

Förbättring av en kvanthybridalgoritm applicerad på proteinveckningsproblemet

Införande av CVaR i optimeringsslingan i QAOA för implementering på HP-modellen

Kandidatarbete i Kemiteknik med fysik [1] & Teknisk fysik [2]

Hampus Brunander [1], Petter Dittmer [2], Jesper Lagerberg [1], Jacob Ljungberg [2], Nils Nerpin [2] & Ludvig Rodung [2]

Institutionen för mikroteknologi och nanovetenskap

CHALMERS TEKNISKA HÖGSKOLA Göteborg, Sverige 2023 www.chalmers.se

Kandidatarbete 2023

Förbättring av en kvanthybridalgoritm applicerad på proteinveckningsproblemet

Införande av CVaR i optimeringsslingan i QAOA för implementering på HP-modellen

Improvement of a Hybrid Quantum Algorithm Applied on the Protein Folding Problem

Introduction of CVaR in the Optimization Loop of QAOA for Implementation on the HP-model

Hampus Brunander Petter Dittmer Jesper Lagerberg Jacob Ljungberg Nils Nerpin Ludvig Rodung



Institutionen för mikroteknologi och nanovetenskap WACQT CHALMERS TEKNISKA HÖGSKOLA Göteborg, Sverige 2023 Förbättring av en kvanthybridalgoritm applicerad på proteinveckningsproblemet Införande av CVaR i optimeringsslingan i QAOA för implementering på HP-modellen Hampus Brunander, Petter Dittmer, Jesper Lagerberg, Jacob Ljungberg, Nils Nerpin & Ludvig Rodung

© Hampus Brunander, Petter Dittmer, Jesper Lagerberg, Jacob Ljungberg, Nils Nerpin & Ludvig Rodung, 2023.

Handledare: Laura García-Álvarez & Göran Johansson, institutionen för mikroteknologi och nanovetenskap Examinator: Per Lundgren, institutionen för mikroteknologi och nanovetenskap

Kandidatarbete 2023 Institutionen för mikroteknologi och nanovetenskap WACQT Chalmers tekniska högskola SE-412 96 Göteborg Telefon +46 31 772 1000

Omslag: Visualisering av CVaR på en normalfördelad sannolikhetsfördelning

Typsatt i LATEX Göteborg, Sverige 2023

Sammandrag

I det här projektet implementeras och undersöks en möjlig förbättring av kvanthybridalgoritmen kvantmekanisk approximativ optimeringsalgoritm (QAOA, Quantum Approximate Optimization Algorithm) på proteinveckningsproblemet. Den undersökta förbättringen kallas Conditional Value at Risk (CVaR), och jämförs med konventionell QAOA för att se om den har några fördelar. Testerna av algoritmen utförs genom simuleringar med programbiblioteket PennyLane för Python. För att modellera proteinveckningsproblemet användes HP-modellen. I slutändan visade QAOA med CVaR på att möjligen vara en liten förbättring jämfört med konventionell QAOA, men fler tester bör göras för att säkerställa om denna förbättring ökar eller minskar med djupet på kvantkretsen och längden på aminosyrakedjan.

Nyckelord: Kvantdator, Kvantalgoritm, QAOA, CVaR, Proteinveckning, HP-modellen, Gittermodellen

Abstract

In this report, the implementation of Conditional Value at Risk (CVaR) on Quantum Approximate Optimization Algorithm (QAOA) is explored on the protein folding problem. The measurements are compared with regular QAOA, to find possible improvements. The tests and simulations are made with the help of the Python library PennyLane, with the HP-model being used to simulate the folding of an amino acid chain. In the end, QAOA with CVaR showed promise to be a small improvement compared to regular QAOA. QAOA with CVaR will have to be implemented on bigger problems to find out whether its improvement over regular QAOA is increasing, or decreasing, with the depth of the quantum circuit and length of the amino acid chain.

Keywords: Quantum Computing, Quantum Algorithm, QAOA, CVaR, Protein Folding, HP-model, Lattice model

Förord

Vi vill tacka våra hårt arbetande handledare Laura García-Álvarez och Göran Johansson, samt ett extra tack till doktorand Hanna Linn för hjälp och guidning. Projektet har gett samtliga studenter en mycket bättre förståelse för kvantberäkning som vi kommer att gynnas av i framtiden.

Hampus Brunander, Petter Dittmer, Jesper Lagerberg, Jacob Ljungberg, Nils Nerpin och Ludvig Rodung Göteborg, maj 2023

Akronym

Nedan är en lista över akronymer som används i arbetet:

CVaR	Conditional Value at Risk CVaR är en metod där endast en del av sannolikhetsfördelningen används för att ta fram ett approximativt medelvärde av hela sannolikhetsfördelningen [1].
NISQ	Noisy Intermediate Scale Quantum En kvantdator bestående av 50 till några hundra stycken kvantbitar vars använ- dande påverkas så pass mycket av störningar att det måste tas hänsyn till vid beräkningar [2], [3].
VQA	Variational Quantum Algorithm Kvantdatoralgoritmer där en parametriserad kvantkrets konstrueras utifrån en kostnadsfunktion. Kostnadsfunktionens grundtillstånd beskriver lösningen till det sökta problemet. Algoritmerna använder både en kvantdator som evalu- erar kostnadsfunktionen och en klassisk dator som optimerar parametrarna i kvantdatorn [4].
QAOA	Quantum Approximate Optimization Algorithm En VQA som har två vektorparametrar som optimeras, $\vec{\gamma}$ och $\vec{\beta}$, vars längd bestäms av kontrolldjupet. Sannolikheten att hitta en korrekt lösning på det sökta problemet ökar med högre kontrolldjup [5].

Nomenklatur

Nedan är en lista över nomenklatur, index, parametrar och variabler som används i arbetet:

Index

k Index på kretslager k

Parametrar

γ_k	Inparameter till kostnadslagret
eta_k	Inparameter till mixerlagret
λ_1	Viktfaktor till energin E_1
λ_2	Viktfaktor till energin E_2
λ_3	Viktfaktor till energin E_3

Variabler

р	Kontrolldjup på QAOA-kretsen
E_{HH}	Antal hydrofoba (H) aminosyror som är närmsta grannar
E_1	Energin för begränsning 1
E_2	Energin för begränsning 2
E_3	Energin för begränsning 3
n	Antal mätningar/sampel i en kvantdatorberäkning/-simulering
q	Antal kvantbitar
$P_{GS}(\vec{\gamma}, \vec{\beta})$	Sannolikheten att mäta grundtillståndet, en konfiguration vars energi är den lägsta möjliga, för ett visst ($\vec{\gamma}, \vec{\beta}$).

Innehåll

Akronym v				
No	omenl	klatur	v	
1	Inle	dning	1	
	1.1	Bakgrund	1	
	1.2	Syfte	3	
	1.3	Avgränsningar	4	
2	Теог	i	4	
	2.1	Kvantmekanisk bakgrund	4	
		2.1.1 Hamiltonianer och vågfunktioner	4	
		2.1.2 Kvantbitar	5	
		2.1.3 Blochsfären	5	
		2.1.4 Kvantgrindar och kvantkretsar	6	
	2.2	Kvantberäkningar och kvantalgoritmer	8	
		2.2.1 Adiabatisk kvantberäkning	8	
		2.2.2 Quantum Approximate Optimization Algorithm	9	
		2.2.3 Conditional Value at Risk	0	
	2.3	HP-model	1	
		2.3.1 Isingmodellen	2	
		2.3.2 Proteinveckning	2	
		2.3.3 Grundtillstånd	3	
•				
3			.4	
	3.1	Programmets utformning	4	
	3.2		6	
	3.3	Analys av variansen for CVaR beroende pa antal sampel	6	
	3.4	Jamforelse av resultat	1	
4	Resu	ultat av testning 1	7	
	4.1	Utvärdering av initialparametrar	7	
	4.2	HPPH-aminosyrasekvensen på 2×2-gittret1	8	
	4.3	HPPH-aminosyrasekvensen på 3×2-gittret1	9	
	4.4	HHHPPH-aminosyrasekvensen på 3×2-gittret2	0	
	4.5	Standardavvikelsen av CVaR beroende på antal sampel	1	
5	Disk	cussion 2	2	
	5.1	Generell resultatdiskussion	2	
	5.2	Initialparametrar	2	
	5.3	Prestation för de olika instanserna 2	2	
		5.3.1 HPPH-aminosyrasekvensen på 2×2-gittret	3	
		5.3.2 HPPH-aminosyrasekvensen på 3×2-gittret,	3	
		5.3.3 HHHPPH-aminosyrasekvesen på 3×2-gittret	3	
	5.4	Konsekvenser av CVaR-avvikelse beroende på antal sampel	4	
	5.5	Förslag inför vidare studier	4	

6	Etiska aspekter	24
7	Slutsats	25
Källförteckning		26
A	Transformering av Hamiltionian till Ising-modellen	Ι

1 Inledning

I följande avsnitt beskrivs bakgrunden till, syftet med och avgränsningar i projektet. En inblick ges till varför det är intressant att utveckla kvantdatorer och kvantalgoritmer. Bakgrunden till proteinveckningsproblemet introduceras, och metoder för att lösa kombinatoriska optimeringsproblem som detta med hjälp av kvantteknologi diskuteras kortfattat. Slutligen presenteras även syftet med projektet samt de avgränsningar som gjorts.

1.1 Bakgrund

Forskning och utveckling inom kvantteknologi har under de senaste åren fått mer och mer uppmärksamhet, och stora ekonomiska satsningar har gjorts inom fältet [6]. Utvecklingsarbete sker både på kommersiella bolag samt på forskningsinstitut och universitet runt om i världen. En av de största utmaningarna inom fältet är drömmen om att utveckla en välfungerande kvantdator. Två aktörer som länge legat i framkant när det kommer till utvecklandet av kvantdatorer är de stora teknikbolagen Google och IBM, men även stora nationella satsningar görs runt om i världen. Exempelvis har Kina totalt tillkännagivit 15 miljarder dollar i offentlig finansiering för utvecklandet av kvantteknologi [6]. I Sverige leds ett nationellt forskningsprogram koordinerat från Chalmers under namnet Wallenberg Centre for Quantum Technology (WACQT). Forskningsprogrammet är en tolvårig miljardsatsning uppstartad 2018 vars huvudprojekt är att: "Utveckla och bygga en kvantdator som kan lösa problem långt utom räckhåll för dagens bästa superdatorer" [7].

Efter en period i början av 1900-talet då vetenskapsmän upptäckt flera brister med dåtidens fysikaliska teorier utvecklades det vi i dag känner till som kvantmekanik [8]. Kvantmekaniken, med idéer om exempelvis superposition och kvantsammanflätning, var på många sätt revolutionerande och betecknade genom sin framväxt ett skifte i vår förståelse av hur universums minsta beståndsdelar beter sig. Ett annat vetenskapsfält som under 1900-talet undergick mycket stora framsteg är datavetenskapen. I takt med att teknologiska framsteg möjliggjort konstruerande av mer och mer avancerade datorer har datavetenskapen parallellt utvecklats till det blomstrande och mångfasetterad fältet vi har idag. I kölvattnet av dessa två epokgörande vetenskapsfält uppstod idén om en kvantdator.

I en banbrytande artikel om kvantdatorer och kvantberäkning från år 1982, resonerade den amerikanske fysikern Richard Feynman kring svårigheterna med att simulera kvantmekaniska system med hjälp av klassiska datorer [8], [9]. Han föreslog då idén om att en dator som istället bygger på kvantmekaniska principer skulle kunna simulera dessa system mycket effektivare. I en klassisk dator kallas de fundamentala beståndsdelarna som används i operationer för bitar. En bit kan vara i två tillstånd, antingen 0 eller 1. Den grundläggande principen som skiljer en kvantdator från en klassisk dator är att dessa fundamentala beståndsdelarna byts mot så kallade kvantbitar. En kvantbit kan precis som en klassisk bit vara i ett av tillstånden 0 eller 1, men den kan även vara i ett superpositionstillstånd av både 0 och 1. Genom att utnyttja det kvantmekaniska fenomenet sammanflätning kan en kvantdator innehållandes q stycken kvantbitar representera 2^q olika tillstånd samtidigt. Detta kan i teorin resultera i en exponentiell förbättring gentemot en klassisk dator vars förmåga att representera tillstånd växer proportionellt mot antalet bitar i datorn. På grund av detta finns förhoppningen att en kvantdator inte bara ska vara användbar för att simulera kvantmekaniska system, utan även för att lösa bland annat klassiska optimeringsproblem [3].

Att realisera en kvantdator i praktiken har visat sig vara en svår utmaning. Det handlar inte bara om att hitta ett fysikaliskt system som kan beskrivas som en superposition av två tillstånd, utan det måste också vara möjligt att styra kvantbitarnas tidsutveckling och utföra mätningar på dem [8]. En kvantdator måste till viss mån vara ett öppet system, det vill säga att det interagerar med sin omgivning. På grund av det förlorar den under tiden för en beräkning successivt de kvantmekaniska egenskaper som gör kvantberäkningen meningsfull. I nuläget finns flera olika idéer om hur en kvantdator kan realiseras med hänsyn till dessa, och flera andra, utmaningar. En av de populäraste varianterna är att använda jonfällor. I dessa kvantdatorer representeras kvantbitarna av tillstånden hos fångade joner. En annan variant som mycket forskningsresurser har investerats i är att bygga supraledande kvantdatorer. Dessa kvantdatorer går istället ut på att skapa artificiella atomer genom supraledande kretsar. Supraledande kvantdatorer är också den typ av kvantdator som Chalmers utvecklar inom WACQT [10]. I januari 2023 hade denna ett chip bestående av 25 kvantbitar. Många av de kvantdatorer som utvecklas idag och som man hoppas kan vara användbara för praktiska tillämpningar under de kommande åren, benämns NISQ-datorer (Noisy Intermediate-Scale Quantum). Dessa består av runt 50 till ett par hundra kvantbitar och påverkas av störningar i så stor grad att det begränsar storleken på de kvantkretsar och likaledes kvantalgoritmer som kan exekveras.

En av de största svårigheterna med kvantberäkning är att även om en kvantdator innehållandes q stycken kvantbitar kan representera 2^q olika tillstånd samtidigt, dikterar kvantmekanikens lagar att när det väl görs en mätning på systemet så fås endast ett av dessa tillstånd tillbaka [8]. Av denna anledning krävs smarta kvantalgoritmer för att en kvantdator ska kunna utföra beräkningar som är effektivare än de beräkningar som kan utföras på klassiska datorer. Ett exempel på en sådan algoritm är Shors algoritm, demonstrerad av Peter Shor år 1994 [11]. Kvantalgoritmen som på ett effektivt sätt kan primtalsfaktorisera stora tal, var ett genombrott då dessa typer av problem inte tros kunna lösas effektivt med klassiska beräkningsalgoritmer. Shors algoritm innebar även att fler fick upp ögonen för kvantdatorfältet, då kvantalgoritmen i teorin möjliggör knäckandet av den frekvent använda krypteringsalgoritmen RSA [12]. För att i praktiken exekvera algoritmen behövs dock många fler kvantbitar än vad som är tillgängligt i dagens bästa NISQ-kvantdatorer [11].

En stor förhoppning inom forskningsfältet är att det går att hitta algoritmer så att alla problem som klassificeras som NP-mellanliggande går att lösa effektivt med kvantdatorer [13]. Med problemklassen NP-mellanliggande menas de problem där tiden det tar för en klassisk dator att exekvera lösningsalgoritmen ökar exponentiellt med antalet bitar som ges som indata, medan tiden det tar att verifiera svaret ökar som mest polynomiskt. Det får dessutom inte gå att effektivt översätta lösningsalgoritmen för ett problem till ett annat [13]–[15]. Eftersom antalet operationer som en klassisk dator kan utföra per sekund är ändligt och hårdvaruberoende, finns problem som i praktiken blir klassiskt olösliga på grund av den långa beräkningstiden. Vanliga exempel på sådana problem är handelsresandeproblemet och Max-Cut [16]. Bättre klassiska algoritmer kan ofta korta ner processen, men i många fall räcker inte det. Kvantdatorer har således potential att överträffa klassiska datorers beräkningsförmåga för en hel mängd olika problem [17].

En vanligt förekommande beräkningsmetod för dagens NISQ-datorer är varierande kvantalgoritmer (VQA, Variational Quantum Algorithms). VQA bygger på att en parametriserad kvantkrets konstrueras utifrån en energi-/kostnadsfunktion, vars grundtillstånd innehåller lösningen till det sökta problemet. VQA exekveras på en så kallad kvanthybriddator som består av både en kvantdator och en klassisk dator. Den parametriserade kvantkretsen exekveras i kvantdatorn medan den klassiska datorn utför en optimering av parametrarna i kvantkretsen [4]. Kvanthybridalgoritmen som används i detta projekt kallas kvantmekanisk approximativ optimeringsalgoritm (QAOA, Quantum Approximative Optimization Algorithm). Algoritmen bygger på att väntevärdet av energin för kvantkretsen minimeras [18], [19]. Kvantkretsen som används i QAOA konstrueras utifrån en Hamiltonian som framtagits för att beskriva energin i det problem som ska lösas [20].

Ett intressant exempel på ett problem som i teorin är lösbart med hjälp av kvantdatorer är det så kalla-

de proteinveckningsproblemet, där lösningstiden för klassiska datorer ökar exponentiellt med längden på aminosyrakedjan [1], [14], [21], [22]. Med proteinveckning menas hur ett protein, som består av en aminosyrakedja, naturligt vecklas ihop beroende på i vilken ordning dess aminosyror är arrangerade. Ett proteins egenskaper beror helt på dess geometriska struktur [23]. Protein som exempelvis Titin, vilket återfinns i människokroppen, kan bestå utav 34350 stycken aminosyror och befinna sig i $3^{34349} \approx 4 \cdot 10^{16388}$ olika konfigurationer, under antagandet att inga strukturella restriktioner för aminosyrorna förekommer. På grund av denna stora mängd möjliga konfigurationer kan uppgiften att beräkna ett proteins naturliga veckning snabbt bli väldigt komplicerat och tidskrävande [24], [25]. Problemet är dock av stort intresse då en effektiv beräkningsmetod potentiellt skulle kunna hjälpa forskare att bland annat ta fram nya läkemedel baserade på nya protein [23]. För att reducera problemets beräkningskomplexitet används vanligtvis HP-modellen introducerad av Ken Dill [26]. I modellen ignoreras alla egenskaper hos de enskilda aminosyrorna, förutom dess hydrofoba (H) eller polära (P) egenskap och görs för att få en uppfattning om hur aminosyrasekvensen veckas i vattenlösning/kroppen.På grund av proteinveckningsproblemets natur är det således en god kandidat för att utvärdera kvantdatorers och kvantalgoritmers effektivitet [27].

I HP-modellen placeras proteinets aminosyror i ett gitter. En förutbestämd energifunktion tilldelar varje bitsträng, motsvarande en konfiguration/veckning, med olika energier [26], [28]. Proteinets optimala struktur definieras då som de konfigurationer med lägsta möjliga energi i just denna modell. Att hitta den optimala strukturen med en klassisk dator, där en bit enbart kan anta antingen tillståndet 1 eller 0, kan ta exponentiell lång tid för större aminosyror, eftersom den enbart kan evaluera energin för en konfiguration åt gången. I en kvantdator, där kvantbitarna kan befinna sig i superpositioner av tillstånden 0 och 1, kan alla konfigurationer representeras samtidigt [8]. Förhoppningen är därmed att det ska gå snabbare att beräkna proteinets optimala struktur med en kvantdator jämfört med en klassisk dator.

1.2 Syfte

Projektets huvudsakliga syfte har varit att utvärdera om en möjlig modifikation av kvanthybridalgoritmen QAOA kan förbättra, alltså accelerera, proteinveckningsberäkningar. För att genomföra denna utvärdering har kvanthybridalgoritmen med och utan den aktuella modifikationen simulerats på klassiska datorer. Data från dessa simuleringar har sparats och bearbetats för att slutsatser kring eventuella förbättringar ska kunna dras.

I traditionell QAOA tas ett väntevärde, av hela den energifördelning som fås ur kvanthybriddatorns kvantkrets. Optimeraren i den klassiska delen av kvanthybriddatorn minimerar vidare detta väntevärde genom att variera kvantkretsens parametrar. Den möjliga modifikationen av QAOA som ska utforskas i projektet innebär att väntevärdet ersätts med Conditional Value at Risk (CVaR). För en mer ingående beskrivning av vad denna modifikation faktiskt innebär se kapitel 2.2.2 och 2.2.3.

Då CVaR kan implementeras med olika värden på hyperparametern α , har tre olika implementeringar jämförts mot traditionell QAOA. Simuleringarna som genomförts har inneburit att lösningar till tre stycken enkla instanser av HP-modellen för proteinveckningsproblemet har beräknats. Syftet har med andra ord inte varit att beräkna lösningar till proteinveckningsproblemet genom att använda QAOA med CVaR, utan att istället använda proteinveckningsproblemet för att utvärdera om QAOA med CVaR är överlägsen traditionell QAOA.

1.3 Avgränsningar

Då syftet med projektet kretsar kring att utforska möjliga förbättringar för QAOA, har resultatet av proteinstrukturoptimeringen och dess energi ej jämförts med resultat från klassiska datorer samt experimentellt bestämda resultat, utan enbart med de resultat som erhålles från traditionell QAOA. Traditionell QAOA är ekvivalent med QAOA med CVaR då hyperparametern $\alpha = 1$, vilket gör att samma kod kan användas för alla tester.

Vid simuleringar av den undersökta kvanthybridalgoritmen har olika många kvantbitar kunnat användas. För att koppla projektet till de NISQ-datorer som utvecklas idag, sattes en övre gräns för antalet kvantbitar till 25 stycken, då det motsvarar det antalet kvantbitar som finns i Chalmers kvantdator. På grund av begränsad beräkningskraft i de persondatorer som i projektet använts för att simulera kvanthybridalgoritmen, begränsades antalet kvantbitar ytterligare till maximalt 18 stycken. I praktiken leder denna begränsning till att de största instanser av HP-modellen som beräkningar utförs på, är aminosyrasekvenser av längd sex på ett 3×2-gitter. Se kapitel 2.3 för en mer ingående beskrivning av HP-modellen.

Kontrolldjup p, som beskrivs mer ingående i kapitel 2.2.2, begränsas i detta projekt till p = [1, 2, 3]. Även denna avgränsning har gjorts på grund av att större värden på p, medför allt för långa beräkningstider.

2 Teori

I följande avsnitt introduceras den teoretiska bakgrund som är nödvändig för att förstå projektets utformning. Bakgrund ges för de grundläggande principer som en kvantdator bygger på och specifikt beskrivs hur kvanthybridalgoritmen QAOA fungerar. Den aktuella modifikationen av QAOA som undersöks i detta projekt, CVaR, beskrivs ingående. HP-modellen för proteinveckning beskrivs, och hur QAOA applicerar HP-modellen genom implementering av en Hamiltonian gås igenom. Slutligen demonstreras vad som menas med aminosyrasekvensernas grundtillstånd.

2.1 Kvantmekanisk bakgrund

Kvantdatorer bygger på en del grundläggande kvantmekanisk teori. För att underlätta förståelsen för projektet ges i detta avsnitt kortfattade genomgångar av de mest relevanta teoretiska delarna.

2.1.1 Hamiltonianer och vågfunktioner

Vågfunktionen är en kontinuerlig komplex funktion som beskriver all möjlig information om ett fysikaliskt system [20]. Denna information går även att beskriva som en tillståndsvektor, $|\Psi\rangle$, i systemets motsvarande tillståndsrum. Tillståndsrummet, \mathcal{H} , är ett komplext vektorrum med en inre produkt, kallat Hilbertrum. För två tillstånd $|\Phi\rangle$ och $|\Psi\rangle$ skrivs den inre produkten som $\langle\Phi|\Psi\rangle \in \mathbb{C}$, och benämns sannolikhetsamplitud. Sannolikheten att mäta tillståndet $|\Phi\rangle$ i ett system i tillståndet $|\Psi\rangle$ ges av $|\langle\Phi|\Psi\rangle|^2 \in [0,1]$. Givetvis gäller att $|\langle\Psi|\Psi\rangle|^2 = 1$, och alla tillståndsvektorer måste därmed vara enhetsvektorer [8].

Tidsutvecklingen för ett kvantsystem beskrivs av den tidsberoende Schrödingerekvationen,

$$i\hbar\frac{\partial}{\partial t}|\Psi\rangle = \hat{H}|\Psi\rangle,$$
 (2.1)

där \hbar är Plancks reducerade konstant och \hat{H} är systemets Hamiltonian, vilken är operatorn som beskriver systemets totala energi [20]. Hamiltonianen är en unitär operator, vilket innebär att $\hat{H}^{\dagger}\hat{H} = I$, där \hat{H}^{\dagger} är operatorns adjunkt och I är identitetselementet. Vidare är den hermitesk, vilket innebär att $\hat{H}^{\dagger} = \hat{H}$, och dess egenvärden är därmed reella [29]. Dessa egenvärden och motsvarande egentillstånd representerar de möjliga energinivåerna för systemet [20]. För ett system med diskreta energinivåer $E_0, E_1, ..., E_n$ med energiegentillstånd $|0\rangle, |1\rangle, ..., |n\rangle$ ges Hamiltonianen av

$$\hat{H} = \sum_{j=0}^{n} E_j |j\rangle \langle j|.$$
(2.2)

Väntevärdet för energin, $\langle E \rangle$ i systemet ges i Diracnotation av

$$\langle E \rangle = \langle \Psi | \hat{H} | \Psi \rangle. \tag{2.3}$$

2.1.2 Kvantbitar

Grunden för kvantberäkning är kvantbiten, vilken realiseras genom att konstruera ett tvånivåsystem. Ett tvånivåsystem är ett fysikaliskt system som beskrivs i ett tvådimensionellt Hilbertrum, och kan därför skrivas som en superposition av två tillståndsvektorer som formar en mätbas för Hilbertrummet. Ofta benämns dessa tillståndsvektorer $|0\rangle$ och $|1\rangle$, likt hur bitar i klassiska datorer benämns som 0 eller 1. En kvantbit kan därmed representera en superposition av dessa två tillstånd, medan en klassisk bit enbart kan representera ett tillstånd åt gången. När flera kvantbitar betraktas växer antalet tillstånd som kan representeras exponentiellt; q stycken kvantbitar kan skapa en superposition av 2^{q} tillstånd. Vågfunktionen för kvantbiten kan skrivas på Diracnotation som

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \qquad \alpha, \beta \in \mathbb{C}$$
(2.4)

och om $|0\rangle = \begin{bmatrix} 1\\0 \end{bmatrix}$ samt $|1\rangle = \begin{bmatrix} 0\\1 \end{bmatrix}$ så kan ekvation 2.4 skrivas på vektorform enligt

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$
 (2.5)

När en mätnings utförs på kvantbiten kollapsar tillståndet till antingen $|0\rangle$ eller $|1\rangle$ med sannolikheterna $|\alpha|^2$ respektive $|\beta|^2$ [8].

Då fler än en kvantbit används kan ett tillstånd beskrivas med hjälp av en mängd kvantbitsträngar, där varje mätbart tillstånd har en unik bitsträng [8]. Exempelvis kan tre kvantbitar i tillstånden $|0\rangle$, $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, respektive $|1\rangle$ beskrivas enligt

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|000\rangle + |010\rangle\right),\tag{2.6}$$

där |000> och |010> är två olika bitsträngar.

2.1.3 Blochsfären

Ett kvanttillstånd kan också representeras geometriskt som en vektor utmed *Blochsfären*, se figur 2.1. Blochsfären har radien r = 1, då sannolikheten att mäta någonting alltid är ett. Blochsfären är en modell som används för att lättare visualisera tvånivåsystem, och är därmed ett bra redskap för att beskriva operationer på enskilda kvantbitar [8]. Kvanttillståndet, se ekvation 2.4, kan i det här fallet beskrivas enligt:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$



(2.7) Figur 2.1: Blochsfärrepresentation av en kvantbit

där θ och ϕ representerar sfäriska koordinater för en punkt på enhetsfären, och $\alpha = \cos \frac{\theta}{2}$ och $\beta = e^{i\phi} \sin \frac{\theta}{2}$.

Även andra mätbaser än $|0\rangle$ och $|1\rangle$ förekommer [8]. Om mätningen sker längs med Blochsfärens *x*-axel istället för dess *z*-axel används mätbasen $|+\rangle$ och $|-\rangle$ istället för $|0\rangle$ och $|1\rangle$. Definitionen av $|+\rangle$ och $|-\rangle$ ges i ekvation 2.11 i kapitel 2.1.4.

2.1.4 Kvantgrindar och kvantkretsar

Vid beräkning med kvantdatorer viktas de enskilda kvantbitarna i kvantbitssträngarna med förhoppning att erhålla en nästintill optimal lösning [8]. Viktningen sker med hjälp av kvantgrindar som opererar på kvantbitarna, likt hur en klassisk dator använder klassiska grindar för att modifiera de klassiska bitarna [13]. Ett exempel på en klassisk grind är *NOT*-grinden, som inverterar värdet av en bit och omvandlar $1 \rightarrow 0$ och $0 \rightarrow 1$. Den klassiska datorn orsakar denna invertering genom att ändra på spänningen, då hög spänning motsvarar ett värde på 1 och låg spänning motsvarar ett värde på 0 [30]. En kvantbit kan befinna sig i en superposition av tillstånden $|0\rangle$ och $|1\rangle$, och kan därför inte inverteras på samma sätt. Om supraledande kvantbitar används kan invertering skapas genom att skicka in mikrovågor med specifik frekvens och amplitud på kvantbitarna [31]. Den nya matematiska beskrivning av det kvanttillstånd som beskrivs i ekvation 2.4 efter att en *NOT*-grind har applicerats på tillståndet är

$$|\psi\rangle = \beta |0\rangle + \alpha |1\rangle, \qquad \alpha, \beta, \in \mathbb{C}$$
 (2.8)

där α och β antar samma värden som innan grinden applicerades [8].

NOT-grinden är dock inte den enda logiska grind som används vid konstruktion utav kvantalgoritmer, utan även andra kvantgrindar, som *CNOT*-grinden (*Controlled NOT*-grinden), *Toffoli*-grinden och $\frac{\pi}{8}$ -grinden används. Eftersom alla kvantgrindar agerar på ett kvanttillstånd, vilket kan beskrivas som en vektor, se ekvation 2.5, kan en kvantgrind beskrivas som en matris. Den tidigare nämnda *NOT*-grinden, vilken ofta benämns *X*, är densamma som Pauli-*X*-matrisen. Den beskrivs enligt följande matris

$$X = \begin{bmatrix} 0 & 1\\ 1 & 0 \end{bmatrix}$$
(2.9)

och ger en matematisk beskrivning av det som sker då kvanttillståndet går från ekvation 2.4 till ekvation 2.8.

En annan vanlig kvantgrind är *Hadamard*-grinden, vilken transformerar mätbastillstånden till jämna superpositionstillstånd. På matrisform har *Hadamard*-grinden formen

$$Ha = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1\\ 1 & -1 \end{bmatrix},$$
 (2.10)

där *Ha* betecknar *Hadamard*-grinden. Vanligtvis betecknas *Hadamard*-grinden som *H*, men för att ej beblanda den med en hydrofob aminosyra (H) eller Hamiltonianen (\hat{H}) används beteckningen *Ha* istället [32]. Om *Hadamard*-grinden appliceras på $|0\rangle$ och $|1\rangle$ fås

$$Ha|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} =: |+\rangle, \qquad Ha|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} =: |-\rangle.$$
(2.11)

Det enda som skiljer sig mellan $Ha|0\rangle$ och $Ha|1\rangle$ är ett plus- och minustecken framför $|1\rangle$, vilket motsvarar en relativ fas på 180°.

Det är möjligt för varje kvantgrind att representeras av en matris, men alla matriser kan inte beskriva en kvantgrind [8]. För att en matris ska beskriva en kvantgrind krävs det att den totala sannolikheten

att mäta något är oförändrad efter grinden har applicerats. Om $|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ är tillståndet innan grinden appliceras och $|\psi'\rangle = \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix}$ är tillståndet efter grinden har applicerats, måste $|\alpha|^2 + |\beta|^2 = |\alpha'|^2 + |\beta'|^2 = 1$, då sannolikheten att mäta någonting alltid är 100%. De matriser som uppfyller detta samband beskriver en kvantgrind, och kallas för unitära [8]. Den matematiska definitionen av en unitär matris är densamma som för en unitär operator och gås igenom i kapitel 2.1.1. Matriserna som beskriver kvantgrindar behöver inte enbart ha dimensionen 2×2, utan alla unitära 2^q × 2^q-matriser kan beskriva en grind. Ett exempel på en 4×4-matris som beskriver en kvantgrind är *CNOT*-grinden, som på matrisform beskrivs enligt följande matris

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
(2.12)

och vars utfall skiljer sig något från *NOT*-grindens [8]. En *CNOT*-grind är en tvåkvantbitsgrind, vilket innebär att den agerar på två kvantbitar samtidigt. I *CNOT*-fallet agerar den som en *NOT*-grind på kvantbit nummer två om kvantbit nummer ett är i tillståndet $|1\rangle$, annars låter den kvantbit nummer två förbli oförändrad. Som utfall från en *CNOT*-grind kan därmed fyra resultat fås, $|00\rangle$, $|01\rangle$, $|10\rangle$ eller $|11\rangle$. I de fall där en kvantdator har tre eller fler kvantbitar kan *CNOT*-grinden agera på två godtyckliga kvantbitar. Precis som i fallet med en kvantbit kan vågfunktionen beskrivas som en superposition av dessa fyra nya tillstånd.

En annan grind är *T*-grinden, som ändrar den relativa fasen 45°. Ett annat vanligt förekommande namn för *T*-grinden är $\frac{\pi}{8}$ -grinden [8]

$$T = \begin{bmatrix} 1 & 0\\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}.$$
 (2.13)

Det är möjligt att approximera alla möjliga kvantgrindar som verkar på en eller flera kvantbitar som en oändlig serie av ett antal olika sorters kvantgrindar. Ett sådant set kallas en universell uppsättning kvantgrindar. Det finns flera möjliga uppsättningar som uppfyller denna definition [13]. Till exempel utgör följande kvantgrindar en sådan uppsättning:

- En kvantgrind som skapar superposition, Hadamard-grinden.
- En kvantgrind som genomför fasrotation på en av kvantbitarna, T-grinden.
- En kvantgrind som kan få två kvantbitar att bli sammanflätade, som CNOT-grinden.

Denna universella uppsättning kvantgrindar kan skrivas {Ha, CNOT, T} [8]. För att representera en godtycklig kvantgrind kan det ofta krävas flera kvantgrindar, speciellt om den grind som ska representeras inte finns med i det nativa setet för kvantdatorn som används för att exekvera algoritmen [8].

Två eller fler kvantbitar är sammanflätade om det går att utvinna information om den andra kvantbiten genom att bara mäta en av kvantbitarna. Matematiskt innebär det att superpositionen inte kan beskrivas som en tensorprodukt av varje enskild kvantbit [13]. Ett exempel på en kvantkrets som resulterar i två sammanflätade kvantbitar visas i kvantkretsdiagrammet i figur 2.2. Kretsen verkar från vänster till höger, och kan resultera i något av dessa fyra möjliga utfall: Båda kvantbitarna efter *CNOT*-grinden är 0, $|00\rangle$, första kvantbiten är 0 medan andra är 1, $|01\rangle$, första är 1 medan andra är 0, $|10\rangle$, eller slutligen, båda kvantbitarna är 1, $|11\rangle$. Eftersom det enbart är möjligt att observera ett av de fyra olika

fallen kan de beskrivas som fyra ortogonala vektorer. Om de olika utfallen beskrivs som följande vektorer

$$|00\rangle = \begin{bmatrix} 1\\0\\0\\0 \end{bmatrix}, \ |01\rangle = \begin{bmatrix} 0\\1\\0\\0 \end{bmatrix}, \ |10\rangle = \begin{bmatrix} 0\\0\\1\\0 \end{bmatrix}, \ |11\rangle = \begin{bmatrix} 0\\0\\0\\1 \end{bmatrix}$$

fås att kvantkretsen i figur 2.2 innan mätningen kan

 $|0\rangle - Ha$ $|0\rangle - \checkmark$

Figur 2.2: En kvantkrets där två kvantbitar förbereds i tillståndet $|0\rangle$. Först appliceras en *Hadamard*-grind på den översta kvantbiten, därefter en *CNOT*-grind där den översta kvantbiten är kontrollkvantbiten. Slutligen mäts den nedersta kvantbiten.

$$(CNOT(Ha \otimes I)) |00\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} |00\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$
(2.14)

där \otimes innebär tensormultiplikation och *I* är identitetsmatrisen. Resultatet av ekvation 2.14 är att båda kvantbitarna antingen befinner sig i tillståndet $|0\rangle$ eller $|1\rangle$. Genom att enbart mäta tillståndet på en av kvantbitarna är det därmed möjligt att avgöra vilket tillstånd den andra kvantbiten befinner sig i.

2.2 Kvantberäkningar och kvantalgoritmer

Nedan beskrivs de kvantalgoritmer och kvantberäkningsmetoder som ligger till grund för eller används direkt i detta projekt.

2.2.1 Adiabatisk kvantberäkning

beskrivas enligt följande ekvation

I adiabatisk kvantberäkning förbereds först en enkel Hamiltonian, \hat{H}_I , vars grundtillstånd är enkelt att bestämma och förbereda. Den utvecklas sedan långsamt till en mer komplicerad Hamiltonian, \hat{H}_F , vars grundtillstånd innehåller lösningen till det problem som studeras [33]. Transformationen som Hamiltonianen genomgår kan beskrivas enligt följande ekvation

$$\hat{H}(t) = s(t)\hat{H}_I + (1 - s(t))\hat{H}_F$$
(2.15)

där s(t) har värdemängden [0, 1] samtidigt som $s(t \le 0) = 1$ och $s(t \ge t_{slut}) = 0$ [34]. Givet att transformationen sker tillräckligt långsamt och att grundtillståndet för \hat{H}_I förbereds vid t = 0, kommer kvantbitstringen som mäts vid $t = t_{slut}$ vara grundtillståndet för \hat{H}_F [33], [34].

Förändringen av Hamiltonianen är tillräckligt långsam om den uppfyller följande samband

$$\max_{t \in [0, t_{slut}]} \left| \frac{\langle E_m(t) | \dot{E}_n(t) \rangle}{E_{nm}(t)} \right| \ll 1, \text{ för alla } n \neq m.$$
(2.16)

 $|E_k(t)\rangle$ är här egentillståndet till $\hat{H}(t)$ med energiegenvärde $E_k(t)$ för excitationsnivå k, $|\dot{E}_k(t)\rangle$ är dess tidsderivata, och $E_{kl}(t)$ är skillnaden i energinivå mellan excitationsnivåerna k och l [35]. Adiabatisk kvantberäkning fungerar dock enbart då spektralgapet, skillnaden mellan grundtillståndets energinivå och första excitationsnivån, är nollskilt för $t \in [0, t_{slut}]$, vilket inte alltid gäller för två godtyckliga \hat{H}_I och \hat{H}_F [34]. Om det inte finns ett spektralgap ökar risken att kvantbitsträngen som mäts inte är grundtillståndet för \hat{H}_F , och en annan kvantbitsträng än den eftersökta erhålles [34]. Eftersom det måste finnas ett spektralgap får Hamiltonianerna \hat{H}_I och \hat{H}_F inte kommutera. Om Hamiltonianerna kommuterar finns det antingen inget spektralgap, eller så har både \hat{H}_I och \hat{H}_F samma grundtillstånd och ingen beräkning behövs [36], [37]. Från ekvation 2.1 fås att kvanttillståndet varierar med tiden *t*. Differentialekvationen går att lösa exakt, och om \hbar antas vara en del av Hamiltonianen fås lösningen på Diracnotation enligt

$$|\Psi(t)\rangle = e^{-i \int_0^t \hat{H}(\tau) d\tau} |\Psi(0)\rangle.$$
(2.17)

Om Hamiltonianerna \hat{H}_I och \hat{H}_F i ekvation 2.15 kommuterar kan ekvation 2.17 utvecklas till

$$|\Psi(t)\rangle = e^{-i\int_0^t (s(\tau)\hat{H}_I + (1 - s(\tau))\hat{H}_F)d\tau} |\Psi(0)\rangle.$$
(2.18)

Däremot om Hamiltonianerna \hat{H}_I och \hat{H}_F inte kommuterar, fås att

$$e^{-i\int_{0}^{t}\hat{H}(\tau)d\tau} \neq e^{-i\int_{0}^{t}s(\tau)\hat{H}_{I}d\tau}e^{-i\int_{0}^{t}(1-s(\tau))\hat{H}_{F}d\tau},$$
(2.19)

och vågfunktionen kan inte uttryckas som i ekvation 2.18 [8]. I de fallen där Hamiltonianerna inte kommuterar kan approximationsmetoden Lie-Trotters produktformel användas

$$\lim_{m \to \infty} \left(e^{iAt/m} e^{iBt/m} \right)^m = e^{i(A+B)t}, \tag{2.20}$$

där A och B är hermiteska matriser och t är reell [8], [38]. För små tidsintervaller, Δt , fås

$$\prod_{k=1}^{m} e^{-i s(k\Delta t)\hat{H}_{I}\Delta t} e^{-i(1-s(k\Delta t))\hat{H}_{F}\Delta t} \approx e^{-i \int_{0}^{t} \hat{H}(\tau)d\tau}$$
(2.21)

där approximationens noggrannhet ökar med m [8].

2.2.2 Quantum Approximate Optimization Algorithm

Quantum Approximate Optimization Algorithm (QAOA) är en kvanthybridalgoritm bestående av både kvantmekaniska och klassiska komponenter. Den har som syfte att approximativt lösa kombinatoriska optimeringsproblem på NISQ-datorer. Algoritmen är en VQA, vilket betyder att den byggs upp av en krets vars kvantgrindar styrs av en mängd parametrar, $(\vec{\gamma}, \vec{\beta})$, vilka sedan varieras för att få algoritmen att fungera på önskat vis [19]. Längden på dessa parametervektorer $(\vec{\gamma}, \vec{\beta})$ är alltid likvärdiga och kallas för kretsens kontrolldjup [39]. Kontrolldjupet benämns hädanefter som enbart djupet *p*.

Likt adiabatisk kvantberäkning använder QAOA två Hamiltonianer: mixerhamiltonianen \hat{H}_M , vars grundtillstånden är känt och enkelt att förbereda, samt kostnadshamiltonianen \hat{H}_C , vars grundtillstånd erhåller lösningen till optimeringsproblemet. Nyckelidén är sedan att i ekvation 2.21 ersätta tidsberoendet med parametrar genom $s(k\Delta t)\Delta t \rightarrow \beta_k$ och $(1 - s(k\Delta t))\Delta t \rightarrow \gamma_k$. Därmed ges det slutgiltiga tillståndet av

$$|\vec{\gamma},\vec{\beta}\rangle = \prod_{k=1}^{p} e^{-i\beta_{k}\hat{H}_{M}} e^{-i\gamma_{k}\hat{H}_{C}} |\psi_{i}\rangle = \prod_{k=1}^{p} \hat{U}_{M}(\beta_{k})\hat{U}_{C}(\gamma_{k}) |\psi_{i}\rangle$$
(2.22)

för ett initialtillstånd $|\psi_i\rangle$, där parametrarna $\vec{\beta} = (\beta_1, \beta_2, ..., \beta_p)$ och $\vec{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_p)$ går att optimera [18]. Operatorerna $\hat{U}_M(\beta_k)$ och $\hat{U}_C(\gamma_k)$ är lika med $e^{-i\beta_k \hat{H}_M}$ respektive $e^{-i\gamma_k \hat{H}_C}$. Precis som för adiabatisk kvantberäkning ökar säkerheten på svaret med djupet p [18].

QAOA fungerar sedan genom följande steg [18]:

- 1. Välj antalet sampel, *n*, ett djup, *p*, och 2*p* initiala parametervärden $\vec{\gamma}$, $\vec{\beta} = \vec{\gamma}_0$, $\vec{\beta}_0$.
- 2. Förbered q stycken kvantbitar i mixerhamiltonianens grundtillstånd.

- 3. Konstruera tillståndet $|\vec{\gamma}, \vec{\beta}\rangle$ genom att utföra operationerna $\hat{U}_C(\gamma_k)$ och $\hat{U}_M(\beta_k)$ alternerande för k = 1, 2, ..., p med en kvantdator. Mät sedan varje kvantbit, vilket resulterar i en bitsträng $|z\rangle$.
- 4. Beräkna $C(z) := \langle z | \hat{H}_C | z \rangle$ med en klassisk dator.
- 5. Upprepa steg 2-4 *n* gånger och uppskatta sedan $\langle E(\vec{\gamma}, \vec{\beta}) \rangle$ till $E_{approx} = \frac{1}{n} \sum_{i=1}^{n} C(z_i)$, samt spara bitsträngen $|z_{best}\rangle$ som ger lägst C(z).
- 6. Om $C(z_{best})$ är optimalt eller bra nog ger algoritmen utdata $C(z_{best})$ och $|z_{best}\rangle$. Annars matas E_{approx} in i en klassisk optimerare vilken uppdaterar $\vec{\gamma}, \vec{\beta}$, varpå steg 3-5 upprepas.



Figur 2.3: Ett flödeschema över QAOA. Kvantbitarna $(|0\rangle_1, ..., |0\rangle_q)$ förbereds i ett tillstånd och skickas in i *p* antal kretslager. Dessa lager består av par av kostnadslager och mixerlager som har olika funktioner. Efter dessa grindar läses tillståndet av med en klassisk dator och ett väntevärde, $\langle \hat{H}_C \rangle$, beräknas. Utifrån detta väntevärde optimeras sedan variablerna $\gamma_1...\gamma_p$ samt $\beta_1...\beta_p$ och processen upprepas [19].

I detta arbete används mixerhamiltonianen $\hat{H}_M = -\sum_{i=1}^q X_i$, där X_i är *NOT*-grinden som opererar på kvantbit *i*. Kvantbitarna förbereds därmed i tillståndet $\left(\frac{1}{\sqrt{2}}\right)^q \sum_z |z\rangle$, vilket är ett grundtillstånd till \hat{H}_M , genom att applicera *Hadamard*-grinden på vardera kvantbit i initialtillståndet $|00...0\rangle$. Kostnadshamiltonianen beskrivs vidare i avsnitt 2.3. I figur 2.3 beskrivs algoritmen för *q* kvantbitar och djup *p* schematiskt genom ett flödesschema. Utifrån denna beskrivning benämns kvantgrindarna som bygger upp \hat{U}_M och \hat{U}_C som mixerlager respektive kostnadslager.

2.2.3 Conditional Value at Risk

Traditionellt sett är det medelvärdet av energin som beräknas och används i optimeringen i QA-OA [18]. I vissa problem kan det däremot vara effektivt att enbart betrakta några av de lägsta energivärdena [19]. Detta fungerar bra för kombinatoriska problem, vilka ger diagonala Hamiltonianer i mätbasen, för då är grundtillståndet ett egentillstånd som motsvaras av den bästa mätningen. Proteinveckning är ett exempel på ett sådant problem, eftersom för att konstruera stabila protein behövs grundtillståndet, alltså tillståndet med lägst energi [19].

Väntevärdet av energin för ett kvantsystem med n sampel ges vanligtvis av

$$\langle E \rangle = \langle \psi(\vec{\gamma}, \vec{\beta}) | \hat{H}(\vec{\gamma}, \vec{\beta}) \rangle | \psi(\vec{\gamma}, \vec{\beta}) \rangle = \frac{1}{n} \sum_{k=1}^{n} C_k(\vec{\gamma}, \vec{\beta})$$
(2.23)

där $C_k(\vec{\gamma}, \vec{\beta})$ är enskilda mätningar av kvantkretsen. CVaR-metoden fokuserar på att förbättra de bästa

möjliga mätningarna istället för medelvärdet, och definieras enligt [19]:

$$CVaR_{\alpha}(X) = \mathbb{E}\left[X \mid X \le F_X^{-1}(\alpha)\right].$$
(2.24)

Här är X en variabel, $\alpha \in (0,1]$ är konfidensintervallet och F_X är den kumulativa fördelningsfunktionen för X. Detta betyder att CVaR är väntevärdet av den lägre α -andelen av fördelningsfunktionen för X, som visas i figur 2.4. Givet att alla mätvärden är sorterade i stigande ordning, med $C_{k+1} \ge C_k$, kan CVaR definieras:

$$CVaR_{\alpha}(X) = \frac{1}{\lceil \alpha n \rceil} \sum_{k=0}^{\lceil \alpha n \rceil} C_k.$$
 (2.25)

En viktig observation är att $\alpha \rightarrow 0$ motsvarar minimivärdet och $\alpha = 1$ motsvarar väntevärdet av hela fördelningsfunktionen av X [19]. Det går inte heller att sätta α till ett godtyckligt litet tal och försöka få fram grundtillståndet på det sättet. Problemet med det är att det kan finnas ett testtillstånd



Figur 2.4: CVaR-modellen utgår ifrån en variabel α definierad inom $0 \le \alpha < 1$. Utifrån vilket värde på α som bestäms kan ett optimerat väntevärde CVaR(X) bestämmas, beroende av fördelningsfunktionen av X [1]. CVaR-värdet är väntevärdet av det rödmarkerade området i figuren, som markeras med ett rött streck. Väntevärdet av hela fördelningsfunktionen benämns $\langle X \rangle$, och utgörs av det blå strecket i figuren. På *y*-axeln visas sannolikheten att mäta ett visst utfall av fördelningsfunktionen för variabeln X.

 $|\varphi(\gamma, \beta)\rangle$ som har ett överlapp ρ med grundtillståndet och CVaR inte kan särskilja på dessa tillstånd givet att $\rho \ge \alpha$ [19]. Ett annat sätt att se på det är att CVaR-metoden inte belönar resultat som har större överlapp än α . För att säkerhetställa ett resultat som har över 5% sannolikhet att hitta grundtillståndet måste $\alpha \ge 0.05$.

Vidare måste även fler mätningar göras för lägre värde på α för att få samma noggrannhet som för väntevärdet. Barkoutsos *et al.* föreslår $\frac{n}{\alpha}$ mätvärden för att få samma noggranhet som väntevärdet med *n* mätpunkter [19]. Även med färre än $\frac{n}{\alpha}$ mätvärden uppnår Barkoutsos *et al.* ett bättre resultat med CVaR-värdet än med enbart väntevärdet, vilket kan förklaras med att CVaR är bättre på att guida den klassiska optimeringsalgoritmen mot ett kvanttillstånd som överlappar väl med den faktiska än modellen utan CVaR. Samma källa påstår att CVaR-QAOA med högre djup och lägre värde på α ger ett bättre värde upp till en viss nivå och rekommenderar $\alpha \in [0,01; 0,25]$.

En fördel med CVaR-funktionen är att den verkar kunna hantera störande brus och fel som introduceras av kvantdatorns hårdvara. En förklaring till detta kan vara att CVaR-metoden fokuserar på det bästa resultatet och därmed ignorerar de sämre resultaten, vilket ofta är de som påverkas av bruset, till skillnad från den vanliga väntevärdes-metoden som värderar alla energinivåer lika mycket [19].

2.3 HP-model

För att genomföra en beräkning på ett system måste en matematisk modell för det som skall undersökas framställas. För att modellera veckningen av protein i vattenlösning har det visat sig att HPmodellen fungerar bra [40]. Därför förklaras det nedan hur den matematiska modellen transformeras för att fungera i en kvantdator.

2.3.1 Isingmodellen

Isingmodellen är en matematisk modell som berör hur interaktioner mellan närliggande spinns påverkar det totala gittrets energi [41]. Gittrets struktur för två dimensioner är uppbyggd så att det är möjligt att gå från en gitterpunkt till en annan genom att addera eller subtrahera det konstanta gitteravståndet i någon av de två ortogonala riktningarna. Med närliggande gitterpunkter menas därmed de punkter som befinner sig ett gitteravstånd från den undersökta punkten.

Varje punkt i gittret har ett eget spinn, σ_i , som kan anta värdet –1 eller +1 [42]. Hamiltonianen för systemet kan därmed skrivas som en summa av de närliggande gitterpunkternas interaktion och den enskilda gitterpunktens energi

$$\hat{H}(\sigma) = -\sum_{i,j} E_{i,j}\sigma_i\sigma_j - \sum_i J_i\sigma_i$$
(2.26)

där $E_{i,j}$ och J_i är passande parametrar [41].

2.3.2 Proteinveckning

Strukturen för proteinen byggs i denna rapport upp på ett tetragoniskt gitter med hjälp av Isingmodellen, där varje ruta numreras från 1 till N [28], [41]. Det är därmed möjligt att beskriva de jämna rutorna som x_s^f och de udda som $x_{s'}^{f'}$ där f är aminosyrornas index och beskriver vilken ordning de sitter i, s är rutnumreringen och x_s^f är 1 om aminosyran f är placerad i ruta s, och 0 annars [28].

För att simulera veckning i vattenlösning "belönar" HP-modellen de veckningar där kluster av hydrofoba aminosyror omringade av polära aminosyrorna förekommer genom att sänka dess energi [28], [43]. Energisänkningen sker enligt följande ekvation

$$E_{HH} = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s, s' \rangle} x_s^f x_{s'}^{f'}$$
(2.27)

där $C(h_f, h_{f'}) = 1$ om två närliggande aminosyror båda är H och inte kommer direkt efter varandra i kedjan, annars blir termen 0 [28].

HP-modellen tar inte enbart hänsyn till om två närliggande aminosyror på gittret både är hydrofoba utan straffar även otillåtna veckningar genom följande tre restriktioner:

- En aminosyra får endast finnas i en gitterpunkt åt gången.
- En gitterpunkt får endast innehålla en aminosyra åt gången.
- Kopplingen mellan de olika aminosyrorna måste enbart ske i vertikal eller horisontell riktning, och gitterpunkterna innehållande aminosyrorna måste angränsa till varandra.

Matematiskt kan restriktionerna uttryckas enligt följande tre ekvationer, i samma ordning som listan ovan:

$$E_1 = \sum_{f} \left(\sum_{s} x_s^f - 1 \right)^2 + \sum_{f'} \left(\sum_{s'} x_{s'}^{f'} - 1 \right)^2$$
(2.28)

$$E_2 = \frac{1}{2} \sum_{f_1 \neq f_2} \sum_{s} x_s^{f_1} x_s^{f_2} + \frac{1}{2} \sum_{f_1' \neq f_2'} \sum_{s'} x_{s'}^{f_1'} x_{s'}^{f_2'}$$
(2.29)

$$E_{3} = \sum_{1 \le f < N} \sum_{s} x_{s}^{f} \sum_{\|s' - s\| > 1} x_{s'}^{f+1} + \sum_{1 \le f < N} \sum_{s'} x_{s'}^{f'} \sum_{\|s - s'\| > 1} x_{s}^{f'+1}.$$
 (2.30)

Energitermerna E_1 , E_2 och E_3 multipliceras med skalärerna λ_1 , λ_2 respektive λ_3 där storleken på λ_i avgör hur stark restriktion E_i ska vara [28].

Restriktionerna får inte vara alltför starka, då för stort värde på λ_i gör variationen i energilandskapet för stor för att optimeraren ska hitta grundtillståndet med någorlunda säkerhet [28]. Med energilandskapet menas, för p = 1, den tredimensionella grafen till funktionen CVaR($\vec{\gamma}, \vec{\beta}$). Vid högre pblir dimensionerna för många för att funktionen ska kunna visualiseras i en graf, men principen är densamma. Då λ_i är stor kan det ske att optimeraren som används i QAOA behöver klättra över en "energitopp" för att nå en dal i energilandskapet med lägre energi. Med ett för stort λ_i ökar risken att optimeraren att ger upp på vägen upp för toppen då den bedömmer att energin inte minskar i den undersökta riktningen och når inte dalen med lägre energi. λ_i får inte heller väljas till för lågt då det riskerar att göra restriktionerna försumbara. Vilket värde på λ_i som är bäst måste avgöras experimentellt [28].

Den slutliga energifunktionen för systemet blir då

$$E = E_{HH} + \sum_{i=1}^{3} \lambda_i E_i$$
 (2.31)

som ska omvandlas till en Hamiltonian för Isingmodellen enligt ekvation 2.26. Den slutgiltiga funktionen fås till

$$\begin{aligned} \hat{H}_{cost}(\sigma) &= -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s, s' \rangle} \frac{1 - \sigma_s^f - \sigma_{s'}^{f'} + \sigma_s^f \sigma_{s'}^{f'}}{4} \\ &+ \lambda_1 \left[\sum_f \left(\sum_s \frac{-\sigma_s^f - 1}{2} \right)^2 + \sum_{f'} \left(\sum_{s'} \frac{-\sigma_{s'}^{f'} - 1}{2} \right)^2 \right] \\ &+ \lambda_2 \left[\frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \frac{1 - \sigma_s^{f_1} - \sigma_s^{f_2} + \sigma_s^{f_2} \sigma_s^{f_1}}{4} + \frac{1}{2} \sum_{f_1' \neq f_2'} \sum_{s'} \frac{1 - \sigma_{s'}^{f'} - \sigma_{s'}^{f'} + \sigma_{s'}^{f'} \sigma_{s'}^{f'}}{4} \right] \\ &+ \lambda_3 \left[\sum_{1 \le f < N} \sum_s \frac{1 - \sigma_s^f}{2} \sum_{||s' - s|| > 1} \frac{1 - \sigma_{s'}^{f+1}}{2} + \sum_{1 \le f < N} \sum_{s'} \frac{1 - \sigma_{s'}^{f'}}{2} \sum_{||s - s'|| > 1} \frac{1 - \sigma_{s'}^{f'+1}}{2} \right] \end{aligned}$$
(2.32)

där härledningen av ekvation 2.32 från ekvation 2.31 går att hitta i Appendix A.

Antalet kvantbitar, q, som beskriver proteinveckningen på ett gitter med längderna L_x , L_y längs med *x*- respektive *y*-axel för en given aminosyrasekvens på n_A aminosyror ges av

$$q = \frac{L_x L_y}{2} n_A. \tag{2.33}$$

2.3.3 Grundtillstånd

Då kvantalgoritmer eller klassiska algoritmer används för att beräkna lösningar för proteinveckningsproblemet, är det just aminosyrasekvensernas grundtillstånd som söks. I detta projekt som kretsar kring att utvärdera QAOA med CVaR jämfört med traditionell QAOA, används dock tre stycken enkla instanser av HP-modellen för proteinveckningsproblemet där grundtillstånden redan är kända. På så sätt kan resultaten av simuleringar alltid verifieras. Nedan presenteras visuella representationer av de olika grundtillstånden för de undersökta instanserna. För HPPH-aminosyrasekvensen på ett 2×2-gitter finns det fyra stycken grundtillstånd, vilka visas i figur 2.5. Alla grundtillstånden har en egen specifik bitsträng på åtta kvantbitar. Sannolikheten att mäta ett grundtillstånd är därmed $\frac{4}{2^8} = 1,5625\%$. För HPPH-aminosyrasekvensen på ett 3×2-gitter finns det istället åtta grundtillstånd, se figur 2.6, vilket medför att sannolikheten för att mäta korrekt bitsträng för grundtillståndet är $\frac{8}{2^{12}} = 0,1953\%$. För HHHPPH-aminosyrasekvensen på ett 3×2-gitter finns två antal grundtillstånd som representeras av bitsträngar på 18 kvantbitar, se figur 2.7. Detta medför att sannolikheten att slumpmässigt mäta rätt bitsträng för grundtillståndet är $\frac{2}{2^{18}} = 0,0007629\%$



Figur 2.5: De fyra olika grundtillstånden för instansen med aminosyra: HPPH på ett 2×2-gitter. De röda cirklarna representerar H-aminosyror, de blåa cirklarna representerar P-aminosyror.



Figur 2.6: De åtta olika grundtillstånden för instansen med aminosyra: HPPH på ett 3×2-gitter. De röda cirklarna representerar H-aminosyror, de blåa cirklarna representerar P-aminosyror.



Figur 2.7: De två olika grundtillstånden för instansen med aminosyra: HHHPPH på ett 3×2-gitter. De röda cirklarna representerar H-aminosyror, de blåa cirklarna representerar P-aminosyror.

3 Metod

I metoden beskrivs arbetets gång för att ta fram koden som använts för att testa QAOA med CVaR applicerat på HP-modellen. Även bakgrund och teori till parameterval gås igenom. Slutligen gås det även igenom varför de parametrar som undersöks är relevanta att undersöka. Koden som utvecklats och använts i detta projekt återfinns på GitHub [44].

3.1 Programmets utformning

Programmet som använts för simulering och datainsamling bygger till stor del på gitbiblioteket 'vqa', utvecklat av David Fitzek, Hanna Linn och Laura García-Álvarez [45]. Biblioteket innehåller verktyg,

funktioner och exempel som underlättar och kompletterar skapande och modifiering av proteingitter enligt HP-modellen. GitHub-biblioteket bygger i sin tur på Pythonbiblioteket PennyLane [46]. PennyLane är ett ramverk för kvantfysikprogrammering, och används i programmet för att utföra själva simuleringen av kvantdatorn.

Programmets utformning kan delas upp i tre olika delar. I första delen definieras proteinets och gittrets form och storlek. Båda två sparas som ett objekt av klassen *CoordinateBased_HPLattice* från vqa-biblioteket. Kostnadshamiltionianen och kostnadslagrets form uttryckt i kvantgrindar beräknas därefter utifrån proteinobjektet. Mixerlagret bestående av NOT-grindar skapas utifrån antalet kvantbitar.

I nästa del används PennyLane för att definiera funktionerna som skapar själva kvantkretsen, qaoa_layer(γ, β) och circuit($\vec{\gamma}, \vec{\beta}$). Grindarna från kostnad- och mixerlagren läggs parvis till i kretsen enligt värdet på djupet p. Notera att varje lagers utformning beror på parametrarna γ och β , vilka hela tiden modifieras av optimeraren. Därför omdefinieras hela kretsen enligt de nya parametrarna vid varje steg som tas av optimeraren.

I den sista delen definieras kostnadsfunktionen $CVaRsamp(\vec{\gamma}, \vec{\beta}, n, \alpha)$, alltså funktionen som ska minimeras av optimeraren. Den utför följande operationer:

- 1. Tar parametrarna $\vec{\gamma}, \vec{\beta}$, antalet sampel *n* och CVaR-konstanten α som invärde.
- 2. Kallar på funktionen $get_probs(\vec{\gamma}, \vec{\beta})$.
- 3. $get_probs(\vec{\gamma}, \vec{\beta})$ kallar i sin tur på funktionen $circuit(\vec{\gamma}, \vec{\beta})$, som med hjälp av $qaoa_layer(\gamma, \beta)$ bygger upp och returnerar den nya kretsen utifrån $\vec{\gamma}, \vec{\beta}$.
- 4. *get_probs* beräknar sannolikhetsfördelningen för alla möjliga proteinkonfigurationer utifrån den nya kretsen, och returnerar denna sannolikhetsfördelning som en vektor.
- 5. *n* stycken sampel tas av den returnerade sannolikhetsfördelningen. På så sätt efterliknas mätprocessen för en riktig kvantdator, där den exakta sannolikhetsfördelningen är okänd. Deras energier beräknas och sorteras i en lista.
- 6. CVaR-värdet beräknas utifrån de $(100 \times \alpha)$ % minsta sampelenergierna, och returneras till optimeraren.

Optimeraren som använts i projektet heter COBYLA (Constrained Optimization By Linear Approximation Optimizer) och kommer från Pythonbiblioteket SciPy [47]. Eftersom kostnadsfunktionen är samplingsbaserad krävs det att optimeraren inte bygger på gradientberäkning [48]. Av de tillgängliga gradientfria optimerarna användes specifikt COBYLA för att den utnyttjats för QAOA i ett flertal tidigare studier [1], [49], [50]. Den är också tillräckligt snabb för hinna utföra alla mätningar inom rapportens tidsgränser. En annan gradientfri optimerare, GPyOpt, övervägdes och testades, men trots att den i många fall var effektivare än COBYLA, valdes den bort då den ansågs vara sämre när det kom till att göra konsekventa jämförelser av antalet iterationer som behövdes för att nå en lösning [51]. Denna jämförelse av antalet iterationer är viktig då det är detta som till stor del avgör hur lång tid kvanthybridalgoritmen behöver för att hitta en lösning.

Optimeraren COBYLA tar som inparametrar kostnadsfunktionen, de valda startparametrarna $\vec{\gamma}_0$, $\vec{\beta}_0$ samt det maximala antalet iterationer och tolerans. Toleransen kan förklaras som den önskade nog-

grannheten för det slutliga svaret. Den sattes till 0,001 för samtliga instanser, vilket grundades i mindre testkörningar där detta värde presterade bäst. Högsta antalet iterationer valdes till 1000 för samtliga instanser, och användes bara som felkontroll. Alla mätningar förväntades avsluta långt innan detta antal iterationer.

Med programmet färdigställt kunde datainsamlingen påbörjas. De olika instanserna kördes i ordning av minst beräkningskraft först.

3.2 Parameterval

I detta delavsnitt diskuteras vilka instanser som simulerats. De värden på α som undersökts är $\alpha = [0,05; 0,10; 0,25; 1]$. Begränsningen i att testa α -värden för CVaR mellan 0,05 och 0,25 sattes med ändamålet att det skulle ge ett mer distinkt resultat vid jämförelse med vanlig QAOA, alltså då $\alpha = 1$.

Proteinkedjorna som undersökts består av fyra eller sex aminosyror. Kedjor med fyra aminosyror simulerades på gitter med dimensioner 2×2 och 3×2, medan det med sex endast simulerades på 3×2-gittret. För varje proteinkedja, och för varje p = [1, 2, 3], simulerades tre fall med olika startparametrar för varje: $\vec{\gamma}_0$, $\vec{\beta}_0 = [[0,1^{\otimes p}][0,1^{\otimes p}]]$, $[[0,5^{\otimes p}][0,5^{\otimes p}]]$ eller $[[1,0^{\otimes p}][1,0^{\otimes p}]]$. Notationen $[a^{\otimes p}]$ definieras i denna rapport som en vektor med längd p, och med värdet a i varje position. Exempelvis är alltså $[[0,1^{\otimes 2}][0,1^{\otimes 2}]] = [[0,1;0,1],[0,1;0,1]]$. Eftersom energiväntevärdet $\langle E(\vec{\gamma}_0, \vec{\beta}_0) \rangle$ för kretsarna som motsvarar de olika startparametrarna kan beräknas utifrån sannolikhetsfördelningen, ger detta ett approximativt mått på hur långt ifrån det globala minimumet som optimeringen börjar. Därför kunde dessa värden väljas relativt godtyckligt, men de hade modifierats om deras motsvarande $\langle E(\vec{\gamma}_0, \vec{\beta}_0) \rangle$ hade visat sig alltför nära varandra i storlek. Detta gjordes i ändamål att undersöka om metodernas prestation i jämförelse med varandra ändras beroende på kvalitén av startparametrarna.

Antalet mätvärden *n* valdes lågt, n = 32 för HPPH-aminosyrasekvensen medan den valdes till n = 1000 för HHHPPH-aminosyrasekvensen, vilket för HHHPPH-aminosyrasekvensen är lågt då den har 2^{18} möjliga utfall på ett 3×2-gitter. Motiveringen var att efterlikna mer realistiska mätningar då det är väldigt osannolikt att mäta grundtillståndet på större instanser. I instansen på 2×2-gittret var detta oundvikligt, eftersom sannolikheten att mäta grundtillståndet, P_{GS} , var $\frac{1}{64}$ redan i initialtillståndet. För de andra konfigurationerna var detta ett mindre problem, eftersom de innehöll många fler möjliga konfigurationer. För att undersöka vilken påverkan detta valet av *n* har på optimeringsprocessen utfördes några testserier med n = 1000 för jämförelse. λ_i valdes så att $[\lambda_1, \lambda_2, \lambda_3] = [2,1;2,4;3,0]$ för alla utförda simuleringar, baserat på tidigare experiment utförda av handledare. Djupet *p* begränsades till ≤ 3 , vilket motiveras i kapitel 1.3.

När en aminosyrasekvens och övriga parametrar valts, exekverades optimeringen 1000 gånger för varje värde av α på 2×2-gittret, och 100 gånger på 3×2-gittret. Antalet körningar minskades på grund av den ökade exekveringstiden vid större instanser. Antalet steg för varje iteration, P_{GS} , samt CVaR-värdet sparas i en CSV-fil.

3.3 Analys av variansen för CVaR beroende på antal sampel

Eftersom bitsträngarna som används för att beräkna *CVaRsamp* är samplade uppstår en varians av värdet som matas in i optimeraren. Ifall denna varians är för stor kan optimeraren ej fungera som tänkt. Med för stor varians menas att den är större än vad skillnaden i värdet från *CVaRsamp* skulle bli med en nödvändigt liten korrigering av $\vec{\gamma}, \vec{\beta}$.

För att få ett mått på hur stor variansen är för olika α , instanser och antal sampel utfördes tester på

detta enligt följande. För en given instans sattes p = 1. Sedan valdes 100 jämnt fördelade punkter i $\gamma\beta$ planet sådana att $(\gamma, \beta) \in [0, 2\pi] \times [0, 2\pi]$. I vardera punkt beräknades sedan värdet på *CVaRsamp* 100 gånger för varje $\alpha \in [0,05; 0,10; 0,25; 1]$. Utifrån dessa värden beräknades sedan standardavviklesen i punkten ut för varje α . Därefter beräknades den genomsnittliga standardavvikelsen i planet ut för varje α , samt även standardavvikelsen för denna. Dessa tester utfördes på HPPH-kedjan för 2×2-gittret och 3×2-gittret med både n = 32 och n = 1000. Vidare gjordes samma test på HHHPPH-kedjan för 3×2gittret med n = 1000, fast med enbart 25 punkter i planet och 30 beräkningar i varje punkt för varje α . Antalet punkter och beräkningar bestämdes i båda fallen enbart med avseende på tiden testerna tog. Resultaten av ovanstående tester plottades sedan för varje instans med standardavvikelsen på *y*-axeln och α på *x*-axeln.

3.4 Jämförelse av resultat

I riktiga kvantdatorkalkylationer är det standard att mäta fler än 1000 mätvärden per tillstånd, för att få en godtagbar uppfattning om sannolikhetsfördelningen. Därför behöver $P_{GS}(\vec{\gamma}, \vec{\beta})$ i slutet av optimeringen endast vara av storleksordningen 0,1% för att man ska vara relativt säker (> 60%) på att mäta ett grundtillstånd åtminstone en gång. Högre sannolikhet för grundtillstånd är därför inte alltid nödvändigt, vilket gör att endast $P_{GS}(\vec{\gamma}, \vec{\beta})$ ej är en bra jämförelsegrund. Därför tas både $P_{GS}(\vec{\gamma}, \vec{\beta})$ och antalet iterationer hänsyn till vid jämförelse. För att bibehålla objektivitet kommer huvudsakligen mätvärden som är bättre eller lika bra i båda dessa aspekter klassas som bättre än något annat värde. Alltså, ett värde med högre $P_{GS}(\vec{\gamma}, \vec{\beta})$, men fler antal utförda iterationer jämfört med något annat värde, bör ej klassificeras som bättre än detta andra värde. Denna strategi har antagits på grund av att det som är viktigast, $P_{GS}(\vec{\gamma}, \vec{\beta})$ eller antalet iterationer, varierar beroende på problemet i fråga.

4 Resultat av testning

Här presenteras resultaten av de mätningar som genomförts i studien. Koden som använts för att genomföra mätningarna återfinns här [44]. För att jämföra QAOA med CVaR mot konventionell QAOA, med väntevärdet av energin, har mätningar med olika parametrar genomförts. Vilka parametrar som varierats och deras respektive värden är beskrivna i metodavsnittet 3.2.

4.1 Utvärdering av initialparametrar

För att gradera initialparametrarna $\vec{\gamma}_0$, $\vec{\beta}_0$ beräknas deras motsvarande tillstånds energiväntevärde $\langle E(\vec{\gamma}_0, \vec{\beta}_0) \rangle$. Dessa värden återfinns i tabell 4.1 nedan. Mätningarna för p = [2, 3] för 3×2, HHHPPH-instansen innebar en för hög beräkningstid, vilket är varför de inte finns med i tabellen.

	$[ec{\gamma}_0,ec{eta}_0]$	2×2, HPPH	3×2, HPPH	3×2, HHHPPH
	[[0,1][0,1]]	0,524	3,579	7,389
<i>p</i> =1	[[0,5][0,5]]	4,082	1,143	-1,055
	[[1,0][1,0]]	-0,089	-0,138	-0,1629
	[[0,1;0,1][0,1;0,1]]	1,449	8,629	
<i>p</i> =2	[[0,5;0,5][0,5;0,5]]	3,308	3,248	
	[[1,0;1,0][1,0;1,0]]	1,327	0,140	
<i>p</i> =3	[[0,1;0,1;0,1][0,1;0,1;0,1]]	2,567	11,830	
	[[0,5;0,5;0,5][0,5;0,5;0,5]]	1,910	3,043	
	[[1,0;1,0;1,0][1,0;1,0;1,0]]	1,763	0,005	

Tabell 4.1: $\langle E(\vec{\gamma}_0, \vec{\beta}_0) \rangle$ av det motsvarande tillståndet för varje uppsättning startparametrar, för samtliga beräknade instanser.

4.2 HPPH-aminosyrasekvensen på 2×2-gittret

Resultaten av simuleringarna på 2×2-gittret visas i figur 4.1. Mätningarna visar att det finns en korrelation mellan storleken på α -parametern och antalet iterationer i optimeraren. Settet av initialparametrar skiljer sig mellan dessa grafer. Ett lägre α ger ett mindre antal iterationer. Det går också att observera att ett högre djup *p* tydligt leder till fler iterationer utförda, men i många fall inte till ett högre $P_{GS}(\vec{\gamma}, \vec{\beta})$.



(a) Mätningar med initalparametrar $[[0,1^{\otimes p}][0,1^{\otimes p}]]$ och 32 sampel.



(c) Mätningar med initialparametrar $[[1,0^{\otimes p}][1,0^{\otimes p}]]$ och 32 sampel.



(b) Mätningar med initialparametrar $[[0,5^{\otimes p}]][0,5^{\otimes p}]]$ och 32 sampel.



(d) Mätningar med initialparametrar $[[1,0^{\otimes p}][1,0^{\otimes p}]]$ och 1000 sampel.

Figur 4.1: Resultat från mätningar av QAOA med CVaR med $\alpha = [0,05; 0,1; 0,25; 1,0]$ för HPPH kedja på 2×2-gittret och p = [1, 2, 3]. På y-axlarna visas sannolikheten att mäta ett av grundtillstånd efter optimeringen, $P_{GS}(\vec{\gamma}, \vec{\beta})$. På x-axlarna visas hur många iterationer som behövdes innan optimeringen avslutades. Punkterna representerar de genomsnittliga resultaten av 1000 mätningar per punkt. För varje punkt visas även standardavvikelsen både i x- och y-led. Samtliga djup p är plottade med olika symboler, p = 1 med cirkel, p = 2 med kryss och p = 3 med triangel. De negativa $P_{GS}(\vec{\gamma}, \vec{\beta})$ beror på en asymmetrisk fördelning som ger upphov till en stor varians, det finns egentligen inga negativa sannolikheter.

4.3 HPPH-aminosyrasekvensen på 3×2-gittret

Resultatet av simulationerna utförda på instansen 3×2, HPPH, presenteras i figur 4.2. I figurerna 4.2a-4.2c är n = 32 och p = [1, 2, 3]. Settet av initialparametrar skiljer sig mellan dessa grafer. I figur 4.2d har antalet sampel ökats till n = 1000. För denna instans har $\alpha = 0,05$ genomgående presterat bäst vid djup p = 3, n = 32, jämfört med de andra α . Med bäst menas att processen krävde färre iterationer samt som den resulterade i högst $P_{GS}(\vec{\gamma}, \vec{\beta})$. I fallen med p = [1, 2] är resultaten inte lika entydiga, men för många av dem har CVaR, alltså fallen med $\alpha \neq 1$, gett bättre resultat jämfört med konventionell QAOA.



(a) Mätningar med initialparametrar $[[0,1^{\otimes p}][0,1^{\otimes p}]]$ och 32 sampel.



(c) Mätningar med initialparametrar $[[1,0^{\otimes p}][1,0^{\otimes p}]]$ och 32 sampel.



(b) Mätningar med initialparametrar $[[0,5^{\otimes p}][0,5^{\otimes p}]]$ och 32 sampel.



(d) Mätningar med initialparametrar $[[1,0^{\otimes p}][1,0^{\otimes p}]]$ och 1000 sampel.

Figur 4.2: Resultat från mätningar av QAOA med CVaR med $\alpha = [0,05; 0,1; 0,25; 1,0]$ för HPPH-kedja på 3×2-gittret. På *y*-axlarna visas sannolikheten att mäta något grundtillstånd efter optimeringen, $P_{GS}(\vec{\gamma}, \vec{\beta})$. På *x*-axlarna visas hur många iterationer som behövts innan optimeringarna avslutades. Punkterna representerar de genomsnittliga resultaten av 100 mätningar per punkt. För varje punkt visas även standardavvikelsen både i *x*- och *y*-led. Samtliga djup *p* är plottade med olika symboler, p = 1 med cirkel, p = 2 med kryss och p = 3 med triangel. De negativa $P_{GS}(\vec{\gamma}, \vec{\beta})$ beror på en asymmetrisk fördelning som ger upphov till en stor varians. Det finns egentligen inga negativa sannolikheter.

4.4 HHHPPH-aminosyrasekvensen på 3×2-gittret

I figur 4.3 är resultaten för simulationerna utförda på instansen 3×2-gittret med HHHPPH-aminosyrasekvensen presenterade. Notera att *x*-axeln har högre upplösning här jämfört med i de tidigare presenterade graferna, och att standardavvikelsen av antalet iterationer generellt sett inte är högre i denna instans än i de mindre instanserna. Notera även att graferna har varierande upplösning längs *y*-axlarna. I denna instans varierade metodernas prestationer mycket för de olika undersökta initialparametrarna. I figur 4.3b gav konventionell QAOA mycket högre $P_{GS}(\vec{\gamma}, \vec{\beta})$ än QAOA med CVaR, men i figur 4.3c gav konventionell QAOA tydligt lägst $P_{GS}(\vec{\gamma}, \vec{\beta})$.



(a) Mätningar med initialparametrar $[[0,1^{\otimes p}][0,1^{\otimes p}]]$, för p = 1.



(c) Mätningar med initialparametrar $[[1,0^{\otimes p}][1,0^{\otimes p}]]$, för p = 1.



(b) Mätningar med initialparametrar $[[0,5^{\otimes p}][0,5^{\otimes p}]]$, för p = 1.

Figur 4.3: Resultat från mätningar av QAOA med CVaR med $\alpha = [0,05;0,1;0,25;1,0]$ för HHHPPHkedja på 3×2-gittret, med 1000 sampel av kvantkretsen vid varje interation. På y-axlarna visas sannolikheten att mäta något av grundtillstånden efter optimeringen, $P_{GS}(\vec{\gamma}, \vec{\beta})$. På x-axlarna visas hur många iterationer som behövts innan optimeringarna avslutats. Punkterna representerar de genomsnittliga resultaten av 1000 mätningar per punkt. För varje punkt visas även standardavvikelsen både i x- och y-led. Endast djup p = 1, är plottad. De negativa P_{GS} beror på en asymmetrisk fördelning som ger upphov till en stor varians. Det finns egentligen inga negativa grundtillståndssannolikheter.

4.5 Standardavvikelsen av CVaR beroende på antal sampel

I figur 4.4 presenteras resultat för standardavvikelsen i CVaR beroende på antalet sampel, vilka beskrevs i avsnitt 3.3. Avvikelsen i antal mätpunkter och antal individuella mätningar i figur 4.4c jämfört med figurerna 4.4a, 4.4b är på grund av beräkningstid, då HHHPPH-instansen var markant mer beräkningstung är HPPH-instanserna. Standardavvikelserna kan jämföras med de energier som kan mätas för given instans. Dessa energispann återfinns i respektive figurtext.



(a) För HPPH-aminosyrasekvensen på 2×2 gittret med 32 sampel och 1000 sampel. Mätning arna är gjorda på 100 jämt fördelade mätpunkter. I de 100 punkterna har 100 individuella mätningar gjorts. Energin för instansen kan anta värden mellan -6,6 till 8,4.



(c) För HHHPPH-aminosyrasekvensen på 3×2 -gittret med 1000 sampel. Mätningarna är gjorda på 25 jämt fördelade mätpunkter. I varje punkt har 30 individuella mätningar gjorts. Energin för instansen kan anta värden mellan -29,4 till 82,2.



(b) För HPPH-aminosyrasekvensen på 3×2 -gittret med 32 sampel och 1000 sampel. Mätningarna är gjorda på 100 jämt fördelade mätpunkter. I de 100 punkterna har 100 individuella mätningar gjorts. Energin för instansen kan anta värden mellan -15,75 till 44,25.

Figur 4.4: Genomsnittliga standardavvikelser samt standardavvikelserna av dessa standardavvikelser för CVaR för olika instanser. Notera att skalan på y-axlarna i (a) och (b) skiljer sig från skalan på y-axeln i (c), och att alla mätvärden för n = 1000 ligger i ungefär samma intervall.

5 Diskussion

I diskussionen jämförs likheter och skillnader i resultaten av de olika mätningarna. Från dessa diskussioner kan sedan vissa slutsatser dras.

5.1 Generell resultatdiskussion

I detta delavsnitt presenteras en kort sammanfattning av de sammanställda resultaten. QAOA med CVaR har i alla utförda simuleringar resulterat i en kortare optimeringsprocess, alltså stannat vid färre iterationer, jämfört med vanlig QAOA. Vilken metod som gett högst grundtillståndssannolikhet $P_{GS}(\vec{\gamma}, \vec{\beta})$ varierar mellan instanserna. Någon typ av QAOA med CVaR har dock presterat bättre i de flesta simulerade fallen även i denna aspekt. Detta tyder på att dessa metoder definitivt är fördelaktiga i vissa situationer, och kan vara värda att studera djupare. Med vidare studier hade det varit möjligt att bestämma vilket α som lämpar sig bäst i allmänthet, samt för vilka problem som dessa metoder ger störst förbättring.

En annan viktig observation som framkommit är att sampelantalet *n* inte bör väljas alltför lågt, även vid små instanser. Även om *n* förefaller tillräckligt högt i förhållande till antalet möjliga tillstånd, kan det, om de möjliga tillstånden är för få, ge en fördelning med alltför hög varians. Detta leder i sin tur till ett beräknat CVaR med samma problematik. Detta visas tydligt i graferna i figur 4.3. Konsekvensen av denna varians är att optimeringen blir mindre effektiv, eftersom energilandskapet blir för varierande. En tillräckligt stabil och högupplöst fördelning krävs för att optimeringen ska fungera på ett tillfredsställande sätt. Sampelantalet *n* behöver dock ej skalas upp proportionellt med antal möjliga konfigurationer för att behålla samma upplösning, utan snarare med skillnaden mellan den högsta och lägsta möjliga energin. Sampelantalet bör därför inte behöva ökas särskilt mycket vid ökning av instansstorlek, givet att variansen vid den ursprungliga instansen var acceptabel.

5.2 Initialparametrar

För samtliga mätningar har olika initialparametrar använts. Hur bra initialparametrarna är avgör hur enkelt eller svårt det är att finna tillstånd med högt $P_{GS}(\vec{\gamma}, \vec{\beta})$. I tabell 4.1 jämförs initialparametrarna, låga värden motsvarar bra initialparametrar. Eftersom varje ökning av djupet p medför två dimensioner till, så är energilandskapen helt olika för olika p. Trots detta så är initialparametrarna [[1,0^{\&p}]] alltid bäst för HPPH-aminosyran.

En tydlig skillnad i tabell 4.1 är värdena på $\langle E(\vec{\gamma}_0, \vec{\beta}_0) \rangle$ för instanserna 3×2, HPPH och 2×2, HPPH. För den större instansen är spridningen av värdena, det vill säga att skillnaden mellan största och minsta $\langle E(\vec{\gamma}_0, \vec{\beta}_0) \rangle$, mycket större än för mindre. Detta tyder på att dåliga initialparametrar för 3×2 har större påverkan än för 2×2. Detta innebär i sin tur att högre djup och större instanser kräver bättre initialparametrar för att kunna genomföra optimeringen effektivt.

5.3 Prestation för de olika instanserna

De tre undersökta instanserna har olika många möjliga tillstånd, men ungefär lika många grundtillståndskonfigurationer. Detta medför att sannolikheten att helt slumpmässigt mäta ett grundtillstånd kommer att variera mycket mellan instanserna. $P_{GS}(\vec{\gamma}, \vec{\beta})$ bör därför endast jämföras mellan mätningar utförda på samma instans. Exempelvis är $P_{GS}(\vec{\gamma}, \vec{\beta}) = 1\%$ ett dåligt resultat för instansen 2×2, HPPH, men ett väldigt bra resultat för 3×2, HHHPPH.

5.3.1 HPPH-aminosyrasekvensen på 2×2-gittret

I figur 4.1b finns tre tydliga uppdelningar beroende på vilket djup p som mätningen har. Ett högre djup ger upphov till fler dimensioner och, i teorin ett mer exakt svar, enligt ekvation 2.21 för adiabatisk kvantberäkning. Dock medför det även en längre beräkningstid och fler iterationer, vilket också kan observeras i figur 4.1b. Från mätningarna framgår trots detta att sannolikheten att mäta grundtillståndet inte alltid ökar med p. Detta kan ses i resultaten för både 2×2- och 3×2-gittret med HPPH-aminosyrasekvensen, se figur 4.1 och figur 4.2.

Sannolikheten $P_{GS}(\vec{\gamma}, \vec{\beta})$ minskar dock inte med djupet för alla mätningar. I figur 4.1b, instansen där HPPH-aminosyrasekvensen beräknas på 2×2-gittret med startparametrar $[[0,5^{\otimes p}][0,5^{\otimes p}]]$, så ökar sannolikheten att mäta grundtillståndet som väntat med djupet. Den troliga förklaringen är dock att initialparametrarna är av högre kvalité för $[\vec{\gamma}_0, \vec{\beta}_0] = [[0,5^{\otimes p}][0,5^{\otimes p}]]$ än för de andra undersökta alternativen. Som kan ses i tabell 4.1, är så fallet för $[\vec{\gamma}_0, \vec{\beta}_0] = [[0,5^{\otimes p}][0,5^{\otimes p}]]$ då de är de enda initialparametrar där $\langle E \rangle$ minskar när djupet ökar.

5.3.2 HPPH-aminosyrasekvensen på 3×2-gittret,

Till skillnad från testerna på 2×2-gittret påverkas sannolikheten att mäta grundtillståndet för HPPHaminosyrasekvensen inte lika mycket av α för de olika djupen på 3×2-gittret, se figur 4.2. För mätningarna med djup p = 3 syns det att $\alpha = 0,05$ genomgående presterat bäst, då de har samma eller högre genomsnittligt $P_{GS}(\vec{\gamma}, \vec{\beta})$ som de andra α -värdena, men med något färre iterationer utförda. För de andra djupen är resultaten mindre tydliga, eftersom standardavvikelserna är stora och de beräknade medelvärdespunkterna ligger endast en eller två iterationer från varandra. Det finns inte heller något tydligt bästa α vid dessa djup, utan det värde som uppvisat bäst prestation varierar för de olika startparametrarna.

Vid jämförelse med resultaten för n = 1000, figur 4.2d, och n = 32, figur 4.2a till 4.2c, för instansen 3×2 , HPPH, kan en tydlig ändring av $P_{GS}(\vec{\gamma}, \vec{\beta})$ observeras. För alla fall förutom då $\alpha = 1, p = 1$, där påverkan på $P_{GS}(\vec{\gamma}, \vec{\beta})$ av den ökande sampelmängden är minimal, har $P_{GS}(\vec{\gamma}, \vec{\beta})$ approximativt dubblerats. En möjlig anledning till detta är att det faktum att en stor del av sampelpunkterna försummas när CVaR-metoden används. Detta bör leda till högre varians av kostnadsfunktionens beräknade värde, jämfört med värdet i konventionell QAOA. Därför kan CVaR-metoderna vara mer känsliga för alltför lågt sampelantal, och vidare förbättras mer av ökat sampelantal.

Trots att $P_{GS}(\vec{\gamma}, \vec{\beta})$ approximativt dubbleras för n = 1000 jämfört med n = 32 kvarstår problemet att sannolikheten att mäta grundtillståndet minskar med djupet. Detta indikerar att det inte är det låga sampelantalet som orsakar att sannolikheten att mäta grundtillståndet minskar med djupet, utan snarare att det är en egenskap av den väg optimeraren tar genom energilandskapet.

5.3.3 HHHPPH-aminosyrasekvesen på 3×2-gittret

I resultaten från HHHPPH-aminosyrasekvensen framkommer en större procentuell variation av det slutliga värdet $P_{GS}(\vec{\gamma}, \vec{\beta})$ för de olika värdena av α . I fallet med initialparametrar $[[0,1^{\otimes p}][0,1^{\otimes p}]]$, vars resultat visas i figur 4.2a, är det högsta genomsnittliga värdet av $P_{GS}(\vec{\gamma}, \vec{\beta})$ ungefär sex gånger mindre än fallet med initialparametrar $[[1,0^{\otimes p}][1,0^{\otimes p}]]$, vars resultat återfinns i figur 4.3c. Utöver detta förändras hur väl de olika värdena av α presterar jämfört med varandra väldigt mycket för olika initialparametrar. Detta är tydligast i figur 4.3b, där $\alpha = 1$ presterar långt bättre än alla versioner av CVaR, trots att $\alpha = 1$ presterar sämst för de båda andra initialparametrarna. Den troliga förklaringen är att den fastnar i ett lokalt minima, som enbart $\alpha = 1$ med fler sampelpunkter använda vid beräkning av $\langle E \rangle$, kan ta sig förbi. På grund av detta kan inga definitiva slutsatser

dras för denna instans angående vilken metod som generellt sett är bäst, men för initialparametrarna $[\vec{\gamma}_0, \vec{\beta}_0] = [[1, 0^{\otimes p}][1, 0^{\otimes p}]]$, figur 4.3c, presterar QAOA med CVaR betydligt bättre.

5.4 Konsekvenser av CVaR-avvikelse beroende på antal sampel

I figur 4.4 återfinns resultaten av undersökningen angående variansen av CVaR, beroende på antalet sampel n som använts i beräkningen. Undersökningen utfördes för att undersöka vilken påverkan n har på optimeringsprocessen, och vidare om dess ökning hade gett mer koherenta resultat och mer korrekt optimering.

I figur 4.4a och 4.4b framgår att standardavvikelsen för HPPH-aminosyrasekvensen på 2×2- respektive 3×2-gitter överstiger 0,5 energienheter i samtliga fall då n = 32, vilket ter sig förhållandevis högt. Eftersom sampling sker vid varje iteration av optimeringsslingan, är en uppskattning av samplingens standardavvikelse relevant. En alltför hög standardavvikelse betyder att samma $\vec{\gamma}, \vec{\beta}$ ger alltför varierande resultat vid två olika mätningar. Optimeraren kan därför inte avgöra om skillnader i det uppmätta värdet beror på skillnader i $\vec{\gamma}, \vec{\beta}$, eller på slumporsakade skillnader i sampelpunkternas energier. Detta kan även bidra till en längre beräkningstid, eftersom det blir svårt för optimeraren att detektera när den väl befinner sig i en minimipunkt. Från figur 4.4a och 4.4b framgår det också att standardavvikelserna då n = 1000 är betydligt lägre än då n = 32. Jämförs detta med figur 4.1d och 4.2d är det tydligt att ett högre antal sampel leder till ett högre P_{GS} . Detta beror antagligen på att i en mätning med fler sampel, och minskad standardavvikelse, minskar sannolikheten att inte sampla några bra tillstånd. Därmed kommer optimeraren med högre sannolikhet få energier som är nära grundtillstånden och därmed få en högre sannolikhet att mäta dessa.

Ytterligare en observation från figur 4.4 är att värdet av standardavvikelsen förblir ungefär konstant när storleken på proteinerna ökar vid 1000 sampel. Detta trots att antalet möjliga utfall för HHHPPH är 2^6 gånger högre än för HPPH på 3×2 -gittret och 2^{10} gånger högre än HPPH på 2×2 . Alltså behöver inte antalet sampel öka med storleken på aminosyrasekvensen för att ge ett ungefär lika statistiskt pålitligt svar.

5.5 Förslag inför vidare studier

För vidare studier av QAOA med CVaR för proteinveckningsproblemet föreslår vi att tester görs på större instanser av HP-modellen. Anledningen är att det hade kunnat visa mer konkret om implementering av CVaR medför en förbättring vid beräkningar som är mer lika verkliga proteinveckningsproblem. För att kunna genomföra den typen av tester behövs antagligen mer beräkningskraft. Vidare kan det vara givande att försöka uppnå högre $P_{GS}(\vec{\gamma}, \vec{\beta})$ genom att öka djupet p. Resultaten i detta arbete visar samtidigt att större djup inte nödvändigtvis implicerar högre $P_{GS}(\vec{\gamma}, \vec{\beta})$, utan att det i stor del beror på initialparametrarna. Därav behöver djupet studeras tillsammans med valet av initialparametrar. Det hade även varit intressant att testa QAOA med CVaR på en riktig kvantdator, för att se hur resultaten skiljer sig ifrån simuleringarna när brus och störningar påverkar.

6 Etiska aspekter

Generellt finns det ett stort antal etiska problem som uppstår om en kvantdator utvecklas som kan lösa problem långt utom räckhåll för dagens bästa superdatorer. Dessa etiska problem kommer inte av teknologin i sig, utan är snarare en effekt av att kraftfullare verktyg för problemlösning kan appliceras på många typer av problem. Det är naturligtvis möjligt att använda exempelvis en mycket kraftfull optimeringsmetod för att optimera hemskheter, exempelvis inom krigföring och massövervakning, som påverkar samhället och individer mycket negativt. Det är därför viktigt att vara medveten om kraften i de verktyg som utvecklas åt framtiden, så att man på betydelsefulla sätt kan säkerställa att dessa verktyg används för allmänhetens bästa.

Ett konkret exempel på vad en kvantdator skulle kunna användas till i framtiden, som kanske inte av alla ses som fullständigt oetiskt, är att med hjälp av Shors algoritm knäcka krypteringsalgoritmen RSA, och genom detta exempelvis komma åt integritetskränkande och hemlig information [52], [53].

I projektet har en modifikation som kan förbättra QAOA för proteinveckning utforskats. Med en förbättrad optimeringsalogritm, en mer exakt beskrivning av proteinveckningen och med en mycket kraftfullare kvantdator än vad vi har idag, skulle strukturen av större protein kunna beräknas. Med detta verktyg skulle forskare snabbare kunna ta fram nya och effektivare läkemedel än vad konventionella tillvägagångsätt tillåtit. Även om nya typer av läkemedel kan medföra risker, anser vi att dessa i allra högsta grad övervägs av alla de positiva konsekvenser som följer av läkemedelsutveckling.

7 Slutsats

Värdet av P_{GS} blev för samtliga simulerade fall större än P_{GS} för en helt slumpmässig mätning då alla tillstånd är lika sannolika. Vi kan alltså konstatera att QAOA med CVaR är en fungerande metod åtminstone för de instanser som undersökts i projektet. Vidare så tycks QAOA med CVaR för de undersökta α vara bättre än traditionell QAOA för de undersökta instanserna, då mätningarna (nästan) alltid kräver färre iterationer, och i de flesta fall uppnår ett högre eller likvärdigt P_{GS} . Standardavvikelsen av CVaR beroende på antal sampel visades vara förhållandevis hög då n = 32, samtidigt som n = 1000 tycktes vara ett tillräckligt högt antal sampel för att minska denna standardavvikelsen markant. Eftersom rapportens omfattning enbart innefattar låga djup, p = [1, 2, 3], och få kvantbitar, $q \leq 18$, kan det vara av intresse att i framtida studier undersöka om denna trend även gäller för större djup och för ett större antal kvantbitar.

Referenser

- A. Robert, P. K. Barkoutsos, S. Woerner och I. Tavernelli, "Resource-efficient quantum algorithm for protein folding," *npj Quantum Information*, årg. 7, nr 1, s. 38, febr. 2021, ISSN: 2056-6387. DOI: 10.1038/s41534-021-00368-4.
- [2] S. Chen, "Quantum Computing Scientists: Give Them Lemons, They'll Make Lemonade," APS News, årg. 28, nr 5, maj 2019. URL: https://www.aps.org/publications/apsnews/ 201905/quantum.cfm.
- [3] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, årg. 2, s. 79, aug. 2018, ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79.
- [4] M. Cerezo, A. Arrasmith, R. Babbush m. fl., "Variational quantum algorithms," *Nature Reviews Physics*, årg. 3, nr 9, s. 625–644, 2021, ISSN: 2522-5820. DOI: 10.1038/s42254-021-00348-9.
- [5] N. Moll, P. Barkoutsos, L. S. Bishop m. fl., "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, årg. 3, nr 3, s. 30503, juni 2018. DOI: 10.1088/2058-9565/aab822.
- [6] J. 'Chow, E. 'Greplová, F. 'Heijman m. fl., "State of Quantum Computing: Building a Quantum Economy," World Economic Forum, tekn. rapport, aug. 2022. URL: https://www3.weforum.org/docs/WEF_State_of_Quantum_Computing_2022.pdf.
- [7] Chalmers Tekniska Högskola, WACQT, maj 2023. URL: https://www.chalmers.se/ centrum/wacqt/.
- [8] M. A. Nielsen och I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, juni 2012, ISBN: 9781107002173. DOI: 10.1017/CB09780511976667.
- [9] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, årg. 21, nr 6-7, s. 467–488, juni 1982, ISSN: 0020-7748. DOI: 10.1007/BF02650179.
- [10] I. Roos, New Swedish quantum computer to be made available to industry, jan. 2023. URL: https://www.chalmers.se/en/current/news/wacqt-new-swedish-quantumcomputer-to-be-made-available-to-industry/.
- P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," i *Proceedings 35th Annual Symposium on Foundations of Computer Science*, IEEE Comput. Soc. Press, 1994, s. 124–134, ISBN: 0-8186-6580-7. DOI: 10.1109/SFCS.1994.365700.
- [12] I. Al Barazanchi, S. A. Shawkat, M. H. Hameed och K. S. Lateef Al-badri, "Modified RSAbased algorithm: a double secure approach," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, årg. 17, nr 6, s. 2818, dec. 2019, ISSN: 2302-9293. DOI: 10.12928/ telkomnika.v17i6.13201.
- [13] E. Rieffel och W. Polak, *Quantum Computing: A Gentle Introduction*, 1. utg. Cambridge: MIT Press, 2011, ISBN: 978-0-262-01506-6.
- [14] D. S. Johnson, "A Catalog of Complexity Classes," i *Algorithms and Complexity*, Elsevier, 1990, s. 67–161. DOI: 10.1016/B978-0-444-88071-0.50007-2.
- [15] V. Kann, "On the approximability of NP-complete optimiztion problems," diss., Royal Institute of Technology, Stockholm, 1992, ISBN: 91-7170-082-X.
- [16] J. Nešetřil och D. Turzí'k, "Solving and approximating combinatorial optimization problems (Towards MAX CUT and TSP)," i 1997, s. 70–85. doi: 10.1007/3-540-63774-5{_}98.

- [17] National Academy of Engineering, *Frontiers of Engineering*. Washington, D.C.: National Academies Press, jan. 2019, ISBN: 978-0-309-48750-4. DOI: 10.17226/25333.
- [18] E. Farhi, J. Goldstone och S. Gutmann, "A Quantum Approximate Optimization Algorithm," *arXiv: Quantum Physics*, 2014. DOI: 10.48550/arXiv.1411.4028.
- [19] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli och S. Woerner, "Improving Variational Quantum Optimization using CVaR," *Quantum*, årg. 4, s. 256, april 2020, ISSN: 2521-327X. DOI: 10.22331/q-2020-04-20-256.
- [20] D. J. Griffiths och D. F. Schroeter, *Introduction to Quantum Mechanics*. Cambridge University Press, aug. 2018, ISBN: 9781316995433. DOI: 10.1017/9781316995433.
- [21] A. Fraenkel, "Complexity of protein folding," *Bulletin of Mathematical Biology*, årg. 55, nr 6, s. 1199–1210, nov. 1993, ISSN: 0092-8240. DOI: 10.1016/S0092-8240(05)80170-3.
- [22] B. Berger och T. Leighton, "Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete.," *Journal of computational biology : a journal of computational molecular cell biology*, årg. 5, nr 1, s. 27–40, 1998, ISSN: 1066-5277. DOI: 10.1089/cmb.1998.5.27.
- [23] E. Callaway, "'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures," *Nature*, årg. 588, nr 7837, s. 203–204, dec. 2020, ISSN: 0028-0836. DOI: 10.1038/d41586-020-03348-4.
- [24] F. Koser, C. Loescher och W. A. Linke, "Posttranslational modifications of titin from cardiac muscle: how, where, and what for?" *The FEBS Journal*, årg. 286, nr 12, s. 2240–2260, juni 2019, ISSN: 1742-464X. DOI: 10.1111/febs.14854.
- [25] R. Zwanzig, A. Szabo och B. Bagchi, "Levinthal's paradox.," *Proceedings of the National Academy of Sciences*, årg. 89, nr 1, s. 20–22, jan. 1992, ISSN: 0027-8424. DOI: 10.1073/pnas. 89.1.20.
- [26] K. F. Lau och K. A. Dill, "A lattice statistical mechanics model of the conformational and sequence spaces of proteins," *Macromolecules*, årg. 22, nr 10, s. 3986–3997, okt. 1989, ISSN: 0024-9297. DOI: 10.1021/ma00200a030.
- [27] H. J. Greenberg, W. E. Hart och G. Lancia, "Opportunities for Combinatorial Optimization in Computational Biology," *INFORMS Journal on Computing*, årg. 16, nr 3, s. 211–231, aug. 2004, ISSN: 1091-9856. DOI: 10.1287/ijoc.1040.0073.
- [28] A. Irbäck, L. Knuthson, S. Mohanty och C. Peterson, "Folding lattice proteins with quantum annealing," *Physical Review Research*, årg. 4, nr 4, okt. 2022, ISSN: 2643-1564. DOI: 10.1103/ PhysRevResearch.4.043013.
- [29] F. Scholtz, H. Geyer och F. Hahne, "Quasi-Hermitian operators in quantum mechanics and the variational principle," *Annals of Physics*, arg. 213, nr 1, s. 74–101, jan. 1992, ISSN: 00034916. DOI: 10.1016/0003-4916(92)90284-S.
- [30] A. Anand Kumar, *Fundamentals of Digital Circuits*, 4. utg. Delhi: PHI Learning Private Limited, juli 2016, s. 132–136, ISBN: 978-81-203-5268-1.
- [31] E. Lucero, M. Hofheinz, M. Ansmann m. fl., "High-Fidelity Gates in a Single Josephson Qubit," *Physical Review Letters*, årg. 100, nr 24, s. 247 001–247 004, juni 2008, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.100.247001.
- [32] D. J. Shepherd, "On the Role of Hadamard Gates in Quantum Circuits," *Quantum Information Processing*, årg. 5, nr 3, s. 161–177, juni 2006, ISSN: 1570-0755. DOI: 10.1007/s11128-006-0023-4.

- [33] T. Albash och D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, årg. 90, nr 1, s. 015 002, jan. 2018, ISSN: 0034-6861. DOI: 10.1103/RevModPhys.90.015002.
- [34] C. C. McGeoch, "Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice," Synthesis Lectures on Quantum Computing, årg. 5, nr 2, s. 1–93, juli 2014, ISSN: 1945-9726. DOI: 10.2200/S00585ED1V01Y201407QMC008.
- [35] M. H. S. Amin, "Consistency of the Adiabatic Theorem," *Physical Review Letters*, årg. 102, nr 22, s. 220 401, juni 2009, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.102.220401.
- [36] E. Farhi, J. Goldstone, S. Gutmann och M. Sipser, "Quantum Computation by Adiabatic Evolution," jan. 2000. doi: 10.48550/arXiv.quant-ph/0001106.
- [37] P. Vikstål, "Continuous-variable quantum annealing with superconducting circuits," diss., 2018. URL: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1254276& dswid=-360.
- [38] M. Suzuki, "Improved Trotter-like formula," *Physics Letters A*, årg. 180, nr 3, s. 232–234, sept. 1993, ISSN: 0375-9601. DOI: 10.1016/0375-9601(93)90701-Z.
- [39] Y. Pan, Y. Tong, S. Xue och G. Zhang, "Efficient Depth Selection for the Implementation of Noisy Quantum Approximate Optimization Algorithm," juli 2022. DOI: 10.48550/arXiv. 2207.04263.
- [40] H. Wu, R. Yang, Q. Fu, J. Chen, W. Lu och H. Li, "Research on predicting 2D-HP protein folding using reinforcement learning with full state space," *BMC Bioinformatics*, årg. 20, nr S25, s. 685, dec. 2019, ISSN: 1471-2105. DOI: 10.1186/s12859-019-3259-6.
- [41] B. A. Cipra, "An Introduction to the Ising Model," *The American Mathematical Monthly*, *ärg.* 94, nr 10, s. 937–959, dec. 1987, ISSN: 0002-9890. DOI: 10.1080/00029890.1987. 12000742.
- [42] S. G. Brush, "History of the Lenz-Ising Model," *Reviews of Modern Physics*, årg. 39, nr 4, s. 883–893, okt. 1967, ISSN: 0034-6861. DOI: 10.1103/RevModPhys.39.883.
- [43] A. Irbäck och C. Troein, "Enumerating Designing Sequences in the HP Model," *Journal of Biological Physics*, årg. 28, nr 1, s. 1–15, 2002, ISSN: 00920606. DOI: 10.1023/A:1016225010659.
- [44] Testing CVaR on QAOA, maj 2023. URL: https://github.com/jacobljungberg/ Testing_CVaR_on_QAOA.git.
- [45] *aqpl-mc2-chalmers/vqa*, 2023. URL: https://github.com/aqpl-mc2-chalmers/vqa.
- [46] V. Bergholm, J. Izaac, M. Schuld m. fl., "PennyLane: Automatic differentiation of hybrid quantumclassical computations," nov. 2018. doi: 10.48550/arXiv.1811.04968.
- [47] P. Virtanen, R. Gommers, T. E. Oliphant m. fl., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, årg. 17, nr 3, mars 2020, ISSN: 1548-7091. DOI: 10.1038/s41592-019-0686-2.
- [48] K. Chandra, A. Xie, J. Ragan-Kelley och E. Meijer, "Gradient Descent: The Ultimate Optimizer," sept. 2019. doi: 10.48550/arXiv.1909.13371.
- [49] M. Alam, A. Ash-Saki och S. Ghosh, "Accelerating Quantum Approximate Optimization Algorithm using Machine Learning," febr. 2020. DOI: 10.48550/arXiv.2002.01089.
- [50] S. Brandhofer, D. Braun, V. Dehn m. fl., "Benchmarking the performance of portfolio optimization with QAOA," *Quantum Information Processing*, årg. 22, nr 1, s. 25, dec. 2022, ISSN: 1573-1332. DOI: 10.1007/s11128-022-03766-5.

- [51] The GPyOpt authors, *GPyOpt: A Bayesian Optimization framework in Python*, 2016. URL: http://github.com/SheffieldML/GPyOpt.
- [52] T. Häner, M. Roetteler och K. M. Svore, "Factoring using 2n+2 qubits with Toffoli based modular multiplication," nov. 2016. DOI: 10.48550/arXiv.1611.07995.
- [53] Xin Zhou och Xiaofei Tang, "Research and implementation of RSA algorithm for encryption and decryption," i *Proceedings of 2011 6th International Forum on Strategic Technology*, IEEE, aug. 2011, s. 1118–1121, ISBN: 978-1-4577-0398-0. DOI: 10.1109/IFOST.2011.6021216.

A Transformering av Hamiltionian till Ising-modellen

För att härleda fram den faktiskta Hamiltonianen som används i programkoden börjar uträkningen med den faktiska energitermen:

$$\begin{split} E &= -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s, s' \rangle} x_s^f x_{s'}^{f'} \\ &+ \lambda_1 \left[\sum_f \left(\sum_s x_s^f - 1 \right)^2 + \sum_{f'} \left(\sum_{s'} x_{s'}^{f'} - 1 \right)^2 \right] \\ &+ \lambda_2 \left[\frac{1}{2} \sum_{f_1 \neq f_2} \sum_s x_s^{f_1} x_s^{f_2} + \frac{1}{2} \sum_{f'_1 \neq f'_2} \sum_{s'} x_{s'}^{f'_1} x_{s'}^{f'_2} \right] \\ &+ \lambda_3 \left[\sum_{1 \le f < N} \sum_s x_s^f \sum_{||s'-s|| > 1} x_{s'}^{f+1} + \sum_{1 \le f < N} \sum_{s'} x_{s'}^{f'} \sum_{||s-s'|| > 1} x_{s'}^{f'+1} \right]. \end{split}$$
(A.1)

Variablerna behövs omvandlas för att passa in i Ising-modellen då den använder sig av variablerna -1, +1 istället för 0 och 1 som annars är praxis

$$x_s^f \to \{0,1\}, \qquad \sigma_s^f \to \{-1,+1\},$$

där variabelbytet av x_s^f uppnås enligt

$$x_s^f = \frac{1 - \sigma_s^f}{2}.$$

För övrigt är

$$C(h_f, h_{f'}) = \begin{cases} 1, \text{ om } h_f = h_{f'} = H\\ 0, \text{ annars,} \end{cases}$$

vilket ger Hamiltonianen:

$$\begin{split} E(\sigma) &= \hat{H}_{cost}(\sigma) = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s, s' \rangle} \frac{1 - \sigma_s^f}{2} \frac{1 - \sigma_{s'}^{f'}}{2} \\ &+ \lambda_1 \left[\sum_f \left(\sum_s \frac{1 - \sigma_s^f}{2} - 1 \right)^2 + \sum_{f'} \left(\sum_{s'} \frac{1 - \sigma_{s'}^{f'}}{2} - 1 \right)^2 \right] \\ &+ \lambda_2 \left[\frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \frac{1 - \sigma_s^{f_1}}{2} \frac{1 - \sigma_s^{f_2}}{2} + \frac{1}{2} \sum_{f_1' \neq f_2'} \sum_{s'} \frac{1 - \sigma_{s'}^{f'_1}}{2} \frac{1 - \sigma_{s'}^{f'_2}}{2} \right] \\ &+ \lambda_3 \left[\sum_{1 \le f < N} \sum_s \frac{1 - \sigma_s^f}{2} \sum_{||s'-s||>1} \frac{1 - \sigma_{s'}^{f+1}}{2} + \sum_{1 \le f < N} \sum_{s'} \frac{1 - \sigma_{s'}^{f'_2}}{2} \sum_{||s-s'||>1} \frac{1 - \sigma_{s'}^{f'+1}}{2} \right] \end{split}$$
(A.2)

eller

$$\begin{split} E(\sigma) &= \hat{H}_{cost}(\sigma) = -\sum_{|f-f'|>1} C(h_f, h_{f'}) \sum_{\langle s, s' \rangle} \frac{1 - \sigma_s^f - \sigma_{s'}^{f'} + \sigma_s^f \sigma_{s'}^{f'}}{4} \\ &+ \lambda_1 \left[\sum_f \left(\sum_s \frac{-\sigma_s^f - 1}{2} \right)^2 + \sum_{f'} \left(\sum_{s'} \frac{-\sigma_{s'}^{f'} - 1}{2} \right)^2 \right] \\ &+ \lambda_2 \left[\frac{1}{2} \sum_{f_1 \neq f_2} \sum_s \frac{1 - \sigma_s^{f_1} - \sigma_s^{f_2} + \sigma_s^{f_2} \sigma_s^{f_1}}{4} + \frac{1}{2} \sum_{f'_1 \neq f'_2} \sum_{s'} \frac{1 - \sigma_{s'}^{f'_1} - \sigma_{s'}^{f'_2} + \sigma_{s'}^{f'_2} \sigma_{s'}^{f'_1}}{4} \right] \\ &+ \lambda_3 \left[\sum_{1 \le f < N} \sum_s \frac{1 - \sigma_s^f}{2} \sum_{||s' - s|| > 1} \frac{1 - \sigma_{s'}^{f+1}}{2} + \sum_{1 \le f < N} \sum_{s'} \frac{1 - \sigma_{s'}^{f'}}{2} \sum_{||s - s'|| > 1} \frac{1 - \sigma_{s'}^{f'+1}}{2} \right]. \end{split}$$
(A.3)

Denna ekvation programmeras sedan in i en enkropps- och tvåkroppsterm i en Hamiltonian i Penny-Lane som appliceras på aminosyrakedjan.

