

Parameter Tuning in Quantum-Inspired Evolutionary Algorithms for Partitioning Complex Networks

Shikha Gupta
Department of Computer Science
University of Delhi
New Delhi, India
shikha.gupta.cs.du@gmail.com

Naveen Kumar
Department of Computer Science
University of Delhi
New Delhi, India
nk.cs.du@gmail.com

ABSTRACT

We propose a numeric variant of quantum-inspired evolutionary algorithm (QIEA) where gene in the quantum chromosome is a superposition of k qubits, thus allowing the genes of the classical chromosome to take numeric values. We also present a modified form of real observation QIEA. Both these techniques are applied to the problem of partitioning a complex network. The algorithm parameters are tuned using an evolutionary bilevel search optimization technique.

Keywords

Genetic Algorithm; Complex Network; Community detection; Quantum-inspired evolutionary algorithm; Modularity; NMI; Evolutionary bilevel Optimization

1. INTRODUCTION

Quantum-inspired evolutionary algorithms (QIEAs) are evolutionary algorithms (EAs) using the quantum computing concepts like quantum bits (qubits), superposition, and unitary transformations of quantum states. A QIEA works with quantum population comprising of chromosomes. Each chromosome is represented as a string of qubits with the advantage that it can represent a linear superposition of states in search space probabilistically. The original QIEA, first implemented by Han and Kim [3], involves measuring the quantum population to generate a classical population followed by an update operation that involves application of quantum gate as variation operator. Survey of extensions and applications of the QIEA [12] categorizes QIEA that are characterized by the qubit representation, measurement process and quantum gates into binary observation QIEA ($bQIEA$), and real observation QIEA ($rQIEA$) based on the type of data assumed in the classical chromosome. Both these variants allow one qubit per gene. In this paper, we propose a numeric variant ($numQIEAm$) that allows the genes of the classical chromosome to take numeric values. Here, every gene in the quantum chromosome is superposition of k qubits. This formulation is useful for combinatorial optimization problems where the data type of a gene in the classical chromosome is numeric. We also present a modified form of $rQIEA$ ($rQIEAm$). Both the

techniques are applied to the problem of partitioning complex networks. The problem is modeled as an evolutionary bilevel optimization problem for parameter tuning [2]. In the rest of the paper, section 2 describes the proposed strategies, section 3 details the experimentation conducted and is followed by concluding remarks.

2. QIEA

The smallest unit of information stored in two-state quantum system is called a qubit which may be in "1" state, "0" state, or in any superposition of "1" and "0" states. The state of a qubit can be represented as $|\Psi\rangle = a|0\rangle + b|1\rangle$, where a and b are complex numbers specifying the probability amplitudes of the corresponding states. $|a|^2$ and $|b|^2$ denote the probabilities that the qubit may be found in states 0 and 1 respectively. The condition $|a|^2 + |b|^2 = 1$ guarantees normalization. Thus, a qubit is defined using a pair of complex numbers (a, b), represented by the vector $[a, b]^T$. The proposed numeric observation QIEA algorithm with

Procedure QIEAm: QIEA with quantum mutation operator

Input: Graph $G = (V, E)$

Output: Best solution B

$t \leftarrow 0$

initialize quantum population $Q(t)$

determine $P(t)$ by measuring $Q(t)$ (Measure function)

repair $P(t)$

evaluate $P(t)$, that is, compute fitness function (network modularity) for each chromosome

store the best solution (that gives a positive gain in network modularity) among $P(t)$ in B

while not termination condition **do**

$t \leftarrow t + 1$

determine $P(t)$ by measuring $Q(t-1)$

evaluate $P(t)$

for all chromosome in $Q(t)$ and $P(t)$ **do**

mutate and repair the chromosome

modify $Q(t)$ using quantum gates (update operation)

update B with the best solution among $P(t-1)$ and $P(t)$

end for

end while

mutation ($numQIEAm$) and the modified real observation QIEA algorithm with mutation ($rQIEAm$), both follow the general steps of the algorithm QIEAm. The specific details for the proposed strategies are elaborated in further sections.

2.1 Binary Observation QIEA

In a binary observation QIEA ($bQIEA$), a chromosome encoding binary representation of the solution uses a representation based on

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2881-4/14/07 ...\$15.00.

<http://dx.doi.org/10.1145/2598394.2605672>.

qubits. Thus, n qubits may be used to represent a *quantum chromosome* of size n as follows,

$$q = \left[\begin{array}{c|c|c|c} a_1 & a_2 & \dots & a_n \\ \hline b_1 & b_2 & \dots & b_n \end{array} \right] \quad (1)$$

Here, the states are in superposition and can be represented as,

$$|\Psi\rangle = a_1 a_2 \dots a_n |00\dots 0\rangle + a_1 a_2 \dots a_{n-1} b_n |00\dots 01\rangle + \\ a_1 a_2 \dots a_{n-2} b_{n-1} a_n |00\dots 010\rangle + \dots + b_1 b_2 \dots b_n |11\dots 1\rangle$$

For example, consider the following 3-qubit chromosome with three pairs of amplitudes,

$$q = \left[\begin{array}{c|c|c} \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} & \frac{-1}{\sqrt{2}} \\ \hline \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{\sqrt{2}} \end{array} \right]$$

Here, the states can be represented as,

$$|\Psi\rangle = \frac{-\sqrt{3}}{4} |000\rangle + \frac{\sqrt{3}}{4} |001\rangle - \frac{1}{4} |010\rangle + \frac{1}{4} |011\rangle \\ - \frac{\sqrt{3}}{4} |100\rangle + \frac{\sqrt{3}}{4} |101\rangle - \frac{1}{4} |110\rangle + \frac{1}{4} |111\rangle$$

This implies that states $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, and $|111\rangle$ occur with probabilities $3/16$, $3/16$, $1/16$, $1/16$, $3/16$, $3/16$, $1/16$, and $1/16$ respectively.

A system of n -qubits can represent 2^n states at the same time but each qubit collapses to a single state 0 or 1 when measured, resulting in a classical binary chromosome. Considering $n=12$, a 12-qubit chromosome

$$a_1|0\rangle + b_1|1\rangle, a_2|0\rangle + b_2|1\rangle, \dots, a_{12}|0\rangle + b_{12}|1\rangle$$

when measured may yield a binary chromosome such as,

$$\boxed{0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0}$$

For the i^{th} qubit in the chromosome, given by probability amplitudes $[a_i, b_i]^T$, a random number r in the range $[0, 1]$ is generated. If $r < |a|^2$, then the corresponding classical bit is set to 0 else to 1. In the context of community detection, the values 0 and 1 in the above chromosome may be interpreted as a community IDs associated with the corresponding nodes. Thus, the above chromosome may be interpreted as,

Node no.	1	2	3	4	5	6	7	8	9	10	11	12
Community ID	0	1	1	1	0	0	1	1	0	1	1	0

That is, nodes 1, 5, 6, 9, and 12 belong to the community 0 while nodes 2, 3, 4, 7, 8, 10, and 11 belong to the community 1.

2.2 Numeric Observation QIEA

The proposed *numQIEAm* algorithm (m for mutation) according to terminology given by Zhang [12], follows the basic steps of the algorithm *QIEAm*. In a numeric observation QIEA (*numQIEAm*), the genes of the classical chromosome take numeric values and genes in the quantum chromosome are each a superposition of k qubits. This allows $nc = 2^k$ possible states for every gene. This formulation is suitable for problems like that of discovering communities in complex networks where each gene in the classical chromosome may take discrete values representing the partition number of the corresponding node. We assume that the partition number values lie in the known range $[1, nc]$, nc being the estimated number of communities, input by the user during algorithm initialization, with a maximum value n , number of nodes in the network. The *quantum chromosome* of size n may be represented with n (k -qubits) as,

$$q = \left[\begin{array}{c|c|c|c} g_1 & g_2 & \dots & g_n \end{array} \right] \quad (2)$$

The i^{th} gene may be represented as,

$$g_i = \left[\begin{array}{c|c|c|c} a_1 & a_2 & \dots & a_k \\ \hline b_1 & b_2 & \dots & b_k \end{array} \right]$$

For example, consider the following 2-qubit gene with two pairs of amplitudes,

$$g = \left[\begin{array}{c|c} \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \\ \hline \frac{1}{\sqrt{2}} & \frac{1}{2} \end{array} \right]$$

Here, the states of a gene are in superposition and can be represented as, $|\Psi_g\rangle = \frac{\sqrt{3}}{2\sqrt{2}} |00\rangle + \frac{1}{2\sqrt{2}} |01\rangle + \frac{\sqrt{3}}{2\sqrt{2}} |10\rangle + \frac{1}{2\sqrt{2}} |11\rangle$. This implies that states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ occur with probabilities $3/8$, $1/8$, $3/8$, and $1/8$ respectively. Renaming the states as 0_d , 1_d , 2_d , 3_d , we can interpret this gene as having 4 possible quantum states, thus allowing a maximum of 4 communities in the network. In general, we can represent the probability amplitudes for state $|\Psi_{g_i}\rangle$ of the k -qubit i^{th} gene g_i as $|\Psi_{g_i}\rangle = [t_{i_1}, t_{i_2}, \dots, t_{i_{nc}}]^T$.

A system of n (k -qubits) can represent $(2^k)^n$ states at the same time and each gene will collapse to a single state between 0 and $nc - 1 = 2^k - 1$ when measured, resulting in a numeric chromosome. For example, a 12 (2-qubit) chromosome represented as, $|\Psi_{g_1}\rangle, |\Psi_{g_2}\rangle, \dots, |\Psi_{g_{12}}\rangle$, further expanded as, $t_{1_1}|0_d\rangle + t_{1_2}|1_d\rangle + t_{1_3}|2_d\rangle + t_{1_4}|3_d\rangle, t_{2_1}|0_d\rangle + t_{2_2}|1_d\rangle + t_{2_3}|2_d\rangle + t_{2_4}|3_d\rangle, \dots, t_{12_1}|0_d\rangle + t_{12_2}|1_d\rangle + t_{12_3}|2_d\rangle + t_{12_4}|3_d\rangle$ when measured may yield a numeric chromosome such as,

$$\boxed{3_d \ 2_d \ 1_d \ 1_d \ 0_d \ 1_d \ 1_d \ 3_d \ 2_d \ 1_d \ 3_d \ 0_d}$$

In the context of community detection, the values 0_d , 1_d , 2_d , and 3_d in the above chromosome may be interpreted as community IDs. That is, nodes 5 and 12 belong to the community 0_d , nodes 3, 4, 6, 7, and 10 belong to community 1_d , nodes 2 and 9 belong to community 2_d , and nodes 1, 8, and 11 belong to the community 3_d .

2.2.1 Quantum Population Initialization

For a quantum chromosome, as given in (2), every k -qubits gene is initialized as:

$$1/\sqrt{nc} |0_d\rangle + 1/\sqrt{nc} |1_d\rangle + \dots + 1/\sqrt{nc} |(nc-1)_d\rangle, \text{ where } nc = 2^k.$$

2.2.2 Measure Function

For the i^{th} gene in the quantum chromosome, given by probability amplitudes $[t_{i_1}, t_{i_2}, \dots, t_{i_{nc}}]^T$, a random number r in the range $[0, 1]$ is generated. If $r < \sum_{i=1}^{nc} |t_{i_j}|^2$, $j = nc, \dots, 1$, then the corresponding classical bit is set to j . Thus, a gene in the classical chromosome takes a value in the range $[1, nc]$.

2.2.3 Mutation

A classical chromosome is selected for mutation with a probability of *mutRate*, an algorithm parameter. For every gene in the selected chromosome, a random number r in the range $[0, 1]$ is generated. If $r < 1/n$, the i^{th} gene is selected for mutation [7]. The selected gene with value x is mutated to value $x' = \lfloor (nc - 1) r \rfloor$. For the i^{th} gene in the corresponding quantum chromosome, the x' -th and x -th amplitudes are interchanged. The chromosome is then repaired, that is, new community ID is assigned to all the neighbors of the node and their corresponding quantum genes are also mutated. The mutation is accepted if the mutated gene in $P(t)$ increases the network modularity.

2.2.4 Repair Function

Tasgin and Bingol (2006) [9] repaired the generated classical chromosome to improve the convergence of the algorithm. A chromosome is randomly picked for repair with probability α and a

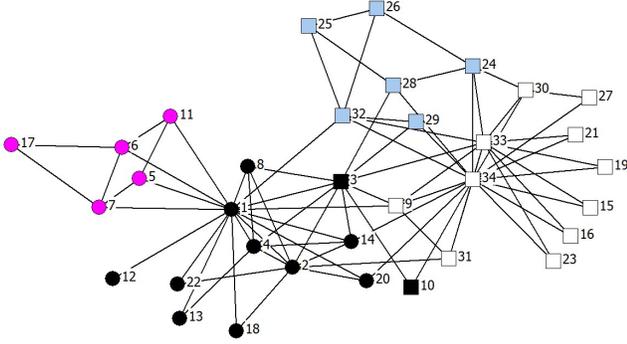


Figure 1: Partitioning structure for Zachary karate club network (NMI=1). Vertex color represents the division into four communities. Vertex shape represents the principle division into two communities

node v is randomly picked with probability β . The community ID of v is then assigned to all of its neighbors in the chromosome. α and β are algorithm parameters.

2.2.5 Update Function

As given by Han and Kim (2000) [3], qubits in $Q(t)$ are updated by applying quantum rotation gate,

$$U(\Delta\theta_i) = \begin{pmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{pmatrix} \quad (3)$$

where, $\Delta\theta_i$ (Table 2), $i = 1, 2, \dots, n$ is the rotation angle of the qubit q_i , towards the fitter state. x_i is the i^{th} gene of the classical chromosome corresponding to the i^{th} quantum gene being mutated. B_i is the i^{th} bit of the best solution (B) achieved so far. Let the x_i^{th} and B_i^{th} probability amplitudes in the quantum gene be t_{x_i} and t_{B_i} respectively. Then the updated probability amplitudes are calculated as $[t'_{x_i}, t'_{B_i}]^T = U(\Delta\theta_i) [t_{x_i}, t_{B_i}]^T$. The magnitude of $\Delta\theta_i$ has an effect on the speed of convergence, but for very large values, the solutions may diverge or converge prematurely to a local optimum.

2.3 Modified rQIEA

The proposed modified *rQIEA* algorithm, called as *rQIEAm* (m for mutation) according to terminology given by Zhang [12], follows the basic steps of the algorithm *QIEAm*. The quantum population in *rQIEAm* is similar to that in *bQIEA*. But the classical population for the two differ in that *rQIEA* allows genes to take real values. Following is the description of modified functions:

Quantum Population Initialization: For the quantum population $Q(t) = \{q_1^1, q_2^1, \dots, q_{pop}^1\}$, $q_i^1(1)$ is initialized as $a_i^1 = 1/\sqrt{2}$, $b_i^1 = 1/\sqrt{2}$, $i = 1, \dots, n$. *pop* being the population size.

Measure: Measure function is modified to suit the problem of community detection. For the i^{th} qubit in the chromosome, given by probability amplitudes $[a_i, b_i]^T$, a random number r in the range $[0, 1]$ is generated. If $r > 0.5$, then the corresponding classical bit is set to $\lfloor (n-1) |b|^2 \rfloor$ else to $\lfloor (n-1) |a|^2 \rfloor$, n being the chromosome size. Thus, a gene in the classical chromosome takes a value in the range $[1, n]$, denoting the community ID of the corresponding node.

Mutation: A chromosome is selected for mutation with a probability of *mutRate*, an algorithm parameter. The i^{th} gene is selected for mutation with probability $1/n$ [7]. The i^{th} gene with value x is then mutated to value $n - x$. The i^{th} qubit in the corresponding quantum

Table 1: Lookup table for quantum rotation gate, applied to i^{th} qubit. Values for θ_1, θ_2 are given in Table 2

x_i	B_i	$f(x) \geq f(B)$	$\Delta\theta_i$
0	0	false	0
0	0	true	θ_2
0	1	false	θ_1
0	1	true	θ_2
1	0	false	$-\theta_1$
1	0	true	$-\theta_2$
1	1	false	0
1	1	true	θ_2

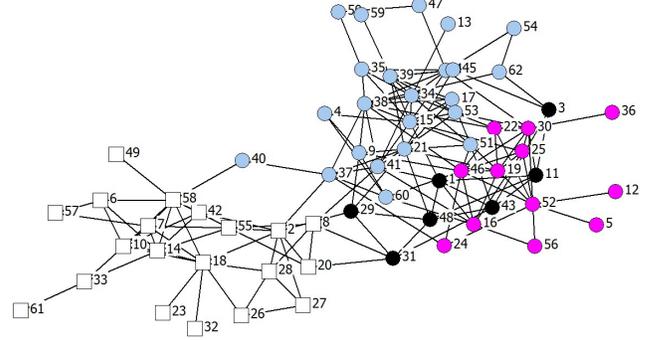


Figure 2: Partitioning structure for Dolphins social network (NMI=1). Vertex color represents the division into four communities. Vertex shape represents the division into two communities

chromosome, given by probability amplitudes $[a_i, b_i]^T$, is mutated to probability amplitudes of $[b_i, a_i]$. The new community ID of the i^{th} gene in the classical chromosome is then propagated to all the neighbors of the node and their corresponding quantum genes are also mutated. The mutation is accepted if the mutated gene in $P(t)$ increases the network modularity.

3. EXPERIMENTAL RESULTS

We make use of MATLAB implementation of NSGA-II as in [8]. The algorithms use the catastrophe operation to avoid premature convergence [10]. We have experimented on two most popular benchmark real-world social networks: Zachary Karate Club Network (Figure 1) [11] and Dolphins Network (Figure 2) [5].

3.1 Partition Similarity Metric

Normalized Mutual Information (NMI) is a popular measure of partition similarity between a pair of partitions [1]. Given a known partitions P_1 (for benchmark datasets) and found partition P_2 (found by proposed algorithm), let $N = (N_{ij})$ be the confusion matrix where the rows and columns correspond to the community structure in P_1 and P_2 respectively and N_{ij} is the number of nodes in the community i that appear in the community j . Then normalized mutual information $NMI(P_1, P_2)$ is defined as:

$$NMI(P_1, P_2) = \frac{-2 \sum_{i=1}^{c_1} \sum_{j=1}^{c_2} N_{ij} \log\left(\frac{N_{ij} n}{N_i N_j}\right)}{\sum_{i=1}^{c_1} N_i \log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{c_2} N_j \log\left(\frac{N_j}{n}\right)} \quad (4)$$

where c_1 and c_2 are the number of communities in P_1 and P_2 respectively and n is the number of nodes in the network. NMI values range between 0 and 1, 1 when P_1 and P_2 are exactly same.

Table 2: Parameter Tuning- NC and NC_I give the number of communities in found and known partitions, Q and Q_I give the modularity of found and known partitions, Misclass. Nodes: Nodes in the known partition that are misclassified in the found partition.

Dataset	Algorithm	Q_I	NC_I	Parameter Solution					Q	NMI	NC	Misclass. Nodes
				θ_1	θ_2	α	β	mutRate				
				Zachary	numQIEAm $k=2$.4188	4	.061				
				.0074	.0038	.52	.13	.13	.4188	1.0	4	none
				.083	.0045	.13	.46	.63	.4188	1.0	4	none
	rQIEAm	.4188	4	.034	0	.51	.25	.78	.3975	.825	4	12
				.054	.0002	.062	.3	.6	.36	1	2	none
		.36	2	.055	.0047	.43	.37	.025	.36	1	2	none
				.035	.0038	.77	.2	.31	.36	1	2	none
				.034	0	.51	.25	.079	.36	1	2	none
Dolphins	numQIEAm $k=3$.5191	4	.033	.0036	.36	.21	.8	.5268	.7830	4	4,9,40,54,60,62
				.51	.005	.31	.22	.033	.5191	1.0	4	none
				.0074	.0038	.52	.13	.13	.5191	1.0	4	none
	rQIEAm	.5191	4	.028	.0027	.49	.41	.23	.5106	.7473	4	3,4,9,21,40,51,63
				.073	.0026	.6	.49	.91	.5147	.7731	4	4,9,21,37,40,51,63
				.03	.0028	.53	.4	.19	.506	.8643	6	13, 36, 40, 51
				.072	.0026	.59	.49	.11	.5064	.8615	6	13, 40, 51, 61

3.2 Parameter Tuning

In evolutionary optimization, bilevel programming problems are those in which an upper level solution is feasible only if it is one of the optimum of a lower level optimization problem. The efficiency of the algorithms proposed in this paper depend on the algorithm parameters. The parameter tuning is an essential task that can be posed as lower level optimization task in a bilevel optimization framework. Community detection problem is then the upper level optimization task and requires searching among the solutions obtained from the lower level task to find an optimal solution corresponding to one or more different (higher level) objectives [2]. Thus, we model our task as an evolutionary bilevel optimization problem where the lower level task tunes the algorithm parameters- θ_1 , θ_2 , α , β , $mutRate$ by using NSGA-II for multi-objective optimization of network modularity and Normalized Mutual Information (NMI) [1]. A good set of parameters will not only maximize modularity but also maximize NMI for benchmark datasets. The results obtained are shown in Table 2.

4. CONCLUSION AND FUTURE WORK

The paper presents novel variants of quantum-inspired genetic algorithms and applies the techniques to the problem of detecting communities in complex networks. The algorithms do not require any decoding step at the end to realize the communities from the solution chromosomes. Experiments on real life benchmark networks show that the methods are able to successfully reveal community structure with high modularity. We plan to study the scalability of approach in future. The proposed QIEA variants may also be tested on other combinatorial optimization problems. We also intend to explore a parallel implementation with a view to reduce the computation time as the quantum-inspired algorithms lend themselves naturally to parallel computation.

5. ACKNOWLEDGEMENTS

The work is supported by University of Delhi research grant no. DRCH/R&D/2013-14/4155.

6. REFERENCES

- [1] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.
- [2] K. Deb and A. Sinha. Solving bilevel multi-objective optimization problems using evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 110–124. Springer, 2009.
- [3] K.-H. Han and J.-H. Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 2, pages 1354–1360. IEEE, 2000.
- [4] E. A. Leicht and M. E. Newman. Community structure in directed networks. *Physical review letters*, 100(11):118703, 2008.
- [5] D. Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(Suppl 2):S186–S188, 2003.
- [6] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.
- [7] G. Ochoa. *Error thresholds and optimal mutation rates in genetic algorithms*. PhD thesis, 2000.
- [8] A. Seshadri. Nsga - ii: A multi-objective optimization algorithm, July 2009.
- [9] M. Tasgin and H. Bingol. Community detection in complex networks using genetic algorithm. In *ECCS '06: Proc. of the European Conference on Complex Systems*, Apr. 2006.
- [10] L. Wang, H. Wu, F. Tang, and D.-Z. Zheng. A hybrid quantum-inspired genetic algorithm for flow shop scheduling. In *Advances in Intelligent Computing*, pages 636–644. Springer, 2005.
- [11] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.
- [12] G. Zhang. Quantum-inspired evolutionary algorithms: a survey and empirical study. *Journal of Heuristics*, 17(3):303–351, 2011.