Quantum Inf Process (2015) 14:3161–3177
DOI 10.1007/s11128-015-1069-y

A bounded-error quantum polynomial-time algorithm for two graph bisection problems

Ahmed Younes[1,2]

Received: 29 May 2015 / Accepted: 5 July 2015 / Published online: 14 July 2015

Abstract The aim of the paper was to propose a bounded-error quantum polynomial-time algorithm for the max-bisection and the min-bisection problems. The max-bisection and the min-bisection problems are fundamental NP-hard problems. Given a graph with even number of vertices, the aim of the max-bisection problem is to divide the vertices into two subsets of the same size to maximize the number of edges between the two subsets, while the aim of the min-bisection problem is to minimize the number of edges between the two subsets. The proposed algorithm runs in $O(m^2)$ for a graph with m edges and in the worst case runs in $O(n^4)$ for a dense graph with n vertices. The proposed algorithm targets a general graph by representing both problems as Boolean constraint satisfaction problems where the set of satisfied constraints are simultaneously maximized/minimized using a novel iterative partial negation and partial measurement technique. The algorithm is shown to achieve an arbitrary high probability of success of $1 - \epsilon$ for small $\epsilon > 0$ using a polynomial space resources.

Keywords Quantum algorithm · Graph bisection · Max-bisection · Min-bisection · Amplitude amplification · BQP · NP-hard

1 Introduction

Given an undirected graph $G = (V, E)$ with a set V of even number of vertices and a set E of unweighted edges, two graph bisection problems will be considered in

✉ Ahmed Younes
ayounes2@yahoo.com; ayounes@alexu.edu.eg

[1] Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Alexandria, Egypt

[2] School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK

the paper: the max-bisection problem and the min-bisection problem. The goal of the max-bisection problem is to divide V into two subsets A and B of the same size so as to maximize the number of edges between A and B, while the goal of the min-bisection problem is to minimize the number of edges between A and B. In theory, both bisection problems are NP-hard for general graphs [9,14].

These classical combinatorial optimization problems are special cases of graph partitioning [13]. The graph partitioning has many applications, for example divide-and-conquer algorithms [24], compiler optimization [21], VLSI circuit layout [5], load balancing [17], image processing [30], computer vision [22], distributed computing [25] and route planning [7]. In practice, there are many general-purpose heuristics for graph partitioning, e.g., [18,31,32] that handle particular graph classes, such as [6,26,32]. There are also many practical exact algorithms for graph bisection that use the branch-and-bound approach [8,23]. These approaches make expensive usage of time and space to obtain lower bounds [1,2,8,16].

On conventional computers, approximation algorithms have gained much attention to tackle the max-bisection and the min-bisection problems. The max-bisection problem, for example, has an approximation ratio of 0.7028 due to [10] which is known to be the best approximation ratio for a long time by introducing the RPR^2 rounding technique into semidefinite programming (SDP) relaxation. In [12], a poly-time algorithm is proposed that given a graph admitting a bisection cutting, a fraction $1 - \varepsilon$ of edges finds a bisection cutting an $(1 - g(\varepsilon))$ fraction of edges where $g(\varepsilon) \rightarrow 0$ as $\varepsilon \rightarrow 0$. A 0.85-approximation algorithm for the max-bisection is obtained in [28]. In [33], the SDP relaxation and the RPR^2 technique of [10] have been used to obtain a performance curve as a function of the ratio of the optimal SDP value over the total weight through finer analysis under the assumption of convexity of the RPR^2 function. For the min-bisection problem, the best-known approximation ratio is $O(\log n)$ [27] with some limited graph classes have known polynomial-time solutions such as grids without holes [11] and graphs with bounded tree width [20].

The aim of the paper was to propose an algorithm that represents the two bisection problems as Boolean constraint satisfaction problems where the set of edges are represented as set of constraints. The algorithm prepares a superposition of all possible graph bisections using an amplitude amplification technique, then evaluates the set of constraints for all possible bisections simultaneously and then amplifies the amplitudes of the best bisections that achieve the maximum/minimum satisfaction to the set of constraints using a novel amplitude amplification technique that applies an iterative partial negation and partial measurement. The proposed algorithm targets a general graph where it runs in $O(m^2)$ for a graph with m edges and in the worst case runs in $O(n^4)$ for a dense graph with number edges close to $m = \frac{n(n-1)}{2}$ with n vertices to achieve an arbitrary high probability of success of $1 - \epsilon$ for small $\epsilon > 0$ using a polynomial space resources.

The paper is organized as follows: Sect. 2 shows the data structure used to represent a graph bisection problem as a Boolean constraint satisfaction problem. Section 3 presents the proposed algorithm with analysis on time and space requirements. Section 4 concludes the paper.

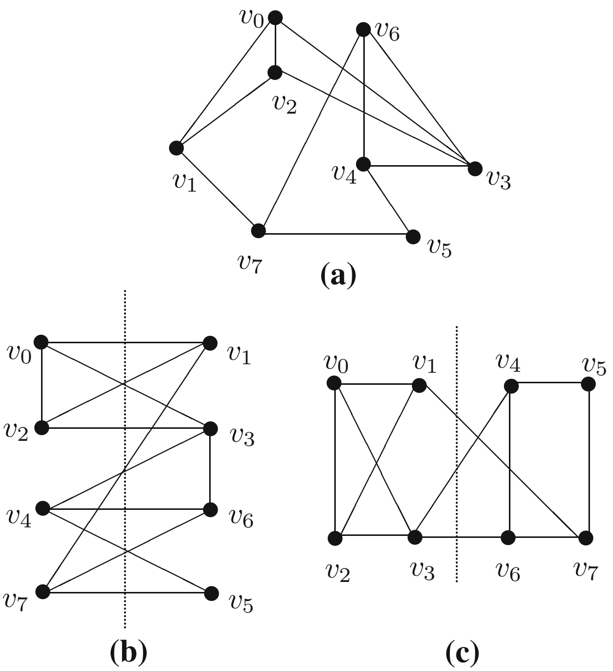

Fig. 1 **a** A random graph with 8 vertices and 12 edges, **b** a max-bisection instance for the graph in **a** with 10 edges connecting the two subsets, and **c**. **b** A min-bisection instance for the graph in **a** with 3 edges connecting the two subsets

2 Data structures and graph representation

Several optimization problems, such as the max-bisection and the min-bisection problems, can be formulated as Boolean constraint satisfaction problems [3,4] where a feasible solution is a solution with as many variables set to 0 as variables set to 1, i.e., balanced assignment, as follows: For a graph G with n vertices and m edges, consider n Boolean variables $v_0, \ldots, v_{n-1}$ and m constraints by associating with each edge $(a, b) \in E$ the constraint $c_l = v_a \oplus v_b$, with $l = 0, 1, \ldots, m-1$, then the max-bisection is the problem that consists of finding a balanced assignment to maximize the number of constraints equal to logic-1 from the m constraints, while the min-bisection is the problem that consists of finding a balanced assignment to maximize the number of constraints equal to logic-0 from the m constraints, such that if a Boolean variable is set to 0, then the associated vertex belongs to the first partition, and if a Boolean variable is set to 1, then the associated vertex belongs to the second partition.

For example, consider the graph G shown in Fig. 1a. Let $G = (V, E)$, where,

$$\begin{aligned} V &= \{0, 1, 2, 3, 4, 5, 6, 7\}, \\ E &= \{(0, 1), (0, 2), (0, 3), \\ &\quad (1, 2), (1, 7), (2, 3), \\ &\quad (3, 4), (3, 6), (4, 5), \\ &\quad (4, 6), (5, 7), (6, 7)\}. \end{aligned} \tag{1}$$

Assume that each vertex $a \in V$ is associated with a Boolean variable v_a, then the set of vertices V can be represented as a vector X of Boolean variables as follows,

$$X = (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7), \tag{2}$$

and if each edge $(a, b) \in E$ is associated with a constraint $c_l = v_a \oplus v_b$, then the set of edges E can be represented as a vector Z of constraints as follows,

$$Z = (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}), \tag{3}$$

such that,

$$\begin{aligned}
&c_0 = (v_0 \oplus v_1), c_1 = (v_0 \oplus v_2), c_2 = (v_0 \oplus v_3),\\
&c_3 = (v_1 \oplus v_2), c_4 = (v_1 \oplus v_7), c_5 = (v_2 \oplus v_3),\\
&c_6 = (v_3 \oplus v_4), c_7 = (v_3 \oplus v_6), c_8 = (v_4 \oplus v_5),\\
&c_9 = (v_4 \oplus v_6), c_{10} = (v_5 \oplus v_7), c_{11} = (v_6 \oplus v_7).
\end{aligned} \tag{4}$$

In general, a bisection G_P for the graph G can be represented as $G_p = (x, z(x))$ such that each vector $x \in \{0, 1\}^n$ of variable assignments is associated with a vector $z(x) \in \{0, 1\}^m$ of constraints evaluated as functions of the variable assignment x. In the max-bisection and the min-bisection problems, the vector x of variable assignments is restricted to be balanced so there are $M = \binom{n}{\frac{n}{2}}$ possible variable assignments among the $N = 2^n$ possible variable assignments, and the solution of the max-bisection problem is to find the variable assignment that is associated with a vector of constraints that contains the maximum number of 1's, and the solution for the min-bisection problem is to find the variable assignment that is associated with a vector of constraints that contains the maximum number of 0's. For example, for the graph G shown in Fig. 1a, a max-bisection for G is ((0, 1, 0, 1, 0, 1, 1, 0), (1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1)) with 10 edges connecting the two partitions as shown in Fig. 1b, and a min-bisection for G is ((0, 0, 0, 0, 1, 1, 1, 1), (0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0)) with 3 edges connecting the two partitions as shown in Fig. 1c. It is important to notice that a variable assignment $x = (0, 1, 0, 1, 0, 1, 1, 0)$ is equivalent to $\overline{x} = (1, 0, 1, 0, 1, 0, 0, 1)$, where $\overline{x}$ is the bit-wise negation of x.

3 The algorithm

Given a graph G with n vertices and m edges, the proposed algorithm is divided into three stages: The first stage prepares a superposition of all balanced assignments for the n variables. The second stage evaluates the m constraints associated with the m edges for every balanced assignment and stores the values of constraints in constraint vectors entangled with the corresponding balanced assignments in the superposition. The third stage amplifies the constraint vector with maximum (minimum) number of satisfied constraints using a partial negation and iterative measurement technique.

3.1 Balanced assignments preparation

To prepare a superposition of all balanced assignments of n qubits, the proposed algorithm can use any amplitude amplification technique, e.g., [15,34,35]. An extra step should be added after the amplitude amplification to create an entanglement between the matched items and an auxiliary qubit $|ax_1\rangle$, so that the correctness of the items in the superposition can be verified by applying measurement on $|ax_1\rangle$ without having to examine the superposition itself. So, if $|ax_1\rangle = |1\rangle$ at the end of this stage, then the superposition contains the correct items, i.e., the balanced assignments, otherwise, repeat the preparation stage until $|ax_1\rangle = |1\rangle$. This is useful for not having to proceed to the next stages until the preparation stage succeeds.

The preparation stage to have a superposition of all balanced assignments of n qubits will use the amplitude amplification technique shown in [35] since it achieves the highest known probability of success using fixed operators and it can be summarized as follows, prepare a superposition of 2^n states by initializing n qubits to state $|0\rangle$ and apply $H^{\otimes n}$ on the n qubits

$$\begin{aligned} |\Psi_0\rangle &= \left(H^{\otimes n}\right)|0\rangle^{\otimes n} \\ &= \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle, \end{aligned} \tag{5}$$

where H is the Hadamard gate and $N = 2^n$. Assume that the system $|\Psi_0\rangle$ is rewritten as follows,

$$|\Psi_0\rangle = \frac{1}{\sqrt{N}}\sum_{\substack{j=0\\ j\in X_T}}^{N-1}|j\rangle + \frac{1}{\sqrt{N}}\sum_{\substack{j=0\\ j\in X_F}}^{N-1}|j\rangle, \tag{6}$$

where X_T is the set of all balanced assignments of n bits and X_F is the set of all unbalanced assignments. Let $M = \begin{pmatrix} n \\ \frac{n}{2}\end{pmatrix}$ be the number of balanced assignments among the 2^n possible assignments, $\sin(\theta) = \sqrt{M/N}$ and $0 < \theta \leq \pi/2$, then the system can be rewritten as follows,

$$|\Psi_0\rangle = \sin(\theta)\,|\psi_1\rangle + \cos(\theta)\,|\psi_0\rangle\,, \tag{7}$$

where $|\psi_1\rangle = |\tau\rangle$ represents the balanced assignments subspace and $|\psi_0\rangle$ represents the unbalanced assignments subspace.

Let $D = WR_0(\phi)\,W^{\dagger}R_\tau(\phi)\,,\ R_0(\phi) = I - (1-e^{i\phi})\,|0\rangle\langle 0|\,,\ R_\tau(\phi) = I - (1 - e^{i\phi})\,|\tau\rangle\langle\tau|$, where $W = H^{\otimes n}$ is the Walsh–Hadamard transform [19]. Iterate the operator D on $|\Psi_0\rangle$ for q times to get,

$$|\Psi_1\rangle = D^q\,|\Psi_0\rangle = a_q\,|\psi_1\rangle + b_q\,|\psi_0\rangle\,, \tag{8}$$

such that,

$$a_q = \sin(\theta)\left(e^{iq\phi}U_q\left(y\right) + e^{i(q-1)\phi}U_{q-1}\left(y\right)\right), \tag{9}$$

$$b_q = \cos(\theta)e^{i(q-1)\phi}\left(U_q\left(y\right) + U_{q-1}\left(y\right)\right), \tag{10}$$

where $y = cos(\delta)$, $\cos\left(\delta\right) = 2\sin^2(\theta)\sin^2(\frac{\phi}{2}) - 1$, $0 < \theta \leq \pi/2$, and U_q is the Chebyshev polynomial of the second kind [29] defined as follows,

$$U_q\left(y\right) = \frac{\sin\left(\left(q+1\right)\delta\right)}{\sin\left(\delta\right)}. \tag{11}$$

Setting $\phi = 6.02193 \approx 1.9168\pi$, $M = \begin{pmatrix} n \\ \frac{n}{2} \end{pmatrix}$, $N = 2^n$ and, $q = \left\lfloor \frac{\phi}{\sin(\theta)} \right\rfloor$, then $\left|a_q\right|^2 \geq 0.9975$ [35]. The upper bound for the required number of iterations q to reach the maximum probability of success is,

$$q = \left\lfloor \frac{\phi}{\sin(\theta)} \right\rfloor \leq 1.9168\pi\sqrt{\frac{N}{M}}, \tag{12}$$

and using Stirling's approximation,

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \tag{13}$$

then, the upper bound for required number of iterations q to prepare the superposition of all balanced assignments is,

$$q \approx 1.9168\sqrt[4]{\frac{\pi^5}{2}n} = O\left(\sqrt[4]{n}\right). \tag{14}$$

It is required to preserve the states in $|\psi_1\rangle$ for further processing in the next stage. This can be done by adding an auxiliary qubit $|ax_1\rangle$ initialized to state $|0\rangle$ and have the states of the balanced assignments entangled with $|ax_1\rangle = |1\rangle$, so that, the correctness of the items in the superposition can be verified by applying measurement on $|ax_1\rangle$ without having to examine the superposition itself. So, if $|ax_1\rangle = |1\rangle$, then the superposition contains the balanced assignments, otherwise, repeat the preparation stage until $|ax_1\rangle = |1\rangle$. This is useful to be able to proceed to the next stage when the preparation stage succeeds. To prepare the entanglement, let

$$\begin{aligned} |\Psi_2\rangle &= |\Psi_1\rangle \otimes |0\rangle \\ &= a_q\left|\psi_1\right\rangle \otimes |0\rangle + b_q\left|\psi_0\right\rangle \otimes |0\rangle, \end{aligned} \tag{15}$$

and apply a quantum Boolean operator U_f on $|\Psi_2\rangle$, where U_f is defined as follows,

$$U_f\,|x,0\rangle = \begin{cases} |x,0\rangle\,, & \text{if} \quad |x\rangle \in |\psi_0\rangle\,, \\ |x,1\rangle\,, & \text{if} \quad |x\rangle \in |\psi_1\rangle\,, \end{cases} \tag{16}$$

and $f : \{0,1\}^n \to \{0,1\}$ is an n inputs single output Boolean function that evaluates to true for any $x \in X_T$ and evaluates to false for any $x \in X_F$; then,

$$\begin{aligned} |\Psi_3\rangle &= U_f\,|\Psi_2\rangle \\ &= a_q\,|\psi_1\rangle \otimes |1\rangle + b_q\,|\psi_0\rangle \otimes |0\rangle\,. \end{aligned} \tag{17}$$

Apply measurement M_1 on the auxiliary qubit $|ax_1\rangle$ as shown in Fig. 2. The probability of finding $|ax_1\rangle = |1\rangle$ is,

$$\Pr(M_1 = 1) = \left|a_q\right|^2 \geq 0.9975, \tag{18}$$

and the system will collapse to,

$$\left|\Psi_3^{(M_1=1)}\right\rangle = |\psi_1\rangle \otimes |1\rangle\,. \tag{19}$$

3.2 Evaluation of constraints

There are M states in the superposition $\left|\Psi_3^{(M_1=1)}\right\rangle$, each state has an amplitude $\frac{1}{\sqrt{M}}$, and then let $|\Psi_4\rangle$ be the system after the balanced assignment preparation stage as follows,

$$|\Psi_4\rangle = \alpha \sum_{k=0}^{M-1} |x_k\rangle, \tag{20}$$

where $|ax_1\rangle$ is dropped from the system for simplicity and $\alpha = \frac{1}{\sqrt{M}}$. For a graph G with n vertices and m edges, every edge (a,b) connecting vertcies $a, b \in V$ is associated with a constraint $c_l = v_a \oplus v_b$, where v_a and v_b are the corresponding qubits for vertices a and b in $|\Psi_4\rangle$, respectively, such that $0 \leq l < m$, $0 \leq m \leq \frac{n(n-1)}{2}$, $0 \leq a, b \leq n-1$ and $a \neq b$, where $\frac{n(n-1)}{2}$ is the maximum number of edges in a graph with n vertices.

To evaluate the m constraints associated with the edges, add m qubits initialized to state $|0\rangle$,

$$\begin{aligned} |\Psi_5\rangle &= |\Psi_4\rangle \otimes |0\rangle^{\otimes m} \\ &= \alpha \sum_{k=0}^{M-1} |x_k\rangle \otimes |0\rangle^{\otimes m}. \end{aligned} \tag{21}$$

For every constraint $c_l = v_a \oplus v_b$, apply two $Cont\_\sigma_X$ gates, $Cont\_\sigma_X(v_a, c_l)$ and $Cont\_\sigma_X(v_b, c_l)$, so that $|c_l\rangle = |v_a \oplus v_b\rangle$. The collection of all $Cont\_\sigma_X$ gates

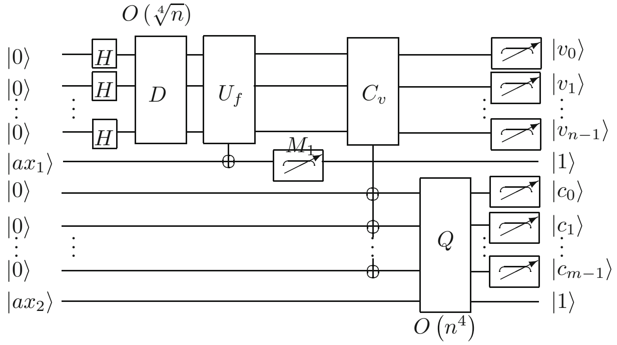

Fig. 2 A quantum circuit for the proposed algorithm

applied to evaluate the m constraints is denoted C_v in Fig. 2, and then the system is transformed to,

$$|\Psi_6\rangle = \alpha \sum_{k=0}^{M-1} \left(|x_k\rangle \otimes \left|c_0^k c_1^k \ldots c_{m-1}^k\right\rangle\right), \tag{22}$$

where σ_X is the Pauli-X gate which is the quantum equivalent to the NOT gate. It can be seen as a rotation of the Bloch Sphere around the X-axis by π radians as follows,

$$\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{23}$$

and $Cont\_U(v, c)$ gate is a controlled gate with control qubit $|v\rangle$ and target qubit $|c\rangle$ that applies a single qubit unitary operator U on $|c\rangle$ only if $|v\rangle = |1\rangle$, so every qubit $\left|c_l^k\right\rangle$ carries a value of the constraint c_l based on the values of v_a and v_b in the balanced assignment $|x_k\rangle$, i.e., the values of v_a^k and v_b^k, respectively. Let $|z_k\rangle = \left|c_0^k c_1^k \ldots c_{m-1}^k\right\rangle$, then the system can be rewritten as follows,

$$|\Psi_6\rangle = \alpha \sum_{k=0}^{M-1} (|x_k\rangle \otimes |z_k\rangle), \tag{24}$$

where every $|x_k\rangle$ is entangled with the corresponding $|z_k\rangle$. The aim of the next stage is to find $|z_k\rangle$ with the maximum number of $|1\rangle$'s for the max-bisection problem or to find $|z_k\rangle$ with the minimum number of $|1\rangle$'s for the min-bisection problem.

3.3 Maximization of the satisfied constraints

Let $|\psi_c\rangle$ be a superposition on M states as follows,

$$|\psi_c\rangle = \alpha \sum_{k=0}^{M-1} |z_k\rangle, \tag{25}$$

where each $|z_k\rangle$ is an m-qubit state and let $d_k = \langle z_k \rangle$ be the number of 1's in state $|z_k\rangle$ such that $|z_k\rangle \neq |0\rangle^{\otimes m}$, i.e., $d_k \neq 0$. This will be referred to as the 1-distance of $|z_k\rangle$.

The max-bisection graph $|x_{\max}\rangle$ is equivalent to find the state $|z_{\max}\rangle$ with $d_{\max} = \max\{d_k,\ 0 \leq k \leq M-1\}$ and the state $|z_{\min}\rangle$ with $d_{\min} = \min\{d_k,\ 0 \leq k \leq M-1\}$ is equivalent to the min-bisection graph $|x_{\min}\rangle$. Finding the state $|z_{\min}\rangle$ with the minimum number of 1's is equivalent to finding the state with the maximum number of 0's, so, to clear ambiguity, let $d_{\max 1} = d_{\max}$ be the maximum number of 1's and $d_{\max 0} = d_{\min}$ be the maximum number of 0's, where the number of 0's in $|z_k\rangle$ will be referred to as the 0-distance of $|z_k\rangle$.

To find either $|z_{\max}\rangle$ or $|z_{\min}\rangle$, when $|\psi_c\rangle$ is measured, add an auxiliary qubit $|ax_2\rangle$ initialized to state $|0\rangle$ to the system $|\psi_c\rangle$ as follows,

$$\begin{aligned} |\psi_m\rangle &= |\psi_c\rangle \otimes |0\rangle \\ &= \alpha \sum_{k=0}^{M-1} |z_k\rangle \otimes |0\rangle . \end{aligned} \tag{26}$$

The main idea to find $|z_{\max}\rangle$ is to apply partial negation on the state of $|ax_2\rangle$ entangled with $|z_k\rangle$ based on the number of 1's in $|z_k\rangle$, i.e., more 1's in $|z_k\rangle$, gives more negation to the state of $|ax_2\rangle$ entangled with $|z_k\rangle$. If the number of 1's in $|z_k\rangle$ is m, then the entangled state of $|ax_2\rangle$ will be fully negated. The mth partial negation operator is the mth root of σ_X and can be calculated using diagonalization as follows,

$$V = \sqrt[m]{\sigma_X} = \frac{1}{2}\begin{bmatrix} 1+t & 1-t \\ 1-t & 1+t \end{bmatrix}, \tag{27}$$

where $t = \sqrt[m]{-1}$, and applying V for d times on a qubit is equivalent to the operator,

$$V^d = \frac{1}{2}\begin{bmatrix} 1+t^d & 1-t^d \\ 1-t^d & 1+t^d \end{bmatrix}, \tag{28}$$

such that if $d = m$, then $V^m = \sigma_X$. To amplify the amplitude of the state $|z_{\max}\rangle$, apply the operator MAX on $|\psi_m\rangle$ as will be shown later, where MAX is an operator on $m+1$ qubits register that applies V conditionally for m times on $|ax_2\rangle$ based on the number of 1's in $|c_0 c_1 \ldots c_{m-1}\rangle$ as follows (as shown in Fig. 3a),

$$\text{MAX} = Cont\_V(c_0, ax_2)Cont\_V(c_1, ax_2)\ldots Cont\_V(c_{m-1}, ax_2), \tag{29}$$

so, if d_1 is the number of $c_l = 1$ in $|c_0 c_1 \ldots c_{m-1}\rangle$, then,

$$\text{MAX}\left(|c_0 c_1 ... c_{m-1}\rangle \otimes |0\rangle\right) = |c_0 c_1 ... c_{m-1}\rangle \otimes \left(\frac{1+t^{d_1}}{2}|0\rangle + \frac{1-t^{d_1}}{2}|1\rangle\right). \tag{30}$$

Amplifying the amplitude of the state $|z_{\min}\rangle$ with the minimum number of 1's is equivalent to amplifying the amplitude of the state with the maximum number of 0's. To find $|z_{\min}\rangle$, apply the operator MIN on $|\psi_m\rangle$ as will be shown later, where MIN is

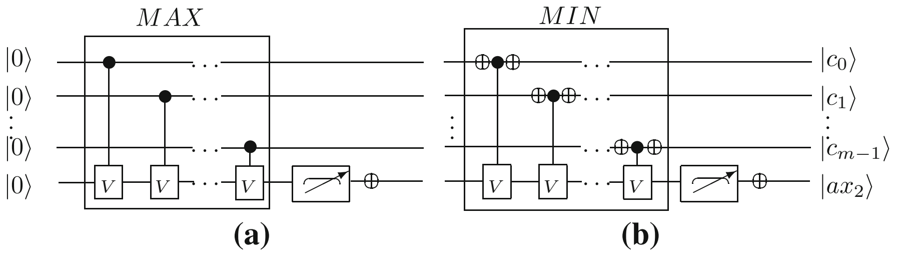

Fig. 3 Quantum circuits for **a** the MAX operator and **b** the MIN operator, followed by a partial measurement then a negation to reset the auxiliary qubit $|ax_2\rangle$

an operator on $m+1$ qubits register that applies V conditionally for m times on $|ax_2\rangle$ based on the number of 0's in $|c_0c_1\ldots c_{m-1}\rangle$ as follows (as shown in Fig. 3b),

$$\mathrm{MIN} = Cont\_V(\overline{c_0}, ax_2)Cont\_V(\overline{c_1}, ax_2)\ldots Cont\_V(\overline{c_{m-1}}, ax_2), \tag{31}$$

where $\overline{c_l}$ is a temporary negation of c_l before and after the application of $Cont\_V(c_l, ax_2)$ as shown in Fig. 3, so, if d_0 is the number of $c_l = 0$ in $|c_0c_1\ldots c_{m-1}\rangle$ then,

$$\mathrm{MIN}\left(|c_0c_1...c_{m-1}\rangle \otimes |0\rangle\right) = |c_0c_1...c_{m-1}\rangle \otimes \left(\frac{1+t^{d_0}}{2}|0\rangle + \frac{1-t^{d_0}}{2}|1\rangle\right). \tag{32}$$

For the sake of simplicity and to avoid duplication, the operator Q will denote either the operator MAX or the operator MIN, d will denote either d_1 or d_0, $|z_s\rangle$ will denote either $|z_{\max}\rangle$ or $|z_{\min}\rangle$, and d_s will denote either $d_{\max 1}$ or $d_{\max 0}$, so,

$$Q\left(|c_0c_1...c_{m-1}\rangle \otimes |0\rangle\right) = |c_0c_1...c_{m-1}\rangle \otimes \left(\frac{1+t^{d}}{2}|0\rangle + \frac{1-t^{d}}{2}|1\rangle\right), \tag{33}$$

and the probabilities of finding the auxiliary qubit $|ax_2\rangle$ in state $|0\rangle$ or $|1\rangle$ when measured is, respectively, as follows,

$$\begin{aligned}\Pr(ax_2 = 0) &= \left|\frac{1+t^d}{2}\right|^2 = \cos^2\left(\frac{d\pi}{2m}\right),\\ \Pr(ax_2 = 1) &= \left|\frac{1-t^d}{2}\right|^2 = \sin^2\left(\frac{d\pi}{2m}\right).\end{aligned} \tag{34}$$

To find the state $|z_s\rangle$ in $|\psi_m\rangle$, the proposed algorithm is as follows, as shown in Fig. 3:

1- Let $|\psi_r\rangle = |\psi_m\rangle$.
2- Repeat the following steps for r times,
  i- Apply the operator Q on $|\psi_r\rangle$.

   ii- Measure $|ax_2\rangle$, if $|ax_2\rangle = |1\rangle$, then let the system post-measurement is $|\psi_r\rangle$, apply σ_X on $|ax_2\rangle$ to reset to $|0\rangle$ for the next iteration and then go to Step (i), otherwise restart the stage and go to Step (1).

3- Measure the first m qubits in $|\psi_r\rangle$ to read $|z_s\rangle$.

For simplicity and without loss of generality, assume that a single $|z_s\rangle$ exists in $|\psi_v\rangle$, although such states will exist in couples since each $|z_s\rangle$ is entangled with a variable assignment $|x_s\rangle$ and each $|x_s\rangle$ is equivalent to $|\overline{x_s}\rangle$; moreover, different variable assignments might give rise to constraint vectors with maximum distance, but such information is not known in advance.

Assuming that the algorithm finds $|ax_2\rangle = |1\rangle$ for r times in a row, then the probability of finding $|ax_2\rangle = |1\rangle$ after Step (2-i) in the first iteration, i.e., $r = 1$ is given by,

$$\Pr{}^{(1)}(ax_2 = 1) = \alpha^2 \sum_{k=0}^{M-1} \sin^2\left(\frac{d_k \pi}{2m}\right). \tag{35}$$

The probability of finding $|\psi_r\rangle = |z_s\rangle$ after Step (2-i) in the first iteration, i.e., $r = 1$ is given by,

$$\Pr{}^{(1)}(\psi_r = z_s) = \alpha^2 \sin^2\left(\frac{d_s \pi}{2m}\right). \tag{36}$$

The probability of finding $|ax_2\rangle = |1\rangle$ after Step (2-i) in the rth iteration, i.e., $r > 1$ is given by,

$$\Pr{}^{(r)}(ax_2 = 1) = \frac{\sum_{k=0}^{M-1} \sin^{2r}\left(\frac{d_k \pi}{2m}\right)}{\sum_{k=0}^{M-1} \sin^{2(r-1)}\left(\frac{d_k \pi}{2m}\right)}. \tag{37}$$

The probability of finding $|\psi_r\rangle = |z_s\rangle$ after Step (2-i) in the rth iteration, i.e., $r > 1$ is given by,

$$\Pr{}^{(r)}(\psi_r = z_s) = \frac{\sin^{2r}\left(\frac{d_s \pi}{2m}\right)}{\sum_{k=0}^{M-1} \sin^{2(r-1)}\left(\frac{d_k \pi}{2m}\right)}. \tag{38}$$

To get the highest probability of success for $\Pr(\psi_r = z_s)$, Step (2) should be repeated until, $\left|\Pr^{(r)}(ax_2 = 1) - \Pr^{(r)}(\psi_r = z_s)\right| \leq \epsilon$ for small $\epsilon \geq 0$ as shown in Fig. 4. This happens when $\sum_{k=0,k\neq s}^{M-1} \sin^{2r}\left(\frac{d_k \pi}{2m}\right) \leq \epsilon$. Since the Sine function is a decreasing function then for sufficient large r,

$$\sum_{k=0,k\neq s}^{M-1} \sin^{2r}\left(\frac{d_k \pi}{2m}\right) \approx \sin^{2r}\left(\frac{d_{ns} \pi}{2m}\right), \tag{39}$$

where d_{ns} is the next maximum distance less than d_s. The values of d_s and d_{ns} are unknown in advance, so let $d_s = m$ be the number of edges, then in the worst case when $d_s = m$, $d_{ns} = m - 1$ and $m = n(n-1)/2$, the required number of iterations r for $\epsilon = 10^{-\lambda}$ and $\lambda > 0$ can be calculated using the formula,

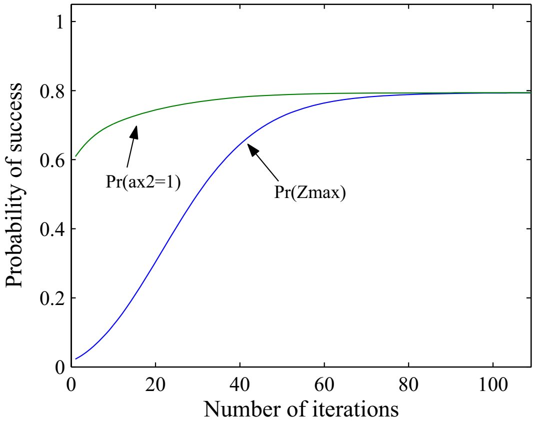

Fig. 4 The probability of success for a max-bisection instance of the graph shown in Fig. 1 with $n = 8$ and $m = 12$ where the probability of success of $|ax_2\rangle$ is 0.6091 after the first iteration and with probability of success of 0.7939 after iterating the algorithm where the probability of success of $|z_{\max}\rangle$, is amplified to reach the probability of success of $|ax_2\rangle$

$$0 < \sin^{2r}\left(\frac{(m-1)\pi}{2m}\right) \leq \epsilon, \tag{40}$$

$$\begin{aligned} r &\geq \frac{\log(\epsilon)}{2\log\left(\sin\left(\frac{(m-1)\pi}{2m}\right)\right)} \\ &= \frac{\log\left(10^{-\lambda}\right)}{2\log\left(\cos\left(\frac{\pi}{2m}\right)\right)} \\ &\geq \lambda\left(\frac{2m}{\pi}\right)^2 \\ &= O\left(m^2\right), \end{aligned} \tag{41}$$

where $0 \leq m \leq \frac{n(n-1)}{2}$. For a complete graph where $m = \frac{n(n-1)}{2}$, then the upper bound for the required number of iterations r is $O\left(n^4\right)$. Assuming that a single $|z_s\rangle$ exists in the superposition will increase the required number of iterations, so it is important to notice here that the probability of success will not be over-cooked by increasing the required number of iteration r similar to the common amplitude amplification techniques.

3.4 Adjustments on the proposed algorithm

During the above discussion, two problems will arise during the implementation of the proposed algorithm. The first one is to finding $|ax_2\rangle = |1\rangle$ for r times in a row which a critical issue in the success of the proposed algorithm to terminate in polynomial time. The second problem is that the value of d_s is not known in advance, where the value of $\Pr^{(1)}(ax_2 = 1)$ shown in Eq. 35 plays an important role in the success of finding $|ax_2\rangle = |1\rangle$ in the next iterations, this value depends heavily on the density of 1's, i.e., the ratio $\frac{d_s}{m}$.

Consider the case of a complete graph with even number of vertices, where the number of egdes $m = \frac{n(n-1)}{2}$ and all $|z_k\rangle$'s are equivalent and each can be taken as $|z_s\rangle$ then,

$$\Pr{}^{(1)}(ax_2 = 1) = M\alpha^2 \sin^2\left(\frac{d_s \pi}{2m}\right). \tag{42}$$

This case is an easy case where setting $m = d_s$ in mth root of σ_X will lead to a probability of success of certainty after a single iteration. Assuming a blind approach where d_s is not known, then this case represents the worst ratio $\frac{d_s}{m}$ where the probability of success will be ≈ 0.5 for sufficient large graph. Iterating the algorithm will not lead to any increase in the probability of both $|z_s\rangle$ and $|ax_2\rangle$.

In the following, adjustments on the proposed algorithm for the max-bisection and the min-bisection graph will be presented to overcome these problems, i.e., to be able to find $|ax_2\rangle = |1\rangle$ after the first iteration with the highest probability of success without a priori knowledge of d_s.

3.4.1 Adjustment for the max-bisection problem

In an arbitrary graph, the density of 1's will be $\frac{d_{\max 1}}{m}$. In the case of a complete graph, there are M states with 1-distance (d_k) equals to $\frac{n^2}{4}$. This case represents the worst density of 1's where the density will be $\frac{n^2}{2n(n-1)}$ slightly greater than 0.5 for arbitrary large n. Iterating the proposed algorithm will not amplify the amplitudes after arbitrary number of iterations. To overcome this problem, add $\mu_{\max}$ temporary qubits initialized to state $|1\rangle$ to the register $|c_0 c_1 ... c_{m-1}\rangle$ as follows,

$$|c_0 c_1 ... c_{m-1}\rangle \rightarrow \left|c_0 c_1 \ldots c_{m-1} c_m c_{m+1} \ldots c_{m+\mu_{\max}-1}\right\rangle, \tag{43}$$

so that the extended number of edges m_{ext} will be $m_{ext} = m + \mu_{\max}$ and $V = \sqrt[m_{ext}]{\sigma_X}$ will be used instead of $V = \sqrt[m]{\sigma_X}$ in the MAX operator, then the density of 1's will be $\frac{n^2+4\mu_{\max}}{2n(n-1)+4\mu_{\max}}$. To get a probability of success $\Pr_{\max}$ to find $|ax_2\rangle = |1\rangle$ after the first iteration,

$$\Pr{}^{(1)}(ax_2 = 1) = M\alpha^2 \sin^2\left(\frac{\pi\left(\frac{n^2}{4} + \mu_{\max}\right)}{2\left(\frac{n(n-1)}{2} + \mu_{\max}\right)}\right) \geq \Pr{}_{\max}, \tag{44}$$

then the required number of temporary qubits $\mu_{\max}$ is calculated as follows,

$$\mu_{\max} \geq \frac{1}{1-\omega}\left(\frac{n^2}{2}(2\omega - 1) - \frac{n}{2}\omega\right), \tag{45}$$

where $\omega = \frac{2}{\pi}\sin^{-1}\left(\sqrt{\frac{\Pr_{\max}}{M\alpha^2}}\right)$ and $\Pr_{\max} < M\alpha^2$, with $M\alpha^2 = 1$ so let $\Pr_{\max} = \delta M\alpha^2$ such that $0 < \delta < 1$. For example, if $\delta = 0.9$, then $\Pr^{(1)}$ $(ax_2 = 1)$ will be at least 90 % as shown in Fig. 5. To conclude, the problem of low density of 1's can be

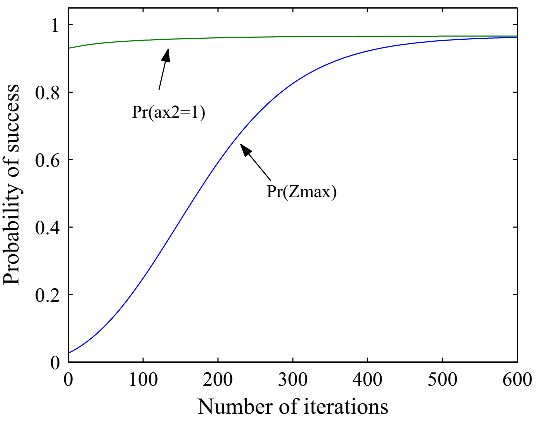

Fig. 5 The probability of success for a max-bisection instance of the graph shown in Fig. 1 with $n = 8$, $m = 12$, $\mu_{\max} = 31$ and $\delta = 0.9$, where the probability of success of $|ax_2\rangle$ is 0.9305 after the first iteration and with probability of success of 0.9662 after iterating the algorithm where the probability of success of $|z_{\max}\rangle$ is amplified to reach the probability of success of $|ax_2\rangle$

solved with a polynomial increase in the number of qubits to get the solution $|z_{\max}\rangle$ in $O\left(m_{ext}^2\right) = O\left(n^4\right)$ iterations with arbitrary high probability $\delta < 1$ to terminate in poly-time, i.e., to read $|ax_2\rangle = |1\rangle$ for r times in a row.

3.4.2 Adjustment for the min-bisection problem

Similar to the above approach, in an arbitrary graph, the density of 0's will be $\frac{d_{\max 0}}{m}$. In the case of a complete graph, there are M states with 0-distance (d_k) equals to $\frac{n(n-1)}{2} - \frac{n^2}{4}$. This case represents the worst density of 0's where the density will be $\frac{n-2}{2(n-1)}$ slightly less than 0.5 for arbitrary large n. Iterating the proposed algorithm will not lead to any amplification after arbitrary number of iterations. To overcome this problem, add $\mu_{\min}$ temporary qubits initialized to state $|0\rangle$ to the register $|c_0c_1...c_{m-1}\rangle$ as follows,

$$|c_0c_1...c_{m-1}\rangle \rightarrow \left|c_0c_1 \ldots c_{m-1}c_mc_{m+1} \ldots c_{m+\mu_{\min}-1}\right\rangle, \tag{46}$$

so that the extended number of edges m_{ext} will be $m_{ext} = m + \mu_{\min}$ and $V = \sqrt[m_{ext}]{\sigma_X}$ will be used instead of $V = \sqrt[m]{\sigma_X}$ in the MIN operator, then the density of 0's will be $\frac{n^2-2n+4\mu_{\min}}{2n(n-1)+4\mu_{\min}}$. To get a probability of success $\Pr_{\max}$ to find $|ax_2\rangle = |1\rangle$ after the first iteration,

$$\Pr^{(1)}(ax_2 = 1) = M\alpha^2 \sin^2\left(\frac{\pi\left(\frac{n(n-1)}{2} - \frac{n^2}{4} + \mu_{\min}\right)}{2\left(\frac{n(n-1)}{2} + \mu_{\min}\right)}\right) \geq \Pr_{\max}, \tag{47}$$

then the required number of temporary qubits $\mu_{\min}$ is calculated as follows,

$$\mu_{\min} \geq \frac{n^2}{4}\left(\frac{2\omega - 1}{1 - \omega}\right) + \frac{n}{2}, \tag{48}$$

where $\omega = \frac{2}{\pi}\sin^{-1}\left(\sqrt{\frac{\Pr_{\max}}{M\alpha^2}}\right)$ and $\Pr_{\max} < M\alpha^2$, with $M\alpha^2 = 1$ so let $\Pr_{\max}$ $= \delta M\alpha^2$ such that $0 < \delta < 1$. For example, if $\delta = 0.9$, then $\Pr^{(1)}$ $(ax_2 = 1)$ will be at least 90 %. To conclude similar to the case of the max-bisection graph, the problem of low density of 0's can be solved with a polynomial increase in the number of qubits, larger than the case of the max-bisection graph, to get the solution $|z_{\min}\rangle$ in $O\left(m_{ext}^2\right) = O\left(n^4\right)$ iterations with arbitrary high probability $\delta < 1$ to terminate in poly-time, i.e., to read $|ax_2\rangle = |1\rangle$ for r times in a row.

4 Conclusion

Given an undirected graph G with even number of vertices n and m unweighted edges, the paper proposed a bounded-error quantum polynomial-time (BQP) algorithm to solve the max-bisection problem and the min-bisection problem, where a general graph is considered for both problems.

The proposed algorithm uses a representation of the two problems as a Boolean constraint satisfaction problem, where the set of edges of a graph are represented as a set of constraints. The algorithm is divided into three stages: The first stage prepares a superposition of all possible equally sized graph partitions in $O\left(\sqrt[4]{n}\right)$ using an amplitude amplification technique that runs in $O\left(\sqrt{\frac{N}{M}}\right)$, for $N = 2^n$ and M is the number of possible graph partitions. The algorithm, in the second stage, evaluates the set of constraints for all possible graph partitions. In the third stage, the algorithm amplifies the amplitudes of the best graph bisection that achieves maximum/minimum satisfaction to the set of constraints using an amplitude amplification technique that applies an iterative partial negation where more negation is given to the set of constrains with more satisfied constrains and a partial measurement to amplify the set of constraints with more negation. The third stage runs in $O(m^2)$ and in the worst case runs in $O(n^4)$ for a dense graph. It is shown that the proposed algorithm achieves an arbitrary high probability of success of $1 - \epsilon$ for small $\epsilon > 0$ using a polynomial increase in the space resources by adding dummy constraints with predefined values to give more negation to the best graph bisection.

References

1. Armbruster, M.: Branch-and-Cut for a Semidefinite Relaxation of Large-Scale Minimum Bisection Problems. Ph.D. thesis, Technische Universitt Chemnitz (2007)
2. Armbruster, M., Fgenschuh, M., Helmberg, C., Martin, A.: A comparative study of linear and semidefinite branch-and-cut methods for solving the minimum graph bisection problem. In: Proceedings ofthe Conference on Integer Programming and Combinatorial Optimization (IPCO), LNCS, vol. 5035, pp. 112–124, (2008)
3. Bazgan, C., Karpinski, M.: On the complexity of global constraint satisfaction. In: Proceeding of 16th Annual International Symposium on Algorithms and Computation, vol. LNCS 3827, Springer, pp. 624–633 (2005)
4. Bazgan, C.: Paradigms of Combinatorial Optimization: Problems and New Approaches. In: Paschos, V.T. (ed) (Chapter 1: Optimal satisfiability), vol. 2, p. 25, Wiley-ISTE, (2010)

5. Bhatt, S.N., Leighton, F.T.: A framework for solving VLSI graph layout problems. J. Comput. Syst. Sci. **28**(2), 300–343 (1984)
6. Delling, D., Goldberg, A.V., Razenshteyn, I., Werneck, R.F.: Graph partitioning with natural cuts. In: IEEE, Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 1135–1146 (2011)
7. Delling, D., Werneck, R.F.: Faster customization of road networks. In: Proceedings of the International Symposium on Experimental Algorithms (SEA). LNCS, vol. 7933, pp. 30–42. Springer, Berlin (2013)
8. Delling, D., Fleischman, D., Goldberg, A.V., Razenshteyn, I., Werneck, R.F.: An exact combinatorial algorithm for minimum graph bisection. Math. Program. (2014). doi:10.1007/s10107-014-0811-z
9. Josep Daza, J., Kamiskib, M.: MAX-CUT and MAX-BISECTION are NP-hard on unit disk graphs. Theor. Comput. Sci. **377**(13), 271–276 (2007)
10. Feige, U., Langberg, M.: The RPR^2 rounding technique for semidefinite programs. J. Algorithms **60**, 1–23 (2006)
11. Feldmann, A.E., Widmayer, P.: An $O(n^4)$ time algorithm to compute the bisection width of solid grid graphs. In: Proceedings of the European Symposium on Algorithms (ESA), LNCS, vol. 6942, pp. 143–154, Springer, (2011)
12. Guruswami, V., Makarychev, Y., Raghavendra, P., Steurer, D., Zhou, Y.: Finding almost-perfect graph bisections. In: Proceedings of ICS, pp. 321–337 (2011)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, London (1979)
14. Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified NP-complete graph problems. Theor. Comput. Sci. **1**, 237–267 (1976)
15. Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. **79**, 325 (1997)
16. Hager, W.W., Phan, D.T., Zhang, H.: An exact algorithm for graph partitioning. Math. Program. **137**, 531–556 (2013)
17. Hendrickson, B., Leland, R.: An improved spectral graph partitioning algorithm for mapping parallel computations. SIAM J. Sci. Comput. **16**(2), 452–469 (1995)
18. Hein, M., Bhler, T.: An inverse power method for nonlinear eigen problems with applications in 1-spectral clustering and sparse PCA. In: Proceedings Advances in Neural Information Processing Systems (NIPS), pp. 847–855 (2010)
19. Høyer, P.: Arbitrary phases in quantum amplitude amplification. Phys. Rev. A **62**, 52304 (2000)
20. Jansen, K., Karpinski, M., Lingas, A., Seidel, E.: Polynomial time approximation schemes for MAX-BISECTION on planar and geometric graphs. SIAM J. Comput. **35**, 110–119 (2005)
21. Johnson, E., Mehrotra, A., Nemhauser, G.: Min-cut clustering. Math. Program. **62**, 133–152 (1993)
22. Kwatra, V., Schdl, A., Essa, I., Turk, G., Bobick, A.: Graph cut textures: image and video synthesis using graph cuts. ACM Trans. Graph. **22**, 277–286 (2003)
23. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. Econometrica **28**(3), 497–520 (1960)
24. Lipton, R.J., Tarjan, R.: Applications of a planar separator theorem. SIAM J. Comput. **9**, 615–627 (1980)
25. Malewicz, G., Austern, M.H., Bik, A.J., Dehnert, J.C., Horn, I., Leiser, N., Czajkowski, G.: Pregel, a system for large-scale graph processing. In: PODC, p. 6. ACM, (2009)
26. Meyerhenke, H., Monien, B., Sauerwald, T.: A new diffusion-based multilevel algorithm for computing graph partitions. J Parallel Distrib. Comput. **69**(9), 750–761 (2009)
27. Rcke, H.: Optimal hierarchical decompositions for congestion minimization in networks. In: Proceedings of the ACM Symposium on Theory of Computing (STOC), pp. 255–263. ACM Press, New York (2008)
28. Raghavendra, P., Tan, N.: Approximating CSPs with global cardinality constraints using SDP hierarchies. In: Proceedings of SODA, pp. 373–387 (2012)
29. Rivlin, T.J.: Chebyshev Polynomials. Wiley, New York (1990)
30. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
31. Sanders, P., Schulz, C.: Distributed evolutionary graph partitioning. In: Proceedings of the Algorithm Engineering and Experiments (ALENEX), pp. 16–29. SIAM (2012)

32. Sanders, P., Schulz, C.: Think locally, act globally: highly balanced graph partitioning. In: Proceedings of the International Symposium on Experimental Algorithms (SEA). LNCS, vol. 7933, pp. 164–175. Springer, Berlin (2013)
33. Xu, Z., Du, D., Xu, D.: Improved approximation algorithms for the max-bisection and the disjoint 2-catalog segmentation problems. J. Comb. Optim. **27**(2), 315–327 (2014)
34. Younes, A., Rowe, J., Miller, J.: Enhanced quantum searching via entanglement and partial diffusion. Phys. D. **237**(8), 1074–1078 (2007)
35. Younes, A.: Towards more reliable fixed phase quantum search algorithm. Appl. Math. Inf. Sci. **7**(1), 93–98 (2013)