**Funkcija**:  
 f(x) = 2x+3

**Evaluator**:  
 g(x) = Ax+B  
**Binarni registar**:  
 br = [bA1, bA0, bB1, bB0]  
**Kvantni registar**:  
 q = [ΘA1, ΘA0, ΘB1, ΘB0]

## Inicijaliziraj populaciju

r1 = [, , , ]

r2 = [, , , ]

## Izmjeri qRegistre (pretvorba qBita u klasični bit)

br1 = [1, 0, 0, 0]

br2 = [1, 0, 1, 0]

## Evaluiraj klasična mjerenja (niz bitova)

g1(x) = 2x+0

g2(x) = 2x+2

## DOK (!UVIJET)

## Pohrani najboljeg (elitizam)

b = r2 = [, , , ]

( br2 = [1, 0, 1, 0] )

## Ostatak populacije izmijeni kvantnim genetskim operatorima

**Kvantna rotacijska vrata**

r1 = [, , , ]

br1 = [1, 0, **0**, 0]

b = [1, 0, **1**, 0]

Istoimene binarne vrijednosti ne mijenjaju kut qbita. U ovom primjeru mijenja se samo 3. qbit:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **qi** | **bi** | **f(q) > f(b)** | **∆θ** |  | | | |
|  |  |  |  |  |  |  |  |
| 0 | 0 | False | 0 | - | - | - | - |
| 0 | 0 | True | 0 | - | - | - | - |
| 0 | 1 | False | φ | +1 | -1 | 0 | ±1 |
| 0 | 1 | True | φ | -1 | +1 | ±1 | 0 |
| 1 | 0 | False | φ | -1 | +1 | ±1 | 0 |
| 1 | 0 | True | φ | +1 | -1 | 0 | ±1 |
| 1 | 1 | False | 0 | - | - | - | - |
| 1 | 1 | True | 0 | - | - | - | - |

φ je ovisan on fitnessu, no radi jednostavnosti primjera, uzet ćemo konstantu: φ =

r’1 = [, , , ]

Kvantna rotacijska vrata promijenila su 3. qubit koji je sada “bliži” stanju |1>.

To znači da ćemo u idućim mjerenjima te jedinke češće izmjeriti taj qbit u stanju 1.

Time se lošija jedinka “približila” boljoj, odnosno preuzela je osobine bolje jedinke.

Iako možemo u idućoj iteraciji algoritma mjerenjem bolje jedinke postići lošiji fitness (jer su mjerenja ‘usmjereno nasumična’), populacija će uvijek prilagođavati svoje atribute najboljoj jedinci, što usmjerava pretragu oko najboljeg rješenja.

**Kvantni operator mutacije (inverzijski)**

Nasumičnim odabirom invertiramo jedan qbit (u primjeru 4.):

r1 = [, , , ]

r’1 = [, , , ]

Vidimo da mutiranjem (invertiranjem) 4. qbita kada se nalazi u superpoziiji ne mijenjamo vrijednost kuta.

No pogledamo li mutaciju iste jedinke u nekoj višoj iteraciji algoritma nad 2. qbitom:

r1 = [, , , ]

r’1 = [, , , ]

Vidimo da smo mutiranjem 2. qbita jedinku približili rješenju problema ([1, 0, 1, 1]), te probabilistički usmjerili pretragu prema boljem rješenju.

## Izmjeri qRegistre (pretvorba qBita u klasični bit)

## Evaluiraj klasična mjerenja (niz bitova)

## POVRATAK NA PETLJU