

# 1. Rešavanje Suodku-a pomoću metaheuristike simuliranog kaljenja

Simulirano kaljenje (eng. simulated annealing, SA) je metaheuristika za problem globalne optimizacije u primjenjenoj matematici, najčešće za pronalazak aproksimacije globalnog optimuma neke funkcije u velikom prostoru pretraživanja. Koristi se uobičajeno u problemima gde je prostor pretraživanja diskretan. Inspiracija za ime dolazi od kaljenja u metalurgiji. Tehnika je to koja koristi zagrijavanje te kontrolirano hlađenje materijala kako bi se povećali njegovi kristali i smanjili njihovi nedostaci. Toplina utiče na atome tako što ih oslobađa od njihovog početnog položaja i dopušta im da se slobodno kreću na višim energetske nivoima. Polagano hlađenje daje atomima veću vjerojatnost da pronađu razmještaj u kojem će imati manju unutarnju energiju nego prije procesa kaljenja. Manja unutarnja energija atoma znači veću stabilnost spoja i bolje karakteristike.

Za analogiju rada algoritma s zakonima fizike potrebno je staviti stvari u ispravan kontekst. Svaki korak rada simuliranog kaljenja zamenjuje trenutno rješenje za nekim, njemu „bliskim“, slučajno odabranim rešenjem. Verovatnost odabira zamenskog rješenja zavisi u tome kolika je razlika u energetskom nivou (ceni) rešenja kandidata i trenutnog rešenja. Također ta verovatnost ovisi i o globalnom parametru  $T$ , koji označava temperaturu i postepeno se smanjuje tijekom rada algoritma kroz višestruke iteracije.

```
procedura simulirano_kaljenje(iO,c0);
    i:=i0; // početno rešenje
    c:=c0;
    Ci:=C(i); // funkcija cilja
    ponavlja
        ponavlja
            j:=susedno_rešenje(i);
            Cj:=C(j);
            ΔC:=Cj-Ci;
            prihvati:=FALSE;
            ako je ΔC<0 tada
                prihvati:=TRUE;
            inače
```

```

        ako je  $\exp(-\Delta C/c) > \text{random}[0,1]$ 
            tada prihvati:=TRUE;
        ako je prihvati=TRUE tada
            i:=j; // prihvati susedno rešenje
            Ci:=Cj;
        do termalne_ravnoteže
            smanji parametar c;
        do zamrzavanja
    kraj.

```

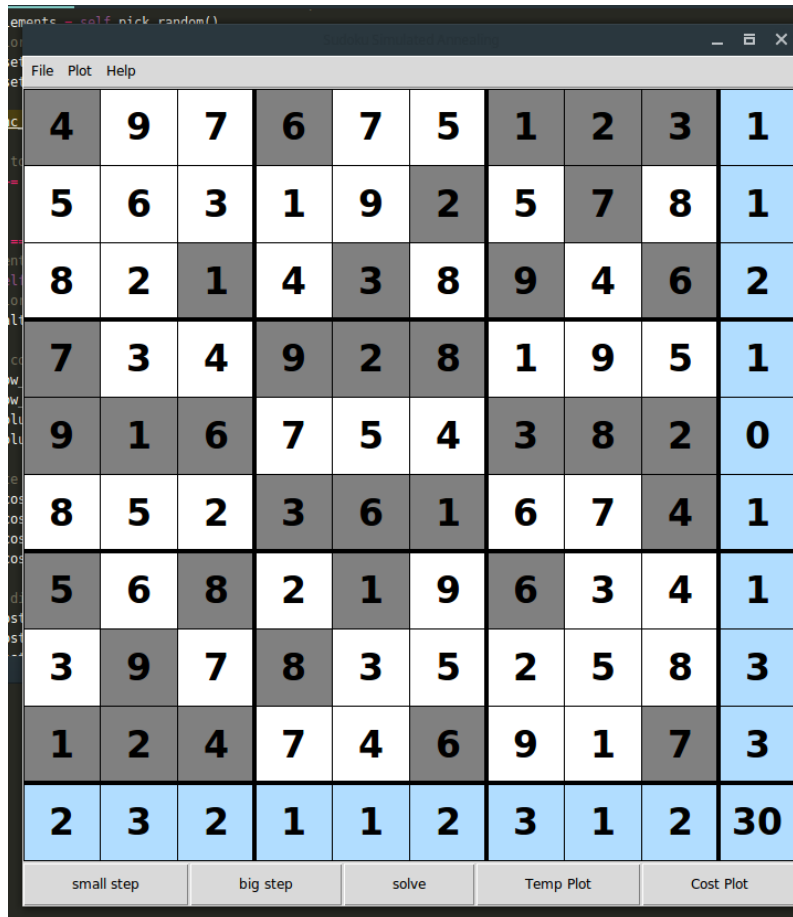
## 1. Rešavanje Sudokua metodom Simuliranog kaljenja

Logika simuliranog kaljenja može se iskoristiti za rešvanje sudokua definisanjem inicijalnog rešenja, susdstva i cene.

*Inicijalno rešenje:* Pseudo-slučajno popunite svaki od praznih kvadrata unutar 3x3 regiona, tako da svi regioni sadrže sve brojeve od 1 do 9 tačno jedanput. Drugim rečima, u inicijalnom rešenju ne morate voditi računa da se u vrstama i kolonama pojavljuju brojevi od 1 do 9 tačno jedanput. Koraci koji slede u algoritmu nikada neće menjati frekvenciju pojavljivanja brojeva od 1 do 9 u svakom 3x3 regionu (tj. uvek će ih biti po devet različitih u svakom regionu). Svako (potencijalno) rešenje koje će biti uzeto u razmatranje tokom pretrage (tokom rada algoritma) će u svakom 3x3 regionu sadržati tačno jedan od brojeva od 1 do 9. Vrste i kolone ne moraju da zadovoljavaju taj zahtev.

*Funkcija cene* koju treba minimizovati: Za svaku vrstu i (kolonu j) vrednost vrste rs i (vrednost kolone cs j , respektivno) je broj simbola (tj. brojeva od 1 do 9) koji nedostaju u vrsti i (koloni j, respektivno). Ukupna cena nekog potencijalnog rešenja je suma svih vrednosti vrsta i vrednosti kolona. Očigledno, rešenje Sudoku ukrštenice će biti ono čija je ukupna cena 0.

*Definisanje susedstva* u prostoru pretrage: Svaki sused trenutnog rešenja se dobija operacijom zamene (eng. swapping) vrednosti dve ne-fiksirane ("non-given") ćelije koje pripadaju istom 3x3 regionu. Pretraga susedstva sledi paradigmu simuliranog kaljenja – tako da se prvo pseudo-slučajno izabere jedna ne-fiksirana ćelija; zatim se pseudo-slučajno izabere druga ne-fiksirana ćelija u istom regionu, i njihove vrednosti se zamene.



### 3. Implementacija

Metaheuristika je implementirana u programskom jeziku Python u okviru sudoku aplikacije.

Aplikacija se sastoji od 4 dela:

1. main.py glavni deo aplikacije gde se nalazi metaheuristika
2. board.py implementira sudoku ploču
3. gui\_sudoku generiše interfejs aplikacije
4. gui\_plot generiše iscrtvanja cene i temperature u funkciji vremena