Quantum-Inspired Algorithms for Covering Arrays

Dimitris E. Simos, Michael Wagner, Ludwig Kampel

Covering Array Optimization

Covering Arrays
- Covering Arrays (CAs) are combinatorial structures used in Combinatorial Testing.
- They guarantee that every $t$-way combination appears in at least one row (test).
- A uniform, binary Covering Array is denoted as $CA(N; t; k)$, where $N$ is the number of rows, $t$ the strength and $k$ the number of columns.
- CAs with the smallest number of rows possible are called optimal CAs.

Quantum-Inspired Evolutionary Algorithms
- First quantum-inspired evolutionary algorithm for CA generation.
- We introduced and evaluated new Mutation and Rotation types.
- We were able to generate various optimal binary CAs for strengths $t = 2, 3, 4$.

Algorithm 1 QEAforCA($n, k, N$)

1. Create $n$-qubit representation $\ket{\Psi}$ of a CA
2. Create candidate solution $\ket{\Phi}$ by observing $\ket{\Psi}$
3. Evaluate $\ket{\Phi}$ based on the number of covered $t$-way interactions
4. $\Phi_n = \Phi(t)$
5. while not termination condition $\Phi_n$
6. $n = n - 1$
7. Create $\Phi_n$ by observing $\ket{\Psi}(n - 1)$
8. if Function $\Phi_n$(true)
9. $\Phi_n = \Phi(t)$
10. end if
11. for all Qubits $q_i$ in $\Phi_n$
12. $q_i = \text{Mutation}(q_i)$
13. $\Phi_n = \text{Rotate}(\Phi_n)$
14. end for
15. end while
16. return $\Phi_n$}

Figure 1: Qubit representation and the reduced version for real-valued amplitudes

Covering Array Generation using IPO-Q
- Combines QEA with the In-Parameter-Order strategy:
  - Expands array using vertical and horizontal extension steps:
    - The blue Qubits in Figure 3 represent the CA from the previous extension step, their state biased towards their old value
    - A new column is added (red Qubits) and QEA attempts to find a CA with the newly added column
    - If QEA fails to generate a CA, additional rows are added (green Qubits)

IPO-Q Evaluation and Future Work
- Guaranteed CA upon termination
- Improved on other IPO variants by reducing the number of rows for certain binary CA instances:

Future Work:
- Generalize our QEA and IPO-Q for higher alphabets.
- Use Quantum Computing to solve Covering Array problems.