

**Optimizarea Luminilor Semafoarelor  
folosind Sisteme de Roiuri de Particule si  
Automate Celulare  
OLSSRPAC**

**Echipa 13**

Muresan Paul, Grupa 935  
Vidrascu Traian, Grupa 931

January 9, 2018

<b>1</b>	<b><u>PROPUNERE</u></b>	<b><u>3</u></b>
<b>2</b>	<b><u>INTRODUCERE ȘI MOTIVAȚIE</u></b>	<b><u>4</u></b>
<b>3</b>	<b><u>DEFINIREA PRECISĂ A PROBLEMEI</u></b>	<b><u>5</u></b>
<b>4</b>	<b><u>ABORDĂRI ÎNRUDITE</u></b>	<b><u>6</u></b>
<b>5</b>	<b><u>METODA DE LUCRU</u></b>	<b><u>7</u></b>
<b>6</b>	<b><u>EXPERIMENTE NUMERICE</u></b>	<b><u>9</u></b>
<b>7</b>	<b><u>CONCLUZII</u></b>	<b><u>11</u></b>
	<b><u>BIBLIOGRAFIE</u></b>	<b><u>10</u></b>

## 1. Propunere

Optimizarea traficului rutier este o problema des intalnita de orasele mari inca de la inceputul secolului XX. Numeroase solutii au aparut in vederea optimizarii acestuia cum ar fi reducerea lui prin incurajarea transportului public sau sincronizarea si gestunua lui prin semafoare. OLSSRPAC este un proiect care isi propune sa analizeze aplicabilitatea algoritmilor asupra problemei de gestionare si sincronizare al traficului.

Proiectul in sine abordeaza doi algoritmi inteligenti pentru a obtine o gestinue proprie asupra unei zone aglomerate. Primul algoritm consta intr-un Sistem de Roi de Particule care incearca sa descopere un optim global, iar al doilea este un Automat Celular care se foloseste de un joc de congestie. Cele doua rezolvari isi propun obtinerea unei sincronizari optime intre semafoare, ameliorand astfel traficul din aceea zona.

In vederea analizarii performatie sau a optimizarii algorimului am folosit programul SUMO(Simulation of Urban Mobility) de la DLR(Deutsche Zentrum fÃ¼r Luft- und Raumfahrt ). Datele de input pe care le-am folosit in SUMO sunt harti rutiere modificare prin OSM (OpenStreetMap) si sumalari sub forma unor fisiere xml generate printr-un tool al programului SUMO.

Mai precis, am folosit datele sintetice de la proiectul RILSA care contineau 4 scenarii, fiecare reprezentand o intersectie clasica cu 4 intrati si iesiri, diferind numai prin numarul de masini din simulare, numarul de benzi sau prin starile semafoarelor. Inafara datelor sintetice am folosit si date reale precum centrul Clujului cu o simulare de 500 sau 300 de masini si piata Avram Iancu cu 300 de masini.

Ca si referinte pentru solutiile noastre am avut urmatoarele lucrari:

1.Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization de Jose GarcÃa-Nieto, Ana Carolina Olivera, and Enrique Alba

2.A swarm intelligent method for traffic light scheduling de Wenbin Hu1, Huan Wang, Liping Yan si Bo Du

## 2. Introducere si Motivatie

In ziua de astazi problema traficului este una foarte mare pe care toate tarile din lumea intai incearca sa o combata prin metode noi. Romania este o tara care incepe sa se urbanizeze din ce in ce mai tare, iar infrastructura strazilor este sortita a nu rezista unei populatii urbane moderne.

Aglomeratia aduce problem severe cum ar fi poluarea orasului, probleme cu locarile de parcare si securitate pe langa stresul populatiei care traieste in oras. Acest lucru a restrictionat mereu dezvoltarea in medii urbane, motiv pentru care inca in ziua de astazi se cauta solutii noi si fiabile.

Infrastructura drumurilor si mai ales a oraselor din Romania este una neoptima pentru traficul urban spre care ne indreptam si avem neaparata nevoie de o optimizare cat mai repede. Avem deja exemple ale acestui esec in orase precum Bucuresti si Cluj-Napoca, in care la o populatie de sub un milion in cazul orasului Cluj-Napoca, avem ore de varf de aproape doua ore de aglomeratie in anumite zile.

Semafoarele sunt cea mai buna solutie pentru gestionarea traficului, iar schimbari in infrastructura sunt costisitoare si de multe ori mult prea greu de implementat, de aceea concluzia la care s-a cazut de comn acord este ca solutia optimizarii luminilor semafoarelor este cea mai buna si fiabila pentru rezolvarea problemelor de congestie.

Simulatoarele de trafic s-au dovedit a fi o solutie foarte buna pentru traficul real, astfel rezolvarile la problemele de congestie pot fi testate printr-un simulator in loc de testari in lumea reala.

Solutia noastra este folosirea algoritmilor inteligenti pentru a rezolva problemele de aglomeratie. Am ales pentru aceasta problema folosirea un Sistem de Roi de Particule care incearca sa descopere in optim global si un Automat Celular care joaca un joc de congestie.

Consideram ca aceste doua solutii vor reusi sa rezolve problema traficului aglomerat deoarece Sistemul de Particule va invata cum sa optimizeze luminile semafoarelor din cazuri ipotetice, iar functia gasita se va putea aplica pe o varietate de strazi si jocul de congestie al Celularului Automat este strans legat de numarul de masini de pe strada, jocul reusind sa optimizeze orice trafic indiferent de strada.

### 3. Definirea Precisa a Problemei

Pana acum, traficul este gestionat cel mai bine de catre semafoare, dar ele nu sunt optime pentru toate tipurile de trafic din timpul zilei si de cele mai multe ori au un ceas incorporat care schimba culoarea semaforului indiferent de cate masini sunt in acel moment. Acest lucru produce adesea congestii si rezulta intr-un trafic aglomerat care la randul sau aduc probleme majore in oras.

Optimizarea semafoarelor de pana acum sunt schimbarea intervalurilor de stari a semafoarelor in functie de ora, insa aceasta abordare se bazeaza pe presupunerea unui anumit numar de masini care vor fi la o anumita ora.

Alte solutii au fost gasite, insa doar pentru drumuri mai putin folosite fata de cele din orase, ca de exemplu senzorii care detecteaza daca este cineva la semafor sau nu.

Propunerea noastra este de a folosi algoritmi inteligenti pentru optimizarea luminilor semafoarelor, sincronizandu-le pentru a gestiona optim traficul astfel incat congestioanarea traficului sa fie cat mai redusa, indiferent de ora sau numarul de masini din intersectie si oras.

Dorim sa gasim o functie si sa aplicam un joc de tip congestie din teoria jocurilor pentru a sincroniza semafoarele din fiecare intersectie ale unui oras. Automatul Celular prezinta joaca un joc inspirat din jocurile de congestie prin care fiecare celula schimba starea unui semafor dintr-o intersectie depinzand de numarul de masini care sunt in intersectie, starea trecuta si starea vecinilor. Sistemul de Roiuri de Particule incearca sa gaseasca o sincronizare optima a starii semafoarelor prin evaluarea solutiei intr-o simulare, iar dupa aceea aplica functia invatata in intersectiile reale.

Sistemul de Roiuri de Particule necesita sa fie antrenat innainte, de preferat pe intersectiile din oras daca nu pe toata harta orasului, intr-o simulare, iar dupa va aplica functia gasita pentru a sincroniza fiecare semafor din oras in cel mai optim mod gasit. Pentru Automatul Celular sincronizarea semafoarelor se face in timp real, nefiind nevoie de cunoastere de dinainte a intersectiilor si semafoarelor, aplicand jocul cu care este setat.

## 4. Abordari Inrudide

Avand in vedere ca aceasta problema are o cerere de rezolvare constanta exista incercari de solutii inteligente prin diferite metode precum Cautare Armonioasa, Inteligenta de Roi pana si o combinatie intre Sitemul de Roi de Particule si Celularul Automat. Pe langa acestea mai exista si solutii cu cautari prin grafuri de adiacenta, fiecare semfor fiind reprezentat ca un nod, iar legaturile dintre ele reprezinta daca o masina poate ajunge de pe strada pe care se afla pe alta daca semaforul i-ar permite.

Multe dintre abordari folosesc SUMO, Simulation of Urban MObility, program specializat pentru simulari ale traficului pe mape reale, dar si mape sintetice, pe care si noi am folosit-o in abordarile noastre asupra problemei, iar altele folosind osmnx, librerie Python.

Fiecare abordare a inceput cu niste date sintetice, intersectii predefinite de mai multe tipuri si semafoare in diferite pozitii, iar dupa au trecut pe cel putin o harta a unui oras real cu traficul simulat cat mai aproape de cel existent, aceasta fiind si abordarea noastra.

Abordarea folosind combinatia dintre Sistemul de Roi de Particule si Automatul Celular au produs rezultate mai bune in optimizarea traficului fata de cum sunt setate semafoarele, dovedind un succes al algoritmului, dar necesitand mai multa certerare pentru a conclusiona daca este un motiv clar de a schimba sincronizarea semfoarelor din orasul testat. Acest algoritm hibrid a avut si rezultate mai bune fata de un Sistem de Roi de Particule pur.

Cautarea Armonioasa Discreta a concluzionat si ea ca optimizeaza mai bine luminile semafoarelor fata de cele curente, pana si fata de o Cautare Armonioasa simpla. Algoritmul cu Inteligenta de Roi a dovedid si el ca este mai optim pe strazile din Malaga in a decongestiona traficul, dovedind astfel ca o abordare inteligenta este de considerat in problema de fata.

Noi consideram ca un Automat Celular sau un Sistem de Roi de Particule sunt de ajuns pentru rezolvarea problemei nefiind nevoie de o imbinare intre ele pentru a gasi o solutie corecta, poate doar mai optima.

## 5. Metoda de Lucru

### Sistem de Roi de Particule

Am preferat alegrea Sitemului de Roi de Particule in detrimentul altor algoritmi inteligenti din mai multe considerente. In primul rand, acest algoritm este cunoscut pentru rapiditatea sa de a converge catre solutiile optime, in plus aceasta proietate al algorimului este utila deoarece pe viitor pot interveni evenimente in trafic ce necesita actualizarea semafoarelor. In ultimul rand, implementarea unui sitem de roi de particule este simpla si depinde de putini parametri de rulare.

Descrierea algoritmului: 1.Reprezentarea solutiei: Reprezentarea solutie sau a postiei unei particule este un vector de intregi. Deoarece in SUMO fiecare semafor are o configurare logica care contine mai multe stari, fiecare avand cate o durata, avand in vedere ca modificam numai durata unei stari o dimesiune al unei particule reprezinta defapt durata unei stari de la un anumit semafor. Mai pe scurt, solutia noastra reprezinta duratele pentru toate starile de la fiecare semafor in parte, toatea find cumulate intr-un singur vector.

#### 2.Functia de evaluare

Obiectivul programului este sa maximizeze numarul de masini care ajung la destinatie si sa diminueze numarul celor care raman in drum. Asa ca functia de evaluare poate fi scrisa in felul urmator

$$F = A^2 - R \quad (1)$$

unde A reprezinta numarul de masini ajunse la destinatie iar R numarul celor care inca sunt in drum. Pentru o prioritizare catre solutiile care favorizeaza numarul de masini ajunse la destinatie s-a decis ridicare numarul acestora la patrat. Numarul de masini ajunse la destinatie precum si celor in drum se obtine prin rularea unui scenariu in programul SUMO, fiecare scenariu are durata in timp real de 500 de secunde.

#### 3. Optimizare

Algoritmul este compus din doua parti, una de simulare si evaluare, iar cealalta este de optimizare. Partea de opimizare este caracteristica unui sistem de roi de particule. Roiul este compus din 30 particule, fiecare particula parte dimensiunea numarului de stari al tuturor semafoarelor. O diminesiune are intervalul [5,50], Iar viteza va fi cuprinsa in acelasi interval. Algoritmul va rula in total 100 de iteratii.Fiecare iteratie este formata din cele doua parti mentionare mai sus. Mai intai se executa simularea si evaluarea particulelor. Apoi in faza optimizare se actualizeaza viteza fiecarei particule prin urmatoarea formula:  $V^i = w * V^i + c1 * random() * (Bcp^i - Cp^i) + c2 * random() * (Bp^i - Cp^i)$

unde  $i$  reprezinta dimensiunea,  $w$  - coeficientul de inertie,  $c1$  - coeficientul de acceleratie,  $c2$  - coeficientul social,  $V$  - viteza,  $Bcp$  - cea mai buna pozitie al particulei curente,  $Cp$  - pozitia curenta al particulei,  $Bp$  - pozitia particulei cel mai buna iar  $\text{random}()$  este cuprins intre 0 si 1. Coeficientii au fost initializati cu valorile  $w:1$ ,  $c1:2$ ,  $c2:2$ .

In urma actualizarii vitezei se face actualizarea pozitiei particulei. Acesta se face dupa formula:  $\text{daca } \text{random}() < s = 1 / (1 + e^{V^i}) \text{ atunci : } Cp^i = Bp^i + V^i$

Dupa executarea celor 100 de iteratii se selecteaza particula cea mai buna iar pozitia ei reprezinta solutia.

### Automat Celular

Am ales sa folosim un Automat Celular pentru rezolvarea problemei congestiei traficului, deoarece am considerat ca un Joc de tip Congestie din Teoria Jocurilor aplicat peste Automatul Celular este o abordare ce trebuie considerata, testata si concluzionat daca este o solutie viabila. Consideram ca un Automat Celular poate avea rezultate considerabile chiar daca nu sunt mai optime fata de un algoritm de machine learning.

Precum si in celelalte solutii ale acestei probleme, noi folosim SUMO pentru a simula traficul si a testa algoritmi. Solutia folosind Automatul Celular este in timp real, programul schimband stările semafoarelor de cand incepe simularea, nefiind nevoie de o pregătire a algoritmului pentru orasul in care se afla. Acest lucru este un beneficiu, deoarece nu este nevoie de timp de pregătire, iar rezultatele pot fi testate instant, anuland timpul de invatare si riscul ca algoritmul sa se poata aplica doar pe acea simulare.

Programul SUMO creaza mai multe stări valide ale unui semafoarelor, pentru a nu se produce accidente si probleme in mersul masinilor din cauza directiilor care sunt disponibile. Fiecare astfel de stare este reprezentata ca o celula in Automatul Celular. Regulile Automatului sunt inspirate din jocurile de tip congestie, fiecare celula isi schimba durata in functie de cate masini au trecut inainte si ce fel de stare este.

#### 1. Functia de schimbare a duratii in functie de stările precedente

Daca o celula reprezinta stop, galben si rosu, dar au trecut inainte multe masini isi va schimba durata in cu cate masini au trecut minus un numar intrus; daca o celula reprezinta verde isi va schimba durata cu cate masini au trecut plus un numar intrus; astfel fiecare celula optimizeaza durata in



functie de cate masini su trecut, respectiv cate sunt defapt pe drum.

$$\begin{cases} C_{itime} > nrMasiniC_{istate} & is \quad not \quad stopC_i = nrMasini - q \\ C_{itime} < nrMasiniC_{istate} & is \quad not \quad stopC_i = nrMasini + q \\ C_{itime} > nrMasiniC_{istate} & is \quad stopC_i = nrMasini - q \\ C_{itime} < nrMasiniC_{istate} & is \quad stopC_i = nrMasini + q \end{cases}$$

unde q reprezinta valoarea asignata sau durata actuala a celulei.

## 2. Functia de schimbare a duratii in functie de starile vecine

La final, fiecare celula isi schimba starea in functie de starile vecine, daca o celula este mai mare decat vecinii sai isi va schimba durata cu un numar random dintre doi indici mici, iar pentru cazut opus se ia un numar random intre doi indici mari, astfel asigurand ca un semafor nu va merge pentru mult prea mult timp daca nu este un trafic aglomerat.

$$\begin{cases} C_{itime} > C_{i-1time} & C_{itime} > C_{i+1time} & C_i = random(xy) \\ C_{itime} < C_{i-1time} & C_{itime} < C_{i+1time} & C_i = random(ab) \end{cases}$$

unde x si y reprezenta valorile de asignare minime si a si b valorile maxime. Valorile trebuie decise in functie de oras, dar pentru simulari am ales valori reprezentative neavand multi factori de luat in considerare.

## 6. Experimente Numerice

Simularile pe care le-am abordat au fost pe intersectii semaforizate create de noi, dar si pe doua parti din Cluj-Napoca, mai precis partile centrale si catre Grigorescu. Fiecare simulare a avut un numar diferit de masini astfel incat sa putem aborda mai multe situatii in experimentele noastre.

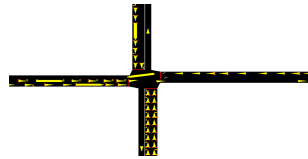
Desi algoritmii sunt diferiti rezultatele sunt apropiate, chiar daca un algoritm invata sistemul de strazi, in timp ce celalalt joaca un joc indiferent unde este simularea. Pentru simularile de test, in care am luat mai multe tipuri de intersectii cu diferite plasamente de semafoare si numere diferite de masini pe fiecare sosea, pentru a putea testa diferite ipostaze ale traficului real.

Pentru fiecare simulare exista: un numar de strazi care se intersecteaza, un numar de masini care se afla pe sosea care au un punct de inceput si un punct final unde trebuie sa ajunga si un numar de semafoare fiecare cu mai multe stari si durata starii.

Cele mai relevante experimente sunt experimentul sisntetic de intersectie in cruce toate cu semafoare, facut sa fie aglomerat, intersectiile din centrul

Clujului, la fel fiind facute sa fie aglomerat si intersectiile din centrul Clujului pana la inceputul la Grigorescu.

Pentru o intersectie in cruce cu 311 de masini pe drum, timpul 1230 folosind Sistemul de Roi de Particule au ajuns 234 in timp ce folosind Automatul Celular masinile care ajung sunt 180-243. Singura, Simularea in timpul 1000 reuseste sa aduca 97 de masini la destinatie din 351.



Pentru cazul din centrul Clujului pentru 351 de masini, timpul 500 rezultatul Sistemului este de 251 masini care ajung la destinatie. Folosind Automatul rezultatele sunt apropiate din 352 de masini ajung 247 la destinatie pentru timpul acesta. Lasand simularea sa ruleze singura pana cand toate masinile ajung la destinatie timpul necesar este de 1700 in timp ce folosind Sistemul este de 1600 si Automatul Celular de 1500-1700, rezultatul nefiind constant catre un numar.



Pentru Cluj-Napoca Centru si Grigorescu lasand simularea sa ruleze singura timpul in care toate masinile ajung la destinatie este de 1600 in timp ce folosind Automatul Celular este de 1300-1500, iar Sistemul este de 1800.



Algoritmii reusesc sa gaseasca o sincronizare buna pentru semafoarele din oras si cele din simulatoare.

## 7. Concluzii

Acest proiect studiaza optimizarea luminilor semafoarelor pe o infrastruc-tura de drummuri urbane. Algoritmii de Sistem de Roi de Particule (PSO) si Automatul Celular sunt propusi ca si o solutie la aceasta problema. Sis-temul de Roi de Particule reuseste sa optimizeze durata semafoarelor dupa ce invata din mai multe simulari, iar rezultatele sunt mai optime, dar ex-ista si cazuri ce produc rezultate nefiabale fata daca simularea ar fi neatinsa, totusi diferenta fiind foarte mica. De asemenea Automatul Celular opti-mizeaza traficul aplicand jocul de tip congestie, dar nu produce rezultate stabile, insa intervalul de solutii pe care le gaseste sunt mai optime decat cele fara interferenta si posibil chiar si fata de Sistemul de Roi de Particule.

1. Algorimtii sunt comparate cu combinatia de Automat Celular plus Sis-tem de Roi de Particule, numit IOCA-PSO, insa rezultatele dovedesc a nu fi la fel de optime fata de combinatie, dar nu sunt cu mult mai neoptime.
2. Rezultatele algoritmilor sunt comparate fata de daca starea semafoarelor nu ar fi schimbata. Amandoi algoritmi reusesc sa gaseasca o solutie mai optima.
3. Rezultatele sunt comparate intre ele, Automatul Celular prezentand metode surprinzatoare fata de Sistemul de Roi de Particule.
4. Cand numarul de vehicule nu este cel testat algoritmul de Sistem de Roi de Particule cat si Automatul Celular nu prezinta probleme in reglemntarea luminilor.
5. Automatul Celular nu poate sa isi dea complet seama daca este aglom-eratie sau doar au fost multe masini pe un drum, rezultand intr-o va-rietate de rezultate, dar toate optime, insa si in situatii ce ar putea fi considerate congestii.

Comparand rezultatele la algoritmi, noi consideram ca desi Automatul Celu-lar nu reuseste sa rezulte intr-o solutie constanta, solutiile pe care le gaseste sunt optime, dovedind astfel ca pentru o asemenea problema poate fi o so-lutie viabila, iar pentru Sistemul de Roiuri de Particule este nevoie de mai multa certetare pentru a putea zice cu siguranta daca solutia este viabila pentru problema traficului.