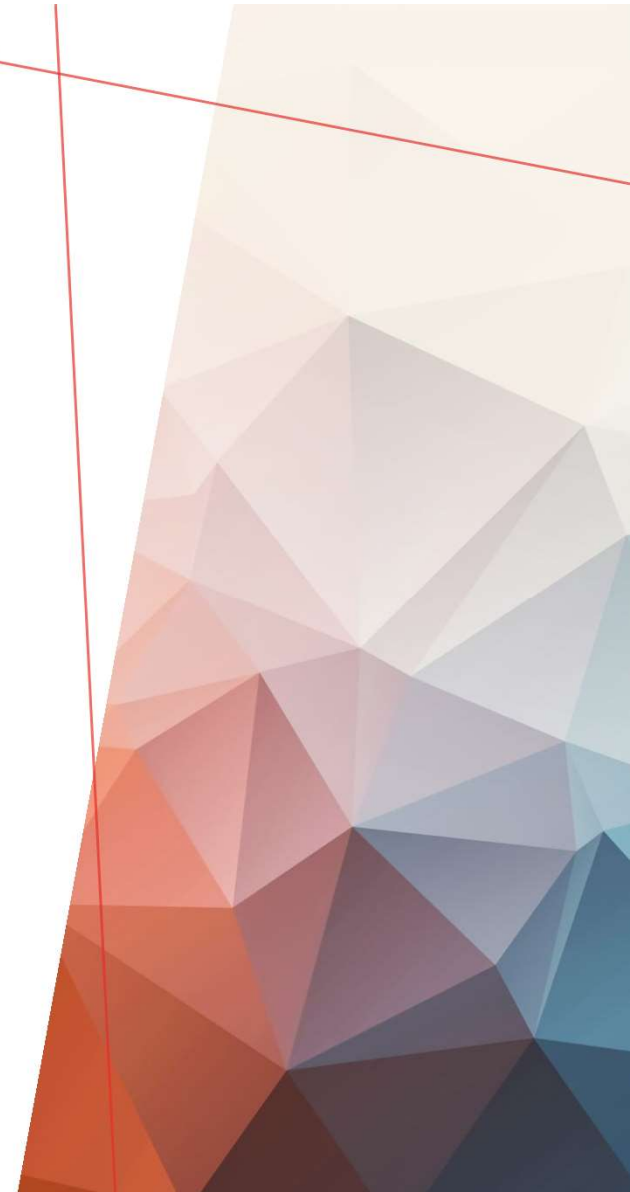


CONSTANTINESCU GABRIELA, GRUPA 331

*PROIECT –  
INTELIGENTA  
ARTIFICIALA*



# *TEMA ALEASA*

- Lucrare: Genetic Algorithm for Solving the Traveling Salesman Problem Using Neighbor-based Constructive Crossover Operator – Akshay Vyas, Dashmeet Kaur Chawla
- Cum functioneaza acest operator?
- Ce parametrii am folosit?
- Ce schimbari am adus?
- Rezultate si concluzii

# *CUM FUNCTIONEAZA NOUL OPERATOR DE INCRUCISARE?*

## Neighbor-based Constructive Crossover Operator

- Operator de incrucisare
- Constructive – cromozomul rezultat din incrucisarea parintilor este construit nod cu nod
- Neighbor-based – in timpul algoritmului, se iau in considerare vecinii (atat din stanga, cat si din dreapta) nodurilor parintilor

# *CUM ARATA ALGORITMUL IN REST?*

- Pentru a stabili performanta incrucisarii propuse, aceasta a fost comparata cu alte 2 tipuri de incrucisare:
  - SPCX – Single Point Crossover – incrucisare printr-un singur punct de taietura
  - SCX – Sequential Constructive Crossover – seamana cu NCX, dar nu se iau vecinii in considerare, ci doar primul nod legitim gasit (se cauta doar in partea dreapta)
- Selectia se face in mod elitist, tinand cont de o probabilitate de reproducere
- In plus, se mai tine cont si de probabilitatea de incrucisare si de cea de mutatie

# *CE SCHIMBARI AM ADUS EU?*

- Pentru mutatie, lucrarea propune 2-swap; eu am rulat atat cu 2-swap, cat si cu 2-opt.  
Am rulat cu 2-opt doar pentru operatorul NCX (nu si SPCX sau SCX)

# *REZULTATE TSP SIMETRIC*

		NCX			SCX			SPCX		
n	Opt val	Best (e)	Avg (e)	Time	Best (e)	Avg(e)	Time	Best (e)	Avg(e)	Time
51	426	460(7.98)	473(11.03)	1.4	476(11.73)	501.9(17.81)	1.28	797(87.08)	932.5(118.89)	0.37
51	426	465(9.15)	495.3(16.28)	3.25	465(9.15)	493.9(15.93)	2.40	518(21.59)	566.9(33.08)	1.99
76	108159	123390 (14.08)	130222.9 (20.39)	2.39	137127 (26.78)	144131.8 (33.25)	2.02	262244 (142.46)	339919.9 (214.27)	0.54
76	108159	116168 (7.4)	133699.8 (23.61)	6.39	124487 (15.09)	136099.3 (25.83)	4.82	149227 (37.97)	174284.9 (61.14)	3.32
101	629	705(12.08)	735.2(16.88)	3.79	837(33.07)	852.8(35.58)	3.09	1860(195.7)	2101.6(234.11)	0.79
101	629	722(14.79)	784.3(24.69)	9.56	769(22.26)	812.4(29.16)	7.86	901(43.24)	1001.7(59.25)	4.63

## *2-SWAP VS 2-OPT*

		2-SWAP			2-OPT		
n	Opt_val	Best (e)	Avg (e)	Time	Best (e)	Avg(e)	Time
51	426	465(9.15)	495.3(16.28)	3.25	440(3.28)	453(6.36)	3.24
76	108159	116168 (7.4)	133699.8 (23.61)	6.39	109477(1.22)	115358(6.66)	6.23
101	629	722(14.79)	784.3(24.69)	9.56	656(4.29)	686.2(9.09)	9.34

- Comparatia este realizata intre rezultatele mele

# *INTERPRETARE REZULTATE*

- Urmand exact aceeasi parametrii ca in lucrare, NCX nu a dat rezultate la fel de bune in rularile mele. SPX si SPCX au dat rezultate mai bune.
- Schimband mutatia din 2-swap in 2-opt, am obtinut rezultate mult mai bune, comparate fiind cu cele din lucrare.
- Timpii de rulare au fost mai mari decat cei din lucrare
- In cadrul rularilor se ajunge la convergenta prematura. Tinand cont de rezultatele bune obtinute, consider ca numarul de generatii 1000 este prea mare.