To appear in Computational Materials Science (`https://doi.org/10.1016/j.commatsci.2022.111886`).

An Optimized Species-Conserving Monte Carlo Method with Potential Applicability to High Entropy Alloys

Aziz Fall,[1,*] Matthew Grasinger,[2] and Kaushik Dayal[3,4,1,5]
[1]*Department of Mechanical Engineering, Carnegie Mellon University*
[2]*Materials and Manufacturing Directorate, Air Force Research Laboratory*
[3]*Department of Civil and Environmental Engineering, Carnegie Mellon University*
[4]*Center for Nonlinear Analysis, Department of Mathematical Sciences, Carnegie Mellon University*
[5]*Pittsburgh Quantum Institute, University of Pittsburgh*
(Dated: November 14, 2022)

We present a species-conserving Monte Carlo (MC) method, motivated by systems such as high-entropy alloys. Current fast local-structure MC methods do not conserve the net concentration of atomic species, or are inefficient for complex atomic systems. By coarse-graining the atomic lattice into clusters and developing a renormalized MC method that takes advantage of the local structure of the atoms, we are able to significantly reduce the number of iterations required for MC simulations to reach equilibrium. In addition, the structure of the method enables easy parallelizability for the future.

1. Introduction

Understanding, and hence optimizing, the composition of complex multi-species materials that display unique physical and chemical properties can be challenging due to the very large number of experiments required and the computational complexity needed to simulate them [GRR19]. For instance, high-entropy alloys (HEA) such as CrMnFeCoNi [GRR19], are an important example of materials that are difficult to discover experimentally, but are also computationally demanding to simulate. Their unique mechanical and electrical properties, radiation tolerance, and oxidization resistance makes them promising for a wide range of applications, including nuclear reactors, combustion chambers, superconducting devices, photothermal conversion, and for applications in biomedical devices [DPAM19, YZ20, GV19, CWL$^+$22, CWL$^+$22, ZYL$^+$18].

An important aspect of understanding HEA, and multi-species materials in general, is to elucidate the local or short-range order of the different atomic types, which is hypothesized to be the origin of their unique properties [WZY$^+$21, CWL$^+$22]. The large number of atomic species in these systems limits their experimental discovery, and the high configurational entropy causes the simulations to have long convergence times [FDG$^+$20]. Simulations of HEA typically predict their short-range order using quantum mechanical calculations coupled with Monte Carlo simulations [TCWSNM$^+$17]. Recent efforts to speed up the simulations have primarily focused on reducing the computational complexity of the quantum mechanical calculations by using data-driven methods to learn a local, reduced-dimensional, empirical interatomic potential that is a function of only nearest neighbor atoms [LZY$^+$21, Sha17, KRN$^+$20]. In this work, we propose a complementary speedup to MC simulations that can enable the discovery of new HEA and materials with complex atomic arrangements by reducing the computational barrier of their simulations.

1.A. Relevant Prior Approaches to Increase the Efficiency of Monte Carlo Simulations

An important advance in a Monte Carlo method that exploits the local structure of atoms to speed up the simulation was proposed by Swendsen and Wang [SW87], and related work in this direction includes [HR04, Lui06, KD91, Kat09]; this class of methods has deep connections to the renormalization group. Another important class of methods are the Kinetic Monte Carlo/N fold way [BKL75, Mur01, Sch08, Vot07]. The approaches based on Swendsen and Wang Algorithm cannot be used to simulate HEA because the fundamental structure of the method does not conserve the total concentration of different atomic species. Kinetic Monte Carlo, on the other hand, is computationally infeasible for HEAs because it requires exhausting all possible pairwise transitions between all local states (i.e., nearest neighbor configurations), and HEAs have vast configurational entropy. We note here that although replica exchange can be used to speed up MC simulations of HEA [LZY$^+$21], it is not a local-structure technique, since it requires non-local exchanges between configurations from parallel runs [SW86].

1.B. The Proposed Approach

An important element of the method proposed in this paper is to identify, via simulated annealing, hierarchical structures of groups of atoms – that we denote as a *cluster* – that do not mix at low temperatures in order to sample configuration space more efficiently. It is well known that naive methods for sampling the canonical ensemble suffer from slow convergence when the potential energy is characterized by a large number of local minima separated by high barriers [Tuc10]. Thus, by "efficient", we

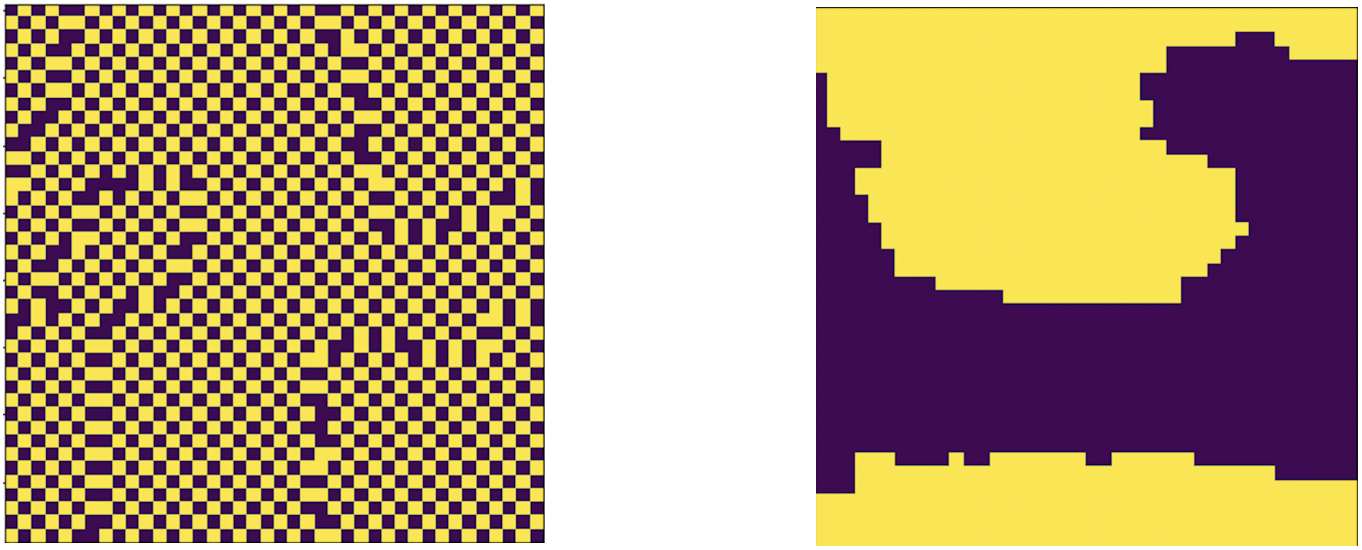
---

[*] afall@andrew.cmu.edu

mean that we design our sampling approach such that it can traverse energy barriers in configurational space without degrading the acceptance rate of the simulation (i.e. the number of accepted moves divided by the total number of iterations). We design our MC method such that we can tune a set of parameters that can control the degree of mixing and growth of clusters of atoms of some predefined structure. Our driving hypothesis – that we demonstrate successfully – is that if we choose the predefined structure of the clusters to be the structure of the clusters of atoms that do not mix at low temperatures, then we can control the degree of mixing and growth of these clusters in our MC sampling scheme to greatly speed up the approach to equilibrium in our simulations at all temperatures. We denote the structure of the cluster of atoms at low temperature to be the low-temperature *Cluster Properties* of the simulation [Fig. 1]. By first finding the optimal cluster structures and then sampling them directly, we effectively decouple the problem into numerical optimization followed by an accelerated Monte Carlo simulation. Once we identify the optimal cluster structures or the low-temperature Cluster Properties, we can control how much clusters of these structures mix and distribute themselves across the simulation cell by changing how often we swap between pairs of atoms on the boundary or interior of a cluster. This is significant because if the simulation finds stable local-structures of atoms or configurations of atoms that correspond to low energy/temperature, the simulation can be made less likely to modify those structures by decreasing the likelihood of swapping single pairs of atoms that are both in the interior of a cluster. It is relevant here to note that sampling using local cluster structures – as opposed to strictly using atom-atom swaps – has possible connections to methods in renormalization more broadly; see, for example, [Swe79, Wil75, Kad66].

A high-level summary of our strategy is outlined below:

1. Find the set of stable local structures or the low temperature atomic configurations
2. Identify all clusters of atoms with these local structures in the simulation
3. Hypothesize a probability distribution for drawing new samples that is optimized for growing and mixing clusters based on the number of interior and boundary atoms of different local structures
4. Use the probability distribution to propose new samples in the MC simulation


(a)

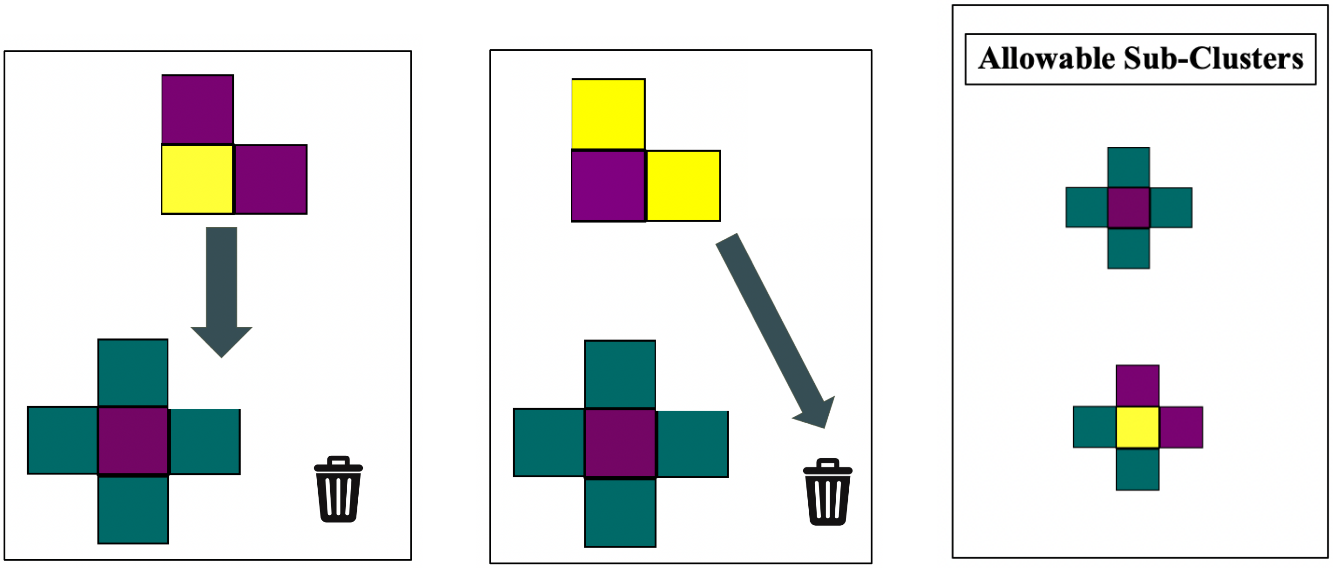
(b)

Figure 1. Example of simulation results for a binary alloy Ising Model system with fixed composition; the colors indicate different spin up and spin down. Left: an example of a low temperature configuration where the coupling constant between the spins is negative ($J < 0$); this results in three low-temperature Cluster Properties, two of which are clusters of aligned spins (i.e., either all aligned up or down) and the other is a cluster in which the spins are anti-aligned. Right: an example of a low temperature configuration where the coupling constant between the spins is positive ($J > 0$). This results in two low-temperature Cluster Properties consisting of clusters of spins that are aligned.

Structure of the Paper. In Section 2, we define key geometric properties of clusters of atoms. In Section 3, we discuss the identification of low-temperature cluster properties. In Section 4, we outline the MC method that accounts for the cluster swaps and detailed balance. In Section 5, we show the results of the proposed method.

2. Geometric Properties of Clusters

We first define a *cluster* as a group of atoms which form a connected subset of a configuration of the system. Eventually we will wish to identify ways of constructing clusters which are at a lower energy state in their interior and, as a consequence, have a significant portion of their energy concentrated at their boundary. The process of constructing clusters is driven by the



(a) A group of atoms joins an existing cluster.

(b) A group of atoms cannot join an existing cluster.

(c) Allowable sub-clusers: the local configurations of atoms in the **AS**.

Figure 2. Rules of cluster growth in a simple example; colors represent the atomic species. In (a), the group of atoms joins the existing cluster since after being added they will have a local configuration that is the same as one of the configurations in the **AS** (shown in (c)). In (b), the group of atoms cannot join the existing cluster since they will not have a local configuration that is the same as one of the configurations in the **AS**, regardless of how it is rotated/reflected or placed.

specification of Allowable Sub-Clusters (**AS**). We give an example of how a group of atoms can join an existing cluster in the simulation in Figures 2a and 2b, and use this example to generalize to different possible clusters. The rules of adding atoms to a cluster, as depicted in Figure 2a, are that

1. The atom or group of atoms must neighbor or be connected to another atom in the cluster
2. The atom or group of atoms can only be added to a cluster if the atom(s) will – upon being added – have a local configuration (i.e., arrangement of neighboring atoms) that is among those in the Allowable Sub-Clusters **AS**.

Definition 1 (Allowable Sub-Cluster (**AS**))**.** An Allowable Sub-Cluster is a possible local configuration (i.e., an arrangement of atoms) that an atom must have to be part of a cluster

Figure 2b is an example by which the set of atoms does not satisfy the second rule, so it is not added to the cluster of atoms. The rules of cluster construction in general should be defined such that any given subset of a cluster is equivalent – up to the symmetry of the Euclidean group, $E(3)$ (i.e., the group of isometries of $\mathbb{R}^3$) – to an element of the Allowable Sub-Clusters when there is no orientational energy between atoms. Since at all times each atom in a cluster must have a local configuration that is in the **AS**, the number of atoms in one local configuration in the **AS** defines the minimum possible cluster size we can build with this property. In general, we can define an **AS** with any number of atoms in its local configurations. We refer to the term Cluster Minsize to define the minimum possible size of a cluster. The possible Cluster Minsizes are denoted Minsize-1 (meaning the minimum possible cluster size is 1), Minsize-2, and so on (Fig. 3). In Figures 2a and 2b, both clusters are Cluster Minsize-5.

Definition 2 (Cluster Minsize)**.** The Cluster Minsize is the minimum possible cluster size given a set of Allowable Sub-Clusters. It is equal to the number of atoms in an Allowable Sub-Cluster.

In general, it is optimal to use only a subset of all the possible local configurations in the **AS** of a given Cluster Minsize and number of unique atomic species. For example, a Cluster Minsize-5 with three possible atomic species will have a total of 3^5 or 243 possible local configurations, but in Figures 2a and 2b we have only two local configurations.

The combination of the Cluster Minsize and **AS** defines the Cluster Property and dictates the rules by which a cluster is allowed to grow. Notice that for Cluster Properties with two or more atomic species, the Cluster Minsize and number of configurations in the **AS** controls the cluster entropy, i.e., the number of possible atomic configurations of a cluster that are still of the same Cluster Property. In general, higher Cluster Minsizes along with fewer configurations in the **AS** result in Cluster Properties with low cluster entropy.

Definition 3 (Cluster Property)**.** A Cluster Property is the combination of Cluster Minsize and a set of Allowable Sub-Clusters that dictates the rules by which a cluster is allowed to grow.

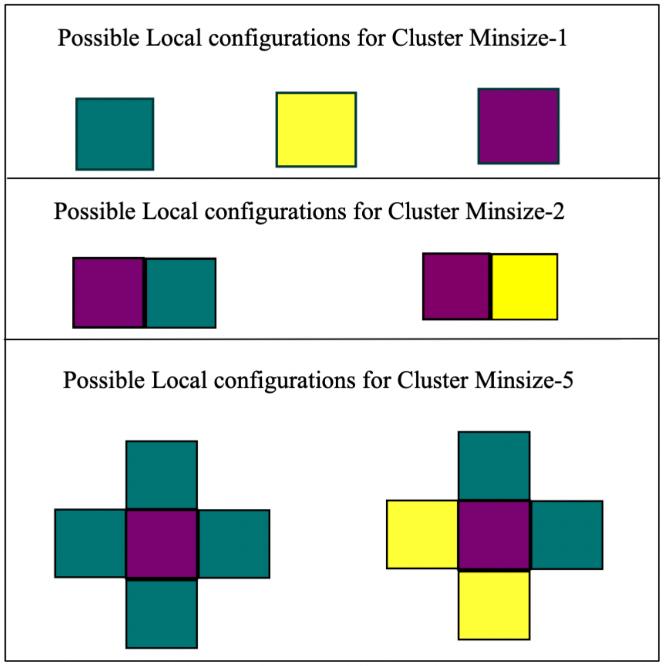
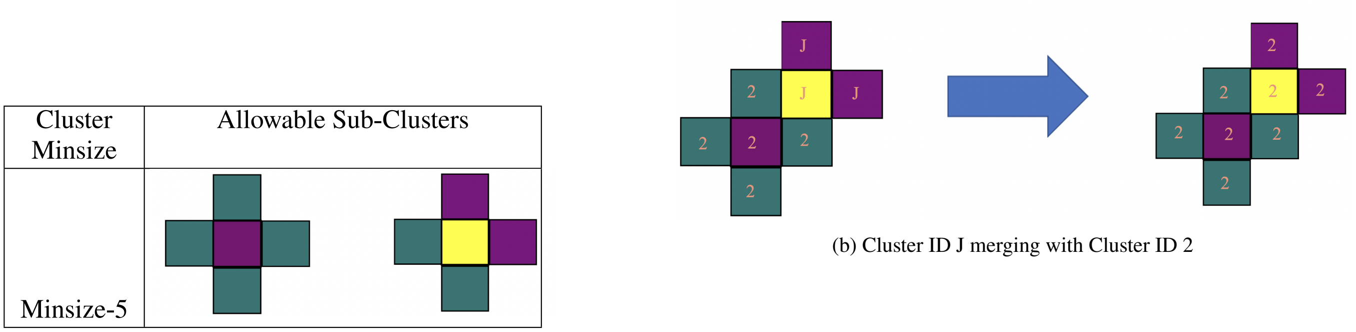


Figure 3. Examples of local configurations for different Cluster Minsizes in 2D. The atomic species are represented by different colored squares.



(a) Cluster Property 1

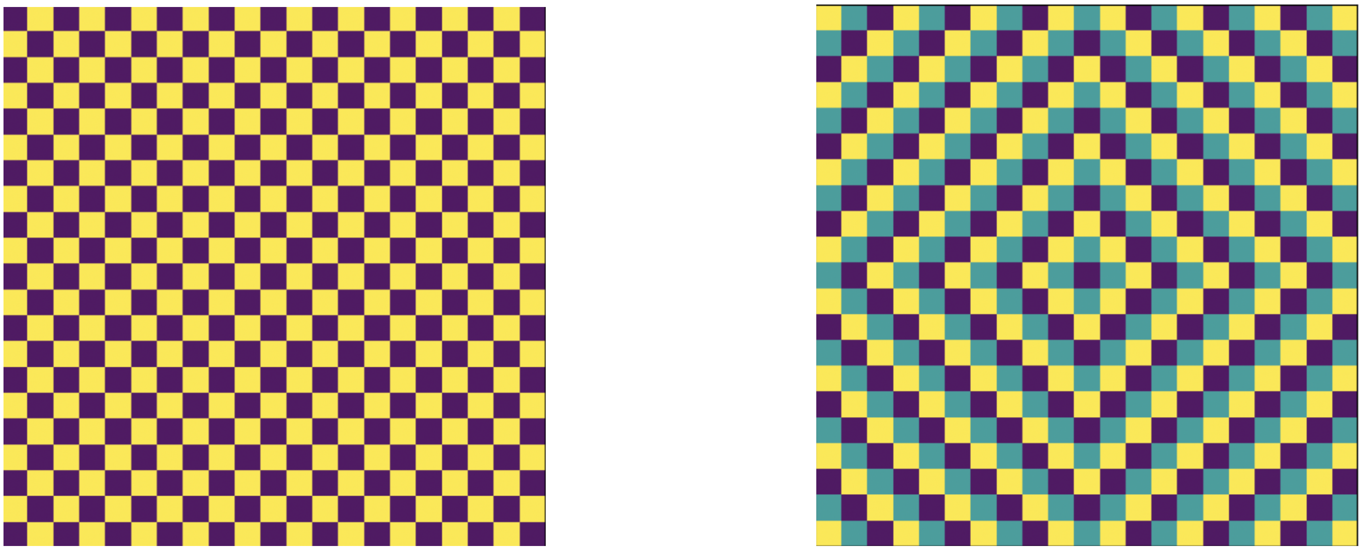
(b) Cluster ID J merging with Cluster ID 2

Figure 4. A example by which an atom can join an existing cluster. A Cluster Property is defined with Cluster Minsize-5 and a subset of 2 local configurations chosen out of the possible 243 (i.e. 3^5) local configurations of three atomic species in Cluster Minsize-5. The atoms marked with the label 2 indicate they belong to cluster ID 2, and those labeled with J are of an unknown cluster ID. Since the atom with the yellow atomic species in the bottom part of the figure matches one of the local configurations of cluster property 1, it joins cluster ID 2 along with its neighbors.

At any time in the simulation, there can be any number of clusters of atoms of the same Cluster Property. Thus we mark clusters with a unique Cluster ID when we need to explicitly track groups of atoms that belong to the same cluster. Figure 4 gives an example of a group of atoms joining an existing cluster with a certain Cluster ID and Cluster Property. The set of rules we have constructed gives us the ability to define clusters of atoms of a wide range of possible atomic structure. Figure 5 shows some examples of the possible structures we can obtain with our set of rules for growing clusters of atoms.

Definition 4 (Cluster ID)**.** A Cluster ID identifies groups of atoms belonging to the same cluster.

Since lower number Cluster Minsizes might match part of the configurations from higher number Cluster Minsizes, we need to add one more definition to uniquely identify the Cluster Properties of clusters in our simulation. We need to assign different Cluster Priorities to the different Cluster Properties to uniquely identify the clusters within our simulation. Figure 6 gives an example of how this definition can uniquely identify Cluster Properties of clusters of atoms within the simulation based on their Cluster Priorities. Note here that the possible Cluster Properties in the simulation needs to be chosen such that at any time in the simulation, each atom can be uniquely associated with a cluster.



(a)

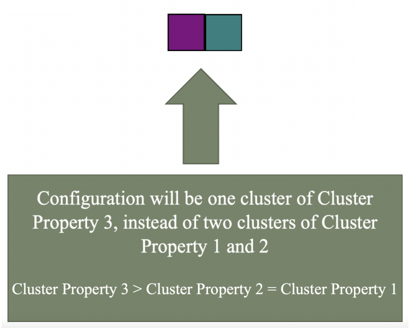
(b)

Figure 5. (a) depicts a cluster of two atomic species with only 2 local configurations out of the possible 32 (i.e. 2^5) local configurations in Cluster Minsize-5. (b) depicts a cluster of three atomic species with 25 local configurations out of the possible 243 (i.e. 3^5) local configurations in Cluster Minsize-5. The local configuration counts were ignored for the atoms on the boundary of the figure.

Definition 5 (Cluster Priority)**.** Cluster Priority is a priority assigned to different Cluster Properties. This is done in order to uniquely determine the Cluster Properties of clusters in the simulation in cases where different Cluster Properties might share overlapping configurations.

Cluster Property	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1	Minsize-1	1	🟦
Cluster Property 2	Minsize-1	1	🟪
Cluster Property 3	Minsize-2	2	🟪🟦

(a) Cluster Properties



Configuration will be one cluster of Cluster Property 3, instead of two clusters of Cluster Property 1 and 2

Cluster Property 3 > Cluster Property 2 = Cluster Property 1

(b) Example Cluster Property Assignment

Figure 6. An example of Cluster Property assignment based on Cluster Priority.

3. Identifying Low-temperature Cluster Properties

We present here a divide-and-conquer scheme to search for the low-temperature Cluster Properties of the simulation. The approach is to start the simulation with finding high cluster entropy Cluster Properties that do not mix, and then refine the structure with lower cluster entropy Cluster Properties. In order to do this, we need to break up a Cluster Property into its lower cluster entropy constituents. We accomplish this by creating Cluster Properties that are composed of other Cluster Properties, called Children Cluster Properties. Children Cluster Properties are thus used to create Parent Cluster Properties (Table. 1). The natural way to partition a Cluster Property into smaller constituents or new lower cluster entropy Cluster Properties then, is to

Parent Cluster Property **Cluster Property 1** Cluster Priority: 3

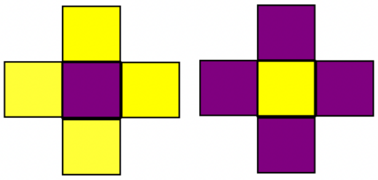
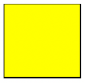
Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.1	Minsize-5	3.2	
Cluster Property 1.2	Minsize-1	3.1	
Cluster Property 1.3	Minsize-1	3.1	

Table 1. Depiction of a Parent Cluster Property composed of Children Cluster Properties. Each Child is assigned a different Cluster Priority

use a subset of its Children to create new Cluster Properties. Thus, the number of ways to generate m unique Cluster Properties from a Parent Cluster Property with N children, is equal to the number of ways to distribute N distinguishable objects into m indistinguishable boxes, such that every box is occupied by at least one object. We call each unique distribution of the N objects or Cluster Properties into m sets, a partition. The number of possible partitions of a Parent Cluster Property with N children into m Cluster Properties is thus

$$\frac{1}{m!}\sum_{k=0}^{m-1}(-1)^k\binom{m}{k}(m-k)^N \tag{3.1}$$

Although the number of possible partitions exponentially grows as a function of the number of children N, we can use certain heuristics to limit the amount of partitions to analyze. For example, we can set the value of m to reduce the number of partitions and restrict ourselves to only partitions that equally distribute the N children into the m sets. If possible, we can also use information about the sign of the coupling constant between pairs of atoms to reduce the amount of children we assign to the Parent Cluster Property.

We can go about finding the low-temperature Cluster Properties of the simulation by hypothesizing a probability function for drawing samples that is optimized for the growth of clusters with the defined Cluster Properties in the simulation. By picking such a function and defining a hierarchical structure for our Cluster Properties, we expect to obtain phase separated or non-mixing clusters of high cluster entropy Cluster Properties before lower cluster entropy Cluster Properties. We start off by first defining a Parent Cluster Property that treats the entire simulation cell as one cluster. We then iteratively partition the Parent Cluster Property into smaller constituents and only save the partition that gives the simulation the lowest energy or gives the clusters in the simulation the lowest degree of mixing. This results in a tree of Cluster Properties where the root node is the Parent Cluster Property that results in the entire simulation cell being one cluster and the leaf nodes are the low-temperature Cluster Properties of the simulation. Algorithm 1 details the steps in which this tree can be constructed.

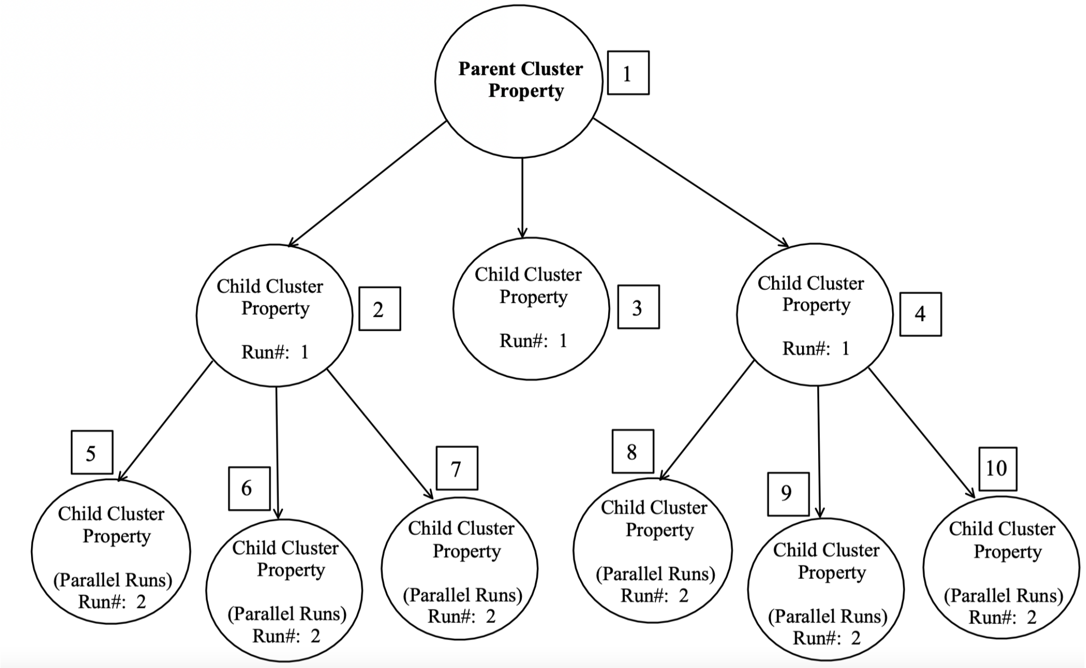
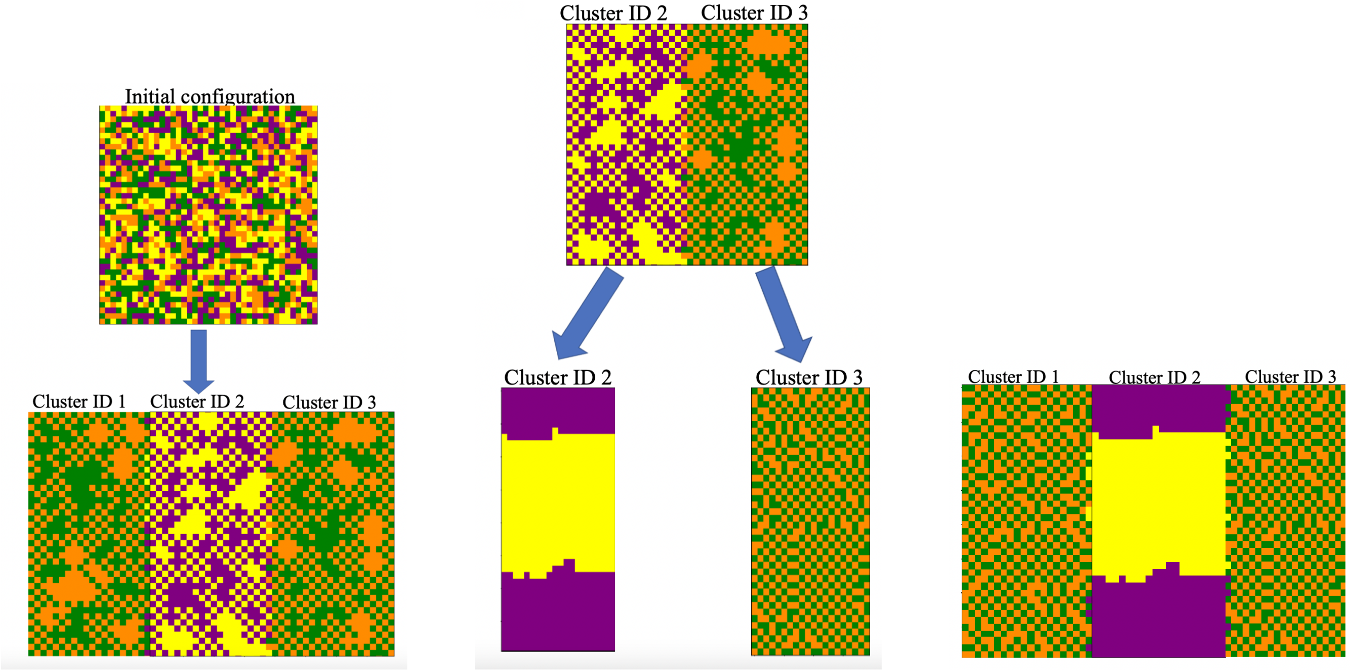


Figure 7. Depiction of a possible Cluster Property tree, where the leaf nodes are the low-temperature Cluster Properties. Only the partitions of the Parent Cluster Property that resulted in the lowest energy in the simulation are saved in each level of the tree. The different levels corresponds to different runs in the simulation. The numbers next to the nodes are the Cluster Property identifiers. The Allowable Sub-Clusters corresponding to each Cluster Property are depicted in Appendix B corresponding to the runs in Figure 8.

Algorithm 1: Steps to find the Cluster Property tree

```
1 Define the Parent Cluster Property of the simulation cell, such that the entire simulation cell is treated as one cluster. This cluster property should be composed of children cluster properties. This is always the root node in the Cluster Property tree.
2 Define $m$, the desired number of cluster properties to use in the simulation.
3 Generate all possible ways to partition the parent cluster property into $m$ children
4 For each partition, run the New MC method at a low temperature, in parallel, using $m$ cluster properties, for a few iterations
5 Choose the partition that resulted in the lowest energy, and continue running the simulation corresponding to this partition until the fluctuation in the number of clusters and their sizes reaches steady state.
6 Save this partition, or the set of $m$ cluster properties that resulted in the lowest energy, into its corresponding place in the tree.
7 Define $n$, the current number of clusters in the simulation. Run $n$ parallel simulations, restricting the simulation cell to only the lattice positions of the clusters for each run. Repeat steps 2-6 for each parallel run treating the corresponding cluster property of the cluster as the Parent Cluster Property.
8 Repeat step 7 for all parallel runs until the simulation reaches equilibrium.
```

This Algorithm results in a tree similar to the one depicted in Figure 7, where the leaf nodes are the low-temperature Cluster Properties of the simulation, since they correspond to the atomic configurations with the lowest energy. This tree only needs to be computed once. Once we have the tree, we can use it to simulate the system at various temperatures by using Algorithm 1 with steps 3, 4, and 6 omitted and using the partitions as defined by the tree. Figures 7 and 8 depict an example by which a Cluster Property tree can be used to simulate the equilibrium structure of an atomic system. At higher simulation temperatures, it might be sufficient to only use or stop at the higher cluster entropy Cluster Properties which are in the lower levels of the



(a) First calculation

(b) Examples from the parallel runs

(c) Final Configuration

Figure 8. An example by which the Cluster Property tree shown in Figure 7 can be used to simulate the equilibrium structure of an atomic system. In the first run, the Parent Cluster Property of the entire simulation cell is partitioned into separate Cluster Properties. This results in phase-separated clusters with high cluster entropies. In the parallel runs, the Cluster Properties corresponding to the clusters are further partitioned and the process is repeated until equilibrium is reached.

tree. If we wish to reduce the number of possible partitions generated from the Parent Cluster Property, we can initially run a separate simulation using Cluster Minsize-1 for each child in the Parent Cluster Property, such that the number of children in the Parent Cluster Property is equal to the number of atomic species; this is done to determine which subsets of atomic species tend to cluster together at low temperatures. We can use this information to design our Parent Cluster Property in subsequent simulations to have fewer children and greater Cluster Minsize numbers. This technique implicitly develops a natural way to break up the Boltzmann distribution into separable computational components. The hierarchical structure we have developed is generally valid in the absence of long-range interaction.

4. Swap-types, Expansion Probabilities, and the Proposed MC Method

The proposed method partitions the possible MC trial moves into three different categories, each with different probabilities. The possible moves in the MC simulation are classified as an exchange of atoms between pairs of boundary-boundary (bb), boundary-interior (bi), and interior-interior (ii) atoms of two different species and Cluster Properties. Here, boundary-boundary refers to an exchange between two boundary atoms of different clusters, and boundary-interior refers to an exchange of atoms between a boundary atom and an interior atom, and interior-interior refers to an exchange of atoms between two interior atoms (Fig. 9). An interior atom is defined to be an atom whose nearest-neighbor atoms all belong to the same cluster. We define the swap-type of a pair of atoms to be an indicator for one of the possible three pairs (i.e., a bb, bi, or ii pair). The goal of this new approach is to propose sample moves according to their likelihood of being accepted at the given temperature.

For example, at low temperatures, boundary-boundary swaps between clusters of atoms are likely to be favored over interior-interior swaps because the former can reduce the boundary energy and entropy is less important at low temperatures. The rate of cluster growth–and, as a result, simulation efficiency–can be optimized by proposing boundary-boundary swaps at a higher probability. Furthermore, we can introduce large changes in configuration space at a single MC step by forming and swapping pairs of clusters. In this case, the two clusters will be initialized from an initial pair of atoms, and expanded concurrently. Neighboring atoms are added to each cluster according to a Bernoulli distribution. For conciseness, we refer to the probability of proposing the initial pair of atoms of being a bb, bi, or ii pair as the swap-type probability of the initial pair. Similarly, the probability of adding atoms to a cluster is referred to as the expansion probability. Since we must always propose an initial pair of atoms to swap at each iteration, we have to enforce the following condition:

$$p^I_{bb} + p^I_{bi} + p^I_{ii} + p^P_{bb} = 1 \tag{4.1}$$

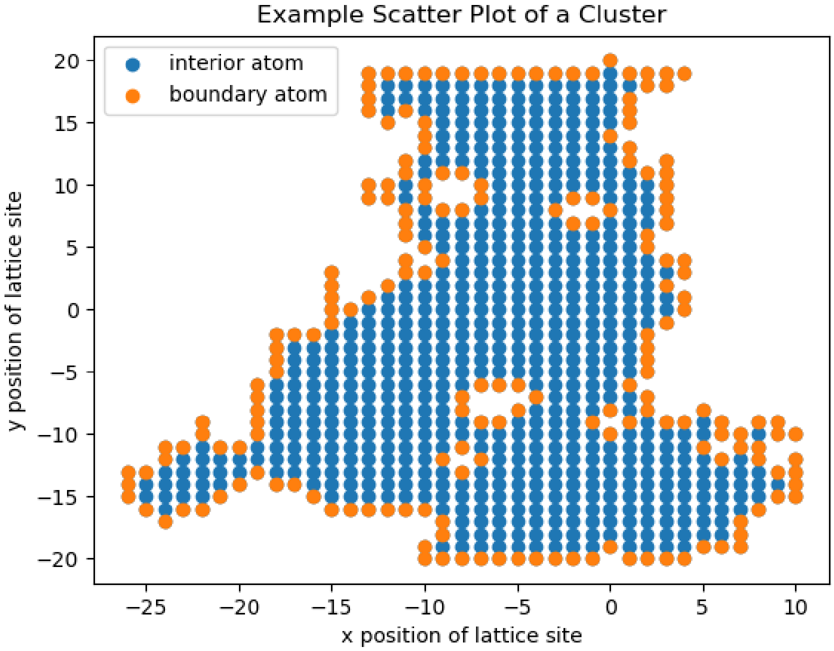


Figure 9. An example with different boundary and interior atoms of a cluster with a Cluster Property that has only one local configuration in its **AS** of Cluster Minsize-1. All the atoms are of the same species.

In the above expression p^I_{bb} is the probability that the proposed initial pair is a bb pair, p^I_{bi} is the probability that the proposed initial pair is a bi pair, and p^I_{ii} is the probability that the proposed initial pair is an ii pair. p^P_{bb} is only non-zero for parallel runs and is needed to avoid the use of periodic boundary conditions in the parallel runs. The set of boundary atoms on the outer boundary of a cluster (i.e. of different Cluster Properties) are included in the parallel runs, but they are only allowed to be exchanged within this set and only within their respective Cluster Properties. Thus p^P_{bb} is the probability of proposing an exchange within this set. Since the boundary atoms on the outer boundary of a cluster are only exchanged within their Cluster Properties, they are not expanded to more than a pair swap in one iteration. Note here that the configurations of these set of atoms (i.e., on the outer boundary of a cluster) are ignored when constructing the final configuration of the simulation cell, since they would have been accounted for from other parallel runs. The expansion probabilities, on the other hand, follows a Bernoulli distribution, such that if the probability of adding a neighboring pair is p then the probability of not adding it is $1 - p$. The neighboring atoms in the neighbor pair are only added with some probability p if each atom in the pair belongs to the same clusters as the atoms in the initial pair; otherwise it is zero. In this new MC scheme, we use different expansion probabilities based on the swap-type of the proposed initial pair; thus the probability of adding a neighboring pair to a swap labeled by $n = 1$ (swap pair) and $n = 0$ (don't swap pair) based on the swap-type of the initial pair is

$$P^e_{bb} = \begin{cases} p^e_{bb} \text{ for } n = 1 \\ 1 - p^e_{bb} \text{ for } n = 0 \end{cases} \tag{4.2}$$

$$P^e_{bi} = \begin{cases} p^e_{bi} \text{ for } n = 1 \\ 1 - p^e_{bi} \text{ for } n = 0 \end{cases} \tag{4.3}$$

$$P^e_{ii} = \begin{cases} p^e_{ii} \text{ for } n = 1 \\ 1 - p^e_{ii} \text{ for } n = 0 \end{cases} \tag{4.4}$$

In the above expressions P^e_{bb}, P^e_{bi}, P^e_{ii} are the probability of adding a neighboring pair based on the swap-type of the initial pair, and p^e_{bb}, p^e_{bi}, p^e_{ii} are the probabilities of accepting the neighboring pair to swap (i.e., the expansion probabilities). To reiterate, the atoms in the neighboring pair must belong to the same pair of clusters as the atoms in the initial pair for the move to be even considered.

5. Simulation Results

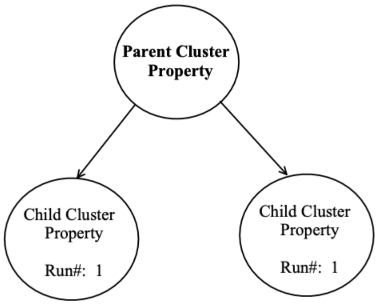
We now focus on an implementation of the technique using a 2D Ising model binary alloy system. We make use of the classical Metropolis-Hastings-Rosenbluth algorithm in order to obey the detailed balance condition [Tuc10]. The form of the acceptance probability for a proposed MC move to go from state x to state x' is given by:

$$A(x \to x') = \min\left(1, \frac{P(x')Q(x' \to x)}{P(x)Q(x \to x')}\right) \tag{5.1}$$

Here P is the desired Boltzmann distribution and Q is the *a priori* probability or the probability of drawing the move in the simulation. We will denote the probability of the forward move, $Q(x \to x')$, as the forward probability, and $Q(x' \to x)$ as the reverse probability.

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.1	Minsize-1	1	🟪
Cluster Property 1.2	Minsize-1	1	🟨

(a) Parent Cluster Property



(b) Cluster Property Tree

Figure 10. (a) shows the Parent Cluster Property that treats the entire simulation cell as one cluster. This Cluster Property was designed for an Ising Model system with $J > 0$. The different colors indicate different spin orientations. (b) shows the corresponding Cluster Property tree used for this system.

Since we are using the Ising Model for this example, we immediately know the low-temperature Cluster Properties of the system. We use a positive coupling constant (i.e., $J > 0$) between the atoms/spins in our system and design the Parent Cluster Property that treats the entire simulation cell as one cluster according to Figure 10a. Our method requires two or more Cluster Properties to start off the simulation, so we need to partition this Parent Cluster Property into different Cluster Properties. Since there are only two Children, there is only one way to partition this Parent Cluster Property into separate Cluster Properties. We thus partition the Parent Cluster Property into two Cluster Minsize-1 Cluster Properties resulting in the Cluster Property tree in Figure 10b.

In this example, a 40×40 cell size was used with each lattice site initialized with probability 0.5 between the possible two atomic species. We optimize the code by only tracking the boundary and interior atoms of different Cluster Properties at each iteration, instead of groups of atoms of a common cluster ID. Since the Cluster Property tree only has two leaf nodes, we do not perform any parallel runs in this example (Fig. 10b). Note here that if we had parallel runs, we would need to compute the group of atoms of common Cluster IDs before executing the parallel runs.

Parameters for $\beta J = 0.1$ (Hot)	**Case 1**	**Case 2**	**Case 3**
p^0_{bb}	0.55	0.55	0.55
p^0_{bi}	0.31	0.31	0.31
p^0_{ii}	0.14	0.14	0.14
p^e_{bb}	0.00	0.26	0.26
p^e_{bi}	0.00	0.30	0.30
p^e_{ii}	0.00	0.30	0.30
c_P	N/A	False	True
Time Complexity per iteration	$O(1)$	$O(m^3)$	$O(m)$

Table 2. Parameters at $\beta J = 0.1$.

We use the following form in (5.2) to estimate the swap-type probabilities for the initial proposed pair:

$$\begin{aligned} p^P_{bb} &= 0 \\ p^I_{bb} &= \frac{p^0_{bb} n_{bb}}{p^0_{bb} n_{bb} + p^0_{bi} n_{bi} + p^0_{ii} n_{ii}} \\ p^I_{bi} &= \frac{p^0_{bi} n_{bi}}{p^0_{bb} n_{bb} + p^0_{bi} n_{bi} + p^0_{ii} n_{ii}} \\ p^I_{ii} &= \frac{p^0_{ii} n_{ii}}{p^0_{bb} n_{bb} + p^0_{bi} n_{bi} + p^0_{ii} n_{ii}} \end{aligned} \tag{5.2}$$

We set $p^P_{bb} = 0$ since we do not have any parallel runs in this simulation. We use this form so that the swap-type probabilities for the initial pair can be written in terms of constant variables in the simulation. p^0_{bb}, p^0_{bi}, and p^0_{ii} are constants in the simulation and are denoted as the base probabilities. n_{bb}, n_{bi}, and n_{ii} in the above equation are the number of pairs of swap-type bb, bi, and ii at each iteration of the simulation.

Note here that if the base probabilities are made equally likely (i.e. $1/3$ each) and the expansion probabilities are set to zero, our method is equivalent to the traditional MC scheme or the non-local Kawasaki Algorithm [LB21].

In order to estimate the best base probabilities to use in our simulation, we begin by plotting the probability that a swap is of a certain swap-type (i.e., (bb), (bi), or (ii)) given that the swap was accepted in the traditional MC simulation. The probabilities were plotted as a function of the interatomic potential between the atoms, J, and inverse temperature, β, normalized by the number of pairs of each swap-type that occurred during the entire simulation (Fig. 11). The base probabilities as a function of βJ were then estimated by fitting a 13-degree polynomial to the different curves in Figure 11. We highlight that while we have used the original MC scheme to construct Figure 11, we could have used any curve with the same trends without relying on the original MC method. While in principle we could have simply optimized for the swap-type probabilities at different temperatures without writing them as functions of the number of pairs of each swap-type (5.2), the optimization will not scale well for different multi-species systems. This is because the optimal probabilities can be sensitive to different proportions of species type in the simulation.

The overall procedure of our simulation is as follows:

1. Propose an initial pair of atoms to swap according to the swap-type probabilities for the initial pair (i.e. p^I_{bb}, p^I_{bi}, and p^I_{ii}). Note that the atoms in the pair should belong to different Cluster Properties and have different species.
2. Expand Swaps using the expansion probabilities and the swap-type of the initial pair (see Appendix A the pseudocode)
3. Compute the forward and reverse probabilities of the move
4. Compute the change in energy of the move
5. Accept/reject the move according to (5.1) to ensure that detailed balance is satisfied
6. Repeat these steps until equilibrium is reached

We have tested our method against the traditional MC scheme using three different approaches, or sets of parameters, and ran them at three different temperatures. In all our simulations, we used a dimensionless coupling constant of $J = 20$. Tables 2, 3, and 4 provide the values for the various probabilities for the different βJ values.

In Tables 2, 3, and 4 p^0_{bb}, p^0_{bi}, and p^0_{ii} are the base probabilities used to propose the initial pair of atoms to swap. p^e_{bb}, p^e_{bi}, and

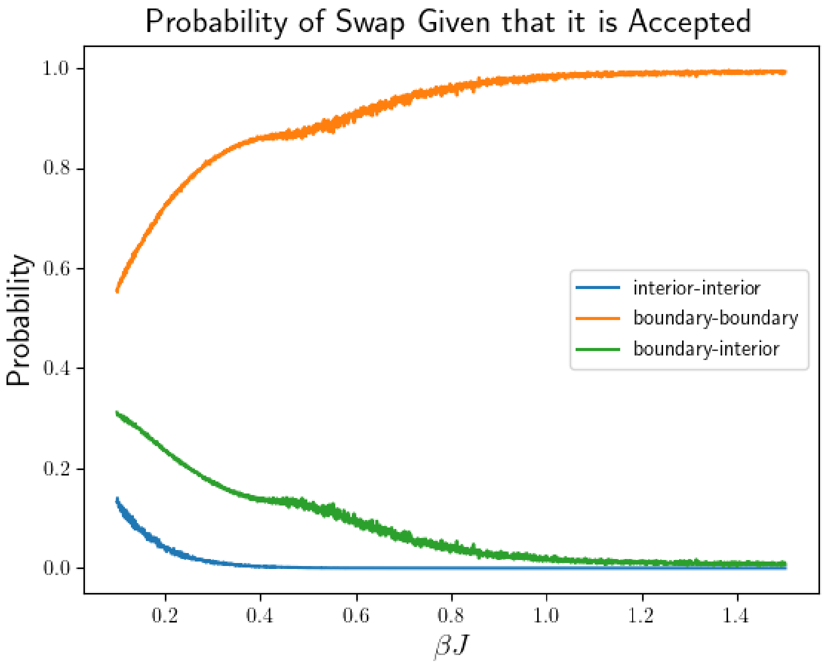


Figure 11. Probabilities of the different swap types as a function of inverse temperature β and atomic coupling constant J.

Parameters for $\beta J = 0.44$ (Transition Temperature)	**Case 1**	**Case 2**	**Case 3**
p^0_{bb}	0.86	0.86	0.86
p^0_{bi}	0.13	0.13	0.13
p^0_{ii}	0.01	0.01	0.01
p^e_{bb}	0.00	0.50	0.50
p^e_{bi}	0.00	0.50	0.50
p^e_{ii}	0.00	0.50	0.50
c_P	N/A	False	True
Time Complexity per iteration	$O(1)$	$O(m^3)$	$O(m)$

Table 3. Parameters at $\beta J = 0.44$.

p^e_{ii} are the different expansion probabilities associated with the swap-type of the initial pair. c_P is a boolean variable that is only valid if the expansion probabilities are greater than zero. If true, the forward and reverse probabilities are computed treating the proposed initial pair of atoms as the only possible initial pair from which the move can be made; if not, the probabilities of the move are computed with respect to all possible initial pairs of atoms between the pair of flipped clusters (i.e., the flipped spin values in Ising Model). m is the average size of one of the flipped clusters, or the average number of flipped spins of an atomic species per iteration.

Our method was compared with the traditional method by plotting the energy of the lattice of atoms against the number of iterations for each case (i.e., the various sets of parameters in the new method) in Figures 12, 13, and 14. To decide if the system had reached equilibrium, we performed a running average of the total energy; if the difference between two consecutive energy values in the running average was less than 3×10^{-4} for a thousand iterations, we concluded that our system had reached equilibrium. This equilibrium condition gave consistent equilibrium energies across the different runs we performed. All energies are reported in normalized units.

The proposed method does significantly better at low temperatures (i.e., high βJ), since the simulation tends to not resample/swap atoms on the interior of low temperature local configurations as much; thus the low temperature clusters phase separate faster. This is because entropy is relatively less important than energy at low temperatures, so boundary-boundary swaps are favored over the other types of swaps. This could also be understood from the fact that higher boundary-boundary swap sampling probabilities will be optimized for growing our defined Cluster Properties. Since our defined Cluster Properties are the low-temperature Cluster Properties (i.e the leaf nodes in the Cluster Property tree), we expect that higher boundary-boundary

Parameters for $\beta J = 1.5$ (Cold)	**Case 1**	**Case 2**	**Case 3**
p_{bb}^{0}	0.984	0.984	0.984
p_{bi}^{0}	0.013	0.013	0.013
p_{ii}^{0}	0.003	0.003	0.003
p_{bb}^{e}	0.000	0.131	0.131
p_{bi}^{e}	0.000	0.340	0.340
p_{ii}^{e}	0.000	0.010	0.010
c_P	N/A	False	True
Time Complexity per iteration	$O(1)$	$O(m^3)$	$O(m)$

Table 4. Parameters at $\beta J = 1.5$

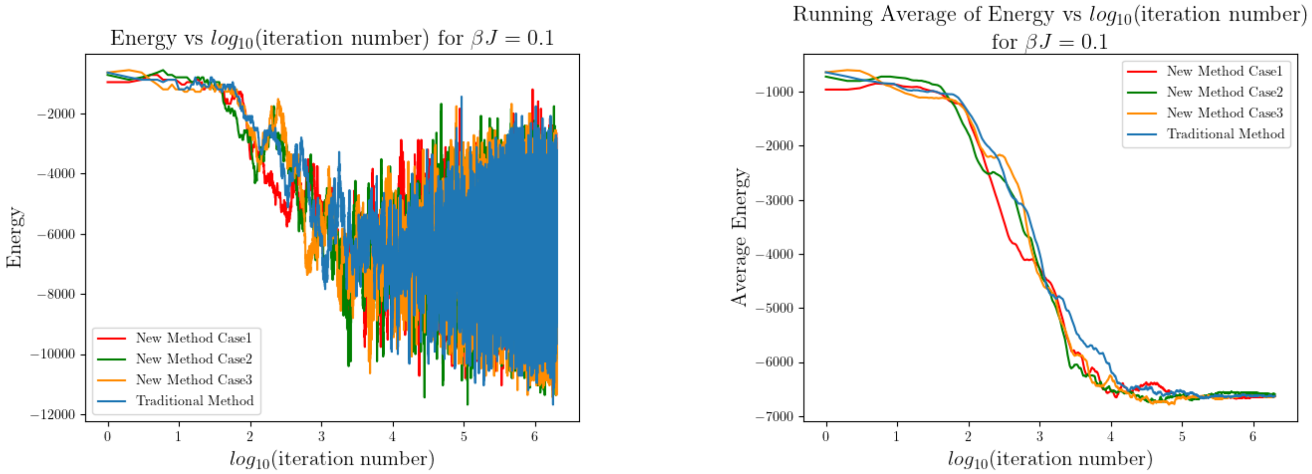
βJ	**Iterations to reach equilibrium:**	**Average Energy**	**Standard Deviation of Energy per Spin**	**Acceptance Rate**
0.1	84,020	-6.63k	0.720	0.8393
0.44	114,965	-42.0k	1.159	0.3390
1.5	753,218	-60.5k	0.073	0.0087

Table 5. Proposed Method Case 1 Simulation Results. The average and standard deviation of the energy are computed after the system reaches thermal equilibrium. The Acceptance Rate is the total number of accepted MC moves divided by the total number of iterations.

swap sampling probabilities will drive the system to equilibrium faster at lower temperatures. Since at high temperatures (i.e., low βJ), clusters of atoms have lower tendencies to phase separate, the performance of our method is comparable to the traditional method.

Overall, the higher acceptance rates and the lower iteration numbers seen in Tables 5, 6, and 7 compared to those of the traditional method in Table 8 gives us confidence that our method converges to equilibrium faster than the traditional one. The consistent values for the average energy and standard deviation of energy per spin between our method and the traditional method, and also from the different cases in our method, gives us confidence in our equilibrium results. The similar results that we see in Tables 6 and 7 suggest that computing the forward and reverse probabilities with $c_P = True$ is sufficient to meet the detailed balance condition, implying that this technique can be achieved with a time complexity of $O(m)$ per iteration.

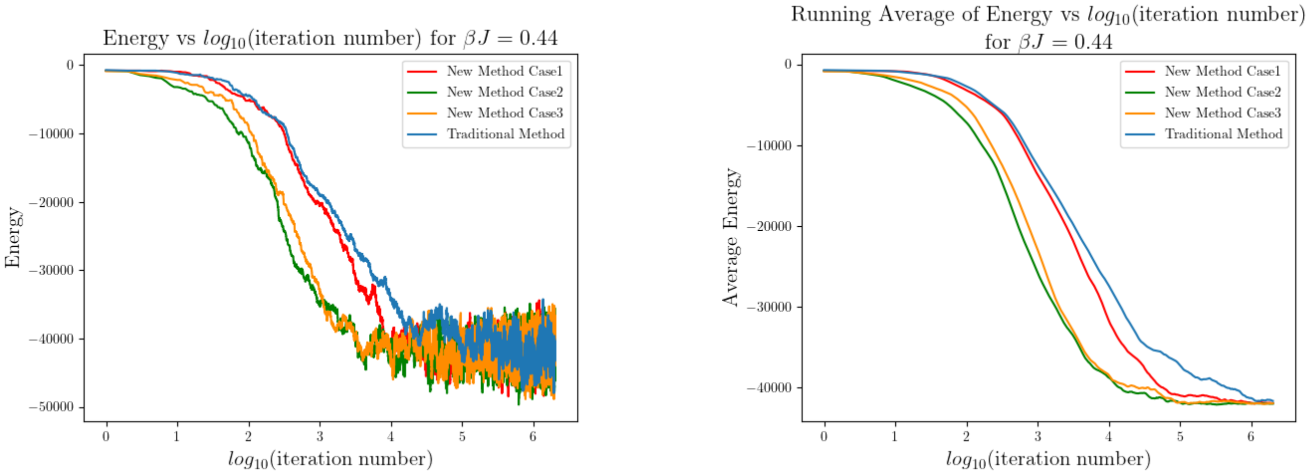
Although this binary alloy system could be easily generalized for any number of atomic species simply by iteratively running this technique with two species at a time and keeping all other species fixed, such an approach would not readily parallelize.



(a)

(b)

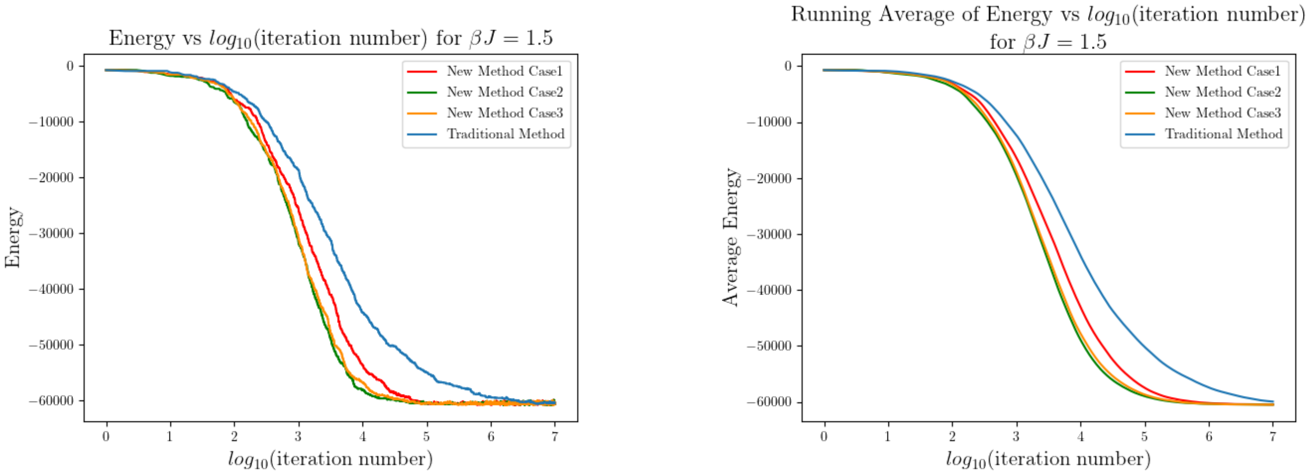
Figure 12. At $\beta J = 0.1$, (a) energy vs. log of iteration number, and (b) running average of total energy vs. log of iteration number.



(a)

(b)

Figure 13. At $\beta J = 0.44$, (a) energy vs. log of iteration number, and (b) running average of total energy vs. log of iteration number.



(a)

(b)

Figure 14. At $\beta J = 1.5$, (a) energy vs. log of iteration number, and (b) running average of total energy vs. log of iteration number.

6. Conclusion

We have developed a fast MC Scheme that conserves the concentration of different particle types. This scheme achieves the goal of making large and informed jumps in configuration space without reducing the MC acceptance rates, and presents the possibility for a divide-and-conquer parallel approach. The significant speedup can enable the understanding of new materials with complex atomic arrangements, and provides a new tool to aid in the discovery of HEA that have unique mechanical,

βJ	**Iterations to reach equilibrium:**	**Average Energy**	**Standard Deviation of Energy per Spin**	**Acceptance Rate**
0.1	76,804	-6.60k	0.726	0.8762
0.44	139,917	-42.0k	1.140	0.3606
1.5	516,942	-60.5k	0.067	0.0070

Table 6. Proposed Method Case 2 Simulation Results. The average and standard deviation of the energy are computed after the system reaches thermal equilibrium. The Acceptance Rate is the total number of accepted MC moves divided by the total number of iterations.

βJ	Iterations to reach equilibrium:	Average Energy	Standard Deviation of Energy per Spin	Acceptance Rate
0.1	97,368	-6.64k	0.734	0.8456
0.44	111,939	-42.0k	1.150	0.3128
1.5	486,220	-60.5k	0.069	0.0069

Table 7. Proposed Method Case 3 Simulation Results. The average and standard deviation of the energy are computed after the system reaches thermal equilibrium. The Acceptance Rate is the total number of accepted MC moves divided by the total number of iterations.

βJ	Equilibrium Reached at Iteration #:	Average Energy	Standard Deviation of Energy per Spin	Acceptance Rate
0.1	105,897	-6.63k	0.719	0.7819
0.44	212,109	-42.0k	1.215	0.1111
1.5	3,472,401	-60.4k	0.070	0.0007

Table 8. Traditional Method Simulation Results. The average and standard deviation of the energy are computed after the system reaches thermal equilibrium. The Acceptance Rate is the total number of accepted MC moves divided by the total number of iterations.

chemical, and electrical properties. The method is not limited to HEA, but it can also be used for the characterization of other systems in which interface effects are important.

Our demonstration of the proof-of-principle has been restricted to binary systems. An important next step, which is straightforward in principle, is to apply and demonstrate this method in the context of realistic multi-species HEA systems. Another avenue for future research is to tune the base and expansion probabilities as a function of the dynamical variables within the simulation, such as number of clusters, average cluster size, and the distribution of cluster size per iteration. Data-driven methods, such as deep reinforcement learning models, are a potential strategy to achieve this: the base and expansion probabilities will be the policy we are learning, and the states are the dynamical variables within the simulation. The goal of this reinforcement learning model would be to maximize for the acceptance probability, our reward function. The acceptance probabilities in this case will only use the accepted moves for which there is a non-zero change in the energy.

The method in this paper can be extended to settings in which the entropic contribution is comparable to the energetic contribution. By using Cluster Properties that are on lower levels in the Cluster Property tree as opposed to the ones on the leaf nodes (Fig. 7); we balance between aggregating atoms in clusters that are energetically favorable while still allowing them to take on more complex geometric configurations that have higher entropy.

Software Availability

A version of the code developed for this work is available at
`https://github.com/azizfall/Local_Structure_MC`

Data Availability

The data of this study will be made available on request.

Acknowledgments

We thank Markus Eisenbach (ORNL) and Yang Wang (PSC) for their insightful comments; AFOSR (MURI FA9550-18-1-0095) and the GEM fellowship to Aziz Fall for financial support; and NSF for XSEDE computing resources provided by Pittsburgh Supercomputing Center. Matthew Grasinger acknowledges the financial support of the Air Force Research Laboratory.

A. Expansion Algorithm

Algorithm 2: Expanding Swaps

```
Input: $x_1^0,\ x_2^0$, lattice positions of proposed initial pair of atoms to swap (Note:  The two positions should have
       opposite spin orientations or different atomic species)
Input: $p_{bb}^e,\ p_{bi}^e,\ p_{ii}^e$, expansion probabilities
Output: $C^a$, Array of lattice positions of atoms that will undergo a change in species
// Array of lattice positions of atoms that will undergo a change in specie or spin flip
$C^a \leftarrow [\,]$
// Append the lattice positions of the initial pair of atoms into $C^a$
$C^a.append(x_1^0)$
$C^a.append(x_2^0)$

$p^e \leftarrow$ Set to either $p_{bb}^e$ , $p_{bi}^e$, or $p_{ii}^e$ based on swap-type of initial pair

// Initialize two empty arrays
$q_1 \leftarrow [\,]$
$q_2 \leftarrow [\,]$

// Append the lattice position of atom 1 in the initial pair into $q_1$
$q_1.append(x_1^0)$
// Append the lattice position of atom 2 in the initial pair into $q_2$
$q_2.append(x_2^0)$
```

```
while $q_1$ is not Empty do
    // Initialize two empty arrays
    $L^1_{nei} \leftarrow [\,]$
    $L^2_{nei} \leftarrow [\,]$

        For every lattice position in $q_1$: Add all its nearest neighbors that have the same
        atomic specie (or belong to the same cluster in case of more complicated cluster
        properties) into $L^1_{nei}$. Do not Add the position if it was ever previously added into $L^1_{nei}$
        for any iteration in the while loop. All lattice positions in $L^1_{nei}$ should be unique.

        Repeat the above step for every lattice position in $q_2$ while adding lattice positions
        into $L^2_{nei}$.

        Note: $L^1_{nei}$ and $L^2_{nei}$ must have different sets of atomic species, even in the case of more
        complicated cluster properties. If we want, we can also restrict the swap-types of the
        expanded pairs to be the same as that of the initial pair.

    // Empty both $q_1$ and $q_2$
    $q_1 \leftarrow [\,]$
    $q_2 \leftarrow [\,]$

    // Find the minimum size between $L^1_{nei}$ and $L^2_{nei}$,
    $m = \mathrm{min}(len(L^1_{nei}),\ len(L^2_{nei}))$

    // Flip a coin $m$ times and set $n$ as the number of successes, with the probability of a success
       being $p^e$
    $n = \mathrm{FlipCoin}(p^e,\ m)$

        Pick a random set of $n$ lattice positions in $L^1_{nei}$ and add it into both $q_1$ and $C^a$

        Pick a random set of $n$ lattice positions in $L^2_{nei}$ and add it into both $q_2$ and $C^a$
end while
```

B. Cluster Properties of Figure 7

[BKL75] Alfred B Bortz, Malvin H Kalos, and Joel L Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, 17(1):10–18, 1975.

[CWL+22] Shuai Chen, Tian Wang, Xiaoyan Li, Yuan Cheng, Gang Zhang, and Huajian Gao. Short-range ordering and its impact on thermodynamic property of high-entropy alloys. *Acta Materialia*, page 118201, 2022.

[DPAM19] Modupeola Dada, Patricia Popoola, Samson Adeosun, and Ntombi Mathe. High entropy alloys for aerospace applications. In *Aerodynamics*. IntechOpen, 2019.

[FDG+20] Alberto Ferrari, Biswanath Dutta, Konstantin Gubaev, Yuji Ikeda, Prashanth Srinivasan, Blazej Grabowski, and Fritz Körmann. Frontiers in atomistic simulations of high entropy alloys. *Journal of Applied Physics*, 128:150901, 2020.

[GRR19] Easo P George, Dierk Raabe, and Robert O Ritchie. High-entropy alloys. *Nature reviews materials*, 4(8):515–534, 2019.

[GV19] Victor Geanta and Ionelia Voiculescu. Characterization and testing of high-entropy alloys from alcrfeconi system for military applications. In *Engineering Steels and High Entropy-Alloys*. IntechOpen, 2019.

[HR04] Alexander K Hartmann and Heiko Rieger. New optimization algorithms in physics. 2004.

[Kad66] Leo P Kadanoff. Scaling laws for ising models near t c. *Physics Physique Fizika*, 2(6):263, 1966.

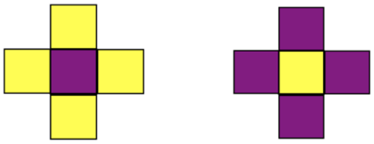
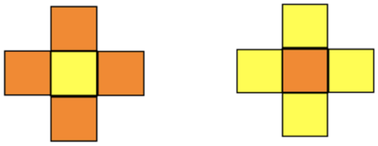
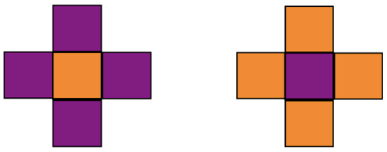
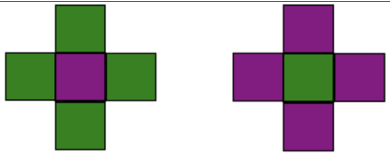
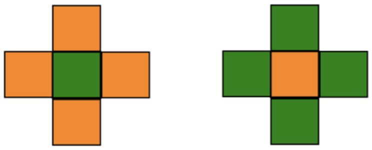
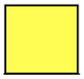
[Kat09] Helmut G Katzgraber. Introduction to monte carlo methods. *arXiv preprint arXiv:0905.1629*, 2009.

[KD91] Daniel Kandel and Eytan Domany. General cluster monte carlo dynamics. *Physical Review B*, 43(10):8539, 1991.

[KRN+20] Tatiana Kostiuchenko, Andrei V Ruban, Jörg Neugebauer, Alexander Shapeev, and Fritz Körmann. Short-range order in face-centered cubic vconi alloys. *Physical Review Materials*, 4(11):113802, 2020.

[LB21] David Landau and Kurt Binder. *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press, 2021.

Parent Cluster Property **Cluster Property 1** Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.10	Minsize-5	3.3	
Cluster Property 1.11	Minsize-5	3.3	
Cluster Property 1.12	Minsize-5	3.3	
Cluster Property 1.13	Minsize-5	3.3	
Cluster Property 1.14	Minsize-5	3.3	
Cluster Property 1.15	Minsize-5	3.3	
Cluster Property 1.16	Minsize-1	3.2	
Cluster Property 1.17	Minsize-1	3.2	
Cluster Property 1.18	Minsize-1	3.2	
Cluster Property 1.19	Minsize-1	3.2	

[Lui06] E Luijten. Introduction to cluster monte carlo algorithms. In *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, pages 13–38. Springer, 2006.

[LZY+21] Xianglin Liu, Jiaxin Zhang, Junqi Yin, Sirui Bi, Markus Eisenbach, and Yang Wang. Monte carlo simulation of order-disorder transition in refractory high entropy alloys: A data-driven approach. *Computational Materials Science*, 187:110135, 2021.

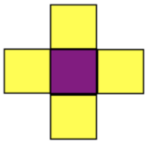
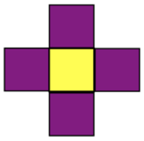
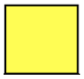
[Mur01] KPN Murthy. An introduction to monte carlo simulation of statistical physics problem. *arXiv preprint cond-mat/0104167*, 2001.

[Sch08] Tim P Schulze. Efficient kinetic monte carlo simulation. *Journal of Computational Physics*, 227(4):2455–2462, 2008.

[Sha17] Alexander Shapeev. Accurate representation of formation energies of crystalline alloys with many components. *Computational Materials Science*, 139:26–30, 2017.

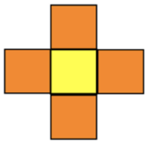
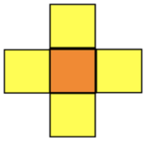
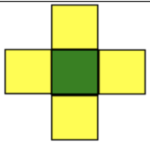
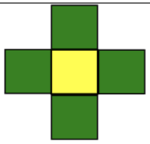
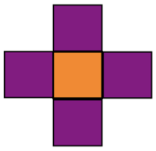
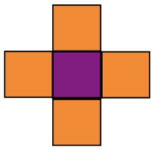
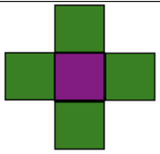
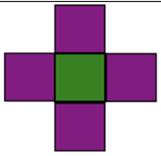
Cluster Property 2

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.10	Minsize-5	3.3	 
Cluster Property 1.16	Minsize-1	3.2	
Cluster Property 1.17	Minsize-1	3.2	

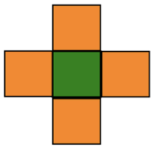
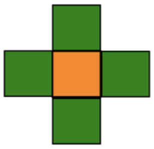
Cluster Property 3

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.11	Minsize-5	3.3	 
Cluster Property 1.12	Minsize-5	3.3	 
Cluster Property 1.13	Minsize-5	3.3	 
Cluster Property 1.14	Minsize-5	3.3	 

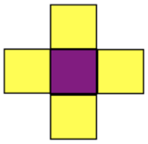
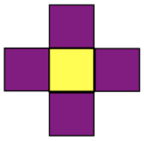
Cluster Property 4

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.15	Minsize-5	3.3	 
Cluster Property 1.18	Minsize-1	3.2	
Cluster Property 1.19	Minsize-1	3.2	

Cluster Property 5

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.10	Minsize-5	3.3	 

Cluster Property 6

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.16	Minsize-1	3.2	

[SW86] Robert H Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Physical review letters*, 57(21):2607, 1986.

[SW87] Robert H Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in monte carlo simulations. *Physical review letters*, 58(2):86, 1987.

[Swe79] Robert H Swendsen. Monte carlo renormalization group. *Physical Review Letters*, 42(14):859, 1979.

[TCWSNM$^+$17] Isaac Toda-Caraballo, Jan S Wróbel S., Duc Nguyen-Manh, Pablo Pérez, and Pedro E. J. Rivera-Díaz-del Castillo. Simulation and modeling in high entropy alloys. *JOM*, 69:2137–2149, 2017.

[Tuc10] Mark Tuckerman. *Statistical mechanics: theory and molecular simulation*. Oxford university press, 2010.

[Vot07] Arthur F Voter. Introduction to the kinetic monte carlo method. In *Radiation effects in solids*, pages 1–23. Springer, 2007.

[Wil75] Kenneth G Wilson. The renormalization group: Critical phenomena and the kondo problem. *Reviews of modern physics*, 47(4):773, 1975.

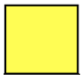
[WZY$^+$21] Yuan Wu, Fei Zhang, Xiaoyuan Yuan, Hailong Huang, Xiaocan Wen, Yihan Wang, Mengyuan Zhang, Honghui Wu, Xiongjun Liu, Hui Wang, Suihe Jiang, and Lu Zhaoping. Short-range ordering and its effects on mechanical properties of high-entropy alloys. *Journal of Materials Science & Technology*, 62:214–220, 2021.

[YZ20] Xuehui Yan and Yong Zhang. Functional properties and promising applications of high entropy alloys. *Scripta Materialia*, 187:188–193, 2020.

[ZYL$^+$18] Xuan Zhang, Jiahao Yao, Bin Liu, Jun Yan, Lei Lu, Yi Li, Huajian Gao, and Xiaoyan Li. Three-dimensional high-entropy alloy–polymer composite nanolattices that overcome the strength–recoverability trade-off. *Nano letters*, 18(7):4247–4256, 2018.

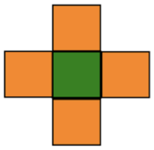
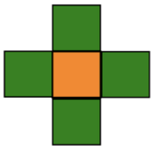
Cluster Property 7

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.17	Minsize-1	3.2	

Cluster Property 8

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.15	Minsize-5	3.3	 

Cluster Property 9

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.18	Minsize-1	3.2	

Cluster Property 10

Cluster Priority: 3

Children	Cluster Minsize	Cluster Priority	Allowable Sub-Clusters
Cluster Property 1.19	Minsize-1	3.2