# Highly efficient tree search algorithm for irreducible site-occupancy configurations

Ji-Chun Lian,<sup>1</sup> Yuan Si,<sup>1</sup> Tao Huang,<sup>1</sup> Wei-Qing Huang<sup>0</sup>,<sup>1,\*</sup> Wangyu Hu,<sup>2</sup> and Gui-Fang Huang<sup>1,†</sup> <sup>1</sup>Department of Applied Physics, School of Physics and Electronics, Hunan University, Changsha 410082, China <sup>2</sup>School of Materials Science and Engineering, Hunan University, Changsha 410082, China

(Received 18 July 2021; accepted 15 December 2021; published 3 January 2022)

We present a universal and extremely efficient tree search algorithm for irreducible site-occupancy configurations (implemented in DISORDER code) that partially avoids the barrier from the combinatorial explosion and allows us to model the compositionally complex materials. The tree search algorithm is developed based on our original algorithm and is leveraging the idea of stopping descending further down the branches of the tree that do not meet the requirements. Meanwhile, the wrongly counted degeneracies of the irreducible site-occupancy configurations, caused by the skipping of some branches of the tree, can be corrected by a degeneracy correction procedure. Using binary face-centered cubic alloys, ternary body-centered cubic alloys, and quaternary simple cubic alloys as examples, we demonstrate that, compared with our original algorithm, the overall efficiency of the tree search algorithm is improved by more than 2 times for binary site occupancy, 70 times for ternary site occupancy, and 50 times for quaternary site occupancy, which is far beyond other similar algorithms. The tree search algorithm developed here can be broadly useful for the modeling of high-entropy alloys and provides support for other methods, such as special quasirandom structures and small set of ordered structures, that require enumerating a set of site-occupancy configurations.

DOI: 10.1103/PhysRevB.105.014201

### I. INTRODUCTION

As one of the most important sources of solid-state structures, site-occupancy disorder systems have been extensively investigated in many domains of condensed matter physics and materials science on account of their structural diversity and promising applications [1-10]. For theoretical calculations studies on site-occupancy disorder, the concrete atomic configurations are always indispensable; they are typically derived from an ordered supercell, and the number of derived configurations increases dramatically with the increase of supercell size, i.e., the so-called combinatorial explosion. Finding a set of irreducible (symmetrically inequivalent) siteoccupancy configurations from all combinatorially distinct configurations by using crystal symmetry is a useful means to reduce the number of generated configurations and thus the computational cost. Up to date, various algorithms with increasing efficiency have been proposed to generate irreducible site-occupancy configurations, such as the algorithm implemented in SOD [11], CRYSTAL [12], ENUMLIB [13-16], SUPERCELL [17], and DISORDER [18]. Nevertheless, the pursuit of more efficient algorithms for generating irreducible site-occupancy configurations to accelerate the process of researching disordered systems still has methodological and practical significance.

High-entropy alloys (HEAs) [19–28], a kind of compositionally complex material, i.e., multiple different types of atoms are randomly arranged on a crystalline lattice, possess diversified applications and have received a great deal of research interest, providing one of the most striking paragons for the practical application of an irreducible site-occupancy configuration generation algorithm. To model the disordered alloys (e.g., HEAs), numerous techniques have been developed [29-37], and the most widely used are coherent potential approximation [29], cluster expansion (CE) [31], special quasirandom structure (SQS) [33], and small set of ordered structures (SSOS) [35]. Among them, the techniques of CE, SQS, and SSOS require enumerating a set of siteoccupancy configurations, which is a subset of the irreducible site-occupancy configurations, while the exhaustive enumeration of all possible irreducible site-occupancy configurations is the only known way that is guaranteed to find optimal SQS and SSOS [38]. However, the search of irreducible siteoccupancy configurations for HEAs suffers from more severe difficulty brought by combinatorial explosion than that for common binary or ternary alloys. Although the tiptop irreducible configuration generation algorithm (implemented in ENUMILB code) for multinary site occupancy recently developed by Morgan, Hart, and Forcade [16] partly avoids the combinatorial explosion, its efficiency is still insufficient for the larger supercell and/or the HEAs with higher entropy. Therefore, developing more efficient search algorithms for irreducible site-occupancy configurations will provide strong support for the wider applications of SQS and SSOS techniques, etc., and promotes the research of HEAs.

In this paper, by abstracting the search of irreducible site-occupancy configurations into a treelike structure, we proposed a tree search algorithm based on our original algorithm to have higher efficiency. We show that, compared with our original algorithm, the efficiency of the tree search

<sup>\*</sup>Corresponding author: wqhuang@hnu.edu.cn

<sup>&</sup>lt;sup>†</sup>Corresponding author: gfhuang@hnu.edu.cn

algorithm is greatly improved, especially for the multinary site-occupancy systems. The reason for this performance improvement is that, in the process of tree search, the branches that do not meet some requirements are blocked, i.e., avoid descending further down the tree, whereas it is equivalent to searching all the branches of the tree in the original algorithm. Additionally, we also proposed a degeneracy correction procedure to correct the wrongly counted degeneracies caused by the skipping of some branches of the tree. To demonstrate its practical application, we apply the tree search algorithm to generate the optimal SQSs for ternary equiatomic facecentered cubic (fcc) and body-centered cubic (bcc) alloys as examples, which is an extremely challenging task for the traditional exhaustive method used for SQS generation. The tree search algorithm for irreducible site-occupancy configurations developed here is universal and efficient, which can be expected to be useful for the modeling of various compositionally complex materials.

### **II. ORIGINAL ALGORITHM RECAP**

To make the paper more coherent, in this section, a brief review of our original algorithm is presented to the readers for a background, while the details of it are given in Ref. [18]. The algorithm for generating irreducible site-occupancy configurations is essentially to deal with three problems: (1) how to represent and enumerate atomic configurations, (2) how to identify symmetrically equivalent atomic configurations, and (3) how to eliminate duplicate atomic configurations. In the following, the treatment of these problems adopted in our original algorithm will be introduced concisely and to the point.

In our algorithm, the representation and enumeration of atomic configurations are based on combinatorics. Consider a supercell with *K* atomic positions that can be occupied by *N* different types of atoms (the vacancies are deemed as a special type of atom), and the number of each type of atom is  $k_i (i \in [1, N])$ , satisfying  $K = \sum_{i=1}^{N} k_i$ . For such a system, the total number (*Nc*) of atomic configurations can be expressed as a multinomial coefficient:

$$Nc(k_1, k_2, \dots, k_N) = \frac{(k_1 + k_2 + \dots + k_N)!}{k_1! k_2! \cdots k_N!} = \frac{\left(\sum_{i=1}^N k_i\right)!}{\prod_{i=1}^N k_i!}.$$

For binary (N = 2) site occupancy, the multinomial coefficient degenerates into the well-known binomial coefficient:

$$Nc(k_1, k_2) = \frac{(k_1 + k_2)!}{k_1! k_2!} = \frac{K!}{k_1! k_2!} = C_K^{k_1} = C_K^{k_2},$$

which inspires us to use binary combinations to represent the atomic configurations for the binary system, where the binary combinations corresponding to the binomial coefficient  $C_n^m$  are the possible ways to choose a subset of size *m* elements, disregarding their order, in a consecutive integer set from 1 to *n*. Meanwhile, for the multinary ( $N \ge 3$ ) site-occupancy case, we can decompose it into several binary site occupancies, which stems from the fact that the multinomial coefficient can be written as the product of several binomial coefficients. For the sake of clarity, we present an example for K = 6,  $k_1 = 3$  (blue), and  $k_2 = 3$  (red), i.e., the binary site occupancy,

as shown in Fig. 1(a), and an example for K = 6,  $k_1 = 2$  (blue),  $k_2 = 2$  (red), and  $k_3 = 2$  (green), i.e., the ternary site occupancy, as shown in Fig. 1(b).

The identification of symmetrically equivalent atomic configurations is achieved by an equivalent atomic matrix, which is constructed from the space group operations of the supercell. Based on the algorithm implemented in SPGLIB [39], the space group operations of an arbitrary supercell can be searched out directly from its structural information. Moreover, the atomic configuration after a space group operation may be beyond the original configurations space; however, it can be returned to the original configuration space by a permutation operation, which is realized by sorting the configuration in numerical order.

The elimination of duplicate configurations by using symmetries is the most time-consuming procedure, so that it becomes the key to the efficiency of our original algorithm as well as the tree search algorithm. Figure 2(a) shows the flowchart of eliminating duplicate configurations for our original algorithm, in which the fast conversion of configurations to integers plays a pivotal role in the high efficiency of our original algorithm.

### **III. TREE SEARCH ALGORITHM DETAILS**

The tree search algorithm is developed based on our original algorithm with some modifications on the procedure of eliminating duplicate configurations. The key concept of the tree search algorithm is to abstract the search of irreducible site-occupancy configurations into a treelike structure and block the branches of the tree that do not meet the requirements. Other basic ideas, such as the enumeration of atomic configurations, the identification of symmetrically equivalent atomic configurations, and the conversion of configurations to integers are consistent with those of the original algorithm. In the following, the implementation details of the tree search algorithm for binary and multinary site occupancy will be elaborated, respectively.

#### A. Binary site occupancy

For binary site occupancy, the tree search algorithm for searching irreducible configurations is based on a treelike structure, as shown in Fig. 3(a): an example for K = 6,  $k_1 =$ 3 (blue), and  $k_2 = 3$  (red). For such an example, the tree search for irreducible configurations is performed as follows: (1) searching irreducible configurations for the binary site occupancy of  $(k_1 = 1, k_2 = 5)$  from the configurations space of {1, 2, 3, 4}; (2) searching irreducible configurations for the binary site occupancy of  $(k_1 = 2, k_2 = 4)$  from the configurations space of {1 2, 1 3, 1 4, 1 5, 3 4, 3 5}; (3) searching irreducible configurations for the binary site occupancy of  $(k_1 = 3, k_2 = 3)$  from the configurations space of {1 2 3, 1 2 4, 1 2 5, 1 2 6, **1 3 4**, **1 3 5**, **1 3 6**, **3 4 5**, 3 4 6, where the bold numbers indicate the irreducible configurations searched out. We can see that the nodes not belonging to the irreducible configurations have no need to descend further down the tree, resulting in a sharp reduction of the configuration space used to search for irreducible configurations.



FIG. 1. The enumeration of (a) binary site-occupancy configurations for K = 6,  $k_1 = 3$  (blue), and  $k_2 = 3$  (red) and (b) ternary site-occupancy configurations for K = 6,  $k_1 = 2$  (blue),  $k_2 = 2$  (red), and  $k_3 = 2$  (green).

For such a treelike structure, however, the branches of the tree are not independent, i.e., the configurations will jump from branch to branch after performing symmetry operations (i.e., space group operations combining permutation operations), which causes difficulty in programming and in counting degeneracies. Although it is possible for us to implement it in our future work, at present, we have to take the second best, i.e., adopting the treelike structure, as shown in Fig. 3(b), whose basic idea is consistent with that presented in Fig. 3(a). The only difference between them is that we combine steps (2) and (3) above into one step, i.e., searching irreducible configurations for the binary site occupancy of  $(k_1 = 3, k_2 = 3)$  from the configurations space of {1 2 3, 1 24, 125, 126, **134**, **135**, **136**, 145, 146, 156, **345**, 3 4 6, 3 5 6}, while step (1) is the same. Obviously, for such a treelike structure, the configuration space used to search for irreducible configurations is also reduced a lot, and most importantly, we have a strategy to decorrelate the branches of the tree. In the following, the specific implementation of the treelike structure presented in Fig. 3(b) will be discussed in detail.

Firstly, we consider a special case, assuming the *K* atomic positions are indistinguishable, i.e., they are symmetrically equivalent. It is conceivable that, for such a case, the configuration that does not start with 1 can be converted to the configuration that starts with 1 by a symmetry operation, and we do not need to know exactly which symmetry operation. In the tree search algorithm, therefore, the configurations that do not start with 1 are discarded directly, forming a reduced configuration space, i.e., {1 2 3, 1 2 4, 1 2 5, 1 2 6, 1 3 4, 1 3 5, 1 3 6, 1 4 5, 1 4 6, 1 5 6} for the example of K = 6 (the six atomic positions are indistinguishable),  $k_1 = 3$  (blue), and  $k_2 = 3$  (red).

However, the counted degeneracies of the irreducible configurations are incorrect for the tree search algorithm, owing to some configurations being skipped, and a degeneracy correction procedure is therefore needed. Fortunately, for the situation we are discussing, the counted degeneracies of all irreducible configurations differ from the corrected degeneracies only by the same constant factor, i.e.,  $\mathbf{D}_{\text{corrected}} = n\mathbf{D}_{\text{counted}}$ , and *n* is the ratio of the number of whole configurations to that of reduced configurations, and have

$$n = \frac{C_K^{k_1}}{C_{K-1}^{k_1-1}} = \frac{K}{k_1},$$

where K-1 and  $k_1 - 1$  are owing to the first atomic positions already being occupied by a blue atom for the reduced configurations. Obviously, to make *n* as large as possible, we can always let  $k_1 \le k_2$ . For the example of K = 6,  $k_1 = 3$ , and  $k_2 = 3$ , the number of reduced configurations is  $C_{K-1}^{k_1-1} = C_5^2 = 10$ , which is half of the number of whole configurations  $C_K^{k_1} = C_6^3 = 20$ , so that **D**<sub>corrected</sub> = 2**D**<sub>counted</sub> for this example. Moreover, it is worth noting that *n* is not necessarily an integer but the product of *n* and the counted degeneracy is an integer.

The above discussion is based on one premise, i.e., the *K* atomic positions are indistinguishable. Although this condition is met in many systems, it is necessary to extend it to the universal situation, i.e., the *K* atomic positions are not all indistinguishable. To this end, we divide the *K* atomic positions into several groups, and the atomic positions in the same group are indistinguishable, while the order of these groups is irrelevant in principle. As shown in Fig. 4, the *K* atomic positions in each group is  $K_i(i \in [1, Ng])$ , and satisfying  $K = \sum_{i=1}^{Ng} K_i$ . Similarly, we can divide all atomic configurations



FIG. 2. The flowchart of eliminating duplicate configurations for (a) the original algorithm and (b) the tree search algorithm.

into Ng groups according to which atomic position group the first label of the atomic configuration belongs to. More specifically, the configurations whose first label belongs to the same atomic position group will be regarded as the same atomic configuration group. However, one can notice that the first label of the last atomic configuration is  $K - k_1 + 1$ . Therefore, the atomic configuration group Ng disappears when  $k_1 > K_{Ng}$ . Similarly, the atomic configuration groups Ng and Ng-1 disappear when  $k_1 > K_{Ng} + K_{Ng-1}$ , and so on. Consequently, the number of atomic configuration groups is Ng' and satisfying  $1 \leq Ng' \leq Ng$ .

Afterward, we define a list **G** with a size of Ng'; its elements are the label of the first atomic position for each group, i.e.,  $\mathbf{G} = (1, K_1 + 1, K_1 + K_2 + 1, \dots, \sum_{i=1}^{Ng'-1} K_i + 1)$ . Like the case of Ng = 1 (i.e., all atomic positions are indistinguishable), we can search the irreducible configurations for each group independently. Specifically, for the group  $l(l \in [1, Ng'])$ , the configuration that does not start with  $\mathbf{G}(l)$  can be converted to the configuration that starts with  $\mathbf{G}(l)$  by a symmetry operation; therefore, the configurations that do not start with  $\mathbf{G}(l)$  are discarded directly. Figure 2(b) shows the flowchart of eliminating duplicate configurations for the tree



FIG. 3. The two treelike structures for binary site occupancy in the special case of K = 6,  $k_1 = 3$  (blue), and  $k_2 = 3$  (red). The solid spheres stand for the irreducible configurations, the hollow spheres with solid lines denote the duplicate configurations, and those with dashed lines represent atoms that do not need to be enumerated in practice.

search algorithm after extending it to the general case. As an aside, the atomic position labels of the same group must be consecutive; otherwise, the atomic configuration groups or the



FIG. 4. The grouping of atomic positions. The symmetrically equivalent atomic positions are regarded as the same group, and the atomic position labels of the same group are consecutive.

branches of the tree, as stated above, may not be independent, i.e., the atomic configuration of one group may be equivalent to that of another group.

It is not difficult to find that the number of reduced configurations for the group *l* is  $Nc(l) = C_{K-K_1-K_2-\cdots K_{l-1}-1}^{k_1-1}$ . Therefore, the ratio of the number of whole configurations to that of reduced configurations is

$$n = \frac{C_K^{k_1}}{C_{K-1}^{k_1-1} + C_{K-K_1-1}^{k_1-1} + C_{K-K_1-K_2-1}^{k_1-1} + \dots + C_{K-K_1-K_2-\dots+K_{N_K-1}-1}^{k_1-1}}.$$

One can see that the above formula becomes  $n = K/k_1$ , i.e., the situation of Ng = 1, when the tail (i.e.,  $C_{K-K_1-1}^{k_1-1} + C_{K-K_1-K_2-1}^{k_1-1} + \cdots + C_{K-K_1-K_2-\cdots-K_{Ng'-1}-1}^{k_1-1}$ ) in the denominator is not present. Accordingly, we have  $n \leq K/k_1$  in which the equal sign holds if and only if Ng' = 1. In addition, it is hard to know how to make *n* as large as possible because it is related to the order of the atomic types (i.e.,  $k_1, k_2$ ) as well as that of the atomic position groups (i.e.,  $K_1, K_2, \ldots, K_{Ng'}$ ). For the sake of simplicity, we still let  $k_1 \leq k_2$ , and on this premise, we should let  $K_1 \geq K_2 \geq K_3 \geq \ldots \geq K_{Ng'}$  to make *n* as close to  $K/k_1$  as possible.

Similarly, we need to perform a degeneracy correction procedure to obtain the corrected degeneracies. Unfortunately, unlike the degeneracy correction for Ng = 1, i.e., the counted degeneracies of all irreducible configurations differ from the corrected degeneracies only by the same constant factor, that for Ng > 1 is much more complicated. Firstly, we define a list of  $(k_{11}, k_{12}, \ldots, k_{1Ng})$ , where  $k_{1l}$  is the number of atoms occupied in atomic position group l for the atomic type corresponding to  $k_1$  and satisfying  $k_1 = \sum_{l=1}^{Ng} k_{1l}$ . Given an irreducible configuration, there is a list  $(k_{11}, k_{12}, \ldots, k_{1Ng})$ corresponding to it, and for such a configuration, its counted degeneracy differs from the corrected degeneracy only by the constant factor, i.e.,  $\mathbf{D}_{corrected} = n'\mathbf{D}_{counted}$ , where

$$n' = \frac{C_{K_1}^{k_{11}} C_{K_2}^{k_{12}} \cdots C_{K_l}^{k_{ll}} C_{K_{l+1}}^{k_{l+1}} C_{K_{l+2}}^{k_{l+2}} \cdots C_{K_{N_g}}^{k_{1N_g}}}{C_{K_1}^{k_{11}} C_{K_2}^{k_{12}} \cdots C_{K_{l-1}}^{k_{l+1}} C_{K_{l+1}}^{k_{l+1}} C_{K_{l+2}}^{k_{l+2}} \cdots C_{K_{N_g}}^{k_{1N_g}}} = \frac{C_{K_l}^{k_{ll}}}{C_{K_l-1}^{k_{l-1}}} = \frac{K_l}{k_{l+1}}$$

for the configurations belonging to group  $l(l \in [1, Ng'])$ . In reality, therefore, for each irreducible configuration belonging to group l, only  $k_{1l}$  in the list of  $(k_{11}, k_{12}, \ldots, k_{1Ng})$  is required, while other elements are useless. It should be noted that, however, for different configurations belonging to the same group, the  $k_{1l}$  may be different.

#### B. Multinary site occupancy

Like binary site occupancy, the tree search algorithm for multinary site occupancy is also based on a treelike structure, as shown in Fig. 5(a): an example for K = 6,  $k_1 = 3$ (blue),  $k_2 = 2$  (red), and  $k_3 = 1$  (green), i.e., the ternary site occupancy. To implement such a treelike structure, we will encounter the same difficulty as that for binary site occupancy, i.e., the branches of the tree are not independent. Obviously, the treelike structure shown in Fig. 5(a) can be degenerated to that shown in Fig. 5(b), just as we did for binary site occupancy. Meanwhile, the implementation of the treelike structure shown in Fig. 5(b) for ternary site occupancy is exactly the same as that shown in Fig. 3(b) for binary site occupancy and shares the same flowchart as presented in Fig. 3(b). Nevertheless, we have an alternative treelike structure [presented in Fig. 5(c)] used to search irreducible configurations for multinary site occupancy, which is much better than that shown in Fig. 5(b). The branch dependence of the tree is caused by the permutation symmetry, which works only on the same color (i.e., atoms of the same type). For



FIG. 5. The three treelike structures for ternary site occupancy in the special case of K = 6,  $k_1 = 3$  (blue),  $k_2 = 2$  (red), and  $k_3 = 1$  (green). The solid spheres represent the irreducible configurations, the hollow spheres with solid lines stand for the duplicate configurations, and those with dashed lines denote the atoms that do not need to be enumerated in practice.

different colors of the treelike structure presented in Fig. 5(c), the permutation symmetry disappears, so the branch dependence of the tree disappears too, i.e., the branches of the tree are independent. As an aside, although a similar treelike structure is also adopted in ENUMLIB code [16], its efficiency is far lower than our algorithm. In the following, the specific implementation of the treelike structure presented in Fig. 5(c) will be discussed in detail.

The tree search of irreducible configurations for multinary site occupancy is hierarchical by colors: blue  $(k_1)$  is the first level, red  $(k_2)$  is the second level, green  $(k_3)$  is the third level, and so on. For binary site occupancy with a stoichiometry of  $k_1 : K - k_1$ , the tree search only needs to proceed down to the first level; for ternary site occupancy with a stoichiometry of  $k_1 : k_2 : K - k_1 - k_2$ , the tree search needs to proceed down to the second level; and for *N*-nary site occupancy with a stoichiometry of  $k_1 : k_2 : ... : k_{N-1} : K - k_1 - k_2 - \cdots - k_{N-1}$ , the tree search needs to go deep into the *N*-1 level. Moreover, for *N*-nary site occupancy with a stoichiometry of  $k_1 : k_2 : ... : k_{N-1} : K - k_1 - k_2 - \cdots - k_{N-1}$ , the procedure of tree search from the first level to the *N*-2 level is fully consistent with that for (*N*-1)-nary site occupancy with a stoichiometry of  $k_1 : k_2 : ... : k_{N-2} : K - k_1 - k_2 - \cdots - k_{N-2}$ . Furthermore, we can also divide the irreducible configurations by colors, such as the irreducible configurations for blue color (BICs), the irreducible configurations for green color (GICs), while the complete irreducible configurations are obtained by merging these irreducible configurations.

Take ternary site occupancy as an example, as shown in Fig. 5(c). The tree search is first carried out on the first level (i.e., the blue color), whose implementation is consistent with that for binary site occupancy, as stated above. For different BICs, their descendants are uncorrelated, so that the search of irreducible configurations for the descendants of each BIC can be performed independently. Moreover, in



FIG. 6. The flowchart of eliminating duplicate configurations for the tree search algorithm of quaternary site occupancy. Note that the letters with different colors represent different variables.

the process of eliminating duplicate configurations for each BIC, the symmetry operations which leave the current BIC unchanged are recorded, i.e., the so-called stabilizer subgroup [16]. For different BICs, their stabilizer subgroup may be different. Thereafter, in the process of eliminating duplicate configurations for the RICs, only the corresponding stabilizer subgroup is applied, while other symmetry operations are skipped directly, which greatly improves the search efficiency.

Based on ternary site occupancy, we can proceed down the tree search for quaternary site occupancy. Similarly, for different RICs, their descendants are uncorrelated too, so that the search of irreducible configurations for the descendants of each RIC can be performed independently. In the process of eliminating duplicate configurations for each RIC, the symmetry operations which leave the current RIC unchanged are also recorded and used to eliminate duplicate configurations for the GICs. Obviously, the stabilizer subgroup for each RIC is a subset of that for its parent (i.e., the BIC), which means that the stabilizer subgroup will get smaller with the deepening of the level and speeds up the search of irreducible configurations. By parity of reasoning, we can proceed down the tree search of irreducible configurations for quinary, senary, and septenary site occupancies, etc.

Obviously, a degeneracy correction procedure is also needed for the tree search algorithm of multinary site occupancy. For irreducible configurations of different colors, we can count their degeneracies respectively in the process of eliminating their duplicate configurations. After the degeneracies of BICs are corrected according to the degeneracy correction procedure for binary site occupancy, as presented in Sec. III A, the correct degeneracies of the merged irreducible configurations can be obtained by multiplying the degeneracies of all colors. Take Fig. 5(c) as an example again. For instance, the counted degeneracy for 1 3 4 is 2, that for 2 5 is 2, 5 6 is 1, and the corrected degeneracy for 1 3 4 is 2  $\times \frac{2}{1} = 4$ . Therefore, the corrected degeneracy for 1 3 4 2 5 is  $4 \times 2 = 8$ , and that for **1 3 4 5 6** is  $4 \times 1 = 4$ . Another example, the counted degeneracy for 3 4 5 is 3, that for 1 2 is 1, 1 6 is 2, and the corrected degeneracy for 3 4 5 is  $3 \times \frac{4}{3} = 4$ . Therefore, the corrected degeneracy for **3** 4 **5** 1 2 is  $4 \times 1 = 4$ , and that for 3 4 5 1 6 is  $4 \times 2 = 8$ . Lastly, Fig. 6 plots the flowchart of eliminating duplicate configurations for the tree search algorithm of quaternary site

TABLE I. The performance comparison of the tree search algorithm and the original algorithm for binary site occupancy in a $2 \times 2$	$2 \times 2$
fcc supercell. $n_1$ is the ratio of the run time of the original algorithm to that of the tree search algorithm, $n_2$ is the ratio of the number of w	vhole
configurations to that of reduced configurations, i.e., $K/k_1$ for this example.	

			Run time (s)				
Stoichiometries	Total configurations	Irreducible configurations	Original algorithm	Tree search algorithm	$n_1$	$n_2$	$n_1/n_2$
1:31	32	1	0.034	0.033	1.030	32.000	0.032
2:30	496	5	0.033	0.032	1.021	16.000	0.064
3:29	4960	14	0.036	0.033	1.075	10.667	0.101
4:28	35 960	71	0.049	0.036	1.355	8.000	0.170
5:27	201 376	223	0.087	0.053	1.636	6.400	0.256
6:26	906 192	874	0.233	0.115	2.027	5.333	0.380
7:25	3 365 856	2706	0.714	0.284	2.518	4.571	0.551
8:24	10 518 300	8043	2.114	0.769	2.748	4.000	0.687
9:23	28 048 800	20123	6.274	2.262	2.774	3.556	0.780
10:22	64 512 240	45 497	16.950	6.229	2.721	3.200	0.850
11:21	129 024 480	88716	38.817	14.884	2.608	2.909	0.897
12:20	225 792 840	154 379	77.104	31.180	2.473	2.667	0.927
13:19	347 373 600	234 803	132.702	57.900	2.292	2.462	0.931
14:18	471 435 600	318 348	202.198	93.186	2.170	2.286	0.949
15:17	565 722 720	379 926	263.182	129.495	2.032	2.133	0.953
16:16	601 080 390	404 582	302.959	155.311	1.951	2.000	0.976
Cumulative	2 448 023 842	1 658 311	1043.487	491.802	2.122	2.280	0.931

occupancy, while that for other multinary site occupancies are similar.

## **IV. PERFORMANCE COMPARISON**

At present, we have elaborated the implementation details of the aforementioned tree search algorithm and have implemented it in DISORDER code [40]. In this section, we will discuss the performance of the tree search algorithm and compare it with the original algorithm through three examples, i.e., the fcc parent lattice used for the binary site-occupancy performance test, the bcc parent lattice used for the ternary site-occupancy performance test, and the simple cubic (sc) parent lattice used for the quaternary site-occupancy performance test. The fcc unit cell possesses the point group symmetry of  $O_h$ , which contains 48 rotation operations and four pure translation operations. Here, a 2 × 2 × 2 fcc supercell (32 atoms), including 1536 space group operations, i.e., 48 rotation operations and 32 pure translation operations, is adopted to test the algorithm performance for binary site occupancy. We enumerate all combinatorially distinct stoichiometries for binary site occupancy, which are used to test the performance of the original algorithm and the tree search algorithm. Table I shows the results of the performance test, from which we can see that the overall efficiency of the tree search algorithm is more than double that of the original algorithm. Additionally, an interesting phenomenon is that, with the increase of  $k_1$ , the ratio  $(n_1)$  of the running time of the original algorithm to that of the tree search algorithm approaches the ratio  $(n_2)$ 

TABLE II. The performance comparison of the tree search algorithm and the original algorithm for ternary site occupancy in a  $2 \times 2 \times 3$  bcc supercell. *n* is the ratio of the run time of the original algorithm to that of the tree search algorithm.

Stoichiometries			Run		
	Total configurations	Irreducible configurations	Original algorithm	Tree search algorithm	n
1:11:12	32 449 872	90 4 2 1	7.025	0.614	11.436
2:11:11	194 699 232	526708	45.880	2.260	20.298
3:10:11	713 897 184	1 904 378	45.880	4.226	42.991
4:10:10	1 963 217 256	5 202 128	543.573	10.234	53.116
5:9:10	3 926 434 512	10 359 102	1124.508	16.516	68.087
6:9:9	6 544 057 520	17 226 595	1949.887	26.816	72.713
7:8:9	8 413 788 240	22 126 976	2610.849	34.105	76.553
8:8:8	9465511770	24 888 438	3046.076	34.503	88.283
Cumulative	31 254 055 586	82 324 746	9509.481	129.275	73.560

			Run		
Stoichiometries	Total configurations	Irreducible configurations	Original algorithm	Tree search algorithm	n
1:6:6:7	931 170 240	11 671 868	882.185	20.259	43.546
2:6:6:6	3 259 095 840	40 814 088	3406.154	69.582	48.952
3:5:6:6	6 518 191 680	81 584 076	7123.834	138.308	51.507
4:5:5:6	9777287520	122 352 972	11 146.139	197.344	56.481
5:5:5:5	11732745024	146 809 254	14 325.498	236.461	60.583
Cumulative	32 218 490 304	403 232 258	36 883.810	661.953	55.720

TABLE III. The performance comparison of the tree search algorithm and the original algorithm for quaternary site occupancy in a  $2 \times 2 \times 5$  sc supercell. *n* is the ratio of the run time of the original algorithm to that of the tree search algorithm.

of the number of whole configurations to that of reduced configurations, which is reflected in the fact that  $n_1/n_2$  gets closer to 1. The reason is that, compared with the preprocessing time of the algorithm, the time of eliminating duplicate configurations, i.e., the procedure that our tree search algorithm works, is gradually dominant with the increase of  $k_1$ .

The performance test for ternary site occupancy is based on a  $2 \times 2 \times 3$  bcc supercell (24 atoms), while that for quaternary site occupancy is grounded in a  $2 \times 2 \times 5$  sc supercell (20 atoms). Because of the inhomogeneous cell expansion, the point group symmetry of  $O_h$  for a bcc unit cell (48 rotation operations and two pure translation operations) is reduced to the point group symmetry of  $D_{4h}$  for a bcc supercell (16 rotation operations and 24 pure translation operations, i.e., 384 space group operations). Similarly, the point group symmetry of  $O_h$  for a sc unit cell (48 rotation operations and one pure translation operations) is also reduced to the point group symmetry of  $D_{4h}$  for a sc supercell (16 rotation operations and 20 pure translation operations, i.e., 320 space group operations). Due to the large number of combinatorially distinct stoichiometries for ternary and quaternary site occupancies, we only select some representative stoichiometries used for performance tests, as presented in Table II for ternary site occupancy and Table III for quaternary site occupancy. It is surprising that the performance of the tree search algorithm is dramatically improved; the overall efficiency of the tree search algorithm is improved by >70 times for ternary site occupancy and >50 times for quaternary site occupancy compared with that of the original algorithm. As an aside, we have reason to believe that the performance will be improved even more for the systems with higher symmetry, i.e., possesses more space group operations.

#### V. APPLICATION EXAMPLE

As an application example, we use our tree search algorithm to generate the optimal SQSs [33] for ternary bcc and fcc  $A_1B_1C_1$  alloys, which are an extremely challenging task for the most widely used SQS generation code, i.e., the GENSQS code in the Alloy Theoretic Automated Toolkit (ATAT) [41]. Specifically, we built 75 symmetrically distinct supercells (27 atoms) from a bcc/fcc primitive cell (1 atom) and obtained 312 446 988 626 irreducible site-occupancy configurations out of 17 090 507 362 500 complete site-occupancy configurations by using our tree search algo-

rithm. In such an extremely huge configuration space and a short time of <2 d (24 threads), we searched out 24 (2) optimal SQSs, whose pair/triplet correlation functions perfectly match those of the corresponding truly random alloys up to the eighth-nearest/third-nearest (ninth-nearest/first-nearest) neighbor, for ternary bcc (fcc) A<sub>1</sub>B<sub>1</sub>C<sub>1</sub> alloys. Note that, according to relevant tests, we estimate that the time may be several years when the GENSQS code is used for the same task and the same calculation conditions. Figure 7



SQS-27 for ternary bcc  $A_1B_1C_1$  alloys



SQS-27 for ternary fcc  $A_1B_1C_1$  alloys

FIG. 7. Atomic arrangements of the optimal special quasirandom structures (SQSs), in their ideal, unrelaxed forms, for (a) bodycentered cubic (bcc) and (b) face-centered cubic (fcc)  $A_1B_1C_1$  alloys. For each type of alloy, only one of the optimal SQSs is randomly selected as a representative. TABLE IV. Structural descriptions of the optimal SQSs (ideal and unrelaxed forms) for bcc and fcc  $A_1B_1C_1$  alloys. The lattice vectors and atomic positions are given in Cartesian coordinates, in units of *a*, the bcc or fcc lattice parameter. For each type of alloy, only one of the optimal SQSs is randomly selected as a representative.

Alloys		bcc A <sub>1</sub> B <sub>1</sub> C <sub>1</sub> (SQS-27)				fcc $A_1B_1C_1$ (SQS-27)		
Lattice vectors	a	1.5	-1.5	1.5	a	-1.5	1.5	0
	b	-2	-3	-1	b	-0.5	-1	-0.5
	c	1	0	-1	с	-0.5	-1	2.5
Atomic positions	А	0	0	0	А	0	0	0
	А	0	-4	0	А	-2	0	2
	А	0.5	-3.5	-0.5	А	-2	-0.5	1.5
	А	0.5	-2.5	-0.5	А	-1	1	0
	А	0	-1	0	А	-1.5	-1	1.5
	А	0.5	-0.5	0.5	А	-1	-1	1
	А	1	-1	0	А	-0.5	-1	1.5
	А	1	-1	1	А	-1	0.5	0.5
	А	-0.5	-1.5	-0.5	А	-1	0	0
	В	0.5	-1.5	-0.5	В	-1.5	0	1.5
	В	0.5	-0.5	0.5	В	-1.5	-0.5	1
	В	-1	-3	-1	В	-1	0	1
	В	1	-3	0	В	-1	-0.5	1.5
	В	0.5	-2.5	0.5	В	-0.5	-0.5	1
	В	1	-2	0	В	-1.5	1	0.5
	В	0.5	-1.5	0.5	В	-1	-1	2
	В	0	-3	-1	В	-1	-1.5	1.5
	В	0	-1	-1	В	-1.5	0	0.5
	С	0	-3	0	С	-1	-0.5	0.5
	С	0	-2	0	С	-0.5	0	0.5
	С	-0.5	-2.5	-1.5	С	-0.5	-0.5	0
	С	-1	-2	-1	С	-1.5	-0.5	2
	С	-0.5	-3.5	-0.5	С	-0.5	-1	0.5
	С	-0.5	-2.5	-0.5	С	-1.5	0.5	0
	С	0	-2	-1	С	-0.5	0.5	0
	С	1	-2	1	С	-2	0	1
	С	1.5	-1.5	0.5	С	-1.5	0.5	1

presents the atomic arrangements of the optimal SQSs (ideal and unrelaxed forms) for bcc and fcc  $A_1B_1C_1$  alloys, respectively (only one of the optimal SQSs is randomly selected as a representative for each type), and Table IV lists the corresponding structural descriptions of these SQSs, which can be widely used as standard SQSs to study various properties of any ternary equiatomic bcc and fcc alloys.

Using the Monte Carlo (MC) simulated annealing technique [42,43] as implemented in the MCSQS code in the ATAT, we also generated an SQS-27 (abbreviated as MC SQS) for ternary bcc  $A_1B_1C_1$  alloys. However, its pair correlation functions merely match those of the truly random alloys up to the second-nearest neighbor, while its triplet correlation functions fail to match. Obviously, the optimal SQSs found by our algorithm are far superior to the MC SQS when evaluating the match of correlation functions. To further demonstrate the superiority, we take a real bcc Mo-Nb-V alloy as an example. We constructed 7 Mo<sub>9</sub>Nb<sub>9</sub>V<sub>9</sub> structures, i.e., six optimal SQSs (randomly selected from 24 optimal SQSs) and one MC SQS of ternary bcc A<sub>1</sub>B<sub>1</sub>C<sub>1</sub> alloys, and calculated their total energies and equilibrium volumes, after full optimization (both lattice and atomic pos itions are relaxed), using density functional theory [44] as implemented in the Vienna Ab initio Simulation Package [45,46]. Figure 8 displays the calculated total energies and equilibrium volumes of the 7 Mo<sub>9</sub>Nb<sub>9</sub>V<sub>9</sub> structures, which are referenced to the average total energy and equilibrium volume of the six optimal SQSs. We can see that, compared with the total energy and equilibrium volume of the optimal SQSs, those of the MC SQS deviate evidently from the zero of energy and volume, i.e., the average total energy and equilibrium volume of the optimal SQSs, which approximately represents the ensemble average of ternary bcc  $A_1B_1C_1$  alloys. Although such deviation in MC SQS is not very big, it would be obvious for other physical properties that are required to be modeled by a longer-range CE. Therefore, the SQSs obtained from our tree search algorithm are much more reliable because their correlation functions are perfectly matched to those of the truly random alloys in a longer range.



FIG. 8. The calculated total energies and equilibrium volumes of body-centered cubic (bcc)  $Mo_9Nb_9V_9$  structures [six optimal special quasirandom structures (SQSs) and 1 Monte Carlo (MC) SQS], in which the average total energy (equilibrium volume) of the six optimal SQSs is chosen as the zero of energy (volume).

### VI. SUMMARY

In summary, we have developed a tree search algorithm for irreducible site-occupancy configurations based on our original algorithm with high efficiency, by leveraging the idea of searching for irreducible site-occupancy configurations in a treelike structure. In the tree search algorithm, only the

- J. Blasco, J. García, J. M. de Teresa, M. R. Ibarra, J. Perez, P. A. Algarabel, C. Marquina, and C. Ritter, Structural, magnetic, and transport properties of the giant magnetoresistive perovskites La<sub>2/3</sub>Ca<sub>1/3</sub>Mn<sub>1-x</sub>Al<sub>x</sub>O<sub>3-δ</sub>, Phys. Rev. B **55**, 8905 (1997).
- [2] D. Chaney, A. Castellano, A. Bosak, J. Bouchet, F. Bottin, B. Dorado, L. Paolasini, S. Rennie, C. Bell, R. Springell, and G. H. Lander, Tuneable correlated disorder in alloys, Phys. Rev. Mater. 5, 035004 (2021).
- [3] Z. W. Li, X. Z. Zhou, and A. H. Morrish, Site occupancies of Si atoms and Curie temperatures for Sm<sub>2</sub>Fe<sub>17-x</sub>Si<sub>x</sub>, Phys. Rev. B 51, 2891 (1995).
- [4] Z. W. Lu, S. H. Wei, and A. Zunger, Electronic structure of ordered and disordered Cu<sub>3</sub>Au and Cu<sub>3</sub>Pd, Phys. Rev. B 45, 10314 (1992).
- [5] S. M. Ramos, M. B. Fontes, E. N. Hering, M. A. Continentino, E. Baggio-Saitovich, F. Neto, E. M. Dinóla Bittar, P. G. Pagliuso, E. D. Bauer, J. L. Sarrao, and J. D. Thompson, Superconducting Quantum Critical Point in CeCoLn<sub>5-x</sub>Sn<sub>x</sub>, Phys. Rev. Lett. **105**, 126401 (2010).
- [6] V. I. Razumovskiy, A. V. Ruban, and P. A. Korzhavyi, Effect of Temperature on the Elastic Anisotropy of Pure Fe and Fe<sub>0.9</sub>Cr<sub>0.1</sub> Random Alloy, Phys. Rev. Lett. **107**, 205504 (2011).
- [7] H. C. Robarts, T. E. Millichamp, D. A. Lagos, J. Laverock, D. Billington, J. A. Duffy, D. O'Neill, S. R. Giblin, J. W. Taylor, G. Kontrym-Sznajd, M. Samsel-Czekała, H. Bei, S. Mu, G. D. Samolyuk, G. M. Stocks, and S. B. Dugdale, Extreme Fermi Surface Smearing in a Maximally Disordered

branches that meet the requirements are needed to descend further down the tree, while other branches are blocked; in the original algorithm, it is equivalent to all the branches of the tree being searched. Therefore, compared with the original algorithm, the efficiency of the tree search algorithm is greatly improved, especially for the multinary site-occupancy systems. Specifically, the performance test results show that the overall efficiency of the tree search algorithm for binary site occupancy in a  $2 \times 2 \times 2$  fcc supercell is more than doubled, and that for ternary (quaternary) site occupancy in a 2  $\times$  2  $\times$  3 bcc (2  $\times$  2  $\times$  5 sc) supercell is increased by > 70(50) times. Moreover, a degeneracy correction procedure is also developed to correct the wrongly counted degeneracies caused by the skipping of some branches of the tree. As an application example, the optimal SQSs (SQS-27) for ternary equiatomic fcc and bcc alloys are searched out by using the tree search algorithm, which can be widely used as standard SQSs to study various properties of any ternary equiatomic bcc and fcc alloys. Our tree search algorithm for irreducible site-occupancy configurations is universal and efficient, which can be exceedingly useful for the modeling of various compositionally complex materials.

#### ACKNOWLEDGMENTS

The authors are grateful to the National Natural Science Foundation of China (Grants No. 52172088 and No. 51772085) and the Natural Science Foundation of Hunan Province (No. 2020JJ4190 and No. 2021JJ30112).

Concentrated Solid Solution, Phys. Rev. Lett. **124**, 046402 (2020).

- [8] N. Shulumba, O. Hellman, Z. Raza, B. Alling, J. Barrirero, F. Mücklich, I. A. Abrikosov, and M. Odén, Lattice Vibrations Change the Solid Solubility of an Alloy at High Temperatures, Phys. Rev. Lett. 117, 205502 (2016).
- [9] P. Soven, Contribution to the theory of disordered alloys, Phys. Rev. 178, 1136 (1969).
- [10] B. M. Way, J. R. Dahn, T. Tiedje, K. Myrtle, and M. Kasrai, Preparation and characterization of  $B_xC_{1-x}$  thin films with the graphite structure, Phys. Rev. B **46**, 1697 (1992).
- [11] R. Grau-Crespo, S. Hamad, C. R. A. Catlow, and N. H. de Leeuw, Symmetry-adapted configurational modelling of fractional site occupancy in solids, J. Phys.: Condens. Matter 19, 256201 (2007).
- [12] S. Mustapha, P. D'Arco, M. De La Pierre, Y. Noel, M. Ferrabone, and R. Dovesi, On the use of symmetry in configurational analysis for the simulation of disordered solids, J. Phys.: Condens. Matter 25, 105401 (2013).
- [13] G. L. W. Hart and R. W. Forcade, Algorithm for generating derivative structures, Phys. Rev. B 77, 224115 (2008).
- [14] G. L. W. Hart and R. W. Forcade, Generating derivative structures from multilattices: algorithm and application to hcp alloys, Phys. Rev. B 80, 014120 (2009).
- [15] G. L. W. Hart, L. J. Nelson, and R. W. Forcade, Generating derivative structures at a fixed concentration, Comput. Mater. Sci. 59, 101 (2012).

- [16] W. S. Morgan, G. L. W. Hart, and R. W. Forcade, Generating derivative superstructures for systems with high configurational freedom, Comput. Mater. Sci. 136, 144 (2017).
- [17] K. Okhotnikov, T. Charpentier, and S. Cadars, Supercell program: a combinatorial structure-generation approach for the local-level modeling of atomic substitutions and partial occupancies in crystals, J. Cheminformatics 8, 17 (2016).
- [18] J.-C. Lian, H.-Y. Wu, W.-Q. Huang, W. Hu, and G.-F. Huang, Algorithm for generating irreducible site-occupancy configurations, Phys. Rev. B 102, 134209 (2020).
- [19] T. Brink, L. Koch, and K. Albe, Structural origins of the boson peak in metals: from high-entropy alloys to metallic glasses, Phys. Rev. B 94, 224203 (2016).
- [20] B. Gludovatz, A. Hohenwarter, D. Catoor, E. H. Chang, E. P. George, and R. O. Ritchie, A fracture-resistant high-entropy alloy for cryogenic applications, Science 345, 1153 (2014).
- [21] Y. Ikeda, K. Gubaev, J. Neugebauer, B. Grabowski, and F. Kormann, Chemically induced local lattice distortions versus structural phase transformations in compositionally complex alloys, npj Comput. Mater. 7, 34 (2021).
- [22] P. Koželj, S. Vrtnik, A. Jelen, S. Jazbec, Z. Jagličić, S. Maiti, M. Feuerbacher, W. Steurer, and J. Dolinšek, Discovery of a Superconducting High-Entropy Alloy, Phys. Rev. Lett. 113, 107001 (2014).
- [23] D. B. Miracle and O. N. Senkov, A critical review of high entropy alloys and related concepts, Acta Mater. 122, 448 (2017).
- [24] G. D. Samolyuk, Y. N. Osetsky, G. M. Stocks, and J. R. Morris, Role of Static Displacements in Stabilizing Body Centered Cubic High Entropy Alloys, Phys. Rev. Lett. **126**, 025501 (2021).
- [25] H. Song, F. Tian, Q.-M. Hu, L. Vitos, Y. Wang, J. Shen, and N. Chen, Local lattice distortion in high-entropy alloys, Phys. Rev. Mater. 1, 023404 (2017).
- [26] M. C. Troparevsky, J. R. Morris, P. R. C. Kent, A. R. Lupini, and G. M. Stocks, Criteria for Predicting the Formation of Single-Phase High-Entropy Alloys, Phys. Rev. X 5, 011041 (2015).
- [27] C. Varvenne, A. Luque, and W. A. Curtin, Theory of strengthening in fcc high entropy alloys, Acta Mater. 118, 164 (2016).
- [28] B. Yin and W. A. Curtin, First-principles-based prediction of yield strength in the RhIrPdPtNiCu high-entropy alloy, npj Comput. Mater. 5, 14 (2019).
- [29] P. Soven, Coherent-potential model of substitutional disordered alloys, Phys. Rev. 156, 809 (1967).
- [30] B. L. Gyorffy, Coherent-potential approximation for a nonoverlapping-muffin-tin-potential model of random substitutional alloys, Phys. Rev. B 5, 2382 (1972).

- [31] J. M. Sanchez, F. Ducastella, and D. Gratias, Generalized cluster description of multicomponent systems, Physica (Amsterdam) 128A, 334 (1984).
- [32] Laks, Ferreira, Froyen, and Zunger, Efficient cluster expansion for substitutional systems, Phys. Rev. B 46, 12587 (1992).
- [33] A. Zunger, S. H. Wei, L. G. Ferreira, and J. E. Bernard, Special Quasirandom Structures, Phys. Rev. Lett. 65, 353 (1990).
- [34] S.-H. Wei, L. G. Ferreira, J. E. Bernard, and A. Zunger, Electronic properties of random alloys: special quasirandom structures, Phys. Rev. B 42, 9622 (1990).
- [35] C. Jiang and B. P. Uberuaga, Efficient *Ab initio* Modeling of Random Multicomponent Alloys, Phys. Rev. Lett. **116**, 105501 (2016).
- [36] V. Sorkin, T. L. Tan, Z. G. Yu, and Y. W. Zhang, Generalized small set of ordered structures method for the solid-solution phase of high-entropy alloys, Phys. Rev. B 102, 174209 (2020).
- [37] F. Tian, A review of solid-solution models of high-entropy alloys based on *ab initio* calculations, Front. Mater. 4, 36 (2017).
- [38] A. van de Walle, Tiwary, P. M. de Jong, D. L. Olmsted, M. Asta, A. Dick, D. Shin, Y. Wang, L. Q. Chen, and Z. K. Liu, Efficient stochastic generation of special quasirandom structures, Calphad 42, 13 (2013).
- [39] A. Togo and I. Tanaka, SPGLIB: a software library for crystal symmetry search, arXiv:1808.01590.
- [40] The tree search algorithm for irreducible site-occupancy configurations has been implemented in the latest version (v0.6.0) of DISORDER code, which is available on GitHub, https://github. com/jichunlian/disorder.
- [41] A. van de Walle, M. Asta, and G. Ceder, The Alloy Theoretic Automated Toolkit: a user guide, Calphad 26, 539 (2002).
- [42] K. A. Mader and A. Zunger, Short- and long-range-order effects on the electronic properties of III-V semiconductor alloys, Phys. Rev. B 51, 10462 (1995).
- [43] I. A. Abrikosov, S. I. Simak, and B. Johansson, Locally selfconsistent Green's function approach to the electronic structure problem, Phys. Rev. B 56, 9319 (1997).
- [44] W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. 140, A1133 (1965).
- [45] G. Kresse and J. Furthmüller, Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set, Phys. Rev. B 54, 11169 (1996).
- [46] G. Kresse and J. Furthmüller, Efficiency of *ab-initio* total energy calculations for metals and semiconductors using a plane-wave basis set, Comp. Mater. Sci. 6, 15 (1996).