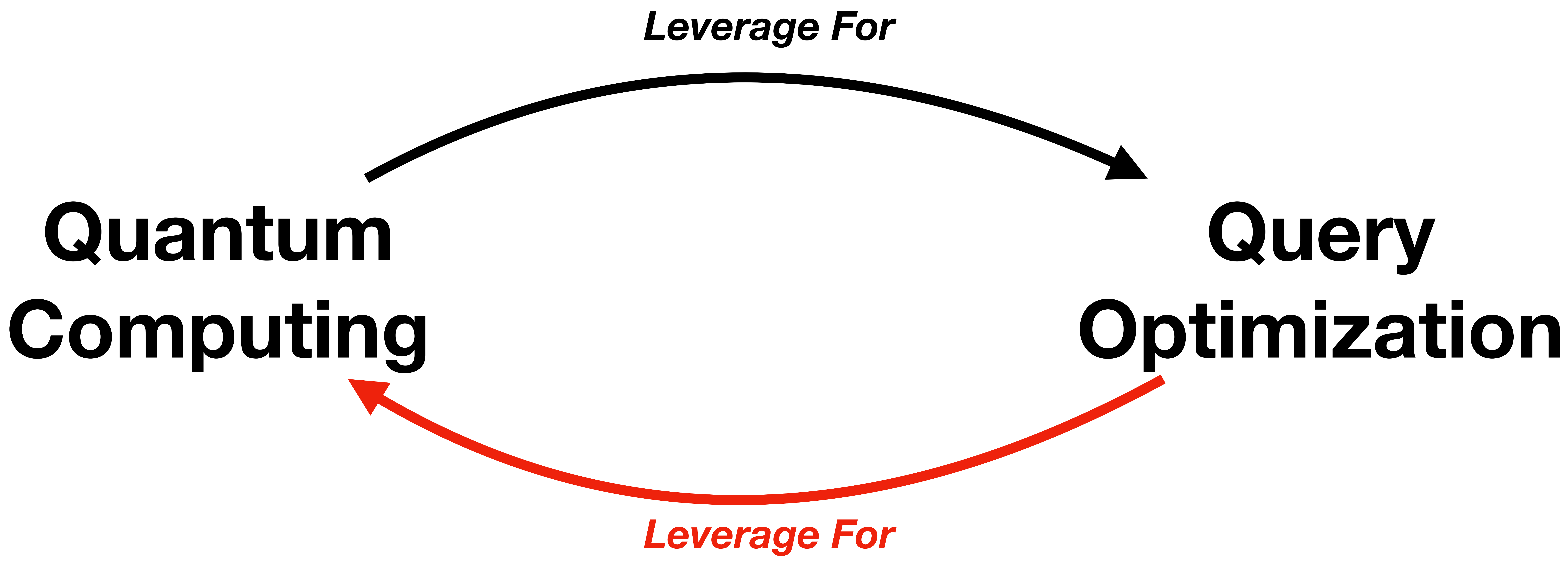
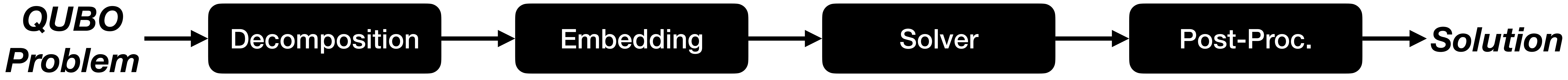
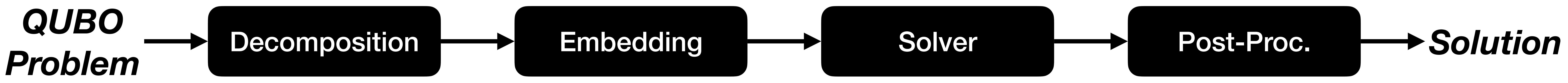
Cost-Based Query Optimization for Quantum Computation

Immanuel Trummer

$$\begin{tikzcd}
\textbf{Quantum Computing} \arrow[r, bend left, "\textit{Leverage For}"] & \textbf{Query Optimization}
\end{tikzcd}$$


Leverage For
Quantum Computing
Query Optimization
Leverage For

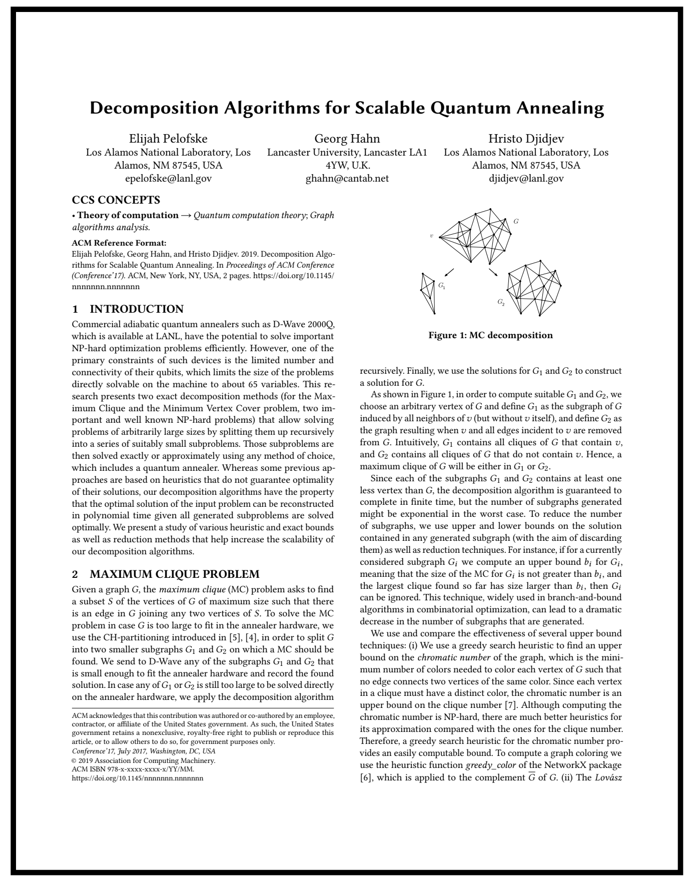

QUBO Problem
Decomposition
Embedding
Solver
Post-Proc.
Solution



Choice of Method

$$\begin{tikzcd}
\textit{QUBO Problem} \arrow[r] & \boxed{\text{Decomposition}} \arrow[r] & \boxed{\text{Embedding}} \arrow[r] & \boxed{\text{Solver}} \arrow[r] & \boxed{\text{Post-Proc.}} \arrow[r] & \textit{Solution}
\end{tikzcd}$$

Choice of Method



Decomposition Algorithms for Scalable Quantum Annealing

Elijah Pelofske
Los Alamos National Laboratory, Los Alamos, NM 87545, USA
epelofske@lanl.gov

Georg Hahn
Lancaster University, Lancaster LA1 4YW, U.K.
ghahn@cantab.net

Hristo Djidjev
Los Alamos National Laboratory, Los Alamos, NM 87545, USA
djidjev@lanl.gov

CCS CONCEPTS

• **Theory of computation** → *Quantum computation theory*; *Graph algorithms analysis.*

ACM Reference Format:
Elijah Pelofske, Georg Hahn, and Hristo Djidjev. 2019. Decomposition Algorithms for Scalable Quantum Annealing. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Commercial adiabatic quantum annealers such as D-Wave 2000Q, which is available at LANL, have the potential to solve important NP-hard optimization problems efficiently. However, one of the primary constraints of such devices is the limited number and connectivity of their qubits, which limits the size of the problems directly solvable on the machine to about 65 variables. This research presents two exact decomposition methods (for the Maximum Clique and the Minimum Vertex Cover problem, two important and well known NP-hard problems) that allow solving problems of arbitrarily large sizes by splitting them up recursively into a series of suitably small subproblems. Those subproblems are then solved exactly or approximately using any method of choice, which includes a quantum annealer. Whereas some previous approaches are based on heuristics that do not guarantee optimality of their solutions, our decomposition algorithms have the property that the optimal solution of the input problem can be reconstructed in polynomial time given all generated subproblems are solved optimally. We present a study of various heuristic and exact bounds as well as reduction methods that help increase the scalability of our decomposition algorithms.

2 MAXIMUM CLIQUE PROBLEM

Given a graph G, the *maximum clique* (MC) problem asks to find a subset S of the vertices of G of maximum size such that there is an edge in G joining any two vertices of S. To solve the MC problem in case G is too large to fit in the annealer hardware, we use the CH-partitioning introduced in [5], [4], in order to split G into two smaller subgraphs G_1 and G_2 on which a MC should be found. We send to D-Wave any of the subgraphs G_1 and G_2 that is small enough to fit the annealer hardware and record the found solution. In case any of G_1 or G_2 is still too large to be solved directly on the annealer hardware, we apply the decomposition algorithm recursively. Finally, we use the solutions for G_1 and G_2 to construct a solution for G.

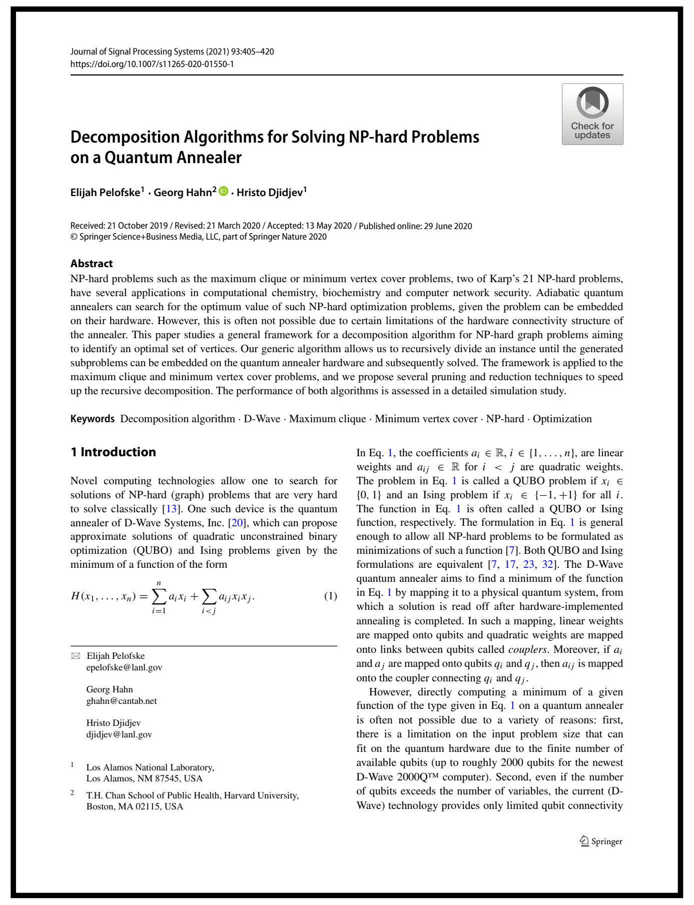
As shown in Figure 1, in order to compute suitable G_1 and G_2, we choose an arbitrary vertex of G and define G_1 as the subgraph of G induced by all neighbors of v (but without v itself), and define G_2 as the graph resulting when v and all edges incident to v are removed from G. Intuitively, G_1 contains all cliques of G that contain v, and G_2 contains all cliques of G that do not contain v. Hence, a maximum clique of G will be either in G_1 or G_2.

Since each of the subgraphs G_1 and G_2 contains at least one less vertex than G, the decomposition algorithm is guaranteed to complete in finite time, but the number of subgraphs generated might be exponential in the worst case. To reduce the number of subgraphs, we use upper and lower bounds on the solution contained in any generated subgraph (with the aim of discarding them) as well as reduction techniques. For instance, if for a currently considered subgraph G_i we compute an upper bound b_i for G_i, meaning that the size of the MC for G_i is not greater than b_i, and the largest clique found so far has size larger than b_i, then G_i can be ignored. This technique, widely used in branch-and-bound algorithms in combinatorial optimization, can lead to a dramatic decrease in the number of subgraphs that are generated.

We use and compare the effectiveness of several upper bound techniques: (i) We use a greedy search heuristic to find an upper bound on the *chromatic number* of the graph, which is the minimum number of colors needed to color each vertex of G such that no edge connects two vertices of the same color. Since each vertex in a clique must have a distinct color, the chromatic number is an upper bound on the clique number [7]. Although computing the chromatic number is NP-hard, there are much better heuristics for its approximation compared with the ones for the clique number. Therefore, a greedy search heuristic for the chromatic number provides an easily computable bound. To compute a graph coloring we use the heuristic function *greedy_color* of the NetworkX package [6], which is applied to the complement $\overline{G}$ of G. (ii) The *Lovász*

Figure 1: MC decomposition

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.
Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn



Journal of Signal Processing Systems (2021) 93:405–420
https://doi.org/10.1007/s11265-020-01550-1

Check for updates

Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer

Elijah Pelofske[1] · Georg Hahn[2] · Hristo Djidjev[1]

Received: 21 October 2019 / Revised: 21 March 2020 / Accepted: 13 May 2020 / Published online: 29 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
NP-hard problems such as the maximum clique or minimum vertex cover problems, two of Karp's 21 NP-hard problems, have several applications in computational chemistry, biochemistry and computer network security. Adiabatic quantum annealers can search for the optimum value of such NP-hard optimization problems, given the problem can be embedded on their hardware. However, this is often not possible due to certain limitations of the hardware connectivity structure of the annealer. This paper studies a general framework for a decomposition algorithm for NP-hard graph problems aiming to identify an optimal set of vertices. Our generic algorithm allows us to recursively divide an instance until the generated subproblems can be embedded on the quantum annealer hardware and subsequently solved. The framework is applied to the maximum clique and minimum vertex cover problems, and we propose several pruning and reduction techniques to speed up the recursive decomposition. The performance of both algorithms is assessed in a detailed simulation study.

Keywords Decomposition algorithm · D-Wave · Maximum clique · Minimum vertex cover · NP-hard · Optimization

1 Introduction

Novel computing technologies allow one to search for solutions of NP-hard (graph) problems that are very hard to solve classically [13]. One such device is the quantum annealer of D-Wave Systems, Inc. [20], which can propose approximate solutions of quadratic unconstrained binary optimization (QUBO) and Ising problems given by the minimum of a function of the form

$$H(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i + \sum_{i<j} a_{ij} x_i x_j. \quad (1)$$

In Eq. 1, the coefficients $a_i \in \mathbb{R}$, $i \in \{1, \ldots, n\}$, are linear weights and $a_{ij} \in \mathbb{R}$ for $i < j$ are quadratic weights. The problem in Eq. 1 is called a QUBO problem if $x_i \in \{0, 1\}$ and an Ising problem if $x_i \in \{-1, +1\}$ for all i. The function in Eq. 1 is often called a QUBO or Ising function, respectively. The formulation in Eq. 1 is general enough to allow all NP-hard problems to be formulated as minimizations of such a function [7]. Both QUBO and Ising formulations are equivalent [7, 17, 23, 32]. The D-Wave quantum annealer aims to find a minimum of the function in Eq. 1 by mapping it to a physical quantum system, from which a solution is read off after hardware-implemented annealing is completed. In such a mapping, linear weights are mapped onto qubits and quadratic weights are mapped onto links between qubits called *couplers*. Moreover, if a_i and a_j are mapped onto qubits q_i and q_j, then a_{ij} is mapped onto the coupler connecting q_i and q_j.

However, directly computing a minimum of a given function of the type given in Eq. 1 on a quantum annealer is often not possible due to a variety of reasons: first, there is a limitation on the input problem size that can fit on the quantum hardware due to the finite number of available qubits (up to roughly 2000 qubits for the newest D-Wave 2000Q™ computer). Second, even if the number of qubits exceeds the number of variables, the current (D-Wave) technology provides only limited qubit connectivity

✉ Elijah Pelofske
epelofske@lanl.gov

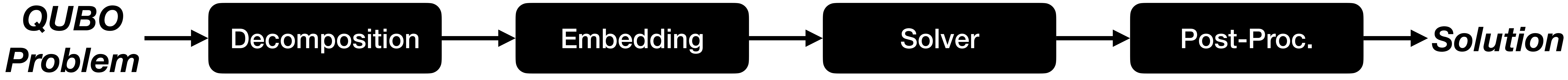
Georg Hahn
ghahn@cantab.net

Hristo Djidjev
djidjev@lanl.gov

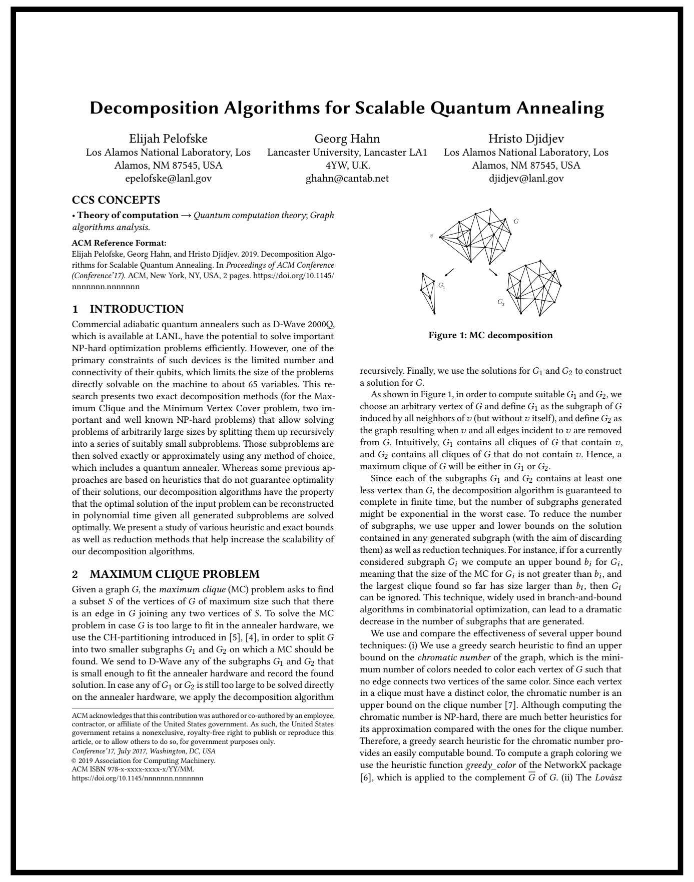
[1] Los Alamos National Laboratory, Los Alamos, NM 87545, USA
[2] T.H. Chan School of Public Health, Harvard University, Boston, MA 02115, USA

Springer

...



Choice of Method



Decomposition Algorithms for Scalable Quantum Annealing

Elijah Pelofske, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, epelofske@lanl.gov
Georg Hahn, Lancaster University, Lancaster LA1 4YW, U.K., ghahn@cantab.net
Hristo Djidjev, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, djidjev@lanl.gov

CCS CONCEPTS

• **Theory of computation** → *Quantum computation theory*; *Graph algorithms analysis.*

ACM Reference Format:
Elijah Pelofske, Georg Hahn, and Hristo Djidjev. 2019. Decomposition Algorithms for Scalable Quantum Annealing. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Commercial adiabatic quantum annealers such as D-Wave 2000Q, which is available at LANL, have the potential to solve important NP-hard optimization problems efficiently. However, one of the primary constraints of such devices is the limited number and connectivity of their qubits, which limits the size of the problems directly solvable on the machine to about 65 variables. This research presents two exact decomposition methods (for the Maximum Clique and the Minimum Vertex Cover problem, two important and well known NP-hard problems) that allow solving problems of arbitrarily large sizes by splitting them up recursively into a series of suitably small subproblems. Those subproblems are then solved exactly or approximately using any method of choice, which includes a quantum annealer. Whereas some previous approaches are based on heuristics that do not guarantee optimality of their solutions, our decomposition algorithms have the property that the optimal solution of the input problem can be reconstructed in polynomial time given all generated subproblems are solved optimally. We present a study of various heuristic and exact bounds as well as reduction methods that help increase the scalability of our decomposition algorithms.

2 MAXIMUM CLIQUE PROBLEM

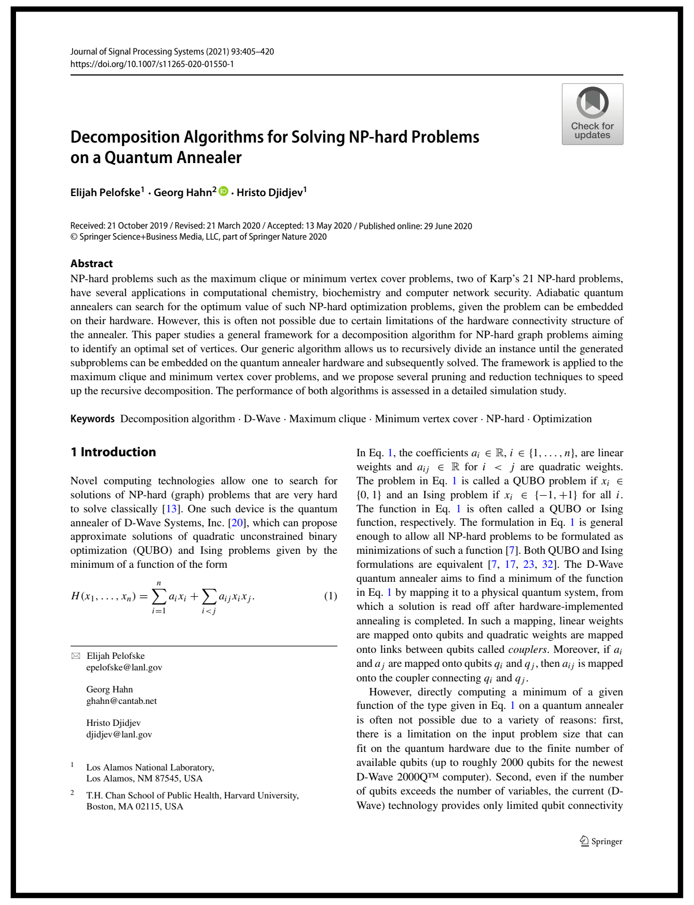
Given a graph G, the *maximum clique* (MC) problem asks to find a subset S of the vertices of G of maximum size such that there is an edge in G joining any two vertices of S. To solve the MC problem in case G is too large to fit in the annealer hardware, we use the CH-partitioning introduced in [5], [4], in order to split G into two smaller subgraphs G_1 and G_2 on which a MC should be found. We send to D-Wave any of the subgraphs G_1 and G_2 that is small enough to fit the annealer hardware and record the found solution. In case any of G_1 or G_2 is still too large to be solved directly on the annealer hardware, we apply the decomposition algorithm recursively. Finally, we use the solutions for G_1 and G_2 to construct a solution for G.

As shown in Figure 1, in order to compute suitable G_1 and G_2, we choose an arbitrary vertex of G and define G_1 as the subgraph of G induced by all neighbors of v (but without v itself), and define G_2 as the graph resulting when v and all edges incident to v are removed from G. Intuitively, G_1 contains all cliques of G that contain v, and G_2 contains all cliques of G that do not contain v. Hence, a maximum clique of G will be either in G_1 or G_2.

Since each of the subgraphs G_1 and G_2 contains at least one less vertex than G, the decomposition algorithm is guaranteed to complete in finite time, but the number of subgraphs generated might be exponential in the worst case. To reduce the number of subgraphs, we use upper and lower bounds on the solution contained in any generated subgraph (with the aim of discarding them) as well as reduction techniques. For instance, if for a currently considered subgraph G_i we compute an upper bound b_i for G_i, meaning that the size of the MC for G_i is not greater than b_i, and the largest clique found so far has size larger than b_i, then G_i can be ignored. This technique, widely used in branch-and-bound algorithms in combinatorial optimization, can lead to a dramatic decrease in the number of subgraphs that are generated.

We use and compare the effectiveness of several upper bound techniques: (i) We use a greedy search heuristic to find an upper bound on the *chromatic number* of the graph, which is the minimum number of colors needed to color each vertex of G such that no edge connects two vertices of the same color. Since each vertex in a clique must have a distinct color, the chromatic number is an upper bound on the clique number [7]. Although computing the chromatic number is NP-hard, there are much better heuristics for its approximation compared with the ones for the clique number. Therefore, a greedy search heuristic for the chromatic number provides an easily computable bound. To compute a graph coloring we use the heuristic function *greedy_color* of the NetworkX package [6], which is applied to the complement $\overline{G}$ of G. (ii) The *Lovász*

Figure 1: MC decomposition



Journal of Signal Processing Systems (2021) 93:405–420
https://doi.org/10.1007/s11265-020-01550-1

Check for updates

Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer

Elijah Pelofske[1] · Georg Hahn[2] · Hristo Djidjev[1]

Received: 21 October 2019 / Revised: 21 March 2020 / Accepted: 13 May 2020 / Published online: 29 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
NP-hard problems such as the maximum clique or minimum vertex cover problems, two of Karp's 21 NP-hard problems, have several applications in computational chemistry, biochemistry and computer network security. Adiabatic quantum annealers can search for the optimum value of such NP-hard optimization problems, given the problem can be embedded on their hardware. However, this is often not possible due to certain limitations of the hardware connectivity structure of the annealer. This paper studies a general framework for a decomposition algorithm for NP-hard graph problems aiming to identify an optimal set of vertices. Our generic algorithm allows us to recursively divide an instance until the generated subproblems can be embedded on the quantum annealer hardware and subsequently solved. The framework is applied to the maximum clique and minimum vertex cover problems, and we propose several pruning and reduction techniques to speed up the recursive decomposition. The performance of both algorithms is assessed in a detailed simulation study.

Keywords Decomposition algorithm · D-Wave · Maximum clique · Minimum vertex cover · NP-hard · Optimization

1 Introduction

Novel computing technologies allow one to search for solutions of NP-hard (graph) problems that are very hard to solve classically [13]. One such device is the quantum annealer of D-Wave Systems, Inc. [20], which can propose approximate solutions of quadratic unconstrained binary optimization (QUBO) and Ising problems given by the minimum of a function of the form

$$H(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i + \sum_{i<j} a_{ij} x_i x_j. \quad (1)$$

In Eq. 1, the coefficients $a_i \in \mathbb{R}$, $i \in \{1, \ldots, n\}$, are linear weights and $a_{ij} \in \mathbb{R}$ for $i < j$ are quadratic weights. The problem in Eq. 1 is called a QUBO problem if $x_i \in \{0, 1\}$ and an Ising problem if $x_i \in \{-1, +1\}$ for all i. The function in Eq. 1 is often called a QUBO or Ising function, respectively. The formulation in Eq. 1 is general enough to allow all NP-hard problems to be formulated as minimizations of such a function [7]. Both QUBO and Ising formulations are equivalent [7, 17, 23, 32]. The D-Wave quantum annealer aims to find a minimum of the function in Eq. 1 by mapping it to a physical quantum system, from which a solution is read off after hardware-implemented annealing is completed. In such a mapping, linear weights are mapped onto qubits and quadratic weights are mapped onto links between qubits called *couplers*. Moreover, if a_i and a_j are mapped onto qubits q_i and q_j, then a_{ij} is mapped onto the coupler connecting q_i and q_j.

However, directly computing a minimum of a given function of the type given in Eq. 1 on a quantum annealer is often not possible due to a variety of reasons: first, there is a limitation on the input problem size that can fit on the quantum hardware due to the finite number of available qubits (up to roughly 2000 qubits for the newest D-Wave 2000Q™ computer). Second, even if the number of qubits exceeds the number of variables, the current (D-Wave) technology provides only limited qubit connectivity

✉ Elijah Pelofske
epelofske@lanl.gov

Georg Hahn
ghahn@cantab.net

Hristo Djidjev
djidjev@lanl.gov

1 Los Alamos National Laboratory, Los Alamos, NM 87545, USA

2 T.H. Chan School of Public Health, Harvard University, Boston, MA 02115, USA

...

Integer programming techniques for minor-embedding in quantum annealers

David E. Bernal[1,4,5], Kyle E. C. Booth[2], Raouf Dridi[3], Hedayat Alghassi[3], Sridhar Tayur[3], and Davide Venturelli[4,5]

[1] Department of Chemical Engineering, Carnegie Mellon University, Pittsurgh PA 15213, USA
bernalde@cmu.edu
[2] Department of Mechanical & Industrial Engineering, University of Toronto, Toronto ON M5S 3G8, Canada
kbooth@mie.utoronto.ca
[3] Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA
{rdridi,halghass,stayur}@andrew.cmu.edu
[4] Quantum AI Lab (QuAIL), NASA Ames Research Center, Moffett Field CA 94035, USA
[5] USRA (RIACS), Mountain View CA 94043, USA
davide.venturelli@nasa.gov

Abstract. A major limitation of current generations of quantum annealers is the sparse connectivity of manufactured qubits in the hardware graph. This technological limitation has generated considerable interest, motivating efforts to design efficient and adroit minor-embedding procedures that bypass sparsity constraints. In this paper, starting from a previous equational formulation by Dridi et al. (arXiv:1810.01440), we propose integer programming (IP) techniques for solving the minor-embedding problem. The first approach involves a direct translation from the previous equational formulation to IP, while the second decomposes the problem into an assignment master problem and fiber condition checking subproblems. The proposed methods are able to detect instance infeasibility and provide bounds on solution quality, capabilities not offered by currently employed heuristic methods. We demonstrate the efficacy of our methods with an extensive computational assessment involving three different families of random graphs of varying sizes and densities. The direct translation as a monolithic IP model can be solved with existing commercial solvers yielding valid minor-embeddings, however, is outperformed overall by the decomposition approach. Our results demonstrate the promise of our methods for the studied benchmarks, highlighting the advantages of using IP technology for minor-embedding problems.

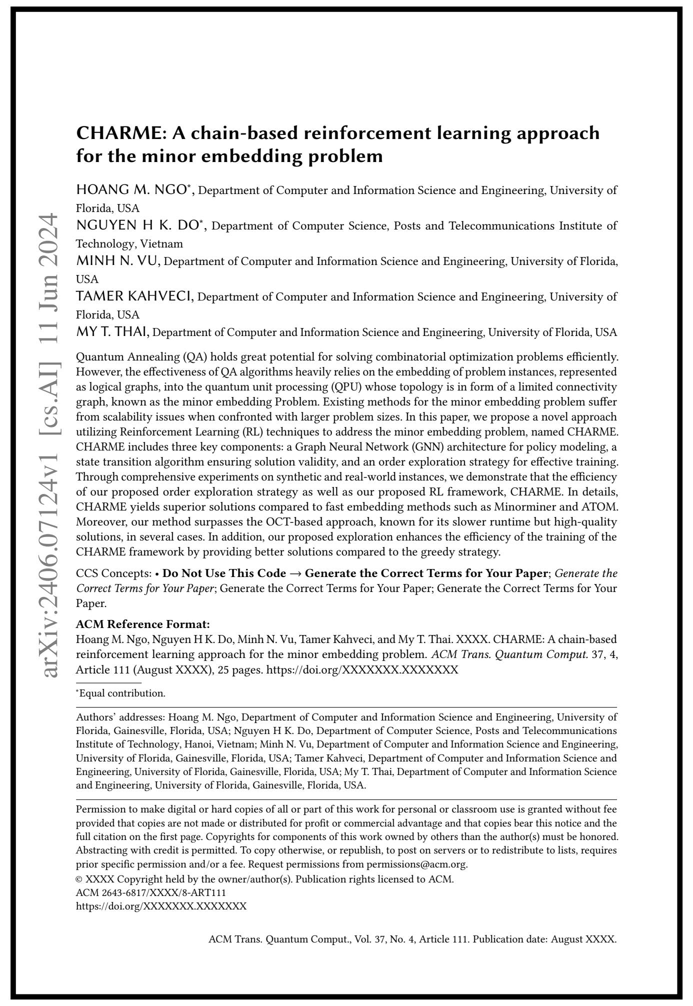
Keywords: Graph minors · Quantum annealers · Integer programming · Decomposition · Algebraic geometry

1 Introduction

Graph minor theory (GMT), the central theme of this work, is prominent across many fields. In quantum computing, GMT is employed to extend the scope of problems that can be represented on current quantum annealing hardware [5, 19]. Mapping a dense problem (logical) graph Y to a sparse (target) graph X can be achieved by constructing connected subgraphs of the target graph X from the high degree logical vertices y. The resulting mapping is called a minor-embedding of Y inside X.

Numerous heuristics for finding minor-embeddings have been proposed [1, 4, 29]. While these approaches are generally fast, they do not provide guarantees on the quality of the produced minor-embeddings nor can they prove the nonexistence of a minor-embedding for infeasible problems. An approach that attempts to address these shortcomings was recently introduced in Dridi et al. [9]. This approach uses tools from algebraic geometry and produces an equational formulation (as opposed to a purely combinatorial approach) to the minor-embedding problem.

In this paper, starting from this equational formulation, we propose integer programming (IP) techniques for tackling the embedding problem. Our proposed approaches differ from the computationally demanding Groebner bases computation used previously and are aimed at more efficiently computing embeddings while retaining the interesting properties that arise from the equational formulation of the problem. Our first approach, detailed in Section 3, directly translates the previous equational formulation to IP, while our second approach decomposes the problem into an assignment master problem and fiber condition checking



CHARME: A chain-based reinforcement learning approach for the minor embedding problem

HOANG M. NGO*, Department of Computer and Information Science and Engineering, University of Florida, USA
NGUYEN H K. DO*, Department of Computer Science, Posts and Telecommunications Institute of Technology, Vietnam
MINH N. VU, Department of Computer and Information Science and Engineering, University of Florida, USA
TAMER KAHVECI, Department of Computer and Information Science and Engineering, University of Florida, USA
MY T. THAI, Department of Computer and Information Science and Engineering, University of Florida, USA

Quantum Annealing (QA) holds great potential for solving combinatorial optimization problems efficiently. However, the effectiveness of QA algorithms heavily relies on the embedding of problem instances, represented as logical graphs, into the quantum unit processing (QPU) whose topology is in form of a limited connectivity graph, known as the minor embedding Problem. Existing methods for the minor embedding problem suffer from scalability issues when confronted with larger problem sizes. In this paper, we propose a novel approach utilizing Reinforcement Learning (RL) techniques to address the minor embedding problem, named CHARME. CHARME includes three key components: a Graph Neural Network (GNN) architecture for policy modeling, a state transition algorithm ensuring solution validity, and an order exploration strategy for effective training. Through comprehensive experiments on synthetic and real-world instances, we demonstrate that the efficiency of our proposed order exploration strategy as well as our proposed RL framework, CHARME. In details, CHARME yields superior solutions compared to fast embedding methods such as Minorminer and ATOM. Moreover, our method surpasses the OCT-based approach, known for its slower runtime but high-quality solutions, in several cases. In addition, our proposed exploration enhances the efficiency of the training of the CHARME framework by providing better solutions compared to the greedy strategy.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

ACM Reference Format:
Hoang M. Ngo, Nguyen H K. Do, Minh N. Vu, Tamer Kahveci, and My T. Thai. XXXX. CHARME: A chain-based reinforcement learning approach for the minor embedding problem. *ACM Trans. Quantum Comput.* 37, 4, Article 111 (August XXXX), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

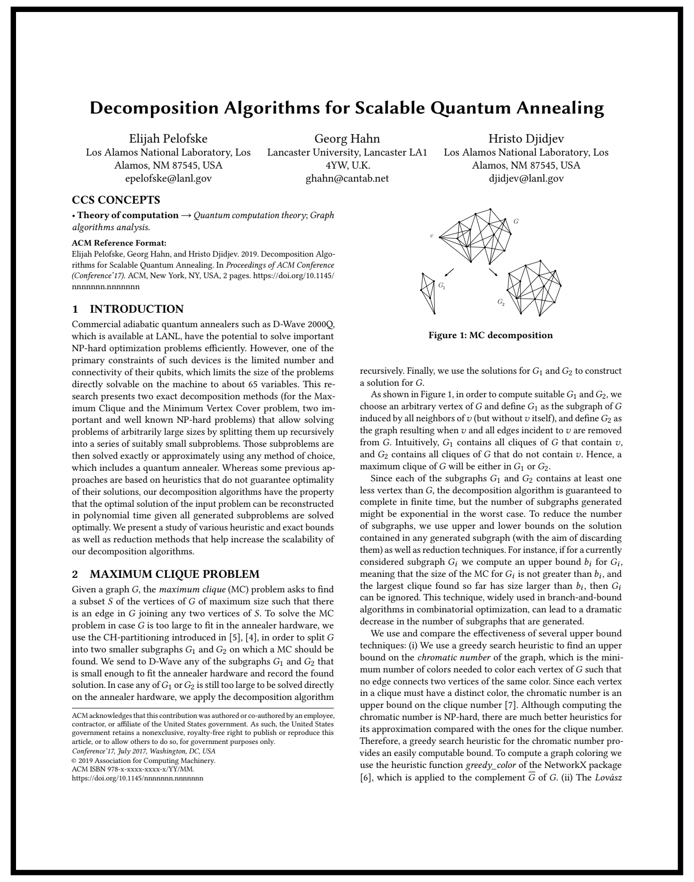
*Equal contribution.

Authors' addresses: Hoang M. Ngo, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; Nguyen H K. Do, Department of Computer Science, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam; Minh N. Vu, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; Tamer Kahveci, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; My T. Thai, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA.

...

$$\begin{tikzcd}
\textit{QUBO Problem} \arrow[r] & \text{Decomposition} \arrow[r] & \text{Embedding} \arrow[r] & \text{Solver} \arrow[r] & \text{Post-Proc.} \arrow[r] & \textit{Solution}
\end{tikzcd}$$

Choice of Method



Decomposition Algorithms for Scalable Quantum Annealing

Elijah Pelofske, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, epelofske@lanl.gov
Georg Hahn, Lancaster University, Lancaster LA1 4YW, U.K., ghahn@cantab.net
Hristo Djidjev, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, djidjev@lanl.gov

CCS CONCEPTS

• **Theory of computation** → *Quantum computation theory*; *Graph algorithms analysis.*

ACM Reference Format:
Elijah Pelofske, Georg Hahn, and Hristo Djidjev. 2019. Decomposition Algorithms for Scalable Quantum Annealing. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Commercial adiabatic quantum annealers such as D-Wave 2000Q, which is available at LANL, have the potential to solve important NP-hard optimization problems efficiently. However, one of the primary constraints of such devices is the limited number and connectivity of their qubits, which limits the size of the problems directly solvable on the machine to about 65 variables. This research presents two exact decomposition methods (for the Maximum Clique and the Minimum Vertex Cover problem, two important and well known NP-hard problems) that allow solving problems of arbitrarily large sizes by splitting them up recursively into a series of suitably small subproblems. Those subproblems are then solved exactly or approximately using any method of choice, which includes a quantum annealer. Whereas some previous approaches are based on heuristics that do not guarantee optimality of their solutions, our decomposition algorithms have the property that the optimal solution of the input problem can be reconstructed in polynomial time given all generated subproblems are solved optimally. We present a study of various heuristic and exact bounds as well as reduction methods that help increase the scalability of our decomposition algorithms.

2 MAXIMUM CLIQUE PROBLEM

Given a graph G, the *maximum clique* (MC) problem asks to find a subset S of the vertices of G of maximum size such that there is an edge in G joining any two vertices of S. To solve the MC problem in case G is too large to fit in the annealer hardware, we use the CH-partitioning introduced in [5], [4], in order to split G into two smaller subgraphs G_1 and G_2 on which a MC should be found. We send to D-Wave any of the subgraphs G_1 and G_2 that is small enough to fit the annealer hardware and record the found solution. In case any of G_1 or G_2 is still too large to be solved directly on the annealer hardware, we apply the decomposition algorithm

Figure 1: MC decomposition

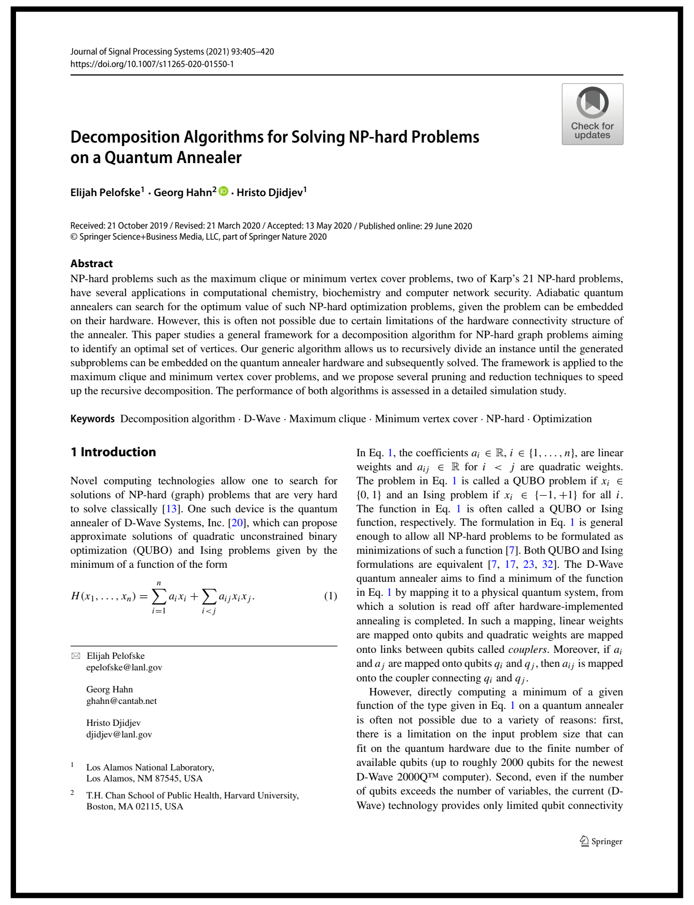
recursively. Finally, we use the solutions for G_1 and G_2 to construct a solution for G.

As shown in Figure 1, in order to compute suitable G_1 and G_2, we choose an arbitrary vertex of G and define G_1 as the subgraph of G induced by all neighbors of v (but without v itself), and define G_2 as the graph resulting when v and all edges incident to v are removed from G. Intuitively, G_1 contains all cliques of G that contain v, and G_2 contains all cliques of G that do not contain v. Hence, a maximum clique of G will be either in G_1 or G_2.

Since each of the subgraphs G_1 and G_2 contains at least one less vertex than G, the decomposition algorithm is guaranteed to complete in finite time, but the number of subgraphs generated might be exponential in the worst case. To reduce the number of subgraphs, we use upper and lower bounds on the solution contained in any generated subgraph (with the aim of discarding them) as well as reduction techniques. For instance, if for a currently considered subgraph G_i we compute an upper bound b_i for G_i, meaning that the size of the MC for G_i is not greater than b_i, and the largest clique found so far has size larger than b_i, then G_i can be ignored. This technique, widely used in branch-and-bound algorithms in combinatorial optimization, can lead to a dramatic decrease in the number of subgraphs that are generated.

We use and compare the effectiveness of several upper bound techniques: (i) We use a greedy search heuristic to find an upper bound on the *chromatic number* of the graph, which is the minimum number of colors needed to color each vertex of G such that no edge connects two vertices of the same color. Since each vertex in a clique must have a distinct color, the chromatic number is an upper bound on the clique number [7]. Although computing the chromatic number is NP-hard, there are much better heuristics for its approximation compared with the ones for the clique number. Therefore, a greedy search heuristic for the chromatic number provides an easily computable bound. To compute a graph coloring we use the heuristic function *greedy_color* of the NetworkX package [6], which is applied to the complement $\overline{G}$ of G. (ii) The *Lovász*

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.
Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn



Journal of Signal Processing Systems (2021) 93:405–420
https://doi.org/10.1007/s11265-020-01550-1

Check for updates

Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer

Elijah Pelofske[1] · Georg Hahn[2] · Hristo Djidjev[1]

Received: 21 October 2019 / Revised: 21 March 2020 / Accepted: 13 May 2020 / Published online: 29 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

NP-hard problems such as the maximum clique or minimum vertex cover problems, two of Karp's 21 NP-hard problems, have several applications in computational chemistry, biochemistry and computer network security. Adiabatic quantum annealers can search for the optimum value of such NP-hard optimization problems, given the problem can be embedded on their hardware. However, this is often not possible due to certain limitations of the hardware connectivity structure of the annealer. This paper studies a general framework for a decomposition algorithm for NP-hard graph problems aiming to identify an optimal set of vertices. Our generic algorithm allows us to recursively divide an instance until the generated subproblems can be embedded on the quantum annealer hardware and subsequently solved. The framework is applied to the maximum clique and minimum vertex cover problems, and we propose several pruning and reduction techniques to speed up the recursive decomposition. The performance of both algorithms is assessed in a detailed simulation study.

Keywords Decomposition algorithm · D-Wave · Maximum clique · Minimum vertex cover · NP-hard · Optimization

1 Introduction

Novel computing technologies allow one to search for solutions of NP-hard (graph) problems that are very hard to solve classically [13]. One such device is the quantum annealer of D-Wave Systems, Inc. [20], which can propose approximate solutions of quadratic unconstrained binary optimization (QUBO) and Ising problems given by the minimum of a function of the form

$$H(x_1, \dots, x_n) = \sum_{i=1}^{n} a_i x_i + \sum_{i<j} a_{ij} x_i x_j. \quad (1)$$

In Eq. 1, the coefficients $a_i \in \mathbb{R}$, $i \in \{1, \dots, n\}$, are linear weights and $a_{ij} \in \mathbb{R}$ for $i < j$ are quadratic weights. The problem in Eq. 1 is called a QUBO problem if $x_i \in \{0, 1\}$ and an Ising problem if $x_i \in \{-1, +1\}$ for all i. The function in Eq. 1 is often called a QUBO or Ising function, respectively. The formulation in Eq. 1 is general enough to allow all NP-hard problems to be formulated as minimizations of such a function [7]. Both QUBO and Ising formulations are equivalent [7, 17, 23, 32]. The D-Wave quantum annealer aims to find a minimum of the function in Eq. 1 by mapping it to a physical quantum system, from which a solution is read off after hardware-implemented annealing is completed. In such a mapping, linear weights are mapped onto qubits and quadratic weights are mapped onto links between qubits called *couplers*. Moreover, if a_i and a_j are mapped onto qubits q_i and q_j, then a_{ij} is mapped onto the coupler connecting q_i and q_j.

However, directly computing a minimum of a given function of the type given in Eq. 1 on a quantum annealer is often not possible due to a variety of reasons: first, there is a limitation on the input problem size that can fit on the quantum hardware due to the finite number of available qubits (up to roughly 2000 qubits for the newest D-Wave 2000Q™ computer). Second, even if the number of qubits exceeds the number of variables, the current (D-Wave) technology provides only limited qubit connectivity

✉ Elijah Pelofske
epelofske@lanl.gov

Georg Hahn
ghahn@cantab.net

Hristo Djidjev
djidjev@lanl.gov

1 Los Alamos National Laboratory, Los Alamos, NM 87545, USA
2 T.H. Chan School of Public Health, Harvard University, Boston, MA 02115, USA

Springer

arXiv:1912.08314v2 [quant-ph] 20 Dec 2019

Integer programming techniques for minor-embedding in quantum annealers

David E. Bernal[1,4,5], Kyle E. C. Booth[2], Raouf Dridi[3], Hedayat Alghassi[3], Sridhar Tayur[3], and Davide Venturelli[4,5]

[1] Department of Chemical Engineering, Carnegie Mellon University, Pittsurgh PA 15213, USA
bernalde@cmu.edu
[2] Department of Mechanical & Industrial Engineering, University of Toronto, Toronto ON M5S 3G8, Canada
kbooth@mie.utoronto.ca
[3] Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA
{rdridi,halghass,stayur}@andrew.cmu.edu
[4] Quantum AI Lab (QuAIL), NASA Ames Research Center, Moffett Field CA 94035, USA
[5] USRA (RIACS), Mountain View CA 94043, USA
davide.venturelli@nasa.gov

Abstract. A major limitation of current generations of quantum annealers is the sparse connectivity of manufactured qubits in the hardware graph. This technological limitation has generated considerable interest, motivating efforts to design efficient and adroit minor-embedding procedures that bypass sparsity constraints. In this paper, starting from a previous equational formulation by Dridi et al. (arXiv:1810.01440), we propose integer programming (IP) techniques for solving the minor-embedding problem. The first approach involves a direct translation from the previous equational formulation to IP, while the second decomposes the problem into an assignment master problem and fiber condition checking subproblems. The proposed methods are able to detect instance infeasibility and provide bounds on solution quality, capabilities not offered by currently employed heuristic methods. We demonstrate the efficacy of our methods with an extensive computational assessment involving three different families of random graphs of varying sizes and densities. The direct translation as a monolithic IP model can be solved with existing commercial solvers yielding valid minor-embeddings, however, is outperformed overall by the decomposition approach. Our results demonstrate the promise of our methods for the studied benchmarks, highlighting the advantages of using IP technology for minor-embedding problems.

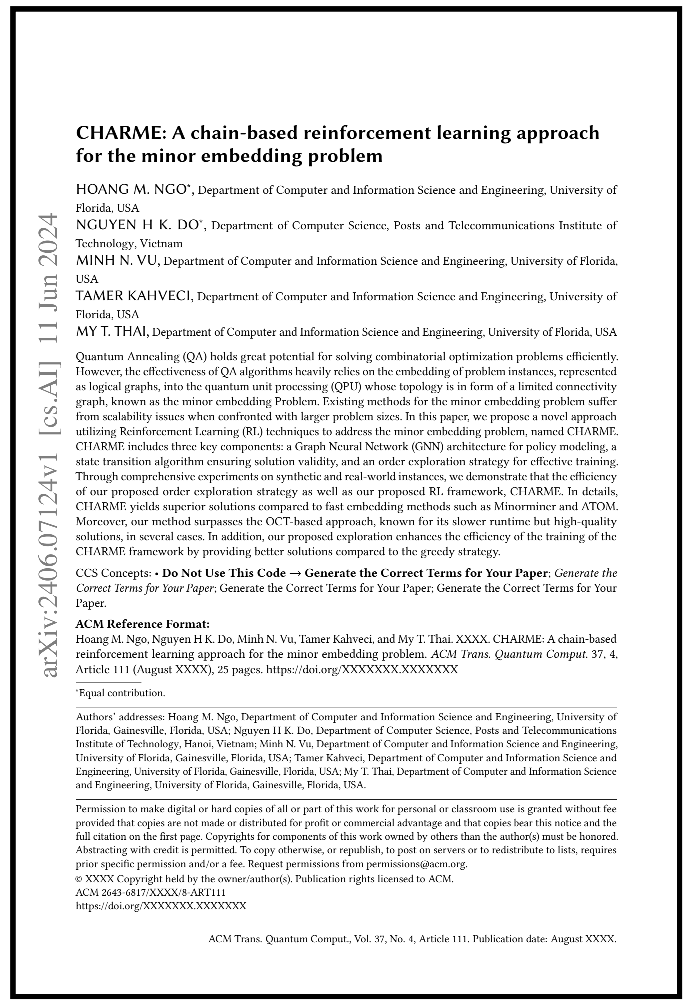
Keywords: Graph minors · Quantum annealers · Integer programming · Decomposition · Algebraic geometry

1 Introduction

Graph minor theory (GMT), the central theme of this work, is prominent across many fields. In quantum computing, GMT is employed to extend the scope of problems that can be represented on current quantum annealing hardware [5, 19]. Mapping a dense problem (logical) graph Y to a sparse (target) graph X can be achieved by constructing connected subgraphs of the target graph X from the high degree logical vertices y. The resulting mapping is called a minor-embedding of Y inside X.

Numerous heuristics for finding minor-embeddings have been proposed [1, 4, 29]. While these approaches are generally fast, they do not provide guarantees on the quality of the produced minor-embeddings nor can they prove the nonexistence of a minor-embedding for infeasible problems. An approach that attempts to address these shortcomings was recently introduced in Dridi et al. [9]. This approach uses tools from algebraic geometry and produces an equational formulation (as opposed to a purely combinatorial approach) to the minor-embedding problem.

In this paper, starting from this equational formulation, we propose integer programming (IP) techniques for tackling the embedding problem. Our proposed approaches differ from the computationally demanding Groebner bases computation used previously and are aimed at more efficiently computing embeddings while retaining the interesting properties that arise from the equational formulation of the problem. Our first approach, detailed in Section 3, directly translates the previous equational formulation to IP, while our second approach decomposes the problem into an assignment master problem and fiber condition checking



arXiv:2406.07124v1 [cs.AI] 11 Jun 2024

CHARME: A chain-based reinforcement learning approach for the minor embedding problem

HOANG M. NGO*, Department of Computer and Information Science and Engineering, University of Florida, USA
NGUYEN H K. DO*, Department of Computer Science, Posts and Telecommunications Institute of Technology, Vietnam
MINH N. VU, Department of Computer and Information Science and Engineering, University of Florida, USA
TAMER KAHVECI, Department of Computer and Information Science and Engineering, University of Florida, USA
MY T. THAI, Department of Computer and Information Science and Engineering, University of Florida, USA

Quantum Annealing (QA) holds great potential for solving combinatorial optimization problems efficiently. However, the effectiveness of QA algorithms heavily relies on the embedding of problem instances, represented as logical graphs, into the quantum unit processing (QPU) whose topology is in form of a limited connectivity graph, known as the minor embedding Problem. Existing methods for the minor embedding problem suffer from scalability issues when confronted with larger problem sizes. In this paper, we propose a novel approach utilizing Reinforcement Learning (RL) techniques to address the minor embedding problem, named CHARME. CHARME includes three key components: a Graph Neural Network (GNN) architecture for policy modeling, a state transition algorithm ensuring solution validity, and an order exploration strategy for effective training. Through comprehensive experiments on synthetic and real-world instances, we demonstrate that the efficiency of our proposed order exploration strategy as well as our proposed RL framework, CHARME. In details, CHARME yields superior solutions compared to fast embedding methods such as Minorminer and ATOM. Moreover, our method surpasses the OCT-based approach, known for its slower runtime but high-quality solutions, in several cases. In addition, our proposed exploration enhances the efficiency of the training of the CHARME framework by providing better solutions compared to the greedy strategy.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

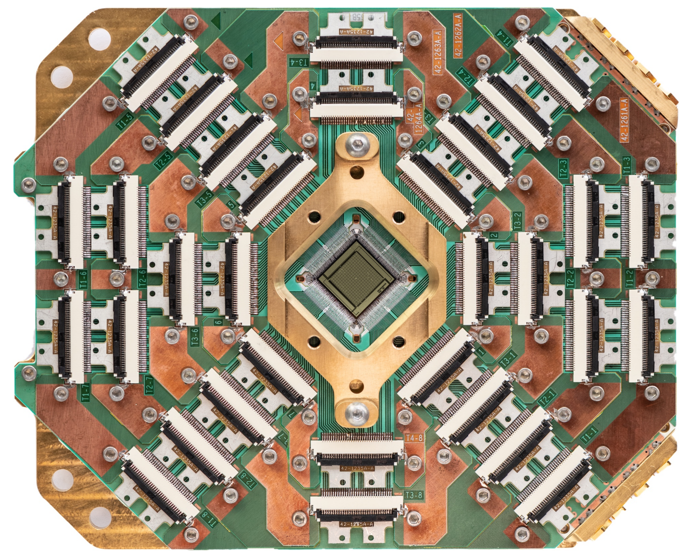
ACM Reference Format:
Hoang M. Ngo, Nguyen H K. Do, Minh N. Vu, Tamer Kahveci, and My T. Thai. XXXX. CHARME: A chain-based reinforcement learning approach for the minor embedding problem. *ACM Trans. Quantum Comput.* 37, 4, Article 111 (August XXXX), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

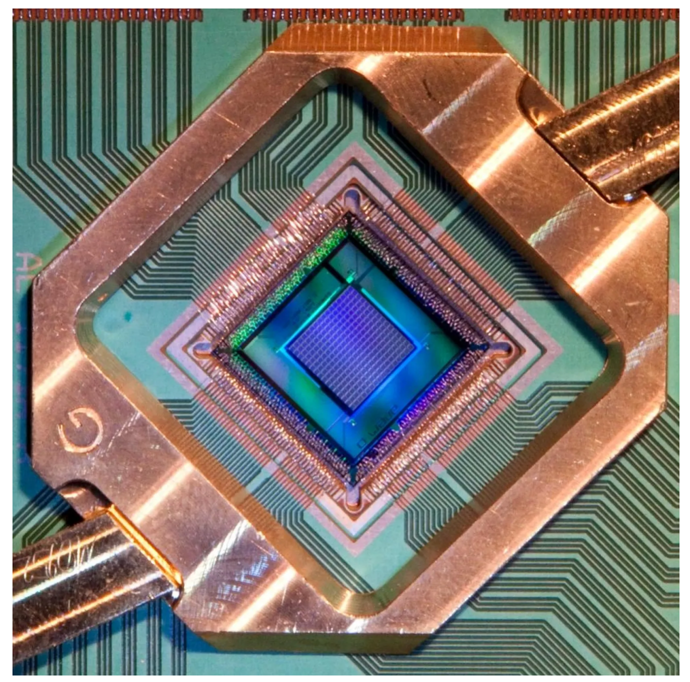
*Equal contribution.

Authors' addresses: Hoang M. Ngo, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; Nguyen H K. Do, Department of Computer Science, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam; Minh N. Vu, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; Tamer Kahveci, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; My T. Thai, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© XXXX Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 2643-6817/XXXX/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

ACM Trans. Quantum Comput., Vol. 37, No. 4, Article 111. Publication date: August XXXX.

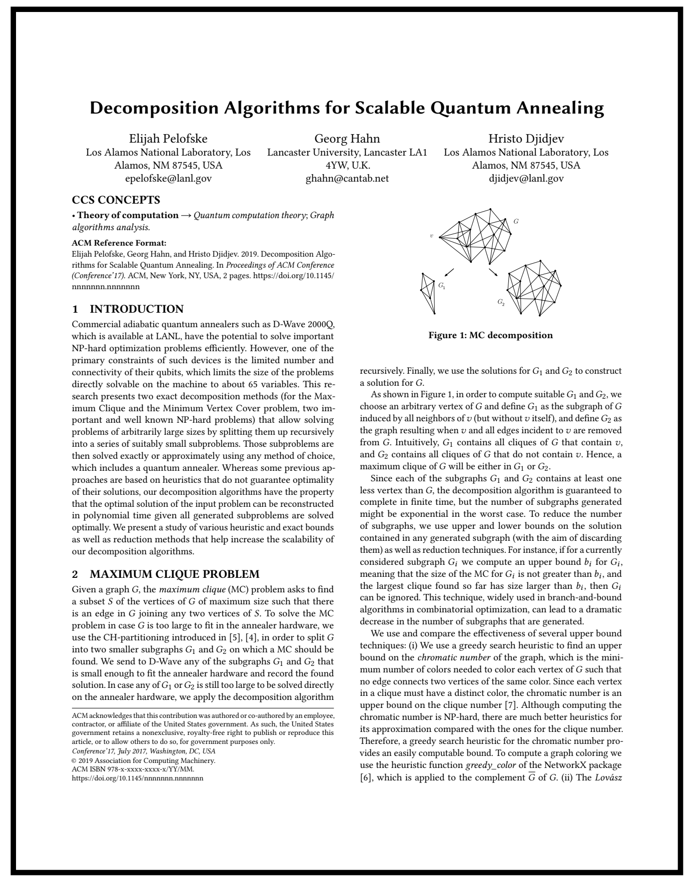




... ...

QUBO Problem → **Decomposition** → **Embedding** → **Solver** → **Post-Proc.** → ***Solution***

Choice of Method



Decomposition Algorithms for Scalable Quantum Annealing

Elijah Pelofske, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, epelofske@lanl.gov
Georg Hahn, Lancaster University, Lancaster LA1 4YW, U.K., ghahn@cantab.net
Hristo Djidjev, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, djidjev@lanl.gov

CCS CONCEPTS

• **Theory of computation** → *Quantum computation theory*; *Graph algorithms analysis.*

ACM Reference Format:
Elijah Pelofske, Georg Hahn, and Hristo Djidjev. 2019. Decomposition Algorithms for Scalable Quantum Annealing. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Commercial adiabatic quantum annealers such as D-Wave 2000Q, which is available at LANL, have the potential to solve important NP-hard optimization problems efficiently. However, one of the primary constraints of such devices is the limited number and connectivity of their qubits, which limits the size of the problems directly solvable on the machine to about 65 variables. This research presents two exact decomposition methods (for the Maximum Clique and the Minimum Vertex Cover problem, two important and well known NP-hard problems) that allow solving problems of arbitrarily large sizes by splitting them up recursively into a series of suitably small subproblems. Those subproblems are then solved exactly or approximately using any method of choice, which includes a quantum annealer. Whereas some previous approaches are based on heuristics that do not guarantee optimality of their solutions, our decomposition algorithms have the property that the optimal solution of the input problem can be reconstructed in polynomial time given all generated subproblems are solved optimally. We present a study of various heuristic and exact bounds as well as reduction methods that help increase the scalability of our decomposition algorithms.

2 MAXIMUM CLIQUE PROBLEM

Given a graph G, the *maximum clique* (MC) problem asks to find a subset S of the vertices of G of maximum size such that there is an edge in G joining any two vertices of S. To solve the MC problem in case G is too large to fit in the annealer hardware, we use the CH-partitioning introduced in [5], [4], in order to split G into two smaller subgraphs G_1 and G_2 on which a MC should be found. We send to D-Wave any of the subgraphs G_1 and G_2 that is small enough to fit the annealer hardware and record the found solution. In case any of G_1 or G_2 is still too large to be solved directly on the annealer hardware, we apply the decomposition algorithm

Figure 1: MC decomposition

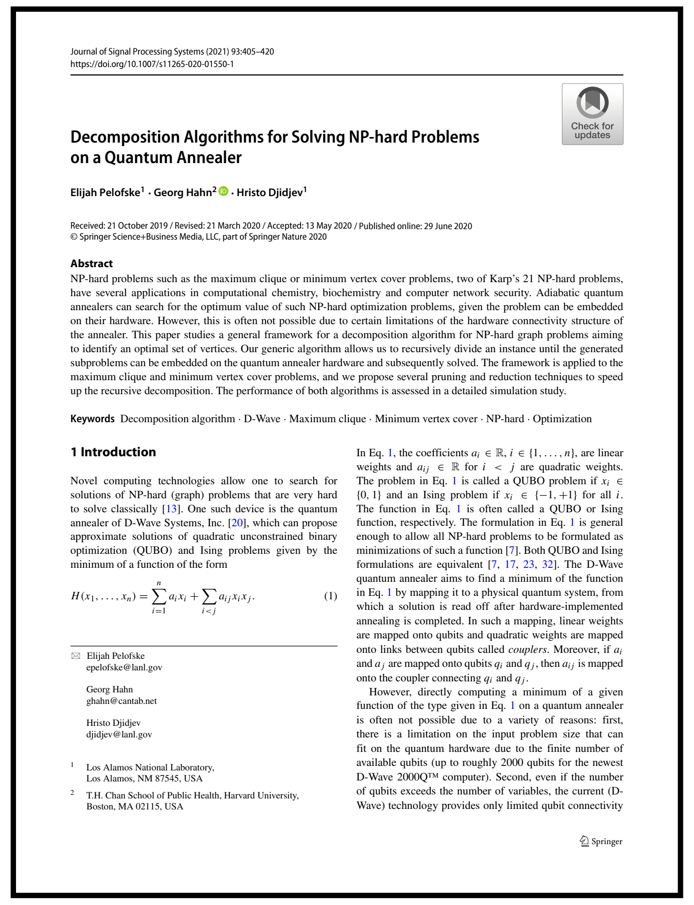
recursively. Finally, we use the solutions for G_1 and G_2 to construct a solution for G.

As shown in Figure 1, in order to compute suitable G_1 and G_2, we choose an arbitrary vertex of G and define G_1 as the subgraph of G induced by all neighbors of v (but without v itself), and define G_2 as the graph resulting when v and all edges incident to v are removed from G. Intuitively, G_1 contains all cliques of G that contain v, and G_2 contains all cliques of G that do not contain v. Hence, a maximum clique of G will be either in G_1 or G_2.

Since each of the subgraphs G_1 and G_2 contains at least one less vertex than G, the decomposition algorithm is guaranteed to complete in finite time, but the number of subgraphs generated might be exponential in the worst case. To reduce the number of subgraphs, we use upper and lower bounds on the solution contained in any generated subgraph (with the aim of discarding them) as well as reduction techniques. For instance, if for a currently considered subgraph G_i we compute an upper bound b_i for G_i, meaning that the size of the MC for G_i is not greater than b_i, and the largest clique found so far has size larger than b_i, then G_i can be ignored. This technique, widely used in branch-and-bound algorithms in combinatorial optimization, can lead to a dramatic decrease in the number of subgraphs that are generated.

We use and compare the effectiveness of several upper bound techniques: (i) We use a greedy search heuristic to find an upper bound on the *chromatic number* of the graph, which is the minimum number of colors needed to color each vertex of G such that no edge connects two vertices of the same color. Since each vertex in a clique must have a distinct color, the chromatic number is an upper bound on the clique number [7]. Although computing the chromatic number is NP-hard, there are much better heuristics for its approximation compared with the ones for the clique number. Therefore, a greedy search heuristic for the chromatic number provides an easily computable bound. To compute a graph coloring we use the heuristic function *greedy_color* of the NetworkX package [6], which is applied to the complement $\overline{G}$ of G. (ii) The *Lovász*

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.
Conference'17, July 2017, Washington, DC, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn



Journal of Signal Processing Systems (2021) 93:405–420
https://doi.org/10.1007/s11265-020-01550-1

Check for updates

Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer

Elijah Pelofske[1] · Georg Hahn[2] · Hristo Djidjev[1]

Received: 21 October 2019 / Revised: 21 March 2020 / Accepted: 13 May 2020 / Published online: 29 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

NP-hard problems such as the maximum clique or minimum vertex cover problems, two of Karp's 21 NP-hard problems, have several applications in computational chemistry, biochemistry and computer network security. Adiabatic quantum annealers can search for the optimum value of such NP-hard optimization problems, given the problem can be embedded on their hardware. However, this is often not possible due to certain limitations of the hardware connectivity structure of the annealer. This paper studies a general framework for a decomposition algorithm for NP-hard graph problems aiming to identify an optimal set of vertices. Our generic algorithm allows us to recursively divide an instance until the generated subproblems can be embedded on the quantum annealer hardware and subsequently solved. The framework is applied to the maximum clique and minimum vertex cover problems, and we propose several pruning and reduction techniques to speed up the recursive decomposition. The performance of both algorithms is assessed in a detailed simulation study.

Keywords Decomposition algorithm · D-Wave · Maximum clique · Minimum vertex cover · NP-hard · Optimization

1 Introduction

Novel computing technologies allow one to search for solutions of NP-hard (graph) problems that are very hard to solve classically [13]. One such device is the quantum annealer of D-Wave Systems, Inc. [20], which can propose approximate solutions of quadratic unconstrained binary optimization (QUBO) and Ising problems given by the minimum of a function of the form

$$H(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i + \sum_{i<j} a_{ij} x_i x_j. \quad (1)$$

In Eq. 1, the coefficients $a_i \in \mathbb{R}$, $i \in \{1, \ldots, n\}$, are linear weights and $a_{ij} \in \mathbb{R}$ for $i < j$ are quadratic weights. The problem in Eq. 1 is called a QUBO problem if $x_i \in \{0, 1\}$ and an Ising problem if $x_i \in \{-1, +1\}$ for all i. The function in Eq. 1 is often called a QUBO or Ising function, respectively. The formulation in Eq. 1 is general enough to allow all NP-hard problems to be formulated as minimizations of such a function [7]. Both QUBO and Ising formulations are equivalent [7, 17, 23, 32]. The D-Wave quantum annealer aims to find a minimum of the function in Eq. 1 by mapping it to a physical quantum system, from which a solution is read off after hardware-implemented annealing is completed. In such a mapping, linear weights are mapped onto qubits and quadratic weights are mapped onto links between qubits called *couplers*. Moreover, if a_i and a_j are mapped onto qubits q_i and q_j, then a_{ij} is mapped onto the coupler connecting q_i and q_j.

However, directly computing a minimum of a given function of the type given in Eq. 1 on a quantum annealer is often not possible due to a variety of reasons: first, there is a limitation on the input problem size that can fit on the quantum hardware due to the finite number of available qubits (up to roughly 2000 qubits for the newest D-Wave 2000Q™ computer). Second, even if the number of qubits exceeds the number of variables, the current (D-Wave) technology provides only limited qubit connectivity

✉ Elijah Pelofske
epelofske@lanl.gov

Georg Hahn
ghahn@cantab.net

Hristo Djidjev
djidjev@lanl.gov

[1] Los Alamos National Laboratory, Los Alamos, NM 87545, USA

[2] T.H. Chan School of Public Health, Harvard University, Boston, MA 02115, USA

Springer

...

arXiv:1912.08314v2 [quant-ph] 20 Dec 2019

Integer programming techniques for minor-embedding in quantum annealers

David E. Bernal[1,4,5], Kyle E. C. Booth[2], Raouf Dridi[3], Hedayat Alghassi[3], Sridhar Tayur[3], and Davide Venturelli[4,5]

[1] Department of Chemical Engineering, Carnegie Mellon University, Pittsurgh PA 15213, USA
bernalde@cmu.edu
[2] Department of Mechanical & Industrial Engineering, University of Toronto, Toronto ON M5S 3G8, Canada
kbooth@mie.utoronto.ca
[3] Tepper School of Business, Carnegie Mellon University, Pittsburgh PA 15213, USA
{rdridi,halghass,stayur}@andrew.cmu.edu
[4] Quantum AI Lab (QuAIL), NASA Ames Research Center, Moffett Field CA 94035, USA
[5] USRA (RIACS), Mountain View CA 94043, USA
davide.venturelli@nasa.gov

Abstract. A major limitation of current generations of quantum annealers is the sparse connectivity of manufactured qubits in the hardware graph. This technological limitation has generated considerable interest, motivating efforts to design efficient and adroit minor-embedding procedures that bypass sparsity constraints. In this paper, starting from a previous equational formulation by Dridi et al. (arXiv:1810.01440), we propose integer programming (IP) techniques for solving the minor-embedding problem. The first approach involves a direct translation from the previous equational formulation to IP, while the second decomposes the problem into an assignment master problem and fiber condition checking subproblems. The proposed methods are able to detect instance infeasibility and provide bounds on solution quality, capabilities not offered by currently employed heuristic methods. We demonstrate the efficacy of our methods with an extensive computational assessment involving three different families of random graphs of varying sizes and densities. The direct translation as a monolithic IP model can be solved with existing commercial solvers yielding valid minor-embeddings, however, is outperformed overall by the decomposition approach. Our results demonstrate the promise of our methods for the studied benchmarks, highlighting the advantages of using IP technology for minor-embedding problems.

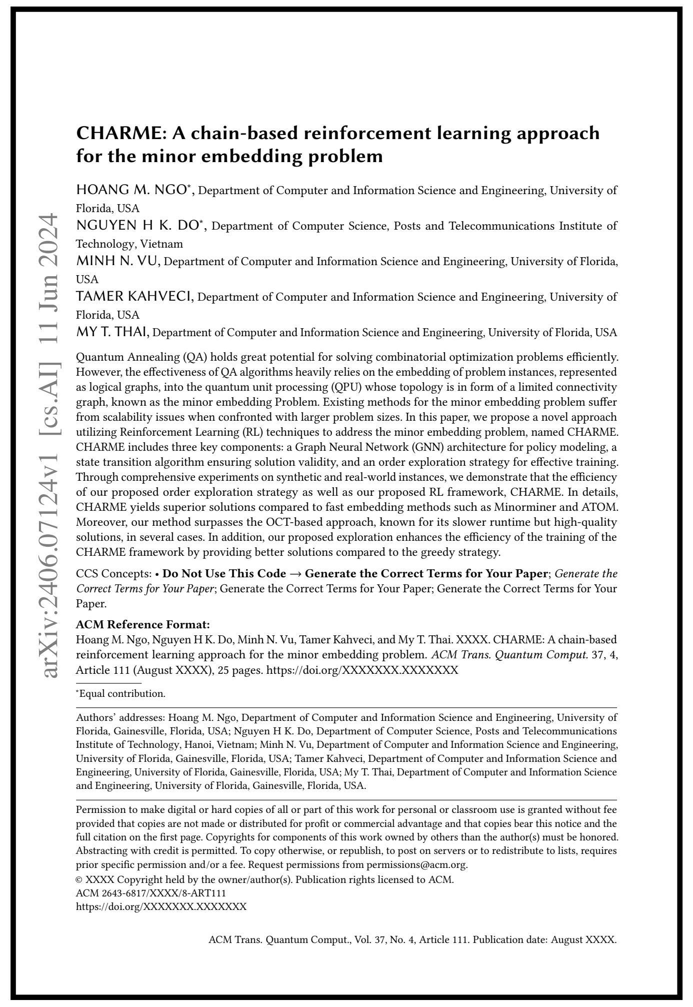
Keywords: Graph minors · Quantum annealers · Integer programming · Decomposition · Algebraic geometry

1 Introduction

Graph minor theory (GMT), the central theme of this work, is prominent across many fields. In quantum computing, GMT is employed to extend the scope of problems that can be represented on current quantum annealing hardware [5, 19]. Mapping a dense problem (logical) graph Y to a sparse (target) graph X can be achieved by constructing connected subgraphs of the target graph X from the high degree logical vertices y. The resulting mapping is called a minor-embedding of Y inside X.

Numerous heuristics for finding minor-embeddings have been proposed [1, 4, 29]. While these approaches are generally fast, they do not provide guarantees on the quality of the produced minor-embeddings nor can they prove the nonexistence of a minor-embedding for infeasible problems. An approach that attempts to address these shortcomings was recently introduced in Dridi et al. [9]. This approach uses tools from algebraic geometry and produces an equational formulation (as opposed to a purely combinatorial approach) to the minor-embedding problem.

In this paper, starting from this equational formulation, we propose integer programming (IP) techniques for tackling the embedding problem. Our proposed approaches differ from the computationally demanding Groebner bases computation used previously and are aimed at more efficiently computing embeddings while retaining the interesting properties that arise from the equational formulation of the problem. Our first approach, detailed in Section 3, directly translates the previous equational formulation to IP, while our second approach decomposes the problem into an assignment master problem and fiber condition checking



arXiv:2406.07124v1 [cs.AI] 11 Jun 2024

CHARME: A chain-based reinforcement learning approach for the minor embedding problem

HOANG M. NGO*, Department of Computer and Information Science and Engineering, University of Florida, USA
NGUYEN H K. DO*, Department of Computer Science, Posts and Telecommunications Institute of Technology, Vietnam
MINH N. VU, Department of Computer and Information Science and Engineering, University of Florida, USA
TAMER KAHVECI, Department of Computer and Information Science and Engineering, University of Florida, USA
MY T. THAI, Department of Computer and Information Science and Engineering, University of Florida, USA

Quantum Annealing (QA) holds great potential for solving combinatorial optimization problems efficiently. However, the effectiveness of QA algorithms heavily relies on the embedding of problem instances, represented as logical graphs, into the quantum unit processing (QPU) whose topology is in form of a limited connectivity graph, known as the minor embedding Problem. Existing methods for the minor embedding problem suffer from scalability issues when confronted with larger problem sizes. In this paper, we propose a novel approach utilizing Reinforcement Learning (RL) techniques to address the minor embedding problem, named CHARME. CHARME includes three key components: a Graph Neural Network (GNN) architecture for policy modeling, a state transition algorithm ensuring solution validity, and an order exploration strategy for effective training. Through comprehensive experiments on synthetic and real-world instances, we demonstrate that the efficiency of our proposed order exploration strategy as well as our proposed RL framework, CHARME. In details, CHARME yields superior solutions compared to fast embedding methods such as Minorminer and ATOM. Moreover, our method surpasses the OCT-based approach, known for its slower runtime but high-quality solutions, in several cases. In addition, our proposed exploration enhances the efficiency of the training of the CHARME framework by providing better solutions compared to the greedy strategy.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

ACM Reference Format:
Hoang M. Ngo, Nguyen H K. Do, Minh N. Vu, Tamer Kahveci, and My T. Thai. XXXX. CHARME: A chain-based reinforcement learning approach for the minor embedding problem. *ACM Trans. Quantum Comput.* 37, 4, Article 111 (August XXXX), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

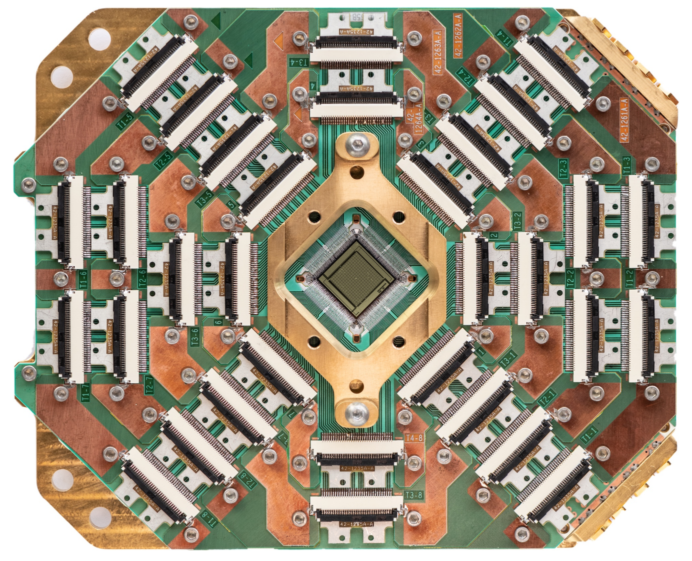
*Equal contribution.

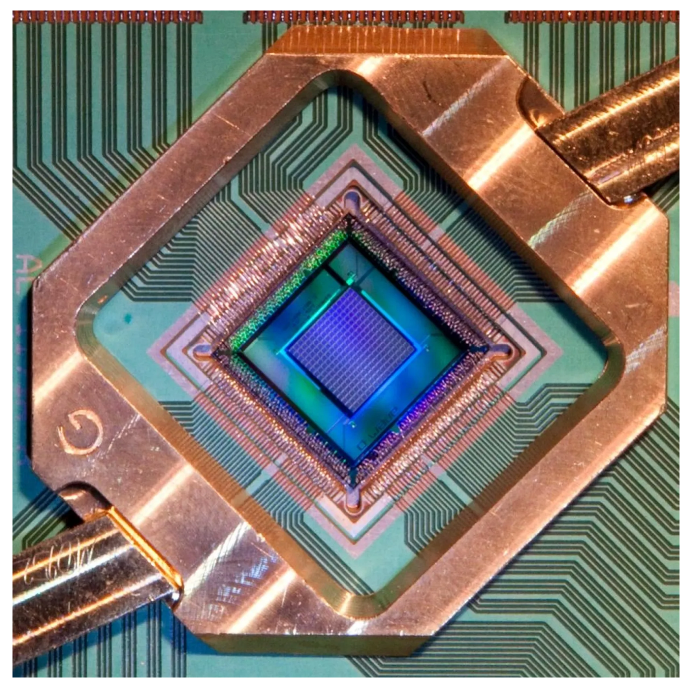
Authors' addresses: Hoang M. Ngo, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; Nguyen H K. Do, Department of Computer Science, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam; Minh N. Vu, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; Tamer Kahveci, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA; My T. Thai, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© XXXX Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 2643-6817/XXXX/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

ACM Trans. Quantum Comput., Vol. 37, No. 4, Article 111. Publication date: August XXXX.

...





...

PHYSICAL REVIEW A **92**, 042310 (2015)

Quantum annealing correction with minor embedding

Walter Vinci,[1,2,3] Tameem Albash,[2,3,4] Gerardo Paz-Silva,[5,6] Itay Hen,[3,4] and Daniel A. Lidar[1,2,3,7]
[1]*Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, USA*
[2]*Department of Physics and Astronomy, University of Southern California, Los Angeles, California 90089, USA*
[3]*Center for Quantum Information Science & Technology, University of Southern California, Los Angeles, California 90089, USA*
[4]*Information Sciences Institute, University of Southern California, Marina del Rey, California 90292, USA*
[5]*Centre for Quantum Computation and Communication Technology (Australian Research Council), Griffith University, Brisbane, Queensland 4111, Australia*
[6]*Centre for Quantum Dynamics, Griffith University, Brisbane, Queensland 4111, Australia*
[7]*Department of Chemistry, University of Southern California, Los Angeles, California 90089, USA*
(Received 13 July 2015; published 9 October 2015)

Quantum annealing provides a promising route for the development of quantum optimization devices, but the usefulness of such devices will be limited in part by the range of implementable problems as dictated by hardware constraints. To overcome constraints imposed by restricted connectivity between qubits, a larger set of interactions can be approximated using minor embedding techniques whereby several physical qubits are used to represent a single logical qubit. However, minor embedding introduces new types of errors due to its approximate nature. We introduce and study quantum annealing correction schemes designed to improve the performance of quantum annealers in conjunction with minor embedding, thus leading to a hybrid scheme defined over an encoded graph. We argue that this scheme can be efficiently decoded using an energy minimization technique provided the density of errors does not exceed the per-site percolation threshold of the encoded graph. We test the hybrid scheme using a D-Wave Two processor on problems for which the encoded graph is a two-level grid and the Ising model is known to be NP-hard. The problems we consider are frustrated Ising model problem instances with "planted" (*a priori* known) solutions. Applied in conjunction with optimized energy penalties and decoding techniques, we find that this approach enables the quantum annealer to solve minor embedded instances with significantly higher success probability than it would without error correction. Our work demonstrates that quantum annealing correction can and should be used to improve the robustness of quantum annealing not only for natively embeddable problems but also when minor embedding is used to extend the connectivity of physical devices.

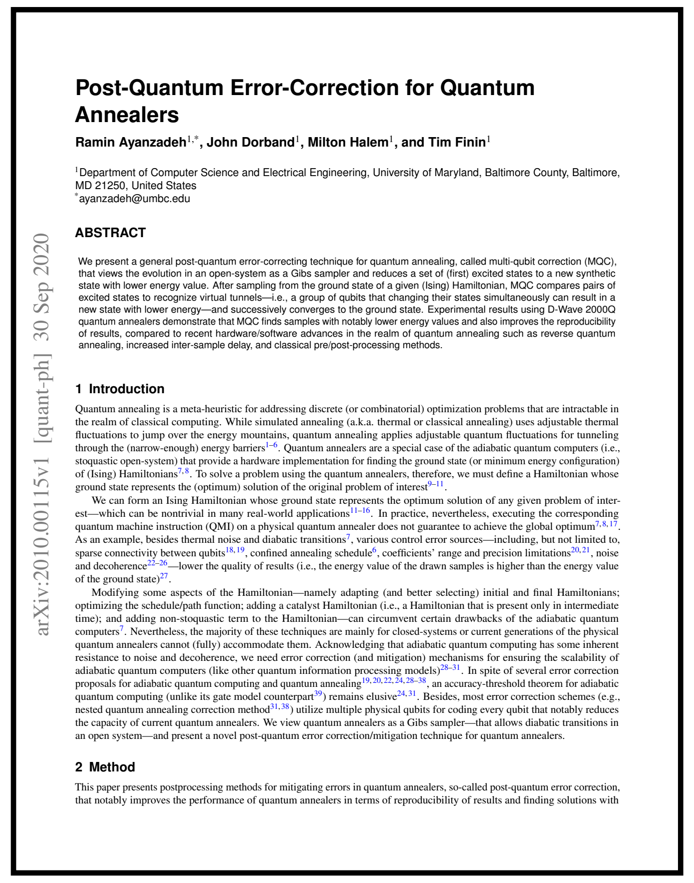
DOI: 10.1103/PhysRevA.92.042310 PACS number(s): 03.67.Ac, 03.65.Yz

I. INTRODUCTION

Quantum computers are in principle able to solve computational problems that are intractable for their classical counterparts [1,2]. Remarkably, rudimentary quantum annealers—a special type of adiabatic quantum computer [3] specifically engineered to implement quantum annealing [4–9] to solve hard optimization problems [10]—are already commercially available [11–13], and their availability to the scientific community has recently stimulated a productive debate [14–24]. The question at stake is whether current experimental quantum annealing can really be considered a form of quantum computation, even when the decoherence time of the component qubits is much smaller than the overall computational time. While theory suggests that the answer to this question can be positive [25,26], as it concerns quantum annealing experiments the answer is still open, in particular due to the "black-box" nature of quantum annealing, which makes it difficult to unambiguously determine the nature of the physical processes taking place between the initialization and readout steps. However, convincing evidence is mounting that quantum effects, including entanglement [27] and multiqubit coherent tunneling [28], play a functional role in explaining the behavior of available annealers. Moreover, open system quantum dynamical models [28,29] have been successful in reproducing experimental data from quantum annealing experiments [16,21,22,28,30]. A closely related question is whether quantum annealers are able to provide quantum speedup, a theoretical possibility [31] despite skepticism based, e.g., on the appearance of exponentially small gaps [32]. While compelling evidence of a quantum speedup is still lacking with current physical annealers, it is also generally recognized that technological improvements (in particular reduced control errors and shorter annealing times) in the next generation of quantum annealers are necessary in order to unambiguously address the issue [33–37]. A careful choice of the benchmark problems is another important consideration [38,39], in particular in light of the fragility of spin glasses to small perturbations (temperature chaos) [40,41].

Ultimately, the usefulness of quantum annealers relies on the promise of achieving quantum information processing on a large scale. This goal can only be met through the use of some form of quantum error correction (QEC), which is necessary to protect the performance of any quantum computation against decoherence [42–44]. The scalability of quantum computing via QEC has been established in the gate model in the form of the celebrated threshold theorem (e.g, Refs. [45,46]). Adiabatic quantum computation and quantum annealing also require some form of error correction in order to preserve their advantages over classical computation. This is true even though adiabatic dynamics is intrinsically robust to some forms of decoherence [47–49]. Several error suppression techniques suitable for adiabatic quantum computing (AQC) have been proposed [50–54]. However, despite valiant theoretical attempts [55] much less is known about how to achieve fault tolerant AQC, and some negative results have

1050-2947/2015/92(4)/042310(30) 042310-1 ©2015 American Physical Society



arXiv:2010.00115v1 [quant-ph] 30 Sep 2020

Post-Quantum Error-Correction for Quantum Annealers

Ramin Ayanzadeh[1,*], John Dorband[1], Milton Halem[1], and Tim Finin[1]

[1]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, United States
*ayanzadeh@umbc.edu

ABSTRACT

We present a general post-quantum error-correcting technique for quantum annealing, called multi-qubit correction (MQC), that views the evolution in an open-system as a Gibs sampler and reduces a set of (first) excited states to a new synthetic state with lower energy value. After sampling from the ground state of a given (Ising) Hamiltonian, MQC compares pairs of excited states to recognize virtual tunnels—i.e., a group of qubits that changing their states simultaneously can result in a new state with lower energy—and successively converges to the ground state. Experimental results using D-Wave 2000Q quantum annealers demonstrate that MQC finds samples with notably lower energy values and also improves the reproducibility of results, compared to recent hardware/software advances in the realm of quantum annealing such as reverse quantum annealing, increased inter-sample delay, and classical pre/post-processing methods.

1 Introduction

Quantum annealing is a meta-heuristic for addressing discrete (or combinatorial) optimization problems that are intractable in the realm of classical computing. While simulated annealing (a.k.a. thermal or classical annealing) uses adjustable thermal fluctuations to jump over the energy mountains, quantum annealing applies adjustable quantum fluctuations for tunneling through the (narrow-enough) energy barriers[1–6]. Quantum annealers are a special case of the adiabatic quantum computers (i.e., stoquastic open-system) that provide a hardware implementation for finding the ground state (or minimum energy configuration) of (Ising) Hamiltonians[7,8]. To solve a problem using the quantum annealers, therefore, we must define a Hamiltonian whose ground state represents the (optimum) solution of the original problem of interest[9–11].

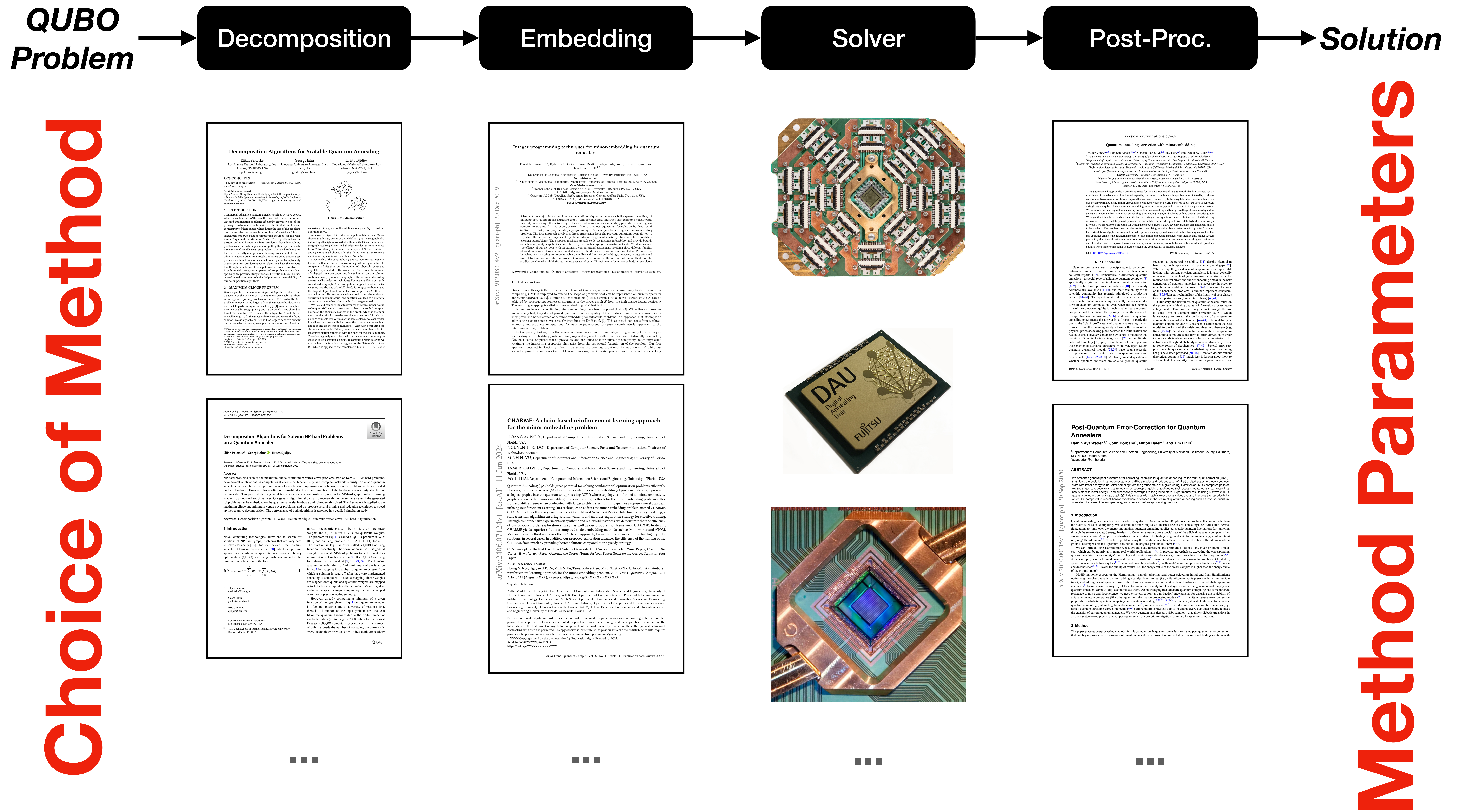
We can form an Ising Hamiltonian whose ground state represents the optimum solution of any given problem of interest—which can be nontrivial in many real-world applications[11–16]. In practice, nevertheless, executing the corresponding quantum machine instruction (QMI) on a physical quantum annealer does not guarantee to achieve the global optimum[7,8,17]. As an example, besides thermal noise and diabatic transitions[7], various control error sources—including, but not limited to, sparse connectivity between qubits[18,19], confined annealing schedule[6], coefficients' range and precision limitations[20,21], noise and decoherence[22–26]—lower the quality of results (i.e., the energy value of the drawn samples is higher than the energy value of the ground state)[27].

Modifying some aspects of the Hamiltonian—namely adapting (and better selecting) initial and final Hamiltonians; optimizing the schedule/path function; adding a catalyst Hamiltonian (i.e., a Hamiltonian that is present only in intermediate time); and adding non-stoquastic term to the Hamiltonian—can circumvent certain drawbacks of the adiabatic quantum computers[7]. Nevertheless, the majority of these techniques are mainly for closed-systems or current generations of the physical quantum annealers cannot (fully) accommodate them. Acknowledging that adiabatic quantum computing has some inherent resistance to noise and decoherence, we need error correction (and mitigation) mechanisms for ensuring the scalability of adiabatic quantum computers (like other quantum information processing models)[28–31]. In spite of several error correction proposals for adiabatic quantum computing and quantum annealing[19,20,22,24,28–38], an accuracy-threshold theorem for adiabatic quantum computing (unlike its gate model counterpart[39]) remains elusive[24,31]. Besides, most error correction schemes (e.g., nested quantum annealing correction method[31,38]) utilize multiple physical qubits for coding every qubit that notably reduces the capacity of current quantum annealers. We view quantum annealers as a Gibs sampler—that allows diabatic transitions in an open system—and present a novel post-quantum error correction/mitigation technique for quantum annealers.

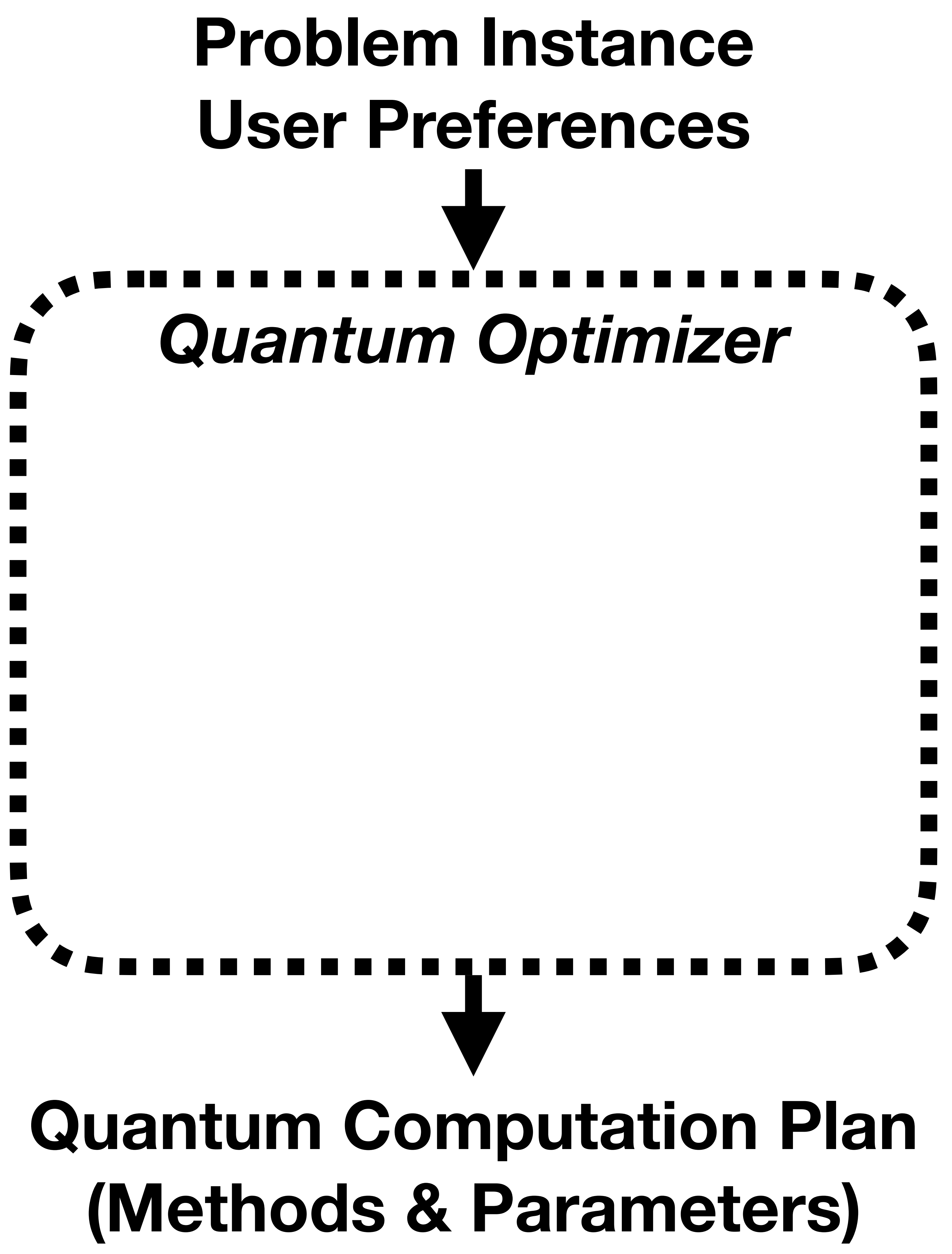
2 Method

This paper presents postprocessing methods for mitigating errors in quantum annealers, so-called post-quantum error correction, that notably improves the performance of quantum annealers in terms of reproducibility of results and finding solutions with

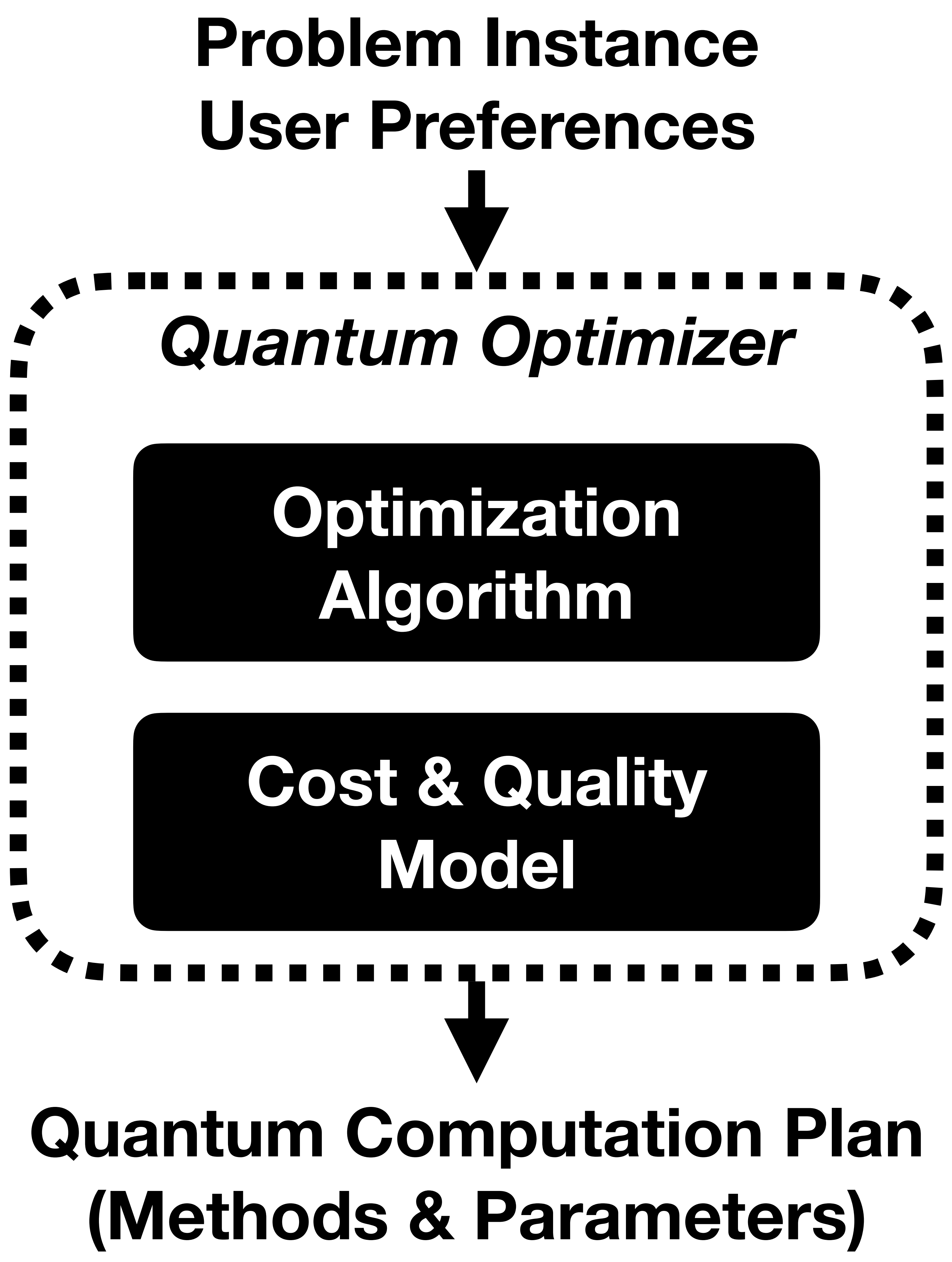
...


QUBO Problem
Decomposition
Embedding
Solver
Post-Proc.
Solution
Choice of Method
Method Parameters
Decomposition Algorithms for Scalable Quantum Annealing
Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer
Integer programming techniques for minor-embedding in quantum annealers
CHARME: A chain-based reinforcement learning approach for the minor embedding problem
DAU Digital Annealing Unit FUJITSU
Quantum annealing correction with minor embedding
Post-Quantum Error-Correction for Quantum Annealers
...
...
...
...

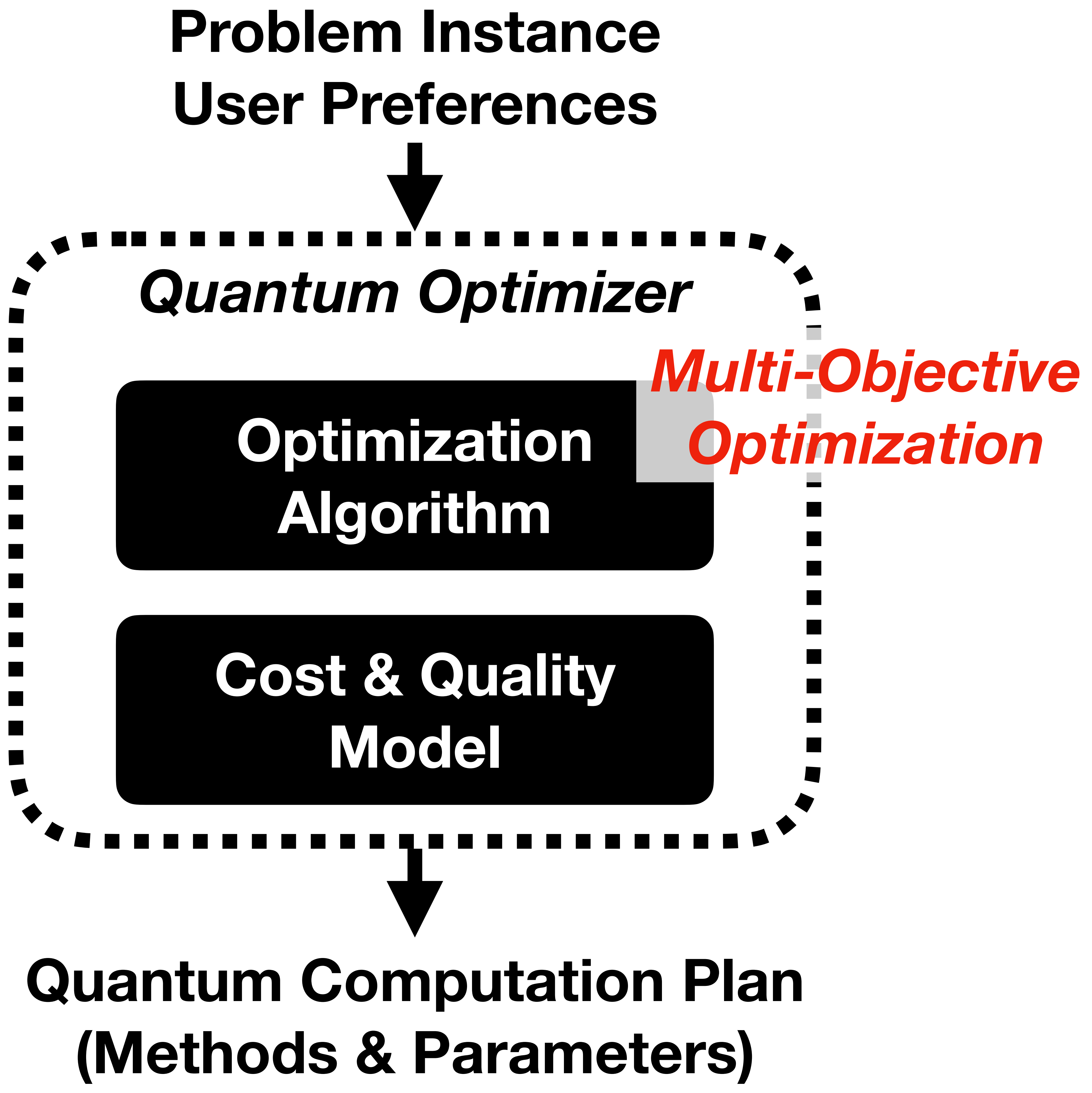
"Query Optimization" for Quantum Computing



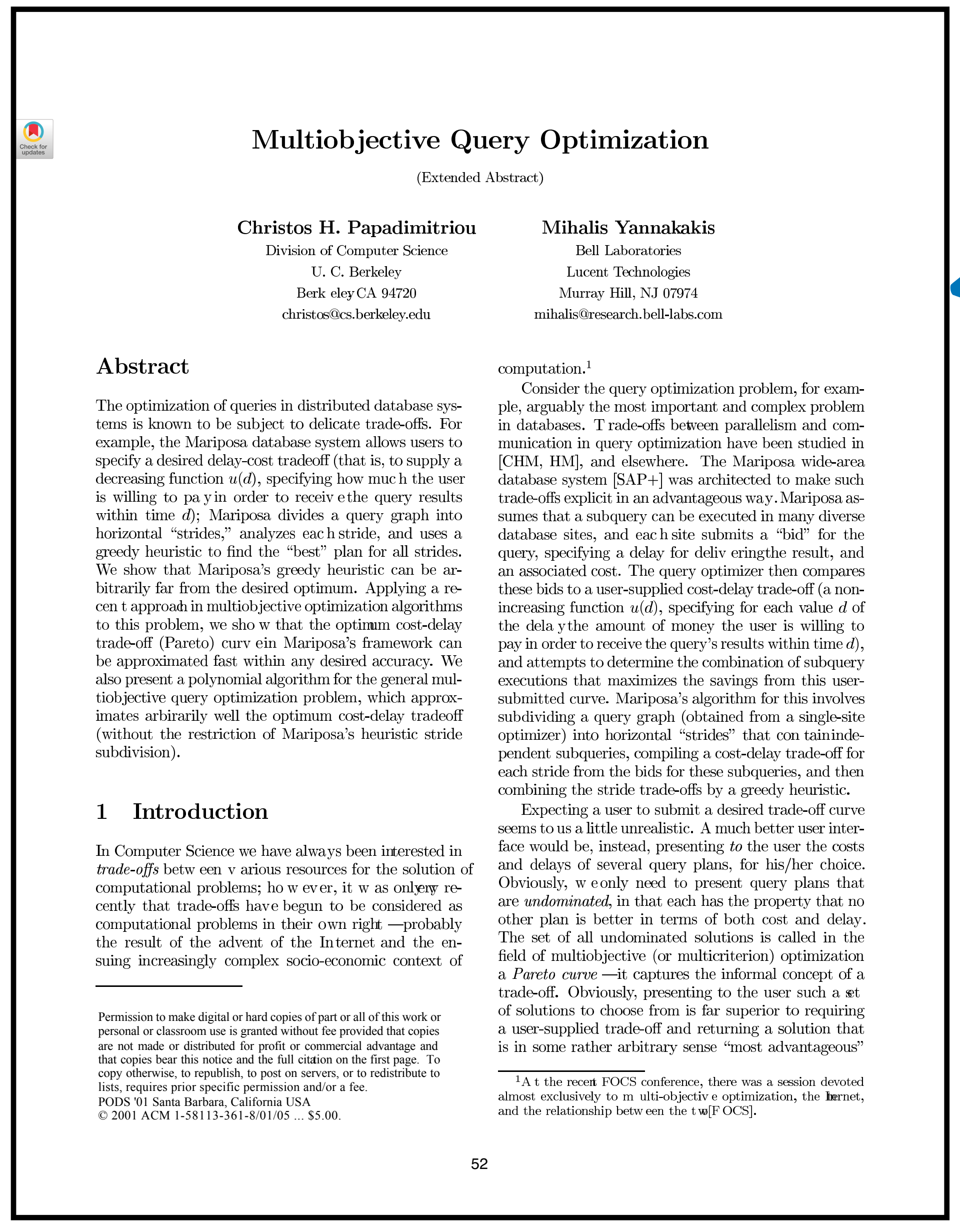
"Query Optimization" for Quantum Computing



"Query Optimization" for Quantum Computing



"Query Optimization" for Quantum Computing



Multiobjective Query Optimization

(Extended Abstract)

Christos H. Papadimitriou
Division of Computer Science
U. C. Berkeley
Berkeley CA 94720
christos@cs.berkeley.edu

Mihalis Yannakakis
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974
mihalis@research.bell-labs.com

Abstract

The optimization of queries in distributed database systems is known to be subject to delicate trade-offs. For example, the Mariposa database system allows users to specify a desired delay-cost tradeoff (that is, to supply a decreasing function $u(d)$, specifying how much the user is willing to pay in order to receive the query results within time d); Mariposa divides a query graph into horizontal "strides," analyzes each stride, and uses a greedy heuristic to find the "best" plan for all strides. We show that Mariposa's greedy heuristic can be arbitrarily far from the desired optimum. Applying a recent approach in multiobjective optimization algorithms to this problem, we show that the optimum cost-delay trade-off (Pareto) curve in Mariposa's framework can be approximated fast within any desired accuracy. We also present a polynomial algorithm for the general multiobjective query optimization problem, which approximates arbirarily well the optimum cost-delay tradeoff (without the restriction of Mariposa's heuristic stride subdivision).

1 Introduction

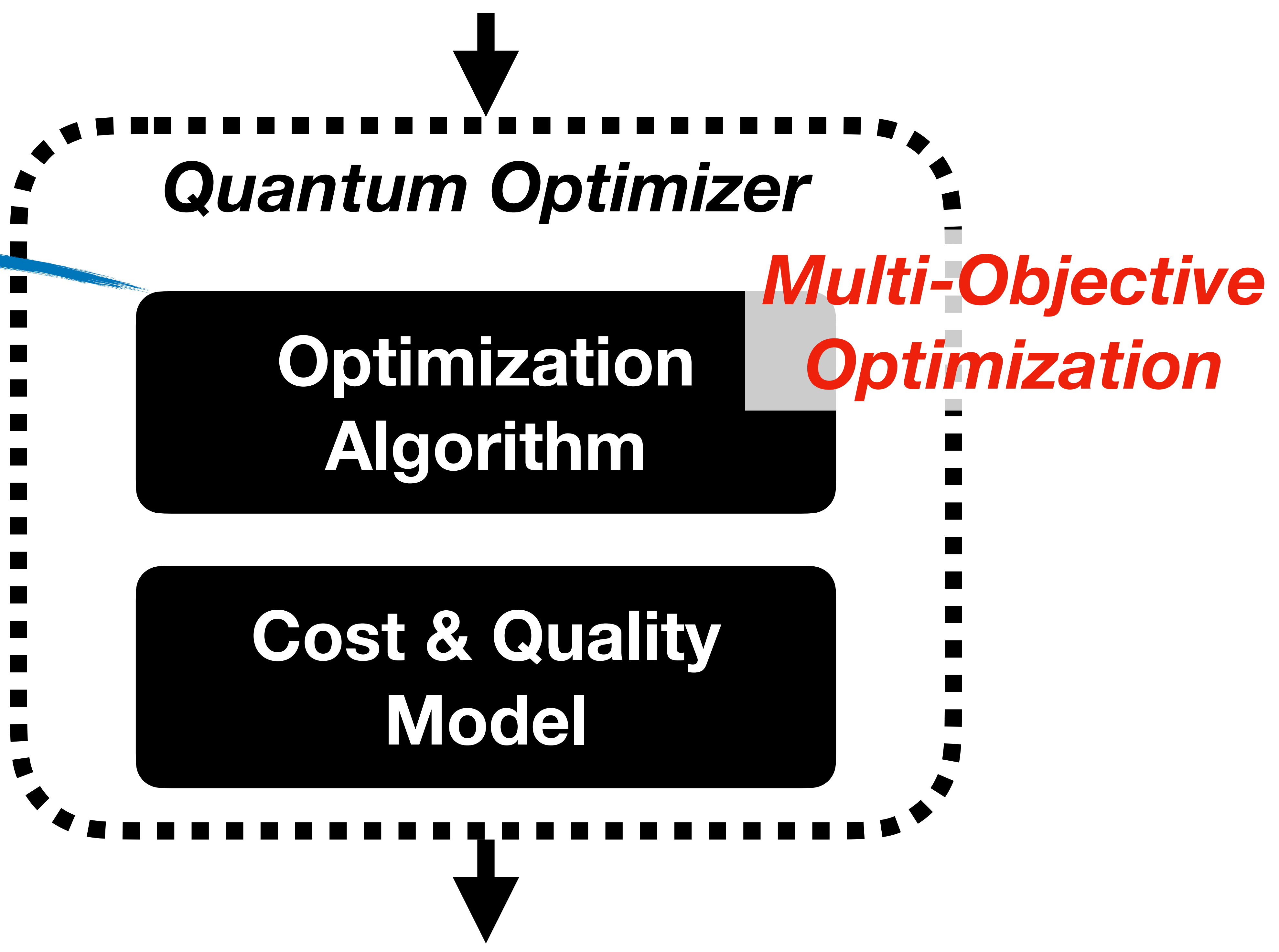
In Computer Science we have always been interested in *trade-offs* between various resources for the solution of computational problems; however, it was only very recently that trade-offs have begun to be considered as computational problems in their own right —probably the result of the advent of the Internet and the ensuing increasingly complex socio-economic context of computation.[1]

Consider the query optimization problem, for example, arguably the most important and complex problem in databases. Trade-offs between parallelism and communication in query optimization have been studied in [CHM, HM], and elsewhere. The Mariposa wide-area database system [SAP+] was architected to make such trade-offs explicit in an advantageous way. Mariposa assumes that a subquery can be executed in many diverse database sites, and each site submits a "bid" for the query, specifying a delay for delivering the result, and an associated cost. The query optimizer then compares these bids to a user-supplied cost-delay trade-off (a non-increasing function $u(d)$, specifying for each value d of the delay the amount of money the user is willing to pay in order to receive the query's results within time d), and attempts to determine the combination of subquery executions that maximizes the savings from this user-submitted curve. Mariposa's algorithm for this involves subdividing a query graph (obtained from a single-site optimizer) into horizontal "strides" that contain independent subqueries, compiling a cost-delay trade-off for each stride from the bids for these subqueries, and then combining the stride trade-offs by a greedy heuristic.

Expecting a user to submit a desired trade-off curve seems to us a little unrealistic. A much better user interface would be, instead, presenting *to* the user the costs and delays of several query plans, for his/her choice. Obviously, we only need to present query plans that are *undominated*, in that each has the property that no other plan is better in terms of both cost and delay. The set of all undominated solutions is called in the field of multiobjective (or multicriterion) optimization a *Pareto curve* —it captures the informal concept of a trade-off. Obviously, presenting to the user such a set of solutions to choose from is far superior to requiring a user-supplied trade-off and returning a solution that is in some rather arbitrary sense "most advantageous"

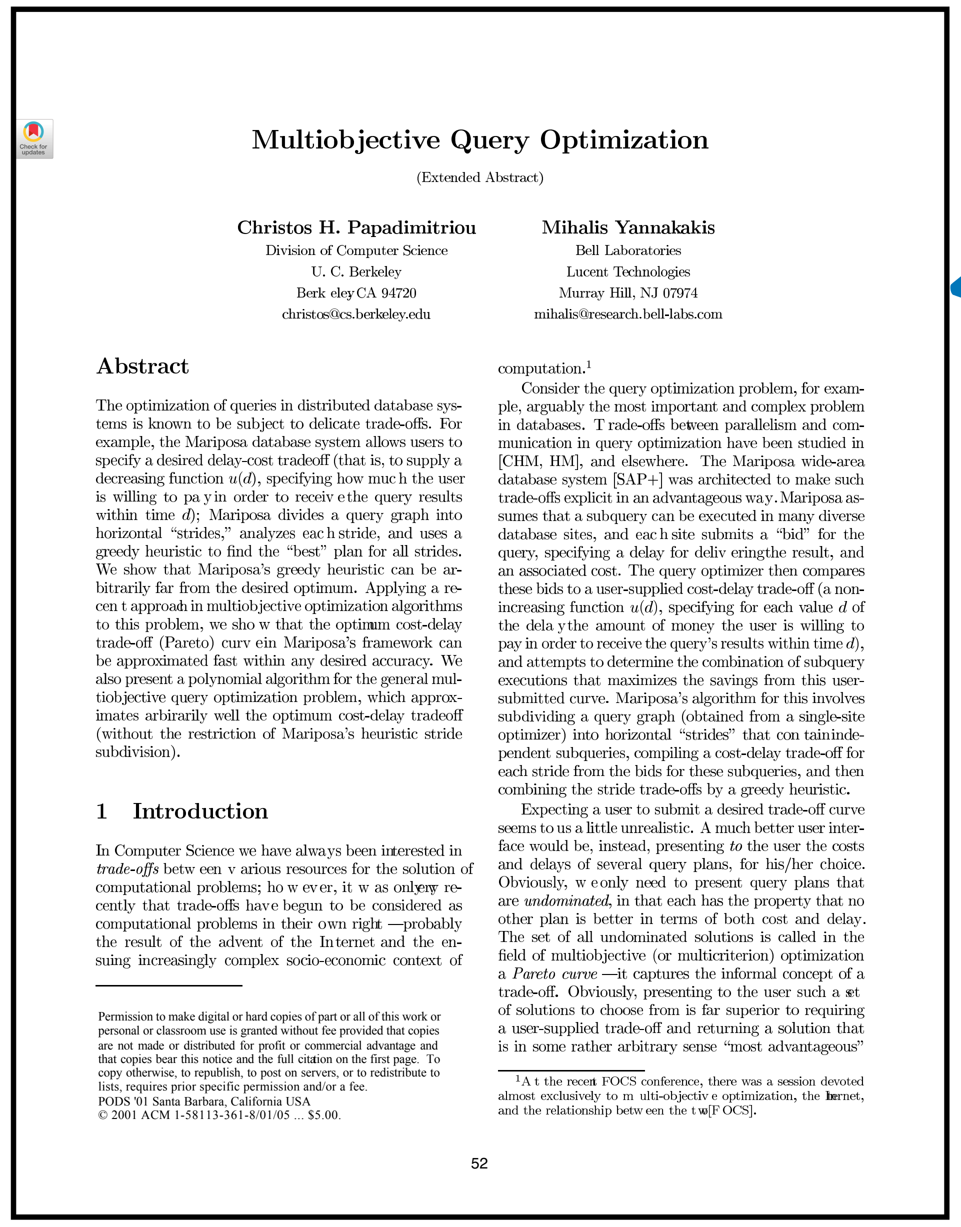
[1]At the recent FOCS conference, there was a session devoted almost exclusively to multi-objective optimization, the Internet, and the relationship between the two [FOCS].

Problem Instance
User Preferences



Quantum Computation Plan
(Methods & Parameters)

"Query Optimization" for Quantum Computing



Multiobjective Query Optimization

(Extended Abstract)

Christos H. Papadimitriou
Division of Computer Science
U. C. Berkeley
Berkeley CA 94720
christos@cs.berkeley.edu

Mihalis Yannakakis
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974
mihalis@research.bell-labs.com

Abstract

The optimization of queries in distributed database systems is known to be subject to delicate trade-offs. For example, the Mariposa database system allows users to specify a desired delay-cost tradeoff (that is, to supply a decreasing function $u(d)$, specifying how much the user is willing to pay in order to receive the query results within time d); Mariposa divides a query graph into horizontal "strides," analyzes each stride, and uses a greedy heuristic to find the "best" plan for all strides. We show that Mariposa's greedy heuristic can be arbitrarily far from the desired optimum. Applying a recent approach in multiobjective optimization algorithms to this problem, we show that the optimum cost-delay trade-off (Pareto) curve in Mariposa's framework can be approximated fast within any desired accuracy. We also present a polynomial algorithm for the general multiobjective query optimization problem, which approximates arbirarily well the optimum cost-delay tradeoff (without the restriction of Mariposa's heuristic stride subdivision).

1 Introduction

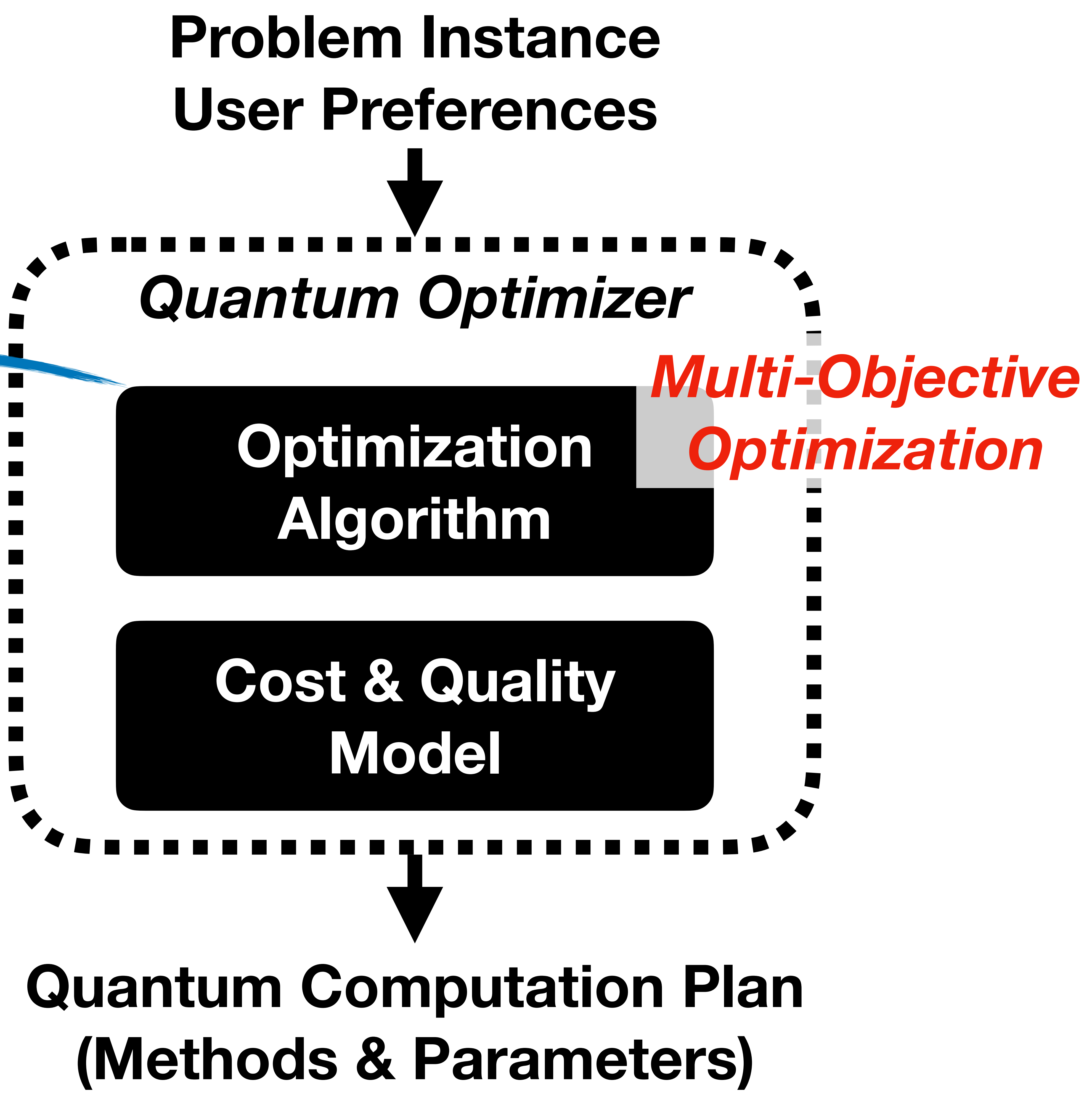
In Computer Science we have always been interested in *trade-offs* between various resources for the solution of computational problems; however, it was only very recently that trade-offs have begun to be considered as computational problems in their own right —probably the result of the advent of the Internet and the ensuing increasingly complex socio-economic context of computation.[1]

Consider the query optimization problem, for example, arguably the most important and complex problem in databases. Trade-offs between parallelism and communication in query optimization have been studied in [CHM, HM], and elsewhere. The Mariposa wide-area database system [SAP+] was architected to make such trade-offs explicit in an advantageous way. Mariposa assumes that a subquery can be executed in many diverse database sites, and each site submits a "bid" for the query, specifying a delay for delivering the result, and an associated cost. The query optimizer then compares these bids to a user-supplied cost-delay trade-off (a non-increasing function $u(d)$, specifying for each value d of the delay the amount of money the user is willing to pay in order to receive the query's results within time d), and attempts to determine the combination of subquery executions that maximizes the savings from this user-submitted curve. Mariposa's algorithm for this involves subdividing a query graph (obtained from a single-site optimizer) into horizontal "strides" that contain independent subqueries, compiling a cost-delay trade-off for each stride from the bids for these subqueries, and then combining the stride trade-offs by a greedy heuristic.

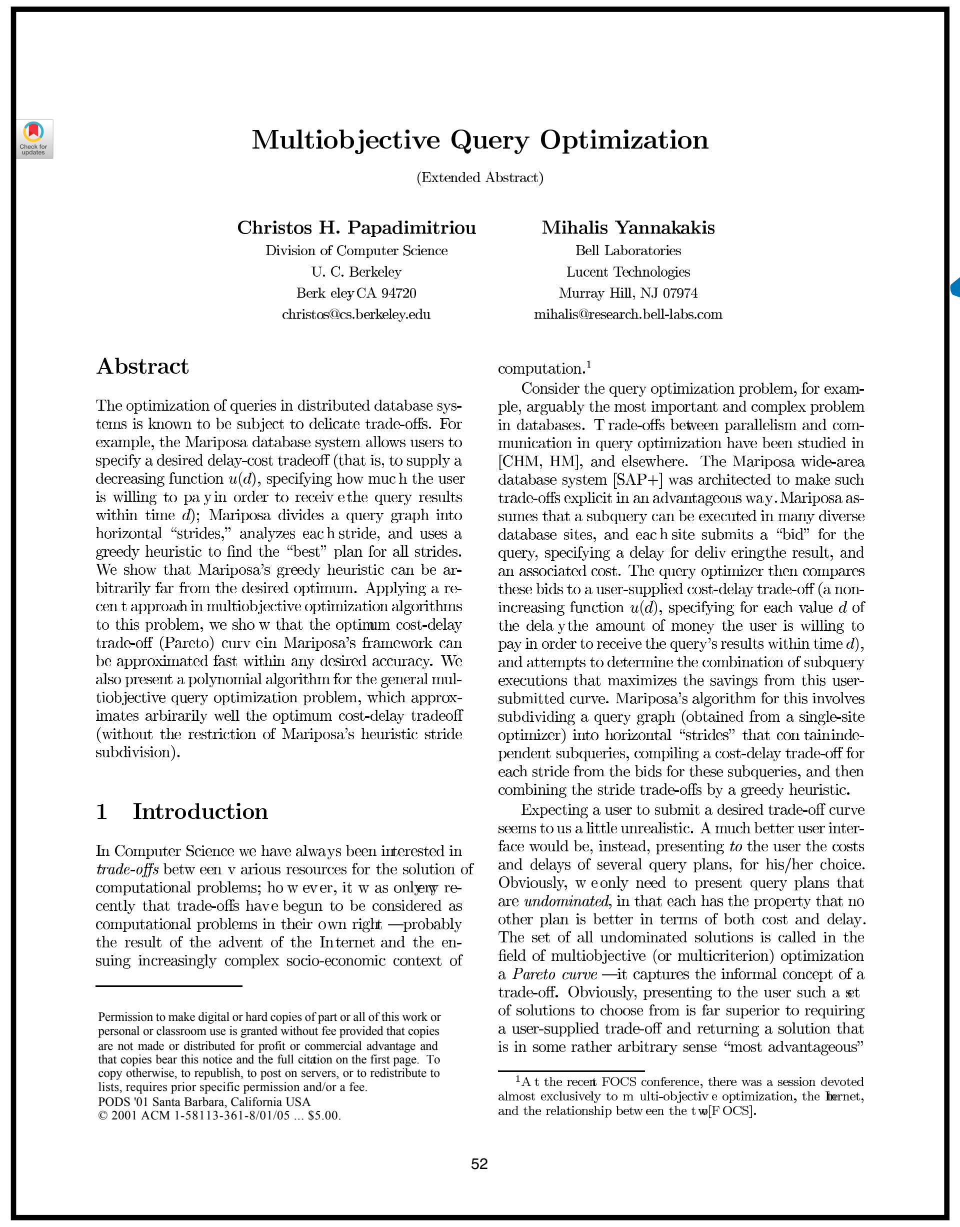
Expecting a user to submit a desired trade-off curve seems to us a little unrealistic. A much better user interface would be, instead, presenting *to* the user the costs and delays of several query plans, for his/her choice. Obviously, we only need to present query plans that are *undominated*, in that each has the property that no other plan is better in terms of both cost and delay. The set of all undominated solutions is called in the field of multiobjective (or multicriterion) optimization a *Pareto curve* —it captures the informal concept of a trade-off. Obviously, presenting to the user such a set of solutions to choose from is far superior to requiring a user-supplied trade-off and returning a solution that is in some rather arbitrary sense "most advantageous"

[1] At the recent FOCS conference, there was a session devoted almost exclusively to multi-objective optimization, the Internet, and the relationship between the two [FOCS].

Approximation Algorithms



"Query Optimization" for Quantum Computing



Multiobjective Query Optimization

(Extended Abstract)

Christos H. Papadimitriou
Division of Computer Science
U. C. Berkeley
Berkeley CA 94720
christos@cs.berkeley.edu

Mihalis Yannakakis
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974
mihalis@research.bell-labs.com

Abstract

The optimization of queries in distributed database systems is known to be subject to delicate trade-offs. For example, the Mariposa database system allows users to specify a desired delay-cost tradeoff (that is, to supply a decreasing function $u(d)$, specifying how much the user is willing to pay in order to receive the query results within time d); Mariposa divides a query graph into horizontal "strides," analyzes each stride, and uses a greedy heuristic to find the "best" plan for all strides. We show that Mariposa's greedy heuristic can be arbitrarily far from the desired optimum. Applying a recent approach in multiobjective optimization algorithms to this problem, we show that the optimum cost-delay trade-off (Pareto) curve in Mariposa's framework can be approximated fast within any desired accuracy. We also present a polynomial algorithm for the general multiobjective query optimization problem, which approximates arbirarily well the optimum cost-delay tradeoff (without the restriction of Mariposa's heuristic stride subdivision).

1 Introduction

In Computer Science we have always been interested in *trade-offs* between various resources for the solution of computational problems; however, it was only very recently that trade-offs have begun to be considered as computational problems in their own right —probably the result of the advent of the Internet and the ensuing increasingly complex socio-economic context of computation.[1]

Consider the query optimization problem, for example, arguably the most important and complex problem in databases. Trade-offs between parallelism and communication in query optimization have been studied in [CHM, HM], and elsewhere. The Mariposa wide-area database system [SAP+] was architected to make such trade-offs explicit in an advantageous way. Mariposa assumes that a subquery can be executed in many diverse database sites, and each site submits a "bid" for the query, specifying a delay for delivering the result, and an associated cost. The query optimizer then compares these bids to a user-supplied cost-delay trade-off (a non-increasing function $u(d)$, specifying for each value d of the delay the amount of money the user is willing to pay in order to receive the query's results within time d), and attempts to determine the combination of subquery executions that maximizes the savings from this user-submitted curve. Mariposa's algorithm for this involves subdividing a query graph (obtained from a single-site optimizer) into horizontal "strides" that contain independent subqueries, compiling a cost-delay trade-off for each stride from the bids for these subqueries, and then combining the stride trade-offs by a greedy heuristic.

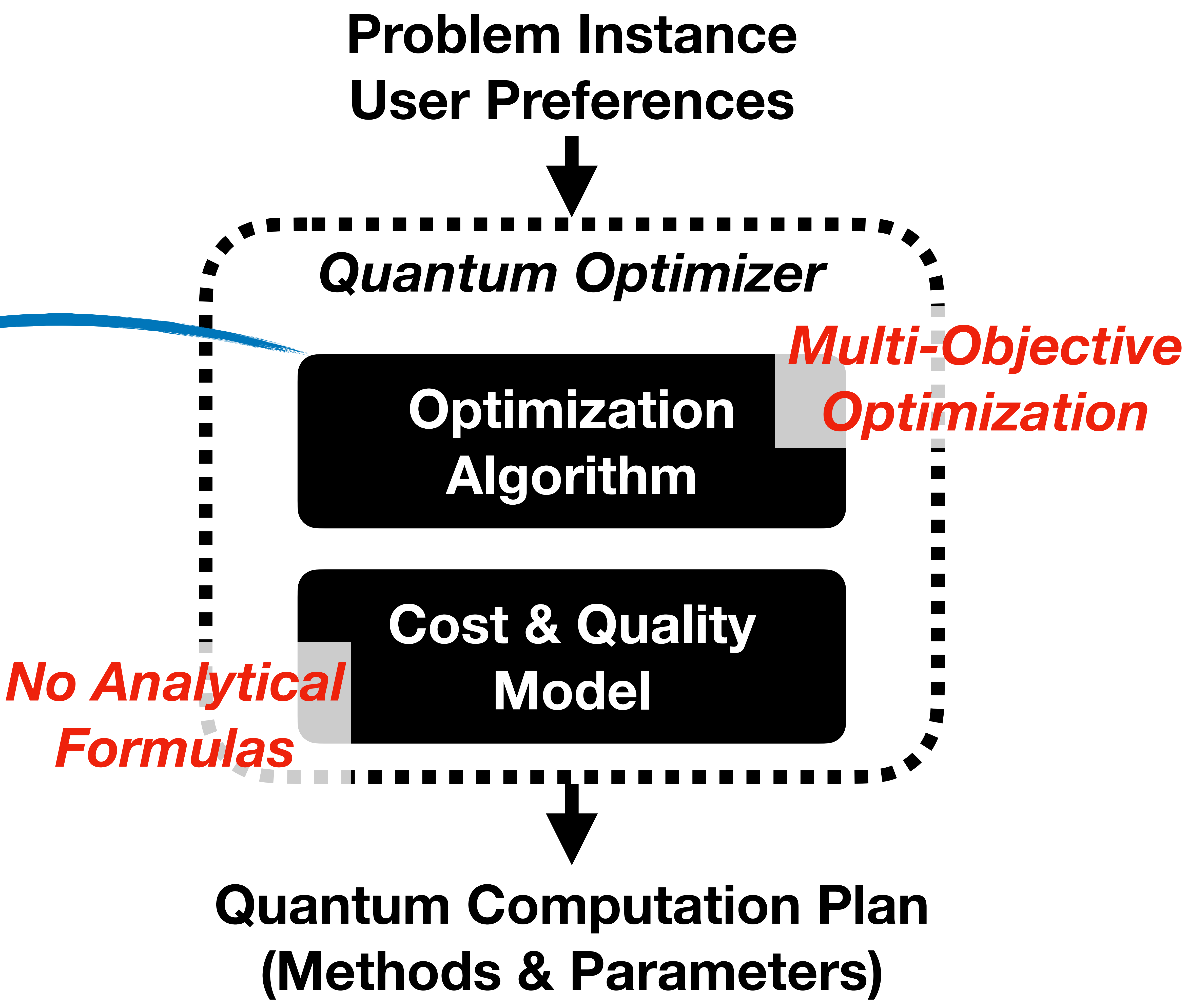
Expecting a user to submit a desired trade-off curve seems to us a little unrealistic. A much better user interface would be, instead, presenting *to* the user the costs and delays of several query plans, for his/her choice. Obviously, we only need to present query plans that are *undominated*, in that each has the property that no other plan is better in terms of both cost and delay. The set of all undominated solutions is called in the field of multiobjective (or multicriterion) optimization a *Pareto curve* —it captures the informal concept of a trade-off. Obviously, presenting to the user such a set of solutions to choose from is far superior to requiring a user-supplied trade-off and returning a solution that is in some rather arbitrary sense "most advantageous"

[1] At the recent FOCS conference, there was a session devoted almost exclusively to multi-objective optimization, the Internet, and the relationship between the two [FOCS].

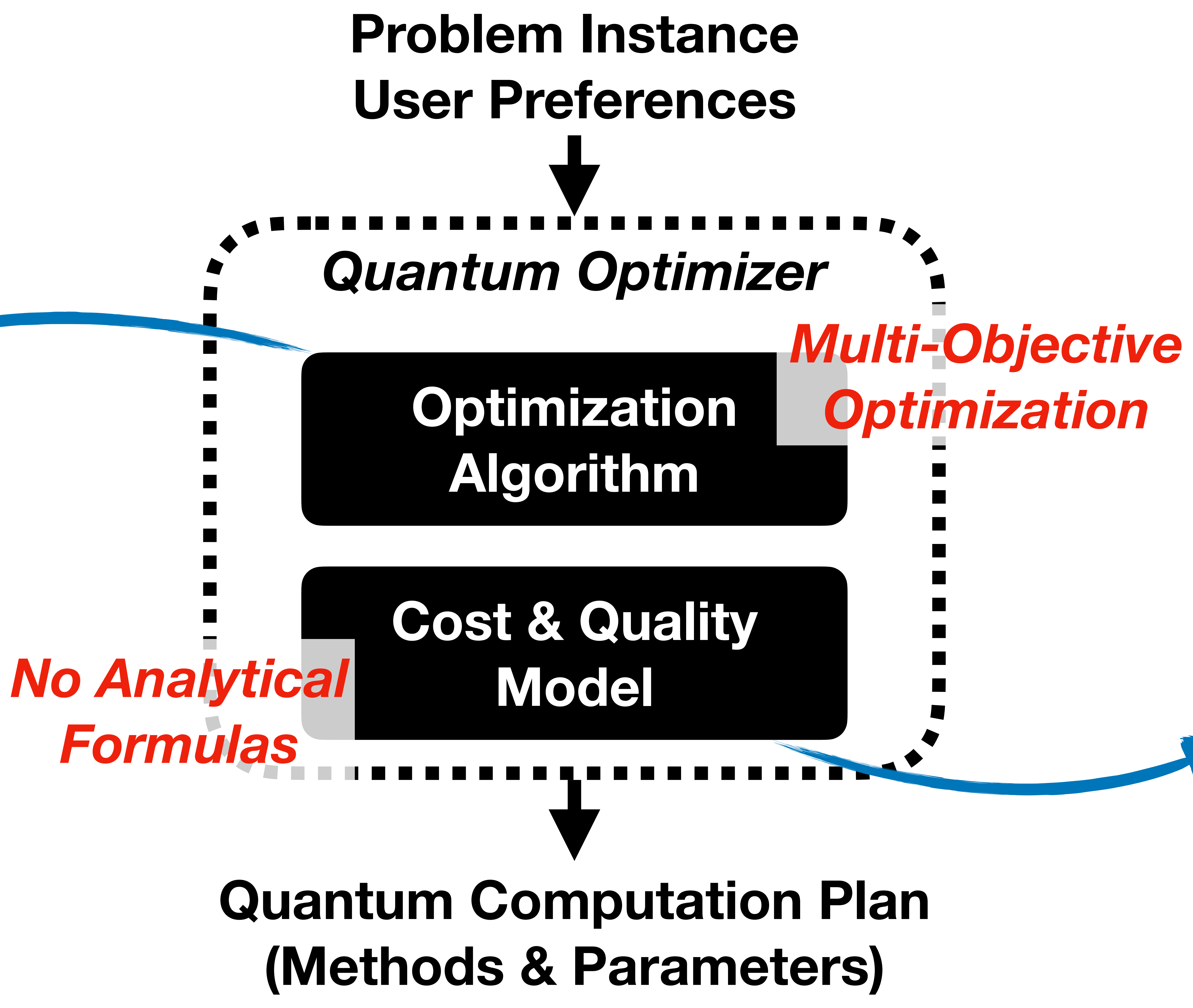
Permission to make digital or hard copies of part or all of this work or personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.
PODS '01 Santa Barbara, California USA
© 2001 ACM 1-58113-361-8/01/05 ... $5.00.

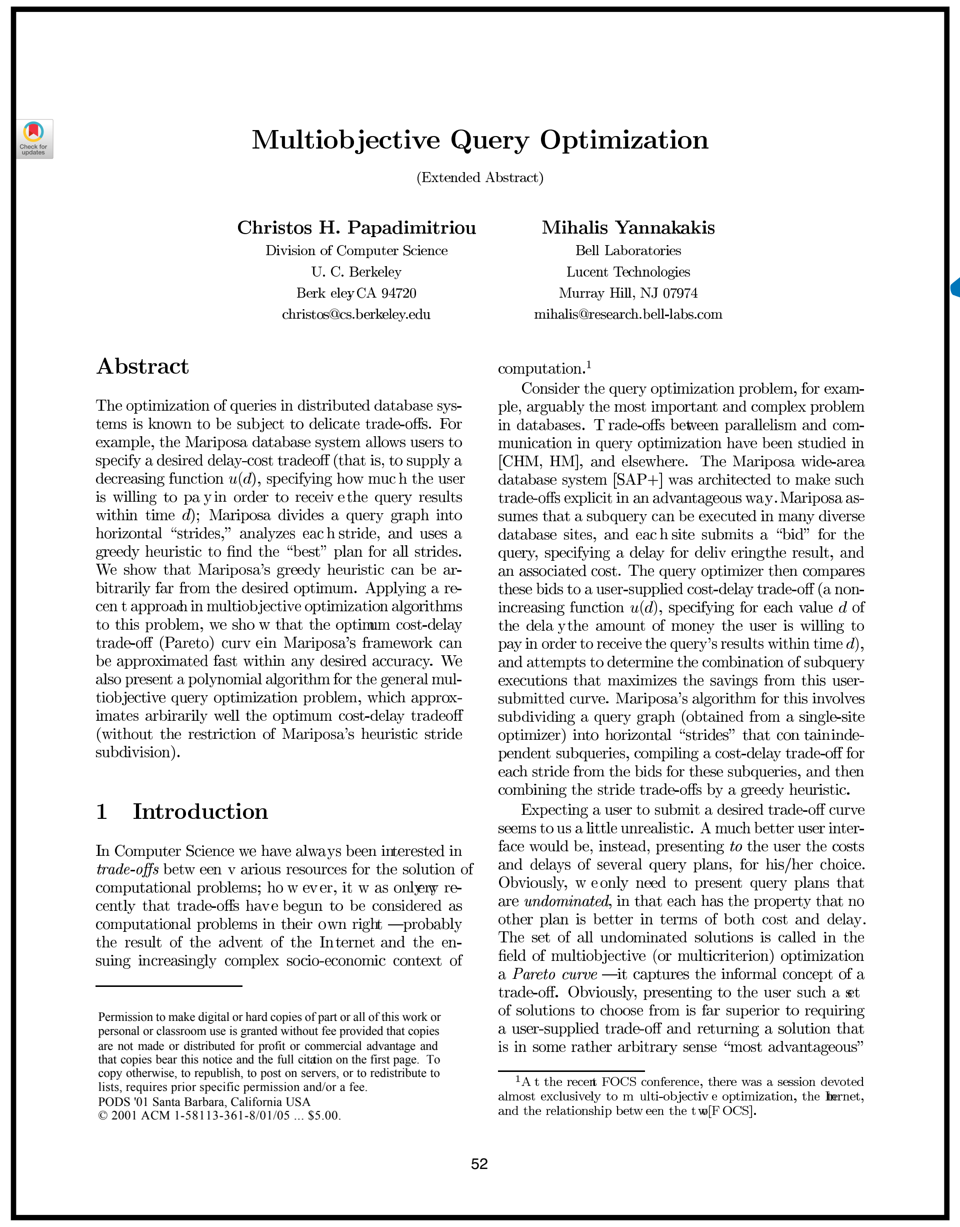
52

Approximation Algorithms



"Query Optimization" for Quantum Computing





Multiobjective Query Optimization

(Extended Abstract)

Christos H. Papadimitriou
Division of Computer Science
U. C. Berkeley
Berk eley CA 94720
christos@cs.berkeley.edu

Mihalis Yannakakis
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974
mihalis@research.bell-labs.com

Abstract

The optimization of queries in distributed database systems is known to be subject to delicate trade-offs. For example, the Mariposa database system allows users to specify a desired delay-cost tradeoff (that is, to supply a decreasing function $u(d)$, specifying how muc h the user is willing to pa y in order to receiv e the query results within time d); Mariposa divides a query graph into horizontal "strides," analyzes eac h stride, and uses a greedy heuristic to find the "best" plan for all strides. We show that Mariposa's greedy heuristic can be arbitrarily far from the desired optimum. Applying a recen t approach in multiobjective optimization algorithms to this problem, we sho w that the optimum cost-delay trade-off (Pareto) curv e in Mariposa's framework can be approximated fast within any desired accuracy. We also present a polynomial algorithm for the general multiobjective query optimization problem, which approximates arbirarily well the optimum cost-delay tradeoff (without the restriction of Mariposa's heuristic stride subdivision).

1 Introduction

In Computer Science we have always been interested in *trade-offs* betw een v arious resources for the solution of computational problems; ho w ev er, it w as only very recently that trade-offs have begun to be considered as computational problems in their own right —probably the result of the advent of the In ternet and the ensuing increasingly complex socio-economic context of computation.[1]

Consider the query optimization problem, for example, arguably the most important and complex problem in databases. T rade-offs between parallelism and communication in query optimization have been studied in [CHM, HM], and elsewhere. The Mariposa wide-area database system [SAP+] was architected to make such trade-offs explicit in an advantageous way. Mariposa assumes that a subquery can be executed in many diverse database sites, and eac h site submits a "bid" for the query, specifying a delay for deliv ering the result, and an associated cost. The query optimizer then compares these bids to a user-supplied cost-delay trade-off (a non-increasing function $u(d)$, specifying for each value d of the dela y the amount of money the user is willing to pay in order to receive the query's results within time d), and attempts to determine the combination of subquery executions that maximizes the savings from this user-submitted curve. Mariposa's algorithm for this involves subdividing a query graph (obtained from a single-site optimizer) into horizontal "strides" that con tain independent subqueries, compiling a cost-delay trade-off for each stride from the bids for these subqueries, and then combining the stride trade-offs by a greedy heuristic.

Expecting a user to submit a desired trade-off curve seems to us a little unrealistic. A much better user interface would be, instead, presenting *to* the user the costs and delays of several query plans, for his/her choice. Obviously, w e only need to present query plans that are *undominated*, in that each has the property that no other plan is better in terms of both cost and delay. The set of all undominated solutions is called in the field of multiobjective (or multicriterion) optimization a *Pareto curve* —it captures the informal concept of a trade-off. Obviously, presenting to the user such a set of solutions to choose from is far superior to requiring a user-supplied trade-off and returning a solution that is in some rather arbitrary sense "most advantageous"

[1] A t the recent FOCS conference, there was a session devoted almost exclusively to m ulti-objectiv e optimization, the Internet, and the relationship betw een the t wo [F OCS].

Approximation Algorithms

Neo: A Learned Query Optimizer

Ryan Marcus[1], Parimarjan Negi[2], Hongzi Mao[2], Chi Zhang[1],
Mohammad Alizadeh[2], Tim Kraska[2], Olga Papaemmanouil[1], Nesime Tatbul[23]
[1]Brandeis University [2]MIT [3]Intel Labs
[1]{ryan, chi, olga}@cs.brandeis.edu [2]{pnegi, hongzi, alizadeh, kraska, tatbul}@mit.edu

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

PVLDB Reference Format:
Ryan Marcus, Parimarjan Negi, Hongzi Mao, Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil, Nesime Tatbul. Neo: A Learned Query Optimizer. *PVLDB*, 12(11): 1705-1718, 2019.
DOI: https://doi.org/10.14778/3342263.3342644

1. INTRODUCTION

In the face of a deluge of machine learning success stories, every database researcher has likely wondered if it is possible to *learn* a query optimizer. Query optimizers are key to achieving good performance in database systems, and can speed up query execution by orders of magnitude. However, building a good optimizer today takes thousands of person-engineering-hours, and is an art only a few experts fully master. Even worse, query optimizers need to be tediously maintained, especially as the system's execution and storage engines evolve. As a result, none of the freely available open-source query optimizers come close to the performance of commercial optimizers offered by IBM, Oracle, or Microsoft.

Due to the heuristic-based nature of query optimization, there have been many attempts to apply learning to query optimizers. For example, almost two decades ago, Leo, DB2's LEarning Optimizer, was proposed [53]. Leo learns from its mistakes by adjusting its cardinality estimations over time. However, Leo still requires a human-engineered cost model, a hand-picked search strategy, and a lot of developer-tuned heuristics. Importantly, Leo only improves its cardinality estimation model, and cannot further optimize its search strategy based on data (e.g., to account for uncertainty in cardinality estimates for join order selection).

More recently, the database community has started to explore how neural networks can be used to improve query optimizers [36, 60]. The majority of this work has focused on replacing a component of the optimizer with learned models. For example, DQ [25] and ReJOIN [35] use reinforcement learning combined with traditional human-engineered cost models to automatically learn search strategies and explore the space of possible join orderings. These papers show that learned search strategies can outperform conventional heuristics on a given cost model. Moreover, in addition to the cost model, these systems still rely on heuristics for cardinality estimation, physical operator selection, and index selection.

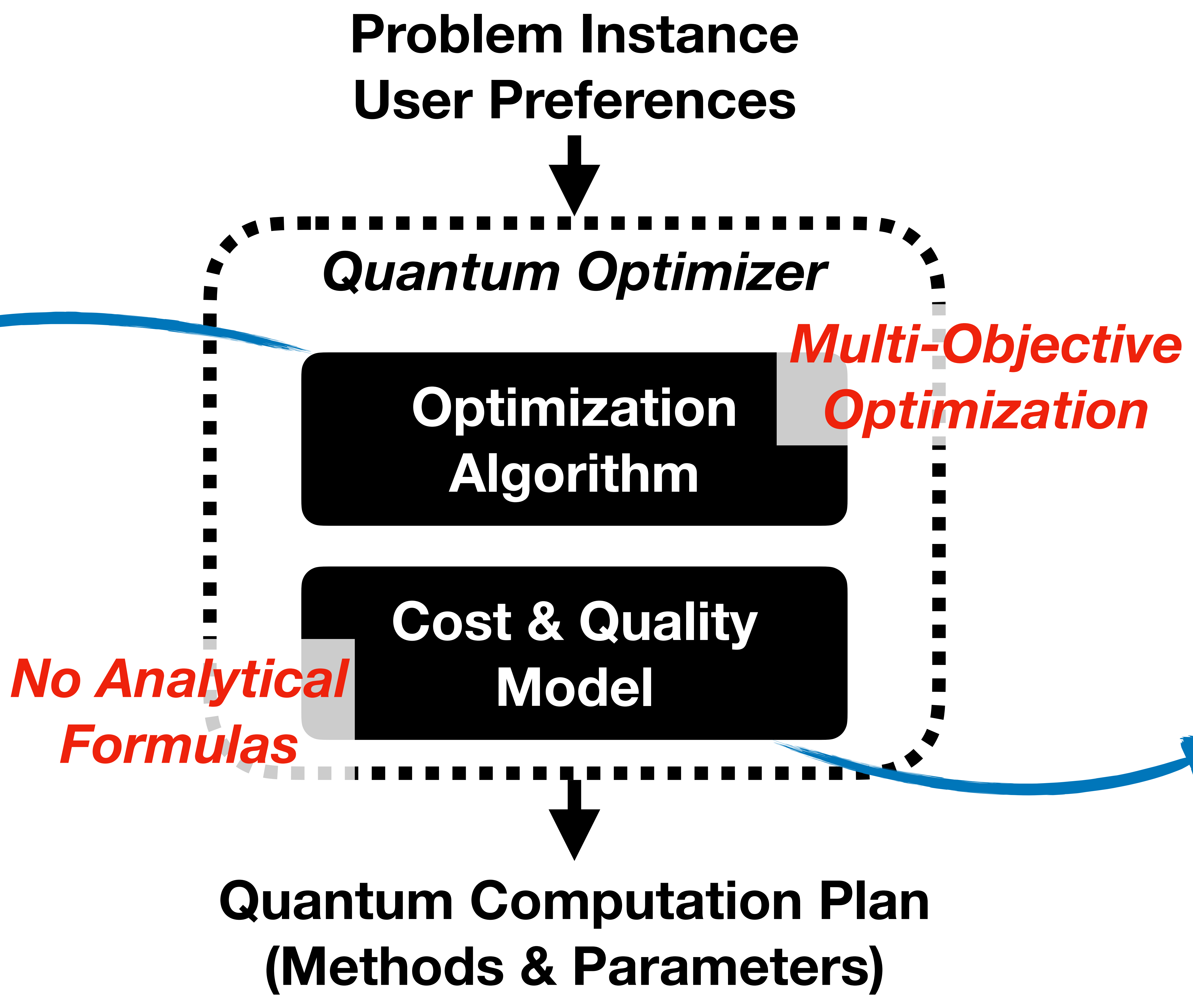
Other approaches demonstrate how machine learning can be used to achieve better cardinality estimates [22, 28, 43, 44]. However, none demonstrate that their improved cardinality estimations *actually lead to better query plans.* It is relatively easy to improve the average error of cardinality estimates, but much harder to improve estimations for the cases that actually improve query plans [27]. Furthermore, unlike join order selection, selecting join operators (e.g., hash join, merge join) and choosing indexes cannot be entirely reduced to cardinality estimation. SkinnerDB [56], showed that adaptive query processing strategies can benefit from reinforcement learning, but it requires a specialized (adaptive) query execution engine and cannot benefit from operator pipelining.

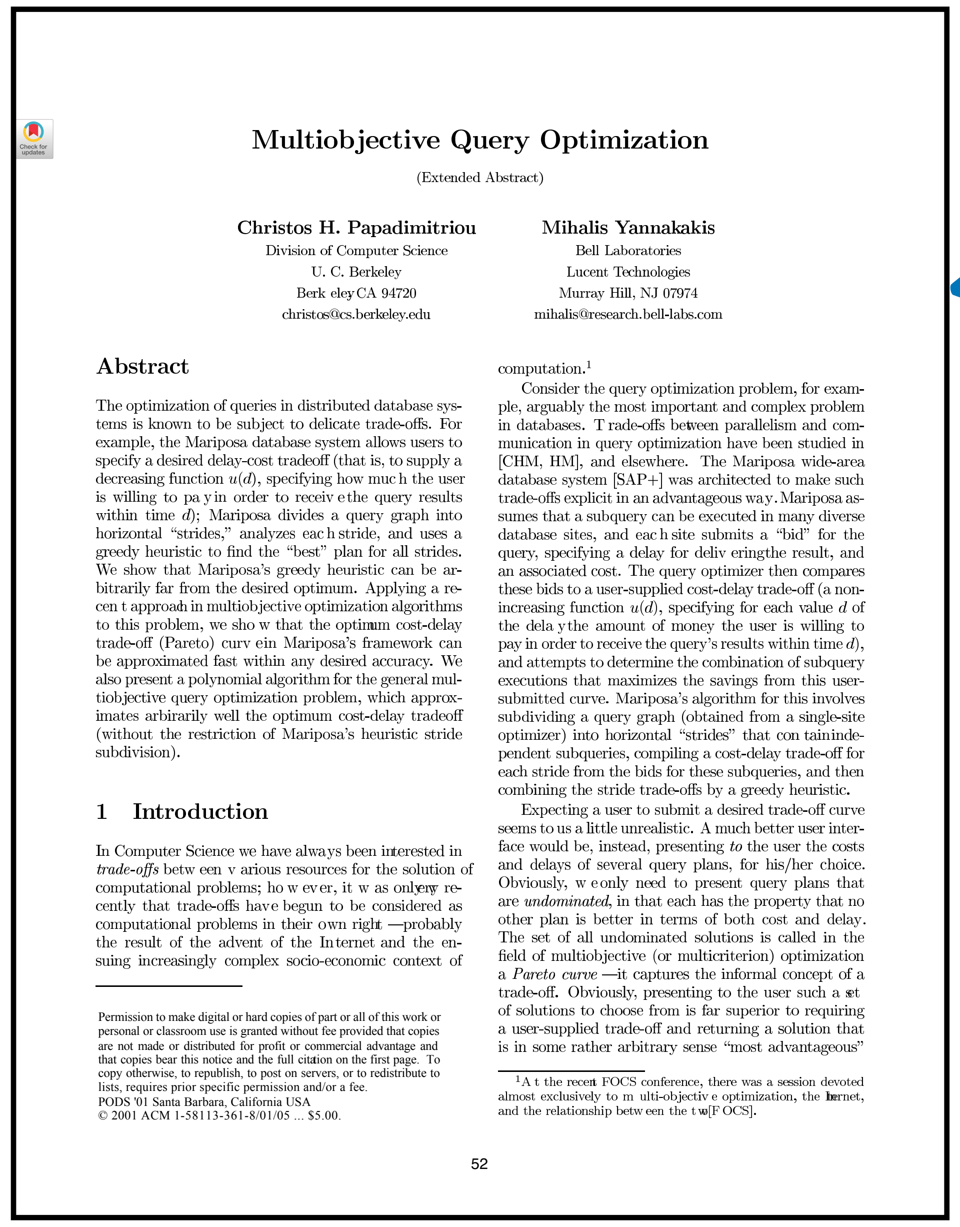
In this paper, we present *Neo* (*Neural Optimizer*), a learned query optimizer that achieves similar or improved performance compared to state-of-the-art commercial optimizers (Oracle and Microsoft) *on their own query execution engines*. Given a set of query rewrite rules to ensure semantic correctness, Neo learns to make decisions about join order, operator, and index selection. Neo optimizes these decisions using reinforcement learning, tailoring itself to the user's database instance and basing its decision on actual query latency.

Neo's design blurs the boundaries between the main components of a traditional query optimizer: cardinality estimation, the cost model, and the plan search algorithm. Neo does not explicitly estimate cardinalities or rely on hand-crafted cost models. Neo combines these two functions in a *value network*, a neural network that takes a partial query plan and *predicts* the best expected runtime that could result from completing this partial plan. Guided by the value network, Neo performs a simple search over the query plan space to make decisions. As Neo discovers better query plans,

Proceedings of the VLDB Endowment, Vol. 12, No. 11
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3342263.3342644

"Query Optimization" for Quantum Computing





Multiobjective Query Optimization

(Extended Abstract)

Christos H. Papadimitriou
Division of Computer Science
U. C. Berkeley
Berk eley CA 94720
christos@cs.berkeley.edu

Mihalis Yannakakis
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974
mihalis@research.bell-labs.com

Abstract

The optimization of queries in distributed database systems is known to be subject to delicate trade-offs. For example, the Mariposa database system allows users to specify a desired delay-cost tradeoff (that is, to supply a decreasing function $u(d)$, specifying how muc h the user is willing to pa y in order to receiv e the query results within time d); Mariposa divides a query graph into horizontal "strides," analyzes eac h stride, and uses a greedy heuristic to find the "best" plan for all strides. We show that Mariposa's greedy heuristic can be arbitrarily far from the desired optimum. Applying a recen t approach in multiobjective optimization algorithms to this problem, we sho w that the optimum cost-delay trade-off (Pareto) curv e in Mariposa's framework can be approximated fast within any desired accuracy. We also present a polynomial algorithm for the general multiobjective query optimization problem, which approximates arbirarily well the optimum cost-delay tradeoff (without the restriction of Mariposa's heuristic stride subdivision).

1 Introduction

In Computer Science we have always been interested in *trade-offs* betw een v arious resources for the solution of computational problems; ho w ev er, it w as only very recently that trade-offs have begun to be considered as computational problems in their own right —probably the result of the advent of the In ternet and the ensuing increasingly complex socio-economic context of computation.[1]

Consider the query optimization problem, for example, arguably the most important and complex problem in databases. T rade-offs between parallelism and communication in query optimization have been studied in [CHM, HM], and elsewhere. The Mariposa wide-area database system [SAP+] was architected to make such trade-offs explicit in an advantageous way. Mariposa assumes that a subquery can be executed in many diverse database sites, and eac h site submits a "bid" for the query, specifying a delay for deliv eringthe result, and an associated cost. The query optimizer then compares these bids to a user-supplied cost-delay trade-off (a non-increasing function $u(d)$, specifying for each value d of the dela y the amount of money the user is willing to pay in order to receive the query's results within time d), and attempts to determine the combination of subquery executions that maximizes the savings from this user-submitted curve. Mariposa's algorithm for this involves subdividing a query graph (obtained from a single-site optimizer) into horizontal "strides" that con tain independent subqueries, compiling a cost-delay trade-off for each stride from the bids for these subqueries, and then combining the stride trade-offs by a greedy heuristic.

Expecting a user to submit a desired trade-off curve seems to us a little unrealistic. A much better user interface would be, instead, presenting *to* the user the costs and delays of several query plans, for his/her choice. Obviously, w e only need to present query plans that are *undominated*, in that each has the property that no other plan is better in terms of both cost and delay. The set of all undominated solutions is called in the field of multiobjective (or multicriterion) optimization a *Pareto curve* —it captures the informal concept of a trade-off. Obviously, presenting to the user such a set of solutions to choose from is far superior to requiring a user-supplied trade-off and returning a solution that is in some rather arbitrary sense "most advantageous"

[1] A t the recent FOCS conference, there was a session devoted almost exclusively to m ulti-objectiv e optimization, the In ternet, and the relationship betw een the two [F OCS].

52

Approximation Algorithms

Neo: A Learned Query Optimizer

Ryan Marcus[1], Parimarjan Negi[2], Hongzi Mao[2], Chi Zhang[1],
Mohammad Alizadeh[2], Tim Kraska[2], Olga Papaemmanouil[1], Nesime Tatbul[23]
[1]Brandeis University [2]MIT [3]Intel Labs
[1]{ryan, chi, olga}@cs.brandeis.edu [2]{pnegi, hongzi, alizadeh, kraska, tatbul}@mit.edu

ABSTRACT

Query optimization is one of the most challenging problems in database systems. Despite the progress made over the past decades, query optimizers remain extremely complex components that require a great deal of hand-tuning for specific workloads and datasets. Motivated by this shortcoming and inspired by recent advances in applying machine learning to data management challenges, we introduce *Neo* (*Neural Optimizer*), a novel learning-based query optimizer that relies on deep neural networks to generate query executions plans. Neo bootstraps its query optimization model from existing optimizers and continues to learn from incoming queries, building upon its successes and learning from its failures. Furthermore, Neo naturally adapts to underlying data patterns and is robust to estimation errors. Experimental results demonstrate that Neo, even when bootstrapped from a simple optimizer like PostgreSQL, can learn a model that offers similar performance to state-of-the-art commercial optimizers, and in some cases even surpass them.

PVLDB Reference Format:
Ryan Marcus, Parimarjan Negi, Hongzi Mao, Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil, Nesime Tatbul. Neo: A Learned Query Optimizer. *PVLDB*, 12(11): 1705-1718, 2019.
DOI: https://doi.org/10.14778/3342263.3342644

1. INTRODUCTION

In the face of a deluge of machine learning success stories, every database researcher has likely wondered if it is possible to *learn* a query optimizer. Query optimizers are key to achieving good performance in database systems, and can speed up query execution by orders of magnitude. However, building a good optimizer today takes thousands of person-engineering-hours, and is an art only a few experts fully master. Even worse, query optimizers need to be tediously maintained, especially as the system's execution and storage engines evolve. As a result, none of the freely available open-source query optimizers come close to the performance of commercial optimizers offered by IBM, Oracle, or Microsoft.

Due to the heuristic-based nature of query optimization, there have been many attempts to apply learning to query optimizers. For example, almost two decades ago, Leo, DB2's LEarning Optimizer, was proposed [53]. Leo learns from its mistakes by adjusting its cardinality estimations over time. However, Leo still requires a human-engineered cost model, a hand-picked search strategy, and a lot of developer-tuned heuristics. Importantly, Leo only improves its cardinality estimation model, and cannot further optimize its search strategy based on data (e.g., to account for uncertainty in cardinality estimates for join order selection).

More recently, the database community has started to explore how neural networks can be used to improve query optimizers [36, 60]. The majority of this work has focused on replacing a component of the optimizer with learned models. For example, DQ [25] and ReJOIN [35] use reinforcement learning combined with traditional human-engineered cost models to automatically learn search strategies and explore the space of possible join orderings. These papers show that learned search strategies can outperform conventional heuristics on a given cost model. Moreover, in addition to the cost model, these systems still rely on heuristics for cardinality estimation, physical operator selection, and index selection.

Other approaches demonstrate how machine learning can be used to achieve better cardinality estimates [22, 28, 43, 44]. However, none demonstrate that their improved cardinality estimations *actually lead to better query plans.* It is relatively easy to improve the average error of cardinality estimates, but much harder to improve estimations for the cases that actually improve query plans [27]. Furthermore, unlike join order selection, selecting join operators (e.g., hash join, merge join) and choosing indexes cannot be entirely reduced to cardinality estimation. SkinnerDB [56], showed that adaptive query processing strategies can benefit from reinforcement learning, but it requires a specialized (adaptive) query execution engine and cannot benefit from operator pipelining.

In this paper, we present *Neo* (*Neural Optimizer*), a learned query optimizer that achieves similar or improved performance compared to state-of-the-art commercial optimizers (Oracle and Microsoft) *on their own query execution engines*. Given a set of query rewrite rules to ensure semantic correctness, Neo learns to make decisions about join order, operator, and index selection. Neo optimizes these decisions using reinforcement learning, tailoring itself to the user's database instance and basing its decision on actual query latency.

Neo's design blurs the boundaries between the main components of a traditional query optimizer: cardinality estimation, the cost model, and the plan search algorithm. Neo does not explicitly estimate cardinalities or rely on hand-crafted cost models. Neo combines these two functions in a *value network*, a neural network that takes a partial query plan and *predicts* the best expected runtime that could result from completing this partial plan. Guided by the value network, Neo performs a simple search over the query plan space to make decisions. As Neo discovers better query plans,

1705

Learned Optimization

Summary

- Use **query optimization ideas** for multi-step quantum computation
    - Approximate multi-objective optimization
    - Learned cost and quality models

?