

Catalogue of Quantum Algorithms for Artificial Intelligence

Leo Sünkel, Thomas Gabor
(QAR-Lab, LMU Munich)



Gefördert durch:



Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages

Quantum Recommendation System

The goal of recommendation systems is to give personal recommendations to users based on their previous actions like purchases or ratings. One assumption is that users can be categorized into different 'types', that is, classes of individuals with shared characteristics such as interests or likes and dislikes. Users from the same type will most likely prefer similar products and thus using this information, the system is able to provide appropriate recommendations for individual users.

A quantum algorithm proposed by Kerenidis and Prakash for this task works as follows. There exists a $m \times n$ matrix P which holds information on user preferences, where m are the users and n the products. An element of this matrix encodes whether a particular user likes or dislikes a particular product. The problem, however, is that the matrix P is unknown or incomplete. Instead of computing the complete matrix P , a time-consuming task, a low-rank approximation matrix is constructed which is then used in the recommendation process. A quantum computer is then used to sample the low-rank approximation matrix in order to get a recommendation.

Up until recently it was thought that this quantum algorithm provides an exponential speedup when compared to the best classical algorithms. However, in the 2018 paper *A quantum-inspired classical algorithm for recommendation systems* by Ewin Tang, a classical algorithm analogous to the quantum algorithm was described that is only polynomially slower than the quantum recommendation system. Thus, the quantum recommendation system no longer provides an exponential speedup to the best classical algorithms.

Quantum computing model: -

Paper: Quantum Recommendation Systems, Iordanis Kerenidis and Anupam Prakash, 2016

Problem type: Recommendation

Implementation available: -

Quantum Autoencoder

Autoencoders are a form of neural network that can be used for dimensionality reduction, i.e., autoencoders learn to represent data in a lower dimensional space. The two main building blocks are the encoder and decoder. The encoder tries to learn a low dimensional encoding for the data while the decoder attempts to reconstruct the original data from the encoding.

A quantum autoencoder could for example be used to compress quantum data. Quantum simulations and state preparation are also applicable domains. State preparation is used as part of other quantum algorithms. There exist several implementation approaches for quantum autoencoders and a selection is described next.

Quantum Autoencoder 1

Quantum computing model: Gate model

Paper: Quantum autoencoders for efficient compression of quantum data, Jonathan Romero, Jonathan P. Olson and Alan Aspuru-Guzik, 2017

Description: In this quantum-gate model approach, the autoencoder is implemented via a programmable circuit that consists of a fixed number of gates each with corresponding parameters. The neural network is then trained via a quantum-classical hybrid approach. State preparation and measurement is done by the quantum computer. Cost computation and optimization is performed by a classical computer to produce new parameters. The resulting parameters are then used as input for the quantum computer and the process repeats until convergence.

Implementation available: -

Quantum Autoencoder 2

Quantum computing model: Gate model

Paper: Quantum generalisation of feedforward neural networks, Kwok Ho Wan, Oscar Dahlsten, Hler Kristjansson, Robert Gardner and M.S. Kim, 2016

Problem type: Classification

Description: In this approach, a classical neural network is generalised to a quantum neural network that is able to compress quantum states and thus can be considered a quantum autoencoder. The neural network is implemented as a quantum circuit and is trained by the use of the optimization method gradient descent. Note that the parameters are classic and thus the

authors term their neural network a 'quantum neural network with classical learning parameters'. They furthermore note that this quantum neural network can perform quantum tasks which a classical network cannot.

Quantum Support Vector Machine

A support vector machine (SVM) is a binary classification algorithm well known in the machine learning community. It has a wide range of applications such as pattern recognition for example. SVMs must be trained on a training data set in order to learn to correctly classify new or unseen data. In particular, the SVM learns to construct a so-called separating hyperplane that separates instances of both classes while attempting to make the margin between instances and the hyperplane as large as possible. Instances on one side of the hyperplane belong to a particular class while the instances on the other side belong a different class.

Quantum Support Vector Machine Algorithm 1

Quantum computing model: Quantum gate model

Paper: Supervised learning with quantum enhanced features spaces, Vojtech Havlicek, Antonio D. Corcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow and Jay M. Gambetta, 2018

Description: Havlicek et al. show that a quantum version of the SVM can be implemented in the following way. Two distinct approaches are available for this problem. The first method uses a variational circuit to compute the separating hyperplane while the second method estimates the kernel function in order to optimize the classifier directly. The latter method is then used as part of a conventional SVM. In both methods the data is provided classically while the quantum state space is used as the feature space. It is furthermore noted that in order to obtain a quantum advantage, the kernel cannot be estimated classically, i.e., if the kernel is too simple, the quantum SVM does not show a quantum advantage over a normal SVM. [*Supervised learning with quantum enhanced feature spaces, Havlicek et al.*]

Implementation available: Qiskit

Quantum Support Vector Machine Algorithm 2

Quantum computing model: Quantum annealing

Paper: Support vector machines on the D-Wave quantum annealer, D. Willsch, M. Willsch, H. De Raedt and K. Michielsen, 2019

Problem type: Classification

Description: The training of quantum support vector machines (QSVM) can also be run on a quantum annealer, as demonstrated by Willsch et al. in a recent paper. To achieve this, the problem, like any problem solved via quantum annealing (QA), must first be formulated as a

QUBO. However, the training of SVMs entails solving equations that contain real numbers whereas a QUBO consists of binary values. Willsch et al. use a special encoding to overcome this and thus are able to formulate the problem as a QUBO.

Willsch et al. investigate the performance of their QSVM on a DW2000Q quantum annealer. They note that the quantum annealer returns in addition to the global optimum, a range of solutions that are close to the optimal. They furthermore note that this is advantageous as the generalization ability may potentially be improved by using a combination of the produced solutions.

In summary, a QSVM can be trained using via QA by formulating the problem as a QUBO. A QA device, such as the DW2000Q, produces optimal and near-optimal solutions and a combination of these solutions can potentially improve the generalization behaviour.

Quantum Generative Adversarial Network

Generative adversarial networks (GANs) are a relatively new form of neural networks that are used to generate new data of a particular form and domain. This data can for example be in the form of text, images or sound. For instance, a GAN could be fed with images of a certain style and the GAN would then attempt to create new images with that exact style. The main goal of a GAN is to create new data that is indistinguishable from real data. To achieve this, GANs consist of two neural networks, a generator and a discriminator. The generator is used to create new data while the discriminator attempts to distinguish between fake, i.e., generated data and real data. During training, the generator learns to generate new data so that the discriminator no longer knows what is real and what is fake. Learning is gradient-based, that is, the generator uses the gradients from the discriminator in order to learn and produce better results. In the next section a selection of quantum generative adversarial networks is discussed.

References:

[1] Quantum generative adversarial networks, Pierre-Luc Dallaire-Demers and Nathan Killoran, 2018

Quantum Generative Adversarial Network 1

Quantum computing model: Gate model

Paper: Quantum generative adversarial networks, Pierre-Luc Dallaire-Demers and Nathan Killoran, 2018

Description: GANs can also be implemented on a quantum computer and one approach proposed by Dallaire-Demers and Killoran works as follows. The generator and discriminator are both implemented as variational quantum circuits, i.e., quantum circuits that take as input a set of parameters. In addition to these circuits, there is a circuit that computes the gradients needed in the learning process. The gradients are used to update the parameters via gradient-descent, which is run on a classical computer, making this a hybrid approach.

In summary, both the generator and discriminator are implemented as quantum circuits running on a quantum computer. The gradients used for learning are also computed via a quantum circuit. However, the parameters for the circuits are updated via gradient descent on a classical computer. Thus, this quantum generative adversarial neural network follows a hybrid approach.

Implementation available: PennyLane

Quantum Generative Adversarial Network 2

Quantum computing model: Gate model

Paper: Quantum Generative Adversarial Networks for Learning and Loading Random Distributions, Christa Zoufal, Aurelien Lucchi and Stefan Woerner, 2019

Description: Quantum generative adversarial networks (qGANs) have many use cases and can for example be applied to quantum state preparation. More specifically, Zoufal et al. show that a qGAN can be used to learn and load random distributions into quantum states. This technique requires less gates than the current state-of-the-art and is thus more efficient. Note that this method can be implemented using the quantum gate model and therefore can be used as part of other quantum gate algorithms. Algorithms that require an efficient state preparation technique in order to provide a quantum advantage could make use of this proposed technique.

The proposed qGAN implements a quantum generator and a classical discriminator, making this a hybrid algorithm. More specifically, the generator is implemented as a quantum variational circuit. Optimization of parameters can be achieved via gradient-based methods. The discriminator is simply a classical neural network.

To summarize, this qGAN is able to learn and load random distributions into quantum states in an efficient manner. The qGAN can be therefore used as a technique for state preparation as part of other algorithms, allowing the possibility to exploit a quantum advantage in said algorithms.

Implementation available: Qiskit

Quantum Reinforcement Learning

Reinforcement learning (RL) is a form of machine learning (ML) in which an agent inhabits some environment and whose goal is to maximize its rewards according to some objective function. An agent basically learns through trial and error by exploring different strategies.

RL can be formulated as a Markov decision process (MDP) and consists of the following components: a set of states, a set of actions, probabilities for state transition, a reward function and an objective function. The strategy that determines what actions to take in what situations is known in RL as a policy. The goal in RL, then, is to learn a policy that maximizes the reward.

Several approaches that attempt to combine RL with quantum computing have been suggested over the last few years. These methods include quantum RL algorithms as well as quantum-enhanced RL algorithms. Both quantum gate and quantum annealing based quantum computers have been investigated.

References:

- [1] Quantum Reinforcement Learning, Daoyi Dong, Chunlin Chen, Hanxiong Li, Tzyh-Jong Tarn, 2008
- [2] Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces, Florian Neukart, David Von Dollen, Christian Seidel, Gabriele Compostella, 2017

Quantum Reinforcement Learning Algorithm 1

Quantum computing model: Gate model

Paper: Quantum Reinforcement Learning, Daoyi Dong, Chunlin Chen, Hanxiong Li, Tzyh-Jong Tarn, 2008

Problem type: Reinforcement learning

Description: With this algorithm, Dong et al. introduce an approach to quantum reinforcement learning (QRL) that takes advantage of effects from quantum physics and works fundamentally different than any classical RL method, however, some similarities still remain. For example, QRL, like classical RL methods, also contain a policy, reward function and an environment. However, Dong et al. note that their QRL algorithm differs to classical RL algorithms in intrinsic parts like representation, policy, parallelism and update operation. States and actions are also different in both approaches. In this QRL method, states are referred to as eigen states and actions as eigen actions and are able to be in a superposition state. Superposition allows the algorithm, among other things, to better balance exploration and exploitation.

Recall that in quantum physics, whenever a qubit in superposition is measured, it collapses and takes on one state according to some probability. The algorithm takes advantage of this

behaviour in the action selection policy. More specifically, an action is measured in relation to some state and hence collapses to one of its eigen actions according to some probability this action is then selected. This means that the probability of actions that are considered good should be amplified. The probability amplitudes must be updated throughout the algorithm. The method to update the probability amplitude is based on the Grover iteration from Grover's algorithm, a famous quantum algorithm for database search. The method contains an oracle or black box that is used to tell whether an action is good or bad.

Loosely formulated, the complete algorithm works as follows. The first step is to initialize the state and action. After this an action is observed and executed to receive the next state and reward. Then the state value and probability amplitudes are updated accordingly. The probability amplitudes are updated in such a way that the probability for good actions is amplified and shrunk for bad ones. This process is done repeatedly. And so, after a number of episodes, the algorithm is able to learn a policy.

Quantum Reinforcement Learning Algorithm 2

Quantum computing model: Quantum annealing

Paper: Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces, Florian Neukart, David Von Dollen, Christian Seidel, Gabriele Compostella, 2017

Problem type: Reinforcement learning

Description: The methods discussed here use a quantum annealing device to solve specific aspects of a reinforcement learning algorithm and can be termed quantum-enhanced reinforcement learning. The authors propose a technique that allows a D-Wave quantum annealing device to be used for finding an optimal policy in a RL setting. In addition to this, they show how a quantum annealer can be used to approximate an improved state value function.

State-value function approximation is done as follows. First n sub-optimal state-value functions are generated. State-value functions can for example be generated by classical Monte Carlo policy evaluation. These sub-optimal state-value functions are then embedded on the QPU in order to approximate an improved state-value function.

In order to learn a policy, Monte Carlo policy iteration is used. This entails choosing random actions and evaluating the reward. This Monte Carlo policy iteration is then partially embedded onto the QPU.

In order to solve these problems via quantum annealing, both problems are formulated as a QUBO and embedded on the D-Wave quantum processing unit.

Quantum Boltzmann Machine

A Boltzmann machine (BM) is a machine learning (ML) algorithm that can be applied to a number of problems that arise in the domain. It consists of a range of connected stochastic binary units that together form a network. These units, sometimes also referred to as neurons, can hold either the value 0 or 1, denoting whether the unit is 'on' or 'off'. Being stochastic, a unit takes on a value according to some probability. These units are further grouped into two distinct groups, the hidden and visible units. While hidden units can be considered free, visible units are bound and are 'clamped' into states by the environment. Units can furthermore be connected by edges with an associated weight. Lastly, Boltzmann machines include a energy function that describes the energy of the system. For every possible configuration of states, the energy function describes the corresponding energy. A BM learns through updating its weights and biases.

A quantum computer can be applied to Boltzmann machines in a number of ways. For example, a quantum computer may be used to train a BM. Several approaches using a quantum computer for training QBM have been investigated in recent years and a selection is discussed next.

References:

[1] Ackley David, Hinton Geoffrey E. and Sejnowski Terrence J., A Learning Algorithm For Boltzmann Machines, Cognitive Science, 1985

Quantum Boltzmann Machine Algorithm 1

Quantum computing model: Quantum annealing

Paper: Quantum Boltzmann Machine, Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, Roger Melko, 2016

Problem type: Boltzmann machine

Description: Amin et al. investigate the possibility of a quantum Boltzmann machine (QBM) in their paper. In this approach, the energy function of the BM is replaced by a Hamiltonian and the classical bits become qubits. The quantum probabilities are bounded and the BM is trained via sampling. This QBM not only takes advantage of the quantum domain for training, but also in its model. More specifically, the QBM is based on a quantum Boltzmann distribution rather than a classical one. Lastly, a quantum annealing (QA) device can be used to train such a QBM. However, it is noted that current QA machines, such as the ones provided by D-Wave, require some modifications to the hardware in order to be used for such a task.

Quantum Boltzmann Machine Algorithm 2

Quantum computing model: Gate model

Paper: Generative training of quantum Boltzmann machines with hidden units, Nathan Wiebe, Leonard Wossnig, 2019

Problem type: Boltzmann machine

Description: The training of quantum Boltzmann machines (QBMs) is examined by Wiebe et al. in a recent paper. The authors propose two different methods to train a QBM. With the first method, a variational bound on the quantum relative entropy is optimized, which allows to easily compute the derivatives. However, this method assumes a specific form of Hamiltonian and corresponding terms. This means that this particular approach is less general than the second, however, it is the more efficient one. In the second approach, the exact expression for the gradient is approximated and is a general method. In particular, the second approach incorporates Fourier series approximations and high-order divided difference formulas.

In summary, Wiebe et al. introduce two new ways to train QBMs, the first method being more efficient while the second is a more general method. The authors note that both approaches are efficient when Gibbs state preparation is efficient and furthermore that finding better ways for this task will be important to be able to train QBMs on near term quantum computers in a practical way.

Quantum Approximate Optimization Algorithm

Quantum computing model: Gate model

Paper: A Quantum Approximate Optimization Algorithm, Edward Farhi, Jeffrey Goldstone, Sam Gutman, 2014

Problem type: Combinatorial optimization

Description: The quantum approximate optimization algorithm (QAOA) is used to solve combinatorial optimization problems. The algorithm does not necessarily find the optimal solution, but rather attempts to find an approximate solution that is close to the optimal. In many cases a sub optimal solution is good enough.

The algorithm works as follows. Given an integer p and a set of initial angles, the quantum computer will create the initial quantum state. Measuring this state in the computational basis results in a string z which is subsequently evaluated according to the objective function. This is repeated and eventually a good approximation is found. However, it can be difficult to find sets of angles and an appropriate strategy is needed. For example, angles can be calculated on a classical computer and then given to the quantum computer, following a quantum-classical approach. The parameter p is crucial in this algorithm as a higher p value also results in an improved quality of the approximation.

Quantum Clustering Algorithms

Clustering algorithms belong to the class of unsupervised learning and are crucial in data mining scenarios. In unsupervised learning, as opposed to supervised learning, there is no training set consisting of data points with their assigned labels and the number of classes might also be unknown. Instead, unsupervised learning algorithms attempt to find underlying, i.e., hidden structure in the given data.

Clustering algorithms are a prominent member of the unsupervised learning class and a number of different approaches and algorithms have been proposed over the years. These methods include partitioning clustering methods, density-based clustering, hierarchical clustering and others. A well-known partitioning clustering algorithm is k-means while DBSCAN is a widely known density-based clustering algorithm. The task of clustering basically entails the partitioning of similar objects into groups while separating dissimilar ones. To determine which objects are similar a similarity measure is needed. A distance like the Euclidean-distance can for example be used as a similarity measure. However, other similarity measures can be applied.

Extending the field to the realm of quantum computing may allow to extract a possible quantum advantage by improving existing algorithms or devising completely new ones.

References:

- [1] Data Mining: Concepts and Techniques, Third edition, Jiawei Han, Micheline Kamber, Jian Pei
- [2] Quantum Clustering Algorithms, Esma Aimeur, Gilles Brassard, Sebastien Gambs, 2007

Quantum Clustering Algorithm 1

Quantum computing model: Gate model

Paper: Quantum Clustering Algorithms, Esma Aimeur, Gilles Brassard, Sebastien Gambs, 2007

Problem type: Unsupervised learning / Clustering

Description: The algorithms presented in the paper by Aimeur et al. improve classical clustering algorithms by incorporating features offered by quantum computation. To this end, the authors present three what they call quantum subroutines which are used to improve classical clustering algorithms. These quantum subroutines are based upon Grover's algorithm. Furthermore, an oracle is used to determine the distance between objects.

The first subroutine is used to find the two points that are the farthest apart. The algorithm starts by selecting two random indexes and the max distance is initialised to the distance between these two points. The next step is to apply Grover's algorithm in order to obtain a new set of points whose distance is larger than the previous max distance. The process repeats until no two

points with a larger distance are found. The second subroutine, which is inspired by an algorithm that can be used to find the c smallest values of a function, is used to find the c closest neighbours to any given data point. Lastly, a subroutine is presented that is used to calculate the median of a set of points. In order to calculate the median, one can calculate for each point the sum distance to all other points. The minimum is then chosen. A black box is used to calculate the distance between points. A further quantum algorithm to find the minimum sum can be then applied.

Using these subroutines, the authors propose to improve three classical clustering algorithms. In particular, they show how these subroutines can be applied to divisive clustering, k -medians and the construction of a c -neighbourhood graph. Divisive clustering is a form of hierarchical clustering, k -medians is a clustering algorithm similar to k -means and a c -neighbourhood graph is used as part of other clustering algorithms.

Quantum Clustering Algorithm 2

Quantum computing model: Adiabatic

Paper: Quantum algorithms for supervised and unsupervised machine learning, Seth Lloyd, Masoud Mohseni, Patrick Rebentrost, 2013

Problem type: Clustering

Description: Lloyd et al. present a technique that uses a quantum computer to improve both supervised and unsupervised learning clustering algorithms. Their algorithms make use of a quantum random access memory (QRAM) which allows for quantum parallel data access. The presented techniques take further advantage of the fact that estimating distances in large vector spaces can be accomplished much faster with a quantum computer than with a classical one.

The first algorithm presented is an algorithm for supervised cluster assignment, i.e. given a vector, assign it to one out of two sets. For each set there exist a number of representative samples. In other words, a clustering already exists and the task is to assign a new vector to one of the two sets. This algorithm is then used as a subroutine in the second algorithm which is a quantum version of the unsupervised clustering algorithm k -means. Furthermore, the authors show that one can reformulate the problem so that it can be solved via the adiabatic algorithm and thereby improving it even further.

Quantum Classification Algorithms

Classification algorithms are a form of supervised learning which means that the class labels for elements in the training set are given and are used during the training process. The goal of a classification algorithm is to predict the correct class label of an object whose label is unknown. In order to achieve this, a model is first built through training using objects and their labels from the training set. The model is able to distinguish data classes and concepts. Once the classifier is built, it is used to classify data objects with unknown labels. Classification algorithms can be applied to many areas such as handwritten digit recognition or to determine whether a bank customer should be allowed to receive a loan. Examples of classification algorithms are nearest-neighbour classification, decision tree and naïve Bayesian classification.

References:

- [1] Jiawei Han, Micheline Kamber and Jian Pei, Data Mining Concepts and Techniques, Third edition, 2012
- [2] Nathan Wiebe, Ashish Kapoor and Krysta M. Svore, Quantum Algorithms for Nearest-Neighbour Methods for Supervised and Unsupervised Learning, 2014

Quantum Classification Algorithm

Quantum computing model: Gate

Paper: Quantum Algorithms for Nearest-Neighbour Methods for Supervised and Unsupervised Learning, Nathan Wiebe, Ashish Kapoor and Krysta M. Svore, 2014

Problem type: Classification

Description: Wiebe et al. propose a quantum version of nearest-neighbour classification algorithm. Their algorithm consists of three main phases and uses either an inner product method or a euclidean method to compute the distance between vectors. The three phases of the algorithm are described next.

In phase one, two oracles are employed to prepare states of a specific form. These states encode the distance between the test vector u and a training vector v_j in an amplitude. Note that this is done for every training vector v_j . In phase two, the distance estimate is stored in as a qubit string. This is done via coherent amplitude amplification and avoids measuring the state. The resulting state is again in a specific form. In the last step Grover's search is used in order to find the closest v_j to the test vector u . Wiebe et al. state that the algorithm applies Grover's search in the form of the Dürr Hoyer minimization algorithm.

These are the main three steps of the quantum nearest-neighbour classification algorithm. Note that the distance between two vectors can be calculated by applying either an inner product method or a Euclidean method on a quantum computer.

Evolutionary Algorithms

Evolutionary Algorithms (EAs) define a class of algorithms solving a range of problems by the means of methods and strategies inspired by Darwinian evolution. EAs can be considered a branch of Artificial Intelligence (AI) and problems they are commonly applied to include a range of optimization problems. EAs apply the principle of survival of the fittest to iteratively improve a population of solutions by allowing more suitable solutions to 'reproduce' and thus survive to the next generation. They furthermore apply mutations in order to slightly change a solution. Many variations of EAs exist, including genetic algorithms, evolutionary programming, evolutionary algorithms and others. Here we describe the main ideas of these algorithms and use the term evolutionary algorithms (EAs) without going into details of each specific area.

A typical evolutionary algorithm (EA) consists of the following components: fitness function, solution representation, mutation operations, crossover operations and selection.

The fitness function, sometimes also referred to as the objective function, describes the function that is to be optimized, i.e., minimized or maximized. It can be seen as a function that measures how good (or bad) a solution is. Solution representations are configurations of variables that are a means to describe a solution to a particular problem and come in many shapes and forms like bit-strings, series of integers or binary variables for example. Mutation operations are used to slightly change a given solution while crossover operations are used to create offspring, i.e., create new solutions by combining the solutions of the parents. Selection is the process of selecting individuals from the population that will function as parents and are used to create offspring (solutions). However, selection is also an important at the end of each generation when it is time to select the individuals from the population that will survive to the next generation and those that will be replaced by the newly created solutions.

A basic EA works as follows. At first a initial population is created. Then the following steps are repeatedly done until the maximum number of generations is reached or a termination criterion is met. A number of parents are selected to produce a number of children, that is, new solutions, via crossover operations. The next step is to apply mutations to these newly created children (solutions) according to some probability. In the last step a subset of the population is selected and replaced by the children (solutions). Once selection and replacement is complete, the process repeats. Each of these cycles is known as a generation. Note that this just describes a basic EA and many variations and strategies exist. Also note that it is important to keep the population diverse in order to avoid premature convergence towards a sub-optimal solution.

References:

[1] Evolutionary Computing, A. E. Eiben and M. Schoenauer, 2005

Quantum(-inspired) Evolutionary Algorithm 1

Quantum computing type: None (Quantum-inspired algorithm running on a classical computer)

Paper: Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization, Kuk-Hyun Han and Jong-Hwan Kim, 2002

Problem type: Optimization

Description: Rather than developing algorithms to be run on quantum computers, quantum-inspired algorithms are designed to run on classical computers. While these algorithms solely run on classical machines, they do take advantage from quantum computing as some components are directly inspired by effects seen in quantum computers. Han et al. propose a quantum-inspired evolutionary algorithm, that is, an evolutionary algorithm that contains parts that are directly inspired by quantum computers albeit still running on a classical machine.

In this particular approach, the EA uses so called Q-bits in its solution representation and Q-gates to update Q-bit individuals. A Q-bit individual consists of a number of Q-bits, that is, its a string of Q-bits. Q-bits are obviously inspired by qubits and allow for the elements of a solution to be in superposition. This representation also increases diversity in the population. Q-gates are used to update the Q-bit individuals and function as a variation operator to allow a Q-bit individual to find better solutions. Additionally, a migration process is proposed that increases the diversity of the probabilities for Q-bit individuals.

Quantum Evolutionary Algorithm 2

Quantum computing model: None (Quantum inspired algorithm running on classical computer)

Paper: A novel quantum swarm evolutionary algorithm and its applications, Yan Wang, Xiao-Yue Feng, Yan-Xin Huang, Dong-Bing Pu, Wen-Gang Zhou, Yan-Chun Liang, Chun-Guang Zhou, 2006

Problem type: Optimization

Description: With this quantum swarm evolutionary algorithm, Wang et al. propose an algorithm that is based on the quantum-inspired evolutionary algorithm proposed by Han et al. [1], which is discussed above. In this approach, a quantum bit is expressed as a quantum angle and a particle swarm optimization (PSO) is then used to update the Q-bit automatically. A quantum swarm consisting of all Q-bits is formed. The best local quantum angles are found and from these the global best values. Using these values, the quantum angles are updated through the Q-gate.

The quantum swarm evolutionary algorithm contains the following 3 main steps. First the population of Q-bit individuals is created and quantum angles are used to encode the Q-bits. Next the states of the quantum angles (Q-bits) are observed. Finally, the population of Q-bit individuals is updated via an improved PSO procedure rather than using a traditional Q-gate like in the QEA. Recall that is algorithm is based on the QEA [1].

[1] Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization, Kuk-Hyun Han and Jong-Hwan Kim, 2002

Quantum Evolutionary Algorithm 3

Quantum computing model: Quantum annealing / classical computer hybrid

Paper: Quantum-Assisted Genetic Algorithm, James King, Masoud Mohseni, William Bernoudy, Alexandre Frechette, Hossein Sadeghi, Sergei V. Isakov, Hartmut Neven and Mohammad H. Amin, 2019

Problem type: Optimization

Description: King et al. propose a quantum/classical hybrid genetic algorithm which they call the quantum-assisted genetic algorithm (QAGA). The QAGA uses a basic classical genetic algorithm as baseline while employing quantum reverse annealing as a mutation operator. Crossover and selection are performed by the classical part of the algorithm. In particular, Isoenergetic cluster moves are used as the crossover method while the selection method simply selects the best N individuals for 'survival'. Mutation is the part that is performed by the quantum annealer (QA) which entails mutating a state via reverse annealing. In reverse annealing, the QA is first initialized to a specific state, that is, a classical state in which the values of the qubits are assigned specific values. During the reverse anneal, the state is put into a superposition of states and runs the annealing algorithm. At the end, a classical state is returned. The thereby mutated states are then fed back into the population of the genetic algorithm which can then continue its classical run.

Quantum Annealing

Quantum annealing (QA) [1, 2] is a meta-heuristic algorithm that is considered a form of quantum computing and is applied to combinatorial optimization problems. In particular, QA takes advantage of effects from quantum physics such as superposition and quantum fluctuations in its computation. It is usually employed fixed in hardware using a special machine called Quantum Annealer. Currently, D-Wave (www.dwavesys.com) is the only commercial provider.

In QA, problems are formulated in such a way that the optimal solution corresponds to the system's ground state, i.e, the lowest point of energy. The Ising model is used as a means to describe a particular problem. However, QA devices, such as the ones provided by D-Wave, allow problems also to be formulated in the equivalent QUBO formulation. QA can furthermore be compared to simulated annealing (SA). Briefly speaking, SA is an algorithm proposed by Kirkpatrick et al. [3] and is used to solve optimization problems. It uses an energy value as a cost function and the goal here, too, is to find the state that represents the minimum energy. However, SA uses thermal fluctuations rather than quantum fluctuations as state transition. The difference is illustrated in the following example. Consider an optimization problem with many local minima and a difficult to find global minimum. While a classical algorithm will attempt to 'climb' out of local minima, QA makes use of quantum fluctuations, or quantum tunnelling, to escape the local minima in a more efficient manner, leading to a potential speed-up in computation.

The QA algorithm runs as follows. In the beginning, the ground state of a Hamiltonian is prepared and functions as the initial state. Note that a Hamiltonian in this context can be seen as a cost function that returns the energy value of a particular solution. The system then evolves following the Schrödinger equation and hopefully remains the low energy state. During the annealing process, the qubits are put into superposition. At the end, when the qubits are measured, the qubits collapse and take on classical states and the system should return the ground state and thus the solution.

In summary, QA is an algorithm that is applied to a specific class of problems, namely combinatorial optimization problems. QA is a form of quantum computing and thus takes advantage of quantum physics in its computation. Problems that should be solved via QA must first be formulated as Ising or QUBO problems. Any problem that can be recast as an Ising/QUBO can potentially be solved via QA, which includes a wide range of problems. These formulations return an energy value for every possible configuration/solution and the optimal solution is the one with the lowest energy, also known as the ground state. Thus, the goal of QA is to find a configuration that represents the ground state of a particular problem.

References:

[1] Quantum annealing: An introduction and new developments, Masayuki Ohzeki and Hidetoshi Nishimori, 2010

[2] Quantum Annealing in the Transverse Ising Model, Tadashi Kadowaki and Hidetoshi Nishimori, 1998

[3] Optimization by Simulated Annealing, S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, 1983

Quantum algorithm for linear systems of equations

Quantum computing model: Gate model

Paper: Quantum algorithm for linear systems of equations, Aram W. Harrow, Avinatan Hassidim and Seth Lloyd, 2009

Description: The quantum algorithm for linear systems of equations, often simply referred to as the HHL algorithm, is a famous quantum algorithm with a wide range of use cases and was proposed by Aram W. Harrow, Avinatan Hassidim and Seth Lloyd in 2009. The HHL algorithm is used to solve a linear system of equations. More specifically, given a Hermitian matrix A and a unit vector b , the goal is to find x such that $Ax = b$ is satisfied. It is assumed that one does not need to know x itself. That is, one is instead interested in an estimate of the expectation value of some operator applied to x . Also note that x is represented by a quantum state. The HHL algorithm provides an exponential speed-up when compared to the best classical algorithm performing the same task, however, some conditions apply. For example, the matrix A should be 'well-conditioned' and sparse.

Variational Quantum Eigensolver

Quantum computing model: Hybrid quantum gate / classical computer

Paper: Quantum optimization using variational algorithms on near-term quantum devices, Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli and Kristan Temme

Problem type: Optimization

Description: The variational quantum eigensolver (VQE) is a hybrid quantum / classical algorithm that can be applied to a range of problems including optimization problems and problems in quantum chemistry. Any problem that can be formulated as a Hamiltonian can potentially be solved by the VQE. Furthermore, the VQE is in particular a method that can run on near-term quantum computers, making this an interesting algorithm for the current state of the field.

The VQE follows a hybrid approach, that is, it uses both a quantum and a classical computer in its computation and works as follows. The algorithm begins by preparing a trial state using the quantum computer. This trial state is created by applying a set of parametrized gates. In the next step, also performed on the quantum computer, the expectation values are measured. The expectation values are then used to calculate the energy, that is, the cost function on the classical computer. In the last step, which is also performed on the classical computer, the parameters are adjusted in order to minimize the cost function. The resulting new parameters are then used in the next iteration of the algorithm. This process continues until the minimum energy and hence the optimal parameters is found. The configuration with the lowest energy corresponds to the solution of the problem.

In summary, the VQE is an algorithm that can solve a wide range of problems formulated as Hamiltonians on near-term quantum computers. In particular, it is a hybrid quantum/classical algorithm, that is, an algorithm that uses both a quantum and a classical computer to solve a particular problem.

Implementation available: Qiskit, Pennylane