scientific reports

Check for updates

OPEN

Power flow analysis using quantum and digital annealers: a discrete combinatorial optimization approach

Zeynab Kaseb[1]✉, Matthias Möller[2], Pedro P. Vergara[1] & Peter Palensky[1]

Power flow (PF) analysis is a foundational computational method to study the flow of power in an electrical network. This analysis involves solving a set of non-linear and non-convex differential-algebraic equations. State-of-the-art solvers for PF analysis, therefore, face challenges with scalability and convergence, specifically for large-scale and/or ill-conditioned cases characterized by high penetration of renewable energy sources, among others. The adiabatic quantum computing paradigm has been proven to efficiently find solutions for combinatorial problems in the noisy intermediate-scale quantum (NISQ) era, and it can potentially address the limitations posed by state-of-the-art PF solvers. For the first time, we propose a novel adiabatic quantum computing approach for efficient PF analysis. Our key contributions are (i) a combinatorial PF algorithm and a modified version that aligns with the principles of PF analysis, termed the adiabatic quantum PF algorithm (AQPF), both of which use Quadratic Unconstrained Binary Optimization (QUBO) and Ising model formulations; (ii) a scalability study of the AQPF algorithm; and (iii) an extension of the AQPF algorithm to handle larger problem sizes using a partitioned approach. Numerical experiments are conducted using different test system sizes on D-Wave's Advantage™ quantum annealer, Fujitsu's digital annealer V3, D-Wave's quantum-classical hybrid annealer, and two simulated annealers running on classical computer hardware. The reported results demonstrate the effectiveness and high accuracy of the proposed AQPF algorithm and its potential to speed up the PF analysis process while handling ill-conditioned cases using quantum and quantum-inspired algorithms.

Keywords Combinatorial power flow analysis, Quantum annealing, QUBO, Hubo, Power systems

Power flow (PF) analysis aims to calculate the complex voltage, i.e., the voltage magnitude and phase angle, of all buses in power systems for given loads, generations, and network topologies. The calculated voltage magnitudes and phase angles are subsequently used to estimate the power and current line flows and, eventually, facilitate the operation and planning of power systems. PF analysis involves solving a set of equations that relate the complex voltage and (active/reactive) power at each bus. From this perspective, a state-space representation of the power system that includes a set of differential-algebraic equations is used. These equations describe the active and reactive power balance at all the buses, as well as the voltage magnitude drop at all lines due to the power losses. The alternating current (AC) PF equations are examples of such equations and can be written as a set of nonlinear and non-convex equations in polar or rectangular coordinates, representing Kirchhoff's laws[1].

The AC PF equations are non-linear and non-convex, which rules out their exact analytical solution. Hence, the state-space representation of the power system is conventionally solved using iterative numerical methods, such as the Gauss-Seidel (GS) and Newton-Raphson (NR) methods[2]. These methods can provide steady-state solutions, i.e., voltages for all buses within specified accuracy boundaries[3,4]. However, the use of conventional PF analysis methods can cause convergence problems in large-scale modern power systems and is very time-consuming. For instance, GS has limitations regarding its dependence on the initial guess solution, accuracy level, ability to handle small network sizes, and difficulty converging for certain system conditions. Similarly, NR is limited by its inability to converge when the Jacobian matrix is singular, computational expense, and poor performance for ill-conditioned cases, such as highly unbalanced operation [5,6]. Therefore, conventional PF

[1]Electrical Sustainable Energy, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands. [2]Applied Mathematics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands. ✉email: Z.Kaseb@tudelft.nl

analysis approaches are becoming ineffective at supporting modern power system operation and planning [7–9]. From this perspective, inadequate PF analysis potentially leads to safety risks due to the inability of PF to accurately handle power generation from large numbers of distributed energy resources [10], as well as to predict blackouts. Hence, it is crucial to develop computationally efficient and numerically stable PF algorithms that can effectively handle the unique challenges of modern power systems[11].

Several advanced algorithms have been developed in the literature to address the challenges faced by PF analysis, including scalability[12–16] and the management of ill-conditioned cases[17–23]. These studies mostly focus on the advancement of programming techniques and the formulation of mathematical models grounded in classical computation principles. Efforts to address the scalability concern focus on reducing the computational challenges for PF analysis of large-scale modern power systems. For example, Lopez et al.[12] developed an extended dynamic programming approach based on the optimality principle of the recursive Hamilton-Jacobi-Bellman equations for electrical distribution systems. The scalability of the proposed approach was demonstrated through experimentation with large-scale power systems and benchmarking against commercial programming solvers. Goncalves et al.[13] formulated a mixed integer linear programming model for the operation planning problem of electrical distribution systems. The proposed formulation leverages linear expressions to model steady-state operation and hence ensures convergence toward the optimal solution for large-scale power systems. Dai et al.[14] proposed a scalable algorithm employing a reduced modeling technique for solving the AC PF problem. Reformulating this problem as a zero-residual least-squares problem with consensus constraints, the authors used a Gauss-Newton-based inexact algorithm to improve scalability. Idema et al.[15] presented iterative linear solvers, which offer enhanced scalability, thus rendering them suitable for the evolving modern power systems. In[16], the authors proposed modifications to NR to address sparsity in large power systems. Specifically, they introduced a sparse PF and integrated parallelization and vectorization techniques to accelerate computations.

With respect to the management of ill-conditioned cases, conventional PF solvers, such as NR, often encounter numerical stability issues during convergence. Consequently, the literature offers robust alternatives to mitigate these challenges. For example, Tostado-Veliz et al.[17], introduced a novel robust PF technique using the current injection form of the PF equations. Similarly, Tripathy[20] applied K.M. Brown's method to solve the PF. This method is a variation of NR that incorporates Gaussian elimination and ensures the utilization of the most recent information at each iteration. The method is particularly effective for solving ill-conditioned non-linear algebraic equations. Tostado-Veliz et al.[21] also presented a four-stage algorithm to address ill-conditioned cases. The algorithm was integrated with an efficient paradigm and is particularly useful in solving systems with reactive constraints of generators. Liu et al.[22], on the other hand, took a different approach where the PF model is transformed into a hypothetical dynamic system by introducing a differential equation on the loading parameter. This innovation enables a non-iterative method based on differential transformation to solve the dynamic system effectively. Iwamoto[23] devise a PF calculation method, characterized by its simplicity, absence of mathematical approximations, and minimal additional computational overhead when incorporated into NR. The method, therefore, offers a pragmatic solution to address ill-conditioned cases.

Despite the significant progress made in PF analysis algorithms, there remains a critical need for scalable, computationally efficient, and numerically stable PF analysis algorithms to meet the growing requirements of modern power systems. In particular, high levels of distributed energy resources, variable loads, and bidirectional power flows require more precise and efficient PF analysis approaches. In addition, the increasing deployment of advanced control and optimization techniques, such as real-time monitoring and control, demand response, and distributed energy resource management, necessitates the development of more sophisticated PF analysis algorithms. Therefore, although state-of-the-art methods have shown improvements, they are not enough to address the needs of modern power systems[11,24]. Considering the next example, the use of NR in the Power Grid Model environment (https://github.com/PowerGridModel) for a 369-node network necessitates approximately 10 seconds to solve a scenario with a specific computer architecture. For a thousand scenarios, the time required would be 1000×10 seconds, or more than 2.7 hours, using the same computer architecture. This duration is not feasible for specific applications, such as stochastic optimal power flow[25], where a large number of calculations should be completed within a few minutes. Consequently, the power systems community is increasingly interested in exploring novel approaches, including quantum computing[26–28], which have shown promising results in various applications and are, therefore, expected to revolutionize the field of PF analysis in the near future[29].

Quantum computing models can be categorized into two key paradigms: (i) the gate-based quantum computing model (GQC) and (ii) the adiabatic quantum computing model (AQC). Optimism for a *quantum speedup* stems from the potential of quantum algorithms to take advantage of unique properties, e.g., superposition and entanglement, which allows computations with fewer operations than classical algorithms. An increasing number of studies, therefore, explore how quantum computing can address the complexities of modern power systems. Most studies implement the GQC paradigm, which can simulate specific computations using quantum gates and discrete time steps[30,31]. GQC offers flexibility in algorithm design and potential quantum speedup for specific problem instances, such as linear system solving. However, GQC algorithms, e.g., the Harrow-Hassidim-Lloyd (HHL) algorithm, demand a large number of qubits and quantum gates even for small-scale problems, which makes them susceptible to noise inherent in current quantum computers. This noise, along with challenges related to the condition number of the system and readout requirements, can significantly limit the expected quantum speedup in power flow problems[32,33].

On the other hand, a few studies focus on the AQC paradigm[34,35]. AQC is similar to GQC in terms of computational power, but it operates in an analog manner, i.e., without quantum gates or discrete time steps[36,37]. The continuous evolution of AQC from an initial ground state to a final ground state of a problem Hamiltonian

makes it more robust to certain types of noise and, therefore, potentially offers a more practical approach to solving complex optimization problems in the near term[38]. For example, a comparative study between a GQC algorithm, i.e., Quantum Alternating Operator Ansatz (QAOA), and an AQC algorithm demonstrated that AQC outperformed GQC on specific hardware platforms[39]. D-Wave's quantum processing units (QPUs) powering the current generation of Advantage™ systems, as an AQC hardware example, serve as specialized accelerators for solving NP-hard optimization problems through a solution method called *quantum annealing*. These QPUs must be supercooled to temperatures close to absolute zero to maintain the quantum properties essential for computation. However, the key feature that enables efficient computation is their ability to control quantum mechanical effects for solving specific types of optimization problems[36]. Note that QPUs facilitate heuristic approaches to finding approximate solutions rather than solving NP-hard problems directly, and they do not guarantee a solution to all NP-hard problems.

The present work uses the AQC paradigm to address PF analysis. A combinatorial PF algorithm and an adiabatic quantum PF (AQPF) algorithm are developed and numerically justified. For the first time, two formulations are implemented based on the Quadratic Unconstrained Binary Optimization (QUBO) and the Ising model formulations for PF analysis. The scalability of the AQPF algorithm and of a partitioned variant is demonstrated numerically by applying them to various power system benchmarks. Experiments are performed for 4-bus, 9-bus, and 14-bus test systems on D-Wave's quantum annealer and Fujitsu's digital annealer, and two simulated annealers running on classical hardware. The results show the effectiveness and the significant potential of the AQPF algorithm in the NISQ and FTQ era. In addition, the proposed partitioned algorithm is introduced for the first time in the literature to the best of the authors' knowledge, which can speed up PF analysis in two different aspects: by leveraging quantum hardware and reducing the number of operations needed for a given test system.

Results

Power flow equations

For all buses i of a power system, the mismatch between the given active and reactive power consumption/injection at buses i, p_i^d and q_i^d, and their net counterparts, p_i and q_i, can be written as

$$p_i^d - p_i = 0 \tag{1}$$

$$q_i^d - q_i = 0 \tag{2}$$

with

$$p_i = \sum_{j=0}^{n} g_{ij}\left(\mu_i\mu_j + \omega_i\omega_j\right) + b_{ij}\left(\omega_i\mu_j - \mu_i\omega_j\right) \tag{3}$$

$$q_i = \sum_{j=0}^{n} g_{ij}\left(\omega_i\mu_j - \mu_i\omega_j\right) - b_{ij}\left(\mu_i\mu_j + \omega_i\omega_j\right) \tag{4}$$

where g_{ij} and b_{ij} are the known real and imaginary parts of the elements in the bus admittance matrix Y, respectively, and μ_i and ω_i are the real and imaginary parts of the complex voltages to be determined such that Eqs. (1)–(2) hold with equality for all buses i. To transform the problem at hand into a form that can be solved by a quantum or digital annealer, Eqs. (3)–(4) are expanded into

$$p_i = \sum_{j=0}^{n} \mu_i g_{ij}\mu_j + \omega_i g_{ij}\omega_j + \omega_i b_{ij}\mu_j - \mu_i b_{ij}\omega_j \tag{5}$$

$$q_i = \sum_{j=0}^{n} \omega_i g_{ij}\mu_j - \mu_i g_{ij}\omega_j - \mu_i b_{ij}\mu_j - \omega_i b_{ij}\omega_j \tag{6}$$

Here, μ_i, μ_j, ω_i, and ω_j are real-valued variables that need to be discretized in order to obtain a *binary* problem, i.e., a discrete combinatorial optimization problem. This work implements two different formulations, i.e., Quadratic Unconstrained Binary Optimization (QUBO) formulation and Ising model formulation, to solve such combinatorial PF.

Combinatorial power flow—QUBO formulation

μ_i and ω_i can be written using a straightforward discretization of the form

$$\mu_i = \mu_i^0 + x_{i,0}^{\mu}\Delta\mu_i - x_{i,1}^{\mu}\Delta\mu_i \tag{7}$$

$$\omega_i = \omega_i^0 + x_{i,0}^{\omega}\Delta\omega_i - x_{i,1}^{\omega}\Delta\omega_i \tag{8}$$

where $x_{i,\{0,1\}}^{\{\mu,\omega\}} \in \{0,1\}$ are *binary decision variables* whose value decides whether the base values μ_i^0 and ω_i^0 are increased $(x_{i,0} = 1 \wedge x_{i,1} = 0)$, decreased $(x_{i,0} = 0 \wedge x_{i,1} = 1)$, or kept at their current value

($x_{i,0} = 0 \wedge x_{i,1} = 0$ or $x_{i,0} = 1 \wedge x_{i,1} = 1$). The last case allows for an ambiguous solution, but at the same time, it ensures that all four combinations – 00, 01, 10, 11 – of the two binary variables $x_{i,0}$ and $x_{i,1}$ are valid bitstrings. In contrast to the often-used one-hot encoding, the above approach does not involve any constraints to rule out invalid bitstrings, which makes it particularly attractive for practical implementation on quantum and digital annealers. Nevertheless, a minor downside of the discretization of Eqs. (7)–(8) is that only three options ('increase', 'decrease', 'keep') are considered at a time, thereby not fully exploiting the full capability of two decision variables. Furthermore, the convergence of the iterative approach to be presented below can be slow if the increments $\Delta\mu_i$ and $\Delta\omega_i$ are chosen to be too small and/or if the base values μ_i^0 and ω_i^0 are far away from the values that satisfy Eqs. (1)–(2). This might be partly mitigated by choosing $\Delta\mu_i$ and $\Delta\omega_i$ adaptively, i.e., updating their values in each iteration loop and, potentially, even choosing individual values for each bus *i*. For net active power injection, substitution of Eqs. (7)–(8) into Eq. (5) yields

$$\begin{aligned}
&p_i = \\
&\#\text{constant term} \\
&\sum_{j=0}^{n} \mu_i^0 g_{ij} \mu_j^0 + \omega_i^0 g_{ij} \omega_j^0 + \omega_i^0 b_{ij} \mu_j^0 - \mu_i^0 b_{ij} \omega_j^0 \\
&\#\text{linearterm} \\
&+ \sum_{j=0}^{n} \sum_{k=0}^{1} (-1)^k \mu_i^0 g_{ij} x_{j,k}^{\mu} \Delta\mu_j + (-1)^k x_{i,k}^{\mu} \Delta\mu_i g_{ij} \mu_j^0 \\
&+ (-1)^k \omega_i^0 g_{ij} x_{j,k}^{\omega} \Delta\omega_j + (-1)^k x_{i,k}^{\omega} \Delta\omega_i g_{ij} \omega_j^0 \\
&+ (-1)^k \omega_i^0 b_{ij} x_{j,k}^{\mu} \Delta\mu_j + (-1)^k x_{i,k}^{\omega} \Delta\omega_i b_{ij} \mu_j^0 \\
&- (-1)^k \mu_i^0 b_{ij} x_{j,k}^{\omega} \Delta\omega_j - (-1)^k x_{i,k}^{\mu} \Delta\mu_i b_{ij} \omega_j^0 \\
&\#\text{quadratic term} \\
&+ \sum_{j=0}^{n} \sum_{k=0}^{1} \sum_{l=0}^{1} (-1)^{k+l} x_{i,k}^{\mu} \Delta\mu_i g_{ij} x_{j,l}^{\mu} \Delta\mu_j \\
&+ (-1)^{k+l} x_{i,k}^{\omega} \Delta\omega_i g_{ij} x_{j,l}^{\omega} \Delta\omega_j \\
&+ (-1)^{k+l} x_{i,k}^{\omega} \Delta\omega_i b_{ij} x_{j,l}^{\mu} \Delta\mu_j \\
&- (-1)^{k+l} x_{i,k}^{\mu} \Delta\mu_i b_{ij} x_{j,l}^{\omega} \Delta\omega_j
\end{aligned} \tag{9}$$

where the terms in the equation can be categorized as follows: constant terms, which do not contain any binary variables, e.g., $\mu_i^0 g_{ij} \mu_j^0$; linear terms, which include a single binary variable per term, e.g., $x_{i,k}^{\mu} \Delta\mu_i b_{ij} \omega_j^0$; and quadratic terms, which involve two binary variables per term, e.g., $x_{i,k}^{\mu} \Delta\mu_i b_{ij} x_{j,l}^{\omega} \Delta\omega_j$. Details on how deriving the equations are provided in the "Supplementary information". A similar expression can be derived for the net reactive power injection by substituting Eqs. (7)–(8) into Eq. (6)

$$\begin{aligned}
&q_i = \\
&\#\text{constant term} \\
&\sum_{j=0}^{n} \omega_i^0 g_{ij} \mu_j^0 - \mu_i^0 g_{ij} \omega_j^0 - \mu_i^0 b_{ij} \mu_j^0 - \omega_i^0 b_{ij} \omega_j^0 \\
&\#\text{linear term} \\
&+ \sum_{j=0}^{n} \sum_{k=0}^{1} (-1)^k \omega_i^0 g_{ij} x_{j,k}^{\mu} \Delta\mu_j + (-1)^k x_{i,k}^{\omega} \Delta\omega_i g_{ij} \mu_j^0 \\
&- (-1)^k \mu_i^0 g_{ij} x_{j,k}^{\omega} \Delta\omega_j - (-1)^k x_{i,k}^{\mu} \Delta\mu_i g_{ij} \omega_j^0 \\
&- (-1)^k \mu_i^0 b_{ij} x_{j,k}^{\mu} \Delta\mu_j - (-1)^k x_{i,k}^{\mu} \Delta\mu_i b_{ij} \mu_j^0 \\
&- (-1)^k \omega_i^0 b_{ij} x_{j,k}^{\omega} \Delta\omega_j - (-1)^k x_{i,k}^{\omega} \Delta\omega_i b_{ij} \omega_j^0 \\
&\#\text{quadratic term} \\
&+ \sum_{j=0}^{n} \sum_{k=0}^{1} \sum_{l=0}^{1} (-1)^{k+l} x_{i,k}^{\omega} \Delta\omega_i g_{ij} x_{j,l}^{\mu} \Delta\mu_j \\
&- (-1)^{k+l} x_{i,k}^{\mu} \Delta\mu_i g_{ij} x_{j,l}^{\omega} \Delta\omega_j \\
&- (-1)^{k+l} x_{i,k}^{\mu} \Delta\mu_i b_{ij} x_{j,l}^{\mu} \Delta\mu_j \\
&- (-1)^{k+l} x_{i,k}^{\omega} \Delta\omega_i b_{ij} x_{j,l}^{\omega} \Delta\omega_j
\end{aligned} \tag{10}$$

Combinatorial power flow— Ising model formulation

As an alternative to the above QUBO formulation, μ_i and ω_i can be represented using an alternative discretization of the form

$$\mu_i = \mu_i^0 + s_{i,0}^{\mu}\Delta\mu_i + s_{i,1}^{\mu}2\Delta\mu_i + s_{i,2}^{\mu}3\Delta\mu_i \tag{11}$$

$$\omega_i = \omega_i^0 + s_{i,0}^{\omega}\Delta\omega_i + s_{i,1}^{\omega}2\Delta\omega_i + s_{i,2}^{\omega}3\Delta\omega_i \tag{12}$$

where $s_{i,\{0,1,2\}}^{\{\omega,\mu\}} \in \{\pm 1\}$ are *spin variables*. As for the binary variables in Eqs. (7)–(8), any combination of spin-variable values yields an admissible solution in the range $\mu_i^0 \pm \{0,2,4,6\}\Delta\mu_i$ and $\omega_i^0 \pm \{0,2,4,6\}\Delta\omega_i$, which allows for more fine-grained control of the increment/decrement at the cost of just two additional variables per bus *i*. To prevent any confusion, we would like to note that the resulting Ising model is *not* equivalent to the previous QUBO formulation but an algorithmic alternative with different properties. In particular, adding an increment or decrement in the QUBO formulation requires introducing two additional binary variables, with one variable representing an increment and the other a decrement. In contrast, the Ising model inherently includes both positive and negative values in its spin variables and hence, needs only one spin variable to represent an increment or a decrement. Based on a sensitivity analysis performed in this work, it is recommended to rescale $\Delta\omega_i$ and $\Delta\mu_i$ by a factor of 3. For net active power injection, substitution of Eqs. (11)–(12) into Eq. (5) yields

$$\begin{aligned}
&p_i = \\
&\#\text{constant term} \\
&\sum_{j=0}^{n} \mu_i^0 g_{ij}\mu_j^0 + \omega_i^0 g_{ij}\omega_j^0 + \omega_i^0 b_{ij}\mu_j^0 - \mu_i^0 b_{ij}\omega_j^0 \\
&\#\text{linear term} \\
&+ \sum_{j=0}^{n}\sum_{k=0}^{2} (k+1)\mu_i^0 g_{ij} s_{j,k}^{\mu}\Delta\mu_j + (k+1)s_{i,k}^{\mu}\Delta\mu_i g_{ij}\mu_j^0 \\
&+ (k+1)\omega_i^0 g_{ij} s_{j,k}^{\omega}\Delta\omega_j + (k+1)s_{i,k}^{\omega}\Delta\omega_i g_{ij}\omega_j^0 \\
&+ (k+1)\omega_i^0 b_{ij} s_{j,k}^{\mu}\Delta\mu_j + (k+1)s_{i,k}^{\omega}\Delta\omega_i b_{ij}\mu_j^0 \\
&- (k+1)\mu_i^0 b_{ij} s_{j,k}^{\omega}\Delta\omega_j - (k+1)s_{i,k}^{\mu}\Delta\mu_i b_{ij}\omega_j^0 \\
&\#\text{quadratic term} \\
&+ \sum_{j=0}^{n}\sum_{k=0}^{2}\sum_{l=0}^{2} (k+1)(l+1)s_{i,k}^{\mu}\Delta\mu_i g_{ij} s_{j,l}^{\mu}\Delta\mu_j \\
&+ (k+1)(l+1)s_{i,k}^{\omega}\Delta\omega_i g_{ij} s_{j,l}^{\omega}\Delta\omega_j \\
&+ (k+1)(l+1)s_{i,k}^{\omega}\Delta\omega_i b_{ij} s_{j,l}^{\mu}\Delta\mu_j \\
&- (k+1)(l+1)s_{i,k}^{\mu}\Delta\mu_i b_{ij} s_{j,l}^{\omega}\Delta\omega_j
\end{aligned} \tag{13}$$

where the three clauses represent constant, linear, and quadratic terms, respectively. A similar expression can be derived for net reactive power injection by substituting Eqs. (11)–(12) into Eq. (6)

$$\begin{aligned}
&q_i = \\
&\#\text{constant term} \\
&\sum_{j=0}^{n} \omega_i^0 g_{ij}\mu_j^0 - \mu_i^0 g_{ij}\omega_j^0 - \mu_i^0 b_{ij}\mu_j^0 - \omega_i^0 b_{ij}\omega_j^0 \\
&\#\text{linear term} \\
&+ \sum_{j=0}^{n}\sum_{k=0}^{2} (k+1)\omega_i^0 g_{ij} s_{j,k}^{\mu}\Delta\mu_j + (k+1)s_{i,k}^{\omega}\Delta\omega_i g_{ij}\mu_j^0 \\
&- (k+1)\mu_i^0 g_{ij} s_{j,k}^{\omega}\Delta\omega_j - (k+1)s_{i,k}^{\mu}\Delta\mu_i g_{ij}\omega_j^0 \\
&- (k+1)\mu_i^0 b_{ij} s_{j,k}^{\mu}\Delta\mu_j - (k+1)s_{i,k}^{\mu}\Delta\mu_i b_{ij}\mu_j^0 \\
&- (k+1)\omega_i^0 b_{ij} s_{j,k}^{\omega}\Delta\omega_j - (k+1)s_{i,k}^{\omega}\Delta\omega_i b_{ij}\omega_j^0 \\
&\#\text{quadratic term} \\
&+ \sum_{j=0}^{n}\sum_{k=0}^{2}\sum_{l=0}^{2} (k+1)(l+1)s_{i,k}^{\omega}\Delta\omega_i g_{ij} s_{j,l}^{\mu}\Delta\mu_j \\
&- (k+1)(l+1)s_{i,k}^{\mu}\Delta\mu_i g_{ij} s_{j,l}^{\omega}\Delta\omega_j \\
&- (k+1)(l+1)s_{i,k}^{\mu}\Delta\mu_i b_{ij} s_{j,l}^{\mu}\Delta\mu_j \\
&- (k+1)(l+1)s_{i,k}^{\omega}\Delta\omega_i b_{ij} s_{j,l}^{\omega}\Delta\omega_j
\end{aligned} \tag{14}$$

Numerical justification of combinatorial power flow

Solving Eqs. (1)–(2) with the aid of a simulated, digital, or quantum annealer amounts to minimizing the sum of all the terms squared, i.e., the Hamiltonian $H(\vec{x})$ and $H(\vec{s})$

$$H(\cdot) = \sum_{i=0}^{n} \left(p_i^d - p_i\right)^2 + \left(q_i^d - q_i\right)^2. \tag{15}$$

Here, $\vec{x}$ and $\vec{s}$ represent vectors of binary and spin variables, respectively. Working out all the terms yields a fourth-order polynomial for the binary/spin variables. The solvers are D-Wave's simulated annealer Neal (https://docs.ocean.dwavesys.com/en/latest/docs_neal/sdk_index.html) (SA_1), D-Wave's quantum-classical hybrid annealer (https://docs.dwavesys.com/docs/latest/doc_leap_hybrid.html) (HA), D-Wave's Advantage™ system (www.dwavesys.com/learn/quantum-computing) (QA), Fujitsu's classical simulated annealer (SA_2), and Fujitsu's digital annealer V3 (www.fujitsu.com/global/services/business-services/digital-annealer) (DA). Since all solvers can handle only quadratic terms, higher-order terms need to be reduced by introducing auxiliary variables. The Python packages PyQUBO (https://pyqubo.readthedocs.io) and DADK are used to develop the QUBO and Ising model based on the Hamiltonian (15), which can perform the reduction of higher-order terms through the functions H.compile().to_qubo() and BinPol.reduce_higher_degree_to_qubo(H), respectively. PyQUBO is used when the solver is SA_1, HA, and QA, while DADK is used in combination with SA_2 and DA to reduce higher-order terms.

We observe differences between the two implementations, PyQUBO and DADK, in terms of the number of auxiliary variables required to encode the same problem sizes, with the DADK package often proving more efficient by requiring fewer auxiliary variables. However, in both implementations, we ensure that the order reduction process accurately preserves the optimal solutions and retains the same variable assignments for the lowest energy solutions. When performing order reduction, it is crucial to consider the specific objectives. In some scenarios, maintaining the entire energy spectrum ordering is important, especially when relative energy differences between states are significant. However, in many practical applications, including this study, the primary goal is to ensure that the optimal solutions remain unchanged after order reduction. References[40,41] provide a broader perspective on various order reduction techniques.

For all solvers, the number of readouts is the same and ranges from 2000 to 100,000, depending on the size of the test system. SA_1 approaches equilibrium by updating at decreasing temperatures until reaching the target temperature, that is 1/kT, where T is the temperature in kelvin and k is Boltzmann's constant. Spins are updated once per point per temperature using the Metropolis-Hastings algorithm. At low temperatures, the distribution focuses on the ground states, and long, smooth temperature schedules ensure that samples match the equilibrium distribution. For HA, the time limit parameter was set to 10 seconds. The chip ID of QA is *Advantage_system5.4*. The minor embedding method is used to map the problem onto the QA. Chain strength is determined by the square of the power consumption variables $\vec{p}^{\,l}$ and $\vec{q}^{\,l}$, as well as the line properties of the test systems, i.e., the admittance matrix *Y*. This ensures that the chain strength is sufficiently large to prevent chain breaks while remaining small enough to maintain the significance of the QUBO terms. The target graph for the embedding is sparse, reflecting the nature of the power test systems, which are not fully connected.

For SA_2 and DA, the time limit is 10 seconds, the scaling bit precision is 64, and the overall timeout is 3,600 seconds. In addition to simulated annealing, DA supports parallel tempering as an alternative optimization algorithm. Parallel tempering runs multiple replicas of the optimization process at different temperatures simultaneously. These replicas can exchange their temperatures with neighboring replicas based on a Metropolis criterion, which, in turn, facilitates a global search for the optimum. SA_2 and DA initiate optimization from a random state in each run, aiming to reach a configuration with the optimal energy value within the decision variable space.

Algorithm 1 represents the pseudo-code of the combinatorial PF algorithm. Initially, the vectors $\vec{p}^{\,l}$ and $\vec{q}^{\,l}$, respectively, representing the active and reactive power demands, and the admittance matrix *Y* are assigned based on the given test system power consumption and topology (lines 1-3). The increments/decrements $\Delta\mu$ and $\Delta\omega$, as well as the real and imaginary voltage vectors $\vec{\mu}$ and $\vec{\omega}$, are initialized with user-specified values (lines 4–7), based on which the initial active and reactive power vectors $\bar{p}$ and $\bar{q}$ are calculated (lines 8-9). Subsequently, the initial Hamiltonian $H(\vec{x})$ or $H(\vec{s})$ (cf. Eq. (15)) is calculated (line 10), providing an approximate solution based on the initial voltage estimates. The residual threshold ϵ is set in line 11, and the iteration counter *it* is initialized to zero before starting the loop.

Once the minimization problem with the Hamiltonian (15) has been solved (line 14), the resulting vectors $\vec{x} = \left[x_{0,0}^{\mu}, \ldots, x_{n,1}^{\omega}\right] \in \{0,1\}^{4n}$ for the QUBO formulation, and $\vec{s} = \left[s_{0,0}^{\mu}, \ldots, s_{n,2}^{\omega}\right] \in \{\pm 1\}^{6n}$ for the Ising model formulation can be used to update the real and imaginary parts of the voltages μ_i and ω_i according to Eqs. (7)–(8), and (11)–(12), respectively (lines 15 and 16). Note that auxiliary variables stemming from the conversion of higher-order terms into quadratic ones are neglected for the voltage updates. If the values of $H(\vec{x})$ or $H(\vec{s})$ drop below a certain user-defined threshold ϵ, the so-computed complex voltages $(\vec{\mu} + j\vec{\omega})$ are accepted as solution values. Otherwise, the complex voltage base values $\mu_i^0 := \mu_i$ and $\omega_i^0 := \omega_i$ are redefined, and the minimization problem is solved with the updated Hamiltonian (15) until convergence is reached. A secondary stopping criterion of the iteration schemes is when all $x_{i,0}^{\mu} = x_{i,1}^{\mu}$ and $x_{i,0}^{\omega} = x_{i,1}^{\omega}$ for the QUBO formulation, and $s_{i,0}^{\mu} + s_{i,1}^{\mu} = -s_{i,2}^{\mu}$ and $s_{i,0}^{\omega} + s_{i,1}^{\omega} = -s_{i,2}^{\omega}$ for the Ising model formulation, meaning that none of the base values μ_i^0 and ω_i^0 are updated and the iteration stalls. Note that this is very unlikely to happen in practice. Sporadic qubit flips most likely trigger at least some updates. Once convergence is reached, p_i and q_i are calculated for all buses *i* via Eqs. (3)–(4). Note that the iterative process in Algorithm 1 is essential for refining the solution to the power flow problem, as the initial solution may not meet the desired accuracy or might be a local minimum.

Initialize $\vec{p}^d = [p_0^d, p_1^d, \ldots, p_n^d]$
Initialize $\vec{q}^d = [q_0^d, q_1^d, \ldots, q_n^d]$
Initialize $Y = \{(g_{ij} + \mathrm{j}b_{ij}) : i, j = 0, 1, \ldots, n\}$
$\Delta\mu \leftarrow 1 \times 10^{-2}$
$\Delta\omega \leftarrow 1 \times 10^{-3}$
$\vec{\mu} = [\mu_0, \mu_1, \ldots, \mu_n] \leftarrow 1$
$\vec{\omega} = [\omega_0, \omega_1, \ldots, \omega_n] \leftarrow 0$
Calculate $\vec{p} = [p_0, p_1, \ldots, p_n]$ from equation (5)
Calculate $\vec{q} = [q_0, q_1, \ldots, q_n]$ from equation (6)
Calculate Hamiltonian H from equation (15)
$\varepsilon \leftarrow 1 \times 10^{-4}$
it $\leftarrow 0$
while $H > \varepsilon$ and it $<$ it$_{\max}$ **do**
    Minimize the QUBO/Ising model
    Update $\vec{\mu}$ from equations (7) or (11)
    Update $\vec{\omega}$ from equations (8) or (12)
    Recalculate $\vec{p}$ from equation (9) or (13)
    Recalculate $\vec{q}$ from equation (10) or (14)
    Recalculate H from equation (15)
    Update $\Delta\mu, \Delta\omega$
    it $\leftarrow$ it $+ 1$
end while

Algorithm 1. Combinatorial power flow algorithm.

Five different annealers, i.e., SA_1, HA, QA, SA_2, and DA, are used to minimize the QUBO/Ising model for which the computational details are presented in Table 1. Experiments are performed on a 4-bus test system with known p_i^d and q_i^d and unknown μ_i and ω_i for all buses i. This implementation results in 136 binary and 300 spin variables for the QUBO and Ising model formulations, respectively, using PyQUBO, and 104 binary and 236 spin variables for the QUBO and Ising model formulations, respectively, using DADK. The difference in variables stems from the fact that the two Python packages implement different strategies to convert higher-order terms into quadratic terms, with DADK adopting a more efficient approach. In the QUBO formulation, QA requires the most iterations to reach the defined threshold $\epsilon = 1 \times 10^{-3}$, while SA_1 takes the most iterations in the Ising model formulation. DA achieves ϵ with the fewest iterations in both the QUBO and Ising model formulations. The Ising model with QA is the most time-consuming per iteration due to the high communication time, including QPU access, delay, programming, readout, and sampling time. It is also important to note that different annealers reach different minimal energies with the same defined ϵ, as shown in Table 1.

Figure 1 illustrates $\vec{\mu} = [\mu_1, \mu_2, \mu_3]$ and $\vec{\omega} = [\omega_1, \omega_2, \omega_3]$ achieved by five annealers using the combinatorial PF algorithm for the 4-bus test system over iterations. Graphs (a-f) show results for the QUBO formulation, while graphs (g-l) depict results for the Ising model formulation. Note that the values of μ_0 and ω_0 are already known for the reference bus, set, respectively, to 1 and 0, and hence are not shown in the graphs. The μ_i and ω_i obtained from the Newton-Raphson classical solver (NR) are also included in the graphs. Figure 1a–c presents the convergence of μ_{1-3} using the QUBO formulation. It is observed that μ_2 converges more rapidly than μ_1 and μ_3, with the primary delay in the convergence of μ_1 and μ_3 being attributed to the slow convergence of QA. QA, in particular, exhibits slower convergence compared to the other annealers. Interestingly, although QA struggles with convergence speed, the values of ω_{1-3} generated by QA oscillate around the corresponding values obtained by NR. This suggests that while QA may eventually approach the correct solution, its path to convergence is less direct, particularly in the QUBO formulation. The slower convergence of μ_1 and μ_3 in the QUBO formulation, therefore, can be directly linked to the performance of QA. On the other hand, the Ising model formulation, depicted in Fig. 1g–l, shows a different trend. Here, SA_1 is the slowest to converge, as corroborated by Table 1. However, the convergence behavior in the Ising model is more uniform across $\mu 1 - 3$ and ω_{1-3}. Therefore, the

Formulation	Compiler	# of variables $[-]$	Compile time $[s]$	Annealer	# of iterations $[-]$	Time per iteration $[s]$	Hamiltonian $[W^2 + Var^2]$
QUBO	PyQUBO	136	0.08	SA_1	35	0.677	8.58×10^{-4}
	PyQUBO	136	0.08	HA	35	9.718	5.37×10^{-4}
	PyQUBO	136	0.08	QA	98	58.598	9.01×10^{-4}
	DADK	104	0.14	SA_2	34	0.16	7.55×10^{-4}
	DADK	104	8.66	DA	24	8.12	8.81×10^{-4}
Ising	PyQUBO	300	1.88	SA_1	69	1.942	9.05×10^{-4}
	PyQUBO	300	1.88	HA	52	10.602	9.42×10^{-4}
	PyQUBO	300	1.88	QA	44	230.2	6.97×10^{-4}
	DADK	236	1.03	SA_2	52	1.06	8.63×10^{-4}
	DADK	236	9.12	DA	23	9.38	8.50×10^{-4}

Table 1. Computational details for the combinatorial power flow algorithm with the QUBO and Ising model formulations based on the 4-bus test system.

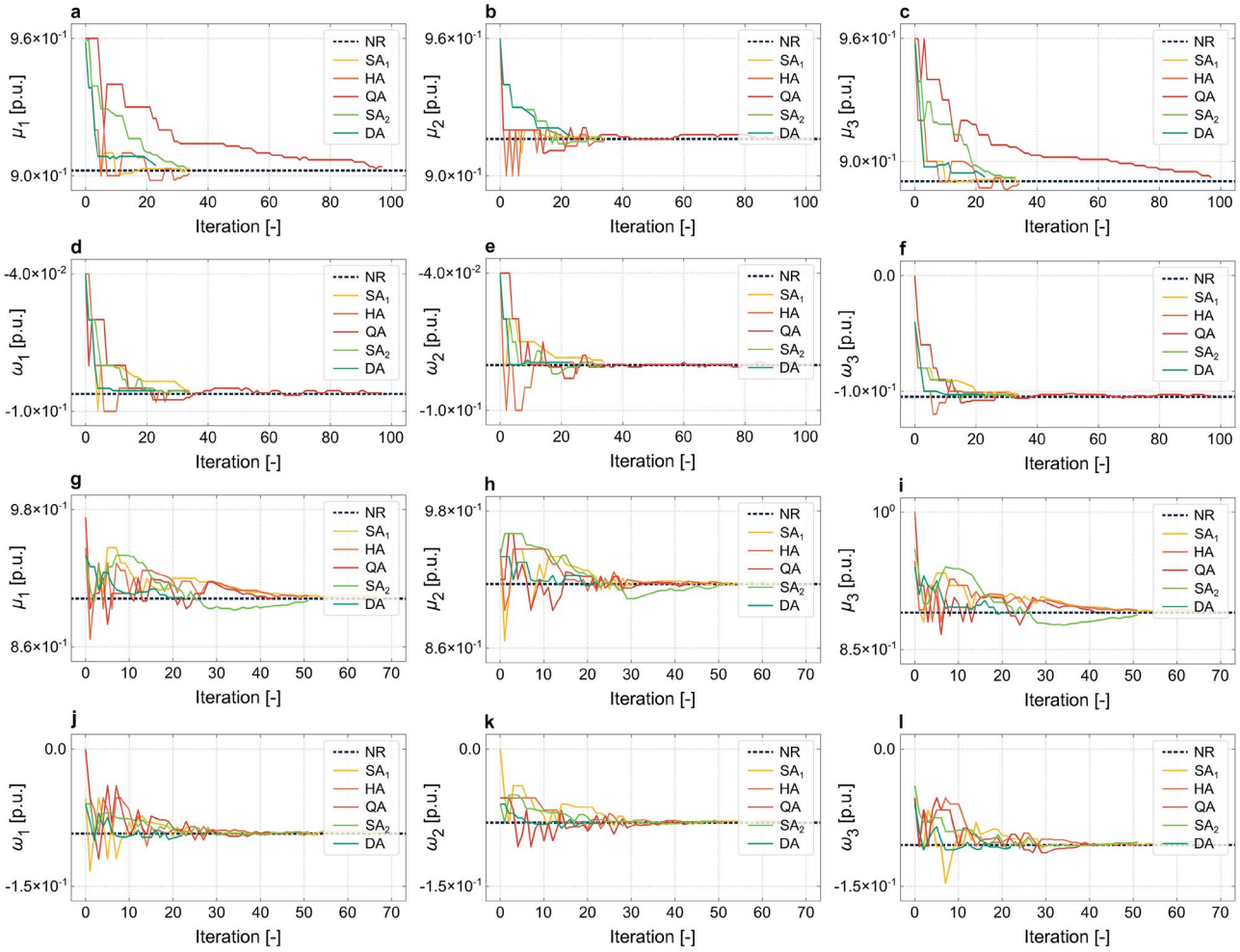

Figure 1. Representation of $\vec{\mu} = [\mu_1, \mu_2, \mu_3]$ and $\vec{\omega} = [\omega_1, \omega_2, \omega_3]$ obtained by the combinatorial power flow algorithm. Experiments are performed using D-Wave's classical simulated annealer Neal (SA_1), D-Wave's quantum-classical hybrid annealer (HA), D-Wave's Advantage™ system (QA), Fujitsu's classical simulated annealer (SA_2), and Fujitsu's digital annealer V3 (DA) using both (**a**–**f**) QUBO and (**g**–**l**) Ising model formulations for the 4-bus test system. The reference bus $i = 0$ is not shown. The graphs include μ_i and ω_i obtained from the Newton-Raphson classical solver (NR).

annealers, when working with the Ising model, exhibit a smoother and more consistent convergence trajectory due to the more fine-grained updates allowed by this formulation.

Numerical justification of adiabatic quantum power flow

In the combinatorial PF algorithm, we assume that the active and reactive powers are available for all buses $i = 0, \ldots, n$, which is inconsistent with the principles of PF analysis. For example, active and reactive powers at the reference bus are unknown, while the complex voltage $\mu + j\omega$ is known. To address this discrepancy, adjustments are necessary to Algorithm 1. Algorithm 2 represents the pseudo-code of the modified algorithm that aligns with the principles of PF analysis. Specifically, for a power system with one reference bus, the active and reactive power values for the reference bus $i = 0$ are excluded from the respective vectors $\vec{p}^{\,tl}$ and $\vec{q}^{\,tl}$, resulting in $\vec{p}^{\,tl} = \left[p_1^d, p_2^d, \ldots, p_n^d\right]$ and $\vec{q}^{\,tl} = \left[q_1^d, q_2^d, \ldots, q_n^d\right]$ (lines 1 and 2). In addition, $\mu_0 = 1$ and $\omega_0 = 0$ are assigned to the reference bus $i = 0$ before the loop starts, ensuring they are not updated during the loop (lines 4 and 5). The remaining of the algorithm is the same as Algorithm 1.

Initialize $\vec{p}^d = [p_1^d, p_2^d, \ldots, p_n^d]$
Initialize $\vec{q}^d = [q_1^d, q_2^d, \ldots, q_n^d]$
Initialize $Y = \{(g_{ij} + \mathrm{j}b_{ij}) : i, j = 0, 1, \ldots, n\}$
Initialize $\mu_0 = 1$
Initialize $\omega_0 = 0$
$\Delta\mu \leftarrow 1 \times 10^{-2}$
$\Delta\omega \leftarrow 1 \times 10^{-3}$
$\vec{\mu} = [\mu_1, \mu_2, \ldots, \mu_n] \leftarrow 1$
$\vec{\omega} = [\omega_1, \omega_2, \ldots, \omega_n] \leftarrow 0$
Calculate $\vec{p} = [p_0, p_1, \ldots, p_n]$ from equation (5)
Calculate $\vec{q} = [q_0, q_1, \ldots, q_n]$ from equation (6)
Calculate Hamiltonian H from equation (15)
$\varepsilon \leftarrow 1 \times 10^{-4}$
it $\leftarrow 0$
while $H > \varepsilon$ and it $<$ it$_{\max}$ **do**
    Minimize the QUBO/Ising model
    Update $\vec{\mu}$ from equations (7) or (11)
    Update $\vec{\omega}$ from equations (8) or (12)
    Recalculate $\vec{p}$ from equation (9) or (13)
    Recalculate $\vec{q}$ from equation (10) or (14)
    Recalculate H from equation (15)
    Update $\Delta\mu, \Delta\omega$
    it $\leftarrow$ it $+ 1$
end while

Algorithm 2. Adiabatic power flow algorithm.

D-Wave's Advantage™ system (QA) and Fujitsu's digital annealer V3 (DA) are used to minimize the QUBO/Ising model developed based on the Hamiltonian at each iteration until convergence is reached. Experiments are performed for the 4-bus test system with known p_i^d and q_i^d and unknown μ_i and ω_i for all load buses $i = 1, 2, 3$, while $\mu_0 = 1$ and $\omega_0 = 0$ are known and p_0^d and q_0^d are unknown for the reference bus $i = 0$. Note that for the AQPF algorithm, the number of binary and spin variables is the same as in the combinatorial PF algorithm. The effectiveness of the AQPF algorithm is justified by comparing $\vec{\mu} = [\mu_1, \mu_2, \mu_3]$ and $\vec{\omega} = [\omega_1, \omega_2, \omega_3]$ obtained by QA and DA with those obtained from NR for the 4-bus test system. The results are detailed in Table 2. It is observed that the Ising model formulation with QA achieves results closest to those derived from NR, followed by the QUBO formulation with QA, for $\epsilon = 1 \times 10^{-3}$. A systematic comparison between the performance of the AQPF algorithm and NR over iterations is provided in the "Supplementary information".

Numerical justification of scalability of AQPF

The scalability of the AQPF algorithm, defined as the ability to handle a larger number of binary/spin variables, is justified through experiments conducted on a larger power system, specifically a 9-bus test system, using the same five annealers as before. In this implementation, PyQUBO generates 666 binary and 1485 spin variables for the QUBO and Ising model, respectively, while DADK generates 249 binary and 900 spin variables for the QUBO and Ising model, respectively. The threshold value for considering the iteration process converged is set to $\epsilon = 1 \times 10^{-3}$.

Figure 2 illustrates the vectors $\vec{\mu} = [\mu_0, \mu_1, \ldots, \mu_8]$ and $\vec{\omega} = [\omega_0, \omega_1, \ldots, \omega_8]$ obtained from SA_1, HA, QA, SA_2, and DA using both the (a and b) QUBO and (c and d) Ising model formulations. Additionally, the graphs include μ_i and ω_i obtained from the Newton-Raphson classical solver (NR). Accordingly, the AQPF algorithm shows scalability, yet using the Ising model with it for the 9-bus test system requires managing more variables than does the use of the QUBO formulation. This presents challenges for embedding in QA due to the partial connectivity of qubits in D-Wave's Advantage™ system. In addition, SA_1 fails to converge for the Ising model formulation. However, HA, SA_2, and DA effectively manage the increased number of spin variables for the 9-bus test system. Note that the computational time per iteration on the quantum hardware, independent of the communication overhead, is substantially less than the microsecond scale. This observation indicates the prospective capability of the AQPF algorithm to address PF challenges, i.e., scalability and handling ill-conditioned cases, in large

	μ_1	μ_2	μ_3	ω_1	ω_2	ω_3
NR	0.902	0.916	0.890	− 0.092	− 0.080	− 0.104
QA (QUBO)	0.904	0.917	0.892	− 0.092	− 0.079	− 0.104
DA (QUBO)	0.904	0.918	0.895	− 0.091	− 0.079	− 0.103
QA (Ising)	0.902	0.916	0.891	− 0.092	− 0.081	− 0.104
DA (Ising)	0.903	0.917	0.891	− 0.094	− 0.080	− 0.107

Table 2. Performance comparison of D-Wave's Advantage™ system (QA) and Fujitsu's digital annealer V3 (DA) with the Newton-Raphson classical solver (NR). The AQPF algorithm is implemented using the QUBO and Ising model formulations for the 4-bus test system. The reference bus $i = 0$ with known values of $\mu_0 = 1$ and $\omega_0 = 0$ is not shown.

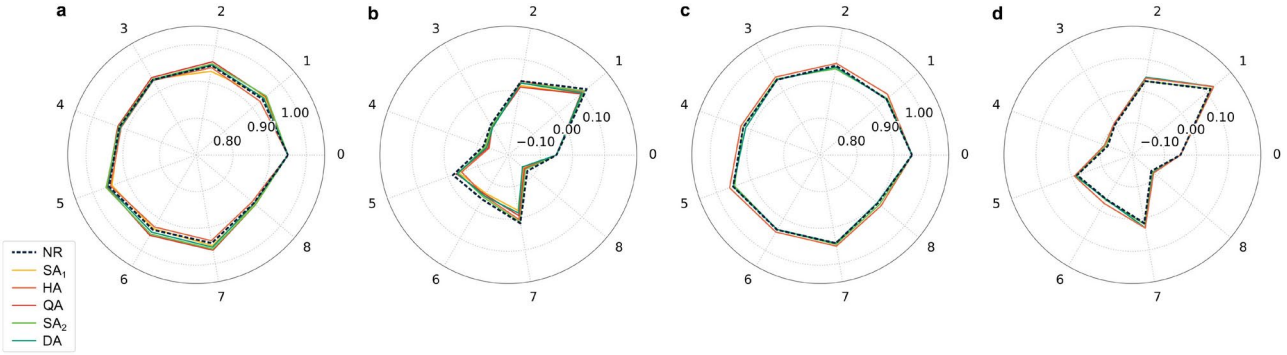

Fig. 2. Representation of (a and c) $\vec{\mu} = [\mu_0, \mu_1, \ldots, \mu_8]$ and (b and d) $\vec{\omega} = [\omega_0, \omega_1, \ldots, \omega_8]$ obtained by the AQPF algorithm. Experiments are performed using D-Wave's classical simulated annealer (SA_1), D-Wave's quantum-classical hybrid annealer (HA), D-Wave's Advantage™ system (QA), Fujitsu's classical simulated annealer (SA_2), and Fujitsu's digital annealer V3 (DA) using both (a and b) QUBO and (c and d) Ising model formulations for the 9-bus test system. The graphs include $\vec{\mu}$ and $\vec{\omega}$ obtained from the Newton-Raphson classical solver (NR).

power systems within a microsecond timeframe, provided that quantum hardware accessibility resembles that of today's classical hardware. However, realizing this potential is impeded by the constraints inherent to current quantum hardware implementations.

Numerical justification of a partitioned AQPF

A heuristic search algorithm is developed and integrated into the AQPF algorithm to provide valid solutions for specific parts of the QUBO/Ising model formulation by excluding one or more buses from the formulations. This is especially important for large power systems as increasing the number of buses exponentially increases the number of binary/spin variables in the QUBO/Ising model formulation. The heuristic search algorithm is employed before using the resulting bitstring $\left[x^{\mu}_{0,0}, \ldots, x^{\omega}_{n,1}\right] \in \{0,1\}^{4n}$ for the QUBO formulation, and $\left[s^{\mu}_{0,0}, \ldots, s^{\omega}_{n,2}\right] \in \{\pm 1\}^{6n}$ for the Ising model formulation to update the real and imaginary parts of the complex voltages, i.e., μ_i and ω_i. The proposed heuristic search algorithm randomly flips one or more bits of the bitstring multiple times to prevent convergence to local optima and ensure a heuristic search of the solution space. The partitioned AQPF algorithm, therefore, offers a distinctive advantage by enabling PF analysis in cases where data is incomplete or inaccurate or computational resources are inadequate.

The algorithm is explored and justified using 4-bus and 14-bus test systems. The experiments are conducted using DA. For the 4-bus test system, bus $i = 3$ is left out of Hamiltonian (15) based on the QUBO formulation (9)–(10), i.e.,

$$H^{4\text{ - bus}}_{\text{part}}(\cdot) = \sum_{\substack{i=0 \\ i \neq 3}}^{n} \left(p^d_i - p_i\right)^2 + \left(q^d_i - q_i\right)^2. \tag{16}$$

The resulting values with $\epsilon = 1 \times 10^{-3}$ are $\vec{\mu} = [1, 0.902, 0.916, 0.891]$ and $\vec{\omega} = [0, -0.092, -0.079, -0.102]$. Note that by excluding bus $i = 3$, the number of binary variables decreases by up to 44%, generating only 58 binary variables using DADK. For the 14-bus test system, three buses, namely, $i = 7, 9, 10$ are left out of Hamiltonian (15) leading to

$$H^{14\text{ - bus}}_{\text{part}}(\cdot) = \sum_{\substack{i=0 \\ i \neq 7,9,10}}^{n} \left(p^d_i - p_i\right)^2 + \left(q^d_i - q_i\right)^2. \tag{17}$$

This implementation results in a total of 392 binary variables, reducing their count by 102 (up to 20%) compared to the AQPF algorithm. For the 14-bus test system, the mean difference between p_i obtained from the partitioned AQPF algorithm and p^d_i obtained from NR is $3.64 \times 10^{-3}[W]$, with a standard deviation of $1.57 \times 10^{-2}[W]$. Similarly, the mean difference between q_i obtained from the partitioned AQPF algorithm and q^d_i obtained from NR is $1.67 \times 10^{-3}[Var]$, with a standard deviation of $1.42 \times 10^{-2}[Var]$. Details on the results obtained by the 14-bus test system are provided in the "Supplementary information".

Discussion

This work introduces adiabatic quantum power flow (AQPF) for the first time in the literature, which leverages reliable quantum annealers[42–44]. Hence, the AQPF algorithm proposed is appropriate for noisy-intermediate-

scale quantum (NISQ) and fault-tolerant quantum (FTQ) hardware as the adiabatic quantum computing model offers more fault tolerance than does the gate-based quantum computing model (GQC) in the NISQ era[36]. Fujitsu's Digital Annealer, with the capacity to handle up to 100,000 decision variables, can test larger power systems compared to state-of-the-art (gate-based) quantum algorithms for power flow (PF) analysis, which can hardly handle test systems with more than five buses for the time being.

The AQPF algorithm adheres to the principles of the adiabatic quantum computing model (AQC), where the system evolves slowly enough to remain in its ground state throughout the computation. This gradual evolution mirrors how the AQPF algorithm iteratively refines voltage estimates to reduce the PF mismatch and ensure steady convergence to the optimal solution. It is important to note that while AQC offers a promising alternative to GQC for certain applications, it also has limitations, particularly in the context of PF analysis. AQC is inherently more suited for discrete optimization problems, whereas PF analysis is a continuous problem. Additionally, the performance of AQC can be highly dependent on problem encoding and the specific characteristics of the quantum hardware. Therefore, further research is needed to identify scenarios where AQC can provide a practical quantum advantage in PF analysis.

An effective partitioned AQPF algorithm is also proposed, which is capable of reducing the number of required (coherent) qubits for a given test system. This advantage is especially very important for performing computations on D-Wave's Advantage™ system. The partitioned AQPF algorithm, when used on various power test systems, shows remarkable capability. It can solve a partitioned formulation, which is a significant achievement, and enables PF analysis even when data is incomplete or inaccurate or when computational resources are insufficient. However, this reduction leads to an increased number of iterations needed to reach a defined threshold compared to the standard AQPF algorithm. Nevertheless, in the short term, this trade-off is beneficial as many NISQ architectures can quickly repeat executions[45].

The proposed AQPF algorithm is a valuable contribution to the literature and holds promise as a pivotal bridge toward harnessing adiabatic quantum capabilities for enhancing PF analysis. It is also noteworthy that the (partitioned) AQPF algorithm can link up with a classical Newton-Raphson (NR) solver. This combination can be applied with a large defined threshold to serve as a warm start for managing ill-conditioned cases. There are several questions that persist in the study of AQPF. These include broadening the scope of the algorithm to incorporate PV buses, devising a systematic approach for independently updating complex voltages for each bus in every iteration, exploring efficient order reduction techniques specifically developed for PF analysis, and eventually implementing the algorithm in real-world power systems.

Methods

Quantum annealing

(QA) is one of the two paradigms encountered in quantum computing. While gate-based quantum computing follows the principles of digital computing, that is, the data stored in the quantum registers is nuancedly controlled by applying a sequence of gates — the quantum program —, quantum annealing can be seen as an analog computing approach tailored toward solving unconstrained optimization problems given in Ising model formulation

$$\min_{\vec{s}\in\{\pm1\}^n} \sum_{i<j} J_{ij} s_i s_j + \sum_i h_i s_i, \tag{18}$$

with interaction terms J_{ij} and external magnetic field terms h_i. Starting from a superposition of all possible states, the system evolves following the time-dependent Schrödinger equation into a state of minimal energy—the ground state of the Hamiltonian. The results reported in this paper have been obtained on D-Wave's Advantage™ system which provides more than 5000 qubits and 35,000 couplers between them responsible for imprinting the problem Hamiltonian. Simply said, a positive weight, $J_{ij} > 0$, stimulates the opposite orientation of the spin qubits s_i and s_j, whereas a negative weight, $J_{ij} < 0$, stimulates $s_i = s_j$.

The main challenges in practical QA are as follows: (i) since physical qubits are not all-to-all connected the graph of the problem Hamiltonian needs to be 'embedded' into the hardware graph, which, in general, is an NP-hard problem; (ii) the number of couplers per qubit is limited (degree 15 for the Advantage™ system) so that a qubit, say, s_i with a lot of non-zero J_{ij}s needs to be expanded into a chain of qubits that are forced to attain the same value via sufficiently larger coupler values; and (iii) the value range of the hardware is limited meaning that J and h are rescaled to $[-2, 2]$ thereby possibly suppressing values that are small in magnitude. While (iii) leads to inaccurate results, (i) and (ii) are a severe problem for scalability since larger problem instances can no longer be embedded and, hence, cannot be solved with QA, cf. Fig. 2.

Digital annealing

(DA) mimics the annealing process on application-specific CMOS hardware that solves many Markov-Chain Monte-Carlo instances in parallel. Fujitsu's DA offers up to 100000 qubits organized in units of 8,192 fully-connected qubits and 64-bit precision. In contrast to D-Wave's QA, Fujitsu's DA accepts problems in QUBO form, which can be obtained from Eq. (18) by setting $s_i = 2x_i - 1$ and converting J and h into a symmetric matrix Q. For the problem sizes studied in this paper, a single unit of 8192 qubits suffices, so that embedding is not a problem.

Reduction of higher-order terms

is necessary to convert the fourth-order polynomial of the Hamiltonian (15) into a quadratic polynomial that can be directly solved on D-Wave's QA and Fujitsu's DA. A possible approach is to introduce auxiliary variables of the form $z_{ij} = x_i x_j$ and replace triplet interactions by

$$x_i x_j x_k = \min_{z_{ij}} \left\{ z_{ij} x_k + \lambda P\left(x_i, x_j; z_{ij}\right) \right\}, \quad \lambda > 0 \tag{19}$$

with the penalty term P defined as follows

$$P\left(x_i, x_j; z_{ij}\right) = x_i x_j - 2\left(x_i + x_j\right) z_{ij} + 3 z_{ij} \tag{20}$$

Likewise, expressions with four binary variables can be replaced by

$$x_i x_j x_k x_l = \min_{z_{ij}, z_{kl}} \left\{ \lambda P\left(x_i, x_j; z_{ij}\right) + \lambda P\left(x_k, x_l; z_{kl}\right) \right\} \tag{21}$$

A comprehensive overview of so-called quadratization techniques is given in[46].

Heuristic search

Algorithm 3 presents the heuristic search approach adopted in this work to avoid getting stuck in local minima and enable a partitioned AQPF algorithm.

```
for k = 0 to 99 do
    if k > 0 then
        Randomly select index and z ∈ {True, False}
        if z == True then
            Flip x⃗^μ / s⃗^μ at index
        else
            Flip x⃗^ω / s⃗^ω at index
        end if
    end if
    Compute μ⃗ and ω⃗ from equations (7)–(8)
    Calculate p⃗ and q⃗ from equations (1)–(2)
    Compute Hamiltonian H from equation (15)
    Append μ⃗, ω⃗, and H to list_mu, list_omega, list_H, respectively
end for
Find index k where H is minimum in list_H
Set μ⃗ and ω⃗ to corresponding values from list_mu[k] and list_omega[k], respectively
```

Algorithm 3. Heuristic search approach.

$\Delta\mu_i$ and $\Delta\omega_i$ update. $\Delta\mu_i$ and $\Delta\omega_i$ are updated as a function of the number of iterations. In the beginning, i.e., $\mathrm{it} = 0$, the increment/decrement is large, i.e., $\Delta\mu_i = 4 \times 10^{-2}$ and $\Delta\omega_i = 2 \times 10^{-2}$, and as the number of iterations increases, it becomes increasingly smaller to refine the values, e.g., $\Delta\mu_i = 1 \times 10^{-4}$ and $\Delta\omega_i = 5 \times 10^{-5}$ for $it = 50$. In this work, the same $\Delta\mu_i$ and $\Delta\omega_i$ are used for all the buses i.

Data availability

The data that support the plots within this article and other findings of this study are available from the corresponding author upon reasonable request.

Code availability

The code for this article is available from the corresponding author upon reasonable request.

Received: 17 May 2024; Accepted: 18 September 2024
Published online: 05 October 2024

References

1. Saadat, H. *Power System Analysis* (McGraw Hill, 1999).
2. Arrillaga, J. & Smith, B. *AC-DC Power System Analysis* (Institution of Electrical Engineers, 1998).
3. Schavemaker, P. & van der Sluis, L. *Electrical Power System Essentials* 2nd edn. (Wiley, 2017).
4. Bienstock, D. & Verma, A. Strong NP-hardness of AC power flows feasibility. *Oper. Res. Lett.* **47**, 494–501. https://doi.org/10.1016/j.orl.2019.08.009 (2019).
5. Al-Jaafreh, M. A. & Mokryani, G. Planning and operation of LV distribution networks: A comprehensive review. *IET Energy Syst. Integr.* **1**, 133–146. https://doi.org/10.1049/iet-esi.2019.0013 (2019).
6. Mokryani, G., Majumdar, A. & Pal, B. C. Probabilistic method for the operation of three-phase unbalanced active distribution networks. *IET Renew. Power Gener.* **10**, 944–954. https://doi.org/10.1049/iet-rpg.2015.0334 (2016).
7. Sexauer, J., Javanbakht, P. & Mohagheghi, S. Phasor measurement units for the distribution grid: Necessity and benefits. In *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)* 1–6 https://doi.org/10.1109/ISGT.2013.6497828 (IEEE, 2013).
8. Guttromson, R. Modeling distributed energy resource dynamics on the transmission system. *IEEE Trans. Power Syst.* **17**, 1148–1153. https://doi.org/10.1109/TPWRS.2002.804957 (2002).

9. Hu, X., Hu, H., Verma, S. & Zhang, Z.-L. Physics-guided deep neural networks for power flow analysis. *IEEE Trans. Power Syst.* **36**, 2082–2092. https://doi.org/10.1109/TPWRS.2020.3029557 (2021).
10. Sharma, N. *et al.* Major blackouts of the decade: Underlying causes, recommendations and arising challenges. In *2021 9th IEEE International Conference on Power Systems (ICPS)* 1–6 https://doi.org/10.1109/ICPS52420.2021.9670166 (IEEE, 2021).
11. Giraldo, J. S., Montoya, O. D., Vergara, P. P. & Milano, F. A fixed-point current injection power flow for electric distribution systems using Laurent series. *Electr. Power Syst. Res.* **211**, 108326. https://doi.org/10.1016/j.epsr.2022.108326 (2022).
12. Lopez, J. C., Vergara, P. P., Lyra, C., Rider, M. J. & da Silva, L. C. P. Optimal operation of radial distribution systems using extended dynamic programming. *IEEE Trans. Power Syst.* **33**, 1352–1363. https://doi.org/10.1109/TPWRS.2017.2722399 (2018).
13. Gonçalves, R. R., Alves, R. P., Franco, J. F. & Rider, M. J. Operation planning of electrical distribution systems using a mixed integer linear model. *J. Control Autom. Electr. Syst.* **24**, 668–679. https://doi.org/10.1007/s40313-013-0055-9 (2013).
14. Dai, X., Cai, Y., Jiang, Y. & Hagenmeyer, V. Rapid scalable distributed power flow with open-source implementation. *IFAC-PapersOnLine* **55**, 145–150. https://doi.org/10.1016/j.ifacol.2022.07.250 (2022).
15. Idema, R., Lahaye, D. J. P., Vuik, C. & van der Sluis, L. Scalable Newton–Krylov solver for very large power flow problems. *IEEE Trans. Power Syst.* **27**, 390–396. https://doi.org/10.1109/TPWRS.2011.2165860 (2012).
16. Zeng, L., Alawneh, S. G. & Arefifar, S. A. GPU-based sparse power flow studies with modified newtonâ€™s method. *IEEE Access* **9**, 153226–153239. https://doi.org/10.1109/ACCESS.2021.3127393 (2021).
17. Tostado-Véliz, M., Kamel, S. & Jurado, F. Power flow solution of Ill-conditioned systems using current injection formulation: Analysis and a novel method. *Int. J. Electr. Power Energy Syst.* **127**, 106669. https://doi.org/10.1016/j.ijepes.2020.106669 (2021).
18. Yang, X. & Zhou, X. Application of asymptotic numerical method with homotopy techniques to power flow problems. *Int. J. Electr. Power Energy Syst.* **57**, 375–383. https://doi.org/10.1016/j.ijepes.2013.12.014 (2014).
19. Sheng, H. & Chiang, H.-D. CDFLOW: A practical tool for tracing stationary behaviors of general distribution networks. *IEEE Trans. Power Syst.* **29**, 1365–1371. https://doi.org/10.1109/TPWRS.2013.2289917 (2014).
20. Tripathy, S., Prasad, G., Malik, O. & Hope, G. Load-flow solutions for Ill-conditioned power systems by a Newton-like method. *IEEE Trans. Power Appar. Syst.* **PAS–101**, 3648–3657. https://doi.org/10.1109/TPAS.1982.317050 (1982).
21. Tostado-Veliz, M. *et al.* Mann-iteration process for power flow calculation of large-scale Ill-conditioned systems: Theoretical analysis and numerical results. *IEEE Access* **9**, 132255–132266. https://doi.org/10.1109/ACCESS.2021.3114969 (2021).
22. Liu, Y., Sun, K. & Dong, J. A dynamized power flow method based on differential transformation. *IEEE Access* **8**, 182441–182450. https://doi.org/10.1109/ACCESS.2020.3028060 (2020).
23. Iwamoto, S. & Tamura, Y. A load flow calculation method for Ill-conditioned power systems. *IEEE Trans. Power Appar. Syst.* **PAS–100**, 1736–1743. https://doi.org/10.1109/TPAS.1981.316511 (1981).
24. Baker, K. Solutions of DC OPF are Never AC Feasible. *arXiv* (2019).
25. Duque, E. M. S. *et al.* Risk-aware operating regions for PV-rich distribution networks considering irradiance variability. *IEEE Trans. Sustain. Energy* **14**, 2092–2108. https://doi.org/10.1109/TSTE.2023.3281890 (2023).
26. Leipold, H. & Spedalieri, F. M. Quantum annealing with special drivers for circuit fault diagnostics. *Sci. Rep.* **12**, 11691. https://doi.org/10.1038/s41598-022-14804-8 (2022).
27. Jing, H., Wang, Y. & Li, Y. Data-driven quantum approximate optimization algorithm for power systems. *Commun. Eng.* **2**, 12. https://doi.org/10.1038/s44172-023-00061-8 (2023).
28. Silva, F. F. C., Carvalho, P. M. S. & Ferreira, L. A. F. M. A quantum computing approach for minimum loss problems in electrical distribution networks. *Sci. Rep.* **13**, 10777. https://doi.org/10.1038/s41598-023-37293-9 (2023).
29. Golestan, S. et al. Quantum computation in power systems. *Energy Rep.* **9**, 584–596. https://doi.org/10.1016/j.egyr.2022.11.185 (2023).
30. Feng, F., Zhou, Y. & Zhang, P. Quantum power flow. *IEEE Trans. Power Syst.* **36**, 3810–3812. https://doi.org/10.1109/TPWRS.2021.3077382 (2021).
31. Feng, F., Zhou, Y.-F. & Zhang, P. Noise-resilient quantum power flow. *iEnergy* **2**, 63–70. https://doi.org/10.23919/IEN.2023.0008 (2023).
32. Pareek, P., Jayakumar, A., Coffrin, C. & Misra, S. Demystifying quantum power flow: Unveiling the limits of practical quantum advantage. Tech. Rep.
33. Kaseb, Z., Möller, M., Balducci, G. T., Palensky, P. & Vergara, P. P. Quantum neural networks for power flow analysis. *Electr. Power Syst. Res.* **235**, 110677. https://doi.org/10.1016/j.epsr.2024.110677 (2024).
34. Halffmann, P., Holzer, P., Plociennik, K. & Trebing, M. A Quantum Computing Approach for the Unit Commitment Problem. 113–120, https://doi.org/10.1007/978-3-031-24907-5_14 (2023).
35. Morstyn, T. Annealing-based quantum computing for combinatorial optimal power flow. *IEEE Trans. Smart Grid* **14**, 1093–1102. https://doi.org/10.1109/TSG.2022.3200590 (2023).
36. McGeoch, C. C. *Adiabatic Quantum Computation and Quantum Annealing Theory and Practice* (Springer, 2014).
37. Beer, K. *et al.* Training deep quantum neural networks. *Nat. Commun.* **11**, 808. https://doi.org/10.1038/s41467-020-14454-2 (2020).
38. Cao, C., Xue, J., Shannon, N. & Joynt, R. Speedup of the quantum adiabatic algorithm using delocalization catalysis. *Phys. Rev. Res.* **3**, 013092. https://doi.org/10.1103/PhysRevResearch.3.013092 (2021).
39. Pelofske, E., Bärtschi, A. & Eidenbenz, S. Short-depth QAOA circuits and quantum annealing on higher-order ising models. *npj Quantum Inf.* **10**, 30. https://doi.org/10.1038/s41534-024-00825-w (2024).
40. Bian, Z. *et al.* Discrete optimization using quantum annealing on sparse Ising models. *Front. Phys.* **2**, 56. https://doi.org/10.3389/fphy.2014.00056 (2014).
41. De Simone, C. *et al.* Exact ground states of Ising spin glasses: New experimental results with a branch-and-cut algorithm. *J. Stat. Phys.* **80**, 487–496. https://doi.org/10.1007/BF02178370 (1995).
42. Asaoka, H. & Kudo, K. Nonnegative/Binary matrix factorization for image classification using quantum annealing. *Sci. Rep.* **13**, 16527. https://doi.org/10.1038/s41598-023-43729-z (2023).
43. Date, P., Arthur, D. & Pusey-Nazzaro, L. QUBO formulations for training machine learning models. *Sci. Rep.* **11**, 10029. https://doi.org/10.1038/s41598-021-89461-4 (2021).
44. Date, P. & Potok, T. Adiabatic quantum linear regression. *Sci. Rep.* **11**, 21905. https://doi.org/10.1038/s41598-021-01445-6 (2021).
45. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510. https://doi.org/10.1038/s41586-019-1666-5 (2019).
46. Dattani, N. Quadratization in discrete optimization and quantum mechanics. arXiv (2019).

Acknowledgements

This work is part of the DATALESs project, with project number 482.20.602, jointly financed by the Netherlands Organization for Scientific Research (NWO) and the National Natural Science Foundation of China (NSFC). This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant number EINF-6569. The authors gratefully acknowledge the Jülich Supercomputing Centre (https://www.fzjuelich.de/ias/jsc) for funding this project by providing computing time on the D-Wave Advantage™ System JUPSI through the Jülich UNified Infrastructure for Quantum computing (JUNIQ). The research received support

from the Center of Excellence RAISE, which receives funding from the European Union's Horizon 2020 - Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733. We would furthermore like to thank Fujitsu Technology Solutions for providing access to the Fujitsu Digital Annealer V3 and, in particular, to Markus Kirsch for his support and custom extensions of the DADK python package.

Author contributions

This work was initiated in discussions with all the authors. Z.K. and M.M. conceived the experiments, analyzed the results, and contributed to writing the manuscript. Z.K. developed the code, conducted the experiments, and created the display items. Z.K., M.M., and P.P.V. reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-73512-7.

Correspondence and requests for materials should be addressed to Z.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.