THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

Open Access
This article is licensed under CC-BY 4.0

pubs.acs.org/JPCA
Article

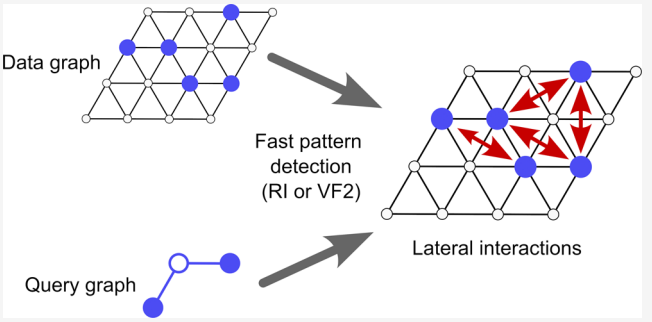
Speeding up the Detection of Adsorbate Lateral Interactions in Graph-Theoretical Kinetic Monte Carlo Simulations

Raz L. Benson, Sai Sharath Yadavalli, and Michail Stamatakis*

Cite This: J. Phys. Chem. A 2023, 127, 10307−10319
Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Kinetic Monte Carlo (KMC) has become an indispensable tool in heterogeneous catalyst discovery, but realistic simulations remain computationally demanding on account of the need to capture complex and long-range lateral interactions between adsorbates. The *Zacros* software package (https://zacros.org) adopts a graph-theoretical cluster expansion (CE) framework that allows such interactions to be computed with a high degree of generality and fidelity. This involves solving a series of subgraph isomorphism problems in order to identify relevant interaction patterns in the lattice. In an effort to reduce the computational burden, we have adapted two well-known subgraph isomorphism algorithms, namely, VF2 and RI, for use in KMC simulations and implemented them in *Zacros*. To benchmark their performance, we simulate a previously established model of catalytic NO oxidation, treating the O* lateral interactions with a series of progressively larger CEs. For CEs with long-range interactions, VF2 and RI are found to provide impressive speedups relative to simpler algorithms. RI performs best, giving speedups reaching more than 150× when combined with OpenMP parallelization. We also simulate a recently developed methane cracking model, showing that RI offers significant improvements in performance at high surface coverages.



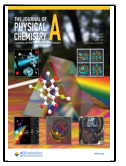
INTRODUCTION

Kinetic modeling techniques play a pivotal role in the mission to engineer new, more sustainable materials for heterogeneous catalysis.[1−6] They complement experimental studies by facilitating a detailed understanding of surface reaction mechanisms and accurate predictions of the catalyst activity and selectivity. The "gold standard" of modern kinetic modeling is kinetic Monte Carlo (KMC), a stochastic simulation method that treats elementary chemical processes as discrete events.[7−16]

Formally, an ensemble of KMC trajectories constitutes a numerical solution to the Markovian master equation for the system, which expresses its time evolution in terms of the rates of transitions between a discrete set of memoryless "states".[8] In the "on-lattice" variations of KMC used to study catalytic surfaces, each state typically corresponds to one configuration of adsorbates on the two-dimensional lattice.[9] The memorylessness arises from the assumption that enough time passes between successive events for the current and past configurations to decorrelate. The transitions are the elementary processes, which include adsorptions, desorptions, diffusional hops, and single-step chemical reactions, with rate constants typically estimated using transition state theory (TST).[10] It is important to recognize that the microscopic TST parameters, which may be determined using density functional theory (DFT) calculations, serve as input to the KMC simulation.

A number of different exact algorithms for KMC are known, some of them under multiple names, but they give statistically equivalent results.[17] Figure 1, taken from ref 15, depicts the general structure common to all KMC algorithms. We will focus in particular on the first-reaction method (FRM), developed originally by Gillespie for well-mixed systems[8] and then formulated for on-lattice chemistry by Jansen and coworkers.[17−19] In the FRM, one identifies every possible lattice process that could occur given the current state of the system and places it in a priority queue based on a provisional occurrence time. This occurrence time is sampled from an exponential distribution with the rate parameter being equal to the appropriate rate constant, which may depend on the local configuration. The process of highest priority (earliest occurrence time) is executed, and the state is updated accordingly. This change of state will affect which processes can take place in future and may change their rate constants, so the occurrence times and process queue are also updated as necessary.[8,17−19]

Received: August 18, 2023
Revised: October 22, 2023
Accepted: October 23, 2023
Published: November 21, 2023



ACS Publications
© 2023 American Chemical Society

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

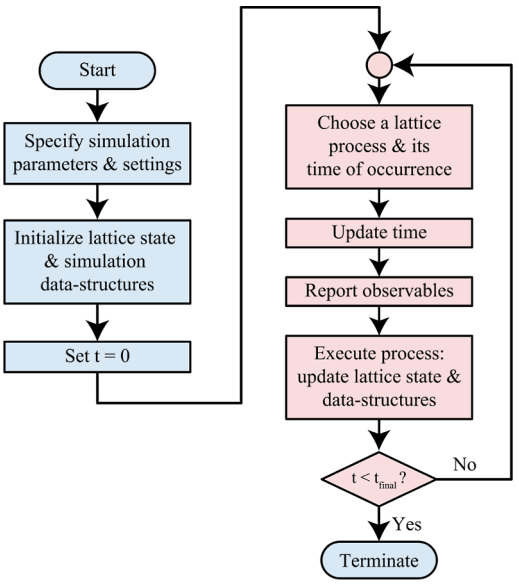


Figure 1. General structure of a KMC algorithm presented as a flowchart. Reproduced with permission from ref 15. Copyright 2022, Authors.

One of the key practical advantages of on-lattice KMC over less expensive microkinetic modeling[20] is its ability to capture the effects of spatially correlated adlayers in a formally exact way. Spatial correlations may arise even in ideal or near-ideal adlayers, for instance, if the diffusion of one or more adsorbates is slow relative to the reaction.[21,22] More commonly, however, correlations are attributed to lateral interactions between adsorbates.[23−30] In KMC, one is at liberty to define the microscopic rate parameters as functions of the adlayer structure, a popular approach to which is to use Brønsted−Evans−Polanyi (BEP) relations that correlate the activation and reaction energies of an elementary event.[31−33] Thus, lateral interactions may be incorporated explicitly, and doing so has been shown to have a strong effect on predictions of catalytic activity.[25,33−36]

Carrying out high-fidelity (i.e., ab initio) calculations of the microscopic rate parameters (e.g., activation energies) for every relevant configuration of adsorbates on the lattice would be prohibitively time-consuming. One can, however, fit mathematical models of lateral interactions to well-selected ab initio data. A popular approach is to represent the total energy of the lattice and its adsorbates using a so-called cluster expansion (CE), in which each term associates a particular geometric pattern or "cluster" of adsorbates with an effective interaction energy. The problem of computing the adlayer energy is thus reduced to the identification of these patterns on the surface.[33,37−39] In principle, the CE can represent any adlayer energy exactly, but, in practice, it must be truncated to a manageable number of terms.[40]

Subject to the memorylessness (Markov) property described above, the KMC method is applicable to surface chemistries of almost arbitrary complexity, involving not just lateral interactions but also multiple binding site types and multidentate adsorbate species. In practice, achieving an efficient implementation that can boast such a high degree of generality presents a significant technical challenge. To this end, Stamatakis and Vlachos[10] formulated a graph-theoretical KMC framework in which the lattice and elementary steps are represented as connected graphs. The elementary steps are then mapped to feasible lattice processes by solving subgraph isomorphism problems, where the elementary step and lattice graphs adopt the roles of the "query" and "data" graph, respectively.[41] Nielsen et al.[33] extended the formalism to incorporate the CE framework for adsorbate lateral interactions. Each cluster or "figure" appearing in the CE Hamiltonian has a graph representation and is mapped to lattice sites involved in lateral interactions by solving a subgraph isomorphism problem, analogous to the means of identifying possible lattice processes.[33] GT-KMC with the FRM is implemented in our Fortran 2003 software package, *Zacros*.[16,33,41−45]

Zacros has been used successfully to simulate a range of catalytic systems.[46−60] Still, while accounting for lateral interactions is possible, it is often extremely time-consuming, especially when the interactions are long-range. Long-range interactions are characterized by the contribution of large clusters to the CE; the cost of solving a subgraph isomorphism problem increases sharply with the number of vertices in the "query graph", which in this case represents the cluster. Several approaches may be adopted to reduce the computational burden without further approximation, arguably the most straightforward of which is shared-memory parallel processing. Nielsen et al. introduced parallelization in *Zacros* with OpenMP, which was shown to offer considerable speedups depending on the CE employed.[33] Applied to a model of catalytic NO oxidation on Pt(111), with a particularly challenging CE encompassing up to eighth nearest-neighbor pairwise interactions between adsorbed oxygen atoms (12 figures in total),[32,61] the parallelization was found to be almost perfectly efficient. The benefit was less pronounced when only shorter-range lateral interactions were considered.[33] In any case, the speedup will always be limited by the number of threads available as well as the number of clusters that need to be detected. More recently, Ravipati et al. implemented a caching scheme to reduce the number of repetitive pattern detections that need to take place. This works by caching the number of lattice instances of each interaction pattern involving the products of each lattice process, then updating this number efficiently (by detecting only the necessary patterns) when a reaction happens in that lattice process's neighborhood. Utilizing this scheme in combination with parallel processing was shown to accelerate KMC execution by a factor of up to 20×, using the same 12-figure CE NO oxidation model mentioned above.[42]

A conceptually different approach to evaluating the CE Hamiltonian was proposed by Hess,[39] in which a look-up table is utilized to specify the interaction energy of a given set of lattice sites as a function of their occupancies. To accelerate the calculations, Hess also developed the supercluster contraction scheme, whereby the number of terms in the CE Hamiltonian is effectively reduced by grouping them together. This requires only inexpensive postprocessing of the original CE and does not introduce any further approximations.[39] The scheme was introduced in tandem with two other algorithmic improvements, namely, the use of subtraction schemes for updating kinetic parameters and the supersite approach to optimize the selection of the next lattice process. It is important to note, however, that the latter two approaches are applicable specifically to "direct method"-KMC,[8] also known as the variable step-size method,[19] whereas *Zacros* implements the first-reaction method as described above. Nonetheless, the supercluster approach shows promise as a standalone

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

acceleration scheme, and further work will be required to determine whether it can be adapted and automated for compatibility with the graph-theoretical CE formalism of Nielsen et al.[33] Other promising recent advancements include the RMC-MKM approach of Kumar and Chatterjee,[62] which employs reverse Monte Carlo (RMC) to enhance microkinetic modeling (MKM) with a short-range order parameter that can capture spatial correlations between adsorbates.

Within the context of GT-KMC, another avenue worth exploring is the optimization of the pattern detection algorithms themselves. In *Zacros* 3.01, which is the most recent release, a simple depth-first search (DFS) strategy is used to solve the relevant subgraph isomorphism problems, incorporating rudimentary techniques to reduce the search space based on ideas introduced by Ullmann.[63] However, subgraph isomorphism is of central importance in a diverse range of fields, including, for instance, cheminformatics,[64] electronic engineering,[65] and cybersecurity.[66] This has driven the development of a large number of alternative algorithms. Notable examples from recent decades include the VF algorithm[67] and its variants (VF2,[68] VF2++,[69] VF3,[70] and VF3-Light[71]), the RI algorithm,[72] McGregor's algorithm,[73] and focus search.[74]

Which algorithm performs best depends on the properties of the graphs under consideration, such as their sizes, their densities, and whether they are planar. Preliminary investigations using a hexagonal lattice have identified the VF2 and RI algorithms as good candidates for detecting lateral interactions in GT-KMC.[75] Routines based on these algorithms have now been implemented in *Zacros* as part of this work, alongside the default "refined DFS" (rDFS) solver. For consistency with refs 33 and 42, we assess the performance using the same series of NO oxidation models studied therein. Quite remarkably, for the 12-figure CE, we obtain acceleration factors exceeding 125× by using VF2 or RI in tandem with shared-memory parallel processing. In stark contrast, neither VF2 nor RI offer any benefit (and in fact slow the KMC execution down) when only first nearest-neighbor interactions are considered. We also simulate a more complex system, namely, a model of methane cracking on Ni(111) developed by Yadavalli et al., which involves a 62-figure cluster expansion.[76] It is shown that RI offers significant speedups when the surface coverage is high such that there are many lateral interaction patterns to detect.

The rest of the paper is organized as follows. Methods contains an overview of the GT-KMC and CE methodology, followed by a discussion of the subgraph isomorphism algorithms under consideration and their implementation in *Zacros*. Results & Discussion details our performance benchmarks, and finally, Conclusions summarizes our findings and their significance.

METHODS

Theory. Here, we provide an overview of the graph-theoretical KMC (GT-KMC) framework and cluster expansion (CE) formalism for treating adsorbate lateral interactions. For a more detailed discussion, the reader is referred to refs 33, 41, 42.

Lattice and Its Coverage. Consider a lattice comprising N_S possible adsorption sites spanning $N_\mathcal{T}$ site types. The foundation of GT-KMC is a connected, undirected graph, $\mathcal{L}=(\mathcal{S}, \mathcal{E})$, which represents this lattice. Each vertex $i\in\mathcal{S}=\{1, \ldots, N_S\}$ represents a unique site, while each edge $(i, j) \in \mathcal{E} \subseteq (\mathcal{S} \times \mathcal{S})$ represents a neighboring relation between two sites. The physical nature of site i is specified by a three-element vector (formally, a tuple) $\mathbf{s}_i\in\mathcal{T}\times\mathbb{R}^2$, where $\mathcal{T}=\{1, \ldots, N_\mathcal{T}\}$. The elements of $\mathbf{s}_i$ denote the site type and the x- and y-coordinates, respectively. The lattice sites can be described collectively by the $N_S\times 3$ array $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_S}\} = \{(s_{i,j})_{j=1}^{3}\}_{i=1}^{N_S}$.

To be able to describe the state (coverage) of the system, suppose that there are $N_\mathcal{A}+1$ different possible surface species (adsorbates). One of these is a vacancy (representing the absence of any adsorbate), and this is given an index of 0. The denticity of species $n \in \mathcal{A}_0=\mathcal{A}\cup\{0\}$, where $\mathcal{A}=\{1, \ldots, N_\mathcal{A}\}$, is given by $d_n \in \{1, \ldots, d_{\max}\}$, where $d_{\max}$ is the maximum denticity of all the species (note that $d_0 = 1$). The state of site i is thus fully described by a three-element vector, $\boldsymbol{\sigma}_i\in\mathcal{B}\times\mathcal{A}_0 \times \{1, \ldots, d_{\max}\}$, where $\mathcal{B}=\{1, \ldots, N_\mathcal{B}\}$ and $N_\mathcal{B}$ is the total number of distinct entities (adsorbates and vacancies) occupying the lattice sites. The elements of σ_i specify the entity label, the species, and the subunit thereof adsorbed on site i, respectively. The state of the entire surface is accordingly described by the $N_S\times 3$ array $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{N_S}\} = \{(\sigma_{i,j})_{j=1}^{3}\}_{i=1}^{N_S}$. One can easily show that $N_\mathcal{B} = \sum_{i\in\mathcal{S}} 1/d_{\sigma_{i,2}}$. Note that any observable property of the system must be invariant under permutations of the entity labels, which are entirely arbitrary.

Elementary Steps. While $\mathbf{s}$ does not change, $\boldsymbol{\sigma}$ evolves as the simulation progresses. This occurs via a chemical reaction mechanism consisting of several possible elementary steps, each of which can be represented by a graph. The graph representing elementary step α is denoted by $\mathcal{R}_\alpha = (\Xi_\alpha, \Psi_\alpha)$, where the vertices $\Xi_\alpha = \{1,\cdots,N_{\Xi\alpha}\}$ correspond to the sites involved and the edges $\Psi_\alpha \subseteq (\Xi_\alpha \times \Xi_\alpha)$ describe their connectivity on the lattice. It is important to highlight that the vertices of $\mathcal{R}_\alpha$ do not correspond to any particular set of lattice sites but rather comprise a generic site pattern, instances of which may occur in several locations on the lattice. Unlike for lattice sites, no positional data needs to be specified for the sites of an elementary step, so the nature of site i of step α can be fully captured by the scalar variable $\xi_{\alpha,i} \in \mathcal{T}_0=\mathcal{T}\cup\{0\}$. The case of $\xi_{\alpha,i} = 0$ arises when the site concerned can be of any type. If necessary, geometric constraints can be defined by specifying the angles between sets of three vertices

$$\mathrm{angle}(\mathcal{R}_\alpha; i, j, k) = \vartheta_{ijk}^{\alpha} \tag{1}$$

where $i, j, k \in \Xi_\alpha$, and $0 \le \vartheta_{ijk}^{\alpha} < 2\pi$. The initial and final coverage patterns of step α are fixed properties of $\mathcal{R}_\alpha$, represented by arrays $\boldsymbol{\theta}_{\alpha,i}^{\mathrm{ini}}, \boldsymbol{\theta}_{\alpha,i}^{\mathrm{fin}} \in \Xi_\alpha \times \mathcal{A}_0 \times \{1, \ldots, d_{\max}\}$ with $i \in \Xi_\alpha$.

To be able to model the kinetics of the system, one needs to know the propensity of each elementary step. We assume that the kinetic rate constant for step α can be fitted to the Arrhenius equation

$$k_\alpha(\boldsymbol{\sigma}, \boldsymbol{\sigma}', T) = A_\alpha(T)\exp\left(-\frac{E_\alpha^{\ddagger}(\boldsymbol{\sigma}, \boldsymbol{\sigma}')}{k_\mathrm{B}T}\right) \tag{2}$$

where $A_\alpha(T)$ is the pre-exponential factor, $E_\alpha^{\ddagger}(\boldsymbol{\sigma},\boldsymbol{\sigma}')$ is the activation energy, k_B is the Boltzmann constant, T is

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

temperature, and $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ describe the lattice state before and after the reaction, respectively. In practice, A_α is usually parameterized using TST.[2,77,78] The activation energy is fitted to a BEP relationship[31]

$$E_{\alpha}^{\ddagger}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') = \max(0, \Delta E_{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\sigma}'), E_{\alpha,0}^{\ddagger} + \omega_{\alpha}[\Delta E_{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') - \Delta E_{\alpha,0}]), \quad (3)$$

where $\Delta E_\alpha(\boldsymbol{\sigma}, \boldsymbol{\sigma}')$ is the reaction energy and ω_α is the proximity factor. The quantities $\Delta E_{\alpha,0}$ and $E_{\alpha,0}^{\ddagger}$ represent the reaction and activation energies in the absence of any spectators (adsorbates not directly involved in the elementary step). The reverse of step α is denoted by $\bar{\alpha}$ and its activation is given by

$$E_{\bar{\alpha}}^{\ddagger}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') = E_{\alpha}^{\ddagger}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') - \Delta E_{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') \quad (4)$$

which stems from the principle of microscopic reversibility.

Adsorbate Lateral Interactions. In principle, one could calculate $\Delta E_\alpha(\boldsymbol{\sigma}, \boldsymbol{\sigma}')$ on the fly using ab initio methods, but this would be prohibitively time-consuming. Instead, ab initio (or other) data collected before the simulation can be used to parameterize a Hamiltonian for the adsorbate layer, $\mathcal{H}(\boldsymbol{\sigma})$, which captures the physical and chemical interactions of the adsorbates with the lattice and with each other. Thus,

$$\Delta E_{\alpha}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') = \mathcal{H}(\boldsymbol{\sigma}') - \mathcal{H}(\boldsymbol{\sigma}) + \Delta E_{\alpha}^{\mathrm{gas}} \quad (5)$$

where $\Delta E_\alpha^{\mathrm{gas}}$ is the change in energy of all gas species involved in step α.

If there is only one monodentate adsorbed species and one site type, the state vector can be expressed in the simplified form $\boldsymbol{\sigma} = \{\sigma_i, \ldots, \sigma_{N_S}\}$, where σ_i is equal to 1 if site i is occupied or 0 if it is unoccupied. Then a suitable functional form for the Hamiltonian is the lattice-gas expansion[79−81]

$$\mathcal{H}(\boldsymbol{\sigma}) = H_0 + h\sum_{i\in S}\sigma_i + \frac{1}{2!}\sum_{i,j\in S}J_{ij}\sigma_i\sigma_j + \frac{1}{3!}\sum_{i,j,k\in S}J_{ijk}\sigma_i\sigma_j\sigma_k + \ldots, \quad (6)$$

where H_0, h, J_{ij}, J_{ijk}, ... are parameters to be fitted. Specifically, H_0 may be interpreted as the energy of the empty lattice, h as an adsorption energy, and J_{ij} and J_{ijk} as two- and three-body effective interaction energies. In practice, the lattice-gas expansion is truncated to a finite number of terms. To recast eq 6 in a graph-theoretical form, we recognize that each product $\sigma_i\sigma_j\ldots$ evaluating to 1 represents a particular pattern of adsorbates interacting on the lattice, while the coefficient $J_{ij\ldots}$ gives the associated effective interaction energy. Given the two-dimensional translational symmetry of the lattice, the value of $J_{ij\ldots}$ must depend only on the geometry of the pattern and not on its absolute position. We can therefore divide the summation into contributions from F distinct pattern geometries or "clusters", also referred to as figures. Each such cluster can be represented by a connected graph $C_\gamma = (\Phi_\gamma, \Pi_\gamma)$, with $\gamma \in \{1, \ldots, F\}$. The associated coverage pattern is described by the scalars $\chi_{\gamma,i} \in \Phi_\gamma$ which take values of 0 or 1, with $i \in \Phi_\gamma$. The Hamiltonian can then be expressed as

$$\mathcal{H}(\boldsymbol{\sigma}) = \sum_{\gamma=1}^{F}\frac{\mathrm{ECI}_\gamma}{\mathrm{GM}_\gamma}\cdot\mathrm{NCE}_\gamma(\boldsymbol{\sigma}) \quad (7)$$

where ECI_γ is the effective cluster interaction for figure γ, GM_γ is the graph multiplicity (essentially a symmetry number), and NCE_γ is the number of instances of figure γ on the lattice (each of which must have the correct coverage pattern specified by $\boldsymbol{\chi}_\gamma$). Note that each ECI equates, in general, to a linear combination of the parameters H_0, h_i, J_{ij}, J_{ijk}, ... of eq 6.

The advantage of eq 7 is that it immediately generalizes to more complex surfaces, involving $N_{\mathcal{T}}$ site types and $N_{\mathcal{A}}$ adsorbate species of arbitrary denticity. The site type of vertex i of cluster γ is specified by $\varphi_{\gamma,i} \in \mathcal{T}$. A three-element vector $\boldsymbol{\chi}_{\gamma,i} \in [\Phi_\gamma \times \mathcal{A}_0 \times \{1, \ldots, d_{\max}\}] \cup \{\&\}$ describes the associated coverage pattern, where & = (&, &, &) denotes a state that is unspecified. This is often the case for intermediate sites in long-range interaction patterns, which can be vacant or occupied by any species; the corresponding graph vertices are dubbed "nonspecific".[42] Geometric constraints can be defined where necessary by specifying the angles between sets of three vertices

$$\mathrm{angle}(C_\gamma; i, j, k) = \phi_{ijk}^{\gamma} \quad (8)$$

where $i, j, k \in \Phi_\gamma$ and $0 \le \phi_{ijk}{}^{\gamma} < 2\pi$.

By way of illustration, let us consider the energy changes that might occur when a CO molecule desorbs from a square lattice. In our simplified model, the cluster expansion contains just two terms, depicted graphically in Figure 2a. The first

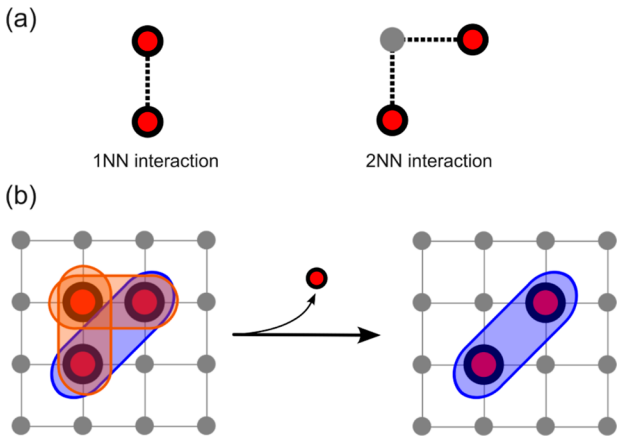


Figure 2. (a) Graphical representation of the 1NN and 2NN lateral interaction patterns contributing to a simplified cluster expansion Hamiltonian for CO on a square lattice. (b) Schematic representation of a CO desorption event with the 1NN and 2NN interactions highlighted in orange and blue, respectively. The desorption event eliminates both 1NN interactions, while the 2NN interaction (in which the central site is nonspecific) remains intact.

involves a 2-site cluster and corresponds to a first nearest-neighbor (1NN) lateral interaction. Both sites are occupied by monodentate CO*; hence, the coverage pattern can be described as

$$\boldsymbol{\chi}_1 = \{(1,1,1), (2,1,1)\} \quad (9)$$

The second term involves a right-angled, 3-site cluster and corresponds to a second nearest-neighbor (2NN) interaction. While the terminal sites are each occupied by CO*, the state of the central site is unspecified. Hence,

$$\boldsymbol{\chi}_2 = \{(1,1,1), (\&,\&, \&), (2,1,1)\} \quad (10)$$

and the geometric constraint is given by

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

$$\text{angle}(C_2;1,2,3)=\pi/2 \tag{11}$$

Figure 2b depicts the initial and final lattice states for a particular CO desorption event involving two spectators. The relevant 1NN and 2NN interaction patterns are colored orange and blue, respectively. As illustrated, the desorption event eliminates two 1NN interactions; the GT-KMC algorithm must correspondingly remove these from the list of valid pattern instances (see ref 42). On the other hand, the 2NN interaction remains intact because the intermediate site thereof is nonspecific. In fact, two such "L"-shaped patterns are present on the lattice. However, they correspond physically to the same effective interaction, and only one will be included in the list of valid pattern instances (for nonspecific sites, our algorithms seek only one valid mapping to avoid unnecessary proliferation of interaction pattern instances).[42]

Subgraph Isomorphism Problems. A key component of the GT-KMC simulation is compiling the list of potential lattice processes, given the state of the adsorbate layer at a particular time. To achieve this, one needs to identify all chemically sound mappings from the sites of each elementary step to the sites of the lattice. In the language of graph theory, this means finding all unique subgraphs of the lattice graph $\mathcal{L}$, that are isomorphic to each elementary step graph $\mathcal{R}_\alpha$.[41] In this "subgraph isomorphism problem", $\mathcal{L}$ takes the role of the "data graph", while $\mathcal{R}_\alpha$ takes the role of the "query graph".[82]

For systems with adsorbate lateral interactions, another key component of the simulation is calculating the lattice state-dependent activation energies using eqs 3 and 4. To do so requires evaluation of the cluster expansion (CE) of eq 7. For each cluster γ, one must count the number of NCE_γ of corresponding lateral interactions on the lattice. Similar to detecting lattice processes, this amounts to solving a subgraph isomorphism problem, except that the query graph is now C_γ instead of $\mathcal{R}_\alpha$.[33] Specifically, we search for each injective function $\mathcal{M}:\Phi_\gamma\rightarrow\mathcal{S}$ that meets the following criteria (in which we drop the index γ for brevity):

Crit. 1:

For every edge connecting two sites i and j of the cluster graph, there is an edge connecting the corresponding sites $\mathcal{M}(i),\mathcal{M}(j)$ in the lattice graph

$$\forall\ (i,j)\in\Pi,\ (\mathcal{M}(i),\mathcal{M}(j))\in\mathcal{E} \tag{12}$$

Crit. 2:

(a) The type of each site i in the cluster graph is compatible with that of its corresponding site $\mathcal{M}(i)$ in the lattice graph

$$\forall\ i\in\Phi,\varphi_i\in\{s_{\mathcal{M}(i),1},0\} \tag{13}$$

(b) The coverage pattern of the cluster graph is compatible with that of its corresponding lattice subgraph

$$\forall\ i\in\Phi,\ \boldsymbol{\sigma}_{\mathcal{M}(i)}\in\{(\mathcal{F}(\chi_{i,1}),\chi_{i,2},\chi_{i,3}),\&\} \tag{14}$$

where

$$\mathcal{F}:\{1,\ldots,\sum_{i\in\Phi}1/d_{\chi_{i,2}}\}\rightarrow\{1,\ldots,\sum_{i\in\mathcal{S}}1/d_{\sigma_{i,2}}\} \tag{15}$$

maps the entities involved in the interaction to those present on the lattice.

(c) Whenever specified, the angles between three vertices of the cluster graph are the same as the angles between the corresponding lattice graph vertices

$$\text{angle}(\mathcal{L};\ \mathcal{M}(i),\mathcal{M}(j),\mathcal{M}(k))=\text{angle}(C;i,j,k) \tag{16}$$

Taken together, Crit. 2(a–c) can be summarized by the requirement that the vertices of C are *compatible* with those of the corresponding lattice subgraph. It is possible to introduce additional criteria under this heading, such as specifying the absolute orientation of the subgraph with respect to a lattice vector.

Note that the lattice process detection problem is defined by analogous criteria, but we focus on lateral interactions as these are more relevant to accelerating KMC simulations (see below). Note also that in practice, one does not need to identify all possible cluster mappings "from scratch" for every new lattice state. After each event occurs, it is only necessary to calculate activation energies for any newly feasible lattice processes, i.e., those in which the product species participate, as well as any processes already queued that are in the neighborhood of that which just took place. Full algorithmic details of the kinetic constant update steps are described in refs 33 and 42. The key point is that only patterns that involve the relevant product species and any neighboring adsorbates need to be detected to calculate a particular activation energy.

Algorithms. *Basic Depth-First Search Algorithms.* The subgraph isomorphism problem is well-studied.[63,67−75,82−84] In the general case, it is NP-complete, with the time taken to solve it scaling exponentially with the size of the query graph.[85] For specific cases, however, efficient solutions can be obtained via carefully constructed rules and procedures for reducing the search space.

Consider a brute-force matching approach in which one loops over the vertices of the query graph (query vertices) and pairs them, in turn, with each of the vertices of the data graph (data vertices). When every query vertex is paired, this constitutes a trial mapping $\tilde{\mathcal{M}}:\Phi\rightarrow\mathcal{S}$, where we have adopted the notation of energetic clusters but dropped the index γ for brevity. If $\tilde{\mathcal{M}}$ meets both (sets of) criteria detailed above, namely, every query edge has a corresponding data edge and any vertex/edge compatibility rules are obeyed, then it describes a subgraph isomorphism $\mathcal{M}=\tilde{\mathcal{M}}$; otherwise, it is discarded. This procedure is guaranteed to find all subgraph isomorphisms and can be visualized as the traversal of the search tree shown in Figure 3. Each node of the tree labeled i,j indicates the pairing of query vertex $i\in\Phi$ with data vertex $j\in\mathcal{S}$, where, in this case, $N_\Phi=2$ and $N_\mathcal{S}=3$. In general, the tree will contain $N_\mathcal{S}^{N_\Phi}$ branches; thus, the time taken to complete the pattern matching grows exponentially with the size of the

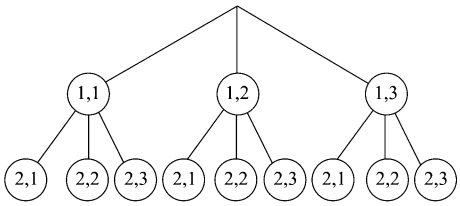


Figure 3. Illustration of the brute-force search tree for subgraph isomorphisms where the query graph contains 2 vertices and the data graph contains 3 vertices. Each node i,j corresponds to a trial mapping of query vertex i to data vertex j.

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

query graph, as noted above. Furthermore, the order in which the nodes are traversed is important. For large data graphs, a breadth-first search (BFS), in which all nodes at a given depth (level) are explored before moving onto the next level, would suffer large memory requirements. A DFS strategy is, therefore, more pragmatic, whereby each branch is explored to its end before backtracking and moving onto the next branch.

While the brute-force DFS strategy, denoted henceforth by bfDFS, circumvents large memory requirements, traversing the entire search tree is still impractical for all but the smallest data graphs. Established subgraph isomorphism algorithms aim to drastically reduce the size of the search space without introducing too much computational overhead. A straightforward refinement to bfDFS is to preemptively reject mappings in which the same data vertex is paired with more than one query vertex. The first, fifth, and ninth branches of the tree in Figure 3 would then be effectively discarded or "pruned". In addition to this, one can check the feasibility of each partial trial mapping each time a new query−data vertex pair is introduced, rather than waiting until every query vertex is paired. If a partial (trial) mapping violates any of the subgraph isomorphism criteria, then it is abandoned, thus pruning the prevailing branch of the search tree.

Both of these "pruning rules" are exploited in the default pattern matching routines of *Zacros* 3.01, used for detecting both lattice processes and lateral interactions. A further refinement exploits the fact that only the patterns involving newly adsorbed product species must be detected following each event. Each trial mapping therefore starts with at least one query−data vertex pair already "locked in" at the root of the search tree. Following ref 42, we can say that the corresponding adsorbate is "fixed" on the pattern, with the number of "fixed" vertex pairs equal to the denticity of that adsorbate. Clearly, it is pointless to try to map lattice sites beyond a "local neighborhood", the size of which depends on that of the pattern under consideration. More precisely, this "local neighborhood" can be found by computing the length l of the longest path that connects a fixed query vertex with a nonfixed one and finding all the data vertices that exist within l edges of the corresponding "fixed" data vertices (the latter being those mapped to the "fixed" query vertices). This way, one obtains the set $\mathcal{S}_{\mathrm{loc}}$ that essentially contains the (nonfixed) data vertices within l edges of the adsorbate. One can rule out matching any data vertices not in $\mathcal{S}_{\mathrm{loc}}$, and include only elements of $\mathcal{S}_{\mathrm{loc}}$ in the trial mappings, leading to a potentially drastic reduction in the size of the search tree. We refer to our "refined" DFS algorithm as rDFS, rather than "Ullmann's algorithm" as in ref 41, so as to avoid confusion with the bit-matrix manipulations described in Ullmann's original paper.[63] However, the pruning rules involved in rDFS are inspired by those described therein.

More Advanced Depth-First Search Algorithms: VF2 and RI. While the rDFS algorithm has been used successfully to solve subgraph isomorphism problems in a range of GT-KMC simulations, the computational burden can be significant when the query graphs are large. This is relevant mostly for the detection of lateral interactions as this is usually the simulation bottleneck. To reduce the search space more aggressively, algorithms such as VF2 and RI start with a strategic choice of the order in which the query vertices will be matched, unlike rDFS in which the order is arbitrary. VF2 adopts a simple, dynamic ordering strategy, whereby one always searches for matches of the query vertex that has the highest degree (number of neighbors) among the neighbors of those previously matched.[68] This way, query vertices with more stringent connectivity requirements are generally matched earlier. In contrast, RI adopts a static ordering strategy in which the sequence $\boldsymbol{\mu}$ of query vertices is determined and stored before any matching takes place. This ordering strategy follows a similar principle as that of the VF2 algorithm but specifically tries to favor query vertices that are "more connected" to those already matched.[72] This is achieved by assigning a lexicographic score to each query vertex $i \notin \boldsymbol{\mu} (i \in \Phi)$ based on the cardinality of three different sets; the highest scoring vertex is then appended to $\boldsymbol{\mu}$. This process is iterated until $\boldsymbol{\mu}$ includes all the query vertices. For a given query vertex $i \notin \boldsymbol{\mu}$ the three sets in order of priority are as follows:

1. S_{vis} is the set of vertices in $\boldsymbol{\mu}$ that are neighbors of vertex i.
2. S_{neig} is the set of vertices in $\boldsymbol{\mu}$ that are neighbors of at least one query vertex outside $\boldsymbol{\mu}$ that is a neighbor of vertex i.
3. S_{unv} is the set of vertices that are neither in $\boldsymbol{\mu}$ nor neighbors of vertices in $\boldsymbol{\mu}$ but are neighbors of vertex i.

The ordering strategies just described provide any benefit only if they are combined with sensible choices for the domain of each query vertex i, denoted by domain(i). This is the set of data vertices to which one attempts to map query vertex i; in the case of rDFS, as we have seen, it includes (rather crudely) all of the unmatched vertices within l edges of the fixed adsorbates. In VF2, domain(i) includes just the unmatched neighbors of the previously matched data vertices.[68] RI is stricter still, with domain(i) comprising just the unmatched neighbors of a *single* data vertex that has previously been matched with a neighbor of i. More precisely, while computing the ordered sequence of query vertices $\boldsymbol{\mu}$, we also determine the "parent" pt(i) of each vertex $i \in \boldsymbol{\mu}$, where pt(i) is defined as the first (i.e., lowest-index) member of $\boldsymbol{\mu}$ that is a neighbor of i. Then, domain(i) consists of the unmatched neighbors of $\mathcal{M}(\mathrm{pt}(i))$.[72]

Finally, while all of the isomorphism algorithms search for matches meeting the same criteria, the order and manner in which these criteria are checked can affect the execution speed. In rDFS and VF2, for each new trial pairing of i with $\tilde{\mathcal{M}}(i)$, one first checks whether the connectivity satisfies Crit. 1, i.e., whether there exists a data edge $(\tilde{\mathcal{M}}(i), \tilde{\mathcal{M}}(j)) \in \mathcal{E}$ for each neighbor j of query vertex i. If so, Crit. 2 is checked by comparing the site types and coverages of query vertex i and data vertex $\tilde{\mathcal{M}}(i)$, as well as any relevant geometric constraints. A different approach is taken in RI to account for the more aggressive connectivity-based pruning rules imposed by this algorithm. Thus, based on the assumption that relatively few trial mappings will fail to satisfy Crit. 1, Crit. 2 is checked first. Furthermore, Crit. 1 is only checked rigorously if $\tilde{\mathcal{M}}$ is first found to satisfy the weaker requirement that data vertex $\tilde{\mathcal{M}}(i)$ has at least as many neighbors as query vertex i.

Implementation in *Zacros*. *Pattern Detection Routines.* The original ("legacy") implementation of rDFS included in *Zacros* 3.01 remains the default option for detecting both lattice processes (above a certain level of complexity; see below) and lateral interactions. We refer to that implementation as rDFS-lgy. An alternative "modern" implementation, rDFS-mdn, of the same algorithm is also included as a method

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

of a stand-alone subgraph isomorphism class. While rDFS-lgy and rDFS-mdn are algorithmically equivalent, the former does not employ recursion; instead, it carries out an explicit loop over trial mappings of the nonfixed query vertices to members of $\mathcal{S}_{loc}$. One may view this as iterating through partial permutations of $\mathcal{S}_{loc}$. Each permutation is built up one site at a time such that invalid ("unfeasible") partial mappings can be immediately rejected.

In contrast, rDFS-mdn is implemented as a recursive subroutine (Scheme S1 in the Supporting Information). The recursion is expected to carry some computational overhead, but the code structure lends itself to the development of more advanced algorithms more naturally than that of rDFS-lgy. Specifically, the "matching" part of the algorithm, in which the next query vertex and its domain are determined, is implemented as a separate method (Scheme S2), as is the "feasibility" check, which determines whether a given (partial) trial mapping satisfies Crit. 1 and 2 (Scheme S3). In fact, Crit. 2 is assessed by calling an external feasibility or "compatibility" function (Scheme S4), which is not part of the subgraph isomorphism module but rather of that handling the lattice state. Note that all the pseudocode presented in Schemes S1−S10, while written with lateral interactions in mind, is equally applicable to lattice process detection by substituting the relevant variables $C \rightarrow \mathcal{R}$, $\Phi \rightarrow \Xi$, $\Pi \rightarrow \Psi$, etc.

Our VF2 implementation (Scheme S5) calls a unique matching function (Scheme S6) but the same feasibility function as rDFS-mdn. In contrast, our RI implementation (Scheme S7) calls both a unique matching function (Scheme S8) and a unique feasibility function (Scheme S9), which enables Crit. 2 to be checked before Crit. 1, as explained in the section Algorithms. Furthermore, the query vertices must be ordered in a preprocessing step, which is carried out by means of calling the RI_Order procedure (Scheme S10).

While lateral interactions are always detected using either rDFS-lgy or one of the recursive subgraph isomorphism procedures, these approaches are applied only to lattice processes when the mechanism (i.e., set of all elementary steps) exceeds a certain level of complexity. Specifically, when the mechanism involves only up to two-site steps and monodentate species, it is faster to follow a simpler approach. For each entity, the code identifies the species and loops over the elementary steps in which it can participate. In the case of a one-site step, provided the site type is correct, the corresponding lattice process is added to the queue. For a two-site step, a second loop is required to cycle over the neighbors of the first site.

Parallelization. Refs 33 and 42 detail how shared-memory (OpenMP) parallelization has already been used to accelerate GT-KMC simulations with *Zacros*. The most costly part of the simulation is the update of rate constants (and, in turn, future event times) following the execution of each event since this is where the detection of lateral interactions takes place. OpenMP parallelization was therefore implemented over the loop that cycles through the scheduled events within the neighborhood of the most recent lattice process. Thus, the new rate constants are calculated in parallel and stored in thread-private arrays.

We have extended the parallelization to work for simulations utilizing the new subgraph isomorphism class for lateral interaction detection. To achieve this, a thread-private instance of this class must be created for each thread. We also create thread-private instances of the class that handles the data required for the external feasibility check.

RESULTS AND DISCUSSION

NO Oxidation on Pt(111). Our first set of benchmarks is performed on the NO oxidation model established by Schneider and co-workers[32,61] and adapted within the KMC framework by Nielsen et al.[33] This model considers atomic oxygen on Pt(111) as the only adsorbate in the system, denoted as O*, and assumes that adsorbed oxygen and gaseous NO are in quasi-equilibrium with gaseous NO_2; thus, the dissociation of gaseous O_2 is the rate-limiting step. Underpinning this model is a hierarchy of CEs for O* developed by Schmidt et al.,[61] who used DFT calculations to fit progressively larger CEs, capturing longer ranges of lateral interactions. Schmidt et al. thus found that a 12-figure CE, including up to eighth nearest-neighbor pairwise interactions and two triplets (three-body interactions), is sufficient to predict formation energies and ground states accurately.[61] Wu et al.[32] then employed this model in a study that combined equilibrium MC with a microscopic rate-averaging scheme to predict the apparent rate of catalytic NO oxidation on Pt(111).

Based on the CEs of Schmidt et al.[61] and the kinetic parameters calculated by Wu et al.,[32] Nielsen et al.[33] built a corresponding set of GT-KMC models for the same NO oxidation reaction. Specifically, they benchmarked an earlier version of *Zacros* in which the detection of lateral interactions was already parallelized but limited to what we now call rDFS-lgy. They found that the execution was more than 4 orders of magnitude slower when using the most complex energetic model (12-figure) compared to the simplest (3-figure), illustrating the need to reduce the computational expense of computing long-range lateral interactions. Parallelizing the rate constant updates using OpenMP was shown to be highly effective for the 12-figure model, with execution speed scaling almost linearly. In contrast, parallelization with the 3-figure model was found to yield only modest speedups, plateauing at around 2× as the number of threads approaches 16. This was explained by the increasing time required to synchronize the threads when collecting the updated rate constants, which eventually becomes the simulation bottleneck when the number of rate constants affected by each event is small (i.e., the lateral interactions are short-range).

We have employed the same NO/Pt(111) reaction model and set of CEs to assess the improvements in performance yielded by our "modern" pattern detection routines. As described in ref 33, NO oxidation proceeds via the following reversible elementary steps

$$\mathrm{NO(g)} + \mathrm{O^*} \underset{k_{red}}{\overset{k_{oxi}}{\rightleftharpoons}} \mathrm{NO_2(g)} + {}^* \tag{17a}$$

$$\mathrm{O_2(g)} + 2{}^* \underset{k_{des}}{\overset{k_{ads}}{\rightleftharpoons}} \mathrm{O^*} + \mathrm{O^*} \tag{17b}$$

$$\mathrm{O^*} + {}^* \underset{k_{diff}}{\overset{k_{diff}}{\rightleftharpoons}} {}^* + \mathrm{O^*} \tag{17c}$$

These constitute NO oxidation, O_2 dissociative adsorption, and O* diffusion, respectively. Details of the graph representations and kinetic parameters can be found in ref 33. The energetic models and corresponding graph patterns are also described in detail in ref 33, but to guide our discussion, in Figure 4, we provide a qualitative illustration of

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

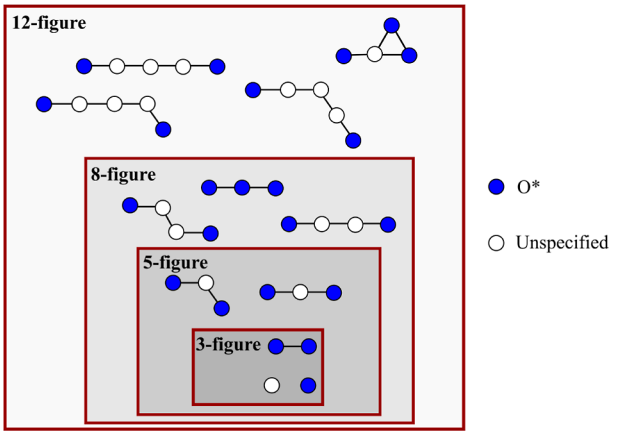


Figure 4. Energetic interaction patterns included in Schmidt et al.'s hierarchy of cluster expansions (CEs) for O* on Pt(111).[61] The underlying lattice is omitted for simplicity; for more detail, see Figure 2 of ref 33. We use blue filled circles to represent lattice sites occupied by O* and white filled circles to represent "nonspecific sites" (i.e., sites with an "unspecified" state), which may participate in the pattern regardless of their occupancy (see the section Adsorbate Lateral Interactions).

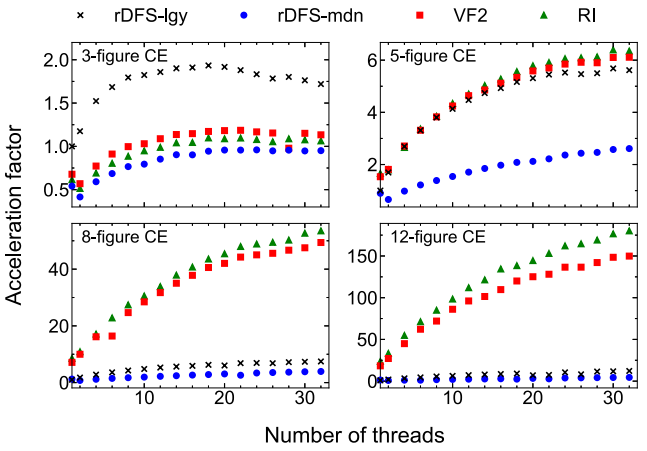


Figure 5. Plots of acceleration factors as a function of the number of threads for GT-KMC simulations of NO oxidation on Pt(111). The acceleration factor is defined with respect to a single-threaded run using rDFS-lgy to detect lateral interactions. These results were obtained using *Zacros* compiled with GNU Fortran.

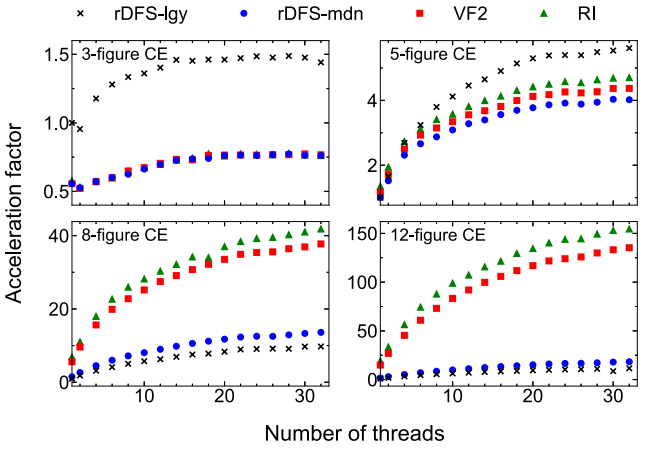


Figure 6. As in Figure 5 but obtained using *Zacros* compiled with Intel Fortran.

the figures included in each CE. The 3-figure expansion includes only point (1-body) terms and the first nearest-neighbor (1NN) effective O* − O* interaction, while the 5-, 8-, and 12-figure expansions cover up to 3NN, 5NN, and 8NN terms, respectively. The 8- and 12-figure expansions also include some triplet terms.

Our simulations employed a lattice size of 42 × 42 (1764 sites), with the temperature fixed at 480 K, total pressure at 1 bar, and the molar fractions of the reactive gas species at y_{O_2} = 0.1, $y_{NO} = 4.139938 \times 10^{-9}$, and $y_{NO_2} = 1.522998 \times 10^{-9}$. Starting from a steady-state adlayer structure, we ran a series of simulations using each combination of CE and pattern detection algorithm, with the rate constant updates parallelized over different numbers of threads. Each simulation was terminated after 1 h of clock time had elapsed. All simulations were carried out on type D nodes of the computational cluster Myriad@UCL, each containing 36 cores and 192 GB of RAM. We obtained two sets of results to compare the GNU and Intel Fortran compilers. Our performance metric is the acceleration factor, defined as the rate of event execution relative to that for a single-threaded run using rDFS-lgy for lateral interaction detection. Note that distinct from a "serial" run, a "single-threaded" run employs the OpenMP-enabled *Zacros* executable with the environment variable OMP_NUM_THREADS set to 1. It is also worth highlighting that the reaction mechanism involves only one- and two-site patterns and monodentate adsorbates, so lattice process detection is always carried out using the simpler method for elementary event detection described in Implementation in Zacros. This is not a simulation bottleneck.

The GNU and Intel acceleration factors are plotted with respect to the number of threads (processors) in Figures 5 and 6, respectively. For the 3-figure CE, as discovered by Nielsen et al.[33] and noted above, parallelization provides only modest speedups relative to single-threaded runs. The maximum acceleration factor is greater for the GNU-compiled version of *Zacros* than for the Intel-compiled version, although, interestingly, this maximum is reached at 16 threads, beyond which the event execution starts to slow down. Using any of the "modern" pattern detection routines also slows down the simulation. The latter is indicative of computational overheads involved in recursion and in initializing and manipulating the data structures needed to represent each mapping $\mathcal{M}$. The comparable performances of rDFS-mdn, VF2, and RI suggest that the "aggressive" approaches to search space reduction adopted in VF2 and RI are, in this case, excessive. This point is explored further below.

For the 5-figure CE, VF2 and RI display similar performance to rDFS-lgy. With the GNU version, rDFS-mdn is considerably slower and parallelizes less efficiently. In the Supporting Information, we show that this effect is somewhat lattice size-dependent as for large lattices, data duplication among threads becomes a bottleneck. Interestingly, with Intel-compiled *Zacros*, rDFS-mdn performs almost as well as the other algorithms. Overall, our results suggest that VF2 and RI offer some algorithmic advantage in terms of reducing the search space, but this is only just able to compensate for the overhead of recursion.

In stark contrast, simulations run with the 8- and 12-figure CEs can be sped up drastically by using either VF2 or RI to detect lateral interactions. Taking the Intel results as an

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

example, single-threaded runs with VF2 progress around 7 and 18 times faster than those using rDFS-lgy for the 8- and 12-figure CEs, respectively. The RI algorithm proves to be marginally faster still, with acceleration factors of roughly 9 and 24. Combining RI with parallelization over 32 processors, the acceleration factors reach 42 (8-figure) and 154 (12-figure). They are even more impressive with GNU Fortran compilation, reaching 53 (8-figure) and 180 (12-figure).

It is pertinent to ask why the VF2 and RI algorithms provide such large speedups relative to rDFS with the larger CEs but none at all with the smallest CE. To address this question, we examined the effectiveness of the search space reduction performed by each algorithm, keeping in mind that the pruning rules are based solely on connectivity and thus do nothing to eliminate partial matches violating Crit. 2 (compatibility). Accordingly, we define the "partial match success rate" (PMSR) to be the fraction of attempted partial matches found to satisfy Crit. 1. The ideal scenario, in which the search tree contains the smallest possible number of branches, is represented by a PMSR of 1.00, i.e., when every partial match attempted has acceptable connectivity.

Table 1 gives the PMSRs of lateral interaction detection for a series of 1 h, single-threaded simulations of NO oxidation on Pt(111). All three algorithms display "ideal" behavior for the 3-figure CE; hence, as Figures 5 and 6 revealed, there is no advantage to the intricate ordering strategies and pruning rules of VF2 or RI relative to the simplistic approach of rDFS. Additionally, larger figures are added to the CE and the PMSR of rDFS drops drastically, with only 5% of attempted partial matches satisfying Crit. 1 in the 12-figure case. In contrast, the PMSR of VF2 drops much more gradually, reaching a minimum of 0.46, which is consistent with Figures 5 and 6 and indicates that the search space reduction is fairly effective, even for large query graphs. Remarkably, the PMSR of RI remains at 1.00 up to and including the 8-figure CE, dropping only slightly to 0.89 for the 12-figure CE. RI is therefore almost as efficient as any algorithm can be for detecting lateral interactions in NO/Pt(111), insofar as one is concerned with minimizing the size of the search tree.

Table 1. Partial Match Success Rates

size of CE	PMSR		
	rDFS	VF2	RI
3	1.00	1.00	1.00
5	0.30	0.68	1.00
8	0.08	0.61	1.00
12	0.05	0.46	0.89

In practice, Table 1 only tells part of the story. For the 12-figure expansion, while RI is almost twice as effective as VF2 at eliminating partial matches that violate Crit. 1, the two algorithms yield similar KMC execution speeds. This is because much of the execution time is spent on checking whether Crit. 2 is satisfied, which involves comparing the site types, occupancies, and geometric properties of cluster and lattice vertices. As explained in More Advanced Depth-First Search Algorithms: VF2 and RI, RI performs this time-consuming step before checking Crit. 1, whereas rDFS and VF2 check Crit. 1 first. Therefore, since ∼90% of partial matches in RI satisfy Crit. 1 (see Table 1), we know that VF2 needs to check Crit. 2 ∼90% as many times as RI. Profiling with Arm MAP[86] revealed that for a single-threaded *Zacros* run, ∼60% of the total time is spent on checking Crit. 2 when RI is used to detect lateral interactions. In contrast, when VF2 is used, checking Crit. 2 accounts for ∼40% of the total time.

Methane Cracking on Ni(111). As a more realistic demonstration of the improvement in performance yielded by our modern pattern detection routines, we used *Zacros* to carry out KMC simulations of methane cracking on Ni(111). The reaction model was developed by Yadavalli et al.,[76] with a complex mechanism involving 2 site types, 5 adsorbate species, and 10 reversible elementary steps. These include all of the C−H activation steps leading from CH_4 up to C + 4H, such that the kinetics of dehydrogenation are captured in detail, whereas the formation of coke is captured at the level of thermodynamics only. A 62-figure CE is used to capture the effective cluster interactions between adsorbed CH_x species (x = 0, ..., 3) and covalently bonded C and CH species. This comprises pairwise interaction parameters up to the 3NN level, as well as several higher-level clusters, with the largest figure in the CE containing 5 sites. For full details of the mechanism and energetics model, the reader is referred to ref 76.

We set out to compare the performance of rDFS-lgy with that of RI, in both cases parallelizing the rate constant updates over 32 processors. Simulations were initialized with an empty lattice, a temperature of 1000 K and a pressure of 10.01 bar. To shed light on how surface coverage affects the execution speed, we stopped and restarted each simulation at KMC time intervals of 0.2 ms, up to a final KMC time of 5 ms. Thus, as a dynamic performance metric, we could estimate the rate of clock time advancement with respect to KMC time

$$S = \frac{\Delta \tau_{\text{clock}}}{\Delta t_{\text{KMC}}} \tag{18}$$

where τ_{clock} and t_{KMC} denote the clock time and KMC time, respectively. Note that *better* performance (i.e., faster KMC execution) is characterized by a *smaller* value of S. As for the NO oxidation system, all methane cracking simulations were carried out on type D nodes of the computational cluster Myriad@UCL. The compiler used was GNU Fortran.

In Figure 7, we plot S as a function of t_{KMC} for the methane cracking model with two different lattice sizes, 10 × 10 and 20 × 20. The results for each lattice size have been averaged over

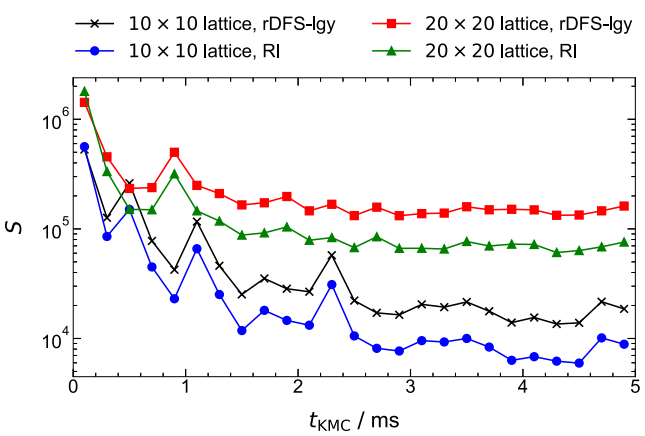


Figure 7. Results of GT-KMC performance benchmarks of the methane cracking model introduced in ref 76. We plot S against t_{KMC}, where S is the clock time advanced per unit of KMC time (see eq 18) and t_{KMC} is the KMC time. These results were obtained using *Zacros* compiled with GNU Fortran.

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

5 repetitions of the simulation (all using identical pseudorandom number streams). Notwithstanding some fluctuation, it is clear that S generally decreases as the simulation progresses, meaning that the KMC time advances more quickly. This is because when the lattice is mostly empty, the process queue contains a large number of adsorption events, each of which increments the KMC time by a relatively small amount. In contrast, as the system approaches steady-state behavior, the surface becomes "poisoned" by carbonaceous species, resulting in slower dynamics and, correspondingly, large interarrival times.

Turning our attention to the method of detecting lateral interactions, rDFS-lgy and RI show similar performance in the early stages of the simulation. This is easy to explain: on a surface with low coverage, the adlayer is almost ideal; thus, there are few interaction patterns to detect. As the CH/C species accumulate on the surface (comprising mostly ring-based structures), long-range interactions between them start to emerge; thus, the fraction of computational time dedicated to solving subgraph isomorphism problems increases. Accordingly, we see that the speedup offered by RI becomes more significant, exceeding 2× at t_{KMC} = 5 ms. Thus, the long time scales involved in the surface poisoning process have become more readily accessible, simplifying the elucidation of the "terminal" state of coking.[76]

CONCLUSIONS

In the graph-theoretical kinetic Monte Carlo (GT-KMC) framework, which is implemented in the *Zacros* software package, adsorbate lateral interactions are described by using the cluster expansion (CE) formalism. There, the formation energy of the adlayer configuration is expanded as a series of effective "cluster" interaction energies. To accurately compute the activation energies of elementary lattice processes, one then has to identify mappings between the clusters/figures of the CE and patterns of sites on the lattice. This amounts to solving a series of subgraph isomorphism problems and is analogous to the way the lattice processes are themselves identified.[33,41]

When the lateral interactions span long distances, the associated pattern detection becomes computationally highly demanding. Previous work to reduce this demand has focused on developing schemes for efficient parallel processing[33] and caching,[42] with fairly promising results. Here, we have built on this work by directly addressing the algorithms used to identify subgraph isomorphisms. The original choice of algorithm for *Zacros*, termed rDFS, adopts a fairly crude approach to reducing the size of the search space that is not always effective. We implemented two more sophisticated algorithms, namely, VF2 and RI, alongside rDFS in a "family" of recursive procedures. The last of these procedures is termed rDFS-mdn for the sake of distinguishability from the "legacy" implementation rDFS-lgy of the same algorithm, which remains in the code as well.

We benchmarked the performance of our "modern" pattern detection routines on a hierarchy of models of catalytic NO oxidation on Pt(111), developed in refs 32, 33, 61. These models treat the interactions between adsorbed oxygen atoms with varying levels of accuracy. The simplest of the models includes just 3 figures in the CE Hamiltonian, incorporating only up to first nearest-neighbor (1NN) pairwise interactions, while the most complex model includes 12 figures, incorporating up to 8NN pairwise interactions alongside some three-body terms. We found that VF2 and RI provide large improvements in performance with the more complex energetic models, yielding acceleration factors up to ∼180 (compared to single-thread runs) when combined with shared-memory parallel processing. On the other hand, rDFS-lgy is the best option when one only needs to detect short-range patterns, for which a simple search-space reduction strategy is adequate.

As a more realistic test, we compared the performance of RI to that of rDFS-lgy for computing lateral interactions in simulations of methane cracking on Ni(111).[76] At low surface coverage, when the number of adsorbate molecules interacting is small, the difference in performance was found to be small as well. However, as the lattice approached a poisoned state, the simulation using RI progressed more than twice as fast.

We conclude that KMC simulations of catalytic surface phenomena may be considerably accelerated by using sophisticated subgraph isomorphism algorithms for the detection of lateral interactions. Further, the acceleration is most significant when the adlayer contains a large number of long-range patterns. We expect that our implementation of VF2 and RI in *Zacros* will advance catalyst discovery by enabling simulations involving complex energetic models that were previously considered intractable.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpca.3c05581.

Pseudocode for the subgraph isomorphism algorithms implemented in *Zacros*, and additional GT-KMC benchmarks of the NO/Pt(111) system with the 5-figure cluster expansion comparing the performances of different algorithms across a range of lattice sizes (PDF)

AUTHOR INFORMATION

Corresponding Author

Michail Stamatakis − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.;* Present Address: Inorganic Chemistry Laboratory, University of Oxford, S Parks Rd, Oxford, OX1 3QR, United Kingdom; orcid.org/0000-0001-8338-8706; Email: michail.stamatakis@chem.ox.ac.uk

Authors

Raz L. Benson − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.;* orcid.org/0000-0002-9746-1860

Sai Sharath Yadavalli − *Department of Chemical Engineering, University College London, London WC1E 7JE, U.K.*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpca.3c05581

Notes

The authors declare the following competing financial interest(s): We are in the process of releasing a new version of our Graph-Theoretical Kinetic Monte Carlo (GT-KMC) software *Zacros* that incorporates the algorithms developed in this study. *Zacros* has been commercialized by UCL Business, the technology transfer office of University College London.

■ ACKNOWLEDGMENTS

R.L.B. and M.S. acknowledge funding from the Leverhulme Trust (project RPG-2017-361) and from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 814416 (ReaxPro). S.S.Y.'s doctoral studies are supported by an Impact Studentship sponsored by the Johnson Matthey Technology Centre. The authors acknowledge the use of the UCL Research Software Development Service (RSD@UCL) as well as the UCL High Performance Computing (HPC) Facility Myriad@UCL and associated support services in the completion of this work. We also thank Dr Ilektra-Athanasia Christidi for helpful discussions.

■ REFERENCES

(1) Neurock, M.; Hansen, E. W. First-principles-based molecular simulation of heterogeneous catalytic surface chemistry. *Comput. Chem. Eng.* **1998**, *22*, S1045−S1060.
(2) Reuter, K.; Scheffler, M. First-principles kinetic Monte Carlo simulations for heterogeneous catalysis: Application to the CO oxidation at Ru O2 (110). *Phys. Rev. B* **2006**, *73*, 045433.
(3) Reuter, K. *Modeling and Simulation of Heterogeneous Catalytic Reactions*; John Wiley and Sons, 2011; pp 71−111, Chapter 3.
(4) Stamatakis, M. Kinetic modelling of heterogeneous catalytic systems. *J. Phys. Consens. Matter* **2015**, *27*, 013001.
(5) Matera, S.; Schneider, W. F.; Heyden, A.; Savara, A. Progress in Accurate Chemical Kinetic Modeling, Simulations, and Parameter Estimation for Heterogeneous Catalysis. *ACS Catal.* **2019**, *9*, 6624−6647.
(6) Chen, B. W.; Xu, L.; Mavrikakis, M. Computational Methods in Heterogeneous Catalysis. *Chem. Rev.* **2021**, *121*, 1007−1048.
(7) Bortz, A.; Kalos, M.; Lebowitz, J. A New Algorithm for Monte Carlo Simulation of Ising Spin Systems. *J. Comput. Phys.* **1975**, *17*, 10−18.
(8) Gillespie, D. T. A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. *J. Comput. Phys.* **1976**, *22*, 403−434.
(9) Fichthorn, K. A.; Weinberg, W. H. Theoretical foundations of dynamical Monte Carlo simulations. *J. Chem. Phys.* **1991**, *95*, 1090−1096.
(10) Stamatakis, M.; Vlachos, D. G. Unraveling the complexity of catalytic reactions via kinetic monte carlo simulation: Current status and frontiers. *ACS Catal.* **2012**, *2*, 2648−2663.
(11) Leetmaa, M.; Skorodumova, N. V. KMCLib: A general framework for lattice kinetic Monte Carlo (KMC) simulations. *Comput. Phys. Commun.* **2014**, *185*, 2340−2349.
(12) Hoffmann, M. J.; Matera, S.; Reuter, K. kmos: A lattice kinetic Monte Carlo framework. *Comput. Phys. Commun.* **2014**, *185*, 2138−2150.
(13) Kunz, L.; Kuhn, F. M.; Deutschmann, O. Kinetic Monte Carlo simulations of surface reactions on supported nanoparticles: A novel approach and computer code. *J. Chem. Phys.* **2015**, *143*, 044108.
(14) Prats, H.; Illas, F.; Sayós, R. General concepts, assumptions, drawbacks, and misuses in kinetic Monte Carlo and microkinetic modeling simulations applied to computational heterogeneous catalysis. *Int. J. Quantum Chem.* **2018**, *118*, No. e25518.
(15) Pineda, M.; Stamatakis, M. Kinetic Monte Carlo simulations for heterogeneous catalysis: Fundamentals, current status and challenges. *J. Chem. Phys.* **2022**, *156*, 120902.
(16) Savva, G. Development, implementation and efficiency optimization of novel methods to accelerate kinetic Monte Carlo simulations of reactive systems. Ph.D. Thesis, University College London, 2022;.
(17) Lukkien, J. J.; Segers, J. P.; Hilbers, P. A.; Gelten, R. J.; Jansen, A. P. J. Efficient Monte Carlo methods for the simulation of catalytic surface reactions. *Phys. Rev. E* **1998**, *58*, 2598−2610.
(18) Jansen, A. P. J. *An introduction to Kinetic Monte Carlo Simulations of Surface Reactions*; Springer: Berlin, Heidelberg, 2012; Vol. *856*, pp 53−55.
(19) Jansen, A. P. J. Monte Carlo simulations of chemical reactions on a surface with time-dependent reaction-rate constants. *Comput. Phys. Commun.* **1995**, *86*, 1−12.
(20) Motagamwala, A. H.; Dumesic, J. A. Microkinetic Modeling: A Tool for Rational Catalyst Design. *Chem. Rev.* **2021**, *121*, 1049−1076.
(21) Matera, S.; Meskine, H.; Reuter, K. Adlayer inhomogeneity without lateral interactions: Rationalizing correlation effects in CO oxidation at RuO2(110) with first-principles kinetic Monte Carlo. *J. Chem. Phys.* **2011**, *134*, 064713.
(22) Savva, G. D.; Benson, R. L.; Christidi, I.-A.; Stamatakis, M. Exact distributed kinetic Monte Carlo simulations for on-lattice chemical kinetics: lessons learnt from medium- and large-scale benchmarks. *Philos. Trans. R. Soc. A* **2023**, *381*, 20220235.
(23) Hess, F.; Over, H. Kinetic Monte Carlo simulations of heterogeneously catalyzed oxidation reactions. *Catal. Sci. Technol.* **2014**, *4*, 583−598.
(24) Gambu, T. G.; Abrahams, R. K.; van Steen, E. Micro-kinetic modelling of CO-TPD from Fe(100)-incorporating lateral interactions. *Catalysts* **2019**, *9*, 310.
(25) Hess, F.; Sack, C.; Langsdorf, D.; Over, H. Probing the Activity of Different Oxygen Species in the CO Oxidation over RuO2(110) by Combining Transient Reflection-Absorption Infrared Spectroscopy with Kinetic Monte Carlo Simulations. *ACS Catal.* **2017**, *7*, 8420−8428.
(26) Hess, F.; Over, H. Rate-Determining Step or Rate-Determining Configuration? the Deacon Reaction over RuO2(110) Studied by DFT-Based KMC Simulations. *ACS Catal.* **2017**, *7*, 128−138.
(27) Farkas, A.; Hess, F.; Over, H. Experiment-Based Kinetic Monte Carlo Simulations: CO Oxidation over RuO2(110). *J. Phys. Chem. C* **2012**, *116*, 581−591.
(28) Sutton, J. E.; Lorenzi, J. M.; Krogel, J. T.; Xiong, Q.; Pannala, S.; Matera, S.; Savara, A. Electrons to Reactors Multiscale Modeling: Catalytic CO Oxidation over RuO2. *ACS Catal.* **2018**, *8*, 5002−5016.
(29) Van Bavel, A. P.; Hermse, C. G.; Hopstaken, M. J.; Jansen, A. P.; Lukkien, J. J.; Hilbers, P. A.; Niemantsverdriet, J. W. Quantifying lateral adsorbate interactions by kinetic Monte-Carlo simulations and density-functional theory: NO dissociation on Rh(100). *Phys. Chem. Chem. Phys.* **2004**, *6*, 1830.
(30) Jansen, A. P. J. Monte Carlo simulations of temperature-programmed desorption spectra. *Phys. Rev. B* **2004**, *69*, 035414.
(31) Michaelides, A.; Liu, Z. P.; Zhang, C. J.; Alavi, A.; King, D. A.; Hu, P. Identification of general linear relationships between activation energies and enthalpy changes for dissociation reactions at surfaces. *J. Am. Chem. Soc.* **2003**, *125*, 3704−3705.
(32) Wu, C.; Schmidt, D. J.; Wolverton, C.; Schneider, W. F. Accurate coverage-dependence incorporated into first-principles kinetic models: Catalytic NO oxidation on Pt (1 1 1). *J. Catal.* **2012**, *286*, 88−94.
(33) Nielsen, J.; D'Avezac, M.; Hetherington, J.; Stamatakis, M. Parallel kinetic Monte Carlo simulation framework incorporating accurate models of adsorbate lateral interactions. *J. Chem. Phys.* **2013**, *139*, 224706.
(34) Vignola, E.; Steinmann, S. N.; Vandegehuchte, B. D.; Curulla, D.; Stamatakis, M.; Sautet, P. A machine learning approach to graph-theoretical cluster expansions of the energy of adsorbate layers. *J. Chem. Phys.* **2017**, *147*, 054106.
(35) Jørgensen, M.; Grönbeck, H. Scaling Relations and Kinetic Monte Carlo Simulations To Bridge the Materials Gap in Heterogeneous Catalysis. *ACS Catal.* **2017**, *7*, 5054−5061.
(36) Huš, M.; Hellman, A. Ethylene Epoxidation on Ag(100), Ag(110), and Ag(111): A Joint Ab Initio and Kinetic Monte Carlo Study and Comparison with Experiments. *ACS Catal.* **2019**, *9*, 1183−1196.
(37) Sanchez, J. M.; Ducastelle, F.; Gratias, D. Generalized cluster description of multicomponent systems. *Phys. A Stat. Mech. Appl.* **1984**, *128*, 334−350.

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

(38) Drautz, R.; Singer, R.; Fähnle, M. Cluster expansion technique: An efficient tool to search for ground-state configurations of adatoms on plane surfaces. *Phys. Rev. B* **2003**, *67*, 035418.
(39) Hess, F. Efficient Implementation of Cluster Expansion Models in Surface Kinetic Monte Carlo Simulations with Lateral Interactions: Subtraction Schemes, Supersites, and the Supercluster Contraction. *J. Comput. Chem.* **2019**, *40*, 2664−2676.
(40) Miller, S. D.; Kitchin, J. R. Uncertainty and figure selection for DFT based cluster expansions for oxygen adsorption on Au and Pt (111) surfaces. *Mol. Simul.* **2009**, *35*, 920−927.
(41) Stamatakis, M.; Vlachos, D. G. A graph-theoretical kinetic Monte Carlo framework for on-lattice chemical kinetics. *J. Chem. Phys.* **2011**, *134*, 214115.
(42) Ravipati, S.; D'Avezac, M.; Nielsen, J.; Hetherington, J.; Stamatakis, M. A Caching Scheme to Accelerate Kinetic Monte Carlo Simulations of Catalytic Reactions. *J. Phys. Chem. A* **2020**, *124*, 7140−7154.
(43) Savva, G. D.; Stamatakis, M. Comparison of Queueing Data-Structures for Kinetic Monte Carlo Simulations of Heterogeneous Catalysts. *J. Phys. Chem. A* **2020**, *124*, 7843−7856.
(44) Ravipati, S.; Savva, G. D.; Christidi, I. A.; Guichard, R.; Nielsen, J.; Réocreux, R.; Stamatakis, M. Coupling the time-warp algorithm with the graph-theoretical kinetic Monte Carlo framework for distributed simulations of heterogeneous catalysts. *Comput. Phys. Commun.* **2022**, *270*, 108148.
(45) Savva, G. D.; Benson, R. L.; Christidi, I.-A.; Stamatakis, M. Large-scale benchmarks of the Time-Warp/Graph-Theoretical Kinetic Monte Carlo approach for distributed on-lattice simulations of catalytic kinetics. *Phys. Chem. Chem. Phys.* **2023**, *25*, 5468−5478.
(46) Prats, H.; Álvarez, L.; Illas, F.; Sayós, R. Kinetic Monte Carlo simulations of the water gas shift reaction on Cu(111) from density functional theory based calculations. *J. Catal.* **2016**, *333*, 217−226.
(47) Kopač, D.; Huš, M.; Ogrizek, M.; Likozar, B. Kinetic Monte Carlo Simulations of Methanol Synthesis from Carbon Dioxide and Hydrogen on Cu(111) Catalysts: Statistical Uncertainty Study. *J. Phys. Chem. C* **2017**, *121*, 17941−17949.
(48) Kopač, D.; Likozar, B.; Huš, M. Catalysis of material surface defects: Multiscale modeling of methanol synthesis by CO2 reduction on copper. *Appl. Surf. Sci.* **2019**, *497*, 143783.
(49) Huš, M.; Kopač, D.; Bajec, D.; Likozar, B. Effect of Surface Oxidation on Oxidative Propane Dehydrogenation over Chromia: An Ab Initio Multiscale Kinetic Study. *ACS Catal.* **2021**, *11*, 11233−11247.
(50) Wang, Y.; Kalscheur, J.; Su, Y.-Q.; Hensen, E. J. M.; Vlachos, D. G. Real-time dynamics and structures of supported subnanometer catalysts via multiscale simulations. *Nat. Commun.* **2021**, *12*, 5430.
(51) Ouyang, M.; Papanikolaou, K. G.; Boubnov, A.; Hoffman, A. S.; Giannakakis, G.; Bare, S. R.; Stamatakis, M.; Flytzani-Stephanopoulos, M.; Sykes, E. C. H. Directing reaction pathways via in situ control of active site geometries in PdAu single-atom alloy catalysts. *Nat. Commun.* **2021**, *12*, 1549.
(52) Kress, P.; Réocreux, R.; Hannagan, R.; Thuening, T.; Boscoboinik, J. A.; Stamatakis, M.; Sykes, E. C. H. Mechanistic insights into carbon-carbon coupling on NiAu and PdAu single-atom alloys. *J. Chem. Phys.* **2021**, *154*, 204701.
(53) Reocreux, R.; Fampiou, I.; Stamatakis, M. The role of oxygenated species in the catalytic self-coupling of MeOH on O pre-covered Au(111). *Faraday Discuss.* **2021**, *229*, 251−266.
(54) Schumann, J.; Bao, Y.; Hannagan, R. T.; Sykes, E. C. H.; Stamatakis, M.; Michaelides, A. Periodic Trends in Adsorption Energies around Single-Atom Alloy Active Sites. *J. Phys. Chem. Lett.* **2021**, *12*, 10060−10067.
(55) Pe na-Torres, A.; Ali, A.; Stamatakis, M.; Jónsson, H. Indirect mechanism of Au adatom diffusion on the Si(100) surface. *Phys. Rev. B* **2022**, *105*, 205411.
(56) Barboun, P. M.; Otor, H. O.; Ma, H.; Goswami, A.; Schneider, W. F.; Hicks, J. C. Plasma-Catalyst Reactivity Control of Surface Nitrogen Species through Plasma-Temperature-Programmed Hydrogenation to Ammonia. *ACS Sustainable Chem. Eng.* **2022**, *10*, 15741−15748.
(57) Choung, S.; Kim, Y.; Moon, J.; Roh, J.; Hwang, J.; Han, J. W. Unveiling the catalyst deactivation mechanism in the non-oxidative dehydrogenation of light alkanes on Rh(111): Density functional theory and kinetic Monte Carlo study. *Catal. Today* **2023**, *411−412*, 113819.
(58) Li, K.; Li, X.; Li, L.; Chang, X.; Wu, S.; Yang, C.; Song, X.; Zhao, Z.-J.; Gong, J. Nature of Catalytic Behavior of Cobalt Oxides for CO2 Hydrogenation. *JACS Au* **2023**, *3*, 508−515.
(59) Zhang, Y.; Wang, B.; Fan, M.; Ling, L.; Zhang, R. Ethane dehydrogenation over the g-C3N4 supported metal single-atom catalysts to enhance reactivity and coking-resistance ability. *Nano Res.* **2023**, *16*, 6142−6152.
(60) Xu, L.; Papanikolaou, K. G.; Lechner, B. A. J.; Je, L.; Somorjai, G. A.; Salmeron, M.; Mavrikakis, M. Formation of active sites on transition metals through reaction-driven migration of surface atoms. *Science* **2023**, *380*, 70−76.
(61) Schmidt, D. J.; Chen, W.; Wolverton, C.; Schneider, W. F. Performance of cluster expansions of coverage-dependent adsorption of atomic oxygen on Pt(111). *J. Chem. Theory Comput.* **2012**, *8*, 264−273.
(62) Kumar, A.; Chatterjee, A. A probabilistic microkinetic modeling framework for catalytic surface reactions. *J. Chem. Phys.* **2023**, *158*, 024109.
(63) Ullmann, J. R. An Algorithm for Subgraph Isomorphism. *J. ACM* **1976**, *23*, 31−42.
(64) Jeliazkova, N.; Kochev, N. AMBIT-SMARTS: Efficient searching of chemical structures and fragments. *Mol. Inf.* **2011**, *30*, 707−720.
(65) Ohlrich, M.; Ebeling, C.; Ginting, E.; Sather, L. SubGemini: Identifying SubCircuits using a fast subgraph isomorphism algorithm. *Proc. 30th ACM/IEEE Des. Autom. Conf. Dallas*: TX: USA, 1993; pp 31−37.
(66) Ning, P.; Xu, D. Learning attack strategies from intrusion alerts. *Proc. 10th ACM Conf. Comput. Commun. Secur. CCS'03.*, 2003; pp 200−209.
(67) Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M. Performance Evaluation of the VF Graph Matching Algorithm. *Proc. 10th ICIAP*, 1999; Vol. *99*, p 1172.
(68) Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M. An improved algorithm for matching large graphs. *Proc. Third IAPR-TC15 Work. Graph-based Represent. Pattern Recognit.*, 2001; pp 149−159.
(69) Jüttner, A.; Madarasi, P. VF2++—An improved subgraph isomorphism algorithm. *Discret. Appl. Math.* **2018**, *242*, 69−81.
(70) Carletti, V.; Foggia, P.; Saggese, A.; Vento, M. Challenging the Time Complexity of Exact Subgraph Isomorphism for Huge and Dense Graphs with VF3. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 804−818.
(71) Carletti, V.; Foggia, P.; Greco, A.; Vento, M.; Vigilante, V. VF3-Light: A lightweight subgraph isomorphism algorithm and its experimental evaluation. *Pattern Recognit. Lett.* **2019**, *125*, 591−596.
(72) Bonnici, V.; Giugno, R.; Pulvirenti, A.; Shasha, D.; Ferro, A. A subgraph isomorphism algorithm and its application to biochemical data. *BMC Bioinf.* **2013**, *14*, S13.
(73) Mcgregor, J. J. Relational consistency algorithms and their application in finding subgraph and graph isomorphisms. *Inf. Sci.* **1979**, *19*, 229−250.
(74) Ullmann, J. R. Bit-vector algorithms for binary constraint satisfaction and subgraph isomorphism. *J. Exp. Algorithmics* **2010**, *15*, 1−6.
(75) Tam, C. Comparison of subgraph isomorphism algorithms with application to kinetic Monte Carlo simulations in catalysis. M.Sc. Thesis, University College London, 2018;.
(76) Yadavalli, S. S.; Jones, G.; Benson, R. L.; Stamatakis, M. Assessing the Impact of Adlayer Description Fidelity on Theoretical Predictions of Coking on Ni(111) at Steam Reforming Conditions. *J. Phys. Chem. C* **2023**, *127*, 8591−8606.

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319

(77) Laidler, K. J.; King, M. C. Development of transition-state theory. *J. Phys. Chem.* **1983**, *87*, 2657−2664.
(78) Hänggi, P.; Talkner, P.; Borkovec, M. Reaction-rate theory: Fifty years after Kramers. *Rev. Mod. Phys.* **1990**, *62*, 251−341.
(79) Stampfl, C.; Kreuzer, H. J.; Payne, S. H.; Pfnür, H.; Scheffler, M. First-principles theory of surface thermodynamics and kinetics. *Phys. Rev. Lett.* **1999**, *83*, 2993−2996.
(80) Stampfl, C. Predicting surface phase transitions from ab initio based statistical mechanics and thermodynamics. *Phase Transitions* **2007**, *80*, 311−332.
(81) Pineda, M.; Stamatakis, M. Beyond mean-field approximations for accurate and computationally efficient models of on-lattice chemical kinetics. *J. Chem. Phys.* **2017**, *147*, 024105.
(82) Shang, H.; Zhang, Y.; Lin, X.; Yu, J. X. Taming verification hardness: An efficient algorithm for testing subgraph isomorphism. *Proc. VLDB Endow.* **2008**, *1*, 364−375.
(83) Foggia, P.; Sansone, C.; Vento, M. A Performance Comparison of Five Algorithms for Graph Isomorphism. *Proc. 3rd IAPR-TC15 Work. Graph-Based Represent. Pattern Recognit*, 2001; pp 188−199.
(84) Lee, J.; Han, W. S.; Kasperovics, R.; Lee, J. H. An in-depth comparison of subgraph isomorphism algorithms in graph databases. *Proc. VLDB Endow.* **2012**, *6*, 133−144.
(85) Garey, M.; Johnson, D. *A Guide to the Theory of NP-Completeness*; Freeman and Company: New York, 1979; pp 18−29.
(86) January, C.; Byrd, J.; Oró, X.; O'Connor, M. *Allinea MAP: Adding Energy and OpenMP Profiling Without Increasing Overhead. Tools for High Performance Computing 2014*: Cham, 2015; pp 25−35.

https://doi.org/10.1021/acs.jpca.3c05581
J. Phys. Chem. A 2023, 127, 10307−10319