A gradient-directed Monte Carlo method for global optimization in a discrete space: Application to protein sequence design and folding

Xiangqian Hu, David N. Beratan,^{a)} and Weitao Yang^{a)}

Department of Chemistry, Duke University, Durham, North Carolina 27708-0354, USA

(Received 8 December 2008; accepted 2 September 2009; published online 21 October 2009)

We apply the gradient-directed Monte Carlo (GDMC) method to select optimal members of a discrete space, the space of chemically viable proteins described by a model Hamiltonian. In contrast to conventional Monte Carlo approaches, our GDMC method uses local property gradients with respect to chemical variables that have discrete values in the actual systems, e.g., residue types in a protein sequence. The local property gradients are obtained from the interpolation of discrete property values, following the linear combination of atomic potentials scheme developed recently [M. Wang *et al.*, *J. Am. Chem. Soc.* **128**, 3228 (2006)]. The local property derivative information directs the search toward the global minima while the Metropolis criterion incorporated in the method overcomes barriers between local minima. Using the simple HP lattice model, we apply the GDMC method to protein sequence design and folding. The GDMC algorithm proves to be particularly efficient, suggesting that this strategy can be extended to other discrete optimization problems in addition to inverse molecular design. © 2009 American Institute of Physics.
[doi:[10.1063/1.3236834](https://doi.org/10.1063/1.3236834)]

I. INTRODUCTION

Global optimization is a key challenge in many research areas.¹ In chemistry, optimization is an essential part of structure prediction, protein folding, and ligand design.² In these cases, the number of molecular conformations increases very rapidly with the number of atoms in a molecule, so determining the global energy minimum structure is non-trivial. Global optimization methods can be divided into deterministic and stochastic classes. Deterministic optimizations have no randomness associated with the optimization algorithms. In contrast, stochastic approaches incorporate probabilistic (random) elements during optimization.³ The problems to which the optimization methods are applied can be categorized in three classes: discrete problems, continuous problems, and mixed-integer problems. Here, we focus on discrete problems. For discrete variables, stochastic methods can be implemented and have been used widely in global optimization. Two popular stochastic approaches are simulated annealing (SA) (Ref. 4) and genetic algorithms (GAs).⁵ Both SA and GAs, however, ignore local information characteristic of the discrete space of interest. A study of Oda *et al.*⁶ showed that the ability of a discrete algorithm to find the global minimum is significantly improved when SA or GA is combined with local searching for Ising spin clusters. For instance, in the hybrid GA incorporated with one-neighborhood search,⁶ each spin from the current generation structure is flipped by single-point mutations, and the fittest individual is selected for further GA operations (such as crossover and selection). Their study suggests that a proper

scheme of normal stochastic global optimization combined with some local searching can reduce the cost of stochastic global searches.

Instead of stochastic approaches that operate directly on discrete variable spaces, several approaches have been proposed to embed discrete spaces in virtual continuous spaces.^{7–10} This strategy has proven to be especially effective in problems of molecular design. In particular, a linear combination of atomic potential (LCAP)^{9,11} approach was developed and used to optimize nonlinear optical (NLO) properties by varying electrostatic potentials, expressed as a linear combination of atomic electrostatic potentials. The optimal structures can be determined without enumerating species or by performing stochastic searches. The LCAP study indicates that it is possible to treat discrete variables as continuous, and then to use local derivative information for the target property to perform continuous optimization.

From our experience with the LCAP approach, the complexity of the continuous virtual property surface depends on the property of interest and the specific library of chemical systems. However, we have found that LCAP gradients assist in guiding local searches in discrete space.^{12,13} Specifically, we showed that the LCAP approach combined with the new gradient-directed Monte Carlo (GDMC) method efficiently optimizes NLO properties in a class of donor-acceptor substituted benzene and porphyrin motifs.¹² Here, we use this GDMC approach for global optimization in discrete spaces, combining derivative information from the continuous interpolation with stochastic Monte Carlo (MC) moves. Local gradients in GDMC are constructed simply by treating the discrete variables as continuous variables. The derivatives then bias the MC moves during the optimization in discrete space. As such, the local searches, with the gradients derived from the virtual continuous surface, are used to enhance the

^{a)}Electronic addresses: weitao.yang@duke.edu and david.beratan@duke.edu.

local searching abilities of SA or GA without requiring extensive exploration of intermediate virtual structures.

In this article, we describe the mathematical background and general procedure for GDMC optimization in Sec. II. GDMC is applied to protein sequence design and protein folding for a simple Hamiltonian in Sec. III. For design and folding, the schemes for obtaining the local gradients are explained. The details of the GDMC procedures in both cases are also described. The results are summarized in Sec. IV.

II. THE GENERAL GDMC METHOD

The GDMC approach described previously¹² can be modified to establish a general tool for global optimization in a discrete space. Mathematically, the constrained discrete global optimization explored here takes the following form:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } h(x) = 0 \quad \text{for } x = x_1, x_2, \dots, x_n, \\ &g(x) \leq 0 \quad \text{for } x = x_1, x_2, \dots, x_n, \end{aligned} \quad (1)$$

where $f(x)$ is an object function, $h(x)$ is a set of equality constraints, $g(x)$ is a set of inequality constraints, and n is the number of discrete variables. Each element in $\{x_i; i=1, \dots, n\}$ is a discrete variable, which means that each element can only take on discrete values. However, we treat each element $\{x_i; i=1, \dots, n\}$ as being a continuous variable for the purpose of obtaining the gradients $\{\partial f / \partial x_i; i=1, \dots, n\}$. Then, using the steepest descent method,¹⁴ a new set of $\{x'_i; i=1, \dots, n\}$ is constructed as

$$x'_i = x_i - \varepsilon \frac{\partial f}{\partial x_i} (\varepsilon > 0), \quad (2)$$

where ε is a step-size parameter. Equation (2) implies that each element in $\{x_i; i=1, \dots, n\}$ will increase or decrease depending on the signs of the gradients $\{\partial f / \partial x_i; i=1, \dots, n\}$. Thus, these gradients can provide useful information regarding the discrete variables to guide the next generation of $\{x'_i; i=1, \dots, n\}$.

Once the gradients $\{\partial f / \partial x_i; i=1, \dots, n\}$ are computed, the general GDMC procedure for the optimization defined in Eq. (1) is as follows.

- Set the iteration index $i=0$ and begin with an initial set of $\{x_k^1; k=1, \dots, n\}$ satisfying the constraints of Eq. (1).
- Set $i=i+1$. Calculate $f_i=f(x^i)$ using $\{x_k^i; k=1, \dots, n\}$ and the gradients $\{\partial f_i / \partial x_k^i; k=1, \dots, n\}$ with respect to all coefficients $\{x_k^i; k=1, \dots, n\}$. Exit if the optimization goal or the maximum number of iterations is reached.
- Make a trial move to generate a new set of $\{x_k^{i+1}; k=1, \dots, n\}$ based on specific schemes following the computed gradients $\{\partial f_i / \partial x_k^i; k=1, \dots, n\}$ with Eq. (2). Accept the trial move with probability $p = \min\{1, e^{-\beta(f_{i+1}-f_i)}\}$ using the Metropolis criterion. If the trial move is accepted, go to step (b); if the trial move is rejected, go to step (c). (Here, β is an empirical parameter.)

The specific schemes in this procedure use the computed gradients to generate a trial move. These schemes need not be based on the steepest decent direction. In addition, the schemes depend on the specific optimization objects. For instance, such a scheme using the gradients to generate a trial protein sequence will be described in Sec. III A for protein sequence design.

The main difference between the GDMC scheme and the classical MC approach is that property gradients are obtained first from a virtual continuous surface in GDMC and used to generate the next set of $\{x_i; i=1, \dots, n\}$ at each optimization step. A key issue explored here is how to construct such a continuous surface from a discrete one so that the derivative information is utilized to direct the selection of the next generation of discrete variables. For example, as implemented in the LCAP approach, linearly combining atomic potentials on each variable site provides a simple approach to link all of the discrete molecules and generate a continuous surface.⁹ In the current study, for protein sequence design, the change from hydrophobic (H) to polar (P) residues is taken as continuous. Then, analytical gradients on the virtual structure surface can be calculated subject to constraints. Nevertheless, for protein folding, there is no effective way to calculate gradients analytically because of the strict structural constraint that each structure in the optimization should be physically meaningful. Therefore, “numerical gradients,” as described below, which satisfy the structural constraints, are calculated and used in GDMC.

III. PROTEIN SEQUENCE DESIGN AND FOLDING WITH LATTICE MODELS

Protein sequence design and protein folding represent central challenges in biology.^{15,16} One challenge is to determine the sequence with the lowest energy for a prescribed three-dimensional (3D) fold (the inverse folding problem). The second challenge is to determine the most stable fold for a specific sequence (the folding problem). Here, to explore these two challenges with the GDMC algorithm, we describe the protein with a simple HP lattice model.^{17–19} In the model, two types of residues can exist on fixed two-dimensional (2D) or 3D lattice sites. The residues at each site may be hydrophobic (H) or polar (P) and the lattice sites can be occupied by only one monomer. A protein chain is configured as a self-avoiding walk (SAW), i.e., a path from one point to another that never crosses itself. A 2D lattice model with 16 lattice sites is shown in Fig. 1.^{7,10} Although the simplicity of HP lattice models limits the model somewhat,^{20–22} protein sequence design and folding with this HP lattice model are nondeterministic polynomial hard problems.^{23,24} Therefore, the simple HP lattice model is often used to benchmark global optimization methods. Here, we focus on the 2D lattice HP models to explore the efficiency of the GDMC optimization for protein sequence design and folding.

In the HP lattice model, the total energy depends on the sum of interactions between all pairs of adjacent “non-bonded” residues in the conformation. For example, in Fig. 1, site 3 has nonbonded interactions with sites 12 and 16 (but not with sites 2 and 4). The interaction energy e depends on

FIG. 1. 4×4 2D HP lattice conformation. Each site is populated by either an H or P monomer. This specific conformation is used in Sect. III A for protein sequence design.

interacting residue types. Note that we do not address the negative design issue (i.e., design with additional constraints).^{25,26} Here, we set $e_{HH}=-2.3$, $e_{HP}=1.0$, and $e_{PP}=0$ (Ref. 7) for protein sequence design; for protein folding, only H-H interactions are nonzero ($e_{HH}=-1.0$).²⁷ The above values were used to compare our GDMC optimizations with corresponding methods in the literature.

The total HP energy⁷ is

$$E = 0.5 \left[\sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N A(i,j) e(S_i, S_j) \right], \quad (3)$$

where

$$A(i,j) = \begin{cases} 1 & \text{if sites } i \text{ and } j \text{ interact} \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

$$e(S_i, S_j) = e_{HP} S_i (1 - S_j) + e_{HP} (1 - S_i) S_j + e_{HH} S_i S_j. \quad (5)$$

In Eq. (3), N is the number of residues and $A(i,j)$ is the interaction matrix for all residues. In Eq. (5), S represents the residue type on each site: $S_i=0$ corresponds to a P residue on the site; $S_i=1$ corresponds to a H residue on the site.

A. Protein sequence design

To design a protein sequence with a given fold, the total energy of a protein conformation is optimized with respect to the residue type (H or P) on each lattice site. Here, we choose the 2D lattice protein fold shown in Fig. 1.⁷ The set $\{S_i, i=1, \dots, N\}$ is optimized to design a protein sequence. In Ref. 7, the S_i value was replaced by a Gaussian function to construct a continuous energy surface, and then continuous optimization of $\{S_i, i=1, \dots, N\}$ was carried out. In our study, however, S_i is a continuous variable. Thus the energy gradient with respect to S_i is

$$B_i = \frac{\partial E}{\partial S_i} = 0.5 \sum_{k=1}^N \sum_{\substack{l=1 \\ l \neq k}}^N A(k,l) \frac{\partial e(S_k, S_l)}{\partial S_i}. \quad (6)$$

If there are no constraints on protein sequence in the HP model, Eq. (3) shows that the minimum energy is found when all sites are H residues, because e_{HH} has the lowest interaction energy. Following the protocol of Ref. 7, the number of H residues (N_H) is constrained for a nontrivial optimization to design a variety of protein sequences,

$$\sum_{i=1}^N S_i = N_H. \quad (7)$$

For this constrained minimization, a new object function of the constrained energy is constructed,

$$L = E + \Lambda \left[\sum_{i=1}^N S_i - N_H \right], \quad (8)$$

where the Lagrangian multiplier is

$$\Lambda = \frac{\left[\sum_{i=1}^N S_i - N_H \right] - \sum_i^N B_i}{N}. \quad (9)$$

Accordingly, the gradient of S_i in GDMC is

$$\frac{\partial L}{\partial S_i} = B_i + \Lambda. \quad (10)$$

After the object function and its gradients with respect to S_i are formulated, the GDMC optimization procedure for protein sequence design is as follows.

- Set the iteration number $i=0$ and begin with a random sequence $\{S_k^1: k=1, \dots, N\}$, S_k^1 is either 0 or 1, maintaining $\sum_k^N S_k^1 = N_H$.
- Set $i=i+1$. Calculate L_i and gradients $\partial L_i / \partial S_k^i$ using Eqs. (8) and (10). Exit if the optimization goal or the maximum number of iterations is reached. (Here, we chose 1000 iterations.)
- Generate a new set of $\{S_k^{i+1}: k=1, \dots, N\}$ using the computed gradients $\partial L_i / \partial S_k^i$. To maintain $\sum_k^N S_k^{i+1} = N_H$, the generation scheme for a trial protein sequence (see below for details) is used. Accept the new sequence with a probability $p = \min\{1, e^{-\beta(L_{i+1}-L_i)}\}$ using the Metropolis rule. (Here, the empirical parameter β is set to 1.2×10^{-3} based on trial runs.)

The generation scheme in the above procedure uses the current i th sequence and its gradients $\partial L_i / \partial S_k^i$ to jump to the next sequence, while satisfying the constraint on the total number of hydrophobic residues. The detailed rules used to generate a new sequence are as follows.

- Using the computed gradients $\{\partial L_i / \partial S_k^i: k=1, \dots, N\}$, all of the N lattice sites are placed in decreasing order based on their gradients. The sites with the N_H lowest gradients are set to be H residues and all others are set to be P residues. The constraint $\sum_k^N S_k^{i+1} = N_H$ is retained.
- If this generated sequence was visited in an earlier step, the site that has the (N_H+1) th lowest gradient is switched to an H residue. Meanwhile, the site that has the N_H th lowest gradient is switched to a P residue to satisfy the constraint $\sum_k^N S_k^{i+1} = N_H$. This switching pro-

TABLE I. Comparison of GDMC with MC performance to design protein sequences for the fixed conformation in Fig. 1. Five MC and GDMC runs were executed independently. N_H is the number of H residues. $E_{\min}$ is the global minimum energy found by enumerating all possible sequences. $E_{\min}^{\text{MC}}$ is the minimum energy obtained from the standard MC search. $E_{\min}^{\text{GDMC}}$ is the minimum energy from GDMC. The number in parentheses for $E_{\min}^{\text{MC}}$ and $E_{\min}^{\text{GDMC}}$ is the smallest number of steps needed to find the global minimum energy during five MC and GDMC runs. The information for the GDMC run, which takes the smallest number of steps to find the lowest energy for each N_H , is shown in the last three columns. “Iteration” is the number of steps in the GDMC optimization. “Answers” indicates how many degenerate sequences with the same lowest energy are found during the optimization. “Successful jumps” denotes how many sequences generated by the prior scheme lower the total energy.

N_H	$E_{\min}^a$	$E_{\min}^{\text{MC}}$	GDMC			
			$E_{\min}^{\text{GDMC}}$	Iteration	Answers	Successful jumps
1	-2.0	-2.0(2)	-2.0(2)	6	5	5
2	-4.3	-4.3(44)	-4.3(7)	7	1	4
3	-6.6	-6.6(72)	-6.6(9)	9	1	5
4	-8.6	-8.6(10)	-8.6(2)	6	4	3
5	-10.9	-10.9(286)	-10.9(2)	14	1	6
6	-12.2	-11.9(540)	-12.2(2)	17	4	7
7	-13.5	-13.5(185)	-13.5(2)	14	6	3
8	-14.8	-14.8(149)	-14.8(3)	32	4	11
9	-16.1	-15.8(53)	-16.1(7)	55	1	27
10	-17.1	-17.1(100)	-17.1(2)	46	8	22
11	-18.4	-18.4(494)	-18.4(2)	34	2	20
12	-19.4	-19.4(10)	-19.4(2)	14	8	11
13	-20.7	-20.7(59)	-20.7(2)	15	1	11
14	-20.7	-20.7(4)	-20.7(2)	16	2	12

^aTaken from Ref. 10.

cedure is repeated until the generated sequence has not been visited, or until we exit if a new sequence cannot be generated after N_H attempts.

In this protocol, the constraint of Eq. (7) is always enforced by the above scheme and the gradients $\{\partial L_i / \partial S_k^i : k=1, \dots, N\}$ are used to guide the MC search. In a sense, the GDMC search is a “semistochastic” algorithm. The local gradients generate the sequence deterministically while the Metropolis criterion stochastically accepts or rejects the sequence so generated.

In the following analysis, the conformation of Fig. 1 is examined in greater detail and the lowest energies obtained by GDMC search are shown in Table I. MC in this table refers to the values of $\{\partial L_i / \partial S_k^i : k=1, \dots, N\}$ being set to random numbers. No gradient information from Eq. (10) is used in conventional MC sampling. Both MC and GDMC are tested in five independent runs. Information from the GDMC run that takes the smallest number of steps to find the lowest energy for each N_H is shown in the last three columns of Table I.

Table I shows that the GDMC search always finds the global minimum energy that is known from exhaustive sequence enumeration.⁷ Several different protein sequences that share the same lowest possible energy are found efficiently. The deterministic continuous optimizations^{7,10} only find one “intermediate” sequence close to one local minimum during one trial run within thousands of steps. The total number of iterations in GDMC, in contrast, is typically less than 50. Furthermore, the successful jumps in GDMC, defined by how many generated sequences use the GDMC scheme to lower the total energy, usually constitute more

than 50% of the total number of iterations. Therefore, compared to the continuous model,^{7,10} our GDMC approach utilizes the smooth energy surface constructed by continuing the discrete variables $\{S_k : k=1, \dots, N\}$ and efficiently explores the real sequences in discrete space.

Table I shows that, although the conventional MC scheme is still robust in its ability to find the global minima for most protein sequence designs, MC fails when $N_H=6$ and $N_H=9$ because of the large number of possible sequences (i.e., $N! / N_H! (N - N_H)!)$ (Ref. 10) when N_H is between 6 and 9. However, using GDMC, all five independent runs for each N_H find the global energy minimum. To find the first global minimum, MC usually requires hundreds of steps (see the number in parentheses for $E_{\min}^{\text{MC}}$ in Table I), while the GDMC method requires fewer than ten steps (see the number in parentheses for $E_{\min}^{\text{GDMC}}$ in Table I). This result suggests that the constructed energy surface of $\{S_k : k=1, \dots, N\}$ is sufficiently smooth that only a few steps are required to reach the global energy minimum using GDMC.

To further examine GDMC performance, another complicated 6×6 2D HP lattice conformation was chosen to design the protein sequence with eight H residues of the 36. The total number of possible sequences with eight H residues is more than 3×10^7 . Beginning with one random initial guess, only 48 iterations were carried out in the GDMC procedure and four degenerate sequences, shown in Fig. 2, were obtained with an optimal energy of -18.1 .⁷ Note that the positions of five H residues are conserved on this 2D lattice, which might suggest that these five adjacent positions form a favorable hydrophobic core.

For protein sequence design, the GDMC approach is not

FIG. 2. The four degenerate sequences with the optimal energy equal to -18.1 were obtained and shown in this figure for a 6×6 2D HP lattice conformation with eight H residues out of 36. All the empty circles indicate that P residues are placed at those sites.

fully stochastic because the analytical gradient information is used to generate new sequences. The local gradients direct the global optimization in the discrete space to find an improved sequence and the Metropolis criterion helps GDMC avoid becoming trapped in local minima through the stochastic moves. The combination of local gradients and the Metropolis criterion enables GDMC to find the global energy minimum efficiently.

B. Protein folding

Protein folding with an HP lattice model causes the interaction matrix A in Eq. (3) to vary as a protein of fixed sequence folds. Thus, $A(i, j)$ is the variable to be optimized toward the lowest energy conformations. However, maintaining the protein structure during folding, a constraint on the A matrix must be imposed: The HP lattice chain must be configured to self-avoid on the lattice.²⁸ Analytical energy gradients with respect to the element $A(i, j)$ subject to the SAW constraint cannot be easily calculated. Thus, numerical gradients with respect to the conformational changes are used. Specifically, the numerical gradients are calculated from the energy changes when one residue is moved to another position along the four directions shown in Fig. 3. Accordingly, other sites should also be moved to maintain an allowed conformation. Here, we use a local move set (i.e., pull moves of Ref. 29) to fold the extended lattice chain in the HP lattice model. Pull moves are complete. That is, any pair of valid conformations is connected to all others by pull moves. Therefore, pull moves determine how other residues respond in 2D HP lattice models when one residue is moved in one direction, as defined in Ref. 29.

According to the pull move rules, four move directions (i.e., indices 1, 2, 3, and 4) are defined in Fig. 3. If the position in one direction (such as P , N , L , or M) is already occupied by the other residue, the move in that direction is not valid. If position P , N , L , or M is vacant, the moves to

FIG. 3. Four move directions (i.e., indices 1, 2, 3, and 4) of residue i . If position L is vacant, residue i can be moved to L . Then residue $(i-1)$ is moved to position C to sustain a valid conformation. If this does not complete the move, residue $(i-2)$ is moved to the position previously held by residue i , residue $(i-3)$ is moved to the position previously held by residue $(i-1)$, and so on, until the move is completed and a new valid conformation is obtained.

these positions are permitted. After one valid move, residue i has a neighbor that is either residue $(i-1)$ or $(i+1)$. The other residues from the $(i-1)$ th residue to the first residue in the chain need to make the necessary moves to sustain a valid protein conformation, according to some specific rules of moving.²⁹ For instance, if residue i is moved to the vacant position L , residue $(i-1)$ is then moved to position C to complete the move and to sustain a valid conformation (as required by the SAW constraint). If the conformation is not valid after this move, residue $(i-2)$ is moved to the position i previously held by residue i . If the conformation is still invalid, residue $(i-3)$ is then moved to the position previously held by residue $(i-1)$ and so on, until the move is completed and a new valid conformation is obtained. The first and last residues in the chain require special pull moves (see Ref. 29). Basically, the strategy of pull moves is that the minimal possible number of residues should change their positions when one residue is moved. Therefore, pull moves are local.

Numerical gradients can be computed for each pull move of any residue. If a new conformation is not valid from a move in one direction, the gradient for this direction is set to be a large positive value. Otherwise, the energy of this new conformation is obtained from Eq. (3). Subsequently, the gradient equals the energy difference between the new conformation and the one prior to the pull move. Hence, numerical gradients for any site under the SAW constraint are readily calculated.

The GDMC procedure in protein folding is as follows.

- Start from an extended structure with the fixed protein sequence. Set $i=1$ and $E_1=0$.
- Set $i=i+1$. Randomly choose a residue to move, i.e., the k th residue. Calculate numerical gradients with respect to four move directions shown in Fig. 3 for the k th residue. Exit if the maximum number of iterations is reached.
- Make a trial move for the k th residue following the

ten sequences. Even for the much longer sequences 9 and 10 in Table II, the GDMC approach with the local pull moves found two lower energies than the MC-pull moves. The two corresponding conformations match those in Refs. 29 and 30. For sequences 1, 3, and 5, GDMC requires more time steps to find the lowest energy than the MC-pull moves; for sequence 7, more trial runs in GDMC were required to find the native conformation compared to the required stochastic MC-pull moves. This result suggests that the landscape for protein folding in the HP model is more complicated than the landscape for protein sequence design. (The complexity of the landscape depends on both the specific protein sequence and the number of residues.) Sometimes, using the local gradient information causes “near-sighted” side effects for global optimization and may mislead the optimization. Eventually, however, the MC moves in GDMC eliminate some side effects of the local gradients and allow the GDMC optimization to escape the local minima. On the whole, GDMC is more likely to find deeper minima for protein folding with the HP lattice model than MC-pull moves without gradient information.

The efficiency of the GDMC search is comparable with that of the CI method.²⁷ CI is an efficient computational strategy^{31–35} to simulate and analyze protein folding for HP lattice models. As shown in Table III, for sequences 4, 5, and 8, GDMC has more hits than CI for searches of the same duration and five trial runs. Moreover, for these sequences, GDMC requires substantially fewer optimization cycles to reach the optimum compared to CI. For sequence 8, the conformation with the lowest energy was found after 61 233 cycles in GDMC compared to 461 099 cycles in CI. For sequences 6 and 7, GDMC requires more iteration time or more trial runs than CI. This is also caused by the near-sighted local gradients in directing the simulation toward the global minimum when the landscape of protein folding is complicated. The MC stochastic moves in GDMC assist in relieving the side effects of local gradients and finally locate the native conformation of sequences 6 and 7. Overall, GDMC is more efficient than CI for finding lower energy minima.

Pull moves are also used in the equienergy (EE) sampling approach of Ref. 36. Pull moves in EE are employed as a mutation operator to change a given state and the generated configuration is accepted or rejected based on the Metropolis–Hastings criterion. With the aid of pull moves, EE samples the entire phase space while considering the entropic contributions to the folding free energy, rather than the energy landscape in the traditional optimization approaches. In contrast, our GDMC approach performs pull moves to compute numerical gradients and the trial move along the direction with the steepest gradient is accepted or rejected by the Metropolis criterion. Here, since we focus on testing the GDMC approach for global optimization, our aim is to find the global energy minimum conformation in the HP lattice models with the fixed sequences.

For HP lattice model protein folding, the analytical energy gradients with respect to the interaction matrix elements are difficult to obtain because the conformation must obey the SAW condition. To implement the GDMC approach, nu-

merical gradients are calculated using the set of local pull moves. Our results demonstrated that GDMC, combined with those local gradients, more efficiently explores the conformation space compared to MC, GA, the MC-pull moves, and the CI method.

IV. CONCLUSIONS

We have described a general global optimization approach for discrete space, i.e., a GDMC approach. GDMC utilizes local gradient information derived for the virtual continuous surface. The Metropolis criterion helps GDMC escape local minima. To demonstrate the efficiency of GDMC, protein sequence design and folding problems were studied for 2D HP lattice models, and the performance of the GDMC procedure for each case was described.

In protein sequence design, the energy function depends on the residue type at each lattice site. Combined with the constraint that arises from the number of H residues in the sequence, the energy gradients with respect to continuously interchanging H and P on each site were calculated. Compared to the continuous optimization in Refs. 7 and 10 and the conventional MC, GDMC requires fewer calculations of real sequences in order to find the sequence with the lowest energy. In one trial run, multiple degenerate sequences with the same lowest energy were obtained.

In protein folding, the SAW constraint was imposed. Numerical gradients were computed using local pull moves. Because the landscape for protein folding is rugged, near-sighted local gradients in GDMC may trap the optimization in some local minima. GDMC jumps out of the local minima and finds the lowest energies reported in the literature. Our results indicate that GDMC explores the conformation surface more efficiently than the MC, GA, MC-pull moves, and CI methods.

In conclusion, combining gradient information with MC moves is efficient for finding the globally optimal protein sequences and folds for this HP lattice model. In particular, when the virtual continuous surface is smooth (e.g., protein sequence design), jumps between two sets of discrete variables guided by the local gradients are particularly efficient. Furthermore, it is important to note that this GDMC approach is general. The method can be expanded to any discrete optimization problem as long as the gradients can be constructed from a reasonably continuous description of discrete space. We also expect that the local gradients can be combined with a GA [i.e., a gradient-directed genetic algorithm (GDGA)] to guide the construction of the offspring generations from the parents in a GDGA.

ACKNOWLEDGMENTS

Support from the University of Pittsburgh Center for Chemical Methodologies & Library Development is gratefully acknowledged (Grant No. 2P50GM067082). D.N.B. thanks the Keck and NIH Foundations for support of computational infrastructure. W.Y. acknowledges partial support

from the National Science Foundation. We thank B. Christopher Rinderspacher, Hao Hu, and Bruce Donald for helpful discussions.

- ¹R. Horst, P. Pardalos, and N. Thoai, *Introduction to Global Optimization*, 2nd ed. (Kluwer, Dordrecht, 2000).
- ²C. A. Floudas, *Deterministic Global Optimization* (Kluwer, Dordrecht, 2000).
- ³J. C. Spall, *Introduction to Stochastic Search and Optimization* (Wiley-Interscience, New York, NY, 2003).
- ⁴S. Kirkpatrick, C. Gelatt, and M. Vecchi, *Science* **220**, 671 (1983).
- ⁵D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA, 1989).
- ⁶A. Oda, H. Nagao, Y. Kitagawa, Y. Shigeta, M. Shoji, H. Nitta, M. Okumura, and K. Yamaguchi, *Int. J. Quantum Chem.* **105**, 645 (2005).
- ⁷S. K. Koh and G. K. Ananthasuresh, *Int. J. Robot. Res.* **24**, 109 (2005).
- ⁸O. A. von Lilienfeld, R. D. Lins, and U. Rothlisberger, *Phys. Rev. Lett.* **95**, 153002 (2005).
- ⁹M. Wang, X. Hu, D. N. Beratan, and W. Yang, *J. Am. Chem. Soc.* **128**, 3228 (2006).
- ¹⁰S. K. Koh, G. K. Ananthasuresh, and C. Croke, *J. Mech. Des.* **127**, 728 (2005).
- ¹¹D. Xiao, W. Yang, and D. N. Beratan, *J. Chem. Phys.* **129**, 044106 (2008).
- ¹²X. Hu, D. N. Beratan, and W. Yang, *J. Chem. Phys.* **129**, 064102 (2008).
- ¹³S. Keinan, X. Hu, D. N. Beratan, and W. Yang, *J. Phys. Chem. A* **111**, 176 (2007).
- ¹⁴J. A. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms* (Springer, New York, 2005).
- ¹⁵B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walters, *Molecular Biology of the Cell* (Garland Science, New York and London, 2002).
- ¹⁶C. Anfinsen, *Biochem. J.* **128**, 737 (1972).
- ¹⁷K. Lau and K. Dill, *Macromolecules* **22**, 3986 (1989).
- ¹⁸H. Chan and K. Dill, *J. Chem. Phys.* **95**, 3775 (1991).
- ¹⁹K. Dill, S. Bromberg, K. Yue, K. Fiebig, D. Yee, P. Thomas, and H. Chan, *Protein Sci.* **4**, 561 (1995).
- ²⁰H. S. Chan, *Proteins* **40**, 543 (2000).
- ²¹H. S. Chan, S. Shimizu, and H. Kaya, *Methods Enzymol.* **380**, 350 (2004).
- ²²H. Kaya and H. S. Chan, *Phys. Rev. Lett.* **85**, 4823 (2000).
- ²³B. Berger and T. Leighton, *J. Comput. Biol.* **5**, 27 (1998).
- ²⁴P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis, *J. Comput. Biol.* **5**, 423 (1998).
- ²⁵K. Yue and K. A. Dill, *Proc. Natl. Acad. Sci. U.S.A.* **89**, 4163 (1992).
- ²⁶G. L. Butterfoss and B. Kuhlman, *Annu. Rev. Biophys. Biomol. Struct.* **35**, 49 (2006).
- ²⁷L. Toma and S. Toma, *Protein Sci.* **5**, 147 (1996).
- ²⁸C. Domb, *Adv. Chem. Phys.* **15**, 229 (1969).
- ²⁹N. Lesh, M. Mitzenmacher, and S. Whitesides, Annual Conference on Research in Computational Molecular Biology, 2003 (unpublished), pp. 188–195.
- ³⁰H. P. Hsu, V. Mehra, W. Nadler, and P. Grassberger, *J. Chem. Phys.* **118**, 444 (2003).
- ³¹R. Unger and J. Moult, *J. Mol. Biol.* **231**, 75 (1993).
- ³²T. C. Beutler and K. A. Dill, *Protein Sci.* **5**, 2037 (1996).
- ³³G. Chikenji, M. Kikuchi, and Y. Iba, *Phys. Rev. Lett.* **83**, 1886 (1999).
- ³⁴F. Liang and W. H. Wong, *J. Chem. Phys.* **115**, 3374 (2001).
- ³⁵W. Huang, Z. Lu, and H. Shi, *Phys. Rev. E* **72**, 016704 (2005).
- ³⁶S. C. Kou, J. Oh, and W. H. Wong, *J. Chem. Phys.* **124**, 244903 (2006).