SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4830

Genetski algoritam inspiriran kvantnom mehanikom

Juraj Fulir

Zagreb, lipanj 2017.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

Zahvaljujem svome mentoru prof. dr. sc. Domagoju Jakoboviću na pomoći i savjetima u razvoju rada te danoj mogućnosti vlastitog odabira teme.

Zahvaljujem svojoj obitelji na podršci i strpljenju pogotovo tijekom kasnih noćnih sati.

SADRŽAJ

1. Uvod

Genetski algoritmi postoje još od 1960-tih kada je John Holland sa svojim studentima i kolegama napravio algoritam koji je simulirao fenomen adaptivnog ponašanja u prirodi. Kasnije su otkrivene mogućnosti i primjene genetskih algoritama na širokom rasponu stvarnih problema iz područja inženjerstva, ekonomije, kemije, robotike i mnogih drugih. Danas su genetski algoritmi podgrupa široke klase evolucijskih algoritama koji simuliranom evolucijom pronalaze rješenja. [8]

U ovom se radu ostvaruje implementacija novije vrste genetskog algoritma koja koristi paradigmu kvantne mehanike za predstavljanje i manipuliranje podacima. Praktični dio implementiran je u sklopu evolucijskog razvojnog okruženja ECF (*Evolutionary Computation Framework*) razvijenog na FER-u. U implementaciji koristim aproksimaciju qubita na koju još nisam naišao u ostalim radovima i člancima, a za koju vjerujem da je isplativija u pogledu memorijskog zauzeća i kompleksnosti implementacije. Algoritam je prilagođen korištenju 3 različite reprezentacije genotipa: niz bitova, niz decimalnih brojeva i bitovni niz s pretvorbom u niz decimalnih brojeva. Vrši se usporedba algoritma s klasičnim genetskim algoritmom na tri različita problema, svaki za jednu reprezentaciju genotipa. Radu se prilažu svi rezultati te korišteni parametri.

Rad je raspoređen na sljedeći način: u poglavlju 2 ukratko se navodi povijest i opisuje značaj kvantne mehanike, objašnjavaju neki temeljni pojmovi potrebni za shvaćanje paradigme računanja kvantnim računalima te trenutna tržišna scena istih. U poglavlju 3 opisuje se struktura klasičnog genetskog algoritma i kvantnog genetskog algoritma te detaljnije genetski algoritam inspiriran kvantnom mehanikom. U poglavlju 4 opisuju se i prikazuju rezultati eksperimenata postignuti algoritmima CGA i GAIQM. U poglavlju 5 predlažu se moguće izmjene i nadogradnje algoritma. U dodatku A prilaže se popis svih podržanih parametara za komponente GAIQM.

2. Kvantna mehanika i kvantna računala

Do početka 19. stoljeća fizičari su imali deterministički pogled na svijet. Razvijanjem tehnologije koja je omogućavala promatranje sve manjih čestica primjećuju se čudni fenomeni koje dotadašnja fizika nije mogla opisati. Iz potrebe za objašnjavanjem tih neobičnih pojava razvijene su podjednako neobične teorije koje, iako su matematički dokazane, nisu olako prihvaćene na znanstvenoj sceni. Jedna takva teorija je pretpostavka diskretne prirode stvarnosti. Iz nje je nastalo jedno čitavo područje fizike zvano '*kvantna mehanika*'. Iako su puno ranije postojala slična razmišljanja o diskretnim i točkastim česticama (Galilei, Bošković), do 19. stoljeća nije postojala ni tehnologija kojom bi se tvrdnje dokazale ni matematika kojom bi se tvrdnje opisale.

Početkom 19. stoljeća, Max Planck objavljuje "*Plankov zakon zračenja*" kojim rješava dugogodišnji problem opisivanja zračenja crnog tijela, popularno nazvan "*ultraljubičasta katastrofa*". Definira povezanost izračene energije s valnom duljinom izračene svjetlosti, ali još bitnije definira diskretizirano zračenje u paketićima energije koje naziva 'količina', latinski 'kvantum'. Njegovim otkrićem kreće revolucija u fizici od kojih ću spomenuti samo nekoliko: otkriće fotoelektričnog efekta (Einstein), ispravljanje (redefiniranje) modela atoma (Rutherford, Bohr), otkriće dvojne prirode čestica (De Broglie), opisivanje valne prirode elektronske ljuske atoma (Heisenberg, Schrödinger, Born). [7]

Danas je kvantni pogled na svijet vrlo uvriježen u krugu fizičara i šire, no još nije u potpunosti shvaćen. Unatoč tome korištenjem fenomena proizišlih iz kvantne fizike stvorili smo razne uređaje i alate koji su danas u širokoj primjeni. Neki takvi uređaji su: tunel dioda, laser, magnetska rezonancija i drugi.

Ukratko, kvantna mehanika je relativno mlada grana fizike, ali je uvelike utjecala na razvoj tehnologije, znanosti i shvaćanja svijeta u kojem živimo.

2.1. Temeljni pojmovi kvantne mehanike

2.1.1. Kvantna superpozicija

Jedan je od osnovnih i nama najvažnijih pojmova kvantne mehanike. Kvantna superpozicija je princip koji definira zbroj (superpoziciju) dvaju ili više kvantnih stanja kao ispravno kvantno stanje. To znači da je kvantno stanje kombinacija više stanja s pridijeljenom vrijednosti (važnosti) svakog stanja.

Kvantnom superpozicijom deterministički je model atoma s definiranim dozvoljenim putanjama elektrona zamijenjen puno preciznijim kvantnim modelom atoma koji ne definira točne putanje elektrona već prostornu distribuciju vjerojatnosti pronalaženja elektrona u određenoj točki prostora. Slikovito možemo reći da je model atoma s kružnim putanjama elektrona oko jezgre zamijenjen modelom jezgre okružene elektronskim oblacima. [7]

Jedan od najpoznatijih alegorija superpozicije je Schrödingerov misaoni eksperiment zvan 'Schrödingerova mačka'. Zamislimo zatvorenu čeličnu kutiju u koju smo stavili mačku i Geigerov brojač unutar kojeg se nalazi uređaj s otrovom. Uređaj u sebi ima komadić radioaktivnog elementa, toliko malen da se u vremenskom rasponu od 1 sat može (ali i ne mora) dogoditi raspad atoma. U slučaju raspada Geigerov bojač aktivira relej koji ispusti čekić na malenu staklenu ampulu cijanovodične kiseline. Ostavimo li sustav zatvorenim na sat vremena možemo reći da je mačka još uvijek živa ako u međuvremenu nije došlo do raspada atoma jer prvi bi raspad otrovao mačku. Psi (valna) funkcija čitavog sustava bi to opisala stanjima žive i mrtve mačke jednakog značaja, odnosno drugačije rečeno jednakih vjerojatnosti pojavljivanja. Stvarno stanje ne možemo znati s potpunom sigurnošću dok je otvorimo kutiju i pogledamo u nju.

Iz primjera vidimo da se sustav (ili čestica) može nalaziti u superpoziciji dok god ne promotrimo (izmjerimo) sustav. Superpozicija je opisana valnom funkcijom, a mjerenjem sustava gubi se superpozicija odnosno događa se kolaps valne funkcije u jedno deterministično stanje. Dakle, mjerenjem sustava dobivamo klasično stanje čestice u

kojem se ona tog trena našla i upravo to je naša poveznica između "kvantnog" i "klasičnog" svijeta koja je nužna za ostvarivanje upotrebljivog kvantnog računala.

2.1.2. Kvantna dekoherencija

Kvantna dekoherencija je kolaps valne funkcije odnosno nepovratni gubitak pohranjene informacije zbog utjecaja okoline koja ga okružuje. U stvarnim kvantnim sustavima relativno je lako ostvariti superpoziciju, ali je veliki problem održati ju. Ovisno o tehnologiji sustava razni uvjeti moraju biti ostvareni kako bi se sustav izolirao od okoline i zaštitio integritet pohranjene informacije. Iz tog su razloga kvantni sustavi danas glomazni i zahtjevni za održavanje.

2.1.3. Spregnuta stanja (engl. *entangled states*)

Spregnuta kvantna stanja povezana su (spregnuta) ako rezultat mjerenja jednog kvantnog stanja utječe na rezultat mjerenja drugog kvantnog stanja. Zanimljivost te veze je da se "informacija" o rezultatu mjerenja jednog stanja u istom trenu prenese drugom stanju. Poznati paradoks koji opisuje tu pojavu je "Einstein–Podolsky–Rosen paradoks".

2.1.4. "No-cloning theorem"

Ovaj je teorem jedna velika zapreka u kvantnoj fizici. Njime se potvrđuje da nije moguće stvoriti identične kopije postojećeg kvantnog stanja. Želimo li stvoriti još jednu česticu nalik ovoj koju već imamo, moramo ponoviti čitav proces stvaranja te čestice. [5]

Iako teorem djeluje vrlo ograničavajuće može se upotrijebiti u kvantnoj kriptografiji. Problem sigurnosti razmjene javnog ključa na kojoj se zasniva moderna sigurnost Interneta može se riješiti kvantnom enkripcijom kojom se garantira nemogućnost kopiranja odaslanog ključa. Čitanjem ključa uništava se njegovo kvantno stanje te ga nije moguće kopirati što pospješuje zaštitu od "*Man-in-the-middle*" napada. [2]

2.2. Kvantni bit (qubit)

Kvantni bit je najmanja jedinica podatka u kvantnim računalima. U literaturi se može pronaći i pod nazivima "*qbit*" i "*Qbit*". Fizički gledano to je zapravo sićušna čestica sa

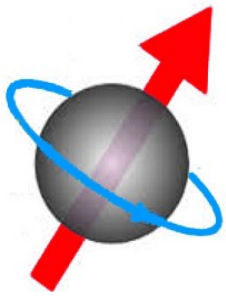
svojstvom superpozicije.

2.2.1. Princip rada

Klasični bit pronalazimo u jednom od dva klasična stanja. Jednom postavljeno stanje se pamti i uvijek ga u njemu možemo pronaći. Njima gradimo binarni brojevni sustav pomoću kojeg zapisujemo podatke na računalu.

Kvantni bit ne pamti jedno klasično stanje već se on istovremeno nalazi u više različitih klasičnih stanja. To se svojstvo naziva superpozicija. Fizički gledano kvantni bit je čestica kojoj je pridružen kvantizirani spin. Spin je kvantizirana inačica kutne količine gibanja rotirajućeg tijela.

Kvantno stanje jednog qubita opisano je vektorom njegovog spina. Kvantno stanje opisuje vjerojatnost pronalaženja qubita u svakom od klasičnih stanja. Konkretno stanje dobivamo mjerenjem qubita kojim se uništava njegovo svojstvo superpozicije te dobivamo česticu u jednom stanju koju možemo koristiti poput klasičnog bita.

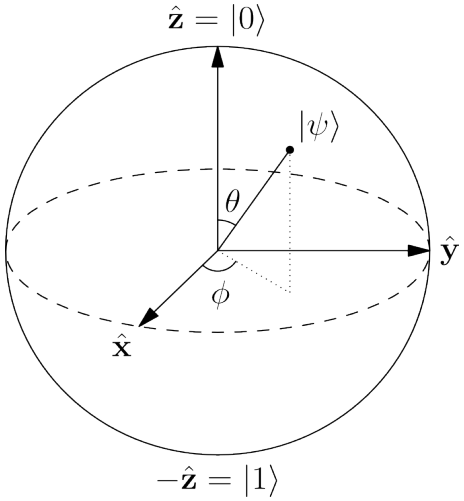


Slika 2.1: Vjerojatnost pronalaženja qubita u određenom stanju opisana je njegovim kvantnim stanjem (spinom).

Sustavi s kvantnim bitovima mogu biti definirani za proizvoljan brojevni sustav, no s obzirom na to da su današnja računala bazirana na binarnom brojevnom sustavu zanima nas definicija qubita s 2 stanja.

2.2.2. Blochova shema

Stanje qubita prikazujemo Blochovom shemom. Jedno kvantno stanje prikazujemo kao vektor iz središta sfere na njezinu površinu. Stanja u kojima možemo pronaći qubit prikazujemo polovima sfere koje definiramo sjecištima sfere i z-osi. [6] [9]



Slika 2.2: Blochova sfera

Na slici vidimo jedno kvantno stanje opisano vektorom. Matematički ga možemo opisati kao linearnu kombinaciju dvaju klasičnih stanja:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

gdje su α i β kompleksni brojevi.

Iz vektora na Blochovoj sferi definiramo konstante iz (2.1):

$$\begin{aligned} |\psi\rangle &= \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\cdot\sin\left(\frac{\theta}{2}\right)|1\rangle \\ &= \cos\left(\frac{\theta}{2}\right)|0\rangle + (cos(\phi) + i\sin(\phi))\cdot\sin\left(\frac{\theta}{2}\right)|1\rangle \end{aligned} \tag{2.2}$$

Vrijednosti α i β su normirane:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.3}$$

Vrijednosti α i β predstavljaju vjerojatnosne amplitude pronalaženja qubita u stanju $|0\rangle$ i $|1\rangle$ respektivno. Iz vjerojatnosnih amplituda računamo vjerojatnosti pronalaženja qubita u svakom od stanja:

$$\begin{aligned} \Pr\{|0\rangle\} &= |\alpha|^2 = \cos^2\frac{\theta}{2} \\ \Pr\{|1\rangle\} &= |\beta|^2 = \sin^2\frac{\theta}{2} \end{aligned} \tag{2.4}$$

Kvantno stanje možemo zapisati vektorom što je pogodno za izvršavanje matričnih operacija nad njime. [6]

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{2.5}$$

2.2.3. Aproksimacija kvantnog stanja

Kvantno stanje prikazano na Blochovoj sferi opisano je s 3 dimenzije. Trodimenzionalne operacije računski su vrlo zahtjevne stoga želimo pojednostaviti model kvantnog stanja. U simulaciji rada s qubitom jedino su nam važne vjerojatnosti pronalaženja qubita u određenom stanju. Iz formula (2.3) i (2.4) zaključujemo da vjerojatnost $|\beta|^2$ ovisi isključivo o vjerojatnosti $|\alpha|^2$:

$$\begin{aligned} |\alpha|^2 + |\beta|^2 &= 1 \\ |\beta|^2 &= 1 - |\alpha|^2 \\ &= 1 - \cos^2\frac{\theta}{2} \\ &= \sin^2\frac{\theta}{2} \end{aligned}$$

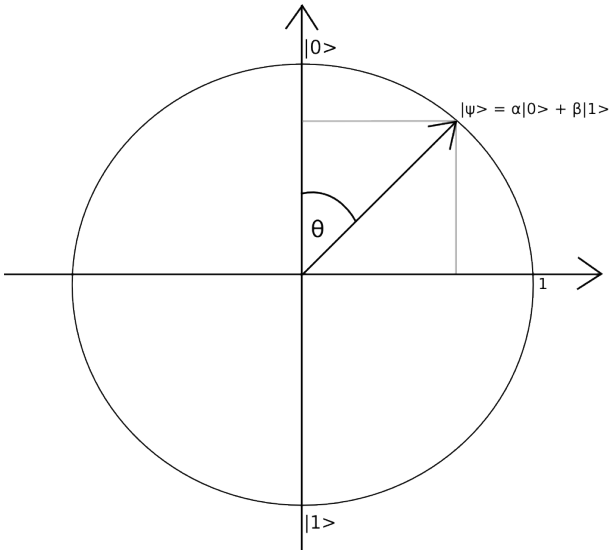
Ako bolje pogledamo kvantno stanje opisano formulom (2.2) vidjet ćemo da α ovisi samo o kutu θ. Želimo se riješiti kompleksne konstante β ili ju barem svesti na realnu. Iz jednakosti (2.6) zaključujemo da je kut ϕ suvišan za računanje vjerojatnosti. To možemo vidjeti i na samoj Blochovoj sferi. Za fiksni kut θ mijenjanjem kuta ϕ rotiramo vektor oko z-osi i ne približavamo se niti jednom polu sfere odnosno vjerojatnosti stanja se ne mijenjaju.

$$|cos(\phi) + i\sin(\phi)| = 1 \tag{2.6}$$

Dakle, parametre α i β sveli smo na realne vrijednosti koje ovise samo o jednom parametru: θ. Naše kvantno stanje možemo opisati formulom:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.7}$$

Aproksimacijom qubita uz zadržane vjerojatnosti stanja dobivamo sljedeću reprezentaciju kvantnog stanja qubita:



Slika 2.3: Aproksimacija qubita

$$\alpha = \cos\frac{\theta}{2} \qquad \beta = \sin\frac{\theta}{2} \tag{2.8}$$

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.9}$$

2.2.4. Kvantni registar

Kvantni registar, poput klasičnog registra, pohranjuje niz qubita te omogućuje nedestruktivno čitanje i zapisivanje istih. Ovdje čitanje predstavlja korištenje (izmjena) kvantnog stanja zapisanog u njemu, a ne mjerenje koje bi uništilo superpoziciju qubita.

$$\begin{bmatrix} q_1 & q_2 & q_3 & \cdots & q_i & \cdots & q_n \end{bmatrix}$$

Slika 2.4: Registar qubita

Simulacija kvantnog registra koju implementiraju članci na koje sam naišao je klasični registar koji pohranjuje vjerojatnosne amplitude qubita pomoću kojih se određuju vjerojatnosti stanja. [3] [4] [2]

Koristeći aproksimaciju qubita zadanu formulom 2.9 zapis stanja možemo dodatno pojednostaviti tako da u klasični registar pohranjujemo samo kuteve qubita pomoću kojih možemo izračunati vjerojatnosti stanja.

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_i & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_i & \cdots & \beta_n \end{bmatrix}$$

Slika 2.5: Registar aproksimiranih qubita zapisom gustoća vjerojatnosti

$$\begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_i & \cdots & \theta_n \end{bmatrix}$$

Slika 2.6: Registar aproksimiranih qubita zapisom kuteva kvantnog stanja

Zadnji sam model kvantnog registra odabrao za implementaciju upravo zbog svoje jednostavnosti i memorijske isplativosti. Kasniji će se operatori također pojednostaviti zahvaljujući ovoj aproksimaciji.

2.3. Kvantni algoritmi i kvantna računala

Kvantno računalo je uređaj koji je sposoban izvršavati matematičke i logičke zadatke korištenjem fenomena kvantne mehanike. Temeljna podatkovna jedinica klasičnog računala je bit, a kvantnog računala qubit. Prednost qubita je što posjeduje svojstvo superpozicije, odnosno može pohraniti više stanja odjednom. Jedan primjer primjene qubita je kvantni registar kao spremnik podataka kojim se opisuje značaj superpozicije:

> Pomoću 1 klasičnog registra veličine n bitova moguće je pohraniti 1 od mogućih 2^n vrijednosti.
>
> Pomoću 1 kvantnog registra veličine n qubita moguće je pohraniti 2^n od mogućih 2^n vrijednosti, odnosno sve vrijednosti odjednom.

Zanimanje za kvantnim računalima počinje rasti 1980-tih godina govorom Richarda Feynmana u kojem Feynman iznosi i obrazlaže zabrinutost da se kvantni sustavi ne mogu učinkovito simulirati klasičnim računalom, umjesto kojeg predlaže osnovni model kvantnog računala. [5]

Kvantni algoritam je algoritam koji je prilagođen izvršavanju na kvantnom računalu. Kvantnim algoritmima moguće je ostvariti značajna ubrzanja nekih NP problema. [1] [2] Primjeri takvih algoritama su:

– Shorov algoritam faktorizacije cijelih brojeva - moguća primjena u otkrivanju javnog ključa RSA
– Groverov algoritam invertiranja funkcije - moguća primjena je pretraživanje baze podataka
– Algoritmi simulacije kvantnih sustava

2.4. Trenutno stanje u svijetu

Porastom snage i pouzdanosti kvantnih računala krenulo se stvarati tržište za njih. Danas neke od najvećih tvrtki ulažu u razvoj kvantnih računala. To su većinom banke, vojska i IT industrija. Danas su najpoznatije dvije tvrtke koje proizvode i razvijaju kvantna računala.

D-Wave

Tvrtka D-Wave razvija i proizvodi kompaktne zatvorene sustave kvantnih računala. Trenutno najnoviji model kvantnog računala tvrtke D-Wave je 2000Q, podržava 2000 qubita i temelji se na adijabatskom kvantnom računanju. Rješava probleme pretraživanja prostora principom sličnim simuliranom kaljenju. Ta su računala nedostupna široj javnosti isključivo radi cijene (15 milijuna američkih dolara, 05.2017.) no neke su ih tvrtke ipak nabavile (Google, NASA, USRA, Volkswagen Group i ostali).

IBM

IBM nudi udaljeni pristup svojim kvantnim računalima koja su razvili. Javni paket uključuje pristup web aplikaciji za dizajniranje kvantnih krugova do 5 qubita te mogućnost izvršavanja eksperimenta na simulatoru ili stvarnom kvantnom procesoru. Pristup stvarnom kvantnom procesoru naplaćuje se 'virtualnim kreditom' koji se dobiva registracijom na sustav i vremenski obnavlja. Svako izvršavanje se naplaćuje ovisno o zadanom broju ponoavljanja eksperimenta. Početkom 2017. godine omogućen je pristup računalu s čak 20 qubita preko web aplikacije. Postoji API za razvoj kvantnih krugova pomoću "kvantnog asemblera" (QASM), a u tijeku je izrada API-a za pristup sustavu kojim bi se omogućio razvoj algoritama i aplikacija u više programskih jezika.

3. Genetski algoritam inspiriran kvantnom mehanikom

U ovom ću odlomku opisati sličnosti između klasičnog i kvantnog genetskog algoritma te primijenjenu aproksimaciju qubita u genetskom algoritmu inspiriranom kvantnom mehanikom. Posebno ću opisati GAIQM i kako sam ga implementirao.

Svi navedeni algoritmi koriste i evaluator. To je procedura koja jedinci pridružuje vrijednost dobrote (engl. *fitness*) prema kojoj se poželjnije jedinke ističu iz populacije. Gradi se ovisno o zadanom problemu i ne ovisi o implementaciji algoritma odnosno jednom napisani evaluator je kompatibilan sa svim algoritmima. Ovisi jedino o načinu na koji se zapisuju podaci koji se evoluiraju tj. ovisi o vrsti genotipa.

3.1. Klasični genetski algoritam (CGA)

3.1.1. Dijelovi

Klasični genetski algoritam zahtjeva nekoliko osnovnih dijelova:

- Genotip
- Operator križanja
- Operator mutacije

Genotip

Podatkovna struktura koja opisuje jednu jedinku. Mora biti kompatibilna s evaluatorom i zahtjeva sebi kompatibilne operatore. Sadrži kromosom u obliku niza bitova.

$$\begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_i & \cdots & b_n \end{bmatrix}$$

Slika 3.1: Registar klasičnih bitova

Operator križanja

Operator koji izvodi križanje (razmjenu) genetskog materijala danih jedinki. Odabire se točka koja će podijeliti kromosom roditelja na 2 dijela. Stvaraju se djeca kojima je jedna polovica roditelja zamijenjena polovicom drugog roditelja.

$$\begin{aligned} Roditelj1 : &\begin{bmatrix} \mathbf{b_{11}} & \mathbf{b_{12}} & | & b_{13} & \cdots & b_{1i} & \cdots & b_{1n} \end{bmatrix} \\ Roditelj2 : &\begin{bmatrix} \mathbf{b_{21}} & \mathbf{b_{22}} & | & b_{23} & \cdots & b_{2i} & \cdots & b_{2n} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} Dijete1 : &\begin{bmatrix} \mathbf{b_{21}} & \mathbf{b_{22}} & b_{13} & \cdots & b_{1i} & \cdots & b_{1n} \end{bmatrix} \\ Dijete2 : &\begin{bmatrix} \mathbf{b_{11}} & \mathbf{b_{12}} & b_{23} & \cdots & b_{2i} & \cdots & b_{2n} \end{bmatrix} \end{aligned}$$

Slika 3.2: Križanje kromosoma zamjenom polovica roditeljskih kromosoma

Operator mutacije

Operator koji vrši izmjenu genetskog materijala dane jedinke. Odabire se bit koji se invertira.

$$\begin{bmatrix} b_1 & b_2 & \mathbf{b_3} & \cdots & b_i & \cdots & b_n \end{bmatrix}$$
$$\begin{bmatrix} b_1 & b_2 & \overline{\mathbf{b_3}} & \cdots & b_i & \cdots & b_n \end{bmatrix}$$

Slika 3.3: Mutacija invertiranjem jednog bita kromosoma

3.1.2. Pseudokod

U nastavku je dan pseudokod općenitog genetskog algoritma koji se izvršava na klasičnom računalu.

Algoritam 1 Klasični genetski algoritam (CGA)

Input: $parametri$
$InicijalizirajPopulaciju$ P(0)
$EvaluirajPopulaciju$ P(0)
B = $DohvatiNajboljegIz$ P(0)
while $NijeDovoljnoDobar$ B **do**
  t $\leftarrow$ t + 1
  $OdaberiRoditeljeIz$ P(t)
  $OperatorKrizanja$
  $OperatorMutacije$
  $EvaluirajPopulaciju$ P(t)
end while
return B

3.2. Kvantni genetski algoritam (QGA)

3.2.1. Dijelovi

Kvantni genetski algoritam uz svoje specifične operatore sadrži i dijelove slične onima iz CGA:

- Kvantni genotip (kvantni registar)
- Kvantna rotacijska vrata
- Inicijalizator populacije
- Operator mjerenja qubita

Kvantni genotip (kvantni registar)

Podatkovna struktura koja opisuje jednu jedinku. Od klasičnog registra se razlikuje u principu pohrane podataka (superpozicija). Zahtjeva operatore sposobne za baratanje qubitima. Dodatno zahtjeva i klasični genotip koji može pohraniti izmjerene vrijednosti qubita i koji je pogodan evaluatoru. Sadrži kromosom u obliku niza qubita.

$$\begin{bmatrix} q_1 & q_2 & q_3 & \cdots & q_i & \cdots & q_n \end{bmatrix}$$

Slika 3.4: Registar qubita

Operator kvantnih rotacijskih vrata

Ovim se operatorom mijenja kvantno stanje qubita. Operator se koristi za unaprjeđenje populacije transformiranjem kromosoma. Matematički se može prikazati kao matrica koja mora biti unitarna odnosno mora vrijediti: $UU^{\dagger} = I$.

Prisjetimo se da smo kvantno stanje qubita opisali vektorom na Blochovoj sferi. Želimo li promijeniti stanje qubita moramo rotirati svojstveni vektor. Upravo nam tome služe kvantna rotacijska vrata. Najvažnija vrata su vrata identiteta i Paulijeva X, Y i Z vrata:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.1}$$

Matematički primjenu kvantnih rotacijskih vrata ostvarujemo umnoškom kvantnog stanja zapisanog u vektor i samih vrata, npr. ostvarenje zrcaljenja oko x-osi:

$$X \cdot |\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \tag{3.2}$$

Kvantna rotacijska vrata u QGA mogu poprimiti više namjena. U prethodnom primjeru zrcaljenjem oko x-osi zamjenjujemo vjerojatnosti pronalaska bita u oba stanja, odnosno ostvarujemo operaciju sličnu inverziji bita koja se koristi za mutaciju kromosoma. Blagom rotacijom kvantnog stanja oko x-osi ili y-osi približavamo se stanju $|0\rangle$ ili $|1\rangle$ te tako ostvarujemo pretraživanje prostora stanja.

Inicijalizator populacije

Svrha inicijalizatora populacije je postaviti qubite u superpoziciju. Ostvaruje se Hadamardovim H vratima:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.3}$$

Matematički prikaz postavljanja qubita u superpoziciju:

$$H \cdot |\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} \alpha + \beta \\ \alpha - \beta \end{bmatrix} \tag{3.4}$$

Na samom početku qubit nije u superpoziciji te se može nalaziti u stanju $|0\rangle$ ili $|1\rangle$. Primjenom Hadamardovih vrata na takve qubite dobivamo dvije istovjetne superpozicije

stanja.

$$
\begin{aligned}
H \cdot |0\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1+0 \\ 1-0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \\
H \cdot |1\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 0+1 \\ 0-1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}
\end{aligned}
\tag{3.5}
$$

Da su superpozicije istovjetne zaključujemo iz formule 2.4 kojom dobivamo jednake vjerojatnosti za oba stanja.

$$
|\alpha|^2 = |\beta|^2 = \frac{1}{2} \tag{3.6}
$$

Operator mjerenja qubita

Operator mjerenja qubita uništava kvantnu superpoziciju qubita i pretvara ju u klasično deterministično stanje, odnosno pretvara qubit u bit. Kvantnim stanjem qubita opisana je vjerojatnost pronalaženja qubita u jednom klasičnom stanju.

3.2.2. Pseudokod

U nastavku je dan pseudokod kvantnog genetskog algoritma koji se izvršava na kvantnom računalu.

Algoritam 2 Kvantni genetski algoritam (QGA)

Input: $parametri$
$InicijalizirajPopulaciju$ Q(0)
$OperatorMjerenja$ Q(0) $\rightarrow$ P(0)
$EvaluirajPopulaciju$ P(0)
B = $DohvatiNajboljegIz$ P(0)
while $NijeDovoljnoDobar$ B **do**
  t $\leftarrow$ t + 1
  $OperatorKvantnihRotacijskihVrata$ Q(t)
  $OperatorMjerenja$ Q(t) $\rightarrow$ P(t)
  $EvaluirajPopulaciju$ P(t)
  B = $DohvatiNajboljegIz$ P(t)
end while
return B

3.3. Genetski algoritam inspiriran kvantnom mehanikom (GAIQM)

Genetski algoritam inspiriran kvantnom mehanikom je simulacija QGA. U literaturi ga možemo pronaći i pod nazivom *Quantum inspired genetic algorithm (QIGA)*. [2]

3.3.1. Dijelovi

GAIQM sadrži dijelove CGA i QGA prilagođene korištenoj aproksimaciji qubita:

- Kvantni genotip (kvantni registar)
- Operator kvantnih rotacijskih vrata
- Kvantni operator mutacije
- Operator katastrofe
- Inicijalizator populacije
- Operator mjerenja qubita

Kvantni genotip (kvantni registar)

Podatkovna struktura koja opisuje jednu jedinku. Od stvarnog kvantnog registra se razlikuje u principu pohrane podataka. Superpozicija qubita zapisana je aproksimacijom kvantnog stanja qubita (formula 2.9) tj. kutem kvantnog stanja.

$$\begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_i & \cdots & \theta_n \end{bmatrix}$$

Slika 3.5: Registar kuteva aproksimiranog kvantnog stanja qubita

Operator kvantnih rotacijskih vrata

Rotaciju qubita proizvoljnim kutem $\delta\theta$ u dvodimenzionalnom prostoru ostvarujemo rotacijskim vratima:

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} \cos(\delta\theta) & -\sin(\delta\theta) \\ \sin(\delta\theta) & \cos(\delta\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3.7}$$

Koristeći aproksimaciju qubita zadanu formulom 2.9 operator rotacije možemo pojednostaviti:

$$\begin{aligned}
\begin{bmatrix}\alpha' \\ \beta'\end{bmatrix} &= \begin{bmatrix}\cos(\delta\theta) & -\sin(\delta\theta) \\ \sin(\delta\theta) & \cos(\delta\theta)\end{bmatrix} \begin{bmatrix}\alpha \\ \beta\end{bmatrix} \\
&= \begin{bmatrix}\cos(\delta\theta) & -\sin(\delta\theta) \\ \sin(\delta\theta) & \cos(\delta\theta)\end{bmatrix} \begin{bmatrix}\cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right)\end{bmatrix} \\
&= \begin{bmatrix}\cos\left(\frac{\theta}{2}\right)\cos(\delta\theta) - \sin\left(\frac{\theta}{2}\right)\sin(\delta\theta) \\ \cos\left(\frac{\theta}{2}\right)\sin(\delta\theta) + \sin\left(\frac{\theta}{2}\right)\cos(\delta\theta)\end{bmatrix} \\
&= \begin{bmatrix}\cos\left(\frac{\theta}{2} + \delta\theta\right) \\ \sin\left(\frac{\theta}{2} + \delta\theta\right)\end{bmatrix} \\
\begin{bmatrix}\cos\left(\frac{\theta'}{2}\right) \\ \sin\left(\frac{\theta'}{2}\right)\end{bmatrix} &= \begin{bmatrix}\cos\left(\frac{\theta+2\delta\theta}{2}\right) \\ \sin\left(\frac{\theta+2\delta\theta}{2}\right)\end{bmatrix}
\end{aligned}$$

Zaključujemo da je matrično množenje moguće zamijeniti formulom znatno manje složenosti:

$$\theta' = \theta + 2 \cdot \delta\theta \tag{3.8}$$

Kut zakreta $\delta\theta$ je moguće zadati pomoću tablice kuteva tzv. *look-up table* u kojoj za svaki mogući slučaj određujemo smjer i veličinu zakreta. Tablicu možemo proizvoljno zadati, a standardni oblik tablice dan je u nastavku:

Tablica 3.1: Standardna *look-up* tablica

q_i	b_i	f(q)>f(b)	$\Delta\theta$	sg(α_i, β_i)			
				$\alpha_i \cdot \beta_i > 0$	$\alpha_i \cdot \beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	$\perp$	0	-	-	-	-
0	0	$\top$	0	-	-	-	-
0	1	$\perp$	ϕ	+1	-1	0	± 1
0	1	$\top$	ϕ	-1	+1	± 1	0
1	0	$\perp$	ϕ	-1	+1	± 1	0
1	0	$\top$	ϕ	+1	-1	0	± 1
1	1	$\perp$	0	-	-	-	-
1	1	$\top$	0	-	-	-	-

Za zadani i-ti qubit neke jedinke q i i-ti qubit najbolje jedinke b, ovisno o odnosu vrijednosti dobrote jedinki odabiremo redak u tablici. Kut zakreta čitamo iz retka na sljedeći način:

$$\delta\theta = \Delta\theta \cdot sg(a, b)$$

gdje $\Delta\theta$ određuje iznos zakreta, a $sg(\alpha_i, \beta_i)$ smjer zakreta. Vrijednost kut ϕ se najčešće odabire iz intervala $[0.005\pi, 0.01\pi]$.

U praksi se pokazalo da male promjene vrijednosti zadanih tablicom uvelike utječu na brzinu i pouzdanost čitavog algoritma. Ovim pristupom određivanja kuta moguće je ostvariti prilagodbu algoritma pojedinom problemu i ostvariti dobre rezultate, no ugađanje konstanti nije učinkovito zbog njihove brojnosti, ima ih čak 40. Stvara se i problem fiksne vrijednosti zakreta zbog koje dolazi do prebrze konvergencije populacije čime ona gubi raznolikost i često završava u lokalnom optimumu. Prevelik kut zakreta sprječava finu pretragu oko pronađenog optimuma što ograničava preciznost pretrage.

Iz tih se razloga koristi adaptivno ugađanje vrijednosti zakreta koje automatski prilagođava iznos kuta zakreta.[10] Umjesto fiksne vrijednosti kuta ϕ zadajemo najveću i najmanju vrijednost kuta između kojih će se interpolirati vrijednost kuta ϕ. Što je jedinka bliža najboljoj jedinci po vrijednosti dobrote, to je kut zakreta manji, odnosno kut zakreta proporcionalan je razlici vrijednosti dobrota jedinki. Formula za određivanje kuta je sljedeća:

$$\delta\theta = l + (h - l) \cdot \frac{f(q) - f(w)}{f(b) - f(w))} \tag{3.9}$$

gdje l i h predstavljaju respektivno iznos donje i gornje granice kuta zakreta, $f(q)$ vrijednost funkcije dobrote za jedinku koju usmjeravamo, a $f(b)$ i $f(w)$ vrijednost funkcije dobrote za respektivno najbolju i najgoru jedinku u trenutnoj populaciji. Ovom je formulom omogućen valjan odabir kuta zakreta istovremeno za probleme minimizacije i maksimizacije.

Kvantni operator mutacije

Koriste se dvije vrste mutacija koje se izvršavaju ili ne ovisno o zadanoj vjerojatnosti izvršavanja: inverzija i zamjena.

Operator inverzije nasumično određuje jedan qubit jedinke te ga invertira. Mutacija je izvedena po uzoru na onu iz QGA. Obavlja se zrcaljenje bita s obzirom na x-os. Aproksimacijom stanja qubita samo jednim kutem (2.8) ograničili smo kut θ na interval $[-\pi, \pi]$ te omogućili pojednostavljeno zrcaljenje kuta:

Algoritam 3 Kvantni operator mutacije za GAIQM

Input: θ
if $\theta > 0$ **then**
  return $\pi - \theta$
else
  return $-\pi - \theta$
end if

Operator zamjene nasumično odabire dva qubita te im zamjenjuje mjesta.

$$\begin{bmatrix} \boldsymbol{\theta_1} & \theta_2 & \boldsymbol{\theta_3} & \cdots & \theta_i & \cdots & \theta_n \end{bmatrix}$$
$$\begin{bmatrix} \boldsymbol{\theta_3} & \theta_2 & \boldsymbol{\theta_1} & \cdots & \theta_i & \cdots & \theta_n \end{bmatrix}$$

Slika 3.6: Zamjena dva nasumično odabrana qubita

Operator katastrofe

Uporabom operatora kvantnih rotacijskih vrata algoritam može konvergirati u lokalni optimum, koji nije globalni optimum, te u njemu zaglaviti. Same mutacije nisu dovoljne da bi se algoritam izvukao, već je potreban snažniji operator.

Operator katastrofe unosi velik poremećaj u populaciju čime ostvaruje veću raznolikost populacije, a time i pretraživanje šireg prostora stanja. Operator sve jedinke populacije izuzev najbolje, vraća u početno stanje superpozicije. Te jedinke tada ponovno konvergiraju ("iz nule"), ali ovog puta odmah kreću prema pronađenoj najboljoj jedinci. Iako se najbolja jedinka nalazi u lokalnom optimumu, nasumično stvorene jedinke započinju svoju konvergenciju iz više različitih stanja povećavajući vjerojatnost pretraživanja dosad neistraženog prostora.

Prečestim pozivanjem operatora može se blokirati svojstvo usmjerenog razvoja populacije čime se pretraživanje pretvara u nasumično. To reguliramo parametrom ponovljenih optimuma. Ideja je pratiti u koliko se uzastopnih generacija ponavlja ista vrijednost najbolje jedinke, a kada taj broj prekorači vrijednost parametra izvršavamo katastrofu.

Algoritam 4 Okidač katastrofe

Input: B *// Vrijednost dobrote najbolje jedinke u populaciji.*
if $prekoracen(parametarOkidanja)$ **then**
  $resetirajBrojac()$
  return true
else if $najboljiSePonovio(B)$ **then**
  $uvecajBrojac()$
else
  $resetirajBrojac()$
  $zapamtiNajboljeg(B)$
end if
return false

Inicijalizator populacije

Inicijalizator postavlja sve qubite u stanje superpozicije $|\psi_0\rangle$. Zbog aproksimacije qubita definirane formulom 2.9 potrebno je sve kuteve inicijalizirati na $\pm\frac{\pi}{2}$.

Operator mjerenja

Operatorom mjerenja superpoziciju pretvaramo u klasično stanje odnosno qubit pretvaramo u bit. Koje će stanje poprimiti pojedini qubit ovisi o njegovom kvantnom stanju prema formuli za izračun vjerojatnosti (2.4).

Algoritam 5 Operator mjerenja

Input: θ_i *// Kut kvantnog stanja i-tog qubita.*
if $nasumicanBroj(0,1) > \cos^2(\theta_i/2)$ **then**
  return 1
else
  return 0
end if

3.3.2. Pseudokod

U nastavku je dan pseudokod genetskog algoritma inspiriranog kvantnom mehanikom koji se izvršava na klasičnom računalu.

Algoritam 6 Genetski algoritam inspiriran kvantnom mehanikom (GAIQM)

Input: $parametri$
$InicijalizirajPopulaciju$ Q(0)
$OperatorMjerenja$ Q(0) $\rightarrow$ P(0)
$EvaluirajPopulaciju$ P(0)
B = $DohvatiNajboljegIz$ Q(0), P(0)
while $NijeDovoljnoDobar$ B **do**
  t $\leftarrow$ t + 1
  if $OkidacKatastrofe$ B **then**
    $OperatorKatastrofe$ B, Q(t)
  end if
  $OperatorKvantnihRotacijskihVrata$ B, Q(t)
  $OperatorMutacije$ Q(t)
  $OperatorMjerenja$ Q(t) $\rightarrow$ P(t)
  $EvaluirajPopulaciju$ P(t)
  B = $DohvatiNajboljegIz$ Q(t), P(t)
end while
return B

3.4. Implementacija algoritma

Algoritam je implementiran na aproksimaciji qubita i kvantnog registra.

Kvantni genotip (QuantumRegister)

Genotip nasljeđuje klasu Binary koja svoj kromosom zapisuje u obliku niza bitova, a nudi i pretvorbu niza bitova u niz realnih brojeva. Definiranim parametrima A.3 određuje se brojnost i preciznost pretvorenih realnih brojeva. Dodatno sadrži i registar koji pamti samo vrijednosti kuteva aproksimiranih kvantnih stanja.

$$
\begin{aligned}
Registar\ bitova &\begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_i & \cdots & b_n \end{bmatrix} \\
Registar\ realnih\ brojeva &\begin{bmatrix} r_1 & r_2 & r_3 & \cdots & r_j & \cdots & r_m \end{bmatrix} \\
Registar\ kuteva &\begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \cdots & \theta_i & \cdots & \theta_n \end{bmatrix}
\end{aligned}
$$

Slika 3.7: $QuantumRegister$ sadrži 3 registra za reprezentaciju informacije

Sadrži metodu $measure()$ kojom se izvršava mjerenje registra. Iteracijom po kutevima računa se vjerojatnost i određuje klasično stanje (bit) koje se zapisuje u registar bitova. Algoritam koji određuje stanje qubita:

Algoritam 7 measure()

for θ_i : registarKuteva **do**
  b_i = randomDouble() > $\cos^2\left(\frac{\theta_i}{2}\right)$
end for

Metodom $resetQubits(StateP)$ ostvaruje se inicijalizacija superpozicije qubita. Parametrima A.3 moguće je odrediti kut početne superpozicije. Ova metoda nasumično postavlja pozitivan ili negativan početni kut za svaki qubit, te nakon toga poziva $measure()$ kako bi se iz qubita dobile vrijednosti bitova. Konačno poziva se metoda $update()$ naslijeđena iz klase $Binary$ koja pretvara binarni zapis registra bitova u realne vrijednosti ovisno o zadanim granicama i preciznosti realnih vrijednosti. Ova se metoda poziva i pri inicijalizaciji samog kvantnog registra.

Napomena: Nasljeđujući $Binary$ genotip sadrži sve potrebne podatkovne strukture i procedure klase $Binary$ te se može jednako koristiti. To znači da evaluatori koji koriste $Binary$ mogu direktno koristiti i $QuantumRegister$ bez potrebe za dodatnim (pomoćnim) genotipom što za taj slučaj ubrzava rad algoritma.

GAIQM

Implementacija GAIQM prati pseudokod 6. Algoritam interno koristi $QuantumRegister$ nad kojim izvršava sve operacije. Kako bi algoritam podržavao različite genotipe dodao sam adapter.

Adapter u obliku metode $adapter(individual, indexOfQreg)$ kopira odgovarajući sadržaj kvantnog registra u radni genotip koji se koristi u evaluatoru. Ovisno o tipu kopira samo realne brojeve ($FloatingPoint$) ili samo bitove ($BitString$ i $Binary$). Posebno za $Binary$ nakon kopiranja poziva se i metoda $update()$ kako bi se osvježili i realni brojevi tog genotipa. Metoda se neće pozvati ako je $QuantumRegister$ jedini zadani genotip. Jedinka može imati više genotipa koji su pohranjeni u niz. Varijabla $indexOfQreg$ služi kao indeks na kojem se nalazi genotip kvantnog registra.

Napomena: Algoritam se služi konvencijom da je $QuantumRegister$ uvijek posljednji genotip u jedinki. Zato treba paziti da se pri definiciji parametara genotipa $QuantumRegister$ uvijek navede posljednji.

MutQuantumInversion

Operator inverzije pri pozivu nasumično odabire jedan qubit u zadanom registru te ga invertira algoritmom 3.

MutQuantumSwap

Operator zamjene pri pozivu nasumično odabire dvije različite pozicije qubita u zadanom registru te im zamjeni vrijednosti kao što je prikazano slikom 3.6.

QuantumRotationGate

Koristi se pojednostavljena formula 3.8 i adaptivna metoda izračuna vrijednosti zakreta 3.9. Tablica predznaka je implementirana neizravno pomoću $if - else$ blokova.

4. Eksperimenti

Eksperiment se sastoji od usporedbe CGA i GAIQM na identičnom eksperimentu uz parametre za koje pojedini algoritam ostvaruje vrlo dobre rezultate. Terminator eksperimenata je maksimalan broj evaluacija koji je zadan ovisno o problemu koji rješava. Svaki eksperiment je ponovljen 40 puta, a prikazani rezultati su usrednjene vrijednosti svih ponovljenih eksperimenata.

4.1. Problem naprtnjače

4.1.1. Opis problema

Zadana je nosivost naprtnjače (engl. *knapsack*) i popis predmeta. Za svaki predmet zadana je njegova težina i vrijednost. Cilj je pronaći najvrjedniju kombinaciju stvari uz uvjet da nije prekoračena nosivost naprtnjače. Dobrota jedinke računa se sumom vrijednosti svih odabranih predmeta. Ovaj eksperiment koristi genotip $BitString$.

4.1.2. Rezultati

Za testiranje koristim nasumično generirani skup predmeta s ograničenom maksimalnom težinom predmeta, maksimalnom vrijednošću predmeta te definiranim brojem predmeta u setu. Koristim skupove od 500 i 1000 predmeta. Statistike skupova i parametri algoritma dani su u nastavku.

Tablica 4.1: Statistike skupova predmeta

Značajka	Skup 500	Skup 1000
Veličina naprtnjače	100	100
Broj predmeta	500	1000
Najveća vrijednost	100	100
Najmanja težina	1	1
Najveća težina	70	100
Prosječna težina	36.598	51.365
Optimalna vrijednost	2922	3616

Tablica 4.2: Korišteni parametri za problem naprtnjače

SteadyStateTournament		GAIQM	
Parametar	Vrijednost	Parametar	Vrijednost
tsize	3	disaster	500
		qrot.ubound	0.2
		qrot.lbound	0.005
		initAngle	0.01
population.size	100	population.size	10
mutation.indprob	0.3	mutation.indprob	0.04
crx.onepoint	1	mut.quantum_swap	1
mut.simple	1	mut.quantum_inversion	1

Pri evaluaciji rješenja može doći do pretrpavanja naprtnjače. Ne prihvaćamo takva rješenja te ih želimo potisnuti. To obavlja operator kazne ugrađen u evaluator koji ovisno o veličini prekoračenja umanjuje ostvareni profit za količinu prekoračene težine pomnoženu sa zadanom konstantom. Koristi se pretpostavljena konstanta 5.

Tablica 4.3: Prosječni rezultati po skupovima

Skup	CGA		
	Pronađena opt. vrijednost	Broj generacija	Broj evaluacija
500	2922	319	63920
1000	3616	700	140115

Skup	GAIQM		
	Pronađena opt. vrijednost	Broj generacija	Broj evaluacija
500	2922	3261	32635
1000	3616	4616	46182

Oba algoritma za oba seta pronalaze optimalno rješenje. Ovdje je posebno naglašena razlika u broju poziva evaluatora. Vidimo da je za set od 1000 predmeta razlika između broja poziva 93933 odnosno GAIQM preko 3 puta manje poziva evaluator.

4.1.3. Utjecaj parametara

GAIQM ima više parametara u odnosu na CGA te zahtjeva više konfiguracije. Iako sama implementacija ima pretpostavljenu većinu parametara, ovisno o problemu potrebno je uštimati pojedine parametre kako bi se ostvarili bolji rezultati. Navedeni primjeri i mjerenja definirani su za prethodno izmjereni skup od 500 predmeta.

Veličina populacije

Većom populacijom ostvaruje se veća raznolikost genetskog materijala što pospješuje pretraživanje neistraženog prostora stanja što znači da će algoritam u manje generacija pronaći optimalno rješenje. Što je veća populacija, to će se po generaciji više puta pozivati evaluator koji je često vrlo zahtjevan i spor. Dakle, želimo pronaći optimalnu veličinu populacije za koju će se minimizirati broj poziva evaluatora.

Tablica 4.4: Prosječan broj evaluacija i generacija ovisno o veličini populacije

Veličina populacije	Broj evaluacija	Broj generacija
3	35496.3	11830.1
10	33542	3352.2
30	35127	1168,9
50	45200	902
100	73990	737.9

Za sve navedene veličine populacija algoritam pronalazi optimalno rješenje. Vidimo da broj evaluacija ovisi o veličini populacije te da za ovaj konkretan problem ima minimum za populaciju 10 jedinki. S druge strane, broj generacija opada porastom veličine populacije što potvrđuje tvrdnju o značaju raznolikosti populacije.

Početni kut

Početnim kutom namještamo stanje u kojem će se pronaći svi qubiti pri inicijalizaciji. Standardno se postavlja na $\frac{\pi}{2}$ no može se mijenjati ovisno o problemu. Za problem naprtnjače ovisno o karakteristikama skupa koji koristimo možemo pretpostaviti da će većina ili manjina predmeta stati u naprtnjaču. Tada početni kut postavljamo na veći ili manji kut respektivno.

Tablica 4.5: Prosječan broj evaluacija i generacija ovisno o početnom kutu

Veličina populacije	Broj evaluacija	Broj generacija
0.005	51443	5142.3
0.01	33542	3352.2
0.05	91165	9114.5
0.1	78141	7812.1
0.5	200000	19998
0.7	200000	19998

Vidimo da naša pretpostavka o svojstvima skupa može uvelike utjecati na brzinu, ali i uspješnost rada algoritma. Pri odabiru ponovno postoje optimalni parametri za koje algoritam postiže najveću brzinu. Na ovom primjeru algoritam za početne kuteve 0.5π i 0.7π ne pronalazi rješenje već se prekida brojem evaluacija dok za kut 0.1π gotovo uvijek pronalazi rješenje.

Granice kuteva adaptivne metode zakreta

Granice kuteva su vrlo osjetljivi parametri izrazito ovisni o problemu. Adaptivnom metodom odabir kuta zakreta bira se kao interpolacija vrijednosti unutar granica proporcionalno odnosu vrijednosti dobrote jedinke s najboljom jedinkom. Donja granica određuje preciznost kojom se pretražuju stanja oko najbolje jedinke trenutne populacije i obično je reda veličine 10^{-2} ili manje. Gornja granica određuje početnu brzinu konvergencije kada jedinka ima najlošiju vrijednost dobrote u odnosu na najbolju vrijednost trenutne populacije i obično je reda veličine 10^{-1} ili manje.

Kada algoritam pronađe lokalni optimum sve jedinke konvergiraju prema njoj. Dakle, premala donja granica može zaglaviti algoritam u lokalnom optimumu te se odabire vrijednost veća od 0. Prevelika gornja granica može uzrokovati prebrzu početnu konvergenciju populacije prema pronađenom lokalnom optimumu u kojem algoritam može zaglaviti. Raspon granica kuteva određuje brzinu konvergencije populacije te se često odabiru granice različite za 1 red veličine.

Operator katastrofe

Operator katastrofe poziva se samo kada se ista vrijednost dobrote pojavi u n uzastopnih generacija. Bez njega algoritam može zapeti u lokalnom optimumu. Pozivamo li operator prečesto možemo usporiti ili čak zaustaviti konvergenciju populacije te algoritam neće moći pronaći optimum u razumnom vremenu. Operator treba pozivati "dovoljno" rijetko, a taj iznos ovisi o problemu i najbolje ga je odabrati prateći ponašanje rezultata tijekom rada algoritma. Uobičajene vrijednosti su reda 100 generacija.

4.2. Traženje globalnog minimuma funkcije (FunctionMin)

4.2.1. Opis problema

Zadane su netrivijalne višedimenzionalne funkcije kojima je potrebno pronaći minimum. Ovaj eksperiment koristi genotip $FloatingPoint$ za klasični algoritam te $QuantumRegister$ kao radni genotip.

Zadane funkcije:

1. Kvadratna
$$f(\mathbf{x}) = \sum_{i=1}^{n}(x_i - i)^2 \tag{4.1}$$

2. Schaffer-ova F6 funkcija
$$f(x, y) = 0.5 + \frac{sin^2(\sqrt{x^2 + y^2}) - 0.5}{[1 + 0.001 \cdot (x^2 + y^2)]^2} \tag{4.2}$$

3. Griewangk
$$f(x_1 \cdots x_n) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) \tag{4.3}$$

4. Ackley
$$f(x_0 \cdots x_n) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e \tag{4.4}$$

5. Rastrigin
$$f(x_1 \cdots x_n) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i)) \tag{4.5}$$

6. Rosenbrock
$$f(x_1 \cdots x_n) = \sum_{i=1}^{n-1}(100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2) \tag{4.6}$$

7. Schaffer-ova F7 funkcija
$$f(x_0 \cdots x_n) = [\frac{1}{n-1}\sqrt{s_i} \cdot (sin(50.0 s_i^{\frac{1}{5}}) + 1)]^2 s_i = \sqrt{x_i^2 + x_{i+1}^2} \tag{4.7}$$

4.2.2. Rezultati

Funkcije ograničavamo na 20 dimenzija/vrijednosti te prostor pretraživanja definiramo na intervalu $[-10, 10]$ preciznošću do 16. decimale.

Tablica 4.6: Korišteni parametri za FunctionMin

SteadyStateTournament		GAIQM	
Parametar	Vrijednost	Parametar	Vrijednost
tsize	3	disaster	300
		qrot.ubound	0.25
		qrot.lbound	0.008
		initAngle	0.5
dimension	20	dimension	20
ubound	10	ubound	10
lbound	-10	lbound	-10
		precision	16
population.size	100	population.size	10
mutation.indprob	0.3	mutation.indprob	0.04
		mut.quantum_swap	0.2
		mut.quantum_inversion	0.8

Za svaki eksperiment postavljena je granica dozvoljenog broja evaluacija na 200000 te vrijednost globalnog optimuma koju moraju postići na 10^{-9}. Algoritmi nikada ne dostižu vrijednosti globalnog optimuma pa je broj evaluacija jedini prekidač programa.

Tablica 4.7: Prosječne vrijednosti po funkcijama

Funkcija	CGA		GAIQM	
	Optimum	Broj generacija	Optimum	Broj generacija
F1	385	1999	387.65	19998
F2	$3.93 \cdot 10^{-2}$	1999	$8.22 \cdot 10^{-2}$	19998
F3	$2.1 \cdot 10^{-3}$	1999	$2.17 \cdot 10^{-2}$	19616.35
F4	$7.37 \cdot 10^{-4}$	1999	$4.33 \cdot 10^{-5}$	16832.85
F5	$5.42 \cdot 10^{-4}$	1999	25.72	19998
F6	26.36	1999	76	19998
F7	1.92	1999	2.1	19998

Za isti broj evaluacija GAIQM prolazi više generacija od CGA. Unatoč tome GAIQM ne pronalazi uvijek bolja rješenja. Dapače, ponekad pronalazi i po nekoliko redova veličine lošija rješenja.

4.3. Regresija neuronske mreže

4.3.1. Opis problema

Za zadane ulaz/izlaz brojčane podatke potrebno je pronaći težine veza između slojeva neuronske mreže (učenje neuronske mreže). Dobrota mreže računa se srednjom kvadratnom greškom (MSE) koju algoritam pokušava minimizirati. Ovaj eksperiment koristi genotip $FloatingPoint$.

Za testiranje koristim dvije funkcije:

$$f(x) = \frac{x^2}{100}, \quad x \in [-9, 9], \quad \Delta x = 1$$
$$g(x) = \sin x, \quad x \in [0, 6.2], \quad \Delta x = 0.1$$

4.3.2. Rezultati

Struktura mreže je fiksirana na $1 \to 5 \to 1$, a aktivacijske funkcije su definirane po slojevima $sigmoidalna \to linearna$. Broj evaluacija ograničen je na 200000.

Tablica 4.8: Korišteni parametri za regresiju neuronske mreže

SteadyStateTournament		GAIQM	
Parametar	Vrijednost	Parametar	Vrijednost
tsize	3	disaster	600
		qrot.ubound	0.1
		qrot.lbound	0.01
		initAngle	0.5
population.size	70	population.size	10
mutation.indprob	0.3	mutation.indprob	0.01
		mut.quantum_swap	1
		mut.quantum_inversion	1

Težine između slojeva ograničavamo na interval $[-10, 10]$ s preciznošću do 16. decimale.

Tablica 4.9: Prosječni rezultati po funkcijama

Funkcija	CGA		GAIQM	
	Optimalna greška	Broj generacija	Optimalna greška	Broj generacija
$f(x)$	$8.14 \cdot 10^{-2}$	2857	$2.07 \cdot 10^{-1}$	39998
$g(x)$	26.14	2857	26.36	39998

GAIQM postiže znatno lošije rezultate u odnosu na CGA. Iako prolazi kroz deseterostruko više generacija ne uspijeva pronaći bolji optimum. Tijekom rada algoritam ne uspijeva izaći iz optimuma te se operator katastrofe redovito poziva pokušavajući pronaći "bolji kut" iz kojeg će pretraživati prostor prema optimumu.

5. Moguće nadogradnje

5.1. Paralelizacija

Stvaranjem više nezavisnih populacija jedinki te povremenom razmjenom najboljih jedinki populacija poboljšava se raznolikost populacije. Također, ostvaruje se šira pretraga prostora stanja jer se jedinke pojedinih populacija počinju "kretati" prema drugim optimumima čime se povećava vjerojatnost pronalaženja dotad neistraženih stanja. [2]

5.2. Podržavanje više-genotipskih jedinki

Trenutno je ostvarena podrška jedinki s jednim genotipom. Definiranjem više genotipa, vrijednosti iz kvantnog registra će se kopirati u prvi zadani genotip (genotip jedinke na indeksu 0). Podršku više genotipa moguće je ostvariti izmjenom 'prilagodnika' te prikladnim zastavicama za odabir genotipova u koje će se kopirati odgovarajuće vrijednosti iz kvantnog registra.

5.3. Ubrzanje algoritma domenom cijelih brojeva

Operacije s podacima dvostruke preciznosti su računski zahtjevnije cjelobrojnih te ih je poželjno zamijeniti. Trenutno se u kvantni registar pohranjuju brojevi dvostruke preciznosti te su sve operacije prilagođene njihovom. Kako je kvantni i klasični dio GAIQM međusobno izoliran, a povezan samo operatorom mjerenja, moguće je drugačije modelirati kvantni dio bez utjecaja na klasični.

Umjesto decimalnih brojeva, u kvantni registar možemo pohranjivati cijele brojeve. Trenutnu definiciju kuta iz domene $[-\pi, \pi]$ možemo preslikati na proizvoljnu cjelobrojnu domenu $[MIN, MAX]$. Tada se sve operacije nad kvantnim registrom trebaju prilagoditi novoj domeni.

Nedostatak ove optimizacije je smanjivanje domene kuteva odnosno kvantnih stanja što može utjecati na preciznost lokalne pretrage prostora stanja kao i uzrokovati prebrzu konvergenciju stanja prema lokalnom optimumu.

6. Zaključak

Genetski algoritam inspiriran kvantnom mehanikom kombinacija je strukture klasičnih genetskih algoritama i paradigme kvantnih računala. Cilj ovog rada je objasniti paradigmu kvantnih računala, istražiti mogućnosti oblikovanja stohastičkih optimizacijskih algoritama uz korištenje simulacije rada kvantnog računala te usporediti novu vrstu genetskog algoritma s klasičnim na nekoliko primjera.

Kako su evaluatori često skupe (spore) procedure, a brzina algoritma ovisi direktno o broju evaluacija poželjno je što manje pozivati evaluator. Rezultati usporedbe klasičnog genetskog algoritma (CGA) i genetskog algoritma inspiriranog kvantnom mehanikom (GAIQM) pokazuju na nadmoć GAIQM po broju poziva evaluatora. To navodi na činjenicu da je GAIQM iznimno pogodan za rad sa zahtjevnim evaluatorima.

Vremenska cijena GAIQM proizlazi iz činjenice da obavlja računske operacije koje se ponavljaju nad svakim qubitom svake jedinke u populaciji. Svoju vremensku zahtjevnost balansira znatno manjom populacijom od CGA. Veličina populacije osigurava veću raznolikost populacije odnosno pretraživanje većeg prostora stanja po iteraciji te se na nju oslanja CGA. GAIQM s druge strane ne zahtijeva veliku populaciju zbog svoje ugrađene nasumičnosti svake jedinke kojom jednako dobro pretražuje prostor stanja.

Rezultati eksperimenata ukazuju na nadmoć GAIQM u kombinatoričkom tipu problema, konkretno na problemu naprtnjače. Na ostalim izvedenim eksperimentima GAIQM daje loše rezultate i ne uspijeva se mjeriti s CGA.

LITERATURA

[1] Sourabh Singh Chauhan. A generalized quantum-inspired evolutionary algorithm for combinatorial optimization problems. 2014. URL http://homepages.spa.umn.edu/~sourabh/quantum_annealing.pdf.

[2] Kuk-Hyun Han. *Quantum-inspired Evolutionary Algorithm*. Doktorska disertacija, Korea, Advanced Institute of Science and Technology, Department of Electrical Engineering and Computer Science, 6 2003. Advisor : Professor Jong-Hwan Kim.

[3] Kuk-Hyun Han i Jong-Hwan Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. U *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, svezak 2, stranice 1354–1360 vol.2, 2000. doi: 10.1109/CEC.2000.870809.

[4] Kuk-Hyun Han i Jong-Hwan Kim. On the analysis of the quantum-inspired evolutionary algorithm with a single individual. U *2006 IEEE International Conference on Evolutionary Computation*, stranice 2622–2629, 2006. doi: 10.1109/CEC.2006.1688636.

[5] M. Hirvensalo. *Quantum Computing*. Natural Computing Series. Springer Berlin Heidelberg, 2013. ISBN 9783662096369. URL https://books.google.hr/books?id=lAmrCAAAQBAJ.

[6] Rafael Lahoz-Beltra. Quantum genetic algorithms for computer scientists. *Computers*, 5(4), 2016. ISSN 2073-431X. URL http://www.mdpi.com/2073-431X/5/4/24.

[7] L. Lederman i D. Teresi. *The God Particle: If the Universe Is the Answer, What Is the Question?* Houghton Mifflin Harcourt, 2006. ISBN 9780547524627. URL https://books.google.hr/books?id=jMOOQDHxWyIC.

[8] M. Mitchell. *An Introduction to Genetic Algorithms*. A Bradford book. Bradford Books, 1998. ISBN 9780262631853. URL https://books.google.hr/books?id=0eznlz0TF-IC.

[9] Maris Ozols. Geometry of a qubit. 2007.

[10] Zhi Jun Wang Huaixiao, Liu Jianyong i Fu Chengqun. The improvement of quantum genetic algorithm and its application on function optimization. *Mathematical Problems in Engineering*, 2013. URL http://dx.doi.org/10.1155/2013/730749.

Dodatak A
Parametri

U nastavku je dan opis svih parametara. Obavezni parametri označeni su **podebljanim tekstom**.

Tablica A.1: Parametri za <GAIQM>

Naziv	Tip	Pretp. vr.	Opis
$ubound$	$double$	0.1	Gornja granica zakreta qubita
$lbound$	$double$	0.01	Donja granica zakreta qubita
$disaster$	$uint$	-1	Broj ponovljenih vrijednosti operatora katastrofe (vrijednost -1 gasi operator katastrofe)

Tablica A.2: Parametri za <KnapsackEvalOp>

Parametri $dimension$, $ubound$, $lbound$ i $precision$ naslijeđeni su iz genotipa Binary.

Naziv	Tip	Pretp. vr.	Opis
eval.knapsackSize	$double$	-	Veličina (nosivost) torbe
eval.itemsFile	$string$	-	Putanja do datoteke s definicijama težina i cijene predmeta
$eval.$ $punishmentFactor$	$double$	5	Faktor kojim se množi prekoračenje nosivosti torbe

Tablica A.3: Parametri za <QuantumRegister>

Parametri $dimension$, $ubound$, $lbound$ i $precision$ naslijeđeni su iz genotipa Binary.

Naziv	Tip	Pretp. vr.	Opis
dimension	$uint$	-	Količina realnih brojeva genotipa
ubound	$double$	-	Maksimalna vrijednost brojeva u genotipu
lbound	$double$	-	Minimalna vrijednost brojeva u genotipu
precision	$uint$	-	Broj znamenki nakon decimalne točke
$initAngle$	$double$	0.5	Početni kut qubita (početno kvantno stanje), množi se s π
$mut.quantum\_inversion$	$double$	0.5	Vjerojatnost poziva operatora
$mut.quantum\_swap$	$double$	0.5	Vjerojatnost poziva operatora

Genetski algoritam inspiriran kvantnom mehanikom

Sažetak

Cilj rada je objasniti koncept pohrane i manipulacije podacima na kvantnim računalima, primjenu tih koncepata na genetski algoritam koji se izvršava na klasičnom računalu te proširenje ECF-a dotičnim algoritmom. Algoritam je primijenjen na 3 različita problema (traženje minimuma, knapsack, regresija neuronske mreže). Rezultati ukazuju na primjenjivost algoritma s vrlo zahtjevnim evaluatorima.

Ključne riječi: genetski algoritam, kvantna mehanika, qubit, knapsack, neuronska mreža.

Genetic algorithm inspired by quantum mechanics

Abstract

The goal of this paper is to explain the concept of data storage and manipulation on quantum computers, applying those concepts on a genetic algorithm that executes in a classical computer and extending the ECF with it. The algorithm was applied on 3 different problems (function minimum search, knapsack problem, regression of a neural network). The results indicate usefulness of the algorithm on very demanding evaluators.

Keywords: genetic algorithm, quantum mechanics, qubit, knapsack, neural network.