Identifying the minimum-energy atomic configuration on a lattice: Lamarckian twist on Darwinian evolution

Mayeul d'Avezac* and Alex Zunger†
National Renewable Energy Laboratory, Golden, Colorado 80401, USA
(Received 11 September 2007; revised manuscript received 24 April 2008; published 4 August 2008)

We examine how the two different mechanisms proposed historically for biological evolution compare for the determination of crystal structures from random initial lattice configurations. The Darwinian theory of evolution contends that the genetic makeup inherited at birth is the one passed on during mating to new offspring, in which case evolution is a product of environmental pressure and chance. In addition to this mechanism, Lamarck surmised that individuals can also pass on traits *acquired* during their lifetime. Here we show that the minimum-energy configurations of a binary $A_{1-x}B_x$ alloy in the full $0 \leq x \leq 1$ concentration range can be found much faster if the conventional Darwinian genetic progression—mating configurations and letting the lowest-energy (fittest) offspring survive—is allowed to experience Lamarckian-style fitness improvements during its lifetime. Such improvements consist of $A \leftrightarrow B$ transmutations of some atomic sites (not just atomic relaxations) guided by "virtual-atom" energy gradients. This hybrid evolution is shown to provide an efficient solution to a generalized Ising Hamiltonian, illustrated here by finding the ground states of face-centered-cubic $Au_{1-x}Pd_x$ using a cluster-expansion functional fitted to first-principles total energies. The statistical rate of success of the search strategies and their practical applicability are rigorously documented in terms of average number of evaluations required to find the solution out of 400 independent evolutionary runs with different random seeds. We show that all *exact* ground states of a 12-atom supercell (2^{12} configurations) can be found within 330 total-energy evaluations, whereas a 36-atom supercell (2^{36} configurations) requires on average 39 000 evaluations. Thus, this problem cannot be currently addressed with confidence using costly energy functionals [e.g., density-functional theory (DFT) based] unless it is limited to $\leq$20 atoms. The computational cost can be reduced at the expense of accuracy: Searching for all *approximate*-minimum-energy configurations (within 3 meV) of a 12- or 36-atom supercell requires on average 30 or 580 total-energy evaluations, respectively. Thus it could be addressed even by costly energy functionals such as density-functional theory.

DOI: 10.1103/PhysRevB.78.064102 PACS number(s): 02.30.Zz, 31.15.−p, 71.15.−m

I. INTRODUCTION

At the heart of solid-state physics and structural inorganic chemistry is the form/function relationship between crystal structure and crystal properties.[1–3] This recognition has led to continued efforts in measuring and cataloging crystal structures[4] and, more recently, to systematic efforts in the theoretical prediction of crystal structures, either from inductive "Pauling-esque" approaches[2,3] or from explicit quantum-mechanical total-energy minimization approaches.[5] There are generally two sets of structural degrees of freedom that need be determined in crystals—the unit-cell lattice structure and the decoration of atomic sites by different atomic types. Historically, a concentric hierarchy of three *crystal*-structure-optimization problems can be distinguished according to the type and amount of information assumed at the outset regarding these degrees of freedom. In the first class of problems [hereafter referred to as type (i)], the system's lattice structure (e.g., fcc) and the decoration of the lattice sites by different atomic types are assumed. Left to optimize are cell-internal degrees of freedom not specified by the space group and the cell-external degrees of freedom, such as volume or the tetragonal $\frac{c}{a}$ ratio.[6–8] In the second type of problems [hereafter referred to as type (ii)], one attempts to populate the N sites of a given skeletal structure—a molecular backbone in chemistry,[9,10] or a Bravais lattice in solid-state physics[11–15]—with M different substituents, e.g., chemical groups or atomic species. This search of the space of M^N possible configurations $\{\sigma\}$ for the minimum-energy structure of an $A_{1-x}B_x$ system on a lattice represents the classic Ising problem[16–18] cast more recently into a cluster-expansion form[13,15] where the interaction energies are determined from first principles. In the third and most ambitious type of search problem [hereafter referred to as type (iii)], both lattice vectors and configurations are determined through energy minimization. Initial progress in this last type of problem was reported recently for both finite clusters[19,20] and infinite periodic solids.[21–24]

Here we discuss type (ii) problems. In historic Ising-Hamiltonian problems, the complexity of the configuration-search space was limited by *a priori* assumptions on the nature and range of the underlying interactions (e.g., first and second neighbors only,[12,25–27] such that the ensuing ground-state configurations were characterized by small unit cells with $N \leq 8$). However, unbiased first-principles determination of the interactions[28–32] revealed subsequently that many binary metals and semiconductor alloys are characterized by nontrivial pair interactions (typically extending to $\sim$10 neighbors), rendering the search space quite complex and requiring new ground-state search methods. Furthermore, since the search space grows exponentially with the number of lattice sites considered, N, obtaining the minimum-energy configurations through exhaustive enumeration[14] is difficult for $N \gg 20$. One has to resort to so-called sampling methods which investigate only a fraction of the full configuration space. Even so, searching for the *exact* configurations with

minimum energy of three-dimensional Ising models[33] is nondeterministic polynomial-time (NP) hard; i.e., the number of trial structures of which the energy is evaluated during the search grows exponentially with the number of atoms, N, in the system.[34] The key question about any such search procedure is therefore how many evaluations of the total energy are needed to obtain the correct minimum-energy configurations with a given degree of confidence.

The number of configurations for which the total energy must be evaluated in order to identify the lowest-energy configurations with a given degree of confidence depends on the extent to which a given material system is dominated by the NP-hard combinatorial issue of site decorations, or by simpler noncombinatorial factors. For example, from a complexity-theory point of view, it is not clear whether lattice-type and unit-cell optimizations are NP hard.[35,36] In metal alloys whose constituents are both nearly isovalent and have the same lattice type [e.g., fcc $Au_{1-x}Pd_x$,[28] fcc $Au_{1-x}Cu_x$,[32] or bcc $Mo_{1-x}Ta_x$ (Ref. 30)], alteration of the lattice type corresponds to easily resolvable high-energy excitations, whereas combinatorial $A \leftrightarrow B$ cross substitutions on the same lattice type [type (ii) optimization] cost little energy and thus correspond to difficult-to-resolve low-energy excitations. In contrast, cross substitutions of cations and anions in valence compounds[1,2] (e.g., As-on-Ga "antisite defect" in GaAs or anion/cation cross substitutions in Al_2O_3 and $MgSiO_3$) correspond to easily resolvable high-energy excitations, whereas alteration of the lattice type corresponds to a low-energy excitation.[8,37] As a result, determining the lattice type becomes the bottleneck for structure optimization of valence compounds. Previous applications of genetic algorithms to type (iii) structure optimization of TiO_2, Al_2O_3, $MgSiO_3$, and $(NH_2)_2CO$ have claimed[21,38] nearly 100% success at finding the correct lowest-energy structure within only ≤ 1000 total-energy evaluations for structures containing as many as $N=80$ atoms. Yet, a systematic and rigorous investigation[39] of type (ii) optimization for a metal-alloy system indicates that even for $N=40$, as many as ~ 3500 evaluations were needed using a genetic algorithm,[40] suggesting that $N=80$ systems cannot be computed in this way on present deterministic computers. Clearly, the challenge is to address and document the success rate of structural optimization problems in systems which are dominated by the NP-hard combinatorial issue of selecting site decorations.

We present efficient search methods for the determination of minimum-energy configurations. As an example, we focus on the face-centered-cubic $Au_{1-x}Pd_x$ alloy within the full composition range, $0 \leq x \leq 1$. This space corresponds to 2^N possible combinations. Instead of using a pure Darwinian genetic algorithm,[41] we introduce a Lamarckian refinement. Historically, two different mechanisms were proposed for evolution. The Darwinian theory of evolution[42] contended that the genetic makeup inherited at birth is the one passed on during mating to new offspring, in which case evolution is a product of environmental pressure and chance. Lamarck[43] surmised that, additionally, traits acquired during an individual's lifetime can be passed on to its offspring. Darwinian genetic algorithms[41] (GAs) are a class of search procedures which mimic natural evolution: populations of configurations are allowed to evolve through: (a) "mating," e.g., the creation of new offspring configurations which retain the characteristics of their parents; and (b) "survival of the fittest," where the fitness of an individual depends on its total energy. Our Lamarckian genetic algorithms (L-GAs) allows additionally for individuals to change their "genetic makeup" through a local fitness-improvement procedure prior to mating. This refinement is not limited to atomic relaxation[19,21–24] but, most importantly, also includes *transmutations of A* atoms into B atoms, guided by the local gradient of the total energy. Indeed, the computational effort needed to obtain the gradients of a functional, such as the *generalized* Ising model in this paper, is oftentimes less than the computational effort necessary for of full evaluation of the total energy. Departing from conventional *discrete* lattice representations,[11,16–18,25,30] where each site is occupied *either* by an A *or* a B atom, we use a continuous representation, as used by Wang *et al.*,[9] where each site i is occupied by a different "virtual" $\langle A_{x_i}B_{1-x_i}\rangle$ atom whose propensity to become pure A or pure B is reflected by the energy gradient with respect to x_i.[44]

In this paper, we examine the ability of genetic algorithm search strategies to address ground-state search problems with bit-string and reciprocal-space matings by using Darwinian evolution and Lamarckian refinements in the form of a virtual-atom minimization. We study both a single-solution problem, searching for the single deepest ground state of a given supercell, and a multiple-solution problem, searching for *all* ground states of a given supercell. The search strategies' success is defined as a statistical quantity. It allows us to rigorously compare the search strategies presented in the paper as well as set practical limits for their use. We find the reciprocal-space Lamarckian GA to be the most efficient search strategy, as well as the most "black box." Solving for the all ground states in the range $0 \leq x \leq 1$ of an Au_xPd_{1-x} alloy on a 12-atom fcc supercell requires on average as many as 330 total-energy evaluations and grows exponentially with the size of the supercell to 39 000 total-energy evaluations for 36 atoms. Hence, searching for *exact* ground states requires a computationally inexpensive functional such as cluster expansion. However, we show that if this constraint is relaxed and one searches for only an approximate result (within 3 meV),[39] the average number of evaluations falls down to 30 for a 12-atom supercell and to 580 for a 36-atom supercell.

II. SEARCHING THE CONFIGURATION SPACE IN TYPE (ii) PROBLEMS

A. Search space of a binary-alloy configuration on a fixed lattice: Supercells and supercell decorations

The search space considered in type (ii) problems consists of all periodic decorations of a binary alloy on a fixed lattice. It can be parametrized by two types of degrees of freedom: (i) the supercell type defining the periodicity of the structure and (ii) the decorations of the sites within this supercell. Thus, as shown in Ref. 11, the space consisting of N-atom configurations can be partitioned according to the translational symmetries into groups of "inequivalent cell shapes" (ICSs). Each ICS contains a number of "same-shape struc-

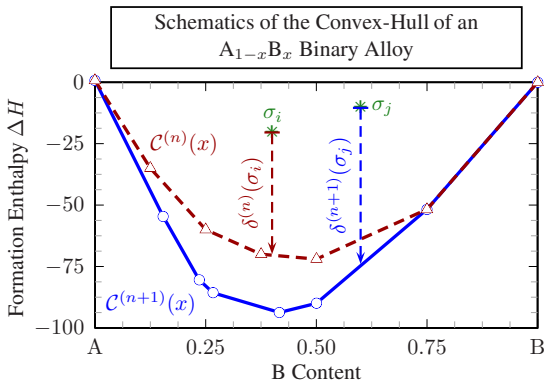

FIG. 1. (Color online) Schematic representation of the convex hull of a binary A_xB_{1-x} alloy at iterations n and $n+1$. It is constructed by linking the lowest lying structures such that $C^{(n)}(x)$ is convex. Structures lying on the convex hull are the only thermodynamically permissible structures at $T=0$ K. As shown in the figure, the values of the objective functions $\delta^{(n)}(\sigma_i)$ and $\delta^{(n)}(\sigma_j)$ of configurations σ_i and σ_j change as the convex hull is refined.

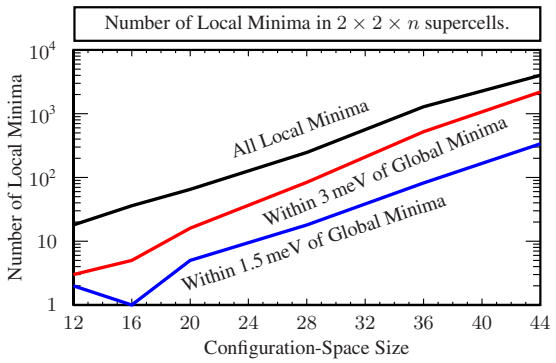

FIG. 2. (Color online) Number of local minima of face-centered-cubic $Au_{1-x}Pd_x$ for $2\times2\times n$ supercells with respect to the number of possible decorations, (2^{4n}). The growth is exponential with the number of atomic sites. The uppermost line accounts for all local minima, the middle line is for local minima with $\Delta H(\sigma) \leq -81.48$ meV, and the bottom line is for local minima with $\Delta H(\sigma) \leq -82.78$ meV. Especially for larger supercells, the local minima are concentrated around energies close to the global minimum. Local minima were obtained using the virtual-atom approach described in the text.

tures," e.g., different $A_pB_{q=N-p}$ decorations of the N periodic lattice sites. It was shown empirically[11] that the number of ICSs evolves as $\sim N^{2/3}$, whereas the number of all same-shape structures (reduced by the symmetries of the lattice) with N atoms evolves as $\sim Ae^{0.6N}$. In order to find the minimum-energy configurations of a binary alloy on a fixed lattice, we follow the strategy outlined in Ref. 11, whereby each ICS is sampled individually. There are 243 ICSs for $N \leq 20$ and 1282 ICSs for $N \leq 32$ which must be explored to search the full configuration space.

In many of the searches performed below, we will look for the ground states in a single inequivalent cell shape, e.g., one $n \times m \times p$ supercell, and sample its decorations. Obviously, obtaining the ground states of the system requires examining all ICSs, one by one. This is not attempted in this paper. The exact algorithmic details for such a procedure can be found in Ref. 11.

B. Definition of the ground states of a binary alloy in the $x \in [0,1]$ concentration range

In lattice statistical mechanics,[12,25,45] the ground-state line is defined via the convex hull $C(x_\sigma)$ (Fig. 1), which is the collection of all thermodynamically stable structures on a given lattice at $T=0$ K. It is constructed in two steps. *First*, one selects *at each concentration* x_i the configuration σ_i with the lowest formation energy $\Delta H(\sigma)$. In the *second* step, one deletes any configuration σ_{x_i} that can disproportionate into a sum of two neighboring configurations σ_{i-1} and σ_{i+1}, with $x_{\sigma_{i-1}} < x_{\sigma_i} < x_{\sigma_{i+1}}$,

$$\Delta H(\sigma_i) > \frac{x_{\sigma_{i+1}} - x_{\sigma_i}}{x_{\sigma_{i+1}} - x_{\sigma_{i-x}}} \Delta H(\sigma_{i-1}) + \frac{x_\sigma - x_{\sigma_{i-1}}}{x_{\sigma_{i+1}} - x_{\sigma_{i-1}}} \Delta H(\sigma_{i+1}). \tag{1}$$

Such configurations will disproportionate since their formation energies are higher than equivalent mixtures of the formation energies of their neighbors. $C^{(n)}(x_\sigma)$ denotes the convex hull constructed from all *known* configurations up to iteration n in the configuration search. To search simultaneously for *all* ground states over $0 \leq x \leq 1$ and over one ICS, we define the fitness of a configuration σ with concentration x_σ as the depth

$$\delta^{(n)}(\sigma) = \Delta H(\sigma) - C^{(n)}(x_\sigma), \tag{2}$$

e.g., as the difference between the formation energy $\Delta H(\sigma)$ and the known convex hull $C^{(n)}(x_\sigma)$ at iteration n (see Fig. 2). In comparison to previous approaches[11,21,38] where the stoichiometry A_pB_q of the search space was fixed, this fitness function allows us to examine the full concentration range $x \in [0,1]$ simultaneously.

It is often the case that the property one is searching for is single valued, e.g., one is searching for the maximum electronic polarizability.[9] As such, we present also in the following the results for seeking the deepest minimum-energy configuration only, in which case the formation enthalpy $\Delta H(\sigma)$ alone is minimized. Appendix B discusses how using the distance from the known convex hull as an objective function can also speed up the search for the deepest ground state only (as opposed to *all* ground states).

C. Exact versus "fuzzy" answers

When searching for the thermodynamically stable decorations at 0 K of a binary alloy, one is confronted with a problem where nothing less than the global optima defined by the convex-hull line are acceptable solutions. Nonetheless, there are problems for which a "fuzzy" answer will do. For instance, one may seek superlattices of quaternary semiconductors[46] with a band gap of *approximately* 310 meV. We will show that such a fuzzy search is remarkably less demanding (by a few orders of magnitude) than the "exact" search required by *exact* ground-state determination.

D. Type of questions asked

We will show that the performance of a search is highly dependent upon the exact nature of the question asked. We present in the following four searches with four different objectives:

Question 1. Find the *exact deepest* ground state of a binary alloy on fixed lattice for a given $n \times m \times p$ supercell (e.g., for one ICS).

Question $1'$. Find the *approximate deepest* ground state of a binary alloy on fixed lattice for a given $n \times m \times p$ supercell.

Question 2. Find *all exact* ground states of a binary alloy on fixed lattice for a given $n \times m \times p$ supercell.

Question $2'$. Find *all approximate* ground states of a binary alloy on fixed lattice for a given $n \times m \times p$ supercell.

The four questions are separated according to whether one seeks multiple answers, e.g., questions 1 and $1'$, for which we search for a number of ground states simultaneously, or singular answers, e.g., questions 2 and $2'$, for which we search for a single ground state or an approximate ground state. Furthermore, we differentiate between exact and fuzzy searches. For instance, we define an answer to question $1'$ as any configuration less than 3 meV from the exact deepest ground state, as obtained from answering question 1. In a similar fashion, we formulate question $2'$ as the search for *all approximate* ground states. More explicitly, we search for a convex-hull line $C_{3\ \mathrm{meV}}(x)$ which is no more than 3 meV from the exact convex hull line $C(x)$ (obtained here from answering question 2) for a given $n \times m \times p$ supercell, e.g., $\forall x \in [0,1]$, $|C_{3\ \mathrm{meV}}(x) - C(x)| \leq 3$ meV. In practice, one does not know $C(x)$ beforehand. Hence, one cannot know without solving question 2 when question $1'$ has been solved. Nevertheless, this formulation allows us to compare the expense of absolute convergence with approximate convergence, as well as compare exact versus fuzzy search goals. Note that each of these questions is termed with respect to a single ICS. The result for all configurations with N lattice sites is recovered by searching each of the $\sim N^{2/3}$ ICSs.

III. MATING AND EVOLVING ALLOYS ON A FIXED LATTICE

The configuration space of metallic binary alloys such as $Au_{1-x}Pd_x$ is extremely complex. Indeed, it is expected that swapping an Au atom for a Pd atom will require little energy. In Fig. 2, a local minimum is defined as any decoration for which swapping any Au atom with a Pd atom (or vice versa) will raise the formation enthalpy. We find that the number of local minima grows exponentially with the size of the supercell. Furthermore, a large number of local minima are no more than 3 meV from the global minima. The complexity of the search space prompts us to use methods which are not hindered by local minima, namely, genetic algorithms (GAs). In Sec. III A below, we describe two general flavors of GA, one inspired from the Darwinian theory of evolution and the other from the Lamarckian theory of evolution. We then describe two possible mating processes, a real-space bit-string mating and a reciprocal-space mating.

A. Lamarckian genetic algorithm versus Darwinian genetic algorithm

Historically, two different theories of evolution attempted to describe the apparition and survival of species in nature. In both theories, individuals are born with traits inherited from their parents and are able to pass on traits to their offsprings. Depending on these traits, individuals may be better fitted to survival than others, such that eventually "good" traits persist across generations, whereas bad traits are lost. What happens in between birth and procreation is where the Lamarckian and the Darwinian theories of evolution differ. In the former case, individuals have the capacity to modify the traits they inherited at birth such that they become more apt at survival. It is the *modified* traits which are passed on to the offspring. In the latter case, traits inherited at birth cannot be modified and are passed on "as is" to the offspring, in which case luck (and environmental pressure), rather than individual self-determination, is the motor of evolution.

A Darwinian GA proceeds as described above by survival of the fittest only. Individuals cannot improve during their lifetimes. Their fitness at birth defines their chances of mating and survival. Lamarckian genetic algorithms are widely used in structural optimization[19,21–24] (e.g., optimization of cell vectors and atomic positions in nanoclusters or periodic crystalline structures). Generally, each newly created offspring undergoes atomic relaxation, thereby improving the traits—the atomic positions—it received at birth. It is the improved atomic positions which it passes on to its own offspring. The goal of such a procedure within GA is to simplify the energy landscape by considering only local minima for mating. We propose here a Lamarckian genetic algorithm where the atomic *species* occupying the lattice sites are relaxed using the virtual-atom scheme described below. We will compare the efficiency of the Lamarckian GA with respect to that of the Darwinian GA. Note that the cluster-expansion functional includes positional relaxation *de facto*.

B. Atomic-specie relaxation with the virtual-atom procedure

The challenge in any type of decoration search is that the search space is *discrete*, each lattice site being occupied by either an A or a B atom. A local refinement of the site occupations can be defined using local $A \leftrightarrow B$ transmutations at some lattice sites. The procedure is complicated by the fact that, *a priori*, one cannot use gradient-directed minimization methods, since gradients do not exist between discrete site occupations. This type of minimization problem where variables are limited to integer values is generally known as "integer programing."[47] They are often treated by relaxing the integer constraints in some way. This approach was taken in, e.g., physics and engineering by Bendsøe and Kikuchi,[48] who were interested in optimizing the spatial distribution of anisotropic material for mechanical load and other properties. Marzari *et al.*[49] considered virtual atoms, or "alchemical potentials," as perturbations to the virtual crystal approximations. In conjunction with Monte Carlo calculations, they recovered the epitaxial phase diagram of $Ga_xIn_{1-x}P$.[49] The idea was then used in the context of molecular decoration by

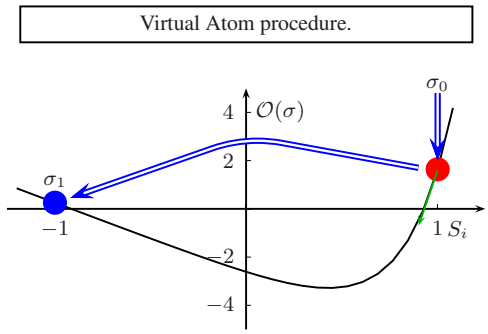

FIG. 3. (Color online) Description of the virtual-atom jump minimizer. The algorithm evaluates $\mathcal{O}$ only at physical configurations and its gradients only between first-neighbor physical points: (i) A physical configuration σ_0 is chosen as a starting point. (ii) The gradient at one lattice site i is evaluated. (iii) If this gradient is negative, then $\mathcal{P}(\sigma_1)$ is evaluated, and the search proceeds from the better structure σ_0 or σ_1. (iv) If, on the other hand, the gradient is negative, the gradient at the next lattice site j is evaluated, and the search proceeds from there.

Wang *et al.*[9] In their "linear combination of atomic potentials," each of the substitutional sites i of a backbone molecule is decorated by a fictitious atom (or chemical group) with a fictitious potential V_i, where $V_i = x_i V_A + (1-x_i) V_B$ is the concentration-weighted linear average of the end-point potentials V_A and V_B. Each lattice site, or molecule site, is occupied by a different virtual atom. By defining a virtual fitness $\delta^{(n)}(\sigma)$ using these virtual atoms, the gradients $\partial\delta^{(n)}(\sigma)/\partial x_i$ can be introduced which represent the "chemical appeal" of an $A \leftrightarrow B$ transmutation at site i.

The original approach of Wang *et al.*[9] consists of minimizing the virtual total energy with respect to the occupations x_i, often resulting in a nonphysical minimum where the sites are occupied by fictitious virtual atoms. Unfortunately, there is no clear link between the virtual minimum and the physical minimum-energy decoration. Furthermore, it is difficult to enforce physicality constraints such as $S_i^2 = \pm 1$ using either Lagrangian multipliers or penalty functions. Generally, "constrained" minimization procedures work by first finding a region of space where the constraints are satisfied and then looking for the minimum within this region. In our case, each region spans a discrete point of the configuration space. Hence, a constrained minimization approach will yield only the closest physical point from the starting point, rather than the physical minimum.[50] The virtual-atom "jump" strategy we have adopted (Fig. 3) circumvents these difficulties. We evaluate $\delta^{(n)}(\sigma)$ and first-order derivatives with respect to $A \leftrightarrow B$ transmutation only at physical points. Proceeding from a starting *physical* configuration σ_0, the gradient $\partial\delta^{(n)}(\sigma_0)/\partial x_i$ at a random site i is evaluated. A negative gradient indicates that transmuting the atom at site i could lead to a lower energy. In that case, the depth $\delta^{(n)}(\sigma_1)$ of this neighboring configuration σ_1 is evaluated, and the procedure iterates from the better configuration σ_0 or σ_1. Otherwise, if the gradient is positive, the procedure iterates with a different lattice site j. The order in which the sites are evaluated is random. Convergence is deemed achieved when every lattice site has been explored without predicting or finding a better neighboring configuration. A similar approach was introduced recently by Keinan *et al.*[10]

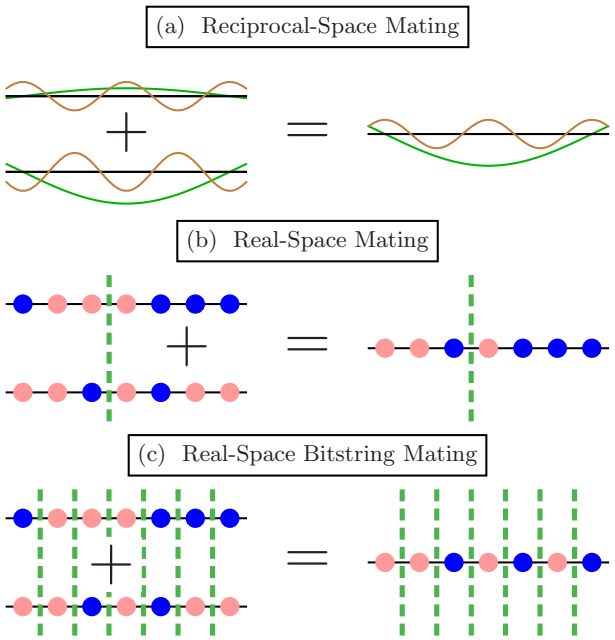

FIG. 4. (Color online) In (a) reciprocal-space mating, individual configurations are identified as waves of A/B material (represented by sinusoidal lines). A new individual is constructed from the A/B material waves of two parents. In (b) real-space mating, the two parents are cut in half by a plane [dashed line in one-dimensional (1D) figure] and the two half structures are spliced together into a new individual. With bit-string mating (c), each lattice site is mapped as a bit on a bit string; this is equivalent to a real-space mating, where each atom is at the center of its own cut-and-splice region.

C. Real-space bit-string search versus reciprocal-space search

Mating is the key to a successful GA: it must be able to identify and pass on favorable traits and patterns from parents to offspring. Previous genetic algorithms for type (iii) general space-group optimization[19,21–23,38,51] and type (ii) configuration-search[11] mate individuals directly in *real space* (see Fig. 4). The former approaches[19,21–23,38,51] generally define a plane in real space which cuts the configuration of each parent into two [see Fig. 4(b)]. Two "half structures" are then chosen and spliced together to form a new individual. The objective is to integrate into one individual the real-space patterns present in each half of the parents. Unfortunately, this procedure also produces patterns straddling the cutting plane which are not inherited but are rather an artifact of the mating procedure. As such, it is likely that these patterns are not particularly fit, i.e., that they do not correspond to low-energy structures. Type (ii) configuration search proceeds by swapping atom types between two parents, e.g., by performing a standard bit-string mating where the occupation of each site i in the ICS is identified by a "spin variable" $S_i = \pm 1$ (where $S_i = 1$ corresponds to an occupation of site i by an A atom and $S_i = -1$ to that by a B atom). It can also be viewed as an extreme version of real-space mating where

each parent is cut into as many "cut-and-splice" regions as there are atoms in the supercell [see Fig. 4(c)]. In addition to the cut-and-splice cross-over operation described above, one also introduces a real-space mutation operator where randomly chosen atoms in the unit cell are flipped from an A to a B atom. The rate with which the mutation operator is applied and the number of sites to flip per mutation are controlled via two parameters. We present here a mating procedure for type (ii) optimizations, reciprocal-space mating, which does not suffer from the drawbacks of cut-and-splice operation and incorporates crossover and mutations in a single operation.

Reciprocal-space mating [Fig. 4(a)] proceeds by characterizing configurations not through the real-space occupations $S_i = \pm 1$ of site i but rather through their Fourier transform, i.e., the structure factor of the configuration $S(\mathbf{k})$,

$$S(\mathbf{k}) = \sum_{j \in \sigma} S_j e^{i\mathbf{R}_j \cdot \mathbf{k}}, \tag{3}$$

where $\mathbf{R}_j$ is the position of lattice site j in the ICS and $\{\mathbf{k}\}$ is the finite set of reciprocal-space vectors $\mathbf{k}$ for which $S(\mathbf{k})$ is nonzero. The set of amplitudes $S(\mathbf{k})$ uniquely identify each configuration σ as a sum of "concentration waves" of materials A and B (see Fig. 4). Reciprocal-space mating proceeds in two steps: (i) an intermediate configuration is created using a cut-and-splice approach in reciprocal space, and (ii) a "discrete" configuration is obtained from the intermediate structure with complex lattice-site occupations. In the first step, two parents are selected and their $S(\mathbf{k})$ are identified. A new *intermediate* configuration is created by setting the value of each of its structure factors $S(\mathbf{k})$ to that of one parent or the other. As such, the intermediate configuration is the interference pattern of inherited concentration waves. When transformed to real space, this intermediate configuration is not necessarily physical in the sense that $S_i \neq \pm 1$ (indeed, S_i could be complex). In the second step, a *physical* offspring configuration is then deduced by mapping each spin to -1 or 1, depending on whether its value in the intermediate configuration is on the left- or right-hand side of the complex plane. This mating procedure has the advantage of directly representing expected patterns of ordered structures, such as ordering (A-B) or clustering (A-A) of first-neighbor pairs. Furthermore, since concentration waves span the whole periodic supercell, the occupation of any one site is correlated with the occupation throughout. In comparison, in real-space mating,[19,21–23] the inheritance of sites on one side of the cutting plane are not correlated with the inheritance of sites on the other. As such, beneficial ordering patterns across the whole configurations are less likely to appear. For instance, ordering or clustering of first-neighbor pairs is inherently broken across the cutting plane. By representing periodic configurations with a natural basis (reciprocal-space waves) for periodic ordering, we expect beneficial ordering patterns to appear and be retained with more ease. Note that the reciprocal-space mating is not restricted to systems with dominant pair interactions or superlattice ground-state structures.

We will examine the success rates of four different genetic search strategies: (1) real-space bit-string Darwinian genetic algorithm (**r**-GA), (2) reciprocal-space Darwinian genetic algorithm (**k**-GA), (3) real-space bit-string Lamarckian genetic algorithm (**r**-LGA), and (4) reciprocal-space Lamarckian genetic algorithm (**k**-LGA). We show that the reciprocal-space Lamarckian genetic algorithm solves the questions in Sec. II D at least as fast, and generally faster, than the other three methods. Furthermore, it is the most *black box* of the four methods, requiring for best efficiency the optimization of only two parameters, the population size and the number of generated offsprings, across a relatively narrow range (see Appendix A).

IV. COMPARING THE SUCCESS RATES OF STOCHASTIC SEARCH STRATEGIES

Genetic algorithms are essentially stochastic. The appearance of traits within a population, e.g., low-energy configurational patterns, is mostly dictated by random choice, even if their persistence is determined by their relative fitness. When starting an evolutionary process, one is never assured how soon good traits will happen. For instance, we will show the reciprocal-space Lamarckian genetic algorithm presented above will recover all thermodynamically stable decorations at 0 K of a $2 \times 2 \times 8$ supercell with on average 7700–10 400 evaluations of the cluster-expansion functional. This rather large interval is a direct result of the stochastic nature of the genetic algorithms. It is not limited however to GA search strategies. Indeed, were we to use a deterministic approach alone, such as the virtual-atom procedure described above, any given run would not be assured to find the global minima. One generally proceeds then by restarting the deterministic approach of choice from different random starting positions.

In the following we rigorously compare the four search strategies by assessing their statistical success rates. More explicitly, we define the degree of confidence $\epsilon(n)$, e.g., the expectation of success, as the ratio of evolutionary runs which have met success with n or less calls to the cluster-expansion functional over the number of performed runs (generally 400). We then explore the behaviors of the four strategies in relation to the size of the system (e.g., the number of atoms considered) for a degree of confidence $\epsilon = 95\%$.

In practice, one does not know *a priori* when a search procedure has converged toward the global minimum. Nonetheless, a statistical study requires the end result to be known. In our case, they were obtained by running the search procedures *ad absurdum*. Such an approach is not conceivable for practical studies. We hope by performing it here to provide reasonable estimations as to how long search procedures should be allowed to go on to obtain results with a good degree of confidence.

V. COMPUTATIONAL DETAILS

A. Fitting the configuration space to a cluster-expansion functional

In principle, all quantities described above could be computed from a *direct* energy functional $E_{\text{direct}}(\sigma)$. Here, we

choose to first fit total energies computed from first principles onto a cluster-expansion functional. We will use a cluster-expansion functional of $Au_{1-x}Pd_x$ which has been previously fitted to a few density-functional-theory (DFT) total energies.[28] The resulting cluster-expansion functional allows us to compute formation enthalpies outside the fitting set to within ~3 meV of DFT energies at a fraction of the cost in computer resources. Indeed, beyond the need for a fast functional to perform the ground-state search detailed in this paper, the cluster-expansion functional can also be used to obtain the formation enthalpy of the random alloy, or even x-T phase diagrams using the Monte Carlo method.[52]

Configurational searches can be performed on explicit Born-Oppenheimer energy surfaces $E_{\text{direct}}(\sigma,\{\mathbf{R}\})$ evaluated on the fly[53,54] or by first parametrizing $E_{\text{direct}}(\sigma,\{\mathbf{R}_{\text{min}}\})$, where $\{\mathbf{R}_{\text{min}}\}$ is the set of relaxed atomic positions of configurations σ. The latter can be given by cluster expansion, which fits a few (~50) total-energy calculations obtained from DFT to a generalized Ising model,

$$\Delta H_{\text{CE}}(\sigma) = J_0 + \frac{1}{N}\left(\sum_i J_i S_i + \sum_{ij} J_{ij} S_i S_j + \cdots\right) + \sum_{\mathbf{k}} \Delta E_{\text{CS}}(\hat{\mathbf{k}},x_\sigma)|S(\mathbf{k})|^2 F(\mathbf{k},x_\sigma), \quad (4)$$

where the occupation of each lattice site i by an A or a B atom in configuration σ is parametrized by the value of a discrete spin variable $S_i = \pm 1$. The reference energy J_0, the on-site interaction-energy J_i, the pair interaction energies J_{ij}, as well as other possible higher-order interaction energies, represent the excess chemical formation energy of each site i in conjunction with its neighboring environment. The long-range strain effects are represented by the last term. It is composed of a sum over the finite set of allowed reciprocal-space vectors.[29] $\Delta E_{\text{CS}}(\hat{\mathbf{k}},x)$ is the cost in energy of maintaining lattice coherence between pure A and pure B along an interface of orientation $\hat{\mathbf{k}}$. $S(\mathbf{k})$ is the structure factor of configuration σ for the reciprocal-space vector $\mathbf{k}$. $F(\mathbf{k},x_\sigma)$ is defined as $F(\mathbf{k},x_\sigma)=e^{-\mathbf{k}^2}[4x_\sigma(1-x_\sigma)]^{-1}$. The random-alloy mixing energies $\Delta H_R(x)$ can be obtained by taking the analytical configurational average of $\Delta H_{\text{CE}}(\sigma)$. *A priori*, optimizing $\Delta H_{\text{CE}}(\sigma)$ is a *discrete* problem since each lattice site is occupied by either an A or a B atom.

As described above, the cluster-expansion functional contains two terms: Ising-type pair and many-body interactions which describe the chemical interactions between atomic species at different lattice sites and a constituent strain part which models the elastic energy. The former can be and is made linear with respect to any given spin. The latter depends on the structure factor of the configuration and cannot be quite as easily linearized.

Once obtained, cluster expansion presents a number of advantages: (i) configurations *outside* the fitting set are predicted with a statistical error of $\leq$3 meV with respect to their local-density approximation (LDA) excess formation energy; (ii) it provides the excess formation energy of *atomically relaxed* configurations; (iii) for configurations of $N \leq 20$ atoms, direct enumeration methods exist[30] which can be used to confirm the findings of the sampling methods presented in this paper; and (iv) evaluating an already generated $\Delta H_{\text{CE}}(\sigma)$ is effortless compared to LDA. Note that since $\Delta H_{\text{CE}}(\sigma)$ reproduces *ab initio* results quite accurately, we expect that the behavior of optimizers which use $\Delta H(\sigma)$ only (as opposed to the virtual-atom approach, which also uses its gradient) will be quantitatively the same for both cluster expansion and *ab initio*.

As an example, we will study in this paper face-centered-cubic $Au_{1-x}Pd_x$ alloy, using the cluster expansion parametrized by Barabash *et al.*[28] They evaluated ~50 energies $\Delta H(\sigma_i)$ of ordered configurations and selected the appropriate pair and many-body interactions $\{J\}$ via a genetic algorithm.[55] $\Delta H_{\text{CE}}(\sigma)$ is obtained from the following iterative procedure:[28–30] (i) the formation enthalpy of ~20 four-atom configurations are computed using density-functional theory, (ii) trial cluster-expansion functionals with optimized sets of interactions are fitted to these configurations, (iii) a configuration search yields the ground states for these functionals, and (iv) the formation enthalpies of predicted ground states are computed with DFT and added to the fitting set. The cluster expansion is deemed converged when no new ground states (with $N \leq 20$ atoms) are found. The formation enthalpy of face-centered-cubic $Au_{1-x}Pd_x$ is described using 13 pair interactions, 5 three-body interactions, and 2 four-body interactions,[28] as well as a constituent strain term.[29]

B. Details of the genetic algorithms

The procedures are parameterized according to the population size and the replacement rate (number of offsprings per generation). Furthermore, the real-space bit-string mating contains two additional variables: one controlling how often crossover is carried out over mutation and another controlling the number of atoms transmuted during each mutation operation. The results presented below have been optimized by hand for maximum efficiency at degree of confidence $\epsilon = 95\%$. A brief description of how these parameters affect the efficiency of the search strategy is given in Appendix A. Overall, the reciprocal-space Lamarckian genetic algorithms require the fewest adjustments to the parameters for maximum efficiency. In general, gradients are much less expensive to obtain within first-principles pertubation theory than total energies. For the sake of simplicity, the computational cost of obtaining gradients in the following statistical study is set to zero. More details on the effect of including the computational cost of the gradient evaluations can be found in Ref. 39. In order to promote diversity within the GA population and enable the search to explore a larger region of the configuration space, we have introduce a "diversity constraint"[39] such that candidate offsprings are rejected if they are clones of individuals in the current population or in the newly generated (and accepted) offspring. As such, there is never more than one copy of a configuration in the population at any given point during the evolutionary run. Furthermore, during the Lamarckian refinements, transmutations which would reproduce an individual of the current population are also disallowed.

VI. RESULTS

In Secs. VI A and VI D, we discuss each of the behavior of the search strategies for each question separately. Ques-

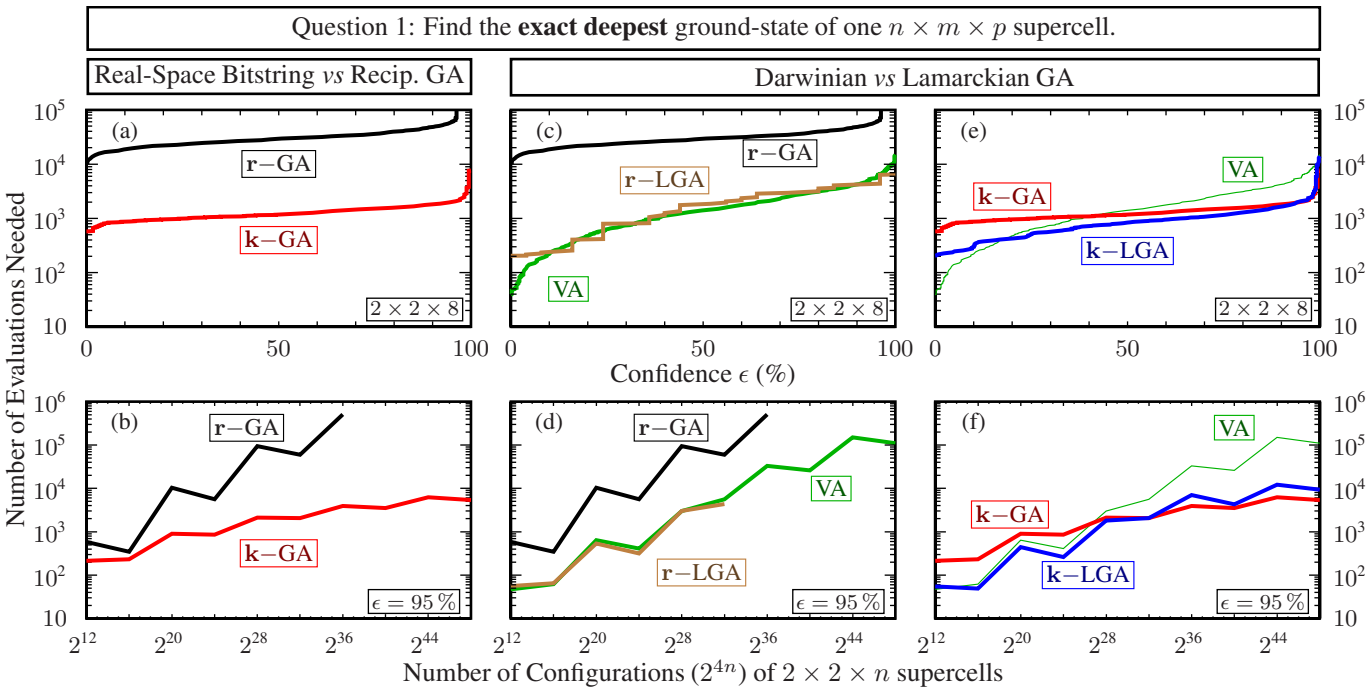

FIG. 5. (Color online) Finding the exact deepest ground states of $n \times m \times p$ supercells of fcc $Au_{1-x}Pd_x$ (question 1): (a) and (b) compare real-space GA (**r**-GA) and reciprocal-space GA (**k**-GA), whereas (c)–(f) compare Darwinian evolution (**r**-GA and **k**-GA) to Lamarckian evolution (**r**-LGA and **k**-LGA). Results for the VA approach with multiple restart from different random starting points are also presented. The upper figure shows the number of total-energy evaluations required to find the deepest ground state of a $2 \times 2 \times 8$ supercell with a degree of confidence $\epsilon(\%)$. Shown in the lower panel is the number of evaluations required for a $2 \times 2 \times n$ supercell with a degree of confidence of $\epsilon = 95\%$. These statistics have been compiled from 400 independent runs for each strategy and supercell. The formation enthalpies of the Au_xPd_{1-x} alloy on a fcc lattice were obtained from a cluster-expansion functional parametrized in Ref. 28. The overall behavior is characterized by an exponential increase in the required number of total-energy evaluations with the size of the system $N=nmp$. Addressing question 1 for systems with more than 20 atoms is hardly feasible with DFT-based functionals.

tions 1, 2, 1′, and 2′ correspond, respectively, to Figs. 5–8. The top three panels of each figure show the average number of evaluations of the cluster-expansion functional required to solve the figure's particular question with an expectation for success ϵ in $2 \times 2 \times 8$ supercells. In the lower panels, we report the number of evaluation required on average by $2 \times 2 \times n$ supercells of increasing sizes and degree of confidence $\epsilon = 95\%$. Statistics were obtained using a minimum of 400 independent evolutionary runs for each search strategy.

A. Question 1: Finding exact deepest ground states of a given $n \times m \times p$ supercell

1. Real-space GA versus reciprocal-space GA

Figures 5(a) and 5(b) compare the use of real-space bit-string and reciprocal-space matings in GA. Figure 5(a) reports the average number of evaluations required to achieve a given expectation for success, or degree of confidence, within a $2 \times 2 \times 8$ supercell. We find that the reciprocal-space mating performs better than the real-space bit-string mating for any degree of confidence ϵ. The reciprocal-space mating generally needs a smaller population size, leading to a shorter "learning period," during which the search strategies evolve the starting population to the region of space containing the deepest ground state, hence resulting in a much lower value at $\epsilon = 0\%$. For most of the degree of confidence range $\epsilon \in [0\%, 100\%]$, the curve is linear. This means that higher success rates come at an exponential increase in the required number of evaluations. Figure 5(b) reports the number of evolutions required to find the exact deepest ground state of $2 \times 2 \times n$ supercells versus n with a degree of confidence of $\epsilon = 95\%$ for real-space bit-string and reciprocal-space GAs. We find again that the reciprocal-space mating is much more efficient. Interestingly, both curves are sawlike, with the number of evaluations required by $2 \times 2 \times n$ supercells with even n smaller than those with odd n. The cluster expansion of $Au_{1-x}Pd_x$ is clustering, e.g., the first-neighbor pair interaction favors Au-Pd patterns over Au-Au or Pd-Pd patterns. This type of arrangement is not possible for odd n, leading to a more complex configuration space.

2. Darwinian evolution versus Lamarckian evolution

Panels (c) and (d) and (e) and (f) of Fig. 5 compare Lamarckian and Darwinian evolutions with real-space bit-string mating and reciprocal-space mating, respectively. We also plot the behavior of the virtual-atom search strategy with multiple random starts. We find that the real-space Lamarckian evolution performs better than its (Darwinian) GA counterpart, i.e., **r**-LGA is faster than **r**-GA and **k**-LGA is faster than **k**-GA. At the same time, the virtual-atom approach and the real-space Lamarckian GA are comparable. Whether it is

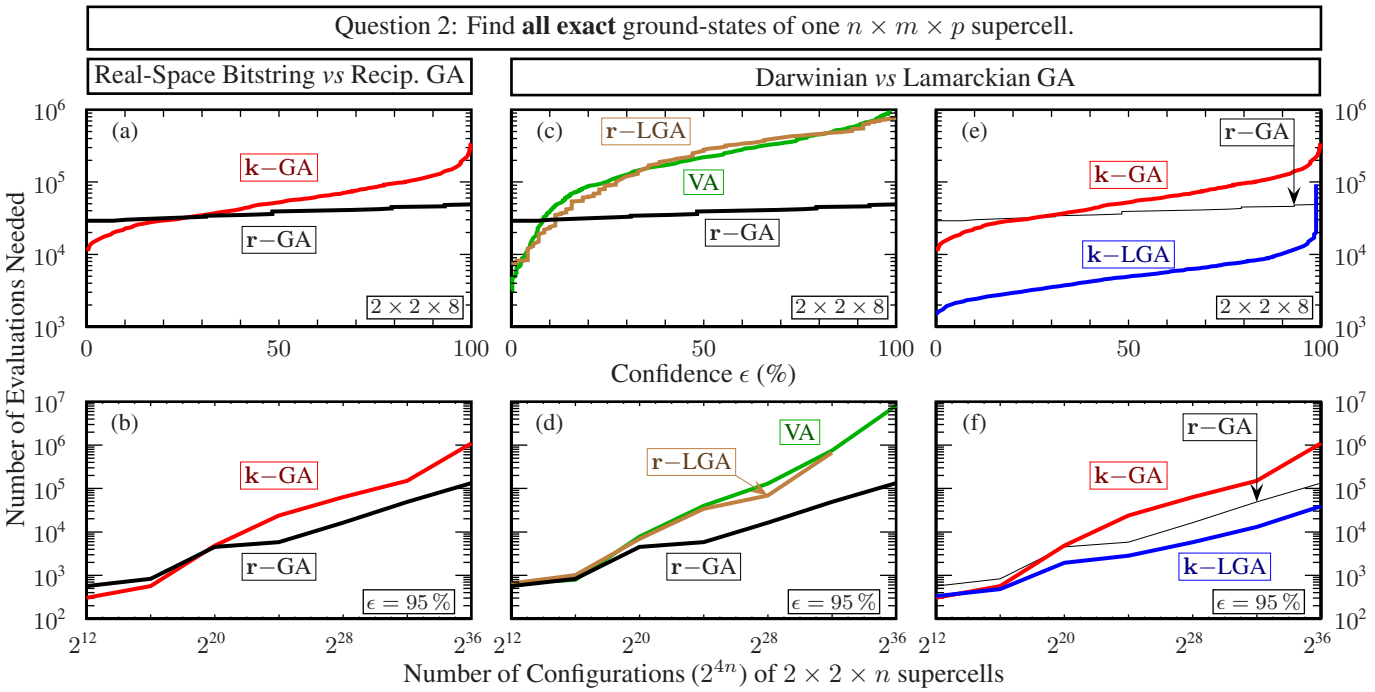

FIG. 6. (Color online) Searching for *all exact* ground-states of $n \times m \times p$ supercells of fcc $Au_{1-x}Pd_x$. See caption of Fig. 5 for more details. Conclusions: Panels (c) and (d) show that the real-space genetic algorithms (**r**-LGA) do not benefit from mating local minima only. The computational cost of searching simultaneously for all exact ground states is less than an order of magnitude higher than searching for the single exact deepest ground-state configuration (see Fig. 5). In general there are 5–12 ground states per supercell.

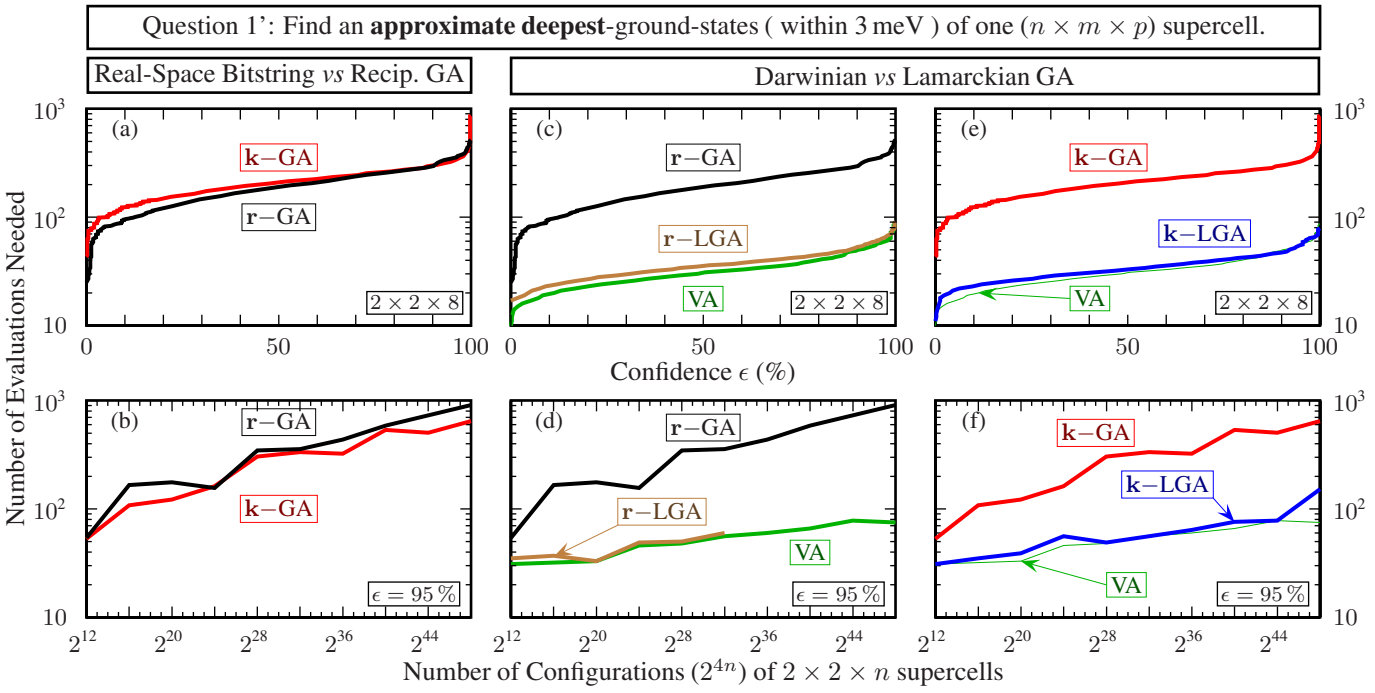

FIG. 7. (Color online) Finding an *approximate deepest* ground state (within 3 meV) of $n \times m \times p$ supercells of fcc $Au_{1-x}Pd_x$ (question 1′). See caption of Fig. 5 for more details. Conclusion: Searching for an *approximate* solution is remarkably less complex than searching for exact solutions. Indeed, the latter is an NP-complete problem (the average number of evaluations increases exponentially with the number of atoms, $N=4n$), whereas the empirical evidence above and in Fig. 2 shows that former may very well not be NP complete (see discussion in text).

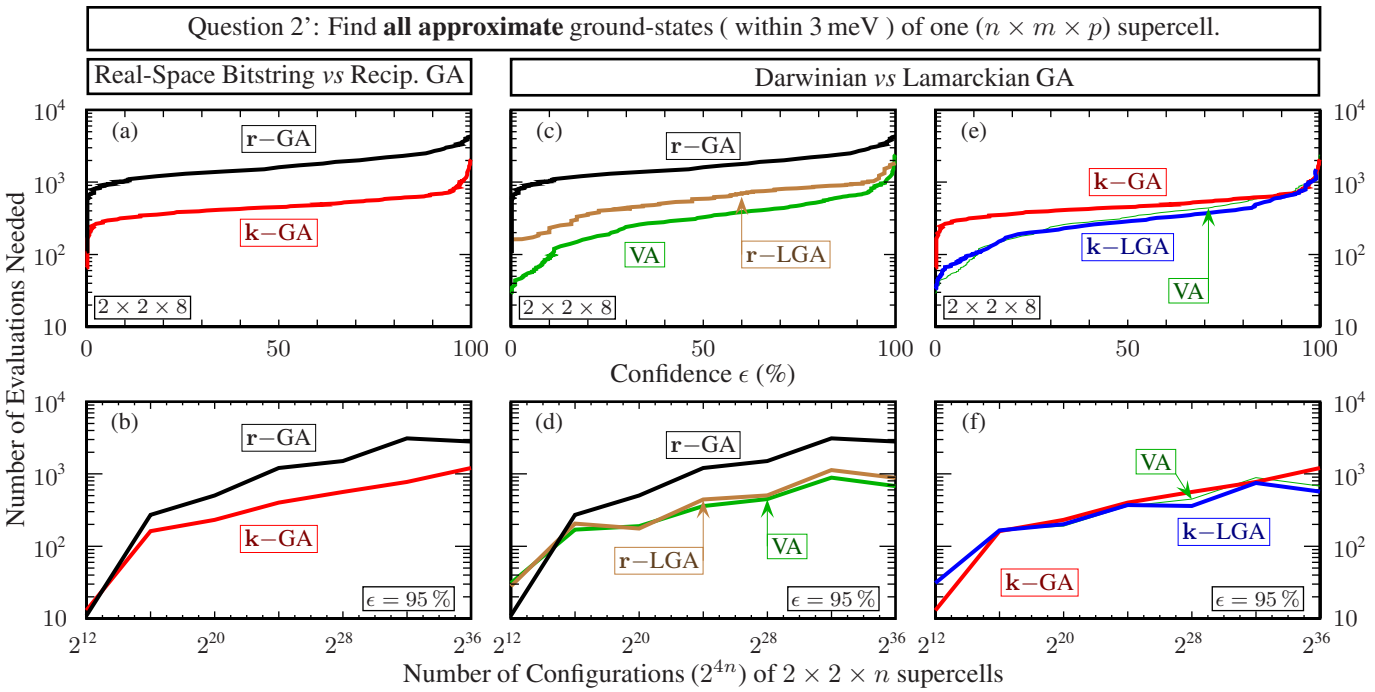

FIG. 8. (Color online) Searching for *all approximate* ground states (within 3 meV, see Sec. II C) of $n \times m \times p$ supercells of fcc $Au_{1-x}Pd_x$ (question 2′). See caption of Fig. 5 for more details. Conclusion: Using a direct method such as the VA approach is remarkably effective when searching for approximate solutions. It is most likely correlated with the large ratio of acceptable solutions (within 3 meV) to exact solutions (see Fig. 2). **k**-LGA is overall the most effective method for finding answers to this search, as well as those in Figs. 5–7.

with respect to the degree of confidence ϵ [panel (c)] or the size of the configuration space [panel (d)], **r**-LGA performs only as well as the virtual-atom approach. It would seem that the computing cost inherent to performing a virtual-atom (VA) minimization overwhelms the Lamarckian-GA benefit of mating local minima only. On the other hand, we find in panels (e) and (f) that reciprocal-space mating performs at least as well with a Lamarckian evolution (**k**-LGA) than it does with a Darwinian evolution (**k**-GA). Overall, the reciprocal-space Lamarckian, **k**-LGA, is the most effective method for finding the exact deepest ground state.

B. Question 2: Finding all exact ground states of a given $n \times m \times p$ supercell

1. Real-space GA versus reciprocal-space GA

Figure 6 reports the number of evaluations with respect to ϵ and the size of the configuration space required to find all exact ground-states of a given supercell. Panels (a) and (b) show that in contrast with question 1 (Fig. 5), the real-space bit-string **r**-GA is more efficient than the reciprocal-space **k**-GA for question 2 at larger configuration space and confidence values. Interestingly, we note that the real-space bit-string strategy is more effective at finding *all* ground states (Fig. 6) than it is at finding only the deepest one (Fig. 5). This may seem a paradoxical result. It stems from the different objective functions which are minimized in the two cases. To find the deepest ground state, we chose to minimize the formation enthalpy $\Delta H(\sigma)$ alone. When searching for all ground states, we minimize the distance to the known convex hull, $\Delta H(\sigma) - C^{(n)}(x_\sigma)$. The latter objective incorporates more information than is needed *a priori* to find the exact deepest ground state. Nonetheless, it makes the search more effective. This result is quite general and is discussed in detail in Appendix B.

2. Darwinian evolution versus Lamarckian evolution

Panels (c) and (d) of Fig. 6 compare the real-space Lamarckian **r**-LGA and Darwinian **r**-GA. We find again that the real-space bit-string Lamarckian **r**-LGA behaves much like the virtual-atom search. In fact, the latter does not solve the present question very effectively, making the real-space **r**-GA a better choice here than its Lamarckian counterpart **r**-LGA. Panels (e) and (f) compare the reciprocal-space Lamarckian and Darwinian evolutions. We find that the reciprocal-space Lamarckian GA is more efficient than either the Darwinian reciprocal-space GA or the Virtual Atom approach by more than an order of magnitude for larger supercells.

C. Question 1′: Finding an approximate deepest ground state of a given $n \times m \times p$ supercell

1. Real-space GA versus reciprocal-space GA

Figure 7 reports the number of evaluations with respect to ϵ and the size of the configuration space required to find an *approximate* deepest ground state of a given supercell. We define an approximate deepest ground state as any configuration within 3 meV of the exact deepest ground state found previously. For all search strategies, solving for an approximate ground state is spectacularly less challenging. Panels

TABLE I. Number of evaluations required to find the exact deepest ground state with 95% degree of confidence using reciprocal-space Lamarckian GA (**k**-LGA) with different population sizes. If the population size is too small, then the search has the propensity to lock itself upon not-quite-optimal traits, resulting in slow convergence. On the other hand, larger populations are slower to evolve. As such, there exists an optimal population size for which the search will converge the fastest.

	Population size							
Supercell	10	20	30	40	50	60	80	100
$2\times2\times3$	60	70	90	100	130			
$2\times2\times4$	60	80	110	130	170			
$2\times2\times5$	5500	1500	440	440	510			
$2\times2\times6$	260	300		370	410	420	490	530
$2\times2\times7$				2200	2500	2100	2100	1800
$2\times2\times8$		13000		2800		2100	2400	2300
$2\times2\times9$						12000	8000	7000

(a) and (b) show little difference between the real-space **r**-GA and the reciprocal-space **k**-GA, with the latter having a slight edge. Figure 2 shows that a very large proportion of local minima are within 3 meV of the deepest ground state; hence these are *acceptable* answers to this question. The relative abundance of solutions to the problem may explain why **r**-GA and **k**-GA display similar success rates.

2. Darwinian evolution versus Lamarckian evolution

In panels (c) and (d) of Fig. 7, we show once again that the real-space bit-string Lamarckian evolution **r**-LGA behaves like the simpler VA approach. However, the virtual-atom approach turns out to be very efficient when searching for the approximate deepest ground state. This is not surprising. Indeed, we have shown in Fig. 2 that a large ratio of the local minima lies close to the exact deepest ground state. In fact, it grows larger with the size of the configuration space for those supercells we have studied. Since local minima can be obtained in polynomial time and since the number of local minima close to the exact deepest ground state increases exponentially, it is most likely that the approximate problem solved in this section is *not* NP complete. Panels (e) and (f) compare the reciprocal-space **k**-GA and its Lamarckian counterpart **k**-LGA. We find once again that the Lamarckian evolutionary strategy with reciprocal-space mating is better than its Darwinian counterpart. Overall Fig. 7 shows that searching for an approximate answer takes less than 100 evaluations, even for a supercell with 48 atoms; hence it is viable with computationally intensive functionals such as DFT.

D. Question 2′: Finding the all approximate ground states of a given $n\times m\times p$ supercell

1. Real-space GA versus reciprocal-space GA

Figure 8 reports the number of evaluations with respect to ϵ and the size of the configuration space required to find all *approximate* ground states of a given supercell. We mean by all approximate ground states that an approximate convex hull $C_{3\ \mathrm{meV}}(x)$ has been found such that $\forall x\in[0,1]$, $|C_{3\ \mathrm{meV}}(x)-C(x)|\leq 3$ meV, with $C(x)$ as the exact convex hull of the supercell. Panels (a) and (b) show that the reciprocal-space **k**-GA is more effective than the real-space **r**-GA. More specifically, panel (a) shows very similar behaviors for both methods, with **k**-GA translated down from **r**-GA. This difference may be assigned to the smaller population-size requirements of **k**-GA (see Table III), which generally implies a faster rate of evolution. Since the **k**-GA generally requires a smaller population, it is generally faster across all $2\times2\times n$ supercells.

2. Darwinian evolution versus Lamarckian evolution

We find in panels (c) and (d) of Fig. 8 that the real-space bit-string Lamarckian evolution **r**-LGA is once again as effective only as the underlying VA minimization. As reported in panels (e) and (f), the reciprocal-space Lamarckian evolution **k**-LGA is the most effective search strategy to find all approximate ground states as well. Overall, searching for an approximate convex hull is at least 2 orders of magnitude faster than searching for the exact convex hull. The VA approach also performs remarkably well. In a sense, this may seem surprising since the answer to question 2′ must include a number of approximate ground-state configurations *across* the concentration range $x\in[0,1]$, and the VA approach can only find one per run. Since the VA approach does solve this question with a similar success rate as that of **k**-GA, it follows that the former approach is quite sensitive to the seed (e.g., to the particular starting point).

VII. SUMMARY

We have performed a systematic statistical study of genetic algorithm's ability to find ground states of binary A_xB_{1-x} alloys on a fixed Bravais lattice with $x\in[0,1]$. In contrast with systems such as valence compounds, the difficulty of finding the minimum-energy structure of a binary metallic alloy lies not solely in finding the correct lattice type but also in the many near-degenerate configurations which exist for a fixed lattice and unit cell. Indeed, such a problem becomes combinatorial and has been shown to be NP hard.[33]

TABLE II. Number of evaluations required to find the exact deepest ground state with 95% degree of confidence using real-space Darwinian GA with mutation (first column) and without mutations (second column). The mutations vastly enhance the efficiency of the search by introducing new traits within the population. It allows the search procedure to explore a larger region of space by providing diversity and reducing its propensity to saturate.

Supercell	Without mutations	With mutations
$2\times2\times3$	580	150
$2\times2\times4$	350	230
$2\times2\times5$	10000	910
$2\times2\times6$	5600	990
$2\times2\times7$	94000	2100
$2\times2\times8$	60000	2100

Furthermore, the complete answer to a ground-state search problem contains in general a number of configurations across the concentration range $x\in[0,1]$. We have shown that combining local refinements with a reciprocal-space mating scheme and using as an objective function the distance to the known convex hull yields a search procedure which efficiently solves ground-state problems. Nonetheless, finding all exact ground-state configurations of an A_xB_{1-x} alloy of even relatively small systems soon becomes intractable ($\sim$330 assessed configurations for $N=12$, $\sim$2800 for $N=24$, and up to $\sim$39 000 for $N=36$) without the use of "indirect" first-principles methods such as cluster expansion. However, this same problem can be solved approximately (within 3 meV) at a much smaller cost in functional evaluations ($\sim$30 assessed configurations for $N=12$, $\sim$380 for $N=24$, and $\sim$580 for $N=36$). Finally, the same search procedure is also quite equal to solving single-valued searches such as the exact deepest ground-state search presented in the paper.

ACKNOWLEDGMENTS

This work was funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Contract No. DE-AC36-99GO10337 for NREL within NREL's Laboratory Directed Research and Development Program. We further acknowledge the use of the Evolutionary Object library[56] and the Alloy Theoretical Automated Toolkit.[57]

APPENDIX A: INFLUENCE OF THE GA PARAMETERS ON THE GENETIC ALGORITHM

All the genetic algorithm results presented in this paper were optimized for efficiency with respect to the population size, the mutation vs crossover frequency, and the number of mutated alleles per mutation operation. These three parameters strongly affect the expectation for success. We find that the number of offsprings per generation does not affect the efficiency of the algorithm quite as strongly. It has been kept at 10% throughout. In general, we find that Lamarckian evolution requires much smaller variations in the population

TABLE III. Population sizes used to obtain the data in Figs. 5–8. Larger populations will be required to explore complex space more fully. As such, the population roughly increases with the size of the supercell. Indeed, from Figs. 5–8 one can see that $2\times2\times n$ supercells with odd n are more difficult to solve than those with even n. This is reflected in the variation of the population sizes.

Method	**r**-GA/M	**k**-GA	**r**-LGA	**k**-LGA	**r**-GA/M	**k**-GA	**r**-LGA	**k**-LGA
		Question 1			Question 1′			
$2\times2\times3$	30	10	5	5	10	10	5	5
$2\times2\times4$	5	20	5	5	5	20	5	5
$2\times2\times5$	500	100	5	40	15	20	5	10
$2\times2\times6$	200	100	5	10	5	20	5	10
$2\times2\times7$	2000	200	20	100	15	40	5	10
$2\times2\times8$	1500	150	50	60	15	40	5	5
$2\times2\times9$	6000	300		200	15	30		5
$2\times2\times10$		200		80	15	50		10
$2\times2\times11$		400		200		40		5
$2\times2\times12$		350		120	25	50		20
		Question 2			Question 2′			
$2\times2\times3$	50	40	10	20	10	10	10	5
$2\times2\times4$	20	60	30	20	20	20	5	5
$2\times2\times5$	200	700	30	40	100	20	5	20
$2\times2\times6$	200	800	60	30	100	20	20	20
$2\times2\times7$	300	1200	50	80	100	40	30	10
$2\times2\times8$	2000	700	30	30	100	60	30	10
$2\times2\times9$	3000	450		100	100	100	50	10

size. Furthermore, the reciprocal-space mating incorporates mutations and crossover in one single operation, without the need for extra parameters. As a result, the Lamarckian reciprocal-space GA is not only the most effective search strategy presented here, it is also the most black-box-like.

We report in Table I the required number of evaluations for answering question 2 with a degree of confidence of 95% using the reciprocal-space Lamarckian GA and with respect to different supercell sizes and populations. We find that population size can have a large effect. Indeed, taking, for instance, a $2\times2\times5$ supercell (third row), using a population size of 20 individuals will require 5500 evaluations, whereas increasing the size to 40 will obtain the answer in no more than 440 evaluations. When evolving small populations, the search may saturate (despite the diversity constraint), e.g., a not-quite-optimal region of the configuration space is found from which the search cannot easily escape because all the individuals in the population contain the same traits from that region. In that case, the genetic algorithm is stuck until a sufficiently favorable mutation happens along. If one increases the population size too far, then the evolving the population will take longer. As such, there exists a population size which is a compromise between the possibility of saturation and the speed of evolution. It is overall less detrimental to operate with a larger population than to risk saturation.

Table II reports the number of evaluations required to solve question 1 with 95% degree of confidence when using bit-string Darwinian GA with and without mutations. The mutation parameters have been optimized. We find that mutations do have a very large impact on the efficiency of the real-space **r**-GA search. Indeed, the mutation operators generally allow the search to explore a larger manifold by introducing new traits into the population. Although not shown here, we have also added real-space mutations to the reciprocal-space **k**-GA search (e.g., in addition to the intrinsic mutations present within reciprocal-space mating). We find that this addition to **k**-GA do not accelerate the search. It would seem that the intrinsic mutations of the reciprocal-space mating are sufficient. As such, the reciprocal-space GAs are more black box than the real-space GA.

Table III gives the population sizes used for each method in Figs. 5–8. The population roughly increases with the size of the supercell, e.g., with the complexity of the explored configuration space. Indeed, as evidenced in Figs. 5–8, $2\times2\times n$ supercells with odd n are more difficult to solve for than those with even n and thus require larger populations. Offsprings are obtained with real-space bit-string mating by applying either a cross-over or a mutation operation, with a probability of 0.2–0.5 for the former. When a mutation operation is applied, each allele has 2–5% chance of being transmuted.

Statistics are collected over long independent evolutionary runs where populations are allowed to evolve until the results have been found with 95% degree of confidence. Practitioners may wonder whether it may not be more efficient to performing several shorter runs with a smaller expectation for success. We show here that, excluding the possibility of saturation discussed above, it is better to let the evolutionary process run its course. Let $\epsilon_0(N)$ be the expectation for success with respect to N, the number of evaluations for a single (statistical) evolutionary run. Then, on average, performing n evolutionary runs would yield a success rate of $\epsilon_n(N)=1-[1-\epsilon_0(N/n)]^n$. This amounts to two rescalings, $N\mapsto\frac{N}{n}$ and $1-\epsilon\mapsto(1-\epsilon)^n$. One can easily convince oneself that the search strategies presented above will become more expensive when using small repetitive evolutionary runs. Indeed, this result is true for any algorithm for which the number of evaluation increases exponentially with the requested degree of confidence, as is the case when solving NP-complete problems such as a configuration space search.

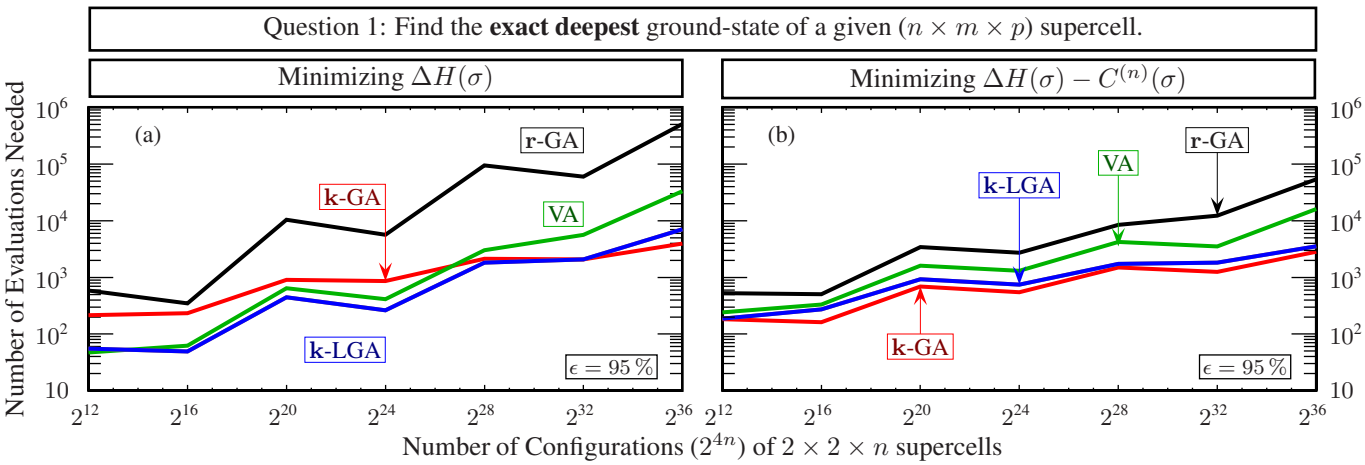

FIG. 9. (Color online) See caption of Fig. 5. The number of evaluations required to find the exact deepest ground states is plotted with respect to the configuration space size for $2\times2\times n$ supercells, when (a) minimizing the formation enthalpy $\Delta H(\sigma)$ and (b) when minimizing the distance to the known convex hull (see Sec. II B) at generation n of the genetic algorithm, $\Delta H(\sigma)-C^{(n)}(\sigma)$. Note that (i) both Darwinian GAs find the answer faster in (b) than in (a) across all supercells and (ii) both Lamarckian GAs and VA find the answer faster in (b) for supercells larger than $n=8$. Although the depth to the known convex hull changes during an evolutionary run, it is nonetheless an easier quantity to minimize than the formation enthalpy alone.

APPENDIX B: SUBTRACTING THE CONVEX HULL GIVES SMOOTH CONFIGURATION SPACE

One general question regarding the strategy we have chosen for all ground states, namely, to minimize $\Delta H(\sigma) - C(x_\sigma)$ rather than $\Delta H(\sigma)$ at fixed concentration x_σ, is that it constitutes a moving target, and thus may actually be more difficult to optimize. We find however that (i) a fixed concentration is a cumbersome constraint during mating operations, especially with the reciprocal-space mating and the Lamarckian refinements described in the paper, and (ii) it requires for N-atom configurations N independent GA minimization. Furthermore, there is evidence that removing the convex hull actually makes the space less complex. Figure 9 plots the required number of evolutions for finding the exact deepest ground state with respect to $2\times2\times n$ supercells for all five search strategies presented here. In panel (a) of Fig. 9, the objective function minimized to solve question 1 is, as mentioned earlier in the text, the formation enthalpy $\Delta H(\sigma)$. On the other hand, in panel (b) the objective function becomes the depth to the known convex hull at iteration n of the genetic algorithm, e.g., the objective function used previously to find *all* ground states, $\delta^{(n)}(\sigma)=\Delta H(\sigma)-C^{(n)}(x_\sigma)$. Remarkably, although this last objective is more complex and, in fact, changes during the course of the evolutionary run, both Darwinian GAs converge faster in panel (b), and both Lamarckian GAs and the virtual-atom approach converge faster in panel (b) for supercells larger than $2\times2\times8$.

*mayeul_davezac@nrel.gov
†alex_zunger@nrel.gov

[1] A. F. Wells, *Structural Inorganic Chemistry* (Oxford University, New York, 1984).
[2] *The Nature of the Chemical Bond*, edited by L. Pauling (Cornell University Press, Ithaca, 1960).
[3] W. Hume-Rothery, R. E. Smallman, and C. W. Haworth, *The Structure of Metals and Alloys* (The Institute of Metals, London, 1969).
[4] *Pearson's Handbook of Crystallographic Data for Intermetallic Phases*, edited by P. Villars and L. D. Calvert (American Society for Metals, Metal Park, OH, 1985).
[5] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods* (University Press, Cambridge, 2004).
[6] M. T. Yin and M. L. Cohen, Phys. Rev. Lett. **50**, 2006 (1983).
[7] S. Froyen and M. L. Cohen, Phys. Rev. B **28**, 3258 (1983).
[8] A. Mujica, A. Rubio, A. Muñoz, and R. J. Needs, Rev. Mod. Phys. **75**, 863 (2003).
[9] M. Wang, X. Hu, D. N. Beratan, and W. Yang, J. Am. Chem. Soc. **128**, 3228 (2006).
[10] S. Keinan, X. Hu, D. N. Beratan, and W. Yang, J. Phys. Chem. A **111**, 146 (2006).
[11] G. Trimarchi, P. Graf, and A. Zunger, Phys. Rev. B **74**, 014204 (2006).
[12] F. Ducastelle, *Order and Phase Stability in Alloys (Cohesion and Structure)* (North-Holland, Amsterdam, 1991).
[13] J. W. D. Connolly and A. R. Williams, Phys. Rev. B **27**, 5169 (1983).
[14] L. G. Ferreira, S.-H. Wei, and A. Zunger, Int. J. Supercomput. Appl. **5**, 34 (1991).
[15] A. Zunger, Proceedings of the NATO ASI on "Statistics and Dynamics of Alloy Phase Transformation (Plenum, New York, 1993), pp. 361–419.
[16] W. Lenz, Phys. Z. **21**, 613 (1920).
[17] E. Ising, Phys. Z. **31**, 253 (1925).
[18] W. L. Bragg and E. J. Williams, Proc. R. Soc. London, Ser. A **145**, 699 (1934).
[19] D. M. Deaven and K. M. Ho, Phys. Rev. Lett. **75**, 288 (1995).
[20] B. Hartke, J. Phys. Chem. **97**, 9973 (1993).
[21] A. R. Oganov and C. W. Glass, J. Chem. Phys. **124**, 244704 (2006).
[22] D. P. Stucke and V. H. Crespi, Nano Lett. **3**, 1183 (2003).
[23] N. L. Abraham and M. I. J. Probert, Phys. Rev. B **73**, 224104 (2006).
[24] G. Trimarchi and A. Zunger, Phys. Rev. B **75**, 104113 (2007).
[25] J. Kanamori and Y. Kakehashi, J. Phys. (Paris), Colloq. **38**, C7 (1977).
[26] M. K. Phani, J. L. Lebowitz, and M. H. Kalos, Phys. Rev. B **21**, 4027 (1980).
[27] A. Finel and F. Ducastelle, Europhys. Lett. **1**, 135 (1986).
[28] S. V. Barabash, V. Blum, S. Müller, and A. Zunger, Phys. Rev. B **74**, 035108 (2006).
[29] D. B. Laks, L. G. Ferreira, S. Froyen, and A. Zunger, Phys. Rev. B **46**, 12587 (1992).
[30] V. Blum and A. Zunger, Phys. Rev. B **72**, 020104(R) (2005).
[31] A. Zunger and S. Mahajan, *Handbook on Semiconductors* (Elsevier, Amsterdam, 1994), Vol. 3, p. 1399.
[32] M. Sanati, L. G. Wang, and A. Zunger, Phys. Rev. Lett. **90**, 045502 (2003).
[33] F. Barahona, J. Phys. A **15**, 3241 (1982).
[34] S. Istrail, Proceedings of the 32nd ACM Symposium on the Theory of Computing (STOC00) (ACM, Portland, OR, 2000), pp. 87–96.
[35] L. T. Wille and J. Vennik, J. Phys. A **18**, L419 (1985).
[36] A. B. Adib, J. Phys. A **38**, 8487 (2005).
[37] J. Crain, G. J. Ackland, and S. J. Clark, Rep. Prog. Phys. **58**, 705 (1995).
[38] C. W. Glass, A. R. Oganov, and N. Hansen, Comput. Phys. Commun. **175**, 713 (2006).
[39] M. d'Avezac and A. Zunger, J. Phys.: Condens. Matter **19**, 402201 (2007).
[40] The search presented in Ref. 39 and in this paper considers type (ii) problems, i.e., *decoration* of an N-atom unit cell on a *fixed* underlying Bravais lattice of $A_{1-x}B_x$, simultaneously throughout the x composition range, whereas the search in Ref. 21 is a type (iii) problem; i.e., it optimizes the lattice type, unit cell, and atomic positions (and hence the lattice decoration) of an A_nB_m structure at fixed stoichiometry n and m. In both methods, structures are relaxed with respect to strain.
[41] N. A. Barricelli, Methodos **1954**, 45.
[42] C. Darwin, *The Origin of Species* (Murray, London, 1859).
[43] J.-B. Lamarck, *Philosophie Zoologique ou Exposition des Con-*

sidérations Relatives à l'Histoire Naturelle des Animaux (Dentu, Paris, 1809).

[44] Gradients of the free energy with respect to the fractional occupation of different sites occur in the concentration-wave theory of ordering (Refs. 58–61). These gradients allow the determination of phase precursors to the partially ordered structure within the random alloy. Nonetheless, the concentration-wave theory (CWT) (Ref. 61) differs in the following ways from the work presented herein: (i) The goal of CWT is to identify precursors of the ground-states within the random-alloy phase, whereas the goal of the virtual-atom approach is to identify the ground states themselves. (ii) CWT takes the derivatives of the free energy of the random alloy, whereas the virtual-atom approach take the derivatives of the total energy of an actual structure. (iii) The site-dependent concentrations introduced in CWT have the physical meaning of a thermodynamic average, whereas here fractional occupations are a mathematical proxy with no physical meaning, introduced only to help obtain the minimum-energy structures.

[45] Z. W. Lu, S.-H. Wei, A. Zunger, S. Frota-Pessoa, and L. G. Ferreira, Phys. Rev. B **44**, 512 (1991).

[46] P. Piquini, A. Zunger, and R. Magri, Phys. Rev. B **77**, 115314 (2008).

[47] L. A. Wolsey, *Integer Programming*, 1st ed. (Wiley, New York, 1998).

[48] M. P. Bendsøe and N. Kikuchi, Comput. Methods Appl. Mech. Eng. **71**, 197 (1988).

[49] N. Marzari, S. de Gironcoli, and S. Baroni, Phys. Rev. Lett. **72**, 4001 (1994).

[50] P. A. Graff and W. B. Jones, J. Comput. Phys. (to be published).

[51] R. L. Johnston, Dalton Trans. **2003**, 4193.

[52] J. Z. Liu, G. Trimarchi, and A. Zunger, Phys. Rev. Lett. **99**, 145501 (2007).

[53] A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **99**, 12562 (2002).

[54] R. Martoňák, A. Laio, and M. Parrinello, Phys. Rev. Lett. **90**, 075503 (2003).

[55] G. L. W. Hart and A. Zunger, Phys. Rev. Lett. **87**, 275508 (2001).

[56] Maarten Keijzer, J. J. Merelo, G. Romero, and M. Schoenauer, Evolving Objects: a general purpose evolutionary computation library. In EA-01, Evolution Artificielle, 5th International Conference in Evolutionary Algorithms, 2001.

[57] A. van de Walle and M. Asta, Modell. Simul. Mater. Sci. Eng. **10**, 521 (2002).

[58] D. de Fontaine, *Sold State Physics* (Academic, New York, 1979), Vol. 34, p 73.

[59] *Theory of Structural Transformations in Solids*, edited by A. G. Khachaturyan (Wiley, New-York, 1983).

[60] R. McCormack, M. Asta, J. J. Hoyt, B. C. Chakoumakos, S. T. Misture, J. D. Althoff, and D. D. Johnson, Comput. Mater. Sci. **8**, 39 (1997).

[61] J. D. Althoff and D. D. Johnson, Comput. Mater. Sci. **8**, 71 (1997).