

LETTER • OPEN ACCESS

Reinforcement learning in discrete action space applied to inverse defect design

To cite this article: Troy D Loeffler *et al* 2021 *J. Phys. Commun.* **5** 031001

View the [article online](#) for updates and enhancements.



LETTER

OPEN ACCESS

RECEIVED

1 December 2020

REVISED

8 February 2021

ACCEPTED FOR PUBLICATION

11 February 2021

PUBLISHED

22 March 2021

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Reinforcement learning in discrete action space applied to inverse defect design

Troy D Loeffler^{1,2,4}, Suvo Banik^{1,2,4}, Tarak K Patra^{2,3}, Michael Sternberg² and Subramanian K R S Sankaranarayanan^{1,2}

¹ Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, Illinois 60607, United States of America

² Center for Nanoscale Materials, Argonne National Laboratory, Lemont, Illinois 60439, United States of America

³ Department of Chemical Engineering, Indian Institute of Technology Madras, Chennai, Tamil Nadu 600036, India

⁴ Both the authors contributed equally.

E-mail: skrssank@uic.edu

Keywords: machine learning, defect design, inverse design

Supplementary material for this article is available [online](#)

Abstract

Reinforcement learning (RL) algorithms that include Monte Carlo Tree Search (MCTS) have found tremendous success in computer games such as Go, Shiga and Chess. Such learning algorithms have demonstrated super-human capabilities in navigating through an exhaustive discrete action search space. Motivated by their success in computer games, we demonstrate that RL can be applied to inverse materials design problems. We deploy RL for a representative case of the optimal atomic scale inverse design of extended defects via rearrangement of chalcogen (e.g. S) vacancies in 2D transition metal dichalcogenides (e.g. MoS₂). These defect rearrangements and their dynamics are important from the perspective of tunable phase transition in 2D materials i.e. 2H (semi-conducting) to 1T (metallic) in MoS₂. We demonstrate the ability of MCTS interfaced with a reactive molecular dynamics simulator to efficiently sample the defect phase space and perform inverse design—starting from randomly distributed S vacancies, the optimal defect rearrangement of defects corresponds a line defect of S vacancies. We compare MCTS performance with evolutionary optimization i.e. genetic algorithms and show that MCTS converges to a better optimal solution (lower objective) and in fewer evaluations compared to GA. We also comprehensively evaluate and discuss the effect of MCTS hyperparameters on the convergence to solution. Overall, our study demonstrates the effectiveness of using RL approaches that operate in discrete action space for inverse defect design problems.

Introduction

Defect dynamics and optimization in 2D transition metal dichalcogenides (TMDs) significantly impact the observed electronic, optical, mechanical and chemical properties [1–4]. A myriad of structural defects can either pre-exist or be introduced in these 2D materials during sample preparation, processing and transfer processes [5, 6]. One finds a significant variation in the nature of the observed defects in these TMDC's—from point (e.g., chalcogen vacancies) to extended (e.g., dislocations and boundaries) defects, which in turn impact their functionality. For example, electrical properties for thin sheets of MoS₂ almost universally reveal *n*-type field effect transistor (FET) characteristics due to chalcogen vacancies, impurities, and metal-like antisite defects pinning the Fermi level of the metal at the metal/TMD contact interfaces [7]. In some cases, the defects are considered detrimental (e.g. electronics) whereas in others they are often beneficial (e.g. catalysis). Exercising precise control over the density and distribution of defects is therefore highly desirable.

The most abundant type of defect in TMDs, such as MoS₂, are the chalcogen (sulphur) mono-vacancies. During sample processing, transfer as well as during the operation of a device, these defects undergo significant reorganization. It has been shown that the S monovacancies can align together to form an extended line defect of

vacancies, which can mediate the cross-over between 2H (semiconductor) and 1T (metallic) phases of MoS₂ [8, 9]. Structurally, 2H phase possesses a trigonal prismatic arrangement of molybdenum (Mo) atom sandwiched between two sulphur (S) atoms, while 1T phase exhibits an octahedral coordination. The transition between 2H and 1T phases involve local rearrangement of S atoms with respect to their central Mo atom. From the perspective of tunable 2H-1T transition, it is desirable to attain a fundamental understanding of the atomic-scale structure and dynamics of defects in 2D TMDs, as well as their role in driving such structural phase transitions. A first step towards achieving this goal is to identify the optimal configurations of the defects in 2D materials, which is the central aim of this paper.

Time scale of defect dynamics in TMDs varies from seconds to hours depending on experimental conditions. Therefore, studying such a long-time process via standard atomistic molecular dynamics simulation is not tractable. Here, we attempt to employ data driven advanced optimization methods such as evolutionary computing and AI to tackle such long timescale phenomena in materials system. Our objective is to develop computational approaches that are suitable for capturing structural phase transitions that involve largescale spatiotemporal rearrangement of atoms and molecules. In particular, we employ and compare the performance of an AI algorithm viz., Monte Carlo Tree Search (MCTS) [10] and an evolutionary optimizer viz., genetic algorithm (GA) [11] in searching the most energetically favorable distribution of atomic defects starting from random defects in a MoS₂ monolayer. We use optimization of the S vacancies in MoS₂ as a representative example. In both the cases, the energy evaluation of the individual candidate structures (leaf nodes for MCTS or genes for evolutionary algorithm) are done using a reactive force field. We discuss the performance and limitations of the two methods in the broader context of inverse materials design problems.

Methods

Monte Carlo tree search

The MCTS is as a powerful global optimization method that has found wide-spread applications in computer games such as Alpha Go, Bridge, Poker and many other video games [12, 13]; and shows promises to tackle materials science problems [14–18]. It is a probabilistic and heuristic search algorithm that integrates a tree search algorithm with reinforcement learning. It builds a shallow tree of nodes (called leaf nodes) where each node represents a point in the search space and downstream pathways are generated by a rollout procedure. Here, each node represents a distinct arrangement of defects on the top layer of a MoS₂ film. The MCTS workflow for exploring the search space of 2D MoS₂ is shown schematically in figure 1. Our objective is to identify the optimal sulphur vacancy in MoS₂ i.e. the vacancy distribution that corresponds to the minimum energy configuration. We conduct MCTS calculations for a constant defect density. The MCTS begins with randomly distributing the vacancies in the top S layer in MoS₂. This candidate serves as the root node of the search tree. The search tree is built in an incremental and iterative way as shown in figure 3 until a pre-defined termination criterion is reached. Once the termination criteria is reached, the search is stopped and the best performing candidate is returned. Our termination criteria is set to be the minimum energy (remaining constant for >5000 MCTS evaluations).

In each MCTS iteration, four steps—selection, expansion, simulations and back-propagation are carried out. A child node is selected during the selection process based on the upper confidence bound (UCB) score [19]. The UCB of a node is defined as

$$ucb_i = \frac{z_i}{v_i} + C \sqrt{\frac{2 \ln v_p}{v_i}}. \quad (1)$$

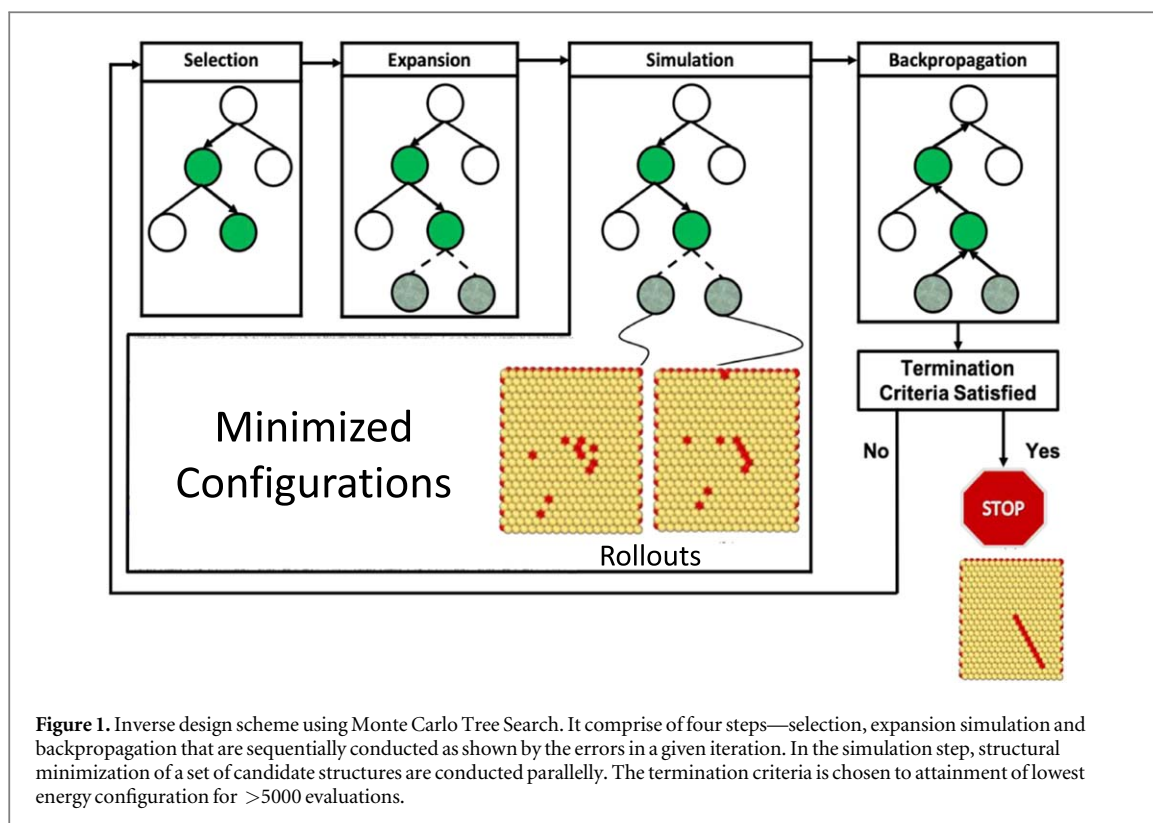
Here, z_i is the accumulated merit of the node (i.e., the sum of the immediate merits of all the downstream nodes), and v_i is the visit count of the node, v_p is the visit count of the parent node and C is a constant for balancing exploration and exploitation. The value of C is controlled adaptively at each node as

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

Here, J is the meta parameter which is set to be one and it increases whenever the algorithm reach a ‘dead end’ node to allow more exploration. At a dead-end node, the number of possible structures narrows to one. This happens when the numbers of $k-1$ candidate structures reach the limit. Here, the J is updated as

$$J \leftarrow J + \max \left\{ \frac{T-t}{T}, 0.1 \right\}, \quad (3)$$

where T is the total number of candidates to be evaluated, and t is the number of candidates for which the surface tension is already evaluated. Whenever a new node is added, the variables are initialized as $v_i = w_i = f_i = 0$ and $u_i = \infty$. Next, we perform the expansion of the tree by adding child nodes to the selected node. In the simulation

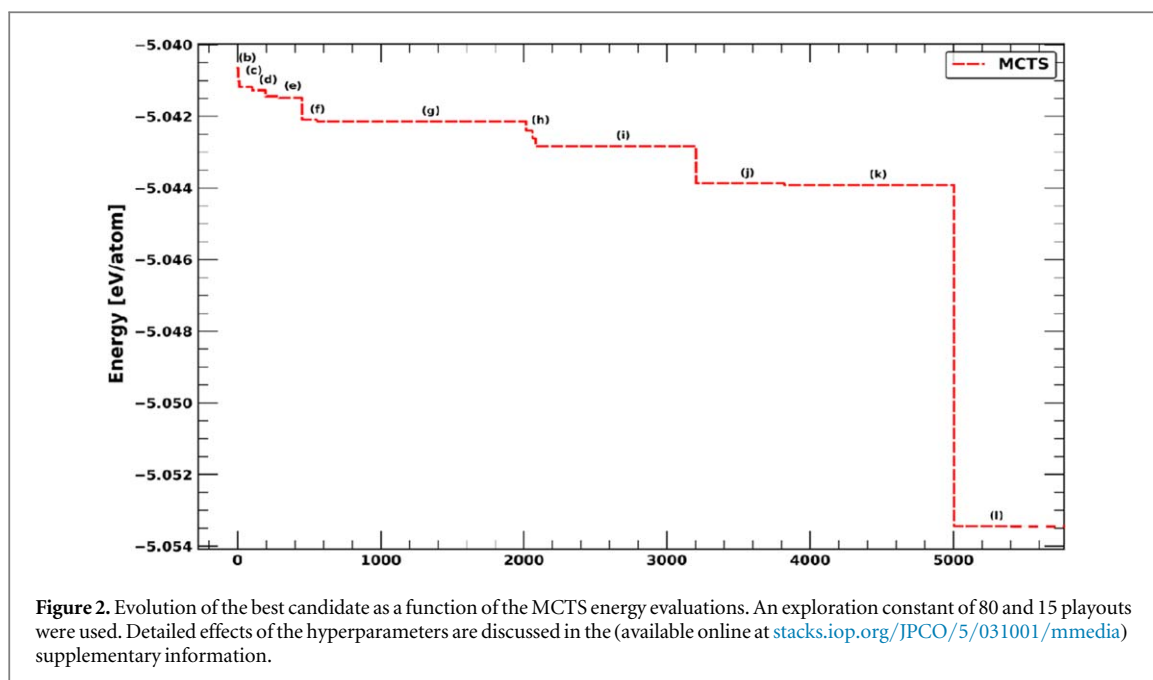


step, a playout is performed from each of the added children. We roll out 10 structures randomly during a playout from a child node. These roll outs are performed via two moves, a small vacancy shift and a large vacancy shift. In the small shift empty sites are swapped with neighboring filled sites to translate the empty site along the lattice. In the last shift an associate/dissociate paradigm is used where either an empty site is moved next to another empty site to created neighbored vacancies or two neighbored vacancies are separated by taking one of the two and randomly placing it throughout the lattice. The energy of the 10 structures are evaluated based on a reactive force field [20]. In this study, no neural networks were used for policy evaluation. Instead of a NN model, the objectives were evaluated based on an atomistic model. The objective in this case is $\min (f(r_{ij}))$, where $f(r_{ij})$ is the total energy of the minimized system configuration.

The optimization is carried out via the MCTS algorithms coupled to an atomistic simulator such as LAMMPS (acronym for *Large-scale Atomic/Molecular Massively Parallel Simulator*), which evaluates the energetics using the atomistic model [20]. In principle, a NN model could also be used for evaluation of the configurational energies. 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. Note that the backpropagation here refers to the update of information in the nodes on the path from the child to root node.

Evolutionary algorithm

Evolutionary algorithm seeks to mimic the process of ‘survival of the fittest’ to design or optimize a system’s variables to identify chosen target properties [21, 22]. This involves establishing a reversible mapping of every possible candidate (in this case, a distinct arrangement of atomic defects in a MoS₂ layer) to a genomic representation, selecting a random initial population of candidates, and determining the fitness -a measure of the degree to which a candidate’s property match to the target property. Here we employ such an evolutionary algorithm, viz., genetic algorithm (GA) to identify energetically most favorable arrangements of vacancies in one of the S planes of a MoS₂ layer. In GA, candidate materials structures are mapped to a genome upon which the evolutionary operations are employed. To describe the vacancy distribution in a given candidate structure, we define a 2D binary genome. This genome is a string containing the current state for each site in the 2D triangular lattice; each site can be in one of the two states, namely: (a) ‘0’ representing the absence of a S atom (vacancy), and (b) ‘1’ indicating that site is occupied by a S atom. For a prescribed vacancy density, the evolutionary search begins with a random population of 32 candidates, each with an arbitrarily chosen but distinct genome, (i.e., S-vacancy distribution). The fitness of each candidate is calculated as the potential energy of the MoS₂ monolayer containing a distribution of S-vacancies as defined by its 2D genome. In each generation, genetic operations, namely, selection, mutation, and crossover are performed on the current population to generate 32 new candidates. The mutation was performed by swapping a vacancy site with a neighboring filled site. The



crossover operations were performed using a typical cut and paste approach. Additional details of these genetic operations can be found in [8] and [11]. The candidates are then ranked by their fitness, and the best (fittest) 32 candidates are passed to the next generation. This procedure is iterated until the GA run converges, i.e., the potential energy of the fittest candidate does not change over a long period of time (~ 100 generations). We note that several case studies are conducted with varying number of population size, and 32 is found to be sufficiently large to ensure necessary structural diversity in the population (in the initial stages), as well as convergence to low energy configurations with reasonable computational costs. Our evolutionary searches using population sizes (>32) have resulted in identical lowest energy configurations, as that from runs with 32 candidates in the gene pool.

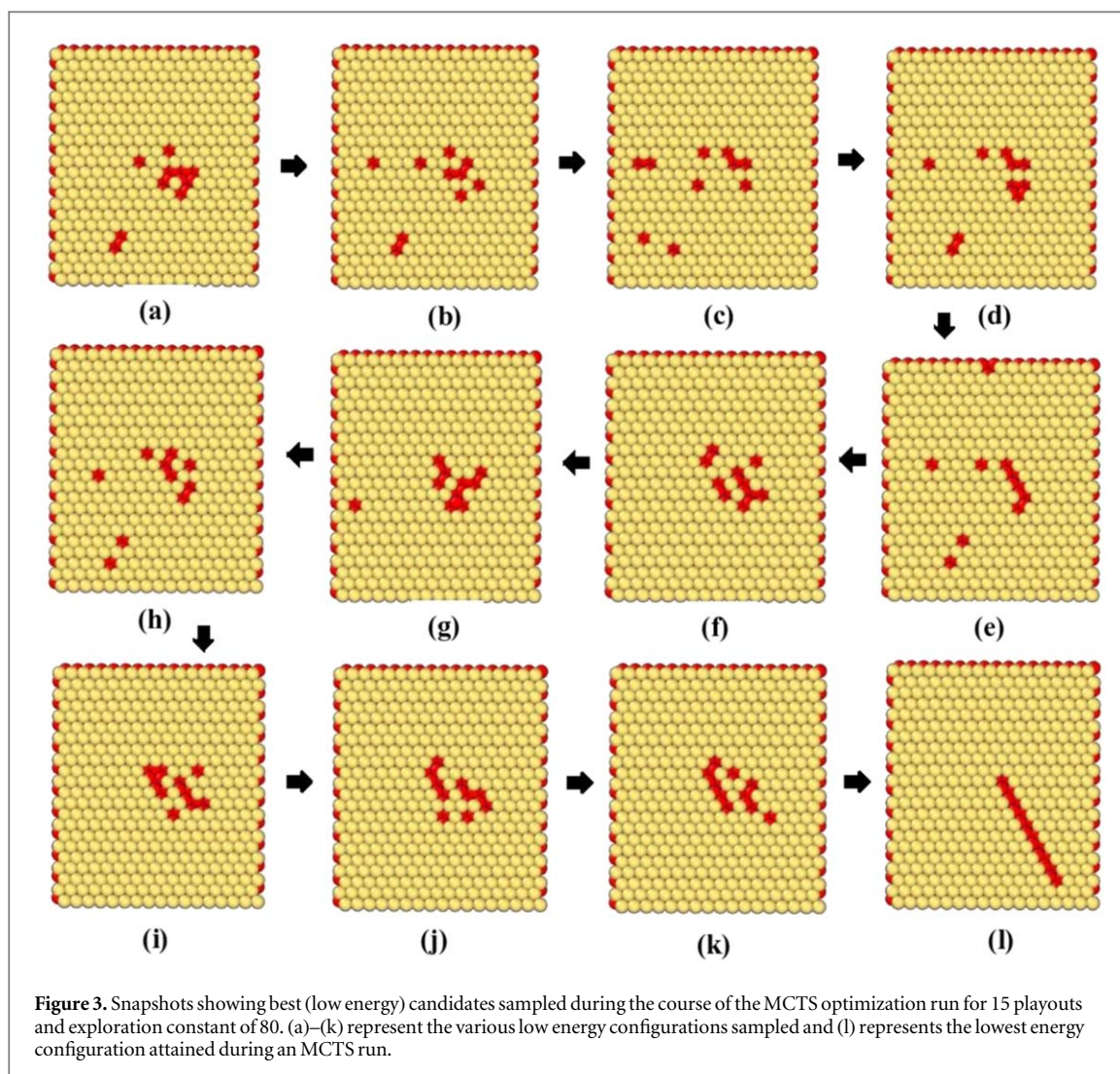
Reactive force-field for objective evaluation

The atomic interactions in the defective MoS₂ structures are modeled using the reactive force field (ReaxFF) [20]. All energy calculations are performed using LAMMPS [23] package. Periodic boundary conditions are employed in the plane of the MoS₂ sheet. LAMMPS is interfaced with an optimizer such as MCTS or GA—the energy evaluations using ReaxFF model of the defective MoS₂ sheets are used to determine the objective function. We note that in a single layer MoS₂, a plane of Mo atoms is sandwiched between two planes of S atoms; in each S plane, the atoms are organized in a 2D triangular lattice. As the interlayer vacancy migration is associated with a very high energy barrier (>5 eV), we restrict our search space to the top layer of S atoms. All the calculations are performed for a defect density of 0.03, which is experimentally realizable. We note that the defect density is defined as the ratio of vacant S sites to the total S sites in the top layer of a MoS₂ film.

Results and discussion

A schematic of the MCTS workflow used for defect optimization is shown in figure 1. Our objective is to minimize the configurational energy of the system for any given distribution of the defects i.e. S vacancies. For a given N i.e. number of S vacancies, we initiate the search by randomly distributing the N vacancies amongst the lattice position for the top S layer in MoS₂. This candidate serves as the root node of the search tree. A search tree is built in an incremental and iterative way as shown in figure 1. The objective evaluation is carried out by performing a structure minimization using LAMMPS.

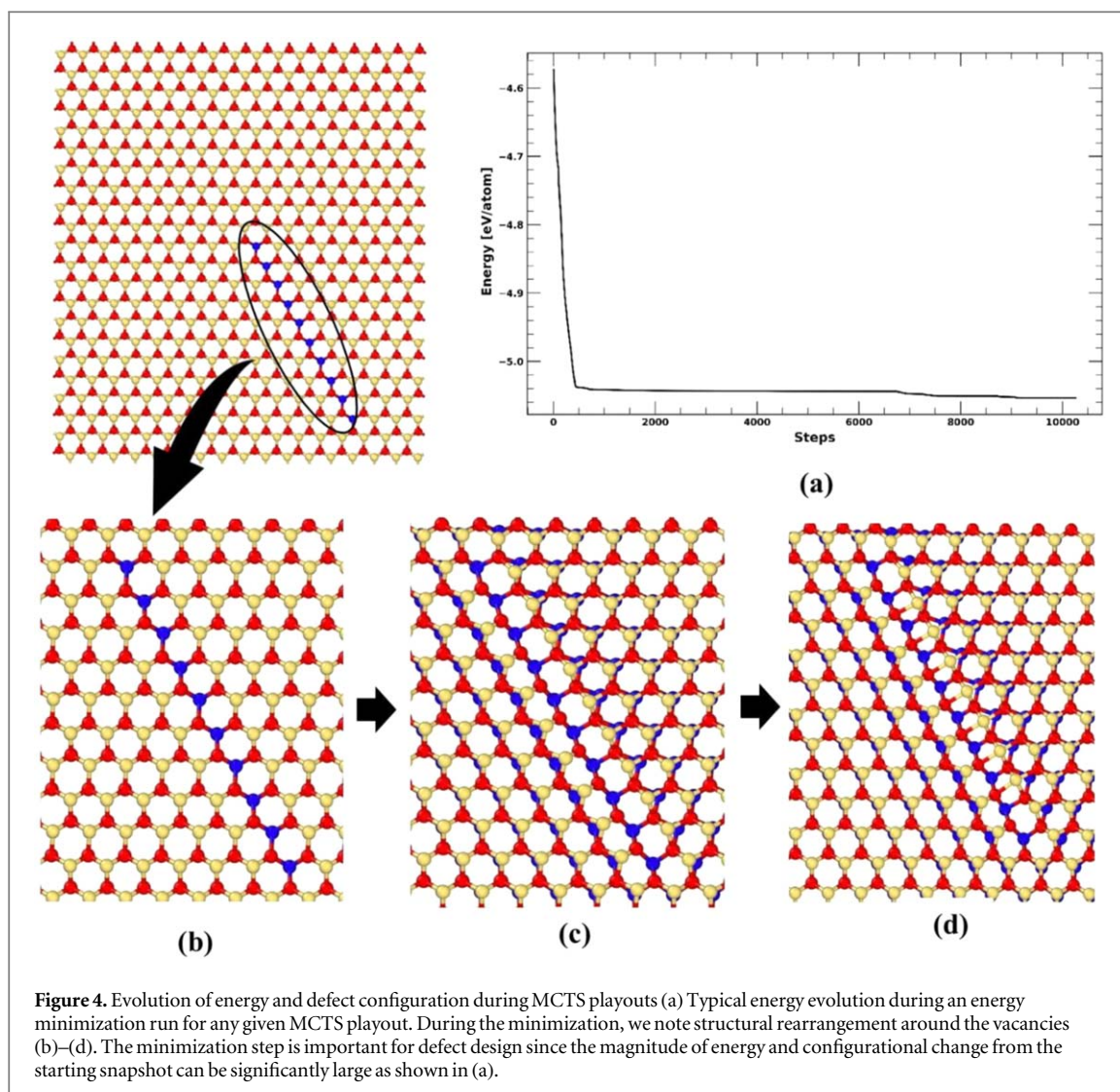
To assess the performance of the MCTS search algorithm, we track the evolution of the best candidate i.e. one with the lowest energy is the search as a function of the number of search evaluation. We note from figure 2 that there is a slow initial drop in the configurational energy for evaluations <1000 , when the search plateaus out and remains constant at ~ 5.042 eV/atom until ~ 2000 evaluations. This is followed by a slow drop until ~ 5000 evaluation, when a sudden and significant drop to the optimal solution is observed. The optimal solution corresponds to the S vacancies arranged in the form of an extended line defect. The objective does not change for the subsequent 5000 evaluations and hence the search is terminated.



We next attempt to understand the MCTS configurational search process and the evolution of the objective shown in figure 2. The initial slower decay in the configurational energy for evaluations <3000 is attributed to the search being primarily in the exploratory stage. During the early stages, the tree expansion is preferred and the search algorithm tends to explore the various defect configurations, which correspond to widely different configurational space. The initial search has high degree of stochasticity and is random—as a result the energetic drops are rather small. Beyond 3000 evaluations, the search is focussed on exploitation of the configurational regions identified by the tree—during this phase, drop is significant and convergence to solution is much more rapid. We note that the exploratory and exploitation stages may differ amongst the different MCTS runs depending on the choice of the hyperparameters. The influence of a few representative MCTS hyperparameters on the search is discussed later in the manuscript.

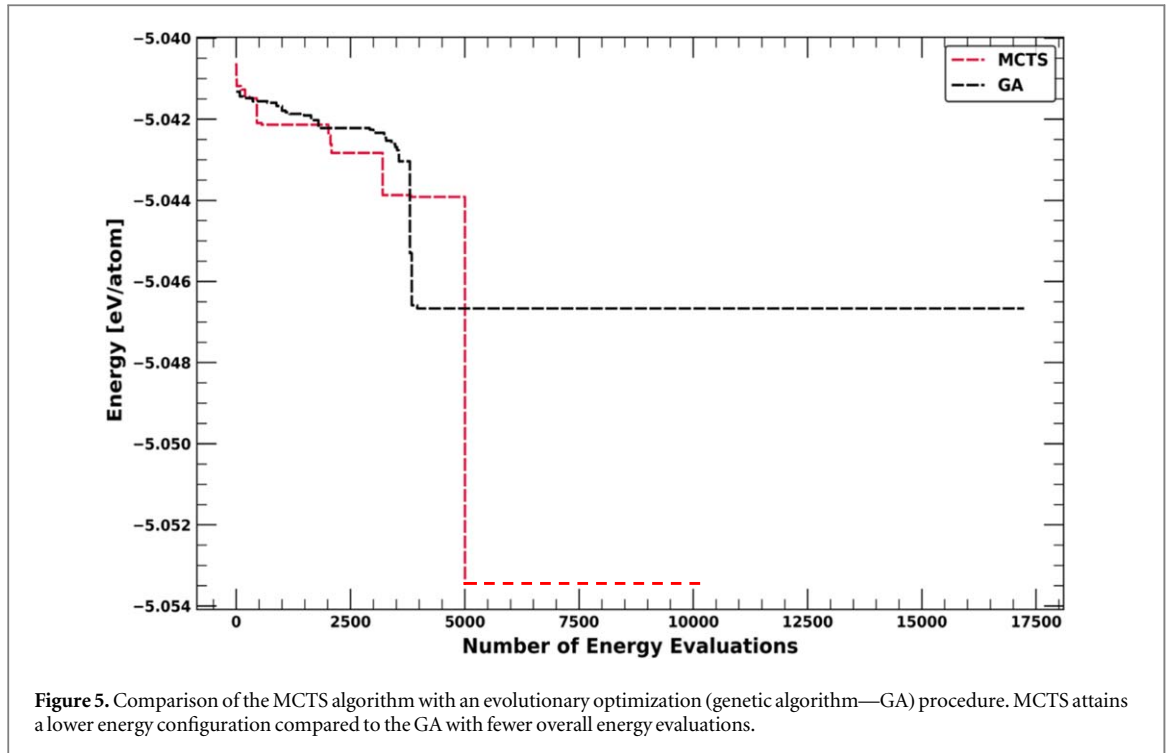
The snapshots shown in figure 3 tracks the best candidate identified during the search process. Snapshots in figures 3(a)–(e) show the configurations obtained during the exploration stage. We can note that configurations sampled primarily have mono-vacancies with a few S di- and S tri-vacancy. During the exploitation stage as shown by snapshots in figures 3(f)–(l), we observe that the configurations sampled have clusters of S vacancies—MCTS explores more in regions where vacancy clustering is preferred signifying that the energetics favor the various mono-vacancies to cluster together. Finally, we see that the most preferred orientation for the S vacancies is to align themselves as a single line defect. This is consistent with previous experimental observation by M. Terrones and co-workers. Their HRTEM studies showed that structural defects in 2D MoS₂ evolve from randomly distributed sulphur monovacancies to distributed line defects to extended coupled line defects induced structural disorders. Such HRTEM observations validate the MCTS prediction and indicate a large variation in the size of low energy extended defects formed in 2D MoS₂.

An important aspect of the MCTS workflow used for defect design is the explicit use of minimization step. We note that that during the playouts, we observe a significant energy/structural deviation of the final minimized configuration compared to the initial (Fig. 4). The number of minimization steps required to



converge varies depending on the nature of the defect—we observe ~ 400 to ~ 15000 minimization steps are required for convergence (Fig. 4). The energy values of the final minimized configurations are used for evaluating the objective function. Thus, even though MCTS moves to manipulate the S vacancies are performed on the MoS_2 lattice, significant structural variations might result upon relaxation even for subtle defect variations. To ensure appropriate mapping between the MCTS configurational moves and the corresponding energies, the objective function should be based on minimized energies to ensure convergence.

We next compare the performance of MCTS with that of an evolutionary approach. We perform GA runs with an initial population starting with randomly distributed S-vacancies, at the desired vacancy density i.e. $N = 10$. The configurational energy of the most stable structure obtained at each generation is tracked and the evolution of the minimum energy configuration is shown as a function of the number of GA evaluations (population * generation) in figure 5. We observe that the GA-identified lowest energy configuration (after 600 generations or 18000 evaluations) for $N = 10$, consist of a combination of lines (of varying sizes) and isolated vacancies, which results in a somewhat higher energy ~ -5.047 eV/atom. Ordering all the vacancies in a single line corresponding to typical line defects seen in experiments, for $N = 10$ yields an energy value of -5.054 eV/atom. Thus, the GA run for $N = 10$ does not reach the global energy minimum even after 18000 evaluations. Defect optimization is often non-trivial and optimization algorithms have difficulties in surmounting suboptimal solutions and tend to slow down near the optimal points as observed in the case of the GA based search. The major advantage the MCTS algorithm has is that it is able to generate a large number of dissimilar structures and balance the relative exploitation and exploration of each of the unique structures. Information from every structure that has been sampled is retained during the optimization and used to determine the next trial. It is able to obtain an approximate idea how much of the phase space is still left to explore and determine if it is worth exploring or if it should instead choose to exploit what it already knows. GA is more localized in scope in that it tends to generate structures that look similar to the ones already in the pool which will generally create

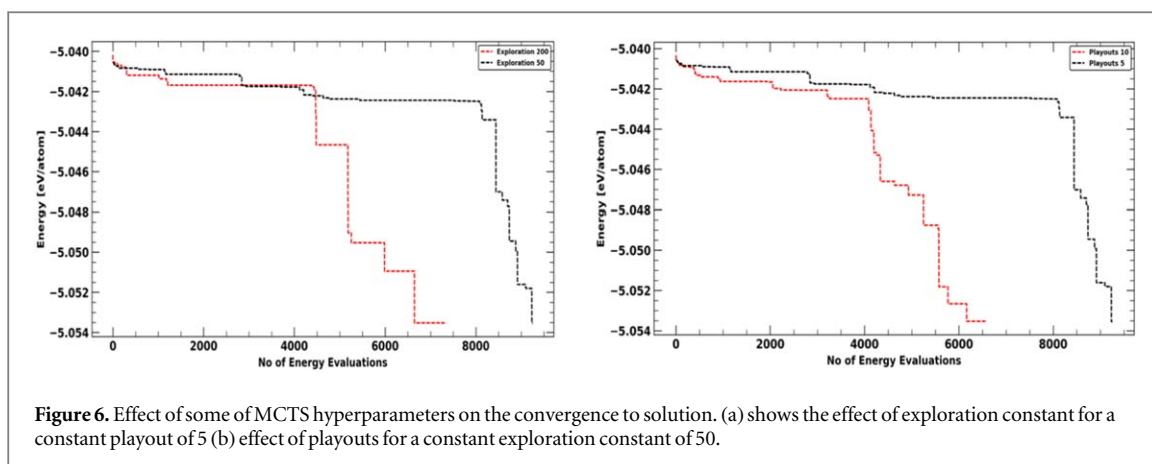


solutions that are perhaps one or two minima well over in the phase space. A major advantage of the MCTS, as evidenced in the case of defect optimization, is that if the search gets trapped in a metastable or suboptimal point, it is expected to quickly find another pathway by growing other branches of the tree utilizing the trade-off mechanism between exploration and exploitation. MCTS is therefore able to reach the optimal solution i.e. the line defect with energy value of ~ -5.054 eV/atom in < 5000 evaluations.

We evaluate the effect of MCTS hyperparameters on the search efficiency. First, we evaluated the effect of MCTS exploration constant. The search is initiated with randomly generated vacancies on the top S monolayer of the MoS₂ sheet. Mainly 4 kind of moves are used for generating a new structure (i.e. a child node) from previously obtained structures. These are swap, associate, dissociate and shift. The MCTS searches through the action space and selects the best possible child based on the upper confidence bounds (UCB formula shown in equation 1). UCB works on balancing between a more explorative randomly sampled search (depends on the exploration) with the quality of the node objective (more exploitation). In the MoS₂ system, the energy of the qualitatively good and the bad structures are very close in terms of magnitude and only differ by few meV/atom, even though the total cohesive energy of the system is on the order of 5 eV/atom. In addition to this subtle energetic difference between the various sampled configurations, it is possible to generate structures that are identical in shape, but may be slightly shifted or rotated from another or can have multiple energetically similar structure with different arrangement of vacancies. For the search to work well, it should initially explore the search space enough and have enough samples to make an optimal selection that can further be exploited. The balance between exploration and exploitation is critical to ensure faster converge to the optimal structure—an increase in tree width improves exploration whereas the tree depth increases the exploitation tendencies of the MCTS.

$$UCB(node) = \text{Min}(S_1, S_2, S_3 \dots S_{n_i}) + C_{\text{explore}} \cdot f_i(\vec{r}_i) \cdot \sqrt{\frac{\log N_i}{n_i}}$$

If the exploration constant is too small, the tree will greedily pick the current best rewards until it hits a local minima. On the other hand, selecting a constant too large will make it that makes the search to be effectively random and reduces the convergence probability significantly to all but nothing. So, a proper selection of exploration constants can make the search converge efficiently in a relatively few number of expensive objective function evaluations. For the sake of illustration, we start with two different explorations constant, each with a fixed number of playouts ~ 5 . For an exploration constant of 200, it took 300 evaluations before the MCTS found energies that are comparatively higher than the one with an exploration constant of 50. The main reason behind this is the search is still exploring even after finding potential structures that can be exploited to find a low energy structure. In contrast, the exploration constant of 50 has swiftly moved towards a local minima. After 300 evaluations, the search with an exploration constant of 200 slowly moves towards the exploitation stage (2nd



term in the UCB equation starts to decrease). This is manifested in figure 6(a)—the configurational energy tends to fall quickly as the exploited structures are more likely to produce a good vacancy alignment upon perturbation.

Next, we study the effect of payouts on the MCTS search. Here, we set the exploration constant to a fixed value of 50 and tried different number of payouts to observe its effect on the convergence rate. Since the payouts are a random measure of the local configuration space around a given node, a larger number of random samples in theory improve the MCTS algorithm's understanding of what potential solutions may exist nearby a search node. Insufficient payouts can lead to an inefficient branch selection because very little information is being discovered per test. But there is a declining rate of return by doing more payouts than is absolutely necessary as this ultimately increases the search time significantly. In such a situation, much of the search is spent on computing redundant or degenerate information. For defect design, a balanced payout is extremely crucial since each structure takes an average of ~ 5000 energy steps for minimization, which slows down the search significantly. We tried two representative payouts for the search as shown in figure 6(b). From the initial progression of the search, it appears that 10 payouts is more efficient than the 5 one and reduces the time to solution. In this MoS₂ system most of the time the energy barrier between a parent and potentially very good child that can be obtained from the parent upon perturbation is considerably high in the context of energy ranges considered. Sometimes it becomes very crucial for MCTS to perform a minimum number of payouts at a node to get a qualitative understanding of the node. But in the context of the given fact that more payout leads to computational inefficiency the maximum payout was kept to 10 which seems to perform a better job in comparison to the search with 5 payouts. For the payout 10 case figure 5(b) the search performs adequate number of payouts at selected node and gets a better qualitative idea of the node based on UCB equation. The chances of a selection of a potentially good parent becomes higher. Thus, the search tends to quickly moves towards a zone where it samples higher number of good offspring and moves quickly towards convergence as compared to the search with payout 5.

Finally, we note that the problem outline above is particularly challenging owing to the subtle differences in the defect energetics. The energetic differences are small because it is easy to generate one of the local minima of the system through naïve random sampling. When the defects are not lined up, the next lowest energy state we have found is to have the defects near each other, but with one filled chalcogen site in between. These are also seen in the snapshots in figure 3. These spotted defects are the easiest minima to access as they can be created even with only a few of the defects (S vacancy) being close to each other.

Benefits of MCTS optimization and related work

Many of the optimization approaches, we looked at such as GA, Bayesian etc., would quickly form these spotted defects and fail to try the line defect of S vacancies. Higher energy configurations do exist even in the case of randomly distributed vacancies. For example, one can simply spread out all the defects away from each other, but there's a high statistical chance that one will get these spotted defects even through naïve or random sampling. As such, our initial design for both the GA and MCTS quickly created one of these configurations. The hard task for sampling, however, is escaping these sub-optimal minima. A major problem with any random based approaches is that they are significantly more likely to generate these local minima than the line defect.

A major advantage the MCTS algorithm has is that it is able to generate a large number of dissimilar structures and balance the relative exploitation and exploration of each of the unique structures [16, 24, 25]. Information from every structure that has been sampled is retained during the optimization and used to determine the next trial. It is able to obtain an approximate idea how much of the phase space is still left to

explore and determine if it is worth exploring or if it should instead choose to exploit what it already knows. In this respect, our strategy outlined seems to clearly perform better compared to local and other global evolutionary search

Conclusion

Defect design and optimization in 2D materials is important for a wide range of applications from catalysis to nanoscale electronics. For example, the nature of atomic scale rearrangement of chalcogen vacancies in 2D TMDCs influences the $2H \leftrightarrow 1T$ phase transitions, which in turn has a profound effect on its electronic properties. We perform reinforcement learning in a discrete action space by using Monte Carlo Tree Search interfaced with a reactive molecular dynamics simulator to find optimal arrangement of S vacancies in MoS_2 . Based on our MCTS search of the exorbitant defect search space, we find that the S vacancies tend to form an extended line defect of vacancies—these findings are corroborated by previous experimental studies. By making suitable comparison with an evolutionary approach, we demonstrate the power of RL search—MCTS attains a significantly lower objective in far fewer evaluations compared to GA. We also perform a detailed evaluation on the effect of different MCTS hyperparameters on the defect inverse design. We believe our RL approach is quite general and can be broadly applied to several materials problems that involve search in discrete action space.

Acknowledgments

The use of the Center for Nanoscale Materials, an Office of Science user facility, was supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357. This research used resources of the National Energy Research Scientific Computing Center, which was supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program of the Argonne Leadership Computing Facility at the Argonne National Laboratory, which was supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. The authors would like to acknowledge the support from the Argonne LDRD-2017-012-N0 project, UIC faculty start-up fund and IIT Madras faculty stat-up fund. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences Data, Artificial Intelligence and Machine Learning at DOE Scientific User Facilities program under Award Number 34532.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

Subramanian K R S Sankaranarayanan  <https://orcid.org/0000-0002-9708-396X>

References

- [1] Schulman D S, Arnold A J and Das S 2018 Contact engineering for 2D materials and devices *Chem. Soc. Rev.* **47** 3037–58
- [2] McDonnell S, Addou R, Buie C, Wallace R M and Hinkle C L 2014 Defect-dominated doping and contact resistance in MoS_2 *ACS Nano* **8** 2880–8
- [3] Yu Z *et al* 2014 Towards intrinsic charge transport in monolayer molybdenum disulfide by defect and interface engineering *Nat. Commun.* **5** 5290
- [4] Sangwan V K, Lee H-S, Bergeron H, Balla I, Beck M E, Chen K-S and Hersam M C 2018 Multi-terminal memtransistors from polycrystalline monolayer molybdenum disulfide *Nature* **554** 500–4
- [5] Zhou W, Zou X, Najmaei S, Liu Z, Shi Y, Kong J, Lou J, Ajayan P M, Yakobson B I and Idrobo J-C 2013 Intrinsic structural defects in monolayer molybdenum disulfide *Nano Lett.* **13** 2615–22
- [6] Addou R, Colombo L and Wallace R M 2015 Surface defects on natural MoS_2 *ACS Appl. Mater. Interfaces* **7** 11921–9
- [7] Bampoulis P, van Bremen R, Yao Q, Poelsema B, Zandvliet H J W and Sotthewes K 2017 Defect dominated charge transport and fermi level pinning in MoS_2 /metal contacts. *ACS Appl. Mater. Interfaces* **9** 19278–86
- [8] Patra T K, Zhang F, Schulman D S, Chan H, Cherukara M J, Terrones M, Das S, Narayanan B and Sankaranarayanan S K R S 2018 Defect dynamics in 2-D MoS_2 probed by using machine learning, atomistic simulations, and high-resolution microscopy *ACS Nano* **12** 8006–16
- [9] Lin Y-C, Dumcenco D O, Huang Y-S and Suenaga K 2014 Atomic mechanism of the semiconducting-to-metallic phase transition in single-layered MoS_2 . *Nat. Nanotechnol.* **9** 391–6 [nnano.2014.64](https://doi.org/10.1038/nnano.2014.64)

- [10] Patra T K, Loeffler T D and Sankaranarayanan S K R S 2020 Accelerating copolymer inverse design using monte carlo tree search. *Nanoscale* **12** 23653–62
- [11] Meenakshisundaram V, Hung J-H, Patra T K and Simmons D S 2017 Designing sequence-specific copolymer compatibilizers using a molecular-dynamics-simulation-based genetic algorithm *Macromolecules* **50** 1155–66
- [12] Silver D *et al* 2016 Mastering the game of go with deep neural networks and tree search *Nature* **529** 484–9
- [13] Browne C B, Powley E, Whitehouse D, Lucas S M, Cowling P I, Rohlfshagen P, Tavener S, Perez D, Samothrakis S and Colton S 2012 A survey of monte carlo tree search methods *IEEE Trans. Comput. Intell. AI Games* **4** 1–43
- [14] Dieb T M, Ju S, Yoshizoe K, Hou Z, Shiomi J and Tsuda K 2017 MDTS: automatic complex materials design using monte carlo tree search. *Sci. Technol. Adv. Mater.* **18** 498–503
- [15] Dieb T M, Ju S, Shiomi J and Tsuda K 2019 Monte Carlo tree search for materials design and discovery *MRS Commun.* **9** 532–6
- [16] Segler M H S, Preuss M and Waller M P 2018 Planning chemical syntheses with deep neural networks and symbolic AI *Nature* **555** 604–10
- [17] Kiyohara S and Mizoguchi T 2018 Searching the stable segregation configuration at the grain boundary by a monte carlo tree search. *J. Chem. Phys. DETC2018* **1** 241741
- [18] Shin K, Tran D P, Takemura K, Kitao A, Terayama K and Tsuda K 2019 Enhancing biomolecular sampling with reinforcement learning: a tree search molecular dynamics simulation method. *ACS Omega* **4** 13853–62
- [19] Kocsis L and Szepesvári C 2006 Bandit based Monte-Carlo planning. *Machine Learning: ECML 2006* ed J Fürnkranz *et al* (Heidelberg: Lecture Notes in Computer Science; Springer: Berlin) pp. 282–93
- [20] Ostadhosseini A, Rahnamoun A, Wang Y, Zhao P, Zhang S, Crespi V H and van Duin A C T 2017 ReaxFF reactive force-field study of molybdenum disulfide (MoS₂). *J. Phys. Chem. Lett.* **8** 631–40
- [21] Goldberg D E 1989 *Genetic Algorithms in Search, Optimization, and Machine Learning* (Massachusetts: Addison-Wesley Publishing Company)
- [22] Mitchell M 1998 *An Introduction to Genetic Algorithms* (Cambridge, MA: MIT Press)
- [23] Plimpton S 1995 Fast parallel algorithms for short-range molecular dynamics *J. Comput. Phys.* **117** 1–19
- [24] White D, Yoo S and Singer J The programming game: evaluating MCTS as an alternative to GP for symbolic regression. GECCO companion 15: *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation* 1521–2
- [25] Patra T K, Loeffler T D and Sankaranarayanan S K R S 2020 Accelerating copolymer inverse design using Monte Carlo tree search *Nanoscale* **12** 23653–62
- [26] Patra T K, Meenakshisundaram V, Hung J-H and Simmons D S 2017 Neural-network-biased genetic algorithms for materials design: evolutionary algorithms that learn. *ACS Comb. Sci.* **19** 96–107