A Generalized Quantum-Inspired Evolutionary Algorithm for Combinatorial Optimization Problems

Julio M. Alegría [1]
julio.alegria@ucsp.pe

Yván J. Túpac [1]
ytupac@ucsp.pe

[1] School of Computer Science
Universidad Católica San Pablo (UCSP)
Arequipa, Peru

2014, September 24

Table of contents

Table of contents

Introduction
Optimization and Heuristics

Universidad Católica San Pablo

Optimization: means finding the maximum or minimum of a certain function.

Let f be a function:

$$\begin{aligned} f: \quad \mathbb{X} \quad &\mapsto \quad \mathbb{R} \\ \mathbf{x} \quad &\mapsto \quad f(\mathbf{x}) \end{aligned} \tag{1}$$

The optimization problem consists in finding:

$$Y^* = \max(\min)\{f(\mathbf{x})\} \tag{2}$$

When the domain of this function is discrete and finite we call it **combinatorial optimization**.

Heuristics and **Metaheuristics:**
Search methods designed to provide sufficiently good solutions even if the optimality of the problem is not guaranteed.

Introduction
Evolutionary Algorithms

Evolutionary Algorithms (EA) are metaheuristic optimization algorithms based on the Neo-Darwinian Paradigm of the Evolution (Neo-Darwinism)

EA Advantages

- Robust and global.
- Highly Parallel.
- Easily adaptable.

EA Disadvantages:

- Evaluation function can be computationally expensive
- Adjusting and Validation can be computationally expensive.

Evolutionary Algorithms

EA Basic Cycle

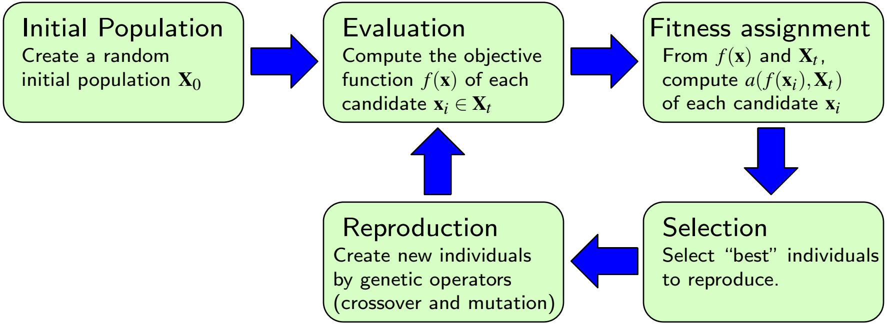


Figure 1: Evolutionary algorithm basic cycle.

Quantum Computing
Definitions

Computing that makes direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data, as a searching.

The main perspective of quantum computing is the proceesing power and consideering:

"Possibilities count, even if they never happen" [Spector, 2004].

In Quantum Computing, the smallest unit of information is the *Q-bit* o *qubit*, which may be in state $|0\rangle$, $|1\rangle$ or in any superposition of the two [Hey, 1999]:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3}$$

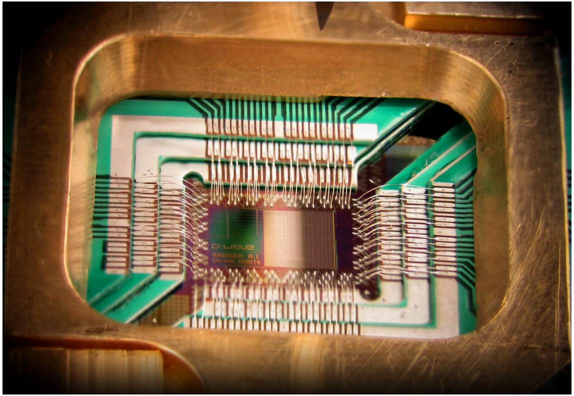
where α and β are amplitudes and:

- $|\alpha|^2$ is the probability of finding the Q-bit in state "0".
- $|\beta|^2$ is the probability of finding the Q-bit in state "1".

Quantum Computing
Some difficults

Main problems to addressing in Quantum Computing:

- Difficulty in implementing a real quantum computer.
- Difficulty in creating algorithms that exploit the processing power of these machines.
- Controlling or removing the *quantum decoherence*.



Quantum Computing
Quantum-Inspired Algorithms

Quantum-Inspired Evolutionary Algorithms:
Classical algorithms (capable of being run on classical computers) that use the principles of quantum computing, in order to improve their performance in problem solving [Moore and Narayanan, 1995].

A Quantum-inspired algorithm must comply the following:

- A numerical representation
- An initial configuration and ending condition
- Divide the problem in more simpler sub-problems
- Identifying the universes (superposition states)
- Associate each problem to an universe
- Computation are independent in each universe
- Some interaction between universes.

Table of contents

Quantum-Inspired Evolutionary Algorithm

Quantum-Inspired Evolutionary Algorithm (QIEA) proposed by [Han and Kim, 2002] addressed the problem of computing time by reducing the size of the population.
QIEA, is an Evolutionary Algorithm composed by individuals, an evaluation function, and a population dynamic:
Instead classical evolutionary algorithm, this model uses:

- A quantum-binary representation of individuals, using Q-bits.
- A variation operator called Q-gate.
- An observation process.

QIEA
Representation

A Q-bit is the smallest unit of information in QIEA, which is defined using a pair of numbers (α, β) as shown in equation 4.

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{4}$$

where the following identity:

$$|\alpha|^2 + |\beta|^2 = 1$$

is satisfied.
A Q-bit individual is a string of Q-bits defined as shown in equation 5.

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{array}\right] \tag{5}$$

QIEA
Q-bit Representation

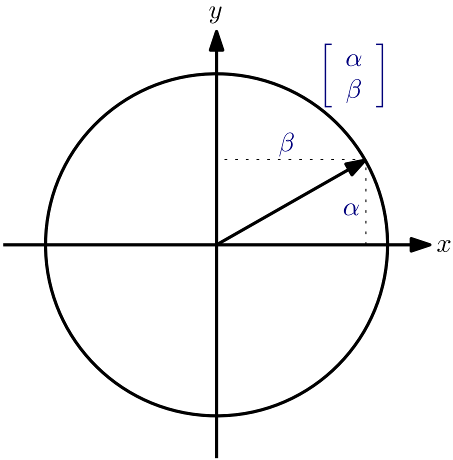


Figure 2: Graphical representation of a circle of unit radius with probability amplitudes to obtain the values 0 and 1 for any Q-bit. **Source:** [Abs da Cruz et al., 2007].

QIEA
Operator

The population updating is performed by a variation operator called **Q-gate**. This operator is defined as a rotation matrix as shown in equation 6.

$$\mathbf{U}(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \quad (6)$$

applied to every column of the quantum individual $\mathbf{q}_i$:

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = \mathbf{U}(\Delta\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$$

and ensuring compliance of the identity $|\alpha_k'|^2 + |\beta_k'|^2 = 1$

QIEA
Operator

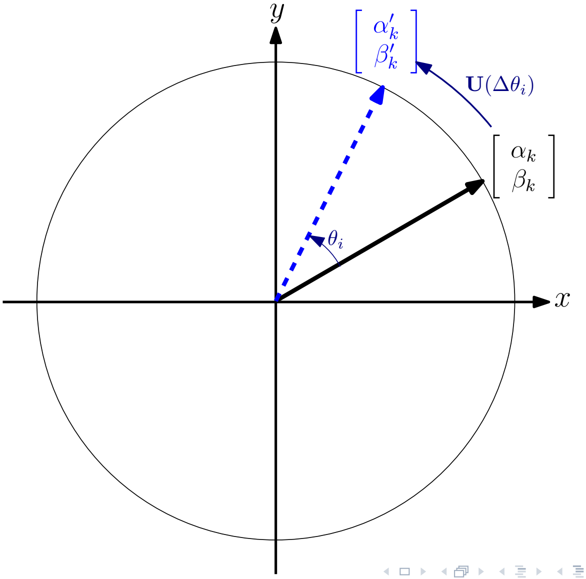


Figure 3: Graphic representation of the rotation of a Q-bit using the Q-gate operator.

QIEA
Algorithm

Algorithm 1 QIEA as proposed in [Han and Kim, 2002].

$t \leftarrow 0$
Initialize Q_t
while not stop-condition **do**
  $t \leftarrow t + 1$
  Generate P_t oberserving the states of Q_{t-1}
  Evaluate P_t
  Store best solution in b_t
  $b_t \leftarrow$ best between b_t and b_{t-1}
  $Q_t \leftarrow$ Q-gate(Q_{t-1})
end while

QIEA
Algorithm

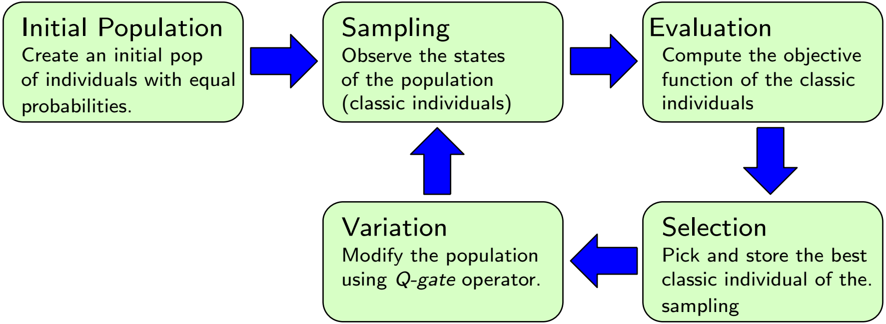


Figure 4: Basic cycle of QIEA.

QIEA
More works on QIEA

Some authors have proposed and implemented other models based on QIEA:

- [Chung et al., 2011] proposes a new initialization method for the QIEA, using priority lists to ensure greater diversity in the initial search area, applying the propose to *unit commitment* problems.
- In [Leon, 2012], a variation of the QIEA is presented which includes an elitism and mutation process of Q-bits. Also, a hill-climbing phase is performed at the end of each cycle. This variation is applied in multi-lateral negotiations on multi-agent systems.
- [Tsai et al., 2012] combines the principles of QIEA with the "Latin squares" to obtain a systematic reasoning mechanism to exploit the best solutions in a search space.
- [Ryu and Kim, 2012] presents a Multiobjective QIEA or MQIEA, which searches a pareto-optimal solution for multiobjective problems.

And a lot of authors have implemented applications using the QIEA of [Han and Kim, 2002]

Table of contents

GQIEA
Generalized Quantum-Inspired Evolutionary Algorithm

All previous works are basis on quantum-binary representation of the QIEA.

Disadvantages:

- Excessive increase of dimensionality.
- The need for a complex coding of solutions.

GQIEA: Generalized Quantum-Inspired Evolutionary Algorithm GQIEA uses not a quantum-binary, but a "quantum-combinatorial" representation. This representation bring us the following advantages:

- Reduce the dimensionality of individuals.
- Simpler coding of individuals.
- Populations with less individuals, which increases efficiency and reduces the algorithm convergence time.

GQIEA

Representation: The generalized q-bit

A generalized Q-bit or GQ-bit is the smallest unit of information in GQIEA, which is defined as a n-uple, where n is the total number of possible states $|S|$ in which the GQ-bit can be found, as shown in equation 7.

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{7}$$

GQIEA
Representation

A GQ individual is a string of GQ-bits defined as shown in equation 8.

$$\left[\begin{array}{c|c|c|c} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,1} & \alpha_{n,2} & \cdots & \alpha_{n,m} \end{array}\right] \tag{8}$$

Notice the analogy with QIEA quantum individual.

GQIEA
Operators

GQIEA also defines a variation operator called generalized Q-gate or GQ-gate. This operator is responsible of driving individuals toward optimal solutions.

Two type of GQ-gate are proposed:

- Arithmetic GQ-gate.
- Geometric GQ-gate.

Let the j-th GQ-bit of a GQ-bit individual, represents the superposition of n states $S = \{S_1, S_2, \ldots, S_n\}$. Let the state S_ϕ be the state observed on the best individual, then the new probabilities of the GQ-bit will be determined as follows:

Arithmetic GQ-gate:

$$\alpha_i = \begin{cases} \alpha_i + \Delta_t & \text{si } i = \phi \\ \alpha_i & \text{si } i \neq \phi \end{cases} \quad , \quad i = 1, 2, \ldots, n \tag{9}$$

Geometric GQ-gate:

$$\alpha_i = \begin{cases} \alpha_i \cdot \Delta_t & \text{si } i = \phi \\ \alpha_i & \text{si } i \neq \phi \end{cases} \quad , \quad i = 1, 2, \ldots, n \tag{10}$$

In both operators, a normalization is performed after the α_i updating, to ensure the compliance of $\sum_i \alpha_i = 1.0$

GQIEA
Algorithm

Algorithm 2 GQIEA proposed in this work.

$t \leftarrow 0$
Initialize Q_t
while not stop-condition **do**
  $t \leftarrow t + 1$
  Generate P_t observing the states of Q_{t-1}
  Evaluate P_t
  Store best solution in b_t
  $b_t \leftarrow$ best between b_t and b_{t-1}
  $Q_t \leftarrow$ GQ-gate(Q_{t-1}, b_t, Δ_t)
end while

Table of contents

Case Study
Assignment Problem

The Assignment Problem (AP) seeks the optimal match between two or more sets. Usually there are two sets: the “tasks” (actions to be assigned) and the “agents” (people or machines) [Pentico, 2007].

Task BGAP: Variation of the AP where multiple tasks can be assigned per agent, and where the goal is to minimize the maximum cost of the assignments made [Mazzola and Neebe, 1988].

$$\text{Minimize} \quad \max_{\forall a \in A} \sum_{\forall t \in T} c_{a,t} \cdot x_{a,t} \tag{11}$$

Case Study
Assignment Problem: Using arithmetic GQ-gate

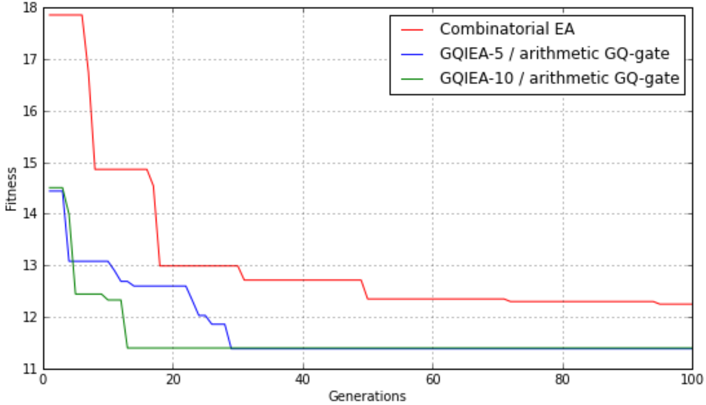


Figure 5: Comparison between the EA for Combinatorial Optimization and two instances of the GQIEA using arithmetic GQ-gate applied to AP (*Task BGAP*).

Case Study
Assignment Problem: Using arithmetic GQ-gate

Algorithm(s)	Parameter	Value
EA for Combinatorial Optimization	Population size	100
EA for Combinatorial Optimization	Selection method	Roulette wheel
EA for Combinatorial Optimization	Crossover	Uniforme
EA for Combinatorial Optimization	Probability of crossover	0.65
EA for Combinatorial Optimization	Probability of mutation	0.08
GQIEA-5	Population size	5
GQIEA-10	Population size	10
GQIEA-5, GQIEA-10	GQ-Gate	Arithmetic
GQIEA-5, GQIEA-10	$\triangle$	0.2

Table 1: Parameters used in the algorithms in Figure 5.

Case Study
Assignment Problem: Using geometric GQ-gate

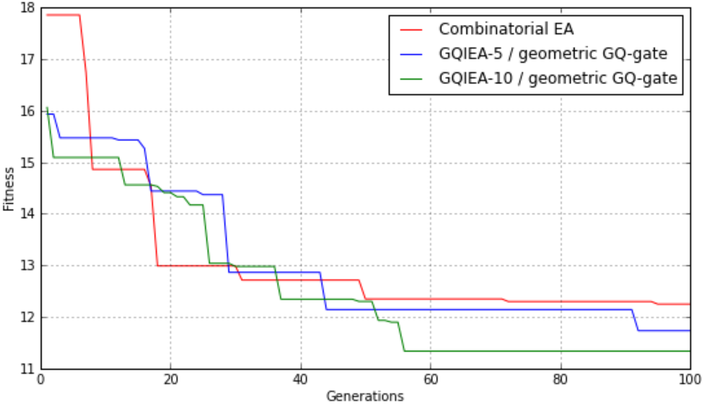


Figure 6: Comparison between the EA for Combinatorial Optimization and two instances of the GQIEA using geometric GQ-gate applied to AP (*Task BGAP*).

Case Study

Assignment Problem: Using geometric GQ-gate

Algorithm(s)	**Parameter**	**Value**
EA for Combinatorial Optimization	Population size	100
EA for Combinatorial Optimization	Selection method	Roulette wheel
EA for Combinatorial Optimization	Crossover	Uniform
EA for Combinatorial Optimization	Probability of crossover	0.65
EA for Combinatorial Optimization	Probability of mutation	0.08
GQIEA-5	Population size	5
GQIEA-10	Population size	10
GQIEA-5, GQIEA-10	GQ-Gate	Geometric
GQIEA-5, GQIEA-10	$\triangle$	1.1

Table 2: Parameters used in the algorithms in Figure 6.

Case Study
Knapsack Problem

The Knapsack Problem (KP) consists of finding the optimum quantity of objects $\{x_i\}_{i=1}^{n}$ that can be included in a container with a maximum weight, while optimizing the total value of the included values, as long as the total weight does not surpass the maximum.
In original KP [] the goal is to maximize the cumulative value of the included objects, given a maximum weight W, as shown in Eq.12

$$\begin{aligned} \text{Maximize} \sum_{\forall k \in K} v_k x_k, \quad x_k \in \{0,1\} \\ \text{such that} \sum_{\forall k \in K} w_k x_k \leq W \end{aligned} \tag{12}$$

where

v_k, w_k are the value and weight of object x_k

x_k indicates the presence or absence of object k ($x_k = 1, 0$ respectively)

Case Study
Knapsack Problem

Tests performed over the KP generalized version: Bounded Knapsack problem, allowing several copies of the same k object, bounded by maximum quantity m_k, as Eq. 13:

$$\begin{aligned} \text{Maximize} \sum_{\forall k \in K} v_k x_k, \quad x_k \in \{0, 1, \ldots, m_k\} \\ \text{such that} \sum_{\forall k \in K} w_k x_k \leq W \end{aligned} \tag{13}$$

And comparing the performance of:

- Classical combinatorial evolutionary algorithm, where each individual is represented as $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, being n the total objects quantity.
- QIEA using quantum individuals (based on Q-bits). Each q-bit is related to the probability of object k be placed into the container.
- The proposal of this work, using an n-String of GQ-Bits,

Case Study
Knapsack Problem: parameters

- The Combinatorial EA uses the following parameters: selection method (roulette wheel, tournament, stochastic remainder), crossover method (one-point, two-point, uniform), mutation method (disorder, change), crossover coefficient $[0.4, 0.9]$, and mutation coefficient $[0.02, 0.2]$.
- QIEA uses only one parameter: theta $[0.01\pi, 0.2\pi]$.
- GQIEA uses just one parameter: Δ, both for arithmetic and geometric GQ-gates.

Universidad Católica San Pablo

Case Study

Knapsack Problem: comparison

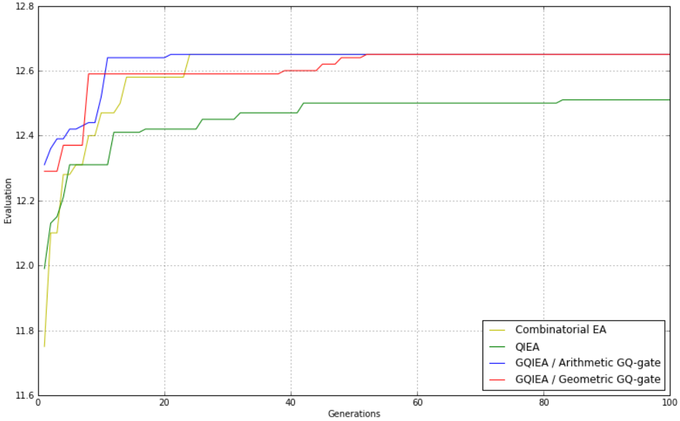


Figure 7: Comparison of the Combinatorial EA, the QIEA and the GQIEA, applied to the Bounded Knapsack Problem..

Case Study

Knapsack Problem: comparison

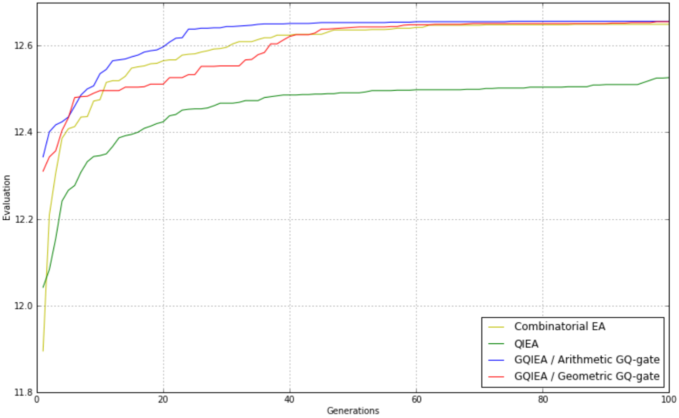


Figure 8: Comparison of an average of 10 executions of the Combinatorial EA, the QIEA and the GQIEA, applied to the Bounded Knapsack Problem.

Case Study
Knapsack Problem: results

Table 3: Results of the algorithms applied to the Bounded Knapsack Problem

Algorithm	Result	Runtime
Combinatorial EA	12.65	145.962 s
QIEA	12.51	49.924 s
GQIEA (Arithmetic QGate)	12.65	31.970 s
GQIEA (Geometric QGate)	12.65	36.970 s

- Three of four algorithms reach the same final result, while one (QIEA) was left behind.
- The algorithm with the slowest runtime was the Combinatorial EA, as it was expected.
- We can also see that both GQIEA had faster runtimes than the QIEA. This because there was a dimensionality reduction.

Table of contents

Conclusions

- A new evolutionary model was proposed, which generalizes the QIEA. The representation of individuals allows the superposition of solutions conformed by any set of finite states, in order to improve the performance achieved by the QIEA in combinatorial optimization problems with expensive evaluation function.
- Two variation operators responsible for driving the individuals to best solutions were proposed, called GQ-gate operators: arithmetic GQ-gate and geometric GQgate.
- The proposed model was compared with two algorithms used in combinatorial optimization: the Combinatorial EA and the QIEA. Outcomes show that the GQIEA has a good quality of results and convergence, improving in most cases the performance of the algorithms with which it was compared.
- GQIEA has better results than traditional Combinatorial Optimization EA, even with a much smaller population in both case studies.

Table of contents

Bibliography I

[Abs da Cruz et al., 2007] Abs da Cruz, A. V., Pacheco, M. A. C., and Vellasco, M. M. B. R. (2007).
Quantum-Inspired Evolutionary Algorithms for Problems based on Numerical Representation.
PhD thesis, Pontificia Universidade Catolica do Rio de Janeiro.

[Chung et al., 2011] Chung, C., Yu, H., and Wong, K.-P. (2011).
An advanced quantum-inspired evolutionary algorithm for unit commitment.
Power Systems, IEEE Transactions on, 26(2):847–854.

[Han and Kim, 2002] Han, K.-H. and Kim, J.-H. (2002).
Quantum-inspired evolutionary algorithm for a class of combinatorial optimization.
Evolutionary Computation, IEEE Transactions on, 6:580–593.

[Hey, 1999] Hey, T. (1999).
Quantum computing: an introduction.
Computing & Control Engineering Journal, 10:105–112.

[Leon, 2012] Leon, F. (2012).
Real-valued quantum-inspired evolutionary algorithm for multi-issue multi-lateral negotiation.
In *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on*, pages 41–48.

Bibliography II

[Mazzola and Neebe, 1988] Mazzola, J. and Neebe, A. (1988).
Bottleneck generalized assignment problems.
Engineering Costs and Production Economics, 14:61–65.

[Moore and Narayanan, 1995] Moore, M. and Narayanan, A. (1995).
Quantum-inspired computing.

[Pentico, 2007] Pentico, D. W. (2007).
Assignment problems: A golden anniversary survey.
European Journal of Operational Research, 176:774–793.

[Ryu and Kim, 2012] Ryu, S.-J. and Kim, J.-H. (2012).
Distributed multiobjective quantum-inspired evolutionary algorithm (dmqea).
In *Robot Intelligence Technology and Applications 2012*, pages 663–670.

[Spector, 2004] Spector, L. (2004).
Automatic Quantum Computer Programming, volume 7 of *Genetic Programming*.
Springer US.

[Tsai et al., 2012] Tsai, J.-T., Chou, J.-H., and Ho, W.-H. (2012).
Improved quantum-inspired evolutionary algorithm for engineering design optimization.
Mathematical Problems in Engineering, 1:27.

A Generalized Quantum-Inspired Evolutionary Algorithm for Combinatorial Optimization Problems

Julio M. Alegría [1]
julio.alegria@ucsp.pe

Yván J. Túpac [1]
ytupac@ucsp.pe

[1] School of Computer Science
Universidad Católica San Pablo (UCSP)
Arequipa, Peru

2014, September 24

Universidad Católica San Pablo