Fuzzy Logic on Quantum Annealers

Amir Pourabdollah, Giovanni Acampora and Roberto Schiattarella

***Abstract*—Quantum computation is going to revolutionize the world of computing by enabling the design of massive parallel algorithms that solve hard problems in an efficient way, thanks to the exploitation of quantum mechanics effects, such as superposition, entanglement and interference. These computational improvements could strongly influence the way how fuzzy systems are designed and used in contexts, such as big data, where computational efficiency represents a non-negligible constraint to be taken into account. In order to pave the way towards this innovative scenario, this paper introduces a novel representation of fuzzy sets and operators based on Quadratic Unconstrained Binary Optimization (QUBO) problems, so as to enable the implementation of fuzzy inference engines on a type of quantum computers known as quantum annealers.**

I. Introduction

Quantum computers have become reality thanks to the effort of some majors in developing innovative technologies that enable the usage of quantum effects in computation, and pave the way towards the design of efficient quantum algorithms to use in different applications domains, from finance and chemistry to artificial and computational intelligence [1]. Indeed, these quantum effects known as superposition, entanglement and interference, arouse a massive parallelism in data processing [2], playing a key role both in defining new computational methodologies and improving the performance of established computational theories, including fuzzy logic. In fact, today's applications of fuzzy systems are increasingly working with large amounts of data, and there is a strong emergence of identifying innovative computational paradigms capable of efficiently managing this type of systems. For example, quantum computation will be able to support fuzzy logic allowing to combine a large number of fuzzy rules in a superposition of quantum states and enable the efficient computation of these rules through entanglement and interference. However, in order to take the first step in this fascinating and futuristic scenario, it is necessary to introduce a quantum representation of the fuzzy sets and of the operators that operate on them. In this paper, an approach based on a Quadratic Unconstrained Binary Optimization (QUBO) problem is used to bridge this gap. A QUBO problem is a NP-Hard problem that cannot be efficiently solved using classical computers, and whose objective is to minimize a quadratic polynomial over binary variables. This kind of problems were largely studied in computer science: among other applications, they have been used to model the traveling salesman problem [3], or to training machine learning models [4]. A QUBO problem has a strong relationship with quantum computation. In fact, it can be shown that any quadratic optimisation problem can be mapped into the search of the ground state of an Ising model [5]. Considering this, a QUBO problem can be efficiently solve on particular quantum computers, known as Quantum Annealers or Adiabatic Quantum Computers. Such devices have been shown to outperform classical computers on several instances [6]. Currently, these quantum devices are accessible through cloud platforms such as the one provided by D-Wave [7]. In particular, D-Wave machines use quantum annealing to solve QUBO problems [8]. Quantum annealing is a metaheuristic whose goal is to find the global minimum of a given objective function over a given set of candidate solutions by exploiting the massive parallelism induced by quantum superposition and entanglement. As a consequence, a QUBO-based representation of fuzzy logic enables a direct implementation of fuzzy reasoning on quantum annealers, opening the way towards quantum versions of fuzzy inference engines. Specifically, in this paper QUBO problems are used to represent fuzzy sets and basic fuzzy operators such as fuzzy union, fuzzy intersection, alpha-cut and maximum, and execute them on a quantum D-Wave annealer as shown in Section V.

Through this new representation, this work aims to show the appropriateness of quantum computers in performing the basic operations of fuzzy systems, and pave the way for future scenarios where quantum fuzzy systems can be computationally attractive compared to their classical counterparts.

II. Related Work

Both quantum mechanics and fuzzy sets theory deal with uncertainties, while both theories differ from those used in classical probability theory. This similarity between the two theories has motivated some research works to bridge between the two, or take the advantages of one for developing methods for the other.

Reviewing the literature identifies three distinctive groups of research directions in bridging between fuzzy logic and quantum computing domains:

- The first direction is on how quantum computing can be used for implementing fuzzy systems.
- Developing quantum-inspired algorithms (which are still implemented in classical computers) is the idea behind the second direction.
- The third direction is on how fuzzy sets theory can be used for simulating or modelling the physical quantum systems.

While the first two approaches are interesting for computer scientists, the last direction is mainly the focus of physicists

A. Pourabdollah is with School of Science and Technology, Nottingham Trent University, Nottingham, United Kingdom, e-mail: amir.pourabdollah@ntu.ac.uk

G. Acampora and R. Schiattarella are with the Department of Physics "Ettore Pancini", University of Naples Federico II, Naples, 80126 Italy e-mail: {giovanni.acampora, roberto.schiattarella}@unina.it.

G. Acampora is with Istituto Nazionale di Fisica Nucleare, Sezione di Napoli, 80126, Italy

and mathematicians. Moreover, there are recent works on theoretical integration of the two domains, e.g., in [9].

From the first group, in [10], the idea of "quantum fuzzy sets", i.e., superposition of many fuzzy sets at once is proposed. In [11], Grover's quantum search algorithms ([12]) is used to implement an inference engine in fuzzy rule-based systems. Also in [13], quantum gates and quantum circuits are used to interpret intuitionistic fuzzy sets by superposition of membership and non-membership degrees. The other example is [14], where the parallelism characteristics of quantum computers have been used for speeding up fuzzy inference calculations.

From the second group, because of the mentioned similarities between the two domains, fuzzy and multi-valued logic is used to represent quantum gates and circuits (e.g., in [15]). Another area of work is on the potential power of fuzzy sets in representing the probabilistic characteristics of quantum mechanics (e.g., in [16]).

Finally, an example of the quantum-inspired classical computation in the third group is improving the robustness of fuzzy controllers by modifying their inference performance based on quantum algorithms (known as *quantum fuzzy inference* in [17]).

The quantum-inspired algorithms are not specific to fuzzy systems, as it also exists for some other computational intelligence methods, e.g. quantum-inspired genetic algorithm [18], quantum fuzzy C-means data clustering [19] and quantum-inspired neuro-fuzzy systems [20]. Some other research works around quantum-inspired computational intelligence are comprehensively reviewed in [21].

The focus of this paper is within the first group of research works, since the aim is to represent fuzzy systems with components of quantum computers. To the best of our knowledge, fuzzy logic concepts have never been represented previously by QUBO problems and an adiabatic model of quantum computing. This will be explained in more details in the next sections.

III. Basic Concepts: QUBO Problem and Adiabatic Model of Computation

Quadratic Unconstrained Binary Optimization problems are traditionally used in computer science. They are defined as optimization problems of functions formulated in Binary Quadratic Model (BQM). Formally, if $\mathcal{Q}$ is an upper-diagonal matrix, which is an NxN upper-triangular matrix of real weights, and X is a vector of binary variables, a QUBO problem consists in minimizing the function:

$$f(X) = \sum_{i=1}^{n}(q_i x_i) + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}(q_{ij} x_i x_j) \tag{1}$$

where q_i and q_{ij} are configurable (linear and quadratic) coefficients. For example, a BQM could be in the form of $f(X) = 2x_i + 5x_2 - 4x_1x_2$.
This kind of problem can be addressed efficiently by quantum annealers. In this model of computing the basic building blocks are the so-called *quantum bits (qubits)*. While a classical bit can take a binary value, a qubit in its superpositioned state can take both 0 and 1 with different "probabilities". Moreover, these qubits can be entangled and if that happens one qubit state depends on another one. In the quantum annealers model, a different setting of qubits is specialised to find the optimum solution for minimising a binary objective function [22]. The quantum computer works out the optimum solution by means of minimising the total energy of the quantum system in an annealing process, that is why this model is also called quantum annealing. Briefly, formulating a problem in adiabatic model is finding q_i and q_{ij}, respectively associated to the superposition and entanglement biases, so that assignments of binary values $x_a, \ldots, x_n$ minimises the objective function, thus represents the solutions to the problem. Then during an annealing phase, the qubits are collapsed to 0 or 1 states, so that the system naturally selects its minimum possible energy. This means that the binary states of the collapsed qubits collectively provide a solution for $f(X)$ minimisation. Similar to any quantum system, the solution is probabilistic, so that the solutions made by a number of runs (called sampling) are being averaged.

In the next section, it will be shown how the adiabatic model can be used for efficiently representing the basic fuzzy logic concepts.

IV. A QUBO Representation of Fuzzy Logic

The idea of representing fuzzy sets with quantum states comes from the similarity between the degrees of membership (a value between 0 and 1), and the probability of collapsing a superpositioned qubit into a binary state. The motivation of this representation is to show the supremacy of algorithms in future quantum computers in dealing with complex fuzzy logic calculations compared to the classical computers. Such a representation of fuzzy sets is applicable in both circuitry (i.e., using quantum gates) and adiabatic (i.e., quantum annealing) models of quantum computing. Here we focus on the adiabatic model because of its relative simplicity and the availability of large number of qubits in the quantum computers developed for the adiabatic model (e.g., D-Wave Systems).

In the following subsection, we first introduce the underlying concept of adiabatic quantum model of computation, then the quantum representation of fuzzy set will be introduced (called *qfuzzy*).

A. A Quantum Representation of Fuzzy Sets

Representing a fuzzy set with qubits, may not initially seem to be any similar to an optimisation problem. As will be shown here, they can be formulated in QUBO so that a quantum computer can treat them as optimisation problems. We will review how to represent a fuzzy set by qubit states in this subsection. In the next section, it will be explained how implementing fuzzy set operations (such as union and intersection) in the quantum model is formulated as a QUBO optimisation problem.

Let A be a fuzzy set with n members (x_i) having membership grades $\mu_A(x_i)$. A can be represented by n qubits (q_i) in superposition state with $p_i(1) = \mu_A(x_i)$; where $p_i(1)$ is the probability of q_i being collapsed to the state 1.

The above definition is applicable to both circuitry and adiabatic models. In the adiabatic model, setting up a qubit system to represent a fuzzy set is equivalent to giving linear biases to the n qubits corresponding to the values of the membership function. There will be no quadratic term necessary for the representation. This is trivial since the members of a fuzzy set are independent. This practically means that a fuzzy set is simply represented by a system of biased (stimulated) qubits called *qfuzzy* system.

The corresponding BQM objective function of a qfuzzy system is thus defined as:

$$f(X) = \sum_{i=1}^{n} \big(\mu_A(x_i) \cdot x_i\big); \quad X = \{x_1, \ldots, x_n\} \tag{2}$$

Obviously, a qfuzzy system does not represent an optimisation problem, thus it is not the subject of annealing. It is actually needed on its stimulated state not in its collapsed state.

The next step towards implementing a quantum-fuzzy inference engine is to implement the basic fuzzy set operations in quantum algorithms. This will be explained in the next section.

B. A Quantum Implementation of Basic Fuzzy Set Operations

Having defined how to represent fuzzy sets as qfuzzy systems, the aim of this section is to show how applying basic fuzzy set operations (Union, Intersection, Maximum and Alpha-cut) on fuzzy sets can be translated to solving QUBO problems over the underlying qfuzzy systems, hence can be implemented on quantum computers.

1) Intersection Operation: Let fuzzy sets A and B, with discrete membership functions $\mu_A(x_i)$ and $\mu_B(x_i)$ over the same universe of discourse $X = \{x_1, x_2, \ldots, x_n\}$ are represented as two qfuzzy systems as follows:

$$\begin{aligned} A &: \textstyle\sum_{i=1}^{n} \big(\mu_A(x_i) \cdot x_i\big) \\ B &: \textstyle\sum_{i=1}^{n} \big(\mu_B(x_i) \cdot x_i\big) \end{aligned} \tag{3}$$

where ":" denotes "represented by".

We choose minimum-intersection and would like to find a new set C represented by a new qfuzzy system represented as:

$$C = A \cap B : \sum_{i=1}^{n} \big(\mu_C(x_i) \cdot x_i\big) \tag{4}$$

$$\mu_C(x_i) = \min\big(\mu_A(x_i), \mu_B(x_i)\big) \tag{5}$$

There is no singular operation existing in quantum computers that can compare or find the minimum of the bias values of any two qubits. However, we can take the advantage of the fact that the annealing naturally leads to the minimum energy of a set of interconnected qubits, thus finds the minimum of an objective function. Therefore the question is which objective function over which qubits is to be minimised.

Let us create another quantum system Y with n qubits. The idea behind this system is that by the end of its annealing, the collapsed qubits act as a 0/1 switch between sets A and B, so that if qubit $y_i = 0$, the ith member of A is selected and if $y_i = 1$ the corresponding member of B is selected.

Let us define a BQM objective function for this system as:

$$f(Y) = \sum_{i=1}^{n} \big((\mu_B(x_i) - \mu_A(x_i)) \cdot y_i\big) \tag{6}$$

It can be seen that if $\mu_A(x_i) < \mu_B(x_i)$, the ith term of (6) takes its minimum when $y_i = 1$. Similarly if $\mu_B(x_i) < \mu_A(x_i)$, this term takes its minimum when $y = 0$. Collectively, $f(Y)$ takes its minimum when $y_i = 0$ for the terms with $\mu_A(x_i) < \mu_B(x_i)$ and $y_i = 1$ for the terms with $\mu_A(x_i) > \mu_B(x_i)$. For when $\mu_A(x_i) = \mu_B(x_i)$, the quantum annealing may randomly collapse y_i to either 0 or 1 randomly, which does not change the result. Finally, when each y_i is observed after the annealing, it indicates which set has the minimum membership function at the ith point.

Once the values y_i are resulted from the annealing of system Y, the qfuzzy system C can be made and represented as:

$$C = A \cap B : \sum_{i=1}^{n} \big(\mu_C(x_i) \cdot x_i\big) \tag{7}$$

$$\mu_C(x_i) = (1 - y_i) \cdot \mu_A(x_i) + y_i \cdot \mu_B(x_i) \tag{8}$$

2) Union Operation: Similarly, the union operation (based on maximum) of qfuzzy systems A and B is defined be introducing an intermediate quantum system Y consisting of n qubits and a BQM objective function defined as:

$$f(Y) = \sum_{i=1}^{n} \big((\mu_A(x_i) - \mu_B(x_i)) \cdot y_i\big) \tag{9}$$

After annealing of Y, it can be similarly shown that:

$$C = A \cup B : \sum_{i=1}^{n} \big(\mu_C(x_i) \cdot x_i\big) \tag{10}$$

$$\mu_C(x_i) = (1 - y_i) \cdot \mu_A(x_i) + y_i \cdot \mu_B(x_i) \tag{11}$$

3) Alpha-cut Operation: The alpha-cut operator takes a fuzzy set and produces a crisp set of values along x-axis for which their membership grade is equal or greater than a given alpha. Let fuzzy set A is represented as a qfuzzy systems as:

$$A : \sum_{i=1}^{n} \big(\mu_A(x_i) \cdot x_i\big) \tag{12}$$

We would like to extract a crisp set Z, in which:

$$Z = \{x \in A \,|\, \mu_A(x) \geq \alpha\} \tag{13}$$

Again, in the lack of any comparison operator, the alpha-cut operator has to be reformulated in BQM. Similar to the max operator, we would need an intermediate qubit system Y in order to binarily flag out the x-values which are the members of the alpha-cut from those who are not. An initial suggestion is that the required BQM objective function should penalises (i.e., produces positive terms for all $\mu_A(x_i) < \alpha$). This initial objective function can then be formulated as:

$$f(Y) = \sum \big(\alpha - \mu_A(x_i)\big) \cdot y_i \tag{14}$$

This function takes its minimum by setting $y_i = 0$ for those x_i with a greater membership grades than α, and setting $y_i = 1$ otherwise. However, the problem is that if there is a x_i point in which $\mu_A(x_i) = \alpha$, the corresponding linear bias in the objective function is zero, thus the result is independent of the corresponding y_i value. This means that the objective function takes its minimum equally for both $y_i = 0$ and $y_i = 1$, in which case the algorithm is unable to identify if x_i is in or is out of the α-cut. To resolve this issue we need to adjust the objective function, so that it produces no zero biases, i.e. it must produce either positive biases for $\mu_A(x_i) \geq \alpha$ or negative biases otherwise. For this, a parameter ϵ is needed that is equal to the half of the membership function resolution, so that the objective function can be adjusted as:

$$f(Y) = \sum \big(\alpha - \epsilon - \mu_A(x_i)\big) \cdot y_i \tag{15}$$

In this case, setting $y_i = 1$ for all the α-cut members would give $f(Y)$ its most negative possible value. Therefore the collapsed values of y_i's after the annealing process is actually a membership flag for any corresponding x_i to the alpha-cut.

Finally, the alpha-cut Z can be redefined as:

$$Z = \{x_i \in A \,|\, y_i = 1; y_i \in Y\} \tag{16}$$

4) Maximum Operation: The maximum operator determines the member(s) of a fuzzy set with maximum membership grade(s). This is particularly useful for defuzzification in FOM/MOM/LOM (First/Mean/Last of Maxima) methods. Let fuzzy set A is represented as a qfuzzy systems:

$$A : \sum_{i=1}^{n} \big(\mu_A(x_i) \cdot x_i\big) \tag{17}$$

We would like to extract a crisp set M, in which:

$$M = \{x \in A \,|\, \mu_A(x) = \max_i \big(\mu_A(x_i)\big)\} \tag{18}$$

Although this look similar to the case of maximum operator, the key difference is that instead of switching between corresponding members of two sets, here the aim is switching between the members of the same set based on whether the member has the maximum membership function or not. For example, if the set takes its single maximum at point m, there will be a single 1 in the mth qubit of Y when collapsed to binary states. This key difference, as will be seen, leads to a substantially different algorithm, than unlike the previously examined operations, adds some quadratic terms to the objective function.

Since there is no maximum operator existing for qubit biases, here we need another quantum system (Y) with n qubits, in which the binary state of the qubit y_i can indicate whether x_i belongs to set M or not. In other words, Y acts as a 0/1 flag so that x_i has taken the maximum membership grade in A only if $y_i = 1$.

The BQM objective function of system Y must be formulated in a way that it increases more if smaller values of $\mu_A(x_i)$ are flagged, compared to the larger values of $\mu_A(x_i)$. Since flagging is equivalent to setting $y_i = 1$, this means that the objective function must be increased (i.e. penalised) by $1 - \mu_A(x_i)$ if its corresponding y_i is 1. Thus we expect the objective function to be in the form of:

$$f(Y) = \sum \big(1 - \mu_A(x_i)\big) \cdot y_i + \ldots \tag{19}$$

A problem with this partial objective function is that it always takes its minimum when all $y_i = 0$. Note that $1 - \mu_A(x_i) \geq 0$ therefore the minimum of $f(Y)$ is zero by setting all y_i to zero. We must then need to penalise the case of all $y_i = 0$. On the other hand, having more than a single zero should be penalised too since this will always increases the value of $f(Y)$.

Without loosing the generality, let us assume that $\mu_A(x_i)$ has a single maximum. In this case, we expect exactly a single 1 in the result, i.e., the sum of all y_i's should not be more nor less than 1. Therefore we define the penalty term as $(\Sigma y_i - 1)^2$. The objective function then becomes:

$$\begin{aligned} f(Y) &= \Sigma\big(1 - \mu_A(x_i)\big)y_i + (\Sigma y_i - 1)^2 \\ &= (1 - \mu_A(x_1)) \cdot y_1 + (1 - \mu_A(x_2)) \cdot y_2 + \\ &+ \ldots + (1 - \mu_A(x_n)) \cdot y_n + \\ &+ y_1^2 + y_2^2 + \ldots + y_n^2 + 1 + \\ &+ 2y_1y_2 + 2y_1y_3 + \ldots + 2y_{n-1}y_n - \\ &- 2y_1 - 2y_2 - \ldots - 2y_n \end{aligned} \tag{20}$$

Note that for binary values, $y_i^2 = y_i$, and that the constant value 1 has no effect on minimising the objective function. Then the equivalent objective function (named the same for simplicity) is defined as:

$$\begin{aligned} f(Y) &= -\mu_A(x_1)y_1 - \mu_A(x_2)y_2 + \ldots \\ &= -\mu_A(x_n)y_n + 2y_1y_2 + 2y_1y_3 + \ldots + 2y_{n-1}y_n \end{aligned} \tag{21}$$

Therefore,

$$f(Y) = -\sum_{i=1}^{n} \mu_A(x_i)y_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 2y_iy_j \tag{22}$$

The above is in BQM form with linear biases $-\mu_A(x_i)$ and all quadratic coefficients equal to 2. Practically, the quadratic terms of this system of qubits are independent of the membership grade, so that they can be pre-programmed.

If there are more than a single maximum points, $f(Y)$ will equally take its minimum by setting $y_i = 1$ for either of them (not for a combination of them). This means that running the quantum annealing algorithm by this point may not give a single answer to the fuzzy set operation (i.e., the maximum operation). Due to the embedded randomness of the quantum annealing, each run of the algorithm may give one of the possible answers.

To find a single answer in the form of a binary bits, we first notice that any given answer can maximise the membership function. Let us assume that the mth bit of Y is found to be 1, and we call $\mu_A(x_m)$ as μ_M. Then we notice that by knowing μ_M, the required maximum operator is a special case

TABLE I
THE LINEAR AND QUADRATIC COEFFICIENTS NEEDS FOR THE OBJECTIVE FUNCTION ASSOCIATED TO THE STUDIED FUZZY SET OPERATORS

Fuzzy set operator	**Linear bias (q_i)**	**Quadratic bias (q_{ij})**
Union	$\mu_A(x_i) - \mu_B(x_i)$	0.0
Intersection	$\mu_B(x_i) - \mu_A(x_i)$	0.0
Alpha-cut	$\alpha - \epsilon - \mu_A(x_i)$	0.0
Maximum (single)	-$\mu_A(x_i)$	2.0

of applying α-cut operator when $\alpha = \mu_M$. The implementation of this part is already explained in the previous subsection.

Therefore, applying the maximum operator can be split into two steps:

- A quantum annealing based on (22) to find a possible binary sequence leading to μ_M.
- Another quantum annealing based on (15) with $\alpha = \mu_M$ to find the final binary sequence.

Now that the binary sequence y_i is determined, the target set M can be simply defined as:

$$M = \{x_i \in A \,|\, y_i = 1\} \tag{23}$$

The resulted y_i bits can also be readily used for FOM/-MOM/LOM defuzzifications: The x_i values corresponding to the left-most, middle and the right-most 1's in the binary sequence are equivalent to the FOM, MOM and LOM of X, respectively.

C. A Summary

As explained in the previous subsections, each studied fuzzy set operator needs associating an objective function $f(Y)$ in binary quadratic form. Table I summarises the linear and quadratic coefficients of $f(y)$ for each fuzzy set operator. With reference to (1), the linear and quadratic coefficients (or biases) are called q_i and q_{ij} respectively.

V. IMPLEMENTIG QUBO-BASED FUZZY LOGIC ON QUANTUM ANNEALERS

Here a sample implementation of a fuzzy set operation is demonstrated. Among the studied fuzzy set operations in the previous section, the maximum and the alpha-cut operations are chosen for this demonstration. We use (22) and (15) for building the associated objective function, minimise it using quantum annealing and show the resulted binary sequences.

The implementation is based on D-Wave System[1]. D-Wave currently provides access to its D-2000 series of quantum computers through D-Wave Leap-2, a cloud-based quantum programming platform. D-Wave also provides some Python libraries for programming using web-based and desktop IDE that connect to the same platform. The details of these libraries is out of the scope of this paper, and can be found in D-Wave Ocean Software Documentation[2]. Here we show a sample program listing and the outputs for the maximum operator.

[1]https://docs.dwavesys.com/docs/latest/index.html
[2]https://docs.ocean.dwavesys.com/en/stable/

We assume a discrete fuzzy set X with 10 points as follows:

$$X = \{0.8/10, 0.3/20, 0.7/30, 0.9/40, 0.7/50, \\ 0.5/60, 0.3/70, 0.2/80, 0.1/90, 0.0/100\}$$

For simplicity, this set has a single maximum value, therefore the algorithm includes a single step. According to table I, the objective function can be programmed based on the given linear and quadratic biases. The Python listing for this program is shown in Listing 1.

Listing 1. Python program for maximum operator

```
import dimod
from dwave.system import LeapHybridSampler

mu = [0.8, 0.3, 0.7, 0.9, 0.7, 0.5,
  0.3, 0.2, 0.1, 0.0]

linear = {}
quadratic = {}

for i in range(len(mu)):
  linear.update({'y'+str(i) : -mu[i]})

for i in range(len(mu)):
  for j in range(i+1, len(mu)):
    quadratic.update({('y'+str(i),'y'+str(j)):2})

bqm = dimod.BinaryQuadraticModel(
  linear, quadratic, 0, 'BINARY')

sampler = LeapHybridSampler()
answer = sampler.sample(bqm)
print(answer)
```

The results of executing the program of Listing 1 on D-Wave quantum computer is shown in Listing 2. As it can be seen, the result is 10 bits, the 4th of which is set to 1. This means that the 4th member of X (40) corresponds to its maximum. As a defuzzifier, this shows that the fuzzy set's MOM is 40.

Listing 2. The results of running Listing 1 on D-Wave quantum computer

```
  y0 y1 y2 y3 y4 y5 y6 y7 y8 y9 energy
  0  0  0  1  0  0  0  0  0  0   -0.9
```

Next, to find a sample $\alpha - cut$ of X, let us assume $\alpha = 0.5$. Bease on I, the required program is shown in Listing 3.

Listing 3. Python program for finding alhpa-cut (alpha=0.5, resolution=0.1)

```
import dimod
from dwave.system import LeapHybridSampler

mu=[0.8, 0.3, 0.7, 0.9, 0.7, 0.5,
  0.3, 0.2, 0.1, 0.0]

linear = {}
quadratic = {}
alpha = 0.5
epsilon = 0.1/2

for i in range(len(mu)):
  linear.update({'y'+str(i) : alpha-epsilon-mu[i]})

for i in range(len(mu)):
  for j in range(i+1, len(mu)):
    quadratic.update({('y'+str(i),'y'+str(j)):0})

bqm = dimod.BinaryQuadraticModel(
  linear, quadratic, 0, 'BINARY')
```

```
sampler = LeapHybridSampler()
answer = sampler.sample(bqm)
print(answer)
```

Listing 4. The results of running Listing 3 on D-Wave quantum computer

```
  y0 y1 y2 y3 y4 y5 y6 y7 y8 y9 energy
  1  0  1  1  1  1  0  0  0  0  -1.35
```

The results of running Listing 3 is shown in Listing 4. It can be seen that the 5 bits are set to 1, that corresponds to the 5 alpha-cut members, as being $Z = \{10, 30, 40, 50, 60\}$.

VI. Conclusions: Towards Quantum Fuzzy Reasoning

Fuzzy sets and operators represents the atomic elements of fuzzy inference engines (e.g. Mamdani, Takagi-Sugeno-Kang), collections of rules able to map inputs to outputs by taking into strongly account vagueness and uncertainty in data. In this paper, both fuzzy sets and fuzzy operators have been represented/formulated using quadratic unconstrained binary optimization problems, where QUBO functions are minimized to evaluate the membership degree of fuzzy sets, and compute the output value of fuzzy operators, such as union, intersections, alpha-cut and maximum. In our research, the minimization of QUBO functions was carried out by means of D-Wave quantum annealers, i.e., a particular type of quantum computer that exploits the annealing phenomenon to find, in an efficient way, optimal solutions of problems formulated as QUBOs. The motivation for the authors in using this model of computation lies in the fact that it is, by its nature, useful in solving optimization problems. However, in future studies the proposed approach will be compared with other quantum computing models, such as circuit model, quantum Turing machine, measurement-based quantum computation or quantum random access machine. Among these, the circuit model is surely the most widespread and used one and future studies involving this computational model may exploit some quantum algorithms present in literature for QUBO problems optimization, such as the those proposed in [23], [24], which respectively leverage approaches based on the Grover's algorithm [12] and quantum genetic algorithms.

Finally, it is important to highlight that the goal achieved by this research is the proof of the quantum computers feasibility in doing basic fuzzy operations. Classically, these operations are carried out efficiently and the aims of the authors was not to prove a quantum advantage in so doing. However, the achieved result represents the first step towards a scenario where quantum computing and fuzzy logic will interact to solve problems in a more efficient way than conventional current approaches. In fact, today's applications of fuzzy systems are increasingly working with large amounts of data or large sets of rules, and there is a strong emergence of identifying innovative computational paradigms capable of efficiently managing this type of systems. Because of this, the QUBO representation of fuzzy logic introduced in this paper, will be used to develop a quantum framework capable of firing fuzzy rules in efficient way by exploiting quantum phenomena as superposition and entanglement and, therefore, the proposed approach can lead to completely new lines of research, where the quantum implementation of fuzzy inference engines will represent an added value both from the theoretical and application point of view.

References

[1] G. Acampora, "Quantum machine intelligence," *Quantum Machine Intelligence*, vol. 1, no. 1, pp. 1–3, 2019. [Online]. Available: https://doi.org/10.1007/s42484-019-00006-5

[2] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467–488, 1982. [Online]. Available: https://doi.org/10.1007/BF02650179

[3] C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti, "A qubo model for the traveling salesman problem with time windows," *Algorithms*, vol. 12, no. 11, p. 224, 2019.

[4] P. Date, D. Arthur, and L. Pusey-Nazzaro, "Qubo formulations for training machine learning models," *Scientific Reports*, vol. 11, no. 1, pp. 1–10, 2021.

[5] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, vol. 2, p. 5, 2014.

[6] H. Ushijima-Mwesigwa, C. F. Negre, and S. M. Mniszewski, "Graph partitioning using quantum annealing on the d-wave system," in *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, 2017, pp. 22–29.

[7] "D-wave system documentation," https://www.dwavesys.com.

[8] P. Date, R. Patton, C. Schuman, and T. Potok, "Efficiently embedding qubo problems on adiabatic quantum computers," *Quantum Information Processing*, vol. 18, no. 4, pp. 1–31, 2019.

[9] H.-x. Li, *Fuzzy Systems To Quantum Mechanics*. World Scientific, 2020, vol. 87.

[10] M. A. Mannucci, "Quantum fuzzy sets: Blending fuzzy set theory and quantum computation," 2006.

[11] G. Acampora, F. Luongo, and A. Vitiello, "Quantum implementation of fuzzy systems through grover's algorithm," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

[12] L. K. Grover, "A fast quantum mechanical algorithm for database search," 1996.

[13] R. Reiser, A. Lemke, A. Avila, J. Vieira, M. Pilla, and A. Du Bois, "Interpretations on quantum fuzzy computing: intuitionistic fuzzy operations$\times$ quantum operators," *Electronic Notes in Theoretical Computer Science*, vol. 324, pp. 135–150, 2016.

[14] G. G. Rigatos and S. G. Tzafestas, "Parallelization of a fuzzy control algorithm using quantum computation," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 451–460, 2002.

[15] R. Leporini, C. Bertini, and F. C. Fabiani, "Fuzzy representation of finite-valued quantum gates," *Soft Comput. https://doi. org/10*, vol. 1007, 2020.

[16] D. Aerts, T. Durt, and B. Van Bogaert, "A physical example of quantum fuzzy sets and the classical limit," *Tatra Mountains Math. Publ*, vol. 1, p. 5, 1992.

[17] L. Litvintseva, I. Ul'yanov, S. Ul'yanov, and S. Ul'yanov, "Quantum fuzzy inference for knowledge base design in robust intelligent controllers," *Journal of Computer and Systems Sciences International*, vol. 46, no. 6, pp. 908–961, 2007.

[18] R. Lahoz-Beltra, "Quantum genetic algorithms for computer scientists," *Computers*, vol. 5, no. 4, p. 24, 2016.

[19] N. Bharill, O. P. Patel, A. Tiwari, L. Mu, D.-L. Li, M. Mohanty, O. Kaiwartya, and M. Prasad, "A generalized enhanced quantum fuzzy approach for efficient data clustering," *IEEE Access*, vol. 7, pp. 50 347–50 361, 2019.

[20] O. P. Patel, N. Bharill, A. Tiwari, and M. Prasad, "A novel quantum-inspired fuzzy based neural network for data classification," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[21] A. Manju and M. J. Nigam, "Applications of quantum inspired computational intelligence: a survey," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 79–156, 2014.

[22] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.

[23] A. Gilliam, S. Woerner, and C. Gonciulea, "Grover adaptive search for constrained polynomial binary optimization," *Quantum*, vol. 5, p. 428, 2021.

[24] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum genetic optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 231–241, 2008.