

Elektrotehnički fakultet, Univerzitet u Beogradu

# Kosta Jovanović

2016/0199



*Izveštaj za prvi domaći iz veštačke inteligencije*

**Tema:** *Zadovoljavanje ograničenja i lokalna pretraga*

Predmetni profesori:

Doc. dr Predrag Tadić

Marija Milutinović

- **Reprezentacija magičnog kvadrata i pomoćne funkcije**

Magični kvadrat predstavljen je klasom *MagicSquare*. Stanje magičnog kvadrata opisano je dovdimenzijskim nizom brojeva od 1 do  $n^2$ , gde je  $n$  red magičnog kvadrata. U klasi su definisane sledeće metode:

1. *genRandSquare()*

Metoda na slučajan način permutuje brojeve unutar magičnog kvadrata.

2. *getNumOfViolated()*

Metoda vraća ukupan broj prekršenih ograničenja za magični kvadrat.

Maksimalan broj prekršnih ograničenja je  $2 * n + 2$ .

3. *getSuccessors(mode,k)*

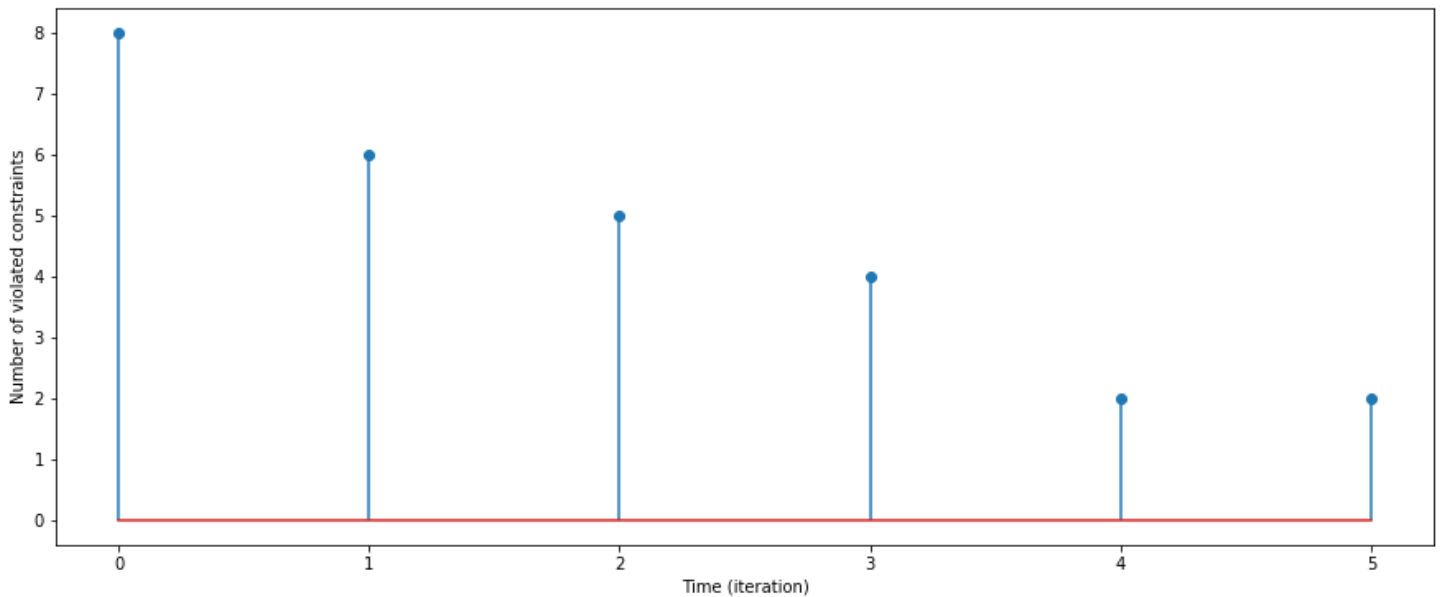
Metoda vraća  $k$  naslednika za trenutno stanje magičnog kvadrata (za  $k = None$  vraća sve naslednike). Metoda sadrži i parameter *mode*, koji može uzeti dve vrednosti "random" ili "top". Ako je prosleđen parametar "random" metoda vraća  $k$  nasumično izabranih naslednika, a ako je prosleđen parametar "top" metoda vraća sortiranu listu od  $k$  najboljih naslednika. Naslednicima se smatraju svi magični kvadrati koji se mogu dobiti zamenom dva elementa trenutnog stanja magičnog kvadrata, takvih kvadrata ima  $\binom{n^2}{2}$ .

- **Nasumična pretraga**

U svakoj iteraciji algoritma na slučajan način bira se jedan naslednik trenutnog stanja. Ako je on bolji, uzima se za trenutno stanje. Algoritam se prekida kada istekne unapred zadat broj iteracija, ili kada je broj prekršenih ograničenja trenutnog stanja jednak 0.

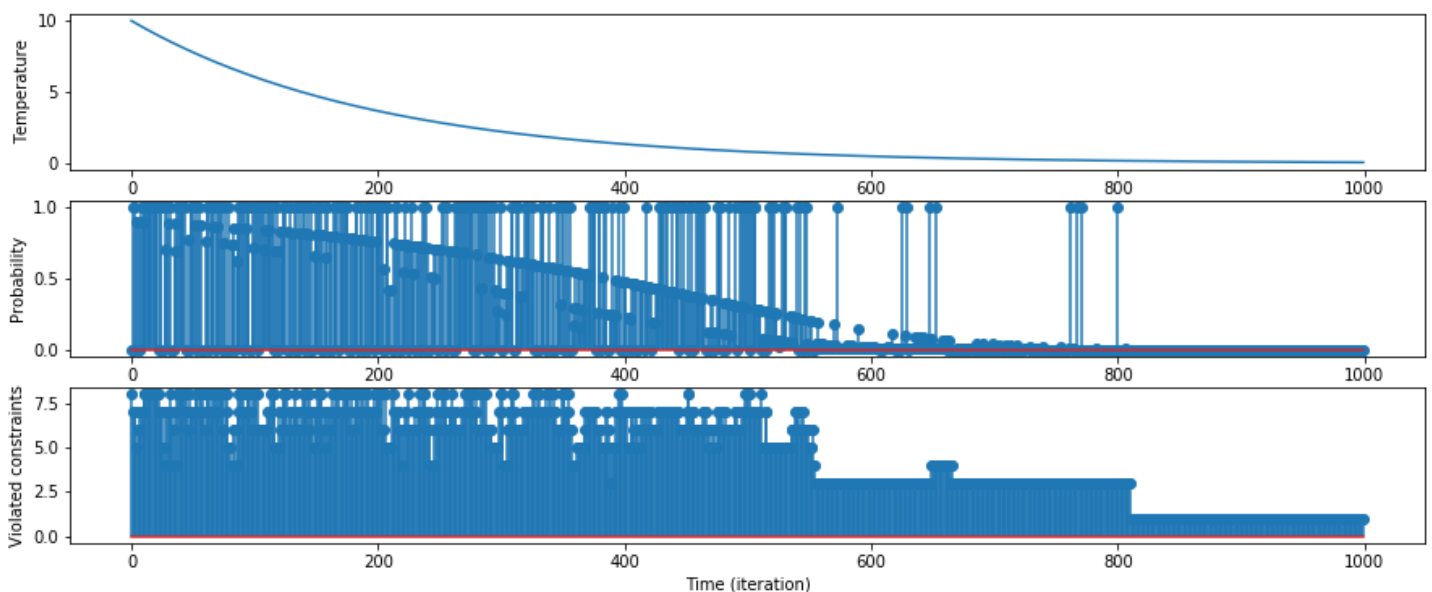
- **Hill climbing algoritam**

Algoritam u svakoj iteraciji uzima najboljeg naslednika trenutnog stanja. Ako je naslednik lošiji, to znači da smo dostigli minimum (lokalni ili globalni) i algoritam se prekida. A ako je naslednik podjednako dobar to znači da smo potencijalno dostigli “ravan” i algoritam prestaje sa radom.



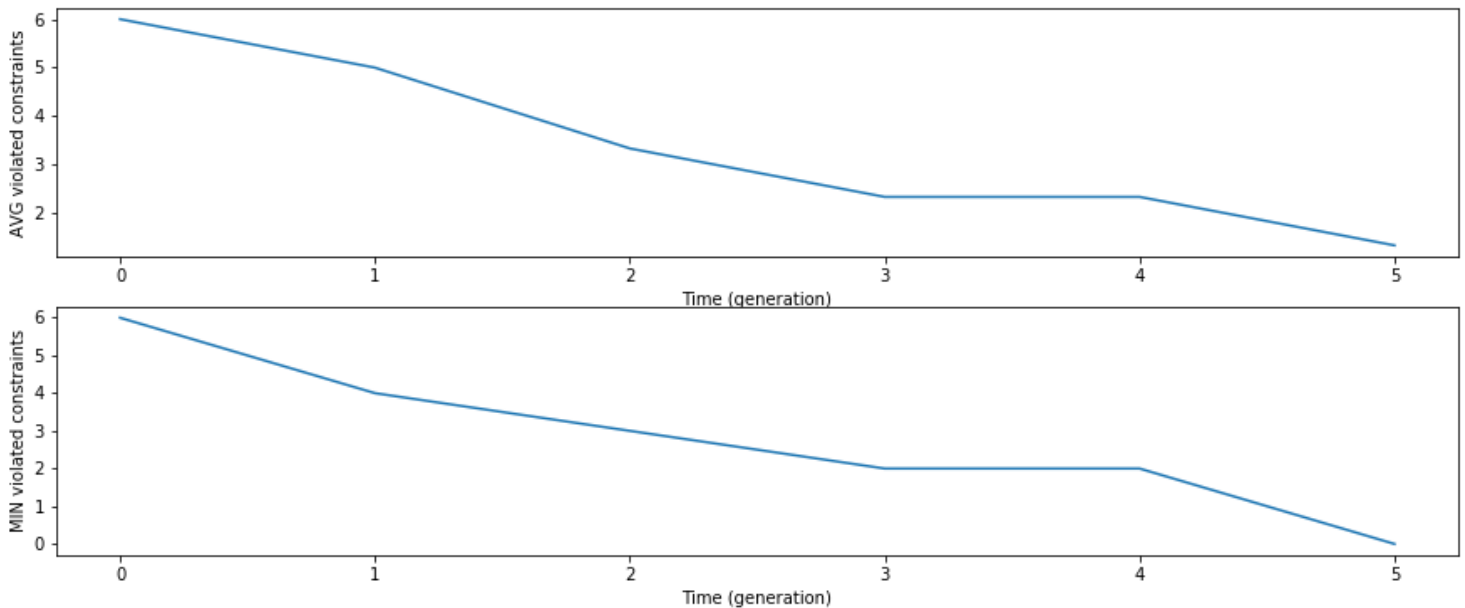
Na slici iznad algoritam je dostigao “ravan”, jer je u 4. i 5. iteraciji broj prekršenih ograničenja isti.

- **Simulirano kaljenje**



Temperatura je snižavana tako što je u svakoj iteraciji množena sa konstantom manjom od 1. Na slici iznad se jasno može videti da što je temperature veća, verovatnoća da lošiji naslednik bude izabran se povećava.

- **Pretraga po snopu**



Može se desiti da se algoritam zaglavi u nekom lokalnom minimum, na primer kada stanja u svakom snopu imaju podjednak broj prekršenih ograničenja i kada se ni u jednom snopu ne može naći naslednik sa manjim brojem prekršenih ograničenja. Rešenje za ovaj problem je da se poveća broj snopova, ili da se pamte stanja iz prethodne iteracije kako ih algoritam ne bi ponovio, jer se, na primer, stanje snopa 1 može pojaviti u naslednicima stanja snopa 2.

- **Genetski algoritam**

- Fitness funkcija:

Za svaku jedinku  $i$ , fitness funkcija  $f_i$  je računata prema formuli

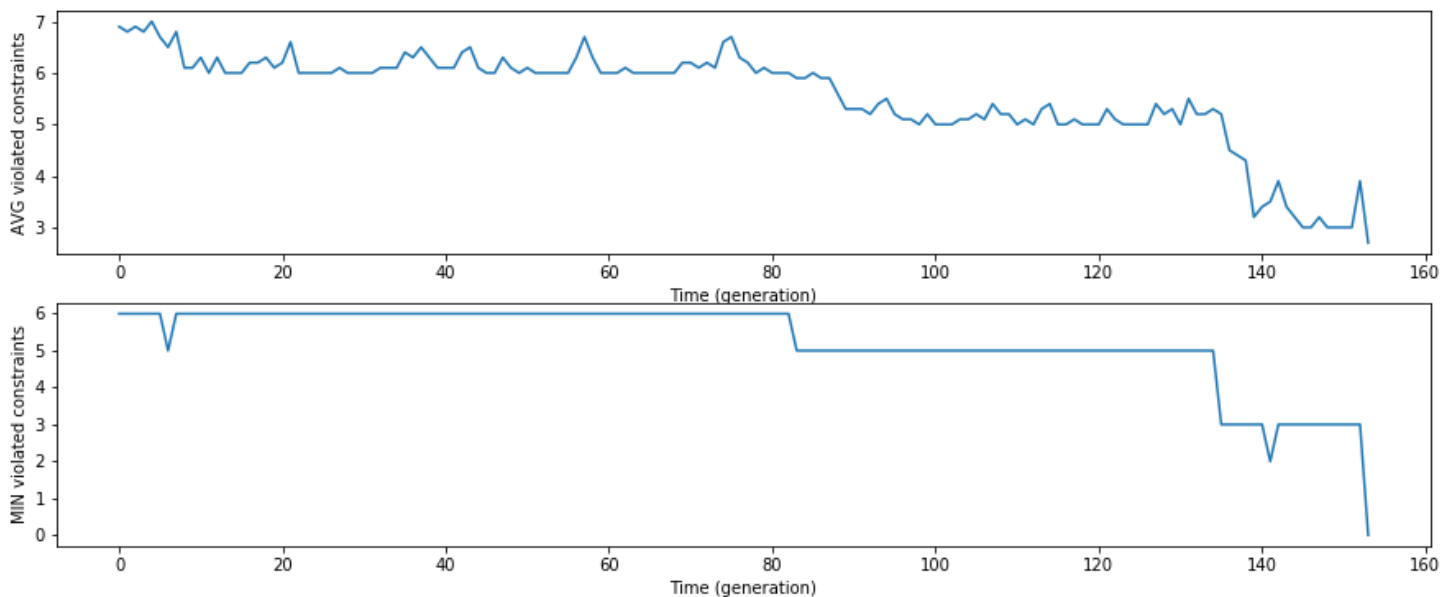
$$f_i = \frac{z_i}{\sum z_j}, \text{ gde je } z_i \text{ broj zadovoljenih ograničenja za jedinku } i.$$

- Ukrštanje:

Pošto je magični kvadrat permutacija, klasične metode ukrštanja (kao što je presecanje) nisu upotrebljive, jer bi došlo do ponavljanja elemenata u kvadratu. Metoda koja je implementirana je predložena u *“Genetic Algorithm Solution of the TSP Avoiding Special Crossover and Mutation written by Göktürk Üçoluk”*. Metoda se sastoji u nalaženju inverzija permutacija oba roditelja, zatim primene klasičnog presecanja na inverzije, i vraćanja nazad u permutaciju. Algoritmi za nalaženje inverzije i permutacije mogu se naći u kodu.

- Mutacija:

Svako dete se mutira sa nekom malom, unapred zadatom verovatnoćom. Mutacija se vrši tako što se na slučajan način izaberu 2 mesta (4 indeksa) u matricama i izvrši se zamena elemenata na tim mestima.



Sa slike iznad se može videti da generacije napreduju kroz vreme, u pogledu minimalnog broja prekršenih ograničenja

## • Poređenje algoritama

- Rezultati za **n=3**

Hiperparametri:

- Simulirano kaljenje: početna temperatura je  $10^\circ$ , konstanta snizavanja temperature je 0.995
- Pretraga po snopu: broj snopova je 3
- Genetski algoritam: veličina populacije je 10, verovatnoća mutacije je 0.05

Algorithm	Average Violated Constraints	STD Violated Constraints	Average Iterations	STD Iterations	Execution Time
Random Search	2.45	0.898610038	981.02	132.8843843	32.1033473
Hill Climbing	2.37	0.955562661	2.44	0.711617875	2.808985949
Simulated Annealing	1.72	1.04	927.74	185.2540213	39.48131108
Beam Search	1.53	0.984428768	810.83	390.5875332	4835.343567
Genetic Algorithm	2.09	1.000949549	946.72	201.8285451	481.3717811

Kao što je i očekivano nasumična pretraga i pretraga usponom daju prosečno najgora rešenja. Pretraga usponom često završi u lokalnom minimumu, a nasumična pretraga nadje rešenje samo ako joj se posreći. Pretraga po snopu daje očekivano najbolje rezultate, ali po ceni vremena izvršavanja, koje je neuporedivo veće nego kod ostalih algoritama. Najbolji balans izmedju tačnosti rešenja i vremena izvršavanja ima simulirano kaljenje.

- Rezultati za **n=4**

Svi hiperparametri su ostali isti, osim veličine populacije kod genetskog algoritma, koja je postavljena na 30.

Algorithm	Average Violated Constraints	STD Violated Constraints	Average Iterations	STD Iterations	Execution Time
Random Search	4.35	1.169401556	1000	0	24.93849778
Hill Climbing	4.24	1.241933976	3.71	1.168717246	10.31714034
Simulated Annealing	3.44	1.070700705	997.52	21.19645253	34.10695314
Beam Search	2.57	1.11584049	930.54	253.1788467	10885.55727
Genetic Algorithm	4.22	1.21371331	993.31	22.1156	1198.888511

Najbolji balans izmedju tačnosti i vremena izvršavanja opet ima simulirano kaljenje. Očekivao sam da će se genetski algoritam bolje pokazati.