Article

Optimality guarantees for crystal structure prediction

https://doi.org/10.1038/s41586-023-06071-y

Received: 19 July 2022

Accepted: 4 April 2023

Published online: 5 July 2023

Check for updates

Vladimir V. Gusev[1,2], Duncan Adamson[1], Argyrios Deligkas[1,4], Dmytro Antypov[1], Christopher M. Collins[3], Piotr Krysta[2], Igor Potapov[2], George R. Darling[3], Matthew S. Dyer[1,3], Paul Spirakis[1,2]✉ & Matthew J. Rosseinsky[1,3]✉

Crystalline materials enable essential technologies, and their properties are determined by their structures. Crystal structure prediction can thus play a central part in the design of new functional materials[1,2]. Researchers have developed efficient heuristics to identify structural minima on the potential energy surface[3–5]. Although these methods can often access all configurations in principle, there is no guarantee that the lowest energy structure has been found. Here we show that the structure of a crystalline material can be predicted with energy guarantees by an algorithm that finds all the unknown atomic positions within a unit cell by combining combinatorial and continuous optimization. We encode the combinatorial task of finding the lowest energy periodic allocation of all atoms on a lattice as a mathematical optimization problem of integer programming[6,7], enabling guaranteed identification of the global optimum using well-developed algorithms. A single subsequent local minimization of the resulting atom allocations then reaches the correct structures of key inorganic materials directly, proving their energetic optimality under clear assumptions. This formulation of crystal structure prediction establishes a connection to the theory of algorithms and provides the absolute energetic status of observed or predicted materials. It provides the ground truth for heuristic or data-driven structure prediction methods and is uniquely suitable for quantum annealers[8–10], opening a path to overcome the combinatorial explosion of atomic configurations.

There are more than 200,000 crystal structures known and held in curated databases as lists of atomic positions[11,12]. Knowing the structure enables the accurate prediction of stability and, in many cases, properties. However, when considering a previously unreported composition without restriction to adopting structures that are included in the databases, the structure cannot be known and must be predicted to permit assessment of its stability and properties. The central feature of crystal structure prediction (CSP) is that it begins with no information about the positions of the atoms in the unit cell and aims to find their exact arrangement[13]. To predict thermodynamically stable compounds, we ask whether there exists a crystal structure for a given composition with an energy below a given threshold, defined by the convex hull[2]. This decision version[14,15] of CSP lies at the heart of in silico material discovery. Over the years, substantial effort has been taken in CSP approaches that aim to quickly identify low-energy structures. However, a formal algorithm, as postulated by the Church–Turing thesis[16], should not only be able to identify such structures but also provide a non-existence proof if the target energy cannot be reached. The difference between finding a solution and proving its optimality is evident in mathematics, where confirmation of conjectures can take decades or even centuries, for example, the Kepler conjecture[17] about the densest sphere packings and its generalizations in high dimensions have only been recently established[17,18]. The formal statement capturing this distinction is probably the most important open problem in computer science: $P = NP$ asks whether efficient ways of finding proofs of optimality exist[19]. Until now, there are no methods for CSP of extended inorganic solids that provide energy optimality guarantees in the continuous space of unknown atomic positions; thus, no formal algorithm for this problem has been presented.

This is in contrast to the general optimization theory, in which formal algorithms for a wide range of problems have been devised and their optimality and approximation guarantees have been thoroughly investigated[19–21]. One of the most general methods to introduce optimality guarantees for a variety of practical problems is integer programming[6]. This method consists of rewriting the problem in a particular form by introducing integer decision variables, constraints and an objective function corresponding to the task. Thus, optimization algorithms can be applied to all encoded problems at once and developed in an abstract setting independent of the actual problem. This universality has led to the widespread use of integer programming in areas such as logistics, manufacturing, healthcare, finance and computer vision[7] and to the development of robust methods and commercial solvers[22,23]. One of the key advances in this area is a class of branch-and-cut optimization algorithms that are capable of rapidly eliminating large parts of the

[1]Leverhulme Research Centre for Functional Materials Design, Materials Innovation Factory, University of Liverpool, Liverpool, UK. [2]Department of Computer Science, University of Liverpool, Liverpool, UK. [3]Department of Chemistry, University of Liverpool, Liverpool, UK. [4]Present address: Department of Computer Science, Royal Holloway, University of London, London, UK. ✉e-mail: p.spirakis@liverpool.ac.uk; m.j.rosseinsky@liverpool.ac.uk

optimization domain from consideration if the current best solution cannot be improved there. Modelling an optimization problem as an integer program addressed with the branch-and-cut method not only allows one to solve much larger problems than possible by brute force[24,25] but also provides numerical upper and lower bounds on the optimal solution during the run and proof of optimality when the run is complete.

These benefits prompted the use of mathematical optimization for diverse material-design challenges such as molecular conformation prediction[26], molecular design[27], protein folding[28], Coulomb glass modelling[29] and substitutions into perovskites[30] and other known parent structures[31]. Benefiting from these demonstrated advantages of combinatorial guarantees, we provide a generally applicable CSP algorithm that addresses the continuous space of possible atomic sites to correctly predict a diverse set of structures. This method determines all the atomic positions previously unknown to the algorithm. The coupling of local minimization to integer programming that we use enables the exploration of the continuous space using strong optimization methods on a discrete space to obtain physical energy guarantees. We start by considering the allocation of all the atoms that define the composition of a material to a suitably dense set of discrete positions within a unit cell treated with periodic boundary conditions. Given a unit cell with a set of positions Pos = {pos_1, pos_2, …, pos_n}, we find an assignment of a given number of atoms of species Types = {typ_1, typ_2, …, typ_k} to them while minimizing the interaction energy. Not all positions have to be occupied. In this work, these positions form a lattice and are specified by their fractional coordinates as $\left(\frac{i}{g}, \frac{j}{g}, \frac{k}{g}\right)$ for integers $i,j,k \in \{0,1,\ldots,g-1\}$, where g is the discretization parameter, equal to the number of positions per side of the unit cell, which defines the discretization or separation between the lattice positions. A typical example is given in Fig. 1, in which the set Pos contains 64 uniformly distributed positions within the unit cell ($g = 4$) and Types = {Sr, Ti, O}.

We start the encoding of the atom allocation problem into an equivalent integer program by introducing binary variables $X^{\mathrm{typ}}_{\mathrm{pos}}$ for every pos and typ, which capture our decision to place an atom of type typ at position pos. The value of $X^{\mathrm{typ}}_{\mathrm{pos}}$ equals 1 if a typ atom occupies pos and 0 otherwise. Because not all variable assignments are physical—for example, both Sr and Ti cannot be located at the same position—we introduce linear constraints to ensure that all solutions of the program correspond to physically reasonable atomic arrangements with the correct stoichiometry (Methods).

The remaining part of the encoding is the objective function—the interaction energy[4,32] of the resulting allocation computed with periodic boundary conditions. Here, we focus on commonly used approaches on the basis of interatomic potentials[3] that represent the energy as the sum of the electrostatic interaction of ions treated as point charges and the repulsion contribution from closely located ions, addressing the widely studied class of ionic materials that enable key technologies[33–35]. Because the electrostatic interaction is long range, special summation methods are used, of which the Ewald sum is the most common[36]. An important observation that is critical for the encoding is that the Ewald sum can be rearranged into a finite sum over all pairs of atoms within a unit cell and these pairwise contributions can be computed independently of the positions of all other atoms within the unit cell (Methods). When repulsion is modelled using two-body potentials either in the form of classical force fields (such as Buckingham or Lennard–Jones) or more-flexible machine learning potentials[32], the total potential energy of a crystal can be written as the sum of pairwise contributions. These individual summands for every possible allocation of a pair of ions can be computed independently and stored in a table. If we denote the constant value from this table that defines the energy contribution of ions typ_1 and typ_2 placed at pos_1 and pos_2, respectively, as $\alpha^{\mathrm{typ}_1,\mathrm{typ}_2}_{\mathrm{pos}_1,\mathrm{pos}_2}$, then the energy of an allocation can be written as

$$E = \sum_{\substack{\mathrm{pos}_1,\mathrm{pos}_2 \in \mathrm{Pos},\\ \mathrm{typ}_1,\mathrm{typ}_2 \in \mathrm{Types}}} \alpha^{\mathrm{typ}_1,\mathrm{typ}_2}_{\mathrm{pos}_1,\mathrm{pos}_2} X^{\mathrm{typ}_1}_{\mathrm{pos}_1} X^{\mathrm{typ}_2}_{\mathrm{pos}_2}. \quad (1)$$

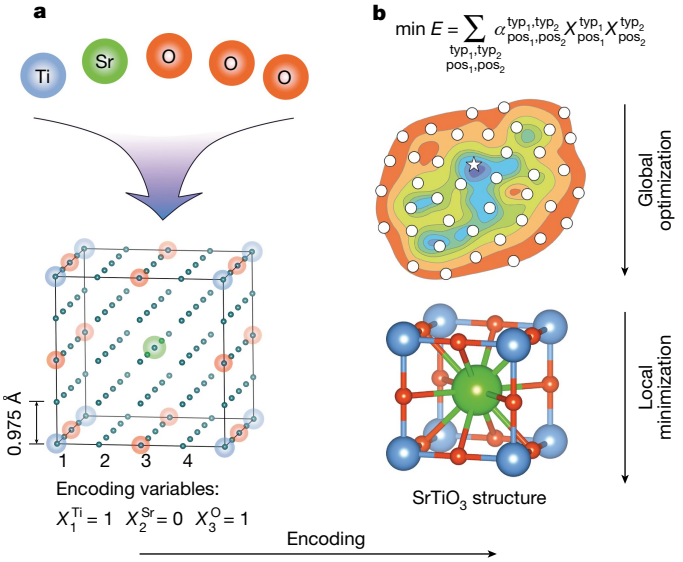

Fig. 1 | CSP using integer programming. a, Atoms defining a specific composition (shown here for $SrTiO_3$) are allocated to a suitably dense set of discrete positions in space under periodic boundary conditions. **b**, The resulting configurations generate candidate crystal structures that lie on the PES (circles). Structure prediction can be done by identifying and then locally minimizing low-energy allocations to afford the lowest energy structure with atomic positions in continuous space. If the space of configurations is well chosen, this leads to the correct crystal structure in a single local minimization of the lowest energy global optimum allocation (star). Evaluation of atomic configurations to identify the globally optimal allocation is achieved by encoding this task as an integer program. This is an established mathematical optimization problem, which can be solved using existing solvers based on powerful algorithms and emerging quantum computers.

The term $\alpha^{\mathrm{typ}_1,\mathrm{typ}_2}_{\mathrm{pos}_1,\mathrm{pos}_2}$ is present if and only if both $X^{\mathrm{typ}_1}_{\mathrm{pos}_1}$ and $X^{\mathrm{typ}_2}_{\mathrm{pos}_2}$ are equal to 1, which corresponds to both these positions being occupied in an allocation, ensuring correct energy values.

The set of variables $X^{\mathrm{typ}}_{\mathrm{pos}}$, chosen constraints and the exact form of the energy function E in equation (1) define an integer program (Supplementary Information) for which the candidate solutions are in one-to-one correspondence with allocations of atoms to lattice positions. It is a binary quadratic program, in which the variables $X^{\mathrm{typ}}_{\mathrm{pos}}$ appear in quadratic terms. Such optimization problems are typically computationally intractable according to the theory of NP-completeness[19]. However, many instances can be solved efficiently using branch-and-cut methods in existing optimizers[22,23]. These optimizers relax the problem by allowing the variables $X^{\mathrm{typ}}_{\mathrm{pos}}$ to take values between 0 and 1, resulting in a more tractable problem with a smaller minimum objective value E. The branch-and-cut method maintains an appropriate set of solutions to the relaxed problem across iterations and, by using cuts, gradually narrows down the relaxed continuous feasibility space of the problem to finally reach the guaranteed optimal binary solution (Methods). We further apply space group symmetry to identify the minimal set of lattice positions that are unique given the symmetry and introduce proximity constraints (Methods), which reduces the run time of optimization methods by focussing on desired subspaces. We refer to the resulting search space of atom allocations to the lattice positions that satisfy all imposed constraints as the configuration space.

Exact solutions of different periodic lattice atom allocation problems can be used to non-heuristically investigate the potential energy surface (PES). The configuration space and the corresponding integer program are specified for a given composition, which enables a branch-and-cut algorithm to find the same global optimum for this problem in every run. With this exact allocation of atoms on the lattice, we then remove

Table 1 | Configuration spaces that lead to prediction of the experimentally determined cubic crystal structures

Compound	Space group of the structure	Number of ions in the unit cell	Cell parameter (Å)	Discretization, *g*	Space group symmetry	Number of unique lattice positions	Time (s)
$SrTiO_3$, *Z*=1	$Pm\bar{3}m$ (221)	5	3.9	4	*P*1 (1)	64	3
$SrTiO_3$, *Z*=8	$Pm\bar{3}m$ (221)	40	7.8	8	*P*23 (195)	56	56
$SrTiO_3$, *Z*=27	$Pm\bar{3}m$ (221)	135	11.7	6	$Pm\bar{3}m$ (221)	20	2
$SrTiO_3$, *Z*=27	$Pm\bar{3}m$ (221)	135	11.7	6	$Pm\bar{3}$ (200)	24	63
$SrTiO_3$, *Z*=27	$Pm\bar{3}m$ (221)	135	11.7	6	*P*23 (195)	28	6,823
Y_2O_3	$Ia\bar{3}$ (206)	80	10.7	8	$Ia\bar{3}$ (206)	17	1
Y_2O_3	$Ia\bar{3}$ (206)	80	10.7	8	$I2_13$ (199)	40	10
Y_2O_3	$Ia\bar{3}$ (206)	80	10.7	16	$Ia\bar{3}$ (206)	124	18
$Y_2Ti_2O_7$[a]	$Fd\bar{3}m$ (227)	88	10.2	8	$Fd\bar{3}m$ (227)	11	1
$Y_2Ti_2O_7$	$Fd\bar{3}m$ (227)	88	10.2	16	$Fd\bar{3}m$ (227)	51	1
$MgAl_2O_4$	$Fd\bar{3}m$ (227)	56	8.2	8	$Fd\bar{3}m$ (227)	11	1
$MgAl_2O_4$	$Fd\bar{3}m$ (227)	56	8.2	16	$Fd\bar{3}m$ (227)	51	1
$MgAl_2O_4$	$Fd\bar{3}m$ (227)	56	8.2	8	*F*23 (196)	22	1
$MgAl_2O_4$[a]	$Fd\bar{3}m$ (227)	56	8.2	8	*P*23 (195)	56	4,085
$Ca_3Al_2Si_3O_{12}$	$Ia\bar{3}d$ (230)	160	11.9	8	$Ia\bar{3}$ (206)	17	1
$Ca_3Al_2Si_3O_{12}$	$Ia\bar{3}d$ (230)	160	11.9	16	$Ia\bar{3}d$ (230)	62	1

The experimentally determined space group symmetry is given in the second column. The perovskite structure ($SrTiO_3$) was investigated for a varying number of formula units (*Z*). Other structure types—namely, bixbyite (Y_2O_3), pyrochlore ($Y_2Ti_2O_7$), spinel ($MgAl_2O_4$) and garnet ($Ca_3Al_2Si_3O_{12}$)—were investigated using different configuration spaces. In all cases, the minima were not only identified but also proved to be optimal. The corresponding periodic lattice atom allocation problems are defined by the dimension of the unit cell and the number of lattice positions per side, *g*. The space group symmetry defines the number of unique lattice positions that is proportional to the size of the integer program.

[a]The configuration space requires local minimization of several low-energy allocations on a lattice before the experimentally determined structure is identified, whereas in all other cases, the global optimum allocation of lattice atoms immediately leads to the correct structure after one local minimization. The representative time needed to identify solutions of the corresponding integer programs alongside the proof of their optimality is given for the hardware specified in the Methods.

the restriction that atoms occupy only lattice positions to predict the crystal structure by local minimization of these optimum configurations. The coupling of local minimization to integer programming enables the exploration of the PES, which is a continuous space, using powerful optimization algorithms on a discrete space (Fig. 1). We examine this approach to CSP (Table 1) on a prototype set of compositions that adopt cubic crystal structures: 21% of all materials in the Inorganic Crystal Structure Database[11] are cubic, including families of ionic materials that have gained considerable attention because of their functional properties such as garnet[33], perovskite[34] and spinel[35].

The garnet structure of the first composition studied, $Ca_3Al_2Si_3O_{12}$, cannot be explained on the basis of the individual sphere packings, requiring instead description as a cylindrical rod packing[37]. This shows its complexity, with distinct 12 (Ca, dodecahedron), 6 (Al, octahedron) and 4 (Si, tetrahedron) coordination of the three electropositive elements by O (oxide occupies a general crystallographic position with no symmetry and is four-coordinated by Al, Si and two Ca atoms). At a unit cell parameter of 11.9 Å with a discretization of 0.75 Å ($g = 16$) in the $Ia\bar{3}d$ space group, the integer programming formulation allocates the four distinct atomic positions required on the 62 unique lattice positions in 1 s on a desktop computer. The integer program also returns a guarantee of optimality for the periodic lattice atom allocation in this run time. This optimal allocation is sufficiently precise that the correct experimentally determined structure (Fig. 2) is immediately recovered by one local minimization of this single configuration, which requires a mean shift of approximately 0.29 Å in atomic positions. The integer program thus identifies a configuration on a lattice that lies in the global minimum basin of the continuous PES and certifies that this is the lowest energy structure possible at this composition under the stated assumptions, because it provides guarantees of optimality.

In addition to $Ca_3Al_2Si_3O_{12}$, we applied this integer programming CSP approach to investigate the PES of the following compositions: $SrTiO_3$, Y_2O_3, $Y_2Ti_2O_7$ and $MgAl_2O_4$. Their experimentally determined structures correspond to the perovskite, bixbyite, pyrochlore and spinel structure types, respectively[38]. We investigated different supercells of $SrTiO_3$ with up to 135 atoms in them to assess the scalability of the approach. The other structures highlight the complexity of multidimensional CSP with 56–160 atoms in the unit cell (Methods). Table 1 reports the configuration spaces that result in prediction of the experimentally determined structures and the times required to guarantee that the solution, and thus the experimentally determined structure itself, is optimal given the composition in each of these archetypal cases.

For every composition (Table 1), local minimization of the single global optimum allocation for a moderate discretization of 0.6–0.7 Å led to the correct structure. This indicates that periodic lattice atom allocations capture crystal structures in continuous space and enable the identification of the global minimum on the PES. The guaranteed bound on the energy difference between the continuous and discrete solutions can be computed and decreases to zero as the discretization becomes finer (Supplementary Information). As the ions are not arbitrarily small, unlike lattice points, it is a physically reasonable hypothesis that a discretization of the order of 30–50% of the shortest interionic distances should enable the correct allocation to be identified. For all the examples in Table 1, discretizations that correspond to a fraction of a bond length lead to globally optimal solutions in the continuous space, reflecting the role of ionic size in determining the structure using repulsive contributions to the energy at short separations, supporting this hypothesis. Moreover, the allocation outcome was insensitive (Extended Data Tables 1–5) to the unit cell parameter—for example, $SrTiO_3$ returns the same allocation over 3–5 Å. A possible explanation is that the change in unit cell size affects the whole pool of low-energy solutions similarly, making the optimal allocation relatively stable and enabling reliable identification of structures.

Although the solution of an integer program is guaranteed to be optimal in the discrete space of the lattice, for coarser discretizations we find that the local minimization of that solution can be different from the global optimum in the continuous space, consistent with the hypothesis above. For lattice discretizations of 1.03–1.49 Å,

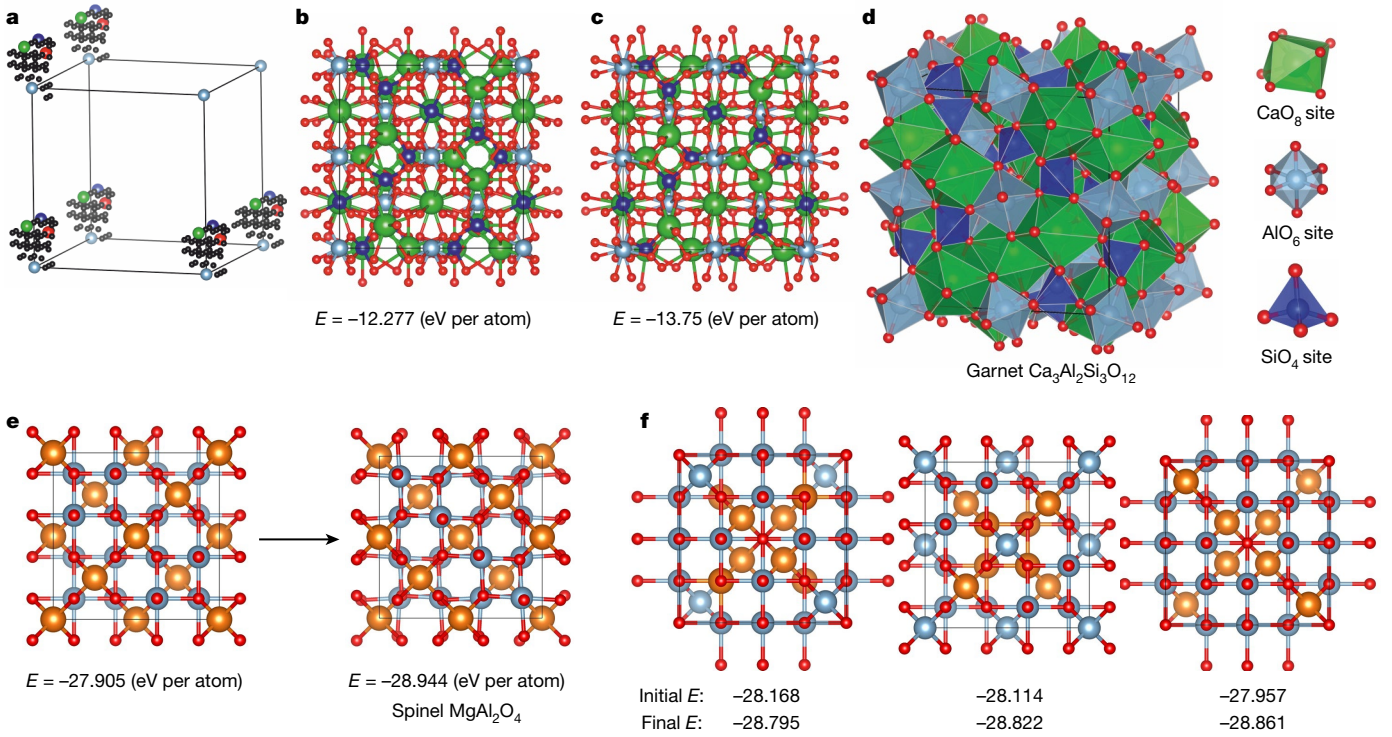

Fig. 2 | Using integer programming to predict garnet ($Ca_3Al_2Si_3O_{12}$) and spinel ($MgAl_2O_4$) structures. a, Unique positions (black spheres) of a lattice with 0.74 Å discretization ($g = 16$) in the $Ia\bar{3}d$ space group. The space group[46] is the group of symmetry operations that describes the symmetry of the unit cell. Atomic positions corresponding to the global optimum solution of the resulting periodic lattice atom allocation problem for $Ca_3Al_2Si_3O_{12}$ are coloured: Ca (green), Al (light blue), Si (dark blue), O (red). **b**–**d**, Local energy minimization of this solution (**b**) affords the correct garnet structure (**c** and **d**). **e**, Periodic lattice atom allocation for 1.02 Å discretization ($g = 8$) and $P23$ space group symmetry for $MgAl_2O_4$ (left) that minimizes into the correct spinel structure (right). **f**, There are allocations in this configuration space that are lower in energy than the structure shown in **e**, as quantified by the initial energy shown, but these allocations do not minimize into the correct structure, as shown by comparison of the final energies with that in **e**. The same discretization with the higher symmetry $F23$ space group provides the lowest energy solution that immediately minimizes into the correct structure.

identification of the correct structure by one local minimization of the global optimum atom allocation was predominantly but not uniformly possible. For $MgAl_2O_4$ with a discretization of 1.03 Å ($g = 8$) and $P23$ space group symmetry, the configuration space became large enough to contain low-energy lattice allocations that belong to local, rather than global, minima of the PES (Fig. 2). Nevertheless, minimization of the four lowest energy arrangements recovered the correct spinel structure. For higher symmetries, the lowest energy allocation even on this coarser lattice provided the correct structure directly (Table 1).

A heuristic partially searches the PES for preferred configurations that are then locally minimized—success relies on identifying a configuration that lies on the walls of the global minimum (Fig. 3). By contrast, integer programming considers all periodic lattice atom allocations simultaneously, identifying the globally optimal configuration. The branch-and-cut algorithm allows us to discard large portions of the configuration space while retaining optimality, leading to brute-force-like energy guarantees without actual brute force. By locally minimizing these exact outcomes from the appropriate discretization and unit cell parameters, we can obtain the guaranteed global minimum in CSP. Further development will integrate this configuration space screening with the space group symmetry and unit cell metric determination that is common to all CSP. Beyond that, as the solver maintains a valid solution, it can identify the global optimum much earlier than proving its optimality (Fig. 3d). By foregoing optimality guarantees, it would be possible to investigate larger configuration spaces and use integer programming either as an independent heuristic tool or to complement allocation decisions in existing heuristic methods.

The current state of the art in computational complexity[6] indicates that there will always be a combinatorial limit to the implementation of integer programs for CSP on classical computers, just as heuristics will eventually run out of capacity to explore complex structural spaces efficiently enough to generate reliable outcomes. Quantum computers have the potential to solve many problems faster than classical computers[39], demonstrating the so-called quantum advantage[40–42]. Although large-scale implementation of quantum computing is not imminent, more-limited forms are increasingly available. One example is the quantum annealer[43], a specialized hardware solver for quadratic unconstrained binary optimization (QUBO) problems[9] alongside other types of Ising machine[10,44]. The QUBO problem involves finding a 0,1 assignment minimizing an objective function containing only products of at most two binary variables. Equation (1) has exactly this form after eliminating the constraints, creating a pathway to overcome the combinatorial explosion in CSP. We have verified the applicability of this approach to CSP by solving the structures of SrO ($g = 2$, $P23$), $SrTiO_3$ ($g = 2$, $Pm\bar{3}m$), ZrO_2 ($g = 4$, $P2_13$) and ZnS ($g = 4$, $P23$) on the 2000Q quantum annealer, which is freely available from D-Wave through Leap[45] (Methods).

The search routines that find the lowest energy periodic lattice atom allocations can be used to predict crystal structures with one subsequent local minimization. Integer programming formulation of this search affords an algorithm that enables guaranteed identification of the global optimum in CSP and enables quantum computers to address the combinatorial challenges that arise. The resulting structures are thus demonstrated to provide the lowest energy possible at a given composition, proving the optimality of the observed structures of archetypal materials under clear assumptions. This provides both ground truth for heuristic and data-driven structure-prediction methods and essential understanding by guaranteeing the energetic status

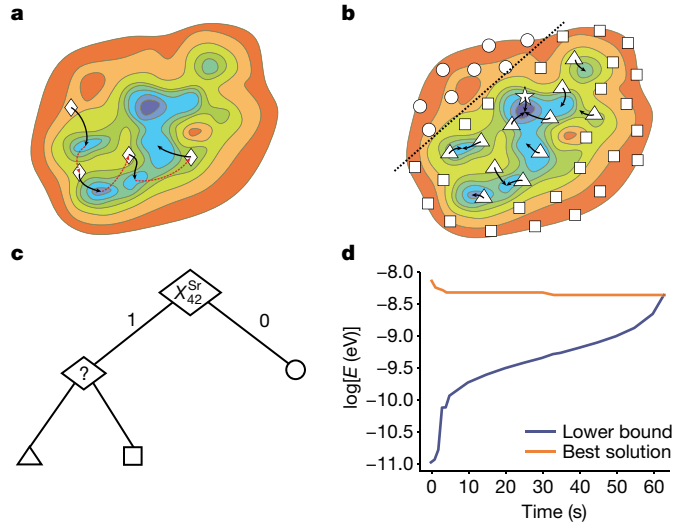

Fig. 3 | Comparison between heuristic and non-heuristic exploration of a PES. **a**, A typical CSP scheme partially explores the surface by iteratively performing local minimization (solid black arrows) of selected atomic configurations (diamonds) in a continuous space. These configurations are derived according to some search strategy (dotted red arrows) defined by a particular heuristic. **b**, CSP leveraging integer programming performs exact global optimization on the whole discrete configuration space of crystals generated by periodic atom allocation on a lattice, followed by local structural minimization of one (star) or more (triangles) low-energy solutions in a continuous space. **c**, Branch-and-cut algorithms designed for integer programs achieve the guaranteed global optimum periodic lattice atom allocation by separating all atomic configurations into branches roughly corresponding to the allocation decisions. Some branches (represented as circles in **b** to indicate their position on the PES) are completely discarded (dotted line in **b**), if all allocations are guaranteed to be worse than the best solution found so far (triangle). More promising branches (triangles) are explored before less promising ones (squares). The branch-and-cut tree is expanded until all configurations are assessed. **d**, An optimization run of the integer program solver on the periodic lattice atom allocation problem for $SrTiO_3$ ($Z = 27$, Table 1). At every moment, a lower bound on the global optimum is available as well as the lowest energy allocation found so far. The run is completed when these energies match, providing a guaranteed solution. The global optimum is identified before the proof of optimality.

of experimentally isolated materials in the laboratory. Development of encodings and implementations that make best use of emerging software and hardware will define a distinct CSP on the basis of optimality, certainty and quantum advantage, enabling new workflows for synthetic prioritization and property prediction.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-023-06071-y.

1. Collins, C. et al. Accelerated discovery of two crystal structure types in a complex inorganic phase field. *Nature* **546**, 280–284 (2017).
2. Oganov, A. R., Pickard, C. J., Zhu, Q. & Needs, R. J. Structure prediction drives materials discovery. *Nat. Rev. Mater.* **4**, 331–348 (2019).
3. Woodley, S. M., Day, G. M. & Catlow, R. Structure prediction of crystals, surfaces and nanoparticles. *Phil. Trans. R. Soc. A* **378**, 20190600 (2020).
4. Oganov, A. R., Saleh, G. & Kvashnin, A. G. (eds) *Computational Materials Discovery* (Royal Society of Chemistry, 2018).
5. Wales, D. J. *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses* (Cambridge Univ. Press, 2003).
6. Wolsey, L. A. *Integer Programming* 2nd edn (Wiley, 2020).
7. Jünger, M. et al. *50 Years of Integer Programming 1958–2008* (Springer, 2010).
8. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014).
9. Berwald, J. J. The mathematics of quantum-enabled applications on the D-Wave quantum computer. *Not. Am. Math. Soc.* **66**, 832–841 (2019).
10. Mohseni, N., McMahon, P. L. & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* **4**, 363–379 (2022).
11. Igor, L. *NIST Inorganic Crystal Structure Database (ICSD)* (National Institute of Standards and Technology, 2018); https://doi.org/10.18434/M32147.
12. Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. The Cambridge Structural Database. *Acta Crystallogr. B* **72**, 171–179 (2016).
13. Woodley, S. M. & Catlow, R. Crystal structure prediction from first principles. *Nat. Mater.* **7**, 937–946 (2008).
14. Adamson, D., Deligkas, A., Gusev, V. & Potapov, I. On the hardness of energy minimisation for crystal structure prediction. *Fundam. Inform.* **184**, 181–203 (2021).
15. Adamson, D., Deligkas, A., Gusev, V. V. & Potapov, I. The complexity of periodic energy minimisation. In *47th International Symposium on Mathematical Foundations of Computer Science* (eds Szeider, S. et al.) Vol. 241, 8:1–8:15 (LIPIcs, 2022).
16. Sipser, M. *Introduction to the Theory of Computation* 3rd edn (Cengage Learning, 2012).
17. Hales, T. C. A proof of the Kepler conjecture. *Ann. Math.* **162**, 1065–1185 (2005).
18. Cohn, H., Kumar, A., Miller, S. D., Radchenko, D. & Viazovska, M. The sphere packing problem in dimension 24. *Ann. Math.* **185**, 1017–1033 (2017).
19. Papadimitriou, C. H. & Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity* (Prentice Hall, 1998).
20. Goemans, M. X. Semidefinite programming in combinatorial optimization. *Math. Program.* **79**, 143–161 (1997).
21. Williamson, D. P. & Shmoys, D. B. *The Design of Approximation Algorithms* (Cambridge Univ. Press, 2011).
22. Gurobi Optimization. *Gurobi Optimizer Reference Manual* (Gurobi Optimization, 2022).
23. Kronqvist, J., Bernal, D. E., Lundell, A. & Grossmann, I. E. A review and comparison of solvers for convex MINLP. *Optim. Eng.* **20**, 397–455 (2019).
24. Applegate, D. L., Bixby, R. E., Chvátal, V. & Cook, W. J. *The Traveling Salesman Problem: A Computational Study* (Princeton Univ. Press, 2011).
25. Elf, M., Gutwenger, C., Jünger, M. & Rinaldi, G. in *Computational Combinatorial Optimization. Lecture Notes in Computer Science* (eds Jünger, M. & Naddef, D.) Vol. 2241, 157–222 (Springer, 2001).
26. Havel, T. F., Kuntz, I. D. & Crippen, G. M. The combinatorial distance geometry method for the calculation of molecular conformation. I. A new approach to an old problem. *J. Theor. Biol.* **104**, 359–381 (1983).
27. Achenie, L., Venkatasubramanian, V. & Gani, R. (eds) *Computer Aided Molecular Design: Theory and Practice* (Elsevier, 2002).
28. Babbush, R., Perdomo-Ortiz, A., O'Gorman, B., Macready, W. & Aspuru-Guzik, A. in *Advances in Chemical Physics* Vol. 155, (eds Rice, S. A. & Dinner, A. R.) Ch. 5, 201–243 (John Wiley, 2014).
29. Pörn, R., Nissfolk, O., Jansson, F. & Westerlund, T. The Coulomb glass - modeling and computational experience with a large scale 0–1 QP problem. *Comput. Aided Chem. Eng.* **29**, 658–662 (2011).
30. Hanselman, C. L. et al. A framework for optimizing oxygen vacancy formation in doped perovskites. *Comput. Chem. Eng.* **126**, 168–177 (2019).
31. Yin, X. & Gounaris, C. E. Search methods for inorganic materials crystal structure prediction. *Curr. Opin. Chem. Eng.* **35**, 100726 (2022).
32. Behler, J. & Csányi, G. Machine learning potentials for extended systems: a perspective. *Eur. Phys. J. B* **94**, 142 (2021).
33. Wang, C. et al. Garnet-type solid-state electrolytes: materials, interfaces, and batteries. *Chem. Rev.* **120**, 4257–4300 (2020).
34. Zhang, W., Eperon, G. E. & Snaith, H. J. Metal halide perovskites for energy applications. *Nat. Energy* **1**, 16048 (2016).
35. Zhao, Q., Yan, Z., Chen, C. & Chen, J. Spinels: controlled preparation, oxygen reduction/evolution reaction application, and beyond. *Chem. Rev.* **117**, 10121–10211 (2017).
36. Toukmaji, A. Y. & Board, J. A.Jr. Ewald summation techniques in perspective: a survey. *Comput. Phys. Commun.* **95**, 73–92 (1996).
37. Andersson, S. & O'Keeffe, M. Body-centred cubic cylinder packing and the garnet structure. *Nature* **267**, 605–606 (1977).
38. Hyde, B. G. & Andersson, S. *Inorganic Crystal Structures* (Wiley, 1989).
39. Bharti, K. et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **94**, 015004 (2022).
40. Zhong, H.-S. et al. Quantum computational advantage using photons. *Science* **370**, 1460–1463 (2020).
41. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
42. Madsen, L. S. et al. Quantum computational advantage with a programmable photonic processor. *Nature* **606**, 75–81 (2022).
43. Johnson, M. W. et al. Quantum annealing with manufactured spins. *Nature* **473**, 194–198 (2011).
44. Inagaki, T. et al. A coherent Ising machine for 2000-node optimization problems. *Science* **354**, 603–606 (2016).
45. McGeoch, C. C., Harris, R., Reinhardt, S. P. & Bunyk, P. I. Practical annealing-based quantum computing. *Computer* **52**, 38–46 (2019).
46. Aroyo, M. I. (ed.) *International Tables for Crystallography* Vol. A, 6th edn, Ch. 1.3 (Wiley, 2006).

Methods

Configuration spaces

Every periodic lattice atom allocation problem, and the resulting configuration space in our computations, is defined by the following parameters:

(1) The number of atoms to allocate, dimensions of the unit cell and the chosen force fields.
(2) The number of potential atomic positions within a unit cell, which is controlled by the discretization parameter g equal to the number of lattice points per cell side.
(3) The proximity constraints that specify how close two ionic species can be to each other.
(4) The desired space group of periodic allocations on a lattice.

To evaluate the applicability of the periodic lattice atom allocation for CSP, we have restricted ourselves to cubic structures and tested the prediction of known cubic materials by minimizing the optimal configurations for a variable unit cell size, symmetry and lattice discretization. Selection of parameters of this kind is an integral part of every CSP code. Thus, to assess integer programming encoding, rather than parameter screening, we limit ourselves to a range of parameters set around the experimentally determined value. The lattice parameters reported in Table 1 are the correct values rounded to the first decimal place. Additional computations (Extended Data Tables 1–5) suggest that any reasonable value close to the experimentally determined value will lead to the same outcome. The potential positions are uniformly distributed within the unit cell and their total number is equal to g^3. Symmetry constraints arise either from the space groups of the experimentally determined structures or from their subgroups. In all reported cases, there was at least one configuration space leading to a successful prediction.

Potential energy of a crystal

The Ewald summation[36] decomposes electrostatic interaction into three terms U^{self}, U^{real} and U^{recip}. This split is controlled by a real parameter α and the summation is performed either in real space over the unit cell copies labelled by n or over the reciprocal lattice vectors $\mathbf{m}$. We denote by V the volume of the unit cell, N the number of ions in it, q_i the charge of the ith ion, $r_{i,j,n}$ the distance between the ith ion in the original cell and the jth ion in the cell labelled by n and $\mathbf{r}_i$ the position vector of the ith ion in the original cell. The constituent parts are

$$U^{\text{self}} = -\frac{\alpha}{\sqrt{\pi}}\sum_{i=1}^{N} q_i^2,\ U^{\text{real}} = \frac{1}{2}\sum_{i,j=1}^{N'} q_i q_j \sum_{n} \frac{\operatorname{erfc}(\alpha r_{ij,n})}{r_{ij,n}},$$

$$U^{\text{recip}} = \frac{1}{2\pi V}\sum_{i,j=1}^{N} q_i q_j \sum_{\mathbf{m}\neq 0} \frac{\exp\left(-(\pi\mathbf{m}/\alpha)^2 + 2\pi i\mathbf{m}(\mathbf{r}_i - \mathbf{r}_j)\right)}{\mathbf{m}^2}.$$

Here, $\mathbf{m}^2$ corresponds to the usual scalar product and the summation up to N' excludes the case of $i = j$ for the original unit cell as it leads to division by zero. Because the reciprocal lattice vectors $\mathbf{m}$ and the distances $r_{i,j,n}$ are well defined as soon as the unit cell and the set of positions Pos are chosen, these sums can be rearranged and the coefficient in front of $q_i q_j$ can be computed independently of the atom allocation. By substituting the charges of elements typ_1, typ_2 for q_i, q_j and pos_1, pos_2 for their positions, we derive the electrostatic contribution part in $\alpha^{\text{typ}_1,\text{typ}_2}_{\text{pos}_1,\text{pos}_2}$. The repulsive part of the potential energy is obtained by direct summation because it has a finite range.

The repulsive contributions depend on the composition of the materials[47–51] and are provided for Y_2O_3, $Y_2Ti_2O_7$, SrO and $SrTiO_3$ (Extended Data Table 6), $MgAl_2O_4$ (Extended Data Table 7), $Ca_3Al_2Si_3O_{12}$ (Extended Data Table 8) and ZrO_2 and ZnS (Extended Data Table 9).

Local minimizations were performed using General Utility Lattice Program[52] and Atomic Simulation Environment[53]. Structures are visualized using Visualization for Electronic and Structural Analysis[54].

Solving integer programs

On a conceptual level, suppose that we are given a minimization problem that depends on binary variables $X^{\text{typ}}_{\text{pos}}$ with a feasible region defined by the constraints, as is the case for the periodic lattice atom allocation problem. Such optimization problems are typically computationally intractable in the sense of the theory of NP-completeness[19]. This theory provides strong evidence for the nonexistence of efficient exact algorithms that always succeed in solving these problems, as in the case of CSP[14]. However, this intractability refers to the difficulty of building universal algorithms that would solve all possible instances. But there are techniques that can efficiently solve most instances. Branch-and-cut is one of the efficient methods that deals with this intractability[6,24].

The first step is the relaxation of the problem to enlarge the set of feasible solutions, most often by eliminating some of the constraints, for example, by letting $X^{\text{typ}}_{\text{pos}}$ be between 0 and 1. The minimum value of the objective function E in such a relaxed feasibility region then drops down, resulting in a lower bound on the original optimal value. Such a lower bound is maintained and refined during the iterative optimization process. In addition to the lower bound, the solvers also compute and maintain an upper bound on the value of the objective E, which is its value at the best feasible solution to the problem so far. This optimization process can gradually tighten the relaxation by adding further valid constraints known as cuts that are feasible for the original binary problem but unfeasible for the relaxed problem. Together with the upper and lower bounds, this enables systematic and efficient exploration of the search space in pursuit of the optimal solution. Branching relies on constraints that divide the relaxed problem into separate subproblems, in which the feasible region of each subproblem is closer to a part of the original binary problem region. Which of those subproblems to explore is decided by the current lower and upper bounds. If the lower bound happens to be larger than the best-known upper bound, this implies that this particular subproblem cannot hold the original optimal binary solution and can thus be omitted. Finally, the optimality of the ultimate binary solution is certified by the fact that its objective value E matches the current lower bound.

The problem set out above is an example of a binary quadratic integer program, in which its variables $X^{\text{typ}}_{\text{pos}}$ appear in quadratic terms. Relaxations of such programs are usually achieved by powerful linearization techniques, in which each product of two variables is replaced by a single variable, thus obtaining a linear term in that new variable. This leads to a relaxed linear or semidefinite program[20], which can then be solved efficiently by a multitude of techniques, such as Dantzig's simplex method or interior point methods[19,55]. Solvers based on optimization theory thus lead to an exact solution to the integer program for which equation (1) is the objective function that considers the entire search space and guarantees the optimality of that solution. Beyond that, many-body potentials can be incorporated into equation (1) by adding products of more than two binary variables.

In practice, we were able to solve relatively large integer programs—for example, Y_2O_3 at $g = 16$ and $Ia\bar{3}$ symmetry contained 248 binary variables and 30,628 quadratic terms. The combinatorics influence the speed, as illustrated by the 100-fold increase in run time for $Z = 27$ $SrTiO_3$ when the space group symmetry is reduced from $Pm\bar{3}m$ to $P23$, reflecting the larger number of allocations possible on the same lattice. Moreover, as our encoding is largely oblivious to different representations of the same crystal, the number of different assignments of binary variables corresponding to equivalent crystals, including the global optimum, substantially increases. This generates equivalent solutions to the program, which are known to slow the branch-and-cut algorithm[7]. The presented integer program for the periodic lattice atom allocation

problem is only one of the many possible encodings and different encodings for the same optimization problem can lead to different run times and theoretical properties[6]. By developing specific encodings that address redundancies in periodic allocation problems, the branch-and-cut identification of the guaranteed minimum can be further accelerated.

Constraints for integer programs

The following constraints are always present and ensure that the resulting periodic lattice atom allocations are correct:

(1) Exclusivity. A constraint $\sum_{\text{typ}\in\text{Types}} X_{\text{pos}}^{\text{typ}} \le 1$ for every pos ∈ Pos to prevent different atoms from occupying the same position.

(2) Stoichiometry. A constraint $\sum_{\text{pos}\in\text{Pos}} X_{\text{pos}}^{\text{typ}} = C_{\text{typ}}$ for every species typ ∈ Types, where C_{typ} is the desired number of atoms of species typ within the unit cell.

Further constraints are used to focus the search on desired structure types and improve the running times

(1) Symmetry. Crystals have symmetry, so it is natural to constrain our search space by fixing the desired space group of the resulting allocation. The space group dictates which positions are symmetrically equivalent and must be occupied by the same chemical element, namely, these are positions belonging to the same crystallographic orbit. For every pair of such positions pos_1 and pos_2, we put $X_{\text{pos}_1}^{\text{typ}} = X_{\text{pos}_2}^{\text{typ}}$ for all typ ∈ Types. Note that we can, in principle, rewrite our encoding right away using these equalities by introducing new variables—one for every orbit and species. But modern solvers do this easily during the pre-solve stage while integrating other constraints as well.

(2) Proximity. The exclusivity constraint can be further strengthened to preclude ions from occupying positions that are too close to each other. To estimate the size of an ion in a typical crystal, we rely on Shannon ionic radii (see below for the values). For every pair of positions pos_1 and pos_2 that are closer than, for example, 75% of the sum of ionic radii of typ_1 and typ_2 atoms in our computations, we include the constraint $X_{\text{pos}_1}^{\text{typ}_1} + X_{\text{pos}_2}^{\text{typ}_2} \le 1$. The selected percentage value is a hyper-parameter used to accommodate lattice distortion and imprecision of radii estimation. Practically, we avoid addition of a large number of such constraints and achieve a similar effect by putting a large positive coefficient in front of the term $X_{\text{pos}_1}^{\text{typ}_1} X_{\text{pos}_2}^{\text{typ}_2}$ in the objective function. If both variables are non-zero, then the energy of such an arrangement is very high and it is excluded from consideration.

The following values of Shannon ionic radii were used during computations[56]: O^{2-}, 1.35 Å; S^{2-}, 1.84 Å; Al^{3+}, 0.39 Å; Y^{3+}, 0.9 Å; Mg^{2+}, 0.57 Å; Ti^{4+}, 0.42 Å; Ca^{2+}, 1.0 Å; Si^{4+}, 0.26 Å; Zr^{4+}, 0.59 Å; Zn^{2+}, 0.6 Å; Sr^{2+}, 1.18 Å.

Quantum computing

Equation (1) contains only products of at most two binary variables and thus can be seen as the objective function of a QUBO problem. Periodic lattice allocation problems have extra constraints such as exclusivity and stoichiometry that we need to eliminate to use quantum annealers. We mimic the effect of these constraints by modifying the objective function. By adding more penalty terms, we can ensure that an optimal solution of the modified program satisfies the constraints. Otherwise, its energy is too high to be optimal. The term $\mu \sum_{\text{typ}_1,\text{typ}_2} X_{\text{pos}}^{\text{typ}_1} X_{\text{pos}}^{\text{typ}_2}$, which is at least μ as soon as two different ion species are placed at pos, enforces the exclusivity condition if μ is a large-enough positive number. By introducing the term $\gamma\left(C_{\text{typ}} - \sum_{\text{pos}\in\text{Pos}} X_{\text{pos}}^{\text{typ}}\right)^2$ with a large-enough positive γ, we ensure that an optimal allocation must have the desired stoichiometry. This results in a QUBO problem that can be directly submitted to a quantum annealer or other hardware QUBO solvers[10,44] or addressed with quantum approximate optimization algorithms on gate-based quantum computers[57]. In practice, quantum annealers produce a range of allocations including the ones that violate these conditions, but their energies are high and they are excluded during postprocessing.

We solved a variety of simple structures on the 2000Q quantum annealer, which is freely available from D-Wave through Leap[45]. There are configuration spaces in which the quantum annealer has been able to identify the global optimum allocation that minimizes into the experimental structure, exactly as achieved with the classical computing approach. Specifically, we have obtained the rock salt structure of SrO ($g = 2$, $P23$), in which the actual space group $Fm\bar{3}m$ was replaced with $P23$ to increase the complexity of the problem, the perovskite structure of $SrTiO_3$ ($g = 2$, $Pm\bar{3}m$), the fluorite structure of cubic zirconia ZrO_2 ($g = 4$, $P2_13$) and the wurtzite structure of ZnS ($g = 4$, $P23$). We used space group symmetry and a small number of lattice positions to limit the size of the resulting integer programs. Each optimization run of the quantum annealer used up to 168 qubits and produced hundreds of allocations in milliseconds including several allocations corresponding to the global optimum. Although the current generation of quantum annealers is limited in terms of the size of programs that they can run—because of a lack of free qubits and their connectivity—and cannot guarantee optimality owing to their high sensitivity to the noisy environment, the technology is being actively developed to overcome these drawbacks[58].

We have deliberately used integer programs of the same form as used in classical computing for clarity and simplicity. The only adjustments were the annealing schedules, which control the optimization process on the quantum computer, and the penalty terms used during constraint elimination of the resulting integer programs (Extended Data Table 10). These parameters had a substantial impact on the outcomes of predictions. The QUBO formulation introduced here can be further adjusted to better suit the quantum annealer. Performance will probably improve by designing CSP encodings that are well suited to the noisy intermediate-scale quantum computers, with the aim of effectively recovering optimality in almost all cases[59]. Hybrid quantum-classical computation[60] could provide notable advantages of quantum computing for CSP even before the full potential of this technology is realized.

Equipment

Our computational experiments were done on a 40-core workstation (two 20-core Intel Xeon E5-2630v4 CPUs) running at 2.2 GHz with 64 Gb of RAM. Gurobi 9.5 was the integer programming solver used. Quantum computations were done on the D-Wave 2000Q quantum annealer through Leap using the associated application programming interface (https://docs.ocean.dwavesys.com/en/stable/).

Data availability

The authors declare that the data supporting the findings of this study are available in the paper and Supplementary Information.

Code availability

An implementation of the integer programming encoding for the periodic lattice allocation problem and subsequent CSP is available at https://github.com/lrcfmd/ipcsp.

47. Collins, C., Darling, G. R. & Rosseinsky, M. J. The Flexible Unit Structure Engine (FUSE) for probe structure-based composition prediction. *Faraday Discuss.* **211**, 117–131 (2018).
48. Binks, D. J. *Computational Modelling of Zinc Oxide and Related Oxide Ceramics*. PhD thesis, Univ. Surrey, (1994).
49. Pedone, A., Malavasi, G., Menziani, M. C., Cormack, A. N. & Segre, U. A new self-consistent empirical interatomic potential model for oxides, silicates, and silica-based glasses. *J. Phys. Chem. B* **110**, 11780–11795 (2006).
50. Woodley, S. M., Battle, P. D., Gale, J. D. & Catlow, C. R. A. The prediction of inorganic crystal structures using a genetic algorithm and energy minimisation. *Phys. Chem. Chem. Phys.* **1**, 2535–2542 (1999).
51. Wright, K. & Jackson, R. A. Computer simulation of the structure and defect properties of zinc sulfide. *J. Mater. Chem.* **5**, 2037–2040 (1995).
52. Gale, J. D. & Rohl, A. L. The General Utility Lattice Program (GULP). *Mol. Simul.* **29**, 291–341 (2003).

53. Larsen, A. H. et al. The atomic simulation environment—a Python library for working with atoms. *J. Phys. Condens. Matter* **29**, 273002 (2017).
54. Momma, K. & Izumi, F. *VESTA 3* for three-dimensional visualization of crystal, volumetric and morphology data. *J. Appl. Crystallogr.* **44**, 1272–1276 (2011).
55. Boyd, S. & Vandenberghe, L. *Convex Optimization* (Cambridge Univ. Press, 2004).
56. Shannon, R. D. Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides. *Acta Crystallogr. A* **32**, 751–767 (1976).
57. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at https://arxiv.org/abs/1411.4028 (2014).
58. Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H. & Oliver, W. D. Perspectives of quantum annealing: methods and implementations. *Rep. Prog. Phys.* **83**, 054401 (2020).
59. Bian, Z. et al. Solving SAT (and MaxSAT) with a quantum annealer: foundations, encodings, and preliminary results. *Inf. Comput.* **275**, 104609 (2020).
60. Ajagekar, A., Humble, T. & You, F. Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Comput. Chem. Eng.* **132**, 106630 (2020).

Acknowledgements We thank the Leverhulme Trust for funding through the Leverhulme Research Centre for Functional Materials Design. V.V.G. thanks M. W. Gaultois for discussions. We thank R. Savani for feedback on the paper.

Author contributions All authors took part in discussions to frame the use of modern optimization approaches in CSP. V.V.G. and A.D. conceptualized the idea of periodic lattice atom allocation. V.V.G. developed Ewald summation and QUBO encodings, implemented the approach and evaluated it on classical computers. D. Antypov and C.M.C. performed supplementary analysis of resulting structures. V.V.G. suggested the use of quantum annealers; V.V.G. and D. Adamson performed evaluation. V.V.G., A.D., D. Antypov, M.S.D. and M.J.R. wrote the first draft of the paper. V.V.G., D. Adamson, C.M.C., P.K., I.P., P.S. and M.J.R. wrote the final draft of the paper. All authors were involved in discussions and evaluation of drafts during the writing process. P.S. and M.J.R. directed the research.

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/s41586-023-06071-y.

Correspondence and requests for materials should be addressed to Paul Spirakis or Matthew J. Rosseinsky.

Peer review information *Nature* thanks C. Richard Catlow and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Extended Data Table 1 | The change in energy of the optimal solution of the periodic lattice atom allocation problem for $SrTiO_3$ with $g=4$ and $P23$ (195) space group symmetry constraint under varying unit cell size

Unit cell size	-30%	-20%	-10%	Correct	+10%	+20%	+30%
eV/atom	infeasible	-22.989	-30.186	-31.704	-30.907	-29.273	-27.441

If the unit cell is too small to accommodate all the ions while satisfying proximity constraints, then no solution is returned. This is the case for the unit cell that is 30% smaller than the experimentally determined structure (2.7 Å vs 3.9 Å). All other periodic lattice atom allocations listed above are locally optimised into the same correct structure despite having very different unit cell sizes (and energies), indicating that exact knowledge of cell parameters is not necessary for a successful application of this technique.

Extended Data Table 2 | The change in energy of the optimal solution of the periodic lattice atom allocation problem for $MgAl_2O_4$ with $g=8$ and $Fd\bar{3}m$ (227) space group symmetry constraint under varying unit cell size

Unit cell size	-20%	-10%	Correct	+10%	+20%
eV/atom	-23.091	-27.254	-27.905	-27.098	-25.735

All optimal configurations minimise into the correct spinel structure with the energy −28.944 eV/atom.

Extended Data Table 3 | The change in energy of the optimal solution of the periodic lattice atom allocation problem for Y_2O_3 with g=16 and $Ia\bar{3}$ (206) space group symmetry constraint under varying unit cell size

Unit cell size	-5%	Correct	+10%	+15%	+20%
eV/atom	-25.252	-26.262	-26.128	-26.251	-25.876

All optimal configurations minimise into the correct bixbyite structure with the energy −27.395 eV/atom. Note that the global optimal solution for the periodic lattice atom allocation problem for Y_2O_3 with the unit cell size -10% minimises into a structure with the energy −26.392 eV/atom, which is higher by 1.002 eV/atom than the correct structure, indicating that the optimal configuration has been changed.

Extended Data Table 4 | The change in energy of the optimal solution of the periodic lattice atom allocation problem for $Y_2Ti_2O_7$ with g=16 and $Fd\bar{3}m$ (227) symmetry constraint under varying unit cell size

Unit cell size	-15%	-10%	Correct	+5%	+10%
eV/atom	-28.680	-32.471	-35.002	-34.919	-34.388

All optimal configurations minimise into the correct pyrochlore structure with the energy −35.154 eV/atom.

Extended Data Table 5 | The change in the optimal solution of the periodic lattice atom allocation problem under varying unit cell size for $Ca_3Al_2Si_3O_{12}$, g=16, $Ia\bar{3}d$ (230)

Unit cell size	-3%	-2%	Correct	+5%	+10%	+15%
eV/atom	-12.109	-12.176	-12.277	-12.327	-12.253	-12.302

All optimal configurations relax into the correct garnet structure with the energy −13.750 eV/atom.

Extended Data Table 6 | Buckingham potential parameters for Y_2O_3, $Y_2Ti_2O_7$, SrO and $SrTiO_3$ with the cut-off radius of 10Å

Interaction	A (eV)	ρ (Å)	C (eV Å^{-6})
$Y^{3+} – O^{2-}$	23000	0.24203	0
$Sr^{2+} – O^{2-}$	1952.39	0.33685	19.22
$Ti^{4+} – O^{2-}$	4590.7279	0.261	0
$O^{2-} – O^{2-}$	1388.77	0.36262	175

Data are from ref. 47.

Extended Data Table 7 | Buckingham potential parameters for $MgAl_2O_4$

Interaction	A (eV)	ρ (Å)	C (eV Å^{-6})
$Mg^{2+} – O^{2-}$	1284.380	0.2997	0
$Al^{3+} – O^{2-}$	1725.000	0.2897	0
$O^{2-} – O^{2-}$	9547.960	0.2192	32.00

Data are from ref. 48.

Extended Data Table 8 | Force-field parameters for $Ca_3Al_2Si_3O_{12}$

Interaction	D_{ij} (eV)	a_{ij} (Å^{-2})	r_0 (Å)	C_{ij} (eV Å^{12})
$Ca^{1.2+} – O^{1.2-}$	0.030211	2.241334	2.923245	5.00
$Si^{2.4+} – O^{1.2-}$	0.340554	2.006700	2.100000	1.00
$Al^{1.8+} – O^{1.2-}$	0.361581	1.900442	2.164818	0.90
$O^{1.2-} – O^{1.2-}$	0.042395	1.379316	3.618701	22.00

This parameter set uses partial charge states on the ions. The repulsive part is a combination of Morse and Lennard-Jones potentials[49] defined as follows:
$D_{ij}((1-\exp(-a_{ij}(r-r_0)))^2-1)+(C_{ij}/r^{12})$.

Extended Data Table 9 | Buckingham potential parameters for ZrO_2 and ZnS

Interaction	A (eV)	ρ (Å)	C (eV Å^{-6})
$Zr^{4+} – O^{2-}$	7290.347	0.2610	0
$O^{2-} – O^{2-}$	25.410	0.6937	32.32
$Zn^{2+} – S^{2-}$	613.356	0.3990	0
$S^{2-} – S^{2-}$	1200.000	0.1490	120.0

Data are from refs. 50,51.

Extended Data Table 10 | Parameters of the quantum annealing runs

Crystal structure	μ	γ	Annealing time, μs
SrO	100	100	200
$SrTiO_3$	100	100	200
ZrO_2	50	50	1000
ZnS	100	100	200

In every case, the quantum annealer produced the correct periodic lattice atom allocation. The number of reads were between 100 and 300. Annealing time is given in microseconds. The presented coefficients μ and γ that were used to offset proximity and stoichiometry constraints to the objective function are defined in Methods.