ADN-ului și a instrumentelor de bio-inginerie - a dovedit că biologia poate fi o sursă de inspirație pentru proiectarea de modele computaționale, dar de asemenea poate ajuta, prin furnizarea unei noi arhitecturi pentru calculatoare, a unei alternative la hardware-ul tradițional pe bază de silicon. În ultimele două decenii s-a dezvoltat în mod dinamic o nouă ramură în informatică numită "Natural Computing".

Principala diferență dintre cele două modele, cele bio-inspirate și cele tradiționale, constă în faptul că operațiile caracteristice celor dintâi sunt inspirate din evoluția celulelor vii și se desfășoară într-un mod masiv paralel. Modelele de calcul de inspirație biologică au potențialul de a rezolva probleme dificil de rezolvat într-un timp rezonabil, profitând de paralelismului masiv făcut posibil de modul în care ADN-ul stochează și procesează informația la nivel molecular.

2 Preliminarii

Rețele de Procesoare Evoluționiste (NEP) ca dispozitive generatoare de limbaje și ca mijloace de rezolvare de probleme au fost descrise în [6] și [18]. NEP-urile, ca dispozitive acceptoare, au fost studiate în [17]; mai târziu, o caracterizare a claselor de complexitate **NP**, **P** și **PSPACE** bazată pe NEP-uri a fost formulată în [14]. NEP universale și unele probleme de complexitate sunt discutate în [13]. Principalele rezultate pot fi găsite în [16].

Au fost realizate implementări software de NEP-uri, a se vedea [7, 8, 12], cele mai multe dintre ele scrise în Java. S-au întâmpinat dificultăți în special în implementarea procesului de filtrare. Ideea principală pentru a simula comportamentul non-deterministic al NEP-urilor a fost să se ia în considerare un model de procesoare, care atribuie fiecărei reguli și fiecărui filtru câte un fir de execuție. În mod evident firele corespunzătoare filtrelor sunt mult mai complicate decât cele asociate operațiilor evolutive. Schimbările de configurație sunt realizate fie printr-un pas de comunicare, fie la un pas evolutiv, dar aceste două etape pot

fi realizate în orice ordine. Acest fapt sugerează că operația de evoluție sau de comunicare poate fi aleasă în funcție de tipul firului asociat procesorului [8]. Filtrele de intrare și ieșire sunt implementate prin extinderea interfeței `Runnable`. Prin urmare, un procesor este părintele unui set de fire de execuție care folosesc toate obiectele asociate procesorului. Când un procesor își începe execuția, se începe rularea, într-un mod cascadă, a firelor asociate regulilor și a firelor de filtrare. După cum se poate vedea, filtrele asociate procesoarelor, mai ales în cazul în care există filtre atât de intrare și cât și de ieșire, par a fi mai greu de implementat. Prin urmare, se propune înlocuirea comunicării bazate pe filtre cu un alt protocol de comunicare. O primă încercare a fost mutarea filtrelor de pe nodurile rețelei pe muchiile acestuia, a se vedea, de exemplu, [9]. Deși această variantă pare, pentru început, mai simplă de implementat, încercările de a o pune în aplicare au întâmpinat dificultăți asemănătoare, din cauza faptului că filtrele asociate muchiilor sunt similare cu cele asociate nodurilor.

În [1] se definește o nouă variantă de NEP ce implică un protocol nou de comunicare și se discută potențialul acestui nou model pentru rezolvarea de probleme. Caracteristica principală și complet nouă a acestei variante este funcția de evaluare, care atribuie fiecărui șir o valoare întreagă, în funcție de valorile atribuite simbolurilor sale. De fapt, nu suntem interesați de valoarea exactă asociată unui șir de caractere, ci doar de semnul acestei valori. Prin intermediul acestei evaluări, se poate spune metaforic că șirurile sunt polarizate electric. Astfel, în cazul în care nodurile sunt și ele polarizate, migrarea cuvintelor de la un nod la altul în funcție de polaritatea celor două noduri pare să fie mai naturală și mai ușor de implementat.

În ceea ce privește limbajele 2-dimensionale, două variante de rețele de procesoare evoluționiste pe picturi au fost definite în [4] și [5]. Aceste studii includ o generalizare 2-dimensională pentru rețele de procesoare evoluționiste cu protocoale de comunicare bazate pe filtre și compară clasele de limbaje rezultate cu clasa de limbaje cu picturi care poate fi local acceptată de către o mulțime de tile-uri (plăci) și cea care poate fi recunoscută de către un sistem de placare. Deși aceste modele par a fi în măsură să accepte o mare varietate de limbaje (de exemplu, complementul oricărui limbaj local, unele limbaje care nu pot fi

recunoscute de către un sistem de placare), sunt prezente anumite limitări, în special cu privire la problema potrivirii 2-dimensionale: anume dacă o pictură conține un anumit model ca sub-pictură. Soluția este prezentă doar pentru cazul în care modelul este format din cel mult trei coloane sau trei rânduri.

3 Structura Tezei

Această teză este împărțită în 5 capitole, după cum urmează:

1. În capitolul introductiv se prezintă motivația ce stă la baza studiului nostru.
2. În capitolul al doilea definim noțiunile și principiile de bază care vor fi folosite pe parcursul acestei teze. Acest capitol este împărțit în 5 secțiuni:
 - Prima secțiune conține definițiile și notațiile de bază cu privire la șiruri de caractere.
 - În partea a doua prezentăm noțiunea de mașină Turing și dăm câteva exemple de diferite modele de mașini Turing.
 - În a treia secțiune descriem pe scurt noțiunea de complexitate timp și definim formal unele clase de complexitate mai importante.
 - În secțiunea a patra discutăm despre clasa de limbaje *NP-complete*.
 - În ultima secțiune definim noțiunea de sisteme de tag-uri ca fiind un model de calcul determinist universal.
3. În capitolul al treilea, intitulat **Rețele de Procesoare Evoluționiste Polarizate**, avem în vedere puterea de calcul a unei noi variante de rețele de procesoare evoluționiste. Fiecare procesor, precum și șirurile care circulă prin întreaga rețea, sunt acum considerate a fi polarizate. În timp ce polarizarea fiecărui procesor este predefinită, polarizarea șirurilor este calculată în mod dinamic prin intermediul unei funcții de evaluare. Acest capitol este structurat după cum urmează:

- Pentru început oferim câteva informații generale despre modele de calcul inspirate din biologie, motivația din spatele unei astfel de cercetări, iar apoi ne referim la alte modele de calcul bazate pe procesoare evoluționiste.
 - În a doua secțiune definim formal noțiunea de *procesor evoluționist polarizat* și a rețelelor bazate pe acest tip de procesor împreună cu un protocol de comunicare specific acestora.
 - În a treia secțiune demonstrăm că *Rețelele de Procesoare Evoluționiste Polarizate* sunt universale, ceea ce înseamnă că toate funcțiile algoritmice calculabile pot fi calculate cu ajutorul acestor rețele. În primul rând demonstrăm că sistemele de tag-uri pot fi simulate de o astfel de rețea cu un număr constant de noduri, în timp ce o mașină Turing poate fi simulată, într-un mod eficient, folosind o rețea cu un număr de noduri liniar dependent de alfabetul de bandă al mașinii simulate [2]. Dacă se dorește simularea unei mașini Turing folosind o rețea cu număr constant de noduri, acest lucru poate fi realizat, dar cu o creștere considerabilă a timpului de calcul [3]. În cele din urmă arătăm că fiecare rețea poate fi simulată de către o mașină Turing și discutăm despre costurile de timp ale unei astfel de simulări [3].
 - În cea de-a patra secțiune încercăm să îmbunătățim rezultatul anterior. Simulăm o mașină Turing folosind o Rețea de Procesoare Evoluționiste Polarizate cu un număr constant de noduri și care are polaritatea fiecărui simbol, definită anterior ca fiind un număr întreg, limitată la mulțimea de valori $\{-1, 0, 1\}$ [19].
 - În ultima secțiune vom exemplifica modul de lucru al Rețelelor de Procesoare Evoluționiste Polarizate prin construirea unei astfel de rețele care rezolvă problema CNF-SAT.
4. În al patrulea capitol, **Calcul pe Limbaje 2-Dimensionale**, definim un nou model de calcul de inspirație biologică care să poată accepta unele limbaje 2-dimensionale în mod similar cu cele prezentate în [4] și [5]. Factorul de noutate fiind protocolul de comunicare bazat pe polaritatea asociată simbolurilor prezente pe rama picturii. Acest model poate fi

privit ca o extensie 2-dimensională a celui prezentat în [2]. Acest capitol este structurat după cum urmează:

- În prima secțiune enunțăm motivația generalizării conceptului de procesoare evoluționiste polarizate pentru lucrul cu picturi.
 - A doua secțiune conține definițiile de bază referitoare la limbajele 2-dimensionale împreună cu câteva exemple.
 - În a treia secțiune descriem formal clasa de limbaje care pot fi acceptate la nivel local de către un set de plăci și clasa de limbaje care pot fi recunoscute de către un sistem de placare.
 - În ultima secțiune prezentăm un nou model de calcul care lucrează pe cuvinte 2-dimensionale. Anume, extindem conceptul de Rețele de Procesoare Evoluționiste Polarizate descrise în capitolul 3 pentru a include lucrul cu picturi dreptunghiulare. Aceste rețele au un protocol de comunicare modificat care asociază fiecărei picturi o cvadruplă cu elemente din mulțimea $\{-, 0, +\}$. Considerăm aceasta ca fiind o generalizare 2-dimensională a noțiunii de polaritate definită anterior pe șiruri. Comparăm clasa de limbaje care pot fi acceptate de modelul nostru cu cele care pot fi acceptate la nivel local de către un set de plăci și cu cele care sunt recunoscute de către un sistem de placare. În cele din urmă dăm o soluție parțială pentru problema de potrivire 2D-dimensională. Considerăm doar cazul în care modelul este o imagine care este alcătuită din cel mult trei rânduri sau trei coloane. Această ultimă secțiune se bazează în principal pe rezultatele publicate în [20].
5. În ultimul capitol, **Concluzii**, prezentăm un scurt rezumat al rezultatelor enunțate și enumerăm o listă de probleme deschise.

4 Rezultate Principale

4.1 Informații Generale

Rețele de Procesoare Evoluționiste (NEP) sunt definite folosind un graf ne-orientat unde fiecare nod conține un procesor capabil să efectueze operații evoluționiste simple (inserare, ștergere sau substituție). Procesoarele plasate în nodurile adiacente pot comunica între ele conform unui protocol predefinit.

În această teză analizăm o variantă mai recentă de NEP-uri, și anume cea alcătuită din Procesoare Polarizate (NPEP) [1]. Această clasă de procesoare diferă de cele menționate în lucrări anterioare precum [16]. În definirea protocolului de comunicare se abandonează metoda bazată pe filtre și se introduce conceptul de nod și cuvânt polarizat.

4.2 Enumerarea Rezultatelor

Deși protocolul de comunicare caracteristic procesoarelor polarizate precum și funcția de evaluare par să ofere mai puțin control, noua variantă este încă universală. Demonstrăm că o NPEP cu un număr constant de noduri, respectiv 15, este universală din punct de vedere al puterii de calcul. Rezultatul acesta este obținut prin elaborarea unei metode pentru simularea sistemelor 2-tag.

Teoremă 4.1. *Pentru orice sistem 2-tag $T = (V, \mu)$ există o Rețea de Procesoare Polarizate Γ cu 15 noduri astfel încât $L(\Gamma) = \{w \mid T \text{ se oprește pe intrarea } w\}$.* [2]

Deoarece un sistem 2-tag poate simula eficient orice mașină Turing deterministă, dar nu și pe cele nedeterministe, ne propunem o simulare a unei mașini Turing nedeterministe folosind o NPEP care să se încadreze în timpul de lucru al mașinii Turing.

Teoremă 4.2. *Pentru orice limbaj recursiv enumerabil L , acceptat în timp $\mathcal{O}(f(n))$ de către o mașină Turing cu alfabetul de lucru U , există o Rețea de Procesoare Polarizate alcătuită din $10 \cdot \text{card}(U)$ noduri care acceptă limbajul L în timpul de lucru $\mathcal{O}(f(n))$.* [2]

Spre deosebire de simularea unui sistem 2-tag, numărul de noduri al unei NPEP pentru simularea unei mașini Turing depinde liniar de cardinalul alfabetului de lucru al mașinii Turing, dar se păstrează complexitatea timp. Dacă vrem să simulăm o mașină Turing cu o rețea de dimensiuni constante, acest lucru este posibil, însă duce la o creștere considerabilă a timpului de calcul.

Teoremă 4.3. *Pentru orice limbaj recursiv enumerabil L , acceptat în timp $\mathcal{O}(f(n))$ de către o mașină Turing cu alfabetul de lucru U , există o NPEP cu 39 de noduri care să accepte limbajul L în $\mathcal{O}((f(n) \cdot \text{card}(U))^2)$.*

De asemenea, aratăm că orice NPEP, cu alfabetul de lucru V și funcția de evaluare φ , poate fi simulată de către o mașină Turing.

Teoremă 4.4. *Pentru orice limbaj L acceptat de către o Rețea de Procesoare Polarizate, cu alfabetul de lucru V și funcția de evaluare φ , în $\mathcal{O}(f(n))$, există o mașină Turing care decide pe L în $\mathcal{O}(f(n)(Kf(n) + Kn))$, unde $K = \max\{|\varphi(a)| \mid a \in V\}$.*

Considerăm apoi o problemă deschisă prezentă în [3]. Vom construi o NPEP cu un număr constant de noduri care simulează o mașină Turing, dar de această dată folosim doar simboluri cu o polaritate asociată din mulțimea de elemente $\{-1, 0, 1\}$, dar în timp mai puțin eficient.

Teoremă 4.5. *Pentru orice limbaj recursiv enumerabil L , acceptat de către o mașină Turing cu alfabetul de lucru U , există o Rețea de Procesoare Polarizate cu 35 de noduri, pentru care evaluarea polarității fiecărui simbol este limitată la mulțimea $\{-1, 0, 1\}$, care să accepte limbajul L .*

În capitolul 4 comparăm clasele de limbaje 2-dimensionale $L(LOC)$ și $L(REC)$ cu clasa de limbaje care pot fi decise de *Rețele Acceptoare de Procesoare Evoluționiste Polarizate de Picturi* (ANPEPP) [20]. Într-o primă etapă, construim o ANPEPP care poate accepta numai imagini pătrate scrise peste un alfabet cu o singură literă - astfel de imagini putând fi acceptate de către un sistem de placare, dar nu pot fi recunoscute local de către un set de plăci.

Teoremă 4.6. $L(ANPEPP) \cap (L(REC) \setminus L(LOC)) \neq \emptyset$.

Arătăm apoi că ANPEPP-urile pot accepta unele limbaže care nu sunt în $L(REC)$. Mai exact, construim o ANPEPP care acceptă doar picturi cu un număr par de rânduri ce au cele două rânduri de mijloc identice.

Teoremă 4.7. $L(ANPEPP) \setminus L(REC) \neq \emptyset$.

Următorul rezultat face referire la complementul oricărui limbaj ce poate fi acceptat local de către un set de plăci.

Teoremă 4.8. *Fie $L \in L(LOC)$ iar L^c complementul lui L , atunci $L^c \in L(ANPEPP)$.*

În cele din urmă prezentăm o soluție parțială pentru problema de potrivire de model.

Teoremă 4.9. *Fie o pictură π cu $l_1(\pi) \leq 3$ sau $l_2(\pi) \leq 3$ și $L = \{\pi' \mid \pi \text{ este o sub-pictură a lui } \pi'\}$, atunci $L \in L(ANPEPP)$.*

Bibliografie

- [1] P. Alarcón, F. Arroyo, V. Mitrana
Networks of polarized evolutionary processors as problem solvers. In *Advances in Knowledge-Based and Intelligent Information and Engineering Systems, Frontiers in Artificial Intelligence and Applications*, IOS Press, (2012), pg 807–815.
- [2] F. Arroyo, S. G. Canaval, V. Mitrana, Șt. Popescu
Networks of Polarized Evolutionary Processors Are Computationally Complete. In *Language and Automata Theory and Applications*, Lecture Notes in Computer Science 8370, Springer-Verlag, Berlin, (2014), pg 101–112.
- [3] F. Arroyo, S. G. Canaval, V. Mitrana, Șt. Popescu
On the Computational Power of Networks of Polarized Evolutionary Processors. Accepted for publication at *Information and Computation*.

- [4] P. Bottoni, A. Labella, V. Mitrana
Accepting Networks of Evolutionary Picture Processors. *Journal Fundamenta Informaticae - Formal Models-Computability, Complexity, Applications, Volume 131 Issue 3-4* (July 2014), pg 337–349.
- [5] P. Bottoni, A. Labella, F. Manea, V. Mitrana, J. Sempere
Networks of Evolutionary Picture Processors with Filtered Connections. In *UC '09 Proceedings of the 8th International Conference on Unconventional Computation* (2009), pg 70–84.
- [6] J. Castellanos, C. Martín-Vide, V. Mitrana, J. M. Sempere
Networks of evolutionary processors. In *Acta Inf.* 39 (2003), pg 517–529.
- [7] M.A. Diaz, L.F. de Mingo, N. Gómez Blas, J. Castellanos
Implementation of massive parallel networks of evolutionary processors (MPNEP): 3-colorability problem. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*, Studies in Computational Intelligence 129, Springer, (2008), pg 399–408.
- [8] M.A. Diaz, L.F. de Mingo, N. Gómez
Networks of evolutionary processors: Java Implementation of a threaded processor. In *International Journal of Information Theories & Applications* 15(2008) pg 37–43.
- [9] C. Drăgoi, F. Manea, V. Mitrana
Accepting networks of evolutionary processors with filtered connections. *J. UCS* 13 (2007), pg 1598–1614.
- [10] D. Giammarresi, A. Restivo,
Recognizable picture languages. In *Int. Journal Pattern Recognition and Artificial Intelligence*. Vol. 6, No. 2 & 3 (1992), pg 241–256.
- [11] D. Giammarresi, A. Restivo,
Two-dimensional languages , in [21], pg 215-267.
- [12] N. Gómez, L.F. de Mingo, J. Castellanos
Networks of evolutionary processors with a self-organizing

- learning. In *Proc. ACS International Conference on Computer Systems and Applications*, IEEE Computer Society Washington, DC, USA, (2008), pg 917–918.
- [13] F. Manea, C. Martín-Vide, V. Mitrana
On the size complexity of universal accepting hybrid networks of evolutionary processors. In *Mathematical Structures in Computer Science* 17 (2007), pg 753–771.
- [14] F. Manea, M. Margenstern, V. Mitrana, M. J. Pérez-Jiménez
A new characterization of NP, P, and PSPACE with accepting hybrid networks of evolutionary processors. In *Theory Comput. Syst.* 46 (2010), pg 174–192.
- [15] F. Manea, C. Martín-Vide, V. Mitrana
Accepting networks of splicing processors. In *New Computational Paradigms*, Springer Berlin Heidelberg, (2005) pg 300–309.
- [16] F. Manea, C. Martín-Vide, V. Mitrana
Accepting networks of evolutionary word and picture processors: A survey. In *Scientific Applications of Language Methods, Mathematics, Computing, Language, and Life: Frontiers in Mathematical Linguistics and Language Theory - Vol. 2*, World Scientific (2010), pg 523–560.
- [17] M. Margenstern, V. Mitrana, M. J. Pérez-Jiménez
Accepting Hybrid Networks of Evolutionary Processors. In *Proc. 10th International Workshop on DNA Computing, DNA 10*, Lecture Notes in Computer Science 3384, Springer-Verlag, Berlin (2004), pg 235–246.
- [18] C. Martín-Vide, V. Mitrana, M. J. Pérez-Jiménez, F. Sancho-Caparrini
Hybrid networks of evolutionary processors. In *Proc. Genetic and Evolutionary Computation - GECCO 2003*, Lecture Notes in Computer Science 2723 Springer-Verlag, Berlin (2003), pg 401–412.
- [19] Șt. Popescu
Networks of Polarized Evolutionary Processors with Elementary Polarization of Symbols, submitted.

[20] Șt. Popescu

Networks of Polarized Evolutionary Picture Processors. Accepted to appear in *Romanian Journal of Information Science and Technology*. Vol. 18, No. 1 (2015).

[21] G. Rozenberg, A. Salomaa (eds.)

Handbook of Formal Languages, vol. I-III. Springer-Verlag, Berlin (1997).