Hybrid quantum-classical metaheuristics for automated machine learning applications

Von Dollen, D.J.

Chapter 4

Quantum-enhanced genetic algorithms

In this chapter, we examine selection operators for evolutionary algorithms. Population-based heuristics are often used in machine learning model selection and hyperparameter optimization [73]. As model selection is a core process in building machine learning pipelines, identifying areas where quantum computing may be used in conjunction with classical heuristics and search may prove beneficial for the machine learning practitioner. In our case, we study the encoding of the selection operator of the evolutionary algorithm as a QUBO, and use the quantum-enhanced selection operator as part of a genetic algorithm for hyperparameter optimization for machine learning models. This chapter has been published as:

D. Von Dollen, S. Yarkoni, D. Weimer, F. Neukart, and T. Bäck. Quantum-enhanced selection operators for evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, page 463–466, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392686. doi: 10.1145/3520304.3528915. URL `https://doi.org/10.1145/3520304.3528915`

4.1 Introduction

Evolutionary algorithms are nature-inspired models drawn from observations of organic evolution. Using selection, recombination, and mutation, this family of algorithms evolves populations to search an optimization domain with respect to individual

fitness within the population. Related to the fields of biology, numerical optimization, and artificial intelligence, these algorithms can also model a collective learning process, where individuals may not only represent a point in the domain of the objective function, but may also represent knowledge of an environment [7]. This family of algorithms is particularly well suited for black-box optimization, where there is no available knowledge of the internal structure of the objective function, in that there is no gradient and the landscape could be highly non-linear and rugged with many local optima.

In this work, we examine the use of a quantum annealing system to find solutions to the problem of optimal selection encoded as a binary quadratic model within an evolutionary algorithm. We examine the trade-off in selective pressure vs. exploration in the evolutionary search and show qualitative gains with respect to fitness and expected run-times in the form of average generations to convergence. We investigate these performance gains with respect to the change in the ratio of μ to n, or the size of the selected parent group and the number of population, and find that the performance gap increases as the ratio approaches μ/n=0.5. This is not without cost, as we also observe additional overhead in compute times, which is incurred by making calls across a network to query a quantum processing unit at each generation, an issue also identified in [193]. We also confirm findings the authors reported in this work, where for fully connected graphs of input QUBOs (Quadratic Unconstrained Binary Optimization) constructed from randomly initialized populations, hybrid quantum-classical algorithms outperform fully quantum solvers by finding lower-energy configurations given the input.

We test our quantum-enhanced algorithms on the IOHprofiler suite for black-box optimization, specifically examining pseudo-Boolean functions [56]. We find that for functions with perturbed fitness, quantum-enhanced selection operators achieve slightly better performance than their classical counterparts. We find that our quantum-enhanced algorithms generally match or outperform their classical counterparts on a majority of test functions, 10 out of 15 test functions.

4.2 Related Works

Quantum-inspired evolutionary algorithms have been well studied over the years, starting with [142], where the classical simulation of quantum mechanical properties was applied to the evolutionary search. This culminated in a large body of work with many variants of quantum-inspired algorithms as described by Zhang in [246].

In surveying quantum-inspired algorithms, Zhang noted three types of algorithms that combine quantum computational properties with evolutionary algorithms [246]. These include:

- Evolutionary Designed Quantum Algorithms (EDQA), which leverage evolutionary algorithms to evolve new designs of quantum algorithms.
- Quantum Evolutionary Algorithms (QEA), where evolutionary algorithms are implemented on a quantum computer.
- Quantum-Inspired Evolutionary Algorithms (QIEA), which are algorithms where the evolutionary process is supplemented by routines inspired by quantum mechanics but implemented using classical hardware.

Along with these, we propose to consider an additional algorithm class. As we are currently in the NISQ era for quantum hardware, we can also examine hybrid quantum-classical algorithms, where some portions or subroutines of the algorithm's execution are performed on a quantum computer, and other portions are performed classically. We call this class quantum-enhanced evolutionary algorithms. Studies of quantum-enhanced evolutionary algorithms can be examined from the perspective of leveraging a quantum device and quantum mechanical properties for selection, crossover, or mutation operators within the heuristic of the evolutionary search.

The idea of a genetic algorithm assisted by quantum annealing was proposed by Chancellor in [38], and the authors of [106] investigated using a quantum-assisted mutation operator, and leveraging reverse annealing runs using a quantum annealer. By performing quasi-local searches using the quantum-assisted mutation operator, the authors were able to show an improvement over forward quantum annealing in finding global optima for a set of input spin-glasses.

More recently, investigations into continuous black box optimization using a quantum-assisted acquisition function have been reported in [98]. In [193], the authors leverage a quantum annealing system to formulate continuous optimization problems cast within a quantum non-linear programming framework, and show applications within the green energy space. Sharabiani et al. also identified the overhead in the computation times in regards to querying a QPU for a subroutine for their optimization algorithm in their work. To our knowledge, there has been no prior investigation into quantum-enhanced routines for selection, which our work addresses.

There also exists a stream of research into the application of Grover's algorithm for unstructured search to global optimization [8]. Although we may not have systems

of the scale and fidelity available today to implement these algorithms on a practical level, this stream of research could be further investigated and realized as quantum systems come online with higher orders of available error-corrected qubits and longer coherence times.

We frame the selection process as a maximum diversity problem, which we encode using a binary quadratic model, where we wish to select a subset of parents for crossover and/or mutation, which preserve a high degree of quality diversity within the parent pool, while preserving a high degree of fitness in regards to the objective function. Maximum diversity problems have been shown to be NP-Hard [58, 48]. This connection motivates our investigation to use a QPU to sample quality solutions for this problem formulation. This type of combinatorial optimization scales as $\mathcal{O}(2^n)$ for an input size n, as there is a binary decision variable for each individual within the population.

4.3 Methods

Evolutionary Algorithms model processes observed in nature, including natural selection, reproduction, and mutation. In terms of global optimization, these processes are leveraged to evolve individual solutions with respect to an objective or fitness function. In the case of a genetic algorithm for pseudo-Boolean optimization, a population of individuals $P(0) := [\mathbf{a_1}, \ ... \ \mathbf{a_n}]$ is initialized, where the values of $\mathbf{a} \in \{0, 1\}^d$ are generated uniformly at random per dimension d. At each generation, individuals (also called chromosomes in the case of genetic algorithms) from a population are selected, recombined to produce offspring, and mutated, resulting in a new population. Over time, individuals converge to the minima of the objective function to be optimized with respect to their fitness values, given by a black-box objective function f. These properties make evolutionary algorithms powerful candidates for optimization for black-box functions where the domain and modality of the function are unknown for both discrete, as in the case of pseudo-Boolean optimization, and continuous optimization.

4.3.1 Selection Operators for Evolutionary Algorithms

In selecting parents from a population pool for mutation and crossover, we may choose from a number of selection operators. These operators may be deterministic or probabilistic and can include the following.

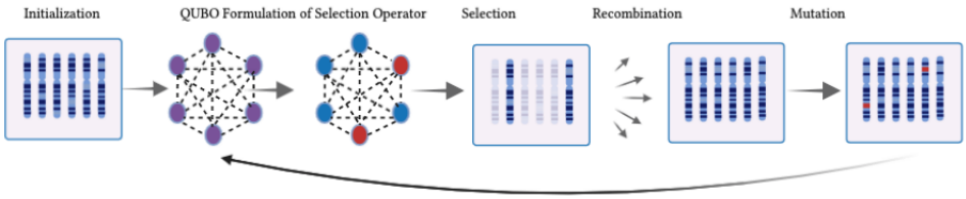

Figure 4.1: Diagram of process flow for the quantum-enhanced selection operator.

- *(μ, λ) selection*: In (μ, λ) selection, μ individuals are selected uniformly at random to create the λ offspring. In this case, only child offspring are included in the subsequent population.
- *$(\mu + \lambda)$ selection*: In $(\mu + \lambda)$ selection, μ individuals are selected uniformly at random to create the λ offspring. $(\mu + \lambda)$ selection is elitist, meaning that parents are included in subsequent generations. In our experiments, we also tracked and recombined the best solution found so far with selected members of the population to create subsequent generations.

We choose these operators in comparison to our quantum-enhanced operator to compare and contrast the trade-offs in exploration vs. exploitation in our evolutionary search.

4.3.2 Introducing A Quantum-Enhanced Selection Operator

When given a set of candidates within a population pool, a natural question is how to select parents with a high degree of fitness, yet are also diverse from one another, with the idea that we want to be able to balance the trade off between exploitation and exploration in the selection mechanism. Low population diversity may lead to more localized search and premature convergence.

In our formulation, we start by defining a matrix $\mathbf{Q}$:

$$\mathbf{Q}_{ij} = \begin{cases} -\alpha |f(\mathbf{a}_i)| & \text{if } i = j \\ -\beta |\operatorname{distance}(\mathbf{a}_i, \mathbf{a}_j)| & \text{if } i \neq j \end{cases} \tag{4.1}$$

Where $f(\mathbf{a}_i)$ is the fitness evaluated by the chromosome $\mathbf{a}_i$ and $\operatorname{distance}(\mathbf{a}_i, \mathbf{a}_j)$ is the pairwise distance metric between chromosomes within the population. Note that for this formulation, the distance metric may be arbitrarily chosen by the practitioner;

in our case, we use *Hamming* distance [84], where we negate the quadratic term to optimize strings with greater distance from each other in the sampled hypercube with the intention of improving diversity within the population.

We define the Hamming distance between two vectors $\mathbf{a}$ and $\mathbf{b}$ as follows:

$$\text{Hamming}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} \mathbf{1}[a_i \neq b_i] \tag{4.2}$$

We introduce the terms α and β as scaling constants, which we may use to adjust the optimization domain for the binary quadratic model for more or less ruggedness to take advantage of the effect of quantum tunneling.

The matrix $\mathbf{Q}$ acts as input for our resulting minimization problem, where we wish to find a minimum energy E through an optimal assignment of qubit values, represented as o_i where $o_i \in \{0, 1\}$ and $i \in \{1, ..., n\}$ to the indices of the population of size n. We constrain the formulation with a penalty term $(\sum_i \mathbf{o}_i - \mu)^2$ so that the sum of the selected subset is of size μ.

$$E(\mathbf{o}) = \sum_{i \leq j} \mathbf{o}_i \mathbf{Q}_{ij} \mathbf{o}_j + \left(\sum_i \mathbf{o}_i - \mu \right)^2 \tag{4.3}$$

Upon sampling a decision vector $\mathbf{o}$, we select parents from the population using the entries of $\mathbf{o}$. If $o_i = 1$, we select a parent from the population at i, otherwise we do not select that member of the population. Using this population subset of size μ, we may then perform crossover and mutation classically, creating a new population and increment to the next generation. For our experiments, we investigate elitist ($QE{-}(\mu{+}\lambda)$) and non-elitist ($QE{-}(\mu, \lambda)$) versions of the quantum-enhanced operator, where the parents are included in the former case and not in the latter case. We also include the heuristic of recombining the selected population with the best solution found so far in the elitist version for both classical and quantum-enhanced operators. We outline the pseudo-code for these algorithms in Algorithm 1.

4.4 Experiments

4.4.1 Benchmarking Quantum, Hybrid, and Classical Solvers

For our version of a quantum-enhanced evolutionary algorithm, we make calls to the D-Wave quantum annealer, using the quantum processing unit (QPU). During the annealing regime, the system starts in a superposition state for all qubit values, and

Algorithm 1 Quantum-Enhanced Genetic Algorithm (*QE-(μ + λ),QE-(μ ,λ)*)

Require: Population size n
Require: Chromosome size d
Require: Maximum generations t
Require: Mutation probability m
Require: Boolean flag *elitist*
Require: Boolean flag *quantum-enhanced*
Initialize $P(0) \leftarrow [\mathbf{a}_1, \ldots, \mathbf{a}_n]$, where $\mathbf{a}_i = [x_0, \ldots, x_d]$, $x_j \in [0, 1]$, uniformly distributed
Initialize $\mathbf{a}^+ \leftarrow \mathbf{a}_1$
for generation $g = 1$ to t **do**
    for $\mathbf{a}_i \in P(g-1)$ **do**
        if $f(\mathbf{a}_i) > f(\mathbf{a}^+)$ **then**
            $\mathbf{a}^+ \leftarrow \mathbf{a}_i$
        end if
    end for
    if *quantum-enhanced* **then**
        Use $P(g-1)$ to construct $\mathbf{Q}$ for QUBO as per 4.1
        Sample solution vector minimizing $E(\mathbf{o})$ from QUBO as per 4.3
        Select $P_\mu(g)$ from $P(g-1)$ where $o_i = 1$ for each o_i in $\mathbf{o}$
    else
        Select μ parents from $P(g-1)$ to form $P_\mu(g)$
    end if
    if *elitist* **then**
        Perform crossover between $\mathbf{a}^+$ and $P_\mu(g)$ to generate $P_\lambda(g)$
        $P_\lambda(g) \leftarrow P_\lambda(g) \cup P_\mu(g)$
    else
        Perform crossover within $P_\mu(g)$ to generate $P_\lambda(g)$
    end if
    Mutate $P_\lambda(g)$ with probability m
    $P(g) \leftarrow P_\lambda(g)$
end for
return $\mathbf{a}^+$

by gradually reducing the amplitude of a transverse field, the system is driven to a ground state. Using quantum-mechanical properties such as entanglement and superposition, we can observe an effect known as *quantum tunneling*, where barriers in the optimization landscape are surpassed, rather than walked or sampled.

The D-Wave 2000Q QPU is composed of 2000 qubits and 5600 couplers, with 128000 Josephson junctions. As the QPU may not have full connectivity, as it uses a *chimera* architecture, a *minor embedding* is created to model the fully connected graph on the chip. For our purposes, we used D-Wave's software tools to automatically create a minor embedding on the QPU to sample our problem [210]. For the D-Wave Clique Sampler, the tool attempts to find clique embedding on the chip of equal chain length. An important parameter, *chain strength*, was set as the maximum absolute value of the linear terms of the initialized binary quadratic model for both quantum samplers and embedding tools [210].

Before approaching the problem of utilizing the quantum-enhanced selection operator, it is natural to question how well a particular solver may find energy minima for the formulation of the binary quadratic model. To determine the quality of the solution, we initialized the populations of solutions at random uniformly with which we constructed $\mathbf{Q}$ matrices according to equations 4.1 and 4.3. We then ran trials for each solver type, with the set of solvers consisting of:

- D-Wave 2000Q - D-Wave Sampler (DwS) [210]
- Leap Hybrid Sampler (LHS) [210]
- Simulated Annealing (SA) [210]
- Steepest Descent (SD) [210]

D-Wave also provides access to a hybrid sampler, which leverages both classical and quantum calls within its subroutine. This technology is proprietary to D-Wave, and therefore we treat this sampler as a black box, and assume that some component of the subroutine leverages calls to a quantum processing unit. For classical solvers, simulated annealing and steepest descent are relatively straightforward in their implementations and may be reviewed and obtained according to the D-Wave documentation [210].

In our tests, we examined the variation of the parameters α and β to see if there was any change in the quality of the solution according to the minimum energy distributions found by each sampler. Generally, we found that the hybrid sampler achieved the best performance, and the fully quantum samplers were less efficient, as shown in

Figure 4.2 and Figure 4.3, when run over the same set of input QUBOs. This could be attributed to the fully connected nature of the input problem, where the embedding found of the decomposition of the fully connected graph for the quantum samplers may not be optimal with respect to the QPU architecture. Since the hybrid sampler achieved the best results for different values of α and β (Figure 4.2 and Figure 4.3), we used this sampler for our experiments with the OneMax function and IOHProfiler functions. For the OneMax experiments, we define the function following [182], where we optimize the bit string $\mathbf{a}$ that maximizes the function:

$$f(\mathbf{a}) = \sum_{i=1}^{d} a_i \tag{4.4}$$

The optimum $\mathbf{a}'$ is essentially a vector of ones, $\mathbf{a}' = [1, 1, ..., 1]$.

4.4.2 IOHExperimenter

For our experiments, we used the IOHProfiler software library [56] for black-box optimization. IOHProfiler contains a suite of pseudo-Boolean functions with which to benchmark optimization algorithms. The functions selected for our benchmarking include function IDs (*fids*) 4-18. The functions 4-17 are variants of OneMax and LeadingOnes, and are *W-model* transformed, using *dummy variables* (DV), *neutrality* (Neu), *epistasis* (Eps), and *fitness perturbation* (FP) as shown in Table 4.1. Function 18 from IOHProfiler is an instance of the low autocorrelation binary sequence problem, where the fitness is determined by the reciprocal over the sequence's autocorrelation.

4.4.3 Experiment Parameters, Performance Metrics and Quality Diversity

In order to compare selection operators as part of a larger heuristic, we set some parameters within the genetic algorithm to be static across all experiments. For our choice of mutation rate, we used a rate of $m = 0.02$. For our chromosome size, d, we set $d = 50$. For the elitist versions of the classical and quantum-enhanced algorithms, we chose to track the best solution found so far and recombined with the parent pool chosen by the selection operator to create offspring. In the non-elitist versions, we only used the parent pool and recombined with members of the population drawn with uniform probability. For our population size, we chose a size $n = 50$ and examined the change in the size of the selected parent pool μ, in relation to the size of the population, as shown in Figure 4.4. For each operator, we set the number of children

Table of *W-model* transformed objective functions					
FID	function	*DV*	*Neu*	*Eps*	*FP*
4	OneMax	$n/2$	1	1	id
5	OneMax	$0.9n$	1	1	id
6	OneMax	n	3	1	id
7	OneMax	n	4	1	id
8	OneMax	n	1	1	r_1
9	OneMax	n	1	1	r_2
10	OneMax	n	1	1	r_3
11	LeadingOnes	$n/2$	1	1	id
12	LeadingOnes	$0.9n$	1	1	id
13	LeadingOnes	n	3	1	id
14	LeadingOnes	n	4	1	id
15	LeadingOnes	n	1	1	r_1
16	LeadingOnes	n	1	1	r_2
17	LeadingOnes	n	1	1	r_3

Table 4.1: Function transformations from IOHProfiler library [56], with ruggedness functions r_1-r_3 mapping various levels of fitness perturbation.

per parent to the value of n/μ. For settings of α and β, for $QE-(\mu,\lambda)$ we set $\alpha = 100$ and $\beta = 100$ and for $QE-(\mu+\lambda)$ we set $\alpha = 1000$ and $\beta = 10$.

In our experiments, we examined the expected run-time, which we define as the average number of generations to the target solution per run, with a total of 20 runs for each experiment. We set a budget of 50 generations for each experiment. We also took into account the best fitness values found in each generation, which we averaged over all runs.

We measured the quality at each generation by taking the average of the pairwise Hamming distances of all members of the population at each generation and averaging over these in each individual run, finally taking the average for all 20 runs.

4.5 Results

We plotted the results as distributions of energies per sampler on random initialization of populations in Figure 4.2. We plotted the gap observed in the change in the ratio of μ/n vs. average generations to convergence over OneMax in Figure 4.4. We plotted the average fitness and log of the average genotype diversity in the OneMax function in Figure 4.6 and Figure 4.5. We tabulated performance results with quantum-enhanced vs. classical algorithms in Tables 4.2, 4.3, 4.4. We highlight the best performance in

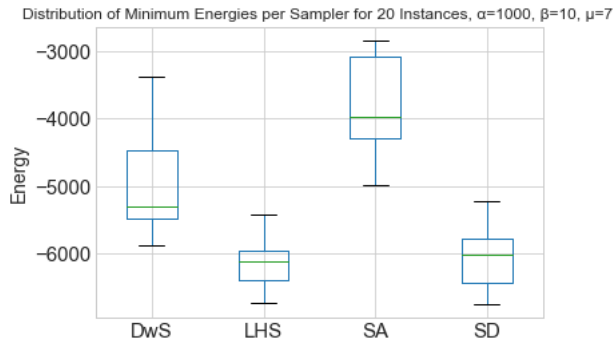

Figure 4.2: Distributions of energies found per sampler on randomly initialized QUBOs, α=1000, β=10, μ=7.

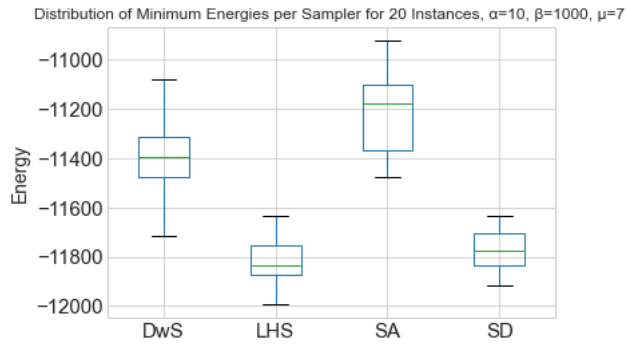

Figure 4.3: Distributions of energies found per sampler on randomly initialized QUBOs, α=10, β=1000, μ=7.

bold, ranked in order of average fitness, average generations to convergence, and average genotype diversity. For *fids* 11,12,13,18 the $QE-(\mu+\lambda)$ operator outperformed other operators on objective functions with increased epistasis and $DV \in [0.5n, 0.9n]$ in terms of fitness and average generations to convergence. For *fids* 7, 10, 14, 15, 16, and 17, the $QE-(\mu,\lambda)$ method outperformed other variants in objective functions with a higher fitness perturbation. We conducted two-sample t-tests comparing quantum-enhanced methods against their classical counterparts; statistically significant differences ($p < 0.05$) were observed in the reported cases. Although we limited the statistical significance testing between classical and quantum enhanced algorithms on individual *fids*, we acknowledge that future work could apply the Bonferroni correction to account for family-wise error rates.

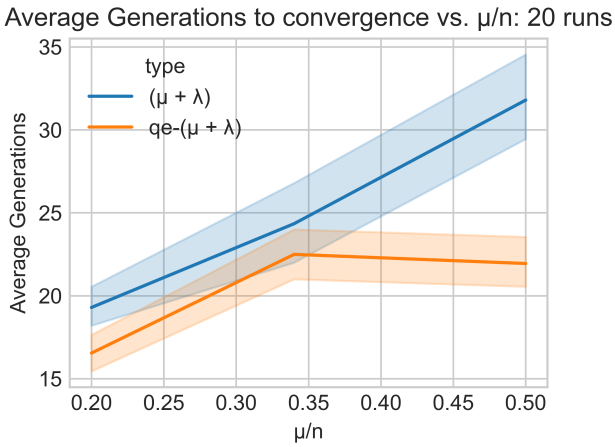

Figure 4.4: Change in average generations to reach global optimum for OneMax function vs. ratio of μ/n.

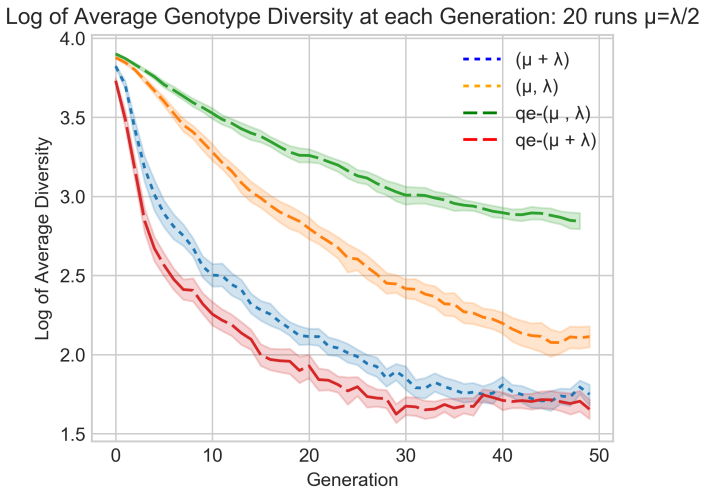

Figure 4.5: Average log genotype diversity for quantum-enhanced vs. classical operators over OneMax objective function.

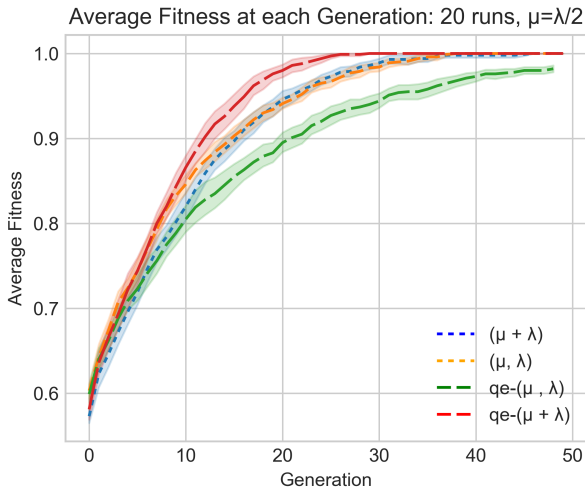

Figure 4.6: Average best fitness for quantum-enhanced vs. classical operators over OneMax objective function.

4.6 Discussion

In examining the performance of the quantum enhanced algorithms, we plotted the average generations to convergence and genotype diversity over the OneMax function, as shown in Figure 4.6 and Figure 4.5. In comparison of the quantum-enhanced methods with their classical counterparts, we see that the quantum-enhanced $QE-(\mu+\lambda)$ achieve a high velocity toward convergence in Figure 4.6, indicating a higher selective pressure with lower average generations to convergence. Taking a closer look at the difference between the average generations for $QE-(\mu+\lambda)$ and its classical counterpart, in Figure 4.4, we notice a gap in expected run-time in generations as the ratio of μ/n approaches 0.5. This could be attributed to the higher weightings of $\alpha = 1000$ vs. $\beta = 10$, where preference is given to fitness within the QUBO of the quantum enhanced selection mechanism. This is likely attributed to the quantum-enhanced mechanism that maintains selective pressure, while also preserving high-quality candidates for recombination as part of the selection mechanism using the QUBO as this ratio approaches 0.5.

We also see that there is a trade-off when comparing the average log genotype diversity of populations as indicated in Figure 4.5, where $QE-(\mu,\lambda)$ shows on average a higher log diversity to its classical counterpart. Surprisingly, when testing the IOH-Profiler suite, we notice that the average genotype diversity is lower for $QE-(\mu,\lambda)$

than (μ, λ). This could also be due to the weighting of the terms $\alpha = 100$ and $\beta = 100$ and the trade-off in optimizing for fitness and lower expected run time vs. diversity. We also see this trade-off when applied to *fids* 4-18 in Tables 4.2-4.4, where the elitist versions achieve higher fitness on functions with dummy variables and neutrality, and non-elitist versions performing well on functions with perturbations of the ruggedness function. This indicates that diversity may help to overcome these perturbations, which may lead searches with higher velocity to converge into local minima, and achieving a proper balance in weightings for terms in the QUBO when optimizing for this criterion is crucial.

In the trade-off of exploration vs. exploitation, for elitist vs. non-elitist algorithms, we note the main difference in recombining the best solution found so far with the selected population in the elitist cases. In the cases where we tuned $\alpha = 1000$, we noticed a decrease in diversity for an increase in velocity. This leads us to believe that there may be some potential in future work toward characterizing the objective function and incorporating a switching mechanism within the optimization routine based upon whether exploitation or exploration may be more or less advantageous.

As we wanted to only examine the effects of the selection operator, we kept the other operators static, but we believe that future work could also examine adaptive population sizes, as well as the effect of adaptive mutation rates, which could help reduce the decreased genotype diversity toward the end of the optimization regime, as observed in Figure 4.5.

4.7 Conclusions

We conclude that our quantum-enhanced selection operator shows some advantages in velocity and exploration within the population selection mechanism, although there is also a trade-off in the latency for compute times on current NISQ chips. In response to the research questions for this thesis, we have shown promising results in response to RQ3. In this case, we find that by leveraging QUBO and hybrid quantum-classical subroutines, we obtain qualitative performance gains when testing over black-box functions with increased perturbation and epistasis. In Chapter 5 we explore using this selection mechanism for evolution strategies for continuous optimization and investigate using the method for hyperparameter optimization and neural architecture search for machine learning. Future work could apply this method to multi-objective optimization. Future work could also incorporate streams of research previously identified in the related works section, such as using Grover's search for global optimization, when

larger QPUs come online. Finally, future work could examine quantum-enhanced metaheuristics when applied to both single- and multi-objective black-box optimization.

4.8 Summary

We present a method of selection for genetic algorithms using a QUBO formulation of the selection operator, and investigate using a quantum annealing system to sample solutions for QUBO in response to RQ3. We observe a balance in the selective pressure and candidate quality for recombination, leading to improved convergence using our method. This may be of benefit toward searching over a hyperparameter space for machine learning models.

For example, as part of this search, members of the population of the quantum-enhanced genetic algorithm may be represented by configurations of model hyperparameters. Machine learning models may be evaluated to determine fitness using these hyperparameters, and this information can be used to formulate a QUBO for selection, as well as perform crossover and mutation. The quantum-enhanced genetic algorithm may be applied in this way and may allow the machine learning practitioner to search the model configuration space of machine learning models more efficiently. We identify this method as a candidate component of a larger AutoML process in response to RQ3 for this thesis.

IOH Function results- ($n = 50$, $d = 50$, 20 trials) Mean Fitness				
IOH *fid*	$QE-(\mu+\lambda)$	$(\mu+\lambda)$	$QE-(\mu,\lambda)$	(μ,λ)
4	25 (+/- 0.0)	**25 (+/- 0)**	25 (+/- 0)	25 (+/- 0)
5	44.95 (+/- 0.21)	**45.0 (+/- 0.0)**	45 (+/- 0.)	45.0 (+/- 0.)
6	16 (+/- 0.0)	15.95 (+/- 0.2)	16 (+/- 0.0)	**16 (+/- 0.0)**
7	43.45 (+/- 1.01)	43.05 (+/- 1.74)	**46.65 (+/- 1.31)**	43.45 (+/-1.65)
8	25.25 (+/- 0.43)	24.85 (+/- 0.57)	25.4 (+/- 0.0)	**25.95 (+/- 0.0)**
9	49.1 (+/- 0.83)	48.9 (+/- 0.7)	49.6 (+/- 0.21)	**49.9 (+/- 0.3)**
10	34.5 (+/- 4.15)	34.5 (+/- 3.1)	**48.0 (+/- 2.0)**	45.5 (+/- 2.29)
11	**25 (+/- 0.0)**	25 (+/- 0.0)	24.3 (+/- 1.3)	23.9 (+/- 1.69)
12	**42.1 (+/- 2.9)**	40.0 (+/- 5.2)	25.45 (+/- 3.2)	23.5 (+/- 3.45)
13	**16 (+/- 0.0)**	16 (+/- 0.0)	16 (+/- 0.0)	16 (+/- 0.0)
14	10.1 (+/- 4.7)	10.25 (+/- 4.14)	**12.35 (+/- 5.8)**	9.65 (+/- 3.2)
15	11.25 (+/- 4.3)	11.8 (+/- 5.3)	**13.3 (+/- 1.9)**	12.15 (+/- 2.15)
16	17.55 (+/- 7.14)	17.4 (+/- 6.0)	**24.95 (+/- 4.2)**	20.85 (+/- 5.1)
17	9.25 (+/- 3.34)	9.5 (+/- 4.15)	**16.9 (+/- 5.1)**	10.75 (+/- 4.26)
18	**4.01 (+/- 0.35)**	3.81 (+/- 0.31)	2.79 (+/- 0.23)	3.09 (+/- 0.32)

Table 4.2: Mean fitness values for 20 runs with a budget 50 generations. Values shown as mean +/- standard deviation over trials.

IOH Function results- ($n = 50$, $d = 50$, 20 trials) Average G				
IOH *fid*	$QE-(\mu+\lambda)$	$(\mu+\lambda)$	$QE-(\mu,\lambda)$	(μ,λ)
4	10.2 (+/- 1.9)	**9.8 (+/- 2.1)**	11.8 (+/- 1.5)	11.85 (+/- 1.95)
5	23.2 (+/- 6.6)	**21.3 (+/- 2.7)**	30.15 (+/- 4.4)	26.55 (+/- 4.9)
6	12.25 (+/- 7.1)	21.3 (+/- 15.0)	9.7 (+/- 2.3)	**8.85 (+/- 2.55)**
7	50 (+/- 0.0)	50 (+/- 0.0)	**50 (+/- 0.0)**	50 (+/- 0.0)
8	46 (+/- 7.45)	49.55 (+/- 1.35)	44.75 (+/- 8.04)	**35.4 (+/- 5.64)**
9	44.2 (+/- 9.5)	48.35 (+/- 5.1)	44.1 (+/- 6.26)	**38.55 (+/- 7.2)**
10	50 (+/- 0.0)	50 (+/- 0.0)	**50 (+/- 0.0)**	50 (+/- 0.0)
11	**17.85 (+/- 4.8)**	20.75 (+/- 5.7)	30.7 (+/- 9.3)	43.55 (+/- 8.17)
12	**48.15 (+/- 3.7)**	45.5 (+/- 5.2)	50 (+/- 0.0)	50 (+/- 0.0)
13	**9.85 (+/- 4.7)**	14.3 (+/- 10.6)	21.65 (+/- 6.5)	22.5 (+/- 9.48)
14	50 (+/- 0.0)	50 (+/- 0.0)	**50 (+/- 0.0)**	50 (+/- 0.0)
15	50 (+/- 0.0)	50 (+/- 0.0)	**50 (+/- 0.0)**	50 (+/- 0.0)
16	50 (+/- 0.0)	50 (+/- 0.0)	**50 (+/- 0.0)**	50 (+/- 0.0)
17	50 (+/- 0.0)	50 (+/- 0.0)	**50 (+/- 0.0)**	50 (+/- 0.0)
18	**50 (+/- 0.0)**	50 (+/- 0.0)	50 (+/- 0.0)	50 (+/- 0.0)

Table 4.3: Average generations (G) to convergence values for 20 runs. Values shown as mean +/- standard deviation over trials.

IOH Function results- ($n = 50$, $d = 50$, 20 trials) Average GD				
IOH *fid*	$QE-(\mu+\lambda)$	$(\mu+\lambda)$	$QE-(\mu,\lambda)$	(μ,λ)
4	0.45 (+/- 0.02)	**0.68 (+/- 0.02)**	0.62 (+/- 0.0)	0.68 (+/- 0.01)
5	0.4 (+/- 0.01)	**0.68 (+/- 0.02)**	0.57 (+/- 0.01)	0.69 (+/- 0.02)
6	0.45 (+/- 0.04)	0.68 (+/- 0.02)	0.63 (+/- 0.01)	**0.69 (+/- 0.02)**
7	0.35 (+/- 0.01)	0.67 (+/- 0.02)	**0.56 (+/- 0.01)**	0.68 (+/- 0.02)
8	0.37 (+/- 0.01)	0.67 (+/- 0.01)	0.57 (+/- 0.01)	**0.67 (+/- 0.01)**
9	0.36 (+/- 0.02)	0.68 (+/- 0.01)	0.55 (+/- 0.01)	**0.68 (+/- 0.02)**
10	0.36 (+/- 0.02)	0.68 (+/- 0.02)	**0.59 (+/- 0.0)**	0.68 (+/- 0.02)
11	**0.45 (+/- 0.02)**	0.68 (+/- 0.01)	0.58 (+/- 0.01)	0.69 (+/- 0.02)
12	**0.40 (+/- 0.01)**	0.68 (+/- 0.02)	0.59 (+/- 0.0)	0.68 (+/- 0.02)
13	**0.48 (+/- 0.04)**	0.69 (+/- 0.01)	0.61 (+/- 0.01)	0.68 (+/- 0.02)
14	0.35 (+/- 0.01)	0.68 (+/- 0.02)	**0.61 (+/- 0.01)**	0.68 (+/- 0.01)
15	0.36 (+/- 0.01)	0.68 (+/- 0.02)	**0.59 (+/- 0.0)**	0.68 (+/- 0.02)
16	0.36 (+/- 0.01)	0.69 (+/- 0.02)	**0.59 (+/- 0.01)**	0.67 (+/- 0.02)
17	0.34 (+/- 0.01)	0.68 (+/- 0.02)	**0.61 (+/- 0.01)**	0.68 (+/- 0.02)
18	**0.34 (+/- 0.01)**	0.69 (+/- 0.01)	0.64 (+/- 0.0)	0.68 (+/- 0.02)

Table 4.4: Average log genotype diversity (GD) to convergence values for 20 runs with a budget 50 generations. Values shown as mean +/- standard deviation over trials.