Science and Technology of Advanced Materials

ISSN: 1468-6996 (Print) 1878-5514 (Online) Journal homepage: http://www.tandfonline.com/loi/tsta20

MDTS: automatic complex materials design using Monte Carlo tree search

Thaer M. Dieb, Shenghong Ju, Kazuki Yoshizoe, Zhufeng Hou, Junichiro Shiomi & Koji Tsuda

To cite this article: Thaer M. Dieb, Shenghong Ju, Kazuki Yoshizoe, Zhufeng Hou, Junichiro Shiomi & Koji Tsuda (2017) MDTS: automatic complex materials design using Monte Carlo tree search, Science and Technology of Advanced Materials, 18:1, 498-503, DOI: 10.1080/14686996.2017.1344083

To link to this article: http://dx.doi.org/10.1080/14686996.2017.1344083

SCIENCE AND TECHNOLOGY OF ADVANCED MATERIALS, 2017
VOL. 18, NO. 1, 498–503
https://doi.org/10.1080/14686996.2017.1344083

Taylor & Francis
Taylor & Francis Group

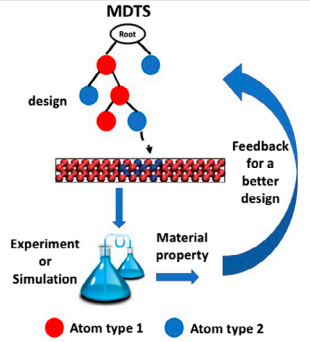
MDTS: automatic complex materials design using Monte Carlo tree search

Thaer M. Dieb[a,b], Shenghong Ju[c], Kazuki Yoshizoe[d], Zhufeng Hou[a], Junichiro Shiomi[a,c] and Koji Tsuda[a,b,d]

[a]National Institute for Materials Science, Tsukuba, Japan;
[b]Graduate School of Frontier Sciences, The University of Tokyo, Kashiwa, Japan;
[c]Department of Mechanical Engineering, The University of Tokyo, Tokyo, Japan;
[d]RIKEN, Center for Advanced Intelligence Project, Tokyo, Japan

ABSTRACT

Complex materials design is often represented as a black-box combinatorial optimization problem. In this paper, we present a novel python library called MDTS (Materials Design using Tree Search). Our algorithm employs a Monte Carlo tree search approach, which has shown exceptional performance in computer Go game. Unlike evolutionary algorithms that require user intervention to set parameters appropriately, MDTS has no tuning parameters and works autonomously in various problems. In comparison to a Bayesian optimization package, our algorithm showed competitive search efficiency and superior scalability. We succeeded in designing large Silicon-Germanium (Si-Ge) alloy structures that Bayesian optimization could not deal with due to excessive computational cost. MDTS is available at https://github.com/tsudalab/MDTS.



ARTICLE HISTORY
Received 25 April 2017
Revised 29 May 2017
Accepted 15 June 2017

KEYWORDS
Materials informatics; Materials design; Monte Carlo tree search; Si-Ge alloy interfacial structure; Python library

CLASSIFICATION
60 New topics/Others; 600 Others: Materials informatics

1. Introduction

Complex materials design is a key topic in materials science and engineering. The design of a complex materials' structure that meets certain criteria is often formulated as the problem of finding the optimal solution from a space of candidates [1,2]. A common problem in solid-state materials design is the structure determination of a substitutional alloys problem [3,4], where atoms or vacancies are assigned to positions in a crystal structure. For example, Ju et al. [4] recently solved the optimal assignments of Silicon (Si) and Germanium (Ge) to a certain crystal structure that achieves minimum and maximum thermal conductance.

To accelerate the materials design process, several experimental design algorithms have been used to find the optimal structure with as few experiments as possible (Figure 1). Experimental design is an iterative process for selecting the next candidates for experiments, where the outcome of the experiments are exploited for making further choices. In many cases, simulators are substituted to experiments, e.g. first-principle calculations. In earlier studies, quantitative structure-property relationship (QSAR) models were mainly used [5]. Recently, Bayesian optimization [6], a technique to select promising candidates using Bayesian learning, has been proven as an effective tool in materials design [1,2,4,7–9]. The difference between Bayesian optimization methods and traditional QSAR models is that the uncertainty of prediction is quantified as predictive variance: the candidates are scored by an acquisition function that takes into account both predicted merit and uncertainty. Bayesian optimization is very effective in finding optimal structures but has problems with scalability, as the acquisition function has to be applied to all candidates. Evolutionary algorithms such as genetic algorithms [10,11] are more scalable, but have many parameters, such as crossover and mutation rates, that must be tuned properly to obtain the best

CONTACT Koji Tsuda tsuda@k.u-tokyo.ac.jp
Supplemental data for this article can be accessed https://doi.org/10.1080/14686996.2017.1344083.

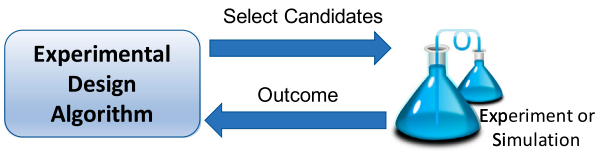


Figure 1. Materials design by an experimental design algorithm. The process starts with an initial random design. The algorithm selects the next candidates for experiments, where the outcome of the experiments are exploited by the algorithm to make further selection.

performance. In most cases, in materials design, the amount of data available a priori is very limited, so tuning parameters using data may not be possible.

In this paper, we propose a novel python library called Materials Design using Tree Search (MDTS). MDTS solves structure determination of substitutional alloys with composition constraints using a Monte Carlo tree search [12], a guided-random best-first search method that showed significant success in computer Go [12,13]. Our library is highly scalable and does not have any tuning parameter.

In experiments, we applied MDTS and an efficient Bayesian optimization implementation [7] to a Si-Ge alloy interface design between two Si leads [4]. The local force field (bonding characteristics) in the structure can change due to substitution. However, in this demonstration case, we did not consider structure relaxation because the force constants of Si and Ge are known to be transferable [14]. On the other hand, there are ways to include the change in the local force constants and the current method can be simply used to incorporate such an effect [15]. The total computational time is decomposed into design time and simulation time. The former represents the selection of the next candidates and the latter simulator time. In terms of the number of calculations to find the optimal solution, Bayesian optimization was better due to its high prediction ability. However, MDTS was comparable or better in terms of total computational time, because Bayesian optimization takes exponential design time with respect to the number of atoms. MDTS is a practical tool that material scientists can easily deploy in their own problems and has the potential to become a standard choice.

2. Method

Consider a black-box function, $f(\mathbf{x})$, where $\mathbf{x}$ is a vector of discrete variables $\mathbf{x} \in \{0, 1, k-1\}^N$. We aim to find the optimal solution $\mathbf{x}^*$ that maximizes $f(\mathbf{x})$ subject to composition constraints

$$\sum_{\ell=1}^{N} I(x_\ell = j) = n_j, \quad j = 0, \ldots, k-1 \tag{1}$$

where I is the indicator function that returns one if the given condition is satisfied and zero otherwise. The constant n_i indicates the number of variables with value i. Notice that $\sum_{j=0}^{k-1} n_j = N$. In an atom assignment problem, $\mathbf{x}$ corresponds to atom types and $f(\mathbf{x})$ is a target property evaluated, for example, through first-principles calculations.

Monte Carlo tree search (MCTS) employs a search tree, where nodes at the ℓth level correspond to value assignment to x_ℓ (Figure 2). A path from the root to a node at level ℓ corresponds to a partial solution with respect to $x_1, \ldots, x_\ell$. In the first round of MCTS, only the root node exists and then the search tree is gradually constructed. To obtain a full solution $\mathbf{x}$, a complete path to a leaf node at the Nth level is necessary. One interesting feature of MCTS is that only a shallow tree is built and the complete paths are obtained via random playouts [12]. A 'playout' creates a solution by starting from a node and determining the remaining variables randomly. The random playout allows us to explore a large candidates space without learning from data. Once a solution has been obtained by a playout, the black-box function $f(\mathbf{x})$ is evaluated and recorded. By combining tree expansion, backtracking and playouts, a large candidate space can be searched systematically. When a predetermined number of calculations is reached, the best solution so far is returned as the final result.

Each node i contains three variables: the visit count v_i represents the number of visits in the search process; f_i denotes the immediate merit of node i evaluated by playout; and the cumulative merit w_i is defined as the sum of all direct merit for all descendant nodes including itself. The Upper Confidence Bound (UCB) score [12] of a node is an index indicating how promising it is to explore the subtree under the node. It is defined based on the cumulative merit and the number of visits as follows:

$$u_i = \frac{w_i}{v_i} + C\sqrt{\frac{2\ln v_{parent}}{v_i}} \tag{2}$$

where C is the constant to balance exploration and exploitation and v_{parent} is the visit count of the parent node. Whenever a new node is added, the variables are initialized as

$$v_i = w_i = f_i = 0, u_i = \infty \tag{3}$$

Each round of MCTS consists of: selection, expansion, simulation and back propagation (Figure 2). In the selection step, the tree is traversed from the root to a leaf by choosing the child with the maximum UCB score at each branch. If there is a tie, the winning child is chosen randomly. Let i denotes the identified leaf, ℓ the level of the node i, $x_1, \ldots, x_\ell$ the partial solution corresponding to the path from the root to i. In the expansion step, children nodes are added under the node i. If the number of atoms j reaches the limit

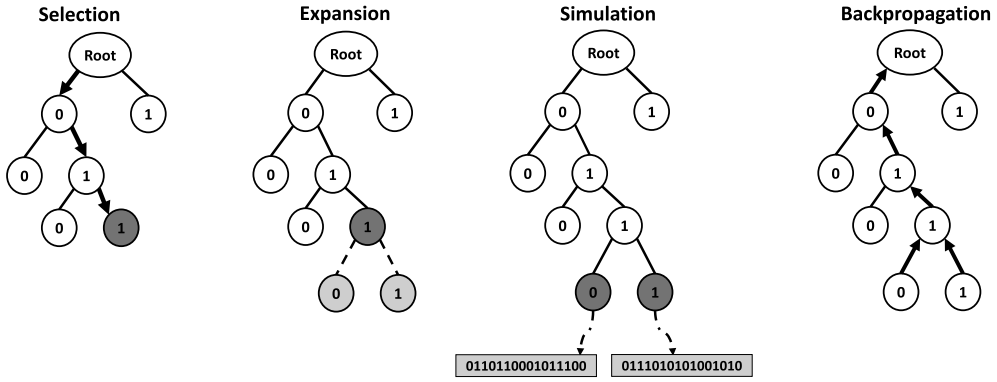


Figure 2. Monte Carlo tree search (MCTS) for a binary atom assignment problem. The candidate space is represented as a tree where each node represents a possible atom assignment. One round of MCTS consists of four steps, Selection, Expansion, Simulation and Backpropagation. In the selection step, a promising leaf node is chosen by following the node with the best UCB score in each branch. The expansion step adds a number of children nodes to the selected one. In simulation, solutions are created by random playouts from the expanded nodes. The backpropagation step updates nodes' information along the path back to the root.

already, i.e. $\sum_{l=1}^{\ell} I(x_l = j) = n_j$ the jth child is not added. In the simulation step, a playout is performed from each of the added children. Notice that the random assignments are made such that the composition constraints are satisfied. With the solutions obtained, a simulator is applied to evaluate $f(\mathbf{x})$ and store the value as the immediate merit of the corresponding nodes. Finally, in the back propagation step, the visit count of each ancestor node of i is incremented by one and the cumulative value is also updated to keep consistency.

The value of C crucially affects the performance of MDTS. According to the analysis by Kocsis and Szepesvári [16], to guarantee the convergence to the optimal solution, C should be proportional to the range between z_{max} and z_{min}, i.e. the maximum and minimum immediate merit observed in downstream nodes. Adjusting C, either statically or dynamically, is a standard technique for applying MCTS (as shown in [12]). Following a similar idea, MDTS controls C adaptively at each node as follows:

$$C = \frac{\sqrt{2}J}{4}(z_{max} - z_{min}) \tag{4}$$

where J is a meta-parameter initially set to one and increased whenever the algorithm encounters a 'dead-end' leaf, to allow more exploration. At a dead-end leaf, the number of possible structures narrows to one. This happens when the numbers of $k-1$ atoms reaches the limit. J is updated as $J \leftarrow J + \max\{\frac{T-t}{T}, 0.1\}$, where T is the total number of candidates to be evaluated and t is the number of candidates for which the black-box function is evaluated. See supplemental material for the algorithm.

3. Experiments and results

In this section, we compare MDTS to a Bayesian optimization package called COMBO [7] in a binary atom assignment problem (notice that MDTS is able to handle multiple atom types assignment problems). The performance of MDTS depends on the variable ordering in $\mathbf{x}$. The following three options were tried: direct (left-to-right), reversed (right-to-left) and random.

MDTS and COMBO were applied to design optimal Si-Ge alloy (Si:Ge=1:1) interfacial structures (Figure 3) with both minimum and maximum thermal conductance [4]. Materials with both minimum (e.g. thermoelectric materials) and maximum (e.g. CPU cooling) interfacial thermal conductance have potential applications. As shown in Figure 3, the system consists of an interface region between two Si leads with infinite thickness. In the interface, there are N positions filled either by Si or Ge. The number of atoms of each type is constrained to $N/2$. The number of possible structures grows rapidly as the number of atoms N increases. For example, at 14, 20 and 26 atoms, the number of possible structures is 3432, 184,756 and 10,400,600,

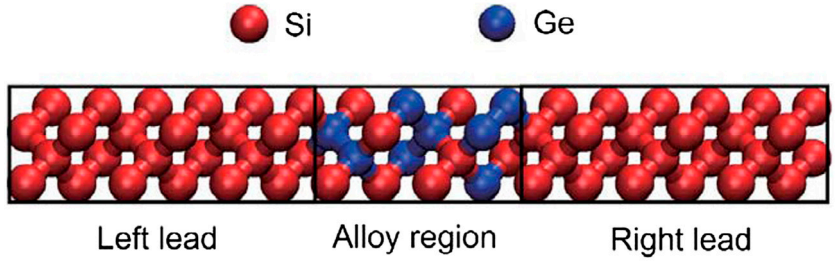


Figure 3. Si-Ge interfacial structure between two Si leads. In this case, the interface region is made up of 16 atoms.

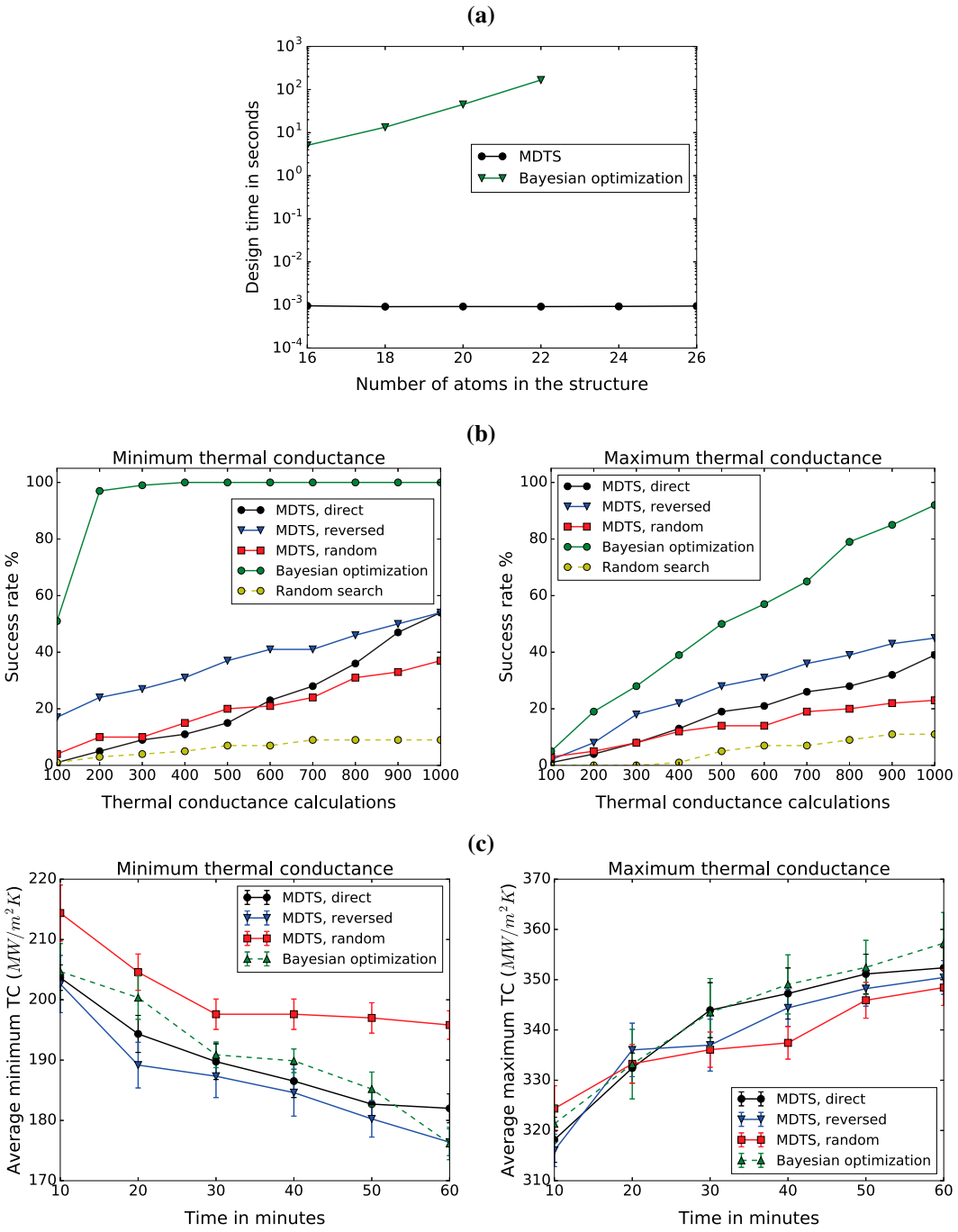


Figure 4. Comparison between MDTS and Bayesian optimization (BO) in finding the structure with minimum and maximum thermal conductance. (a) Design time for choosing a candidate structure against the number of atoms in the interfacial structure N. The time for BO grows exponentially as N increases. Results averaged over 10 runs, each for 30 solutions. (b) The fraction of optimal structure discovery (i.e. success rate) for both minimum and maximum thermal conductance in 100 runs against the number of thermal conductance calculations. The number of atoms is 16 ($N = 16$). BO takes fewer calculations to find the optimal structure. (c) Optimal observed thermal conductance (minimum and maximum) against total computational time including both design and simulation time ($N = 22$). The result is averaged over 10 runs. Here, the efficiency of the two methods is comparable. For $N < 22$, BO was more efficient and MDTS was more efficient for $N > 22$.

respectively. The thermal conductance was computed using the atomistic Green's function implemented in the ATK-Classical Simulator of Atomistix ToolKit (ATK) [17,18]. SiGe Tersoff [19,20] potential was used to describe the atom interactions. The size of the supercell in the transverse direction (perpendicular to the direction of heat conduction) is 1 unit cell, i.e. 5.43 Å × 5.43 Å, and periodic boundary conditions were used. See Ref. [4] for further details.

Since the process of simulation-based structure optimization involves an experimental design algorithm and a simulation algorithm, the total computational time is divided into two parts: *design time* and *simulation time*. The design time per structure against the

number of atoms is shown in Figure 4(a). Bayesian optimization shows an exponential increase in design time, because it needs to compute a score for every candidate structure. On the other hand, the design in MDTS takes only a tree traversal, whose computational cost is scarcely affected by the number of atoms. Figure 4(b) shows the fraction of optimal structure discovery over 100 runs (i.e. success rate) for both minimum and maximum thermal conductance against the number of thermal conductance calculations at $N = 16$. Bayesian optimization required a smaller number of calculations to achieve the same level of success rate due to its sophisticated prediction algorithm. Nevertheless, the performance of MDTS was better than random search, indicating its substantial capability of learning from data. Among the three variable orderings of MDTS, the reversed order was best. Random order performance was lowest in this particular case, likely because the existence of neighbourhood relations may be crucial for the optimal thermal conductance. Despite better learning capability, the advantage of Bayesian optimization in total computational time is rapidly wiped out, as N increases, because of the exponentially increasing design time. At $N = 22$, the speed of thermal conductance minimization and maximization of MDTS and Bayesian optimization is comparable as shown in Figure 4(c). At $N = 26$, however, Bayesian optimization becomes significantly slower: it takes about 15 times more time than the $N = 22$ case. This result shows that MDTS should be chosen over Bayesian optimization unless the problem size is sufficiently small.

4. Conclusion

In this paper, we presented MDTS: a materials design library based on Monte Carlo tree search. MDTS is an open source project and interested researchers can join in the development of MDTS. The balance between design time and simulation time is an important factor in automatic materials design. Efficient design methods including MDTS are most useful when the simulation time is short. The long design time of a more inefficient machine-learning based approach can appear less problematic when the simulation time is longer. In future work, it would be necessary to pursue an adaptive approach that can balance optimality and design time in a variable manner. Additionally, we plan to make MDTS more customizable for diverse materials design problems with possibly different kinds of constraints.

Acknowledgements

We would like to thank David A. duVerle for fruitful discussions. We also would like to thank anonymous referees for their comments and suggestions to improve the manuscript.

Disclosure statement

Authors declare no conflict of interest.

Funding

This work was supported by the 'Materials research by Information Integration' Initiative (MI2I) project and CREST [grant number JPMJCR16Q5] from Japan Science and Technology Agency (JST). It was also supported by Grant-in-Aid for Scientific Research on Innovative Areas 'Nano Informatics' [grant number 25106005] from the Japan Society for the Promotion of Science (JSPS).

References

[1] Seko A, Togo A, Hayashi H, et al. Prediction of low-thermal-conductivity compounds with first-principles anharmonic lattice-dynamics calculations and bayesian optimization. Phys Rev Lett. 2015;115:205901.
[2] Balachandran PV, Xue D, Theiler J, et al. Adaptive strategies for materials design using uncertainties. Sci Rep. 2016;6:19660.
[3] Okhotnikov K, Charpentier T, Cadars S. Supercell program: a combinatorial structure-generation approach for the local-level modeling of atomic substitutions and partial occupancies in crystals. J Cheminf. 2016;8(1):17. DOI: 10.1186/s13321-016-0129-3
[4] Ju S, Shiga T, Feng L, et al. Designing nanostructures for phonon transport via Bayesian optimization. Phys Rev X. 2017;7:021024.
[5] Coulinga D, Bernotb R, Dochertyb KM, et al. Assessing the factors responsible for ionic liquid toxicity to aquatic organisms via quantitative structure-property relationship modeling. Green Chem. 2006;8:82–90.
[6] Snoek J, Larochelle H, Adams R. Practical Bayesian optimization of machine learning algorithms. Adv Neural Inf Process Syst. 2012;2951–2959.
[7] Ueno T, Rhone T, Hou Z, et al. COMBO: an efficient Bayesian optimization library for materials science. Mater Discov. 2016;4:18–21.
[8] Seko A, Maekawa T, Tsuda K, et al. Machine learning with systematic density-functional theory calculations: application to melting temperatures of single-and binary-component solids. Phys Rev B. 2014;89:054303.
[9] Kiyohara S, Oda H, Tsuda K, et al. Acceleration of stable interfacestructure searching using a kriging approach. Jpn J Appl Phys. 2016;55:045502.
[10] Patra TK, Meenakshisundaram V, Hung J, et al. Neural-network-biased genetic algorithms for materials design: Evolutionary algorithms that learn. ACS Comb Sci. 2017;19(2):96–107.
[11] Paszkowicz W, Harris KD, Johnston RL. Genetic algorithms: a universal tool for solving computational tasks in materials science. Comput Mater Sci. 2009;45(1):ix–x.
[12] Browne C, Powley E, Whitehouse D, et al. A survey of Monte Carlo tree search methods. IEEE Trans Comput Intell AI in Games. 2012;4(1):1–43.
[13] Silver D, Huang A, Maddison C, et al. Mastering the game of Go with deep neural networks and tree search. Nature. 2016;529:484–489.
[14] Murakami T, Hori T, Shiga T, et al. Probing and tuning inelastic phonon conductance across finite-thickness interface. Appl Phys Express. 2014;7:121801.
[15] Murakami T, Shiga T, Hori T, et al. Importance of local force fields on lattice thermal conductivity reduction in $PbTe_{1-x}Se_x$ alloys. EPL. 2013;102:46002.
[16] Kocsis L, Szepesvári C. Bandit based monte-carlo planning. European conference on machine learning. Berlin: Springer; 2006. p. 282–293.

[17] Griebel M, Hamaekers J. Molecular dynamics simulations of the elastic moduli of polymer-carbon nanotube composites. Comput Meth Appl Mech Eng. 2004;193:1773–1788.
[18] Griebel M, Knapek S, Zumbusch G. Numerical simulation in molecular dynamics. Vol. 5, Texts in computational science and engineering. Springer-Verlag, Berlin Heidelberg; 2007.
[19] Tersoff J. Modeling solid-state chemistry: interatomic potentials for multicomponent systems. Phys Rev B. 1989;39:5566(R).
[20] Tersoff J. Erratum: Modeling solid-state chemistry: interatomic potentials for multicomponent systems. Phys Rev B. 1990;41:3248.