Engineering Grover Adaptive Search: Exploring the Degrees of Freedom for Efficient QUBO Solving

Luigi Giuffrida[1], Deborah Volpe[1*], Giovanni Amedeo Cirillo[1], Maurizio Zamboni[1] and Giovanna Turvani[1]

[1*]Department of Electronics and Telecommunications, Politecnico di Torino, Corso Castelfidardo, 39, Torino, 10129, Italy.

*Corresponding author(s). E-mail(s): deborah.volpe@polito.it;
Contributing authors: luigi.giuffrida@studenti.polito.it;
giovanni_cirillo@polito.it; maurizio.zamboni@polito.it;
giovanna.turvani@polito.it;

1 Introduction

In this supplementary materials document, the QUBO formulation is taken up and deepened, presenting all problems employed for benchmarking and their characteristics. Moreover, additional results are reported and commented on, and further indications about how to read the results are reported.

2 Other relevant QUBO problems

Quadratic Unconstrained Binary Optimization (QUBO) is a mathematical formulation able to represent an exceptional variety of combinatorial optimization problems [1]. **Quadratic** refers to the highest power applied on the involved variables, which can assume only 0 and 1 values (thus **Binary**). **Unconstrained** means that no constraints are applied in a standard way, and **Optimization** is because this model is used to minimize or maximize the obtained objective function.
It can be expressed as:

$$\text{minimize/maximize} \quad y = \mathbf{x}^t \cdot Q \cdot \mathbf{x}\,, \tag{1}$$

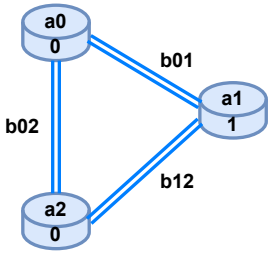


Fig. 1: An example of three-variable QUBO model

where $\mathbf{x}$ is a vector of binary variables (*e.g.* [0,1,1,0,1]) and Q is a square matrix of constants, depending on the problem. Starting from the symmetric form of the matrix, it is possible to obtain the upper triangular one by replacing the q_{ij} coefficient with $q_{ij} + q_{ji}$ for all i and $j > i$ and then replace all q_{ij} for $j < i$ with zero. Equation 2 reports the symmetric matrix of a simple four-variable problem, then in Equation 3 the upper triangular matrix referred to the same problem is shown.

$$Q_{\text{symmetric}} = \begin{pmatrix} a_0 & \frac{b_{01}}{2} & \frac{b_{02}}{2} & \frac{b_{03}}{2} \\ \frac{b_{01}}{2} & a_1 & \frac{b_{12}}{2} & \frac{b_{13}}{2} \\ \frac{b_{02}}{2} & \frac{b_{12}}{2} & a_2 & \frac{b_{23}}{2} \\ \frac{b_{03}}{2} & \frac{b_{13}}{2} & \frac{b_{23}}{2} & a_3 \end{pmatrix}. \tag{2}$$

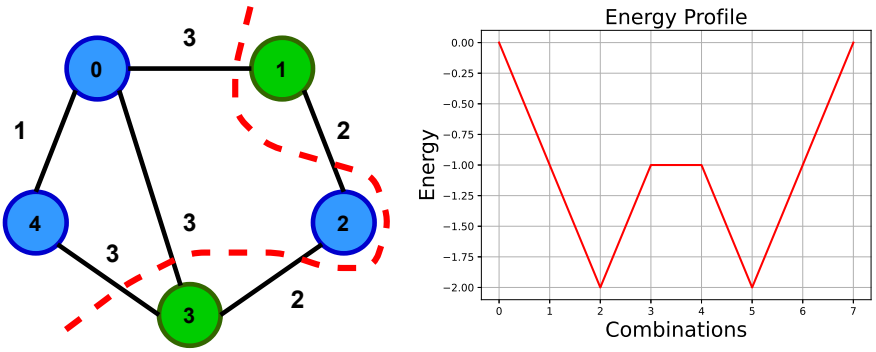
$$Q_{\text{triangular}} = \begin{pmatrix} a_0 & b_{01} & b_{02} & b_{03} \\ 0 & a_1 & b_{12} & b_{13} \\ 0 & 0 & a_2 & b_{23} \\ 0 & 0 & 0 & a_3 \end{pmatrix}. \tag{3}$$

The QUBO can be also written as:

$$\text{Obj}(c, a_i, b_{ij}, q_i) = c + \sum_i q_i \cdot a_i + \sum_{i<j} b_{ij} \cdot q_i q_j, \tag{4}$$

where $q_i \in [0, 1]$ is a binary variable, $q_i q_j$ is a coupler that allows two variables to influence each other, a_i is a weight or bias associated with a single variable and b_{ij} is a strength which controls the influence of variables i and j. The a_i and b_{ij} are respectively the terms on the main diagonal and off-diagonal of Q. Despite what the name would suggest, problems including additional constraints can be formulated with the QUBO model, introducing quadratic penalties to the objective function:

$$\text{minimize/maximize} \quad y = f(\mathbf{x}) + \lambda g(\mathbf{x}), \tag{5}$$



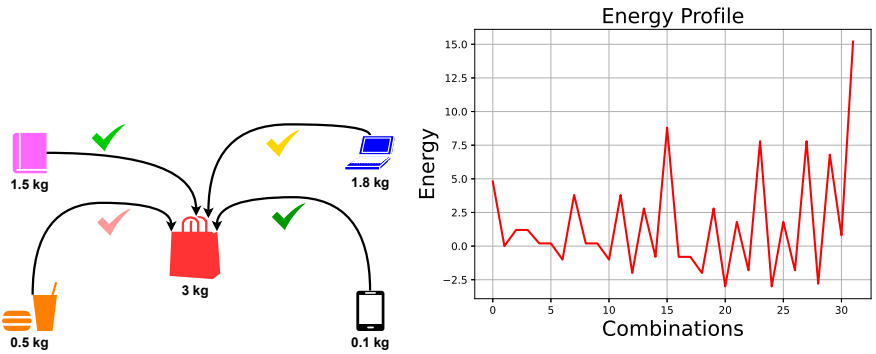
(a) Five-node max-cut. (b) Three-node max-cut energy profile.

Fig. 2: Max-cut problem.

where λ is a positive penalty parameter to be multiplied by the constraint function $g(\mathbf{x})$, which is evaluated during the standard execution of the optimizer. However, the choice of λ is critical; if it is too high, the cost function becomes too flat, thus making it difficult to evaluate the effective quality of the solutions, while a too low value of its could imply the negligibly of the corresponding constraint $g(\mathbf{x})$ and consequently the unreliability of the obtained solutions.
A good way to find an acceptable value is to perform an estimation of the original objective function and take λ as a certain percentage of it (usually in the range **75%-150%**) [1]. QUBO formulation can be assisted by Python libraries, such as qubovert [2], PyQUBO [3, 4] and dimod [5], characterized by routines for automatic insertion of some relevant constraints in the problem function and by the possibility to interface the defined QUBO problems with different solvers, *e.g.* based on simulated annealing [6].
In the following subsections, relevant problems and their QUBO formulations are reported. Each of them was employed as a benchmark for evaluating the performance of the new proposed strategies for the Grover Adaptive Search (GAS) solver.

2.1 Max-cut problem

One of the most relevant CO problems is the **maxcut** [7] [8] (Figure 2a), which can be described as partitioning a graph into two complementary subsets S and $\overline{S}$, maximizing the sum of weights over all the edges across the two vertex subsets. This problem can be used to solve several real-world problems in network design, statistical physics, VLSI design and circuit layout design [9].
In order to obtain the QUBO formulation, the introduction of a binary variable for each node — satisfying $x_j = 1$ if the associated node j is in one set and $x_j = 0$ if it is in the other set — is required. Viewing a cut as severing edges joining two sets, the quantity $x_j + x_i - 2x_j x_i$ identifies whether the edge (i, j)



(a) Four-object knapsack.

(b) Three-object (five-variable) knapsack energy profile.

Fig. 3: Knapsack problem.

is in the cut. If the expression is equal to 1, then exactly one among x_i and x_j equals 1, which implies edge (i, j) is in the cut; otherwise, when it is equal to zero, the edge is not in the cut. Summing the contributions of each edge, the following QUBO formulation can be obtained:

$$\text{Maximize} \quad y = \sum_{(i,j)\in E} w_{i,j} \cdot (x_j + x_i - 2x_j x_i)\,, \tag{6}$$

where $w_{i,j}$ is the weight of the edge that connects the i-th and the j-th node. Max-cut problems are characterized by symmetric energy profiles (as shown in Figure 2b); in fact, a solution and its complement (*e.g.* [0,1,1,0,1] and [1,0,0,1,0]) have the same energy because the obtained two sub-sets are interchangeable. Consequently, the energy profile is characterized by at least a repetition of each value of the objective function, which could make these problems unfeasible for varying the number of Grover rotations in GAS according to **fixed pattern** strategy. Moreover, the starting cost function is characterized by only negative or null values, which means that it is not the optimal condition for the Grover search application, so a pre-conditioning step can be particularly useful and effective in this case.

2.2 Knapsack problem

Another well-known CO problem is the **knapsack** [10] one, whose conceptual scheme is reported in Figure 3a. Its goal consists in defining — for a set of objects X, each of which is labelled as x_i and is characterized by a weight w_i — the best sub-set to be put into a bag, ensuring that the total weight does

not exceed a threshold W:

$$0 < \sum_{i=1}^{\dim(X)} w_i x_i \leq W \,. \tag{7}$$

Since a *strict* inequality is present, QUBO formulation requires auxiliary variables [1, 11], whose number could be partially limited by involving integer weights instead of real, which implies more sophisticated number representations as floating-point. The cost function can be finally written as:

$$f_{\text{knapsack}}(\mathbf{x}) = f_{\text{inequality}}(\mathbf{x}) - \sum_i p_i x_i \,, \tag{8}$$

where $p_i x_i$ are preference contributions to force the choice of specific objects. In particular, higher is the p_i associated with the i^{th} object, higher is the preference for its insertion. When the terms p_i are all equal, the problem requires maximizing the total number of objects inserted in the knapsack, considering the weight limitation.
This problem can also be employed for an abstract description of a resource exploitation optimization problem in an industrial environment.
An example of a cost function profile for this problem is reported in Figure 3b. In this example, the number of positive values of the starting cost function is higher than the number of negative ones, which is an ideal condition for the Grover Search (GS) algorithm application. However, the number of repeated values is relatively high, which can compromise the effectiveness of some strategies, such as **fixed pattern** variation for the number of Grover rotations.

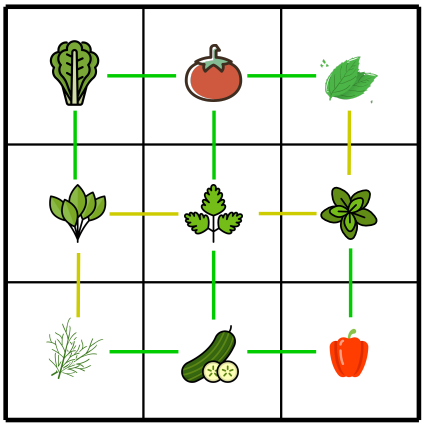
2.3 Garden optimization

Another important family is that associated with the **well-positioning problems**, as the **garden optimization** [12], whose target is to find an optimal placement of n plants in n pots (as shown in Figures 4a and 4b). The associated QUBO problem involves n^2 variables, one for each plant-pot pair. The x_{ij} variable assumes value one if the plant j is in the pot i. A placement is valid if some requirements are satisfied. First of all, each pot has to be filled with exactly one plant:

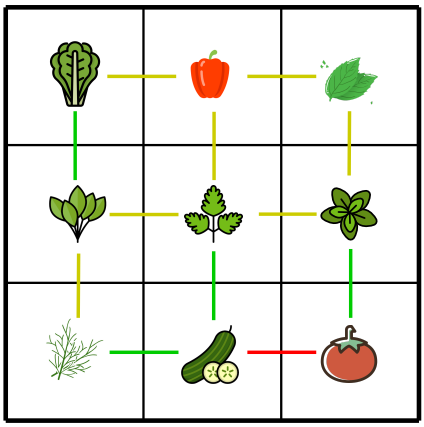
$$\forall i: \quad \sum_{j=1}^{n} x_{ij} = 1 \,. \tag{9}$$

Then, all available plants must be placed in the garden:

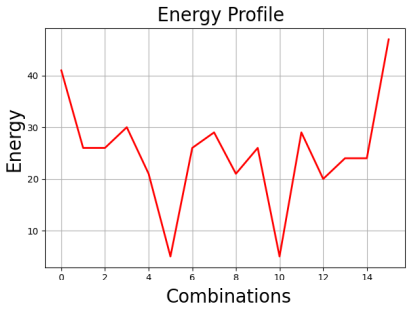
$$\forall j: \quad \sum_{i=1}^{n} x_{ij} = 1 \,. \tag{10}$$



(a) Optimal solution of a nine-pot garden optimization.



(b) Not optimal solution of a nine-pot garden optimization.



(c) Two-pot (four-variable) garden optimization energy profile.

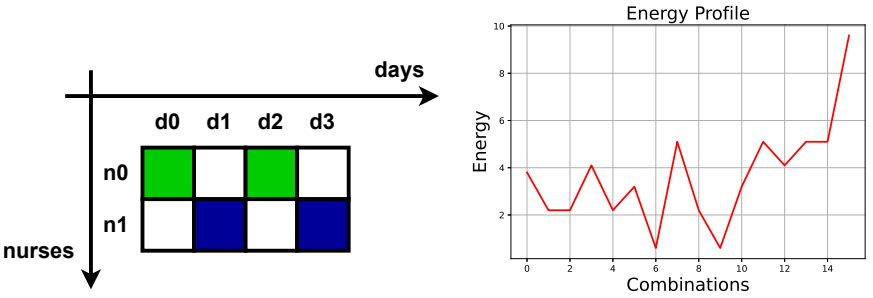
Fig. 4: Garden optimization problem.

In the end, tall plants shall not shadow smaller ones:

$$\forall i, j: \quad (i \mod 2 - s_j)^2 x_{i,j} = 0, \tag{11}$$

where $s_j \in [0, 1]$ is a binary flag assuming value 0 (1) if the j^{th} plant is tall (small), forcing it in even (odd) rows.
The figure of merit for placement optimization is the affinity among plant species. Indeed, some species can be placed close to each other, while others cannot.
The final cost function can be written as:

$$f_{\text{garden}}(\mathbf{x}) = -\sum_{i,i'=1}^{n} J_{ii'} \left(1 + \sum_{j,j'=1}^{n} x_{ij} C_{jj'} x_{i'j'}\right) + \lambda_1 \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{n} x_{ij}\right)^2 +$$



(a) Two-nurse and four-day (eight-variable) nurse scheduling problem.

(b) Two-nurse and two-day (four-variable) nurse scheduling problem energy profile.

Fig. 5: Nurse scheduling problem.

$$+\lambda_2 \sum_{j=1}^{n}\left(1-\sum_{i=1}^{n} x_{ij}\right)^2 + \lambda_3 \sum_{i=1}^{n}\sum_{j=1}^{n}(i \text{ rem } 2 - s_j)^2 x_{ij}, \quad (12)$$

where $J_{ii'}$ and $C_{jj'}$ are the terms of the adjacent J and companions C matrices, respectively. $J_{ii'}$ is equal to 1 if pots i and i' are adjacent, while $C_{jj'}$ is equal to -1 when there is a friendly relationship among plants j and j' plants, and equal to 0 and 1 in case of neutral and antagonist relationships respectively. In Figure 4a an example of an optimal solution, in which the number of friendly relationships (green) is maximized, is reported, then in Figure 4b a not optimal solution is reported, where an antagonist relationship (red) is present, and the number of friendly relationships is not the highest possible. In Figure 4c, an example of problem energy profile is reported. It is possible to notice that all the values of the initial cost function are positive, which is not the best starting condition for Grover search, so a pre-conditioning step could be effective for this type of problem and that the number of repetitions is very limited, thus making effective the **fixed pattern strategy** for number of Grover rotation variation, as shown in the article.

2.4 Nurse scheduling optimization problem

The **timetabling problems**, like **nurse scheduling optimization**[13], can be written in QUBO formulation. The target is to find the optimal schedule for nurses working in a hospital over a fixed timetable of shifts (Figure 5a). Given N nurses and D working days, the associated QUBO model involves $N \cdot D$ variables, one for each nurse-day pair, which is equal to 1 if the n^{th} nurse works in the d^{th} day. Three constraints characterize this problem. The first one is called *hard shift constraint* and requires that the schedule has to assure in each day d a nurse effort $\sum_{n=1}^{N} E(n)x_{n,d}$ sufficient to satisfy the associated

workload $W(d)$:

$$\forall d: \quad \sum_{n=1}^{N} E(n) x_{n,d} = W(d)\,. \tag{13}$$

The second one is called *hard nurse constraint*, which ensures that no nurse works for two consecutive days. A positive correlation constant a is used to penalize the schedule in which the n^{th} nurse works on two consecutive days:

$$f_{\text{nurse}}(\mathbf{x}) = \sum_{n=1}^{N} \sum_{d=1}^{D-1} a \cdot x_{n,d} \cdot x_{n,d+1}\,. \tag{14}$$

The last one, which is called the *soft nurse constraint*, assures that all nurses should work approximately the same number of days $F = D/N$:

$$\forall n: \quad \sum_{d=1}^{D} x_{n,d} = F\,. \tag{15}$$

The final cost function can be written as:

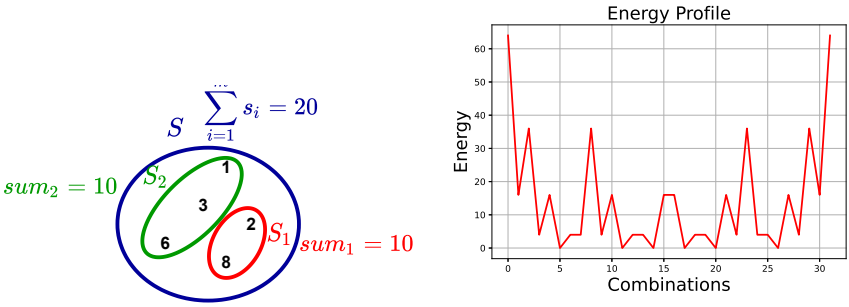
$$f_{\text{nurse}}(\mathbf{x}) = \sum_{n=1}^{N} \sum_{d=1}^{D-1} a x_{n,d} x_{n,d+1} + \lambda_1 \sum_{n=1}^{N} \left(\sum_{d=1}^{D} x_{n,d} - F \right)^2 + \\ + \lambda_2 \sum_{d=1}^{D} \left(\sum_{n=1}^{N} E(n) x_{n,d} - W(d) \right)^2. \tag{16}$$

Figure 5b shows an example of a problem energy profile. It is possible to observe that all the values of the starting cost function are positive, which is not the best starting condition for Grover search, so a pre-conditioning step could be effective for this type of problem and that the number of repetition is very limited, thus making effective the **fixed pattern strategy** for number of Grover rotation variation, as shown in the article.

2.5 Number partitioning problem

The **number partitioning** [1] is an optimization problem in which the target is to partition a set $S = \{s_1, s_2, ..., s_m\}$ of m positive integers into two subsets S_1 and S_2, such that the sum of numbers in each one is equal (as shown in Figure 6a). The QUBO formulation involves m binary variables, one for each number in the set S. The variable x_i assumes a value of 1 if the i^{th} number is assigned to the subset S_1 and 0 otherwise. Consequently, the sum for the first subset is given by:

$$\text{sum}_1 = \sum_{i=1}^{m} s_i x_i\,, \tag{17}$$



(a) Five-variable number partitioning problem.

(b) Five-number number partitioning problem energy profile.

Fig. 6: Number partitioning optimization problem.

while the sum for the second is:

$$\text{sum}_2 = \sum_{i=1}^{m} s_i - \sum_{i=1}^{m} s_i x_i \,. \tag{18}$$

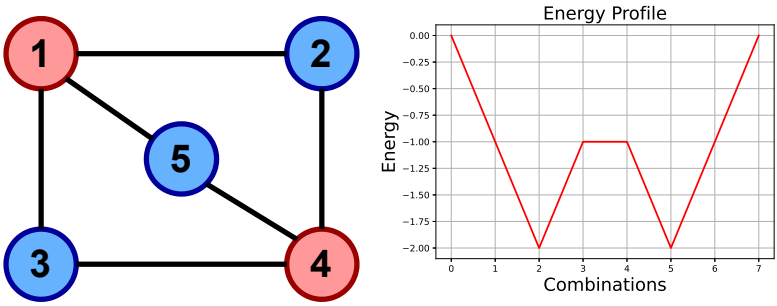
The difference among the two sum is so given by:

$$\text{diff} = \text{sum}_2 - \text{sum}_1 = \sum_{i=1}^{m} s_i - 2\sum_{i=1}^{m} s_i x_i \,, \tag{19}$$

which must be minimized for achieving the target of the problem. Therefore, the final cost function is equal to:

$$f_{\text{number}}(\mathbf{x}) = \left(\sum_{i=1}^{m} s_i - 2\sum_{i=1}^{m} s_i x_i\right)^2 . \tag{20}$$

Number partitioning problems are characterized by symmetric energy profiles (as shown in Figure 6b); in fact, a solution and its complement (*e.g.* [0,1,1,0,1] and [1,0,0,1,0]) have the same energy because the obtained two sub-sets are interchangeable. Consequently, the energy profile is characterized by at least a repetition of each value of the objective function, which could make these problems unfeasible for varying the number of Grover rotations according to **fixed pattern** strategy. Moreover, the starting cost function is characterized by only positive or null values, which means that it is not the optimal condition for the Grover search application, and so a pre-conditioning step can be particularly useful and effective in this case.



(a) Five-nodes minimum vertex cover problem.

(b) Three-nodes minimum vertex cover problem energy profile.

Fig. 7: Minimum vertex cover optimization problem.

2.6 Minimum vertex cover problem

Another optimization problem is the **minimum vertex cover**[1], which involves a graph as the max-cut one. Considering an undirected graph with a set of vertices V and edges E, a vertex cover is the subset of vertices such that each edge is incident to at least one vertex of the subset itself. Therefore, the problem target is to find a cover with the minimum number of vertices in the subset (as shown in Figure 7a). The QUBO formulation involves one binary variable for each vertex x_i, which assumes value one if the node is in the cover, i.e. in the subset. The cost function is composed of two parts; the former optimizes the number of nodes in the subset:

$$\text{Minimize} \quad y_1 = \sum_{i \in V} x_i \,, \tag{21}$$

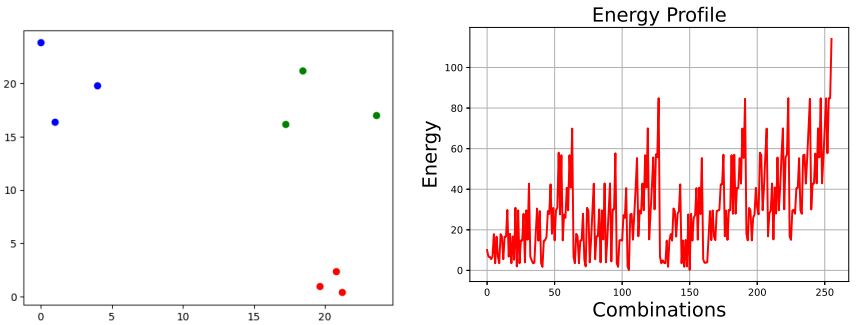
while the latter assures that each edge is incident to at least one vertex in the subset, imposing that for each couple of nodes i and j connected $x_i + x_j$ is equal to or greater than 1:

$$y_2 = \sum_{i,j \in E} \Big(1 - x_i - x_j + x_i x_j \Big) \,. \tag{22}$$

Consequently the final cost function is equal to

$$f_{\text{vertex}}(\mathbf{x}) = \sum_{i \in V} x_i + \lambda \sum_{i,j \in E} \Big(1 - x_i - x_j + x_i x_j \Big) \,. \tag{23}$$

Minimum vertex cover problems are characterized by the energy profiles shown in Figure 7b.



(a) Nine-data and three-cluster (27-variable) data clustering optimization.

(b) Two-data and two-cluster (four-variable) data clustering optimization energy profile.

Fig. 8: Data clustering optimization problem.

2.7 Cluster separation problem

Data clustering [14] is an important task of unsupervised learning whose target is to divide a group of N data into K subgroups (cluster) of similar elements, where similarity criteria can be simply the smallest distance (as shown in Figure 8b). This problem can be written in terms of an optimization problem.

The simplest possible QUBO formulation involves $N \cdot K$ binary variables, one for each data-cluster pair. The x_{nk} variable assumes value one if the n^{th} data is in the k^{th} cluster. A valid clustering has to satisfy some requirements, which depend on the application. The simplest version of these is reported in the following. First of all, each data can be assigned to exactly one cluster:

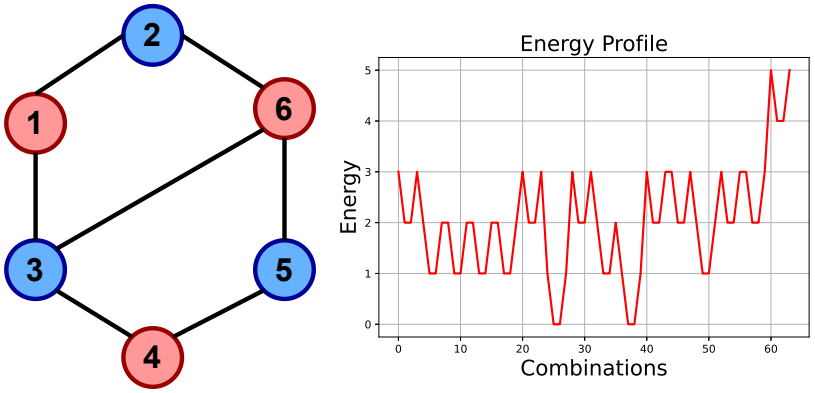
$$\forall n : \quad \sum_{k=1}^{K} x_{nk} = 1 \,. \tag{24}$$

Then, data must be equally distributed among the clusters:

$$\forall k : \quad \sum_{n=1}^{N} x_{nk} = \frac{N}{K} \,. \tag{25}$$

The figure of merit for optimization is to minimize the total distance D among data in all cluster:

$$D = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} d_{ij} x_{ik} x_{jk} \,, \tag{26}$$



(a) Two-colour and six-nodes (twelve-variable) graph coloring problem.

(b) Two-colour and three-nodes (six-variable) graph coloring energy profile.

Fig. 9: Graph coloring optimization problem.

where the distance among the i^{th} and the j^{th} data is computed as:

$$d_{ij} = \sqrt{(c1_i - c1_j)^2 + (c2_i - c2_j)^2 \ldots}\,, \tag{27}$$

where $c1, c2 \ldots$ are the coordinates (or features) associated with each data. In conclusion, the final cost function can be written as:

$$f_{\text{cluser}}(\mathbf{x}) = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{K} d_{ij} x_{ik} x_{jk} + \lambda_1 \sum_{n=1}^{N}\left(\sum_{k=1}^{K} x_{nk} - 1\right)^2 + \\ + \lambda_2 \sum_{k=1}^{K}\left(\sum_{n=1}^{N} x_{nk} - \frac{N}{K}\right)^2. \tag{28}$$

In Figure 8b an example of problem energy profile is reported. It is possible to notice that all the values of the starting cost function are positive, which is not the best starting condition for the Grover search, so that a pre-conditioning step could be effective for this type of problem and that the number of repetitions is usually expected to be limited, thus making effective the **fixed pattern strategy** for the number of Grover rotation variation.

2.8 Graph coloring problem

Another optimization problem is the **graph coloring**[1], which involves a graph as the max-cut and minimum vertex cover one. The problem target is to assign different colour labels to adjacent nodes, as shown in Figure 9a. It has a wide range of applications in industrial and scientific fields, e.g., printed circuit

design. Considering K colours and N nodes, the QUBO formulation requires $N \cdot K$ binary variable, i.e. one for each node-colour pair, which assumes value one if the k^{th} colour is assigned to the n^{th} node. A valid solution has to satisfy some requirements. First of all, each node has to be assigned to exactly one colour:

$$\forall n : \quad \sum_{k=1}^{K} x_{nk} = 1 \,. \tag{29}$$

Moreover, adjacent nodes have to be assigned to different colours:

$$\forall k : \quad \forall (i,j) \quad \text{adjacent node} : \quad x_{ik} + x_{jk} \leq 1 \,, \tag{30}$$

which can be expressed as:

$$\forall k : \quad \sum_{i,j \in E} x_{ik} x_{jk} = 0 \,, \tag{31}$$

where E is the set of edges of the graph. In conclusion, the final cost function can be written as:

$$f_{\text{coloring}}(\mathbf{x}) = \lambda_1 \sum_{n=1}^{N} \left(\sum_{k=1}^{K} x_{nk} - 1 \right)^2 + \lambda_2 \sum_{k=1}^{K} \sum_{i,j \in E} x_{ik} x_{jk} \,. \tag{32}$$

In Figure 9b an example of problem energy profile is reported. It is possible to notice that all the values of the starting cost function are positive, which is not the best starting condition for the Grover search, so a pre-conditioning step could be effective for this type of problem and that the number of repetitions is quite high, which can compromise the effectiveness of some strategies, such as **fixed pattern** variation for the number of Grover rotations.

3 Additional Results

This section provides additional results and some guidelines for reading them. All tests are performed by using for the quantum part the **QasmSimulator** available in Qiskit, which runs locally, and by using as execution platform a single-process Intel(R) Xeon(R) Gold 6134 CPU @ 3.20GHz opta-core, Model 85 [15] with a memory of 10296102+ KiB. The maximum size of considered problems is limited by the number of simulable qubits on the available platform, which is twenty. Consequently, considering that at least half of the qubits are devolved for the value register, the highest number of binary variables involved was eight.

Each problem was solved by exploiting all possible strategies combination for varying the number of Grover rotations and for dynamically modifying the threshold associated with the stop condition. A brief recap of the tested strategies is reported in Tables 1 and 2 and a brief recap of the tested strategies is reported are already present in the literature, and which of these are presented

Table 1: Strategies supported for varying the number of Grover rotations for each Grover search execution. The **fixed pattern** and **random sat** strategies are already present in literature. Then all the other ones are presented in the article for the first time.

Strategy	Update method
Random sat [16] (**RND sat**)	r is randomly sampled from an interval, initialized to $[0, 1]$, whenever the cost function is shifted and is increased at every positive value sampled. The upper bound of the interval is saturated $m_{\text{new}} = \min\left\{m_{\text{old}} + 1,\, 2^{n_{\text{key}}/2}\right\}$, where n_{key} is the number of binary variables. Available in Qiskit (**QSKT**)
Random noSat (**RND noSat**)	r is randomly sampled from an interval, initialized to $[0, 1]$, whenever the cost function is shifted and is increased at every positive value sampled.
Linear sat (**LIN sat**)	r is set to 0 when a negative sample is obtained and is linearly increased whenever a positive one is measured. The upper bound of the interval is saturated $m_{\text{new}} = \min\left\{m_{\text{old}} + 1,\, 2^{n_{\text{key}}/2}\right\}$, where n_{key} is the number of binary variables.
Linear noSat (**LIN noSat**)	r is set to 0 when a negative sample is obtained and is linearly increased whenever a positive one is measured.
Fixed pattern [17] (**PAT**)	r at each GS call is read by a list of pre-computed values.

in the article for the first time.

Our preliminary simulation studies showed that the optimal combination of strategies depends on the energy profile of the considered problem. For example, taking into account the success probability, the **pattern** strategy performed well with jagged and irregular energy profiles, such as the garden problem. In contrast, other strategies performed better with more regular energy profiles, such as the max-cut problem. This is consistent with results reported in [18].

Generally, both r and t selection strategies affect the success probability. Choosing the proper r permits to perform an effective GS, while a fine-tuned t permits to avoid the early stopping of the GAS and the declaration of sub-optimal minima.

3.1 How read the results

The analyzed results are reported both in tabular and graphical forms. The figures of merit considered in the tables are chosen to consider the trade-off

Table 2: Strategies supported for the stop condition management. The article proposes dynamic strategies (linear, logarithmic and adaptive) for the first time. Then the fixed threshold was already employed in the literature.

Strategy	Update method	When update
FIXED (**FIX**)	t_{fixed}	Never
Linear (**LIN**)	$t_{\text{new}} = \left\lceil -\frac{t_{\max}}{t_{\min}} \frac{n_S}{v} + t_{\max} \right\rceil$ where $t_{\min}$ and $t_{\max}$ are the minimum and maximum values of t, n_S is a counter of the executed classical iterations and v is a control parameter of the linear scaling slope	whenever a negative value of the cost function is measured
Logarithmic (**LOG**)	$t_{\text{new}} = \left\lceil \frac{\log_{10}(1.1)(t_{\max}-t_{\min})}{\log_{10}(1.1+n_S/v)} + (t_{\min}-1) \right\rceil$ where $t_{\min}$ and $t_{\max}$ are the minimum and maximum values of t, n_S is a counter of the executed classical iterations and v is a control parameter of the logarithmic scaling slope	whenever a negative value of the cost function is measured
Adaptive (**ADPT**)	$t_{\text{new}} = t_{\text{old}} \cdot 1.2$, if a negative value is sampled after $0.8 \cdot t_{\text{old}}$ or more positive ones; $t_{\text{new}} = t_{\text{old}} \cdot 0.8$, if a negative value is sampled after $0.2 \cdot t_{\text{old}}$ or less positive ones	whenever a negative value of the cost function is measured

between the success probability and the time required for obtaining a solution. In particular, the **probability of success** (**POS**) — i.e. the probability of obtaining the optimal solution — **the average number of total Grover rotations** per GAS execution $R = \sum_{\textbf{GS calls}} r$, and the **average number of classical iterations** (C) are considered. The **the average number of total Grover rotations**, specifically, is a metric for estimating the amount of quantum computation required for solving the problem, while the **average number of classical iterations** is a metric for recognizing the total number of GAS iterations required for solving a problem. All these metrics can be computed by repeating the same problem execution with the same solver multiple times.

The graphical representation gives a visual representation of the compromise between accuracy and execution time of each solver. The cumulative distributions permit to understand if a strategies combination provides results closer to

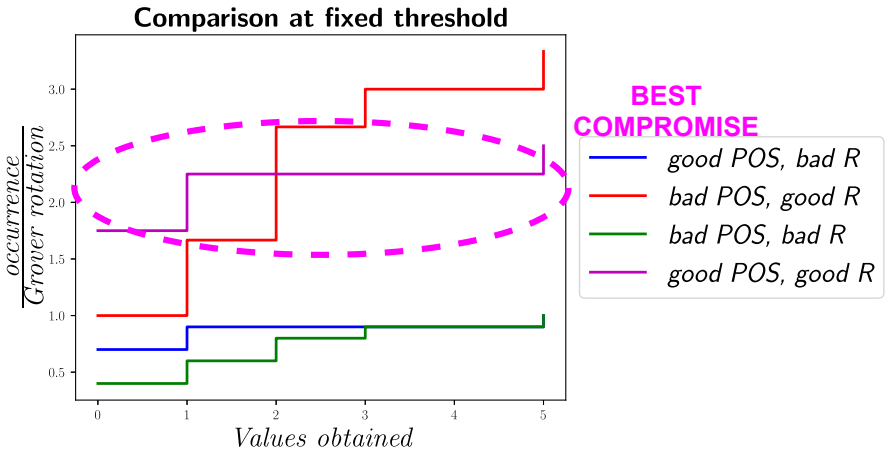
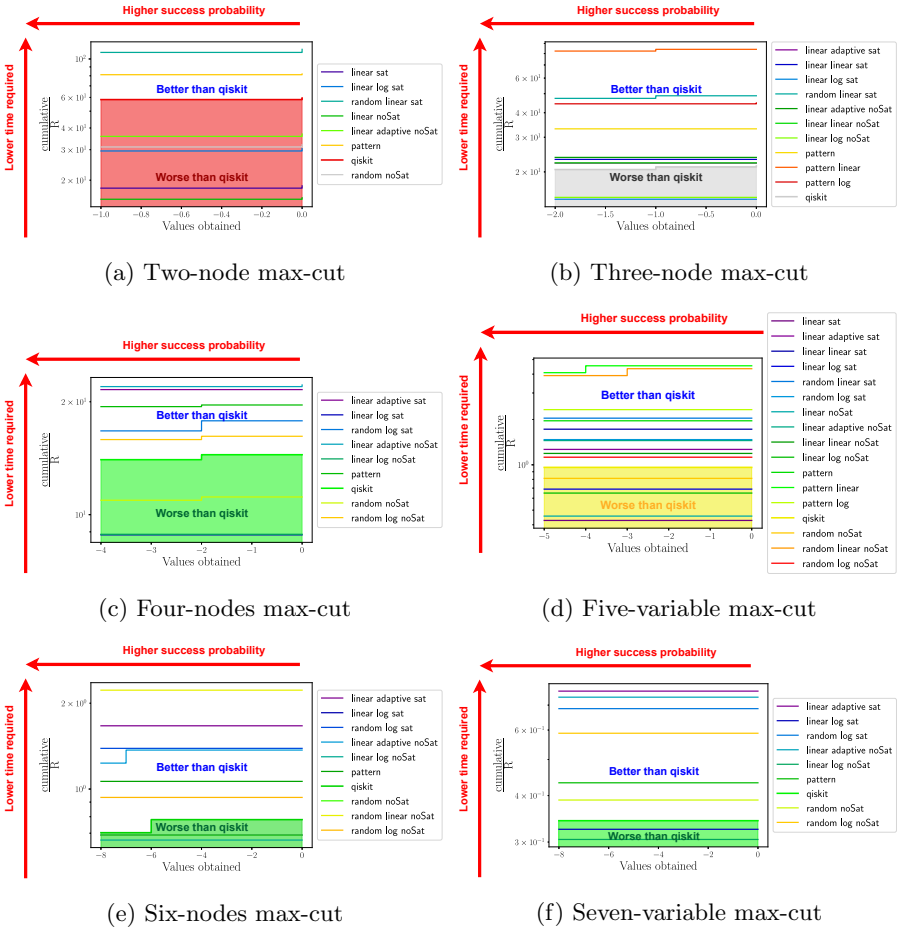


Fig. 10: Example of the graph in the four possible extreme conditions: good success probability (POS) but a high number of Grover rotations R, bad success probability (POS) but a low number of Grover rotations R, bad success probability (POS) but a high number of Grover rotations R and good success probability (POS) but low number of Grover rotations R. The best compromise, the last mentioned case, is also underlined.

the optimal one. At the same time, the scaling factor $\frac{1}{R}$ recognizes the strategies combination with lower execution time, thus also allowing a potential time distinction between two strategies combinations with analogous cumulative distributions. For these reasons, the best strategies can be considered those where the cumulative increases more significantly on the left part of the plot (where the lowest cost values are located) and which are characterized by higher values on the y-axis (associated with a lower execution time). Moreover, to make more accessible the comparison with Qiskit, the region of the plot below its cumulative is colored with the same color.
A meaningful example about how to read the reported graph is reported in Figure 10. In particular, the weighted cumulative is presented for the four possible extreme cases:

- *good POS bad R*, in which the success probability reported is equal to 70%, but with an average number of average total Grover rotations equal to 10;
- *bad POS good R*, in which the success probability reported is equal to 30%, but with an average number of average total Grover rotations equal to 3;
- *bad POS bad R*, in which the success probability reported is equal to 40%, but with an average number of total average Grover rotationss equal to 10;
- *good POS good R*, in which the success probability reported is equal to 70%, but with an average number of total average Grover rotationss equal to 4.

It is possible to recognize the weighted cumulative distributions with lower success probability because these still grow long after the first step. The ones



(a) Two-node max-cut

(b) Three-node max-cut

(c) Four-nodes max-cut

(d) Five-variable max-cut

(e) Six-nodes max-cut

(f) Seven-variable max-cut

Fig. 11: Cumulative distributions of the values obtained at the end of GAS, divided by the average total number of Grover rotations R. The complete statistical data for the considered problems are reported in Tables 3, 4 and 5.

with a high number of total Grover rotations are those mainly concentrated towards the bottom of the graph.

3.2 Max-cut problem

This subsection reports the results obtained with different sizes of max-cut problems. From Table 3 and Figures 11a and 11b, it is possible to notice that for the small size of max-cut problems, the **random saturated** strategy for varying the number of Grover rotations and the **linear** strategy for dynamically reducing the threshold for the stop condition seem to be the best compromise between a high success probability and a low number of both average Grover

Table 3: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a two-node max-cut ($w_{i,j}$ = rand(range(0, 2))) and a three-node max-cut ($w_{i,j}$ = rand(range(0, 2))) .

Adopted strategy		**Max-cut with 2 variables**			**Max-cut with 3 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.98	1.68	5.69	0.96	4.67	7.92
RND sat	**ADPT**	0.94	1.00	4.51	0.94	2.16	5.86
	LIN	**0.96**	**0.88**	**4.28**	**0.97**	**2.05**	**5.68**
	LOG	0.93	1.06	4.76	0.92	2.96	6.47
RND noSat	**FIX**	**0.97**	**3.12**	**5.65**	**0.96**	**5.17**	**7.62**
	ADPT	0.93	1.30	4.41	0.92	2.44	5.72
	LIN	0.93	1.20	4.31	0.92	2.07	5.52
	LOG	0.88	1.80	4.69	0.90	3.45	6.57
LIN sat	**FIX**	0.97	5.39	5.78	1.00	9.81	7.6
	ADPT	0.90	2.70	4.42	1.00	4.51	5.76
	LIN	0.94	2.40	4.29	**1.00**	**4.32**	**5.69**
	LOG	**0.97**	**3.29**	**4.7**	1.00	6.89	6.82
LIN noSat	**FIX**	0.98	6.31	5.69	1.00	10.51	7.35
	ADPT	**0.97**	**2.71**	**4.42**	**1.00**	**4.22**	**5.69**
	LIN	0.92	2.60	4.27	1.00	4.50	5.73
	LOG	0.96	3.36	4.75	1.00	6.74	6.56
PAT	**FIX**	**0.99**	**1.22**	**5.68**	1.00	3.02	8.27
	ADPT	0.93	0.16	4.29	0.93	1.46	6.48
	LIN	0.87	0.05	4.19	**0.98**	**1.19**	**6.05**
	LOG	0.92	0.23	4.59	0.99	2.23	7.29

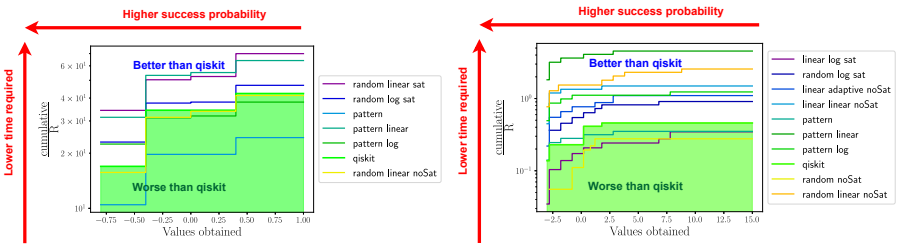
rotations (R) and classical iterations (C).

The same best strategies combination can be identified for the medium size of problem, as shown in Table 4 and in Figures 11c and 11d.

For larger problem dimensions, this strategies combination is overcome by **random saturated with adaptive** variation of the threshold and **linear not saturated with linear** variation of the threshold (as shown in Table 5 and in Figures 11e and 11f), even if it continues to guarantee a good compromise between probability of success and time. The potential of linear growth techniques for the Grover rotations number and the adaptive strategy for the variation of the threshold value is, therefore, better expressed for more significant problems.

Table 4: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a four-node max-cut ($w_{i,j} = \text{rand}(\text{range}(0, 2))$) and ten times a five-node max-cut ($w_{i,j} = \text{rand}(\text{range}(0, 2))$).

Adopted strategy		**Max-cut with 4 variables**			**Max-cut with 5 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.97	6.92	9.04	1.00	10.40	10.00
RND sat	**ADPT**	0.87	3.09	6.49	**1.00**	**3.60**	**7.80**
	LIN	**0.97**	**2.82**	**6.34**	1.00	4.90	7.60
	LOG	0.94	5.62	7.86	1.00	6.80	7.80
RND noSat	**FIX**	0.98	8.97	9.19	1.00	12.30	10.50
	ADPT	0.93	2.75	6.45	0.80	4.30	6.90
	LIN	**0.97**	**2.88**	**6.41**	**0.90**	**2.30**	**6.70**
	LOG	0.98	6.18	8.26	1.00	8.90	9.40
LIN sat	**FIX**	1.00	15.20	8.97	1.00	23.40	9.90
	ADPT	**1.00**	**4.64**	**5.98**	1.00	7.90	7.80
	LIN	0.98	4.73	6.07	**1.00**	**5.80**	**7.60**
	LOG	1.00	11.32	7.95	1.00	14.50	7.80
LIN noSat	**FIX**	1.00	16.14	8.09	1.00	21.90	9.70
	ADPT	**0.99**	**4.51**	**5.98**	**1.00**	**6.90**	**7.00**
	LIN	0.99	4.96	6.02	1.00	8.40	6.80
	LOG	1.00	11.28	8.04	1.00	15.40	8.30
PAT	**FIX**	**0.99**	**5.10**	**9.93**	1.00	5.10	10.50
	ADPT	0.87	2.05	6.99	**1.00**	**3.30**	**8.10**
	LIN	0.91	2.16	7.03	0.90	2.20	7.30
	LOG	0.97	4.01	8.99	1.00	4.30	9.60



(a) Two-variable knapsack (b) Five-variable knapsack

Fig. 12: Cumulative distributions of the values obtained at the end of GAS, divided by the average total number of Grover rotations R. The complete statistical data for the considered problems are reported in Table 6.

Table 5: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, ten times a six-node max-cut ($w_{i,j}$ = rand(range(0, 2))) and a seven-node max-cut ($w_{i,j}$ = rand(range(0, 2))).

Adopted strategy		**Max-cut with 6 variables**			**Max-cut with 7 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.90	12.80	13.20	1.00	29.20	18.30
RND sat	**ADPT**	**0.90**	**4.40**	**7.90**	**0.80**	**7.30**	**11.30**
	LIN	0.80	3.00	7.40	0.70	4.90	9.00
	LOG	1.00	7.20	10.10	1.00	14.60	14.60
RND noSat	**FIX**	1.00	14.50	12.90	1.00	25.70	17.10
	ADPT	0.50	3.40	8.10	**0.80**	**8.60**	**11.10**
	LIN	**1.00**	**4.50**	**8.00**	0.70	4.60	9.00
	LOG	1.00	10.70	11.10	1.00	17.00	13.70
LIN sat	**FIX**	1.00	24.00	12.40	1.00	49.90	15.90
	ADPT	**1.00**	**6.00**	**7.50**	**1.00**	**13.10**	**10.70**
	LIN	0.80	5.80	7.30	0.80	9.10	8.10
	LOG	1.00	14.50	9.90	1.00	30.80	13.40
LIN noSat	**FIX**	1.00	23.10	11.70	1.00	48.4	16.00
	ADPT	0.90	7.30	8.00	1.00	13.6	10.10
	LIN	**0.90**	**5.70**	**6.70**	**0.90**	**9.50**	**8.60**
	LOG	1.00	15.10	9.60	1.00	32.80	13.30
PAT	**FIX**	1.00	9.40	13.40	1.00	23.10	18.10
	ADPT	0.80	2.40	8.00	0.70	8.50	11.90
	LIN	0.80	2.00	7.20	0.70	3.30	8.90
	LOG	**0.90**	**5.70**	**11.00**	**0.80**	**12.30**	**14.00**

3.3 Knapsack problem

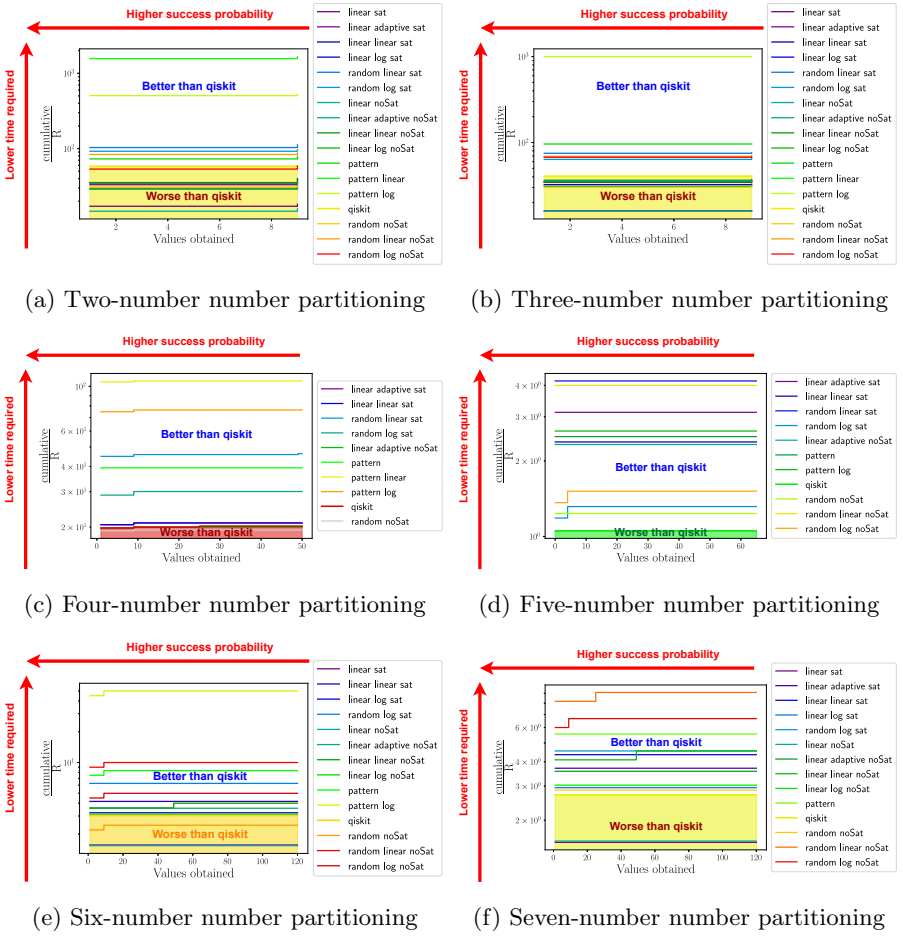
This subsection reports the results obtained with knapsack problems of different sizes. From Table 6 and Figures 12a and 12b, it is possible to notice that for the two-variable problem, the **random saturated** strategy for varying the number of Grover rotations and the **adaptive** strategy for dynamically reducing the threshold for the stop condition seem to be the best compromise between a high success probability and a low number of both average Grover rotations (R) and classical iterations (C). Instead, for the five-variable and seven-variable problems reported in the article, the best combination is **random not saturated with adaptive** variation of the threshold.
These results show the effectiveness of the adaptive strategy for varying the threshold.

Table 6: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a two-object knapsack (two-variable, considering those auxiliary, p_i = rand(range(1,3)), w_i = rand(range(1,3)) and W = rand(range(1,3))) problems and ten times a three-object knapsack (five-variable, considering those auxiliary, p_i = rand(range(1,3)), w_i = rand(range(1,3)) and W = rand(range(3,5))) problems.

Adopted strategy		**Knapsack with 2 variables**			**Knapsack with 5 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.40	2.36	8.27	0.20	21.80	20.00
RND sat	**ADPT**	**0.53**	**1.62**	**6.25**	**0.40**	**6.40**	**9.20**
	LIN	0.49	1.43	6.1	0.10	2.50	7.40
	LOG	0.49	2.13	7.61	0.40	11.00	14.00
RND noSat	**FIX**	0.25	5.85	9,97	0.20	36.10	27.6
	ADPT	**0.45**	**3.24**	**7.27**	**0.40**	**4.10**	**8.50**
	LIN	0.38	2.42	6.43	0.20	3.90	7.90
	LOG	0.25	4.44	8.95	0.00	16.90	17.20
LIN sat	**FIX**	0.20	8.75	5.85	0.00	64.70	27.20
	ADPT	0.32	4.64	3.24	0.10	6.70	8.10
	LIN	**0.30**	**4.33**	**2.42**	**0.10**	**6.30**	**8.00**
	LOG	0.20	8.07	4.44	0.20	28.90	16.90
LIN noSat	**FIX**	0.29	10.91	10.97	0.10	57.10	26.40
	ADPT	0.27	5.63	7.91	0.30	9.10	9.50
	LIN	**0.25**	**4.42**	**7.09**	**0.50**	**6.70**	**7.70**
	LOG	0.23	9.14	9.71	0.10	31.10	18.70
PAT	**FIX**	0.43	4.11	8.89	0.30	28.50	17.50
	ADPT	0.47	1.55	6.42	0.20	6.30	10.10
	LIN	**0.49**	**1.56**	**6.39**	**0.30**	**2.20**	**7.40**
	LOG	0,59	2.63	7.71	0.30	8.10	12.50

3.4 Number partitioning problem

This subsection reports the results obtained with number partitioning problems of different sizes. From Table 7 and Figures 13a and 13b, it is possible to notice that for a small size of number partitioning problems the **fixed pattern** strategy for varying the number of Grover rotations and the **linear** strategy for dynamically reducing the threshold for the stop condition seem to be the best compromise between a high success probability and a low number of both average Grover rotations (R) and classical iterations (C).
The same best strategies combination can be identified for the four-number



(a) Two-number number partitioning

(b) Three-number number partitioning

(c) Four-number number partitioning

(d) Five-number number partitioning

(e) Six-number number partitioning

(f) Seven-number number partitioning

Fig. 13: Cumulative distributions of the values obtained at the end of GAS, divided by the average total number of Grover rotations R. The complete statistical data for the considered problems are reported in Tables 7 and 8.

problem, as shown in Table 8 and in Figure 13c, while for five-number problem, as shown in Table 8 and in Figure 13d, the best strategy combination is the **random not saturated with linear** variation of the threshold.
For larger problem dimension, the best strategies combination is the same of the five-variable one (as shown in Table 12 and in Figures 13e and 13f). The results obtained with large-size problems show the limits of the **fixed pattern strategy** for varying the number of Grover rotations with this class of problems.

Table 7: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred time a two-node number partitioning ($S_i = \text{rand}(\text{range}(0, 5))$) and a three-node number partitioning ($S_i = \text{rand}(\text{range}(0, 5))$).

Adopted strategy		**Number partitioning with 2 variables**			**Number partitioning with 3 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.98	1.68	5.88	0.99	2.44	5.24
RND sat	**ADPT**	0.92	1.01	4.52	0.99	1.34	4.17
	LIN	0.92	0.89	4.23	**0.99**	**1.32**	**4.03**
	LOG	**0.96**	**1.04**	**4.73**	0.98	1.54	4.29
RND noSat	**FIX**	0.98	3.17	5.70	0.98	3.25	5.35
	ADPT	**0.93**	**1.44**	**4.47**	**0.98**	**1.41**	**4.08**
	LIN	0.92	1.10	4.27	0.98	1.42	4.09
	LOG	0.91	1.70	4.68	0.99	1.47	4.35
LIN sat	**FIX**	0.94	5.47	5.83	1.00	6.24	5.45
	ADPT	**0.96**	**2.89**	**4.63**	0.98	2.84	4.18
	LIN	0.88	2.51	4.20	**0.98**	**2.81**	**4.14**
	LOG	0.97	3.31	4.63	0.99	3.06	4.25
LIN noSat	**FIX**	0.93	6.30	5.67	1.00	6.30	5.53
	ADPT	0.87	2.91	4.44	**0.99**	**2.84**	**4.18**
	LIN	0.90	2.54	4.25	0.98	2.69	4.18
	LOG	**0.95**	**3.23**	**4.63**	0.97	3.15	4.42
PAT	**FIX**	0.92	1.26	5.78	1.00	1.04	5.22
	ADPT	0.91	0.25	4.42	0.98	0.05	4.10
	LIN	**0.94**	**0.06**	**4.28**	**1.00**	**0.00**	**4.04**
	LOG	0.96	0.19	4.60	1.00	0.10	4.38

3.5 Minimum vertex cover problem

In this subsection, the results obtained with minimum vertex cover problems of different sizes are reported. From Table 10 and Figures 14a and 14b, it is possible to notice that for a small size of minimum vertex cover problems the **linear not saturated** strategy for varying the number of Grover rotations and the **linear** strategy for dynamically reducing the threshold for the stop condition seem to be the best compromise between a high success probability and a low number of both average Grover rotations (R) and classical iterations (C).

The same best strategies combination can be identified for the four-variable problem, as shown in Table 11 and in Figure 14e, while for the five-variable one the best combination is **linear saturated with linear** variation of the threshold, as shown in Table 11 and in Figure 14f.

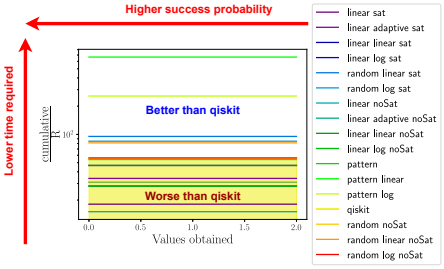
Table 8: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a four-numbers number partitioning problem (S_i = rand(range(0, 5))) and ten times a five-numbers number partitioning problem (S_i = rand(range(0, 5))).

Adopted strategy		**Number partitioning with 4 variables**			**Number Partitioning with 5 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.98	4.95	7.26	1.00	9.50	9.70
RND sat	**ADPT**	0.91	2.57	5.67	0.90	1.80	6.60
	LIN	**0.97**	**2.16**	**5.37**	**1.00**	**2.40**	**6.20**
	LOG	0.96	3.33	6.19	0.90	7.60	8.00
RND noSat	**FIX**	0.99	5.17	7.37	1.00	8.10	9.40
	ADPT	**0.95**	**2.49**	**5.66**	0.80	2.20	5.70
	LIN	0.92	2.45	5.54	**1.00**	**2.50**	**5.90**
	LOG	0.95	3.30	6.17	0.90	6.60	8.70
LIN sat	**FIX**	0.99	10.84	7.4	1.00	15.40	7.80
	ADPT	0.98	4.77	5.54	**1.00**	**3.20**	**6.00**
	LIN	**0.98**	**4.77**	**5.44**	1.00	4.20	6.00
	LOG	0.99	6.72	6.26	1.00	10.60	7.40
LIN noSat	**FIX**	0.99	10.78	7.29	1.00	15.60	8.90
	ADPT	**0.99**	**5.01**	**5.83**	**1.00**	**4.30**	**6.10**
	LIN	0.96	4.64	5.42	0.90	4.20	6.60
	LOG	1.00	6.62	6.29	1.00	10.80	7.60
PAT	**FIX**	1.00	2.54	7.58	**1.00**	**4.00**	**9.50**
	ADPT	0.96	1.05	5.83	1.00	2.10	6.90
	LIN	**0.99**	**0.94**	**5.59**	0.80	0.90	6.00
	LOG	0.98	1.31	6.01	1.00	3.80	9.50

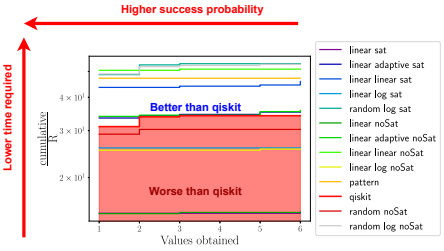
For larger problem dimension, these strategies combinations are overcome by **linear not saturated with adaptive** variation of the threshold (as shown in Table 5 and in Figures 11e and 11f). The potential of linear growth techniques for the Grover rotation number and the adaptive strategy for the variation of the threshold value is, therefore, better expressed for larger problems. However, for this type of problem, choosing the best strategies combination seems less easy than in the other cases because there are more variations in the results of different sizes.

Table 9: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, ten times a six-number number partitioning (S_i = rand(range(0, 5))) and a seven-number number partitioning (S_i = rand(range(0, 5))).

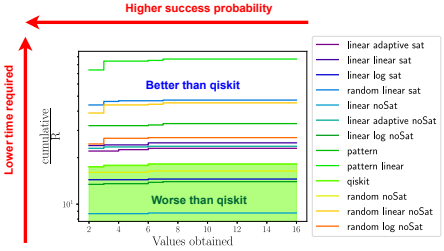
Adopted strategy		**Number partitioning with 6 variables**			**Number partitioning with 7 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	1.00	3.30	5.90	1.00	3.70	6.00
RND sat	**ADPT**	0.80	1.30	5.30	**1.00**	**1.20**	**4.80**
	LIN	0.80	1.20	4.60	0.80	0.90	4.20
	LOG	**1.00**	**1.60**	**4.30**	1.00	2.20	5.00
RND noSat	**FIX**	0.90	4.10	6.90	1.00	3.50	6.30
	ADPT	0.90	1.20	4.90	0.90	2.00	5.00
	LIN	**0.90**	**1.00**	**4.40**	**0.90**	**1.10**	**4.70**
	LOG	0.90	2.00	5.10	0.90	1.50	4.80
LIN sat	**FIX**	1.00	6.40	6.30	1.00	6.50	6.70
	ADPT	0.80	2.70	4.60	1.00	2.70	4.50
	LIN	**1.00**	**2.40**	**5.20**	**1.00**	**2.30**	**4.40**
	LOG	1.00	3.10	4.50	1.00	3.40	5.30
LIN noSat	**FIX**	1.00	6.50	6.30	1.00	6.40	6.10
	ADPT	**1.00**	**2.80**	**4.50**	**1.00**	**2.80**	**4.60**
	LIN	0.90	2.50	5.20	0.90	2.20	4.70
	LOG	1.00	3.20	4.90	1.00	3.30	5.00
PAT	**FIX**	0.90	1.20	6.50	**1.00**	**1.80**	**7.20**
	ADPT	0.70	0.10	4.20	0.60	0.00	4.00
	LIN	0.80	0.10	4.70	0.70	0.00	4.20
	LOG	**0.90**	**0.20**	**5.10**	0.80	0.40	5.10



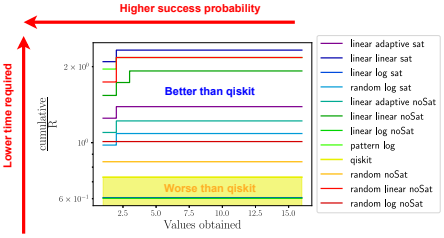
(a) Two-node minimum vertex cover



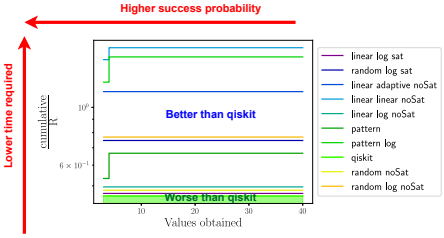
(b) Three-node minimum vertex cover



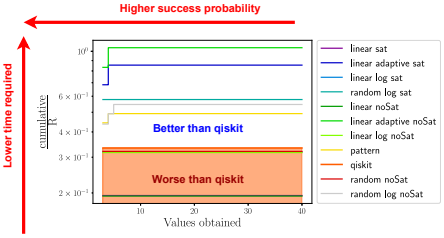
(c) Four-nodes minimum vertex cover



(d) Five-variable minimum vertex cover



(e) Six-nodes minimum vertex cover



(f) Seven-variable minimum vertex cover

Fig. 14: Cumulative distributions of the values obtained at the end of GAS, divided by the average total number of Grover rotations R. The complete statistical data for the considered problems are reported in Tables 10, 11 and 12.

Table 10: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a two-node minimum vertex cover and a three-node minimum vertex cover.

Adopted strategy		**Minimum vertex cover with 2 variables**			**Minimum vertex cover with 4 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	1.00	1.80	6.38	0.91	2.93	6.65
RND sat	**ADPT**	1.00	1.11	5.12	0.81	1.50	4.86
	LIN	**1.00**	**1.05**	**4.71**	0.85	1.18	4.64
	LOG	1.00	1.18	5.28	**0.91**	**1.87**	**5.51**
RND noSat	**FIX**	1.00	3.20	6.39	0.96	3.30	6.84
	ADPT	1.00	1.30	4.99	0.80	1.35	4.75
	LIN	**1.00**	**1.23**	**4.59**	0.83	1.16	4.68
	LOG	1.00	1.78	5.35	**0.92**	**1.88**	**5.62**
LIN sat	**FIX**	1.00	5.53	6.35	0.99	6.74	6.64
	ADPT	1.00	2.94	4.99	0.95	2.84	5.02
	LIN	**1.00**	**2.14**	**4.76**	0.95	2.18	4.86
	LOG	1.00	3.55	5.40	**1.00**	**3.87**	**5.79**
LIN noSat	**FIX**	1.00	6.65	6.48	0.99	6.76	6.51
	ADPT	1.00	3.24	5.24	0.95	2.80	5.05
	LIN	**1.00**	**2.14**	**4.81**	**0.99**	**1.96**	**4.76**
	LOG	1.00	3.56	5.37	0.99	3.91	5.78
PAT	**FIX**	1.00	1.85	6.87	**1.00**	**2.12**	**7.33**
	ADPT	1.00	0.28	4.53	0.79	0.39	4.73
	LIN	**1.00**	**0.15**	**4.55**	0.71	0.05	4.46
	LOG	1.00	0.39	5.06	0.81	0.42	5,29

Table 11: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a four-node minimum vertex cover and ten times a five-node minimum vertex cover.

Adopted strategy		**Minimum vertex cover with 4 variables**			**Minimum vertex Cover with 5 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.96	5.50	8.20	1.00	13.70	15.00
RND sat	**ADPT**	0.87	2.41	6.19	0.60	5.70	9.70
	LIN	**0.93**	**2.12**	**6.03**	0.20	3.10	7.30
	LOG	0.92	3.99	7.24	**0.90**	**9.20**	**11.00**
RND noSat	**FIX**	0.98	6.10	8.52	1.00	11.90	13.50
	ADPT	0.89	2.05	5.89	0.80	6.60	9.90
	LIN	0.86	2.21	5.93	**0.80**	**4.60**	**8.00**
	LOG	**0.91**	**3.71**	**7.11**	1.00	9.90	13.50
LIN sat	**FIX**	0.98	11.44	8.18	1.00	25.00	12.40
	ADPT	0.96	4.35	5.93	0.90	7.20	8.90
	LIN	0.96	4.01	6.04	**0.90**	**4.30**	**7.30**
	LOG	**0.99**	**6.88**	**6.95**	1.00	16.50	10.80
LIN noSat	**FIX**	0.99	11.41	8.46	1.00	25.10	13.10
	ADPT	0.97	4.22	5.95	**0.90**	**8.20**	**9.10**
	LIN	**0.97**	**3.92**	**5.95**	0.80	5.20	7.00
	LOG	0.96	7.15	7.09	1.00	16.60	11.40
PAT	**FIX**	**0.97**	**3.01**	**8.36**	0.70	8.00	12.40
	ADPT	0.85	1.27	6.21	0.40	3.30	9.30
	LIN	0.85	1.15	6.10	0.50	2.60	7.70
	LOG	0.91	2.32	7.50	**0.90**	**4.60**	**10.40**

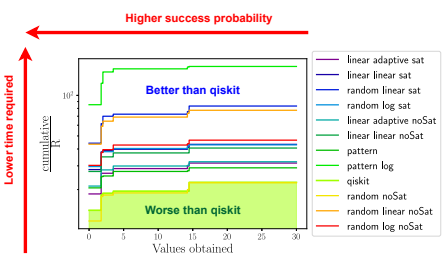
Table 12: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, ten times a six-node minimum vertex cover and a seven-node minimum vertex cover.

Adopted strategy		**Minimum vertex cover with 6 variables**			**Minimum vertex cover with 7 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	1.00	21.90	15.50	1.00	30.00	17.30
RND sat	**ADPT**	0.70	4.70	9.00	0.70	9.20	11.60
	LIN	0.60	3.10	7.90	0.60	5.40	9.50
	LOG	**1.00**	**13.40**	**14.20**	**1.00**	**17.30**	**15.90**
RND noSat	**FIX**	1.00	20.80	18.30	1.00	31.20	21.10
	ADPT	**0.90**	**6.70**	**10.70**	**0.70**	**9.30**	**11.80**
	LIN	0.50	3.60	6.90	0.60	6.60	10.00
	LOG	1.00	13.00	15.90	0.80	18.30	14.50
LIN sat	**FIX**	1.00	31.00	13.30	1.00	51.5	17.40
	ADPT	0.70	7.50	8.70	**0.80**	**11.70**	**10.90**
	LIN	0.70	4.70	7.50	0.70	7.70	9.70
	LOG	**1.00**	**21.40**	**13.50**	1.00	31.40	15.40
LIN noSat	**FIX**	1.00	32.60	13.80	1.00	51.80	18.40
	ADPT	**1.00**	**8.70**	**8.70**	**0.80**	**9.60**	**10.80**
	LIN	0.90	5.90	7.80	0.70	7.70	9.50
	LOG	1.00	20.20	12.50	1.00	31.60	15.70
PAT	**FIX**	0.80	15.00	15.00	**0.90**	**20.30**	**17.70**
	ADPT	0.70	5.40	10.30	0.40	6.10	10.80
	LIN	0.30	1.40	6.50	0.30	3.10	9.00
	LOG	**0.80**	**6.40**	**11.90**	0.60	11.60	14.50

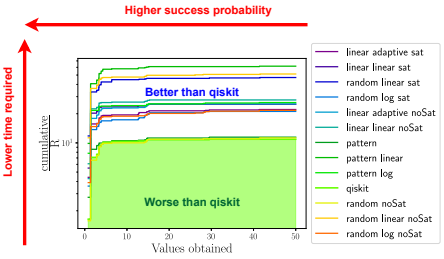
3.6 Cluster separation problem

This subsection reports the results obtained with cluster separation problems of different sizes. From Table 13 and Figures 15a and 15b, it is possible to notice that for four-variable and six-variable of cluster separation problems the **random not saturated** strategy for varying the number of Grover rotations and the **linear** strategy for dynamically reducing the threshold for the stop condition seem to be the best compromise between a high success probability and a low number of both average Grover rotations (R) and classical iterations (C).

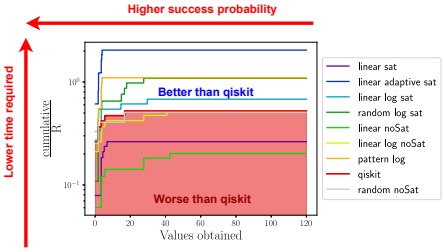
On the other hand, for what concerns the eight-variable problem, the best strategies combination is **linear not saturated with adaptive** variation of the threshold (as shown in Table 14 and in Figure 15c). The potential of linear growth techniques for the Grover rotation number and the adaptive strategy for the variation of the threshold value is, therefore, better expressed for larger problems. However, the available data are insufficient to determine the best strategies for this class of problem.



(a) Two-data clustering separation



(b) Three-data clustering separation



(c) Four-data clustering separation

Fig. 15: Cumulative distributions of the values obtained at the end of GAS, divided by the average total number of Grover rotations R. The complete statistical data for the considered problems are reported in Table 13.

Table 13: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a two-data cluster separation (4 variables) and a three-data cluster separation (6 variables).

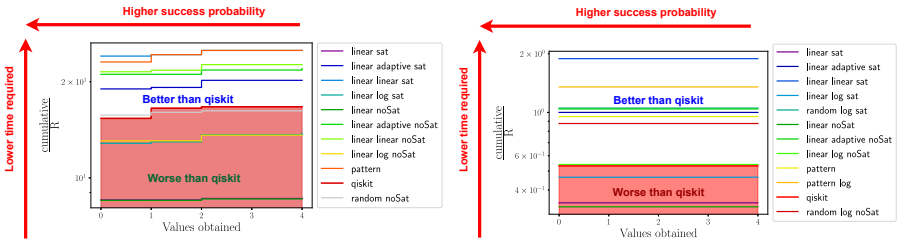
Adopted strategy		**Cluster separation with 4 variables**			**Cluster separation with 6 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.62	4.42	8.66	0.15	9.17	13.12
RND sat	**ADPT**	0.56	1.43	5.71	0.19	2.56	6.68
	LIN	0.53	1.20	5.02	**0.25**	**2.12**	**6.68**
	LOG	**0.69**	**2.33**	**6.68**	0.17	4.74	9.45
RND noSat	**FIX**	0.52	4.45	8.59	0.14	8.89	12.85
	ADPT	0.54	1.57	5.84	0.23	2.38	6.80
	LIN	**0.56**	**1.29**	**5.17**	**0.22**	**1.95**	**6.70**
	LOG	0.65	2.15	6.42	0.18	4.62	9.41
LIN sat	**FIX**	0.43	10.12	9.87	0.09	24.97	18.86
	ADPT	**0.59**	**3.18**	**5.90**	0.20	4.53	7.18
	LIN	0.65	2.31	5.52	**0.22**	**4.00**	**7.47**
	LOG	0.57	4.91	7.14	0.12	11.19	11.63
LIN noSat	**FIX**	0.38	9.83	9.47	0.07	24.59	18.47
	ADPT	0.66	3.11	5.89	0.17	3.97	6.79
	LIN	**0.67**	**2.46**	**5.59**	**0.22**	**3.61**	**7.32**
	LOG	0.59	5.19	7.61	0.11	11.77	12.3
PAT	**FIX**	**0.71**	**3.45**	**8.87**	0.24	8.73	12.2
	ADPT	0.42	0.38	4.75	0.18	1.86	7.10
	LIN	0.33	0.07	4.50	0.16	1.62	6.90
	LOG	0.52	0.61	5.65	**0.30**	**3.84**	**9.42**

Table 14: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, ten times a four-data cluster separation (8 variables).

Adopted strategy		**Cluster separation with 8 variables**		
r	**t**	**POS**	**R**	**C**
QSKT	–	0.50	19.10	25.70
RND sat	**ADPT**	**0.10**	**2.70**	**7.30**
	LIN	0.10	2.90	6.90
	LOG	0.10	9.20	13.60
RND noSat	**FIX**	0.50	19.70	24.00
	ADPT	0.20	2.70	8.50
	LIN	0.10	2.60	6.90
	LOG	**0.40**	**7.40**	**12.70**
LIN sat	**FIX**	0.30	38.00	27.50
	ADPT	**0.30**	**4.90**	**8.30**
	LIN	0.10	3.90	7.80
	LOG	0.40	14.70	13.10
LIN noSat	**FIX**	0.30	49.50	36.30
	ADPT	0.10	3.60	7.30
	LIN	0.00	3.30	6.90
	LOG	**0.40**	**19.00**	**16.30**
PAT	**FIX**	0.10	10.30	12.90
	ADPT	0.20	1.60	6.90
	LIN	0.10	1.80	6.90
	LOG	**0.30**	**9.10**	**11.40**

3.7 Graph colouring problem

This subsection reports the results obtained with graph colouring problems of different sizes. From Table 15 and Figures 16a and 16b, it is possible to notice that the **linear saturated** strategy for varying the number of Grover rotations and the **linear** strategy for dynamically reducing the threshold for the stop condition seem to be the best compromise between a high success probability and a low number of both average Grover rotations (R) and classical iterations (C) for both the considered problems.
Unfortunately, this type of problem scales too fast for performing tests with a larger or intermediate number of variables.



(a) Four-variables graph coloring (b) Six-variable graph coloring

Fig. 16: Cumulative distributions of the values obtained at the end of GAS, divided by the average total number of Grover rotations R. The complete statistical data for the considered problems are reported in Table 15.

Table 15: Statistical results concerning the effectiveness of each strategy combination. In particular, **probability of success** (**POS**), **average number of total Grover rotations** (**R**) and **average number of classical iterations** (**C**) in a GAS execution are reported. The reported data are obtained by solving, with each strategies combinations, one hundred times a two-node two-colour graph coloring (4 variables) and ten times a three-node two-colour graph coloring (6 variables).

Adopted strategy		**Graph coloring with 4 variables**			**Graph coloring with 6 variables**		
r	**t**	**POS**	**R**	**C**	**POS**	**R**	**C**
QSKT	–	0.92	5.99	8.72	1.00	18.80	13.40
RND sat	**ADPT**	0.85	2.98	6.54	**1.00**	**5.30**	**8.40**
	LIN	0.84	2.34	6.27	0.70	4.20	7.30
	LOG	**0.87**	**3.37**	**7.34**	1.00	9.50	10.70
RND noSat	**FIX**	**0.97**	**6.17**	**8.92**	0.90	15.50	12.60
	ADPT	0.87	2.56	6.44	**0.90**	**5.50**	**8.00**
	LIN	0.83	2.19	6.06	0.80	4.60	7.70
	LOG	0.88	4.04	7.55	1.00	11.40	10.40
LIN sat	**FIX**	0.99	11.60	8.45	1.00	29.10	11.20
	ADPT	0.94	4.95	6.29	1.00	10.00	9.00
	LIN	**0.96**	**3.99**	**6.11**	**1.00**	**5.30**	**7.60**
	LOG	0.93	7.25	7.11	1.00	21.50	9.80
LIN noSat	**FIX**	0.99	11.64	8.46	1.00	30.50	12.10
	ADPT	0.94	4.55	6.20	1.00	9.60	9.00
	LIN	**0.96**	**4.42**	**6.08**	**0.90**	**7.50**	**7.20**
	LOG	0.93	7.35	7.30	1.00	18.50	10.40
PAT	**FIX**	0.92	3.99	9.21	1.00	10.50	13.20
	ADPT	0.75	1.60	6.67	0.80	3.30	8.50
	LIN	0.79	1.15	6.03	**0.90**	**2.70**	**7.40**
	LOG	**0.89**	**2.47**	**7.84**	1.00	7.40	12.10

References

[1] F. Glover, G. Kochenberger, and Y. Du, *A tutorial on formulating and using QUBO models*, 2018. DOI: https://doi.org/10.48550/arXiv.1811.11538.

[2] J. T. Iosue, *qubovert Documentation*. 2019. [Online at https://qubovert.readthedocs.io/en/stable/,; accessed 10-May-2022]

[3] J. T. Iosue, *pyqubo*.[Online at https://github.com/recruit-communication,; accessed 10-May-2022]

[4] M. Zaman, K. Tanahashi, and S. Tanaka, *PyQUBO: Python Library for Mapping Combinatorial Optimization Problems to QUBO Form*, IEEE Transactions on Computers, vol. 71, no. 4, pp. 838–850, 2022, doi: 10.1109/TC.2021.3063618. DOI: https://doi.org/10.1109/TC.2021.3063618.

[5] *Python package Dimod*. [Online at https://github.com/dwavesystems/dimod,; accessed 10-May-2022]

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by Simulated Annealing*, Science, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671. DOI: https://doi.org/10.1126/science.220.4598.671.

[7] C. Cook, H. Zhao, T. Sato, M. Hiromoto, and S. X.-D. Tan, *GPU-based Ising computing for solving max-cut combinatorial optimization problems*, Integration, vol. 69, pp. 335–344, 2019. DOI: https://doi.org/10.1016/j.vlsi.2019.07.003.

[8] Y. Haribara, S. Utsunomiya, K. Kawarabayashi, and Y. Yamamoto, *A coherent Ising machine for MAX-CUT problems: Performance evaluation against semidefinite programming relaxation and simulated annealing*, arXiv preprint arXiv:1501.07030, 2015. DOI: https://doi.org/10.1007/978-4-431-55756-2_12.

[9] F. Liers, T. Nieberg, and G. Pardella, *Via Minimization in VLSI Chip Design Application of a Planar Max-Cut Algorithm*, 2011. [Online at http://e-archive.informatik.uni-koeln.de/630/,; accessed 10-May-2022]

[10] M. W. Coffey, *Adiabatic quantum computing solution of the knapsack problem*, arXiv, 2017. DOI: https://doi.org/10.48550/arxiv.1701.05584.

[11] A. Verma and M. Lewis, Variable Reduction For Quadratic Unconstrained Binary Optimization. arXiv, 2021. DOI: https://doi.org/10.48550/arxiv.2105.07032.

[12] C. D. Gonzalez Calaza, D. Willsch, and K. Michielsen, *Garden optimization problems for benchmarking quantum annealers*, Quantum Information Processing, vol. 20, no. 9, pp. 1–22, 2021. DOI: https://doi.org/10.1007/s11128-021-03226-6

[13] K. Ikeda, Y. Nakamura, and T. S. Humble, *Application of quantum annealing to nurse scheduling problem*, Scientific reports, vol. 9, no. 1, pp. 1–10, 2019. DOI: https://doi.org/10.48550/arXiv.1904.12139

[14] P. Miasnikof, S. Hong, and Y. Lawryshyn, *Graph Clustering Via QUBO and Digital Annealing*. arXiv, 2020. DOI: https://doi.org/10.48550/ARXIV.2003.03872

[15] *Intel Xeon Gold 6134 Processor - Product Specification*. [Online] https://ark.intel.com/content/www/us/en/ark/products/120493/intel-xeon-gold-6134-processor-24-75m-cache-3-20-ghz.html, accessed 25-October-2021.

[16] A. Gilliam, S. Woerner, and C. Gonciulea, *Grover Adaptive Search for Constrained Polynomial Binary Optimization*, Apr. 2021. DOI: https://doi.org/10.22331/q-2021-04-08-428 10.22331/q-2021-04-08-428.

[17] W. P. Baritompa, D. W. Bulger, and G. R. Wood, *Grover's Quantum Algorithm Applied to Global Optimization*, Jan. 2005. DOI: https://doi.org/10.1137/040605072.

[18] Y. Liu and G. J. Koehler, *Using modifications to Grover's Search algorithm for quantum global optimization*, European Journal of Operational Research, vol. 207, no. 2, pp. 620–632, 2010. DOI: https://doi.org/10.1016/j.ejor.2010.05.039