Quantum Circuit Optimisation: Exploring Reduction Methods with Higher-Order Terms: Supplementary Material

[redacted]

[redacted]

[redacted]

[redacted]

***Abstract*—With reference to the main paper, we go into more detail about global properties of the graph structure.**

***Index Terms*—Quantum Computing, QAOA, Graphs, Pseudo Boolean Function, QUBO, PUBO**

I. Graph properties

Fig. 1 gives an overview of the graphs and their relation to the polynomials in an iteration step t: A graph is created from a Pseudo-Boolean Function (PBF) $f_t : \{0,1\}^n \to \mathbb{R}$. When a reduction step takes place, a new graph G_{t+1} emerges from the (partly) reduced PBF f_{t+1}. However, it is unclear, how the previous graph $G_t(V_t, E_t)$ relates to the new graph $G_{t+1}(V_{t+1}, E_{t+1})$. We therefore identify that the total size of multi-edges will strictly decrease. Furthermore, we discover that a node's degree will not increase during the reduction.

For the following analysis, we presuppose an algorithm that only chooses multi-edges $\{v_i, v_j\}^\beta, \beta > 1$ and terminates when there are no multi-edges left. Recall that any of the proposed variable selection types (*i.e.*, *Sparse*, *Medium* and *Dense*) are allowed to choose multi-edges. In fact, the *Dense* type always selects a multi-edge from $\{\{v_i, v_j\}^{\beta_1} | \ \forall \{v_a, v_b\}^{\beta_2} \in E_t : \beta_1 \geq \beta_2\}$, since it searches for a pair that appears most among all monomials. We denote the number of edges between two nodes by a superscript. For example, $\{v_a, v_b\}^\beta$ represents $\beta \in \mathbb{N}$ edges between nodes v_a and v_b. Iff $\beta > 1$, we call $\{v_a, v_b\}^\beta$ a multi-edge. If $\beta = 1$, we usually omit the superscript. We extend the formal definition of P_S to allow for monomials in S. For example, $P_{\{x_1 x_2 x_3\}} = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}$.

$$\begin{tikzcd}
G_t(V_t, E_t) \arrow[r, dashed, "\text{Inductive step}"] & G_{t+1}(V_{t+1}, E_{t+1}) \\
f_t(x_0, \ldots, x_n) \arrow[u] \arrow[r, "\text{Reduction}"] & f_{t+1}(x_0, \ldots, x_n, y_t) \arrow[u]
\end{tikzcd}$$

Figure 1: Schematic overview of a multi-graph evolution of a single reduction step. The dashed arrow locates the following theorems.

More formally, we start at an undirected multigraph $G_0(V_0, E_0)$, which emerges from the starting polynomial f_0. An edge $\{v_a, v_b\} \in E_0$ if x_a and x_b occur in the same monomial in the starting polynomial $f_0(x_1, \ldots, x_n)$, where the variables represent the nodes V_0 in the graph. A reduction step then leads to a new undirected multigraph $G_1(V_1, E_1)$. Inductively, a reduction step creates f_{t+1} from f_t and therefore creates $G_{t+1}(V_{t+1}, E_{t+1})$, until there are no more multi-edges left: $e = \{v_a, v_b\}^\beta, \beta \leq 1 \ \forall e \in E$. By $\deg_{G_t}(v_i)$, we denote the degree of node v_i in the graph G_t. We further assume that any polynomial is simplified, that is monomials, which contain the same variables, are totalled. For example, $f_1(x_1, x_2) = 2x_1x_2 - 4x_1x_2 = -2x_1x_2$.

Theorem 1. *The total size of multi-edges in the corresponding graphs will strictly decrease with every reduction step $t \mapsto t+1$:*

$$\sum_{\substack{e=\{v_i,v_j\}^\beta \in E_t, \\ \beta > 1}} \beta < \sum_{\substack{e=\{v_i,v_j\}^\beta \in E_{t+1}, \\ \beta > 1}} \beta. \tag{1}$$

Furthermore, the degree of nodes will not increase with every reduction step:

$$\deg_{G_t}(v_i) \geq \deg_{G_{t+1}}(v_i) \ \forall v_i \in V_t. \tag{2}$$

Take into consideration, that this does not characterise the degree of the newly introduced variable in the current step: $\deg_{G_{t+1}}(y_t), y_t \in V_{t+1} \setminus V_t$.

Proof. Let $G_t(E_t, V_t)$ be the corresponding graph to $f_t(x_1, \ldots, x_n)$. Let $x_i x_j$ be the variable pair that is going to be reduced by the new variable y_t. Let $\{v_i, v_j\}^\beta \in E_t, \beta \geq 2$ be the corresponding multi-edge. The following proof examines the effect of reduction for an arbitrary monomial m in f_t. At first, we proof that the degree of any node will not increase during a reduction step: $\deg_{G_t}(v_x) \geq \deg_{G_{t+1}}(v_x) \forall v_x \in V_t$. For this, we classify a monomial m in f_t into one of four categories:

- C1: m is not part of f_t: $m \notin f_t$.
- C2: m is part of f_t and neither contains x_i nor x_j: $m \in f_t \wedge x_i \notin m \wedge x_j \notin m$.
- C3: m is part of f_t and contains either x_i or x_j: $x_i \in m \dot{\vee} x_j \in m$.
- C4: m is part of f_t and contains x_i and x_j: $x_i \in m \wedge x_j \in m$.

Take into consideration that these categories depict all possible monomials of f_t and are pairwise disjoint. Fig. 2 shows their structure.

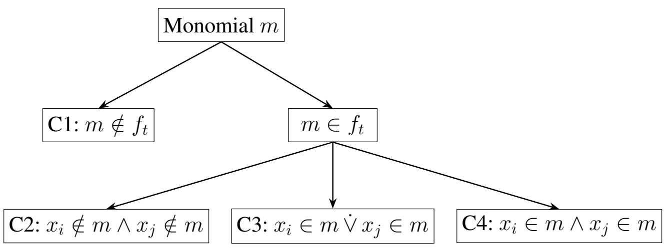

Figure 2: Overview of the proof's categories C1, C2, C3 and C4, where the latter three inherit the property $m \in f_t$.

Recall that the reduction replaces all occurrences of $x_i x_j$ by y_t. It further adds a penalty term $\mathrm{p}(x_i, x_j, y_t) = 3\underbrace{y_t}_{\text{a)}} + \underbrace{x_i x_j}_{\text{b)}} - 2\underbrace{x_i y_t}_{\text{c)}} - 2\underbrace{x_j y_t}_{\text{d)}}$ that only depends on x_i, x_j and y_t.

Category 1: $m \notin f_t$
A monomial that is not part of f_t (*i.e.*, $m \notin f_t$) can only change by the reduction step if it is introduced by the penalty term. The penalty term subdivides into a), b), c) and d). a) will not introduce an edge to G_{t+1}, since its two-combination set is empty ($P_{\{y_t\}} = \emptyset$). The node pair $\{v_i, v_j\}^\beta$, corresponding to b), must be in G_t, since the algorithm chooses $x_i x_j$. For its corresponding edge $\{v_i, v_j\}^\beta \in G_t$, $\beta \geq 2$ applies according to prerequisites. If b) is not part of f_t (*i.e.* $x_i x_j \notin f_t$), then b) introduces an edge $\{v_i, v_j\}$ and therefore increases the degree of v_i and v_j by one respectively [1]. If b) is part of f_t (*i.e.* $x_i x_j \in f_t$), category 4 applies. Since y_t is the new variable, c) and d) must not be part of f_t, that is $x_i y_t \notin f_t \wedge x_j y_t \notin f_t$. Therefore, the degree of v_i and v_j increases by one respectively. At the same time, the replacement part removes $\{v_i, v_j\}^\beta$. The proof for this can be found in category 4. For $\beta \geq 2$, which is guaranteed by the algorithms choice, the degree of v_i and v_j decreases by at least two respectively. We conclude that $\deg_{G_t}(v_x) \geq \deg_{G_{t+1}}(v_x) \, \forall v_x \in \{v_i, v_j\}$. Fig. 3 visualises the above stated.

Category 2: $m \in f_t \wedge x_i \notin m \wedge x_j \notin m$
Without loss of generality, let $m = x_a x_b x_c \ldots \in f_t$, where $x_i \notin m \wedge x_j \notin m$. m is not affected by the replacement part of the reduction, since $x_i \notin m \wedge x_j \notin m$. Beyond that, m is is not affected by the penalty part, since y_t is the new variable. Hence, a), c) and d) do not affect m in terms of algebraic simplification. Moreover, b) does not affect m, since $x_i \notin m \wedge x_j \notin m$. We conclude that the two-combination set $P_{\{m\}}$ of m is invariant under reduction. As a consequence of that, any edge introduced by $P_{\{m\}}$ does not change under reduction.

[1]For example: $f(\vec{x}) = x_1 x_2 x_3 + x_1 x_2 x_4 \overset{\text{reduc}}{\to} y_t x_3 + y_t x_4 \overbrace{\underbrace{x_1 x_2}_{\text{b)}} + \ldots}^{\mathrm{p}(x_1, x_2, y_t)}$.

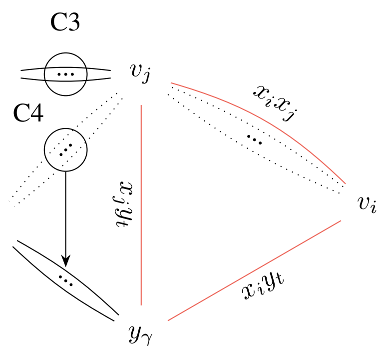

Figure 3: Effect of a reduction on v_i and v_j. Dotted lines: Removed edges. Orange lines: Edges introduced by the penalty term. $\{v_i, v_j\}^\beta$ corresponds to $x_i x_j$, which is replaced by y_t. The remapping of C4 edges ($x_i x_j \in m$) and the invariance of C3 edges ($x_i \in m \dot{\vee} x_j \in m$) is analogous for v_i (not drawn).

Category 3: $x_i \in m \dot{\vee} x_j \in m$
Without loss of generality, let $m = x_a x_b x_c x_d \ldots x_i$ be a monomial, where $x_j \notin m$. Its two-combination set $P_{\{m\}}$ has the following elements:

$$\begin{aligned} P_{\{m\}} = \{ & \{v_a, v_b\}, \{v_a, v_c\}, \{v_a, v_d\}, \ldots, \{v_a, v_i\}, \\ & \{v_b, v_c\}, \{v_b, v_d\}, \ldots, \{v_b, v_i\}, \\ & \{v_c, v_d\}, \ldots, \{v_c, v_i\}, \\ & \ldots \}. \end{aligned} \tag{3}$$

Analogously to category 2, it is unaffected by the replacement part, as well as the penalty part. The same argument applies for $m = x_a x_b x_c x_d \ldots x_j$.

Category 4: $x_i \in m \wedge x_j \in m$
Without loss of generality, let $m_t = x_a x_b x_c x_d \ldots x_i x_j$. Analogously to category 3, the two-combination set $P_{\{m_t\}}$ of m_t has the following elements:

$$\begin{aligned} P_{\{m_t\}} = \{ & \{v_a, v_b\}, \{v_a, v_c\}, \{v_a, v_d\}, \ldots, \{v_a, v_i\}, \{v_a, v_j\}, \\ & \{v_b, v_c\}, \{v_b, v_d\}, \ldots, \{v_b, v_i\}, \{v_b, v_j\}, \\ & \{v_c, v_d\}, \ldots, \{v_c, v_i\}, \{v_c, v_j\}, \\ & \ldots, \\ & \{v_i, v_j\} \}. \end{aligned} \tag{4}$$

Since the replacement part of the reduction sets $y_t = x_i x_j$, the resulting monomial $m_{t+1} = x_a x_b x_c x_d \ldots y_t$, which leads to

$$
\begin{aligned}
P_{\{m_{t+1}\}} = \{&\{v_a, v_b\}, \{v_a, v_c\}, \{v_a, v_d\}, \ldots, \{v_a, y_t\},\\
&\{v_b, v_c\}, \{v_b, v_d\}, \ldots, \{v_b, y_t\},\\
&\{v_c, v_d\}, \ldots, \{v_c, y_t\},\\
&\ldots\}.
\end{aligned} \tag{5}
$$

Therefore, $P_{S_{\text{lost}}} = P_{\{m_t\}} \setminus P_{\{m_{t+1}\}} = \{\{v_a, v_i\}, \{v_a, v_j\}, \{v_b, v_i\}, \{v_b, v_j\}, \{v_c, v_i\}, \{v_c, v_j\}, \ldots, \{v_i, v_j\}\}$ and $P_{S_{\text{new}}} = P_{\{m_{t+1}\}} \setminus P_{\{m_t\}} = \{\{v_a, y_t\}, \{v_b, y_t\}, \{v_c, y_t\}, \ldots\}$. Let $x_\gamma \in m_t$ be an arbitrary variable in m_t not equal to x_i or x_j and let v_γ be its corresponding node. We can see that from v_γ's point of view, $\{v_\gamma, v_i\}$ and $\{v_\gamma, v_j\}$ are replaced by $\{v_\gamma, y_t\}$. In other words, v_γ is no longer connected to both v_i and v_j, but rather to y_t after the reduction step (see Fig. 3). We conclude that $\deg_{G_t}(v_\gamma) > \deg_{G_{t+1}}(v_\gamma)$ and therefore $\deg_{G_t}(v_\gamma) \geq \deg_{G_{t+1}}(v_\gamma)$, which is based on the fact that v_γ is unaffected by the penalty term. Moreover, $\{v_i, v_j\} \in P_{S_{\text{lost}}}$. The replacement part therefore removes the single edge $\{v_i, v_j\}$ from m_t's local point of view. Since this is true for all category 4 monomials, the replacement part will in total remove all edges between v_i and v_j. As a prerequisite, the algorithm chooses an edge $\{v_i, v_j\}^\beta$ with $\beta \geq 2$. Hence, at least two edges are removed. Category 1 raises the special case $m_t = x_i x_j$. The replacement part leads to $m_{t+1} = y_t$, that is a temporary removed edge. The penalty term then introduces $x_i x_j$, that is the same edge. In total the degree of v_i and v_j does not change in this special case, but can be lower if there are further monomials containing $x_i x_j$ as described above. In summary, we can conclude: $\deg_{G_t}(v_x) \geq \deg_{G_{t+1}}(v_x)\, \forall v_x \in \{v_i, v_j\}$.

All in all, $\deg_{G_t}(v_x) \geq \deg_{G_{t+1}}(v_x)\, \forall v_x \in V_t$, since the categories depict all possible monomials, which concludes the first part of the proof.

Category 4 gives the reason why the multi-edge $\{v_i, v_j\}^\beta$ with $\beta \geq 2$, is removed (replacement part) and then reintroduced as a single edge (penalty term part b)). It remains to show that the reduction does not increase the size of multi-edges on other nodes. Category 2 and 3 monomials are unaffected by the reduction. Hence, their induced edges in G_{t+1} do not change. From the argument in category 1, we can see that c) and d) from the penalty term are not part of f_t (*i.e.*, $x_i y_t \notin f_t \wedge x_j y_t \notin f_t$) and therefore create two new edges in G_{t+1}, namely $\{v_i, y_t\}$ and $\{v_j, y_t\}$. Both edges are unique and therefore cannot form a multi-edge. Furthermore, part a) does not lead to an edge in G_{t+1}. Edges $e \in E_t$ stemming from category 4 monomials are remapped to y_t (see Fig. 3, $P_{S_{\text{lost}}}$ and $P_{S_{\text{new}}}$). Consequently, the total size of multi-edges strictly decreases with every reduction step: $\sum_{e=\{v_i,v_j\}^\beta \in E_t,\, \beta>1} \beta < \sum_{e=\{v_i,v_j\}^\beta \in E_{t+1},\, \beta>1} \beta$. $\square$

Corollary 1.1. *Let f_0 be the starting polynomial and f_t be the last polynomial. The graph corresponding to f_t (*i.e., *G_t) has no multi-edges, since their size strictly decreases. We therefore conclude that the multi-edge selecting algorithm that uses Boros [1] reduction method terminates.*

In summary, we are left with a graph that has no multi-edges left, but may still represent a degree-k polynomial, with $k > 2$ (*i.e.*, a not-yet quadratised polynomial) [2]. Regarding the whole multi-reduction method, it is interesting to know whether further reduction steps introduce multi-edges again. We now assume an algorithm that operates on a polynomial f_t whose corresponding graph G_t has no multi-edges. Since there are no multi-edges in G_t it selects a degree-k monomial, where $k > 2$ and reduces it via Boros [1] method. We call the resulting polynomial f_{t+1} and its graph G_{t+1}. In this second phase, the variable selection types do not differ, since there are no common variable pairs among monomials.

Theorem 2. *If G_t has no multi-edges, then G_{t+1} has no multi-edges under the effect of reduction via Boros method [1].*

Proof. We know that multi-edges are introduced in G_t whenever there are two or more two-combination sets that are not disjoint: $\exists P_{S_i}, P_{S_j} : P_{S_i} \cap P_{S_j} \neq \emptyset \implies \exists \{v_a, v_b\}^\beta \in E_t : \beta > 1$.
This is logically equivalent to $\forall \{v_a, v_b\}^\beta \in E_t : \beta \leq 1 \implies \forall P_{S_i}, P_{S_j} : P_{S_i} \cap P_{S_j} = \emptyset$. Since $\beta \in \mathbb{N}$, we can rewrite: $\forall \{v_a, v_b\}^\beta \in E_t : \beta = 1 \implies \forall P_{S_i}, P_{S_j} : P_{S_i} \cap P_{S_j} = \emptyset$. In other words we can conclude that, starting from the graph G_t, which has no multi-edges, the polynomial f_t has no monomials that share elements from their two-combination sets. Let $x_i x_j$ be the variable-pair that will be replaced by y_t. As the previous argument states, there is exactly one monomial m containing $x_i x_j$. The replacement part of the reduction introduces y_t which is unique. Hence, the two-combination set $P_{\{m\}}$ of m is still disjoint to the other two-combination sets of f_t. The penalty part introduces $3 \underbrace{y_t}_{\text{a)}} + \underbrace{x_i x_j}_{\text{b)}} - 2 \underbrace{x_i y_t}_{\text{c)}} - 2 \underbrace{x_j y_t}_{\text{d)}}$.
$P_{\{y_t\}} = \emptyset$, c) and d) are unique and therefore $P_{\{y_t\}}$, $P_{\{x_i, y_t\}}$ and $P_{\{x_j, y_t\}}$ are pairwise disjoint from each other and disjoint from the other two-combination sets. After the reduction, $x_i x_j$ was replaced by y_t. Hence, $P_{\{x_i, x_j\}}$ is also disjoint from the other two-combination sets after the reduction. $\square$

Corollary 2.1. *Starting from a non-quadratic polynomial f_0 ($\deg(f_0) > 2$), the combined multi-reduction algorithm terminates in a quadratic polynomial.*

References

[1] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 155–225, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166218X01003419

[2] For example $f(x_1, \ldots, x_n) = x_1 x_2 x_3 + x_4 x_5 x_6 + \ldots + x_{n-2} x_{n-1} x_n$.