ORCID: 0000-0002-2590-6023
О. О. БОГАТИРЬОВ
кандидат фізико-математичних наук, доцент,
ННІ інформаційних та освітніх технологій,
Черкаський національний університет імені Богдана Хмельницького, Черкаси, Україна,
a.o.bogatyrev@gmail.com

DOI: 10.31651/2076-5851-2025-199-210 PACS 02.70.Ns, 02.70.-c, 07.05.Tp

АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ ДЛЯ ПОШУКУ НАЙБЛИЖЧИХ СУСІДІВ У МОЛЕКУЛЯРНІЙ ДИНАМІЦІ: ТОЧНІ МЕТОДИ В РЕАЛЬНОМУ ПРОСТОРІ

Анотація. Ефективний пошук найближчих сусідів є визначальним чинником продуктивності обчислювальної молекулярної динаміки (МД): саме на оцінювання локальних взаємодій у межах радіуса обрізання потенціалу припадає основна частка обчислювальних витрат симуляції, а безпосереднє попарне обчислення відстаней має квадратичну складність $O(N^2)$ і стає непридатним для систем реалістичного розміру. Метою статті є систематизація та порівняльний аналіз алгоритмів і структур даних для пошуку найближчих сусідів у молекулярній динаміці, а також оцінювання співвідношення їх обчислювальної ефективності та масштабованості. У роботі розмежовано два принципово різні режими такого пошуку. Перший – реальний тривимірний простір, у якому обчислюються міжатомні сили; для нього детально проаналізовано класичні структури даних: списки комірок (cell lists), списки Верле зі «шкіркою» та просторові дерева (KD-дерева, octree), що знижують складність від квадратичної до майже лінійної або $O(N \log N)$. Другий режим постав із поширенням машинно-навчених міжатомних потенціалів, де кожне локальне атомне оточення кодується багатовимірним вектором-дескриптором; у ньому пошук сусідів керує процедурами активного навчання, відбору репрезентативних конфігурацій та оцінювання невизначеності, а точні деревоподібні структури зазнають «прокляття розмірності», що зумовлює перехід до наближених методів (Ball-Trees, локально-чутливе хешування, HNSW). Як методичний внесок запропоновано обчислювальний експеримент на канонічній модельній системі – леннард-джонсівському флюїді – для кількісного зіставлення повного перебору, списків комірок, списків Верле та KD-дерева залежно від розміру системи, радіуса «шкірки» та густини. Експеримент підтвердив теоретичні оцінки складності (нахили 2.01 для перебору та 0.96–1.05 для масштабованих методів) і виявив різницю майже на три порядки між перебором і списком комірок за $N \approx 10^6$. Зроблено висновок, що в низьковимірному режимі реального простору структури на основі комірок і дерев є близькими до оптимальних і слугують базовою лінією для оцінювання високовимірного режиму, дослідження якого на реальному матеріалі є предметом окремої роботи.

Ключові слова: молекулярна динаміка; пошук найближчих сусідів; структури даних; список Верле; список комірок; KD-дерева; обчислювальна складність; леннард-джонсівський флюїд; масштабованість; алгоритми.

1. Вступ

Молекулярна динаміка (МД) є одним із фундаментальних методів обчислювальної та прикладної фізики, що дозволяє відтворювати еволюцію систем багатьох взаємодіючих частинок шляхом чисельного інтегрування рівнянь руху. У

фізиці твердого тіла та матеріалознавстві МД-моделювання є незамінним для дослідження структурних, механічних і теплових властивостей матеріалів та наноструктур на атомному масштабі [1, 2]. Практична застосовність методу до систем реалістичного розміру визначається передусім ефективністю одного критичного кроку обчислювального циклу – визначення множини сусідів кожної частинки в межах радіуса обрізання потенціалу, на яке припадає основна частка часу симуляції.

Безпосереднє обчислення відстаней між усіма парами частинок має обчислювальну складність $O(N^2)$ за кількістю частинок N, що унеможливлює його застосування до систем із мільйонів атомів. Класичним розв'язанням цієї проблеми, що сформувалося ще в ранніх роботах із комп'ютерного моделювання рідин, є просторове розбиття області на комірки (cell lists, linked-cell) та побудова списків Верле (Verlet lists) із додатковим буфером – «шкіркою» (skin radius), яка дозволяє оновлювати список не на кожному кроці інтегрування [3, 4]. Доповнені просторовими деревними структурами – KD-деревами та octree, особливо ефективними за неоднорідного розподілу густини [5, 6], – ці методи знижують складність пошуку до O(N) або O(N log N). Оскільки пошук відбувається у фізичному просторі низької (тривимірної) розмірності, зазначені структури даних є близькими до оптимальних, і задачу пошуку сусідів для розрахунку короткодіючих сил у класичній МД можна вважати алгоритмічно розв'язаною.

Водночас розвиток машинно-навчених міжатомних потенціалів (machine-learning interatomic potentials) докорінно розширив роль пошуку найближчих сусідів у МД. У таких підходах кожне локальне атомне оточення подається інваріантним вектором-дескриптором (симетрійні функції, SOAP, ACE) розмірністю в десятки й сотні ознак [9, 10, 15], унаслідок чого задача пошуку сусідів виникає вже не у тривимірному фізичному просторі, а у багатовимірному просторі дескрипторів, де вона керує процедурами активного навчання, відбору репрезентативних конфігурацій та оцінювання невизначеності прогнозу [14]. У цьому режимі точні деревоподібні структури зазнають «прокляття розмірності» [7], що зумовлює потребу в наближених методах пошуку – метричних деревах (Ball-Trees), локально-чутливому хешуванні [8] та графових структурах (HNSW) [13]. Таким чином, у сучасній МД співіснують два принципово різні режими пошуку найближчих сусідів, кожен зі своїм оптимальним класом структур даних.

У науковій літературі ці два режими, як правило, розглядаються відокремлено: одні роботи зосереджені на оптимізації списків сусідів для розрахунку сил [4, 12], інші – на структурах пошуку для машинного навчання потенціалів [10, 14], тоді як єдиного порівняльного аналізу, який пов'язував би обидва режими спільною мовою обчислювальної складності та емпіричної ефективності, бракує. Дана робота є першою частиною такого аналізу і присвячена низьковимірному режиму реального простору. Для нього, окрім систематизації відповідних алгоритмів і структур даних, наведено власний обчислювальний експеримент на канонічній модельній системі – леннард-джонсівському флюїді, – який кількісно зіставляє ефективність повного перебору, списків комірок, списків Верле та KD-дерева за зростання розміру системи.

Метою статті є систематизація та порівняльний аналіз алгоритмів і структур даних для пошуку найближчих сусідів у молекулярній динаміці в режимі реального простору, а також експериментальне оцінювання співвідношення їх обчислювальної ефективності та масштабованості. Стаття структурована так: у розділі 2 формалізовано задачу пошуку сусідів та метрики відстані з урахуванням періодичних граничних умов; у розділі 3 проаналізовано структури даних реального простору; у розділі 4 оглядово розглянуто високовимірний режим пошуку у просторі дескрипторів і наближені методи, повне експериментальне дослідження яких на реальному матеріалі винесено в окрему

роботу; у розділі 5 викладено методику обчислювального експерименту; у розділі 6 наведено та обговорено його результати; у розділі 7 сформульовано висновки.

2. Формалізація задачі пошуку найближчих сусідів

Нехай задано систему з N частинок із координатами r_i, i = 1, …, N, у кубічній комірці зі стороною L та періодичними граничними умовами. У молекулярній динаміці з короткодіючими потенціалами для кожної частинки і потрібно визначити множину її сусідів – усіх частинок j, відстань до яких не перевищує радіуса обрізання потенціалу r_c. Це задача пошуку в межах фіксованого радіуса (fixed-radius near neighbours). Відстань між частинками обчислюється за евклідовою метрикою з урахуванням конвенції мінімального образу (minimum image convention), за якої враховується найближчий з періодичних образів частинки:

$$r_{ij} = \left|\Delta r_{ij} - L \cdot round\left(\Delta r_{ij}/L\right)\right|,\ N(i) = \left\{j : r_{ij} \le r\_c\right\} \quad (1)$$

Поряд із пошуком у фіксованому радіусі в задачах причинного відбору конфігурацій та машинного навчання потенціалів використовують пошук k найближчих сусідів (k-nearest neighbours), у якому для кожного запиту повертається задана кількість найближчих об’єктів незалежно від абсолютної відстані. Принципова відмінність двох режимів, розглянутих у статті, полягає у розмірності простору пошуку: у реальному просторі вона дорівнює трьом (d = 3), тоді як у просторі дескрипторів атомного оточення вона сягає десятків і сотень ($d \gg 3$), що докорінно змінює ефективність застосовних структур даних. Наївний алгоритм, що обчислює всі попарні відстані, має складність $O(N^2)$ незалежно від розмірності й слугує базовою лінією для порівняння.

3. Структури даних реального простору

3.1. Списки комірок (cell lists)

Метод списків комірок (linked-cell) полягає у розбитті моделювальної комірки на регулярну сітку кубічних комірок зі стороною, не меншою за r_c. Кожна частинка приписується до своєї комірки за O(1) операцій, після чого потенційні сусіди частинки шукаються лише в її власній та 26 суміжних комірках (у тривимірному випадку). За однорідного розподілу густини середня кількість частинок у комірці є сталою, тож загальна складність побудови та обходу списку сусідів становить O(N), а потрібний обсяг пам’яті – O(N) [3, 4]. Це робить списки комірок практично оптимальним інструментом для однорідних систем. Псевдокод побудови списку наведено в алгоритмі 1, а схему сітки комірок із 3×3-околом частинки на рис. 1.

```
Вхід: {r_i}, i = 1..N;  розмір комірки s ≥ r_c;  розмір системи L.
Вихід: списки сусідів N(i) у радіусі r_c.
M ← floor(L / s)        // число комірок уздовж осі
head[1..M³] ← 0;  next[1..N] ← 0
for i ← 1 to N do       // розклад частинок по комірках
  c ← індекс комірки частинки i за r_i
  next[i] ← head[c];  head[c] ← i
for i ← 1 to N do       // формування списку сусідів
  for each c′ ∈ {комірка i та 26 суміжних} do
    j ← head[c′]
    while j ≠ 0 do
      if j ≠ i ∧ |r_i − r_j| ≤ r_c then  N(i) ← N(i) ∪ {j}
      j ← next[j]
```

Алгоритм 1. Побудова списку комірок (linked-cell) [3, 4].

Algorithm 1. Construction of the cell (linked-cell) list.

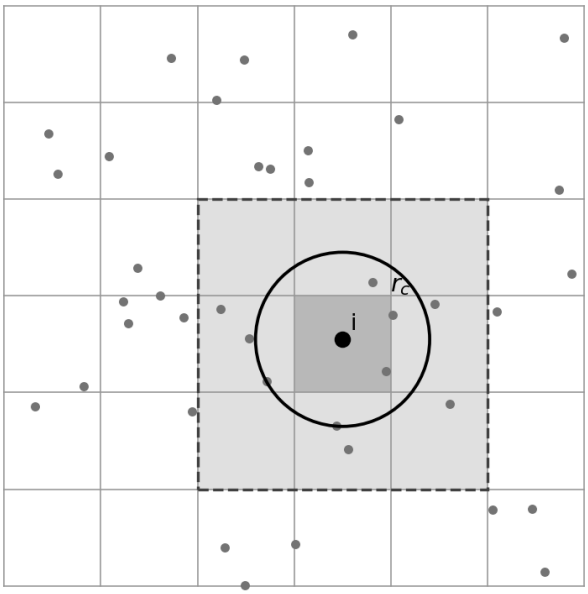

Рис. 1. Сітка списку комірок [3, 4]: мічена частинка i, її «домашня» комірка та 3×3-окіл (штрихова межа); коло – радіус обрізання r_c.
Fig. 1. Cell-list grid: tagged particle i, its home cell and the 3×3 neighbourhood (dashed boundary); the circle is the cutoff radius r_c.

3.2. Списки Верле зі «шкіркою»

Список Верле зберігає для кожної частинки перелік сусідів у межах розширеного радіуса r_list = r_c + r_skin, де r_skin – буферна «шкірка». Оскільки за один крок інтегрування частинки зміщуються незначно, той самий список повторно використовується протягом багатьох кроків і перебудовується лише тоді, коли сумарне зміщення найбільш рухомих частинок може порушити його коректність (типовий критерій – переміщення частинки більш ніж на r_skin/2). Список зазвичай будується за допомогою списку комірок, що дає амортизовану складність O(N) на крок. Вибір r_skin є компромісом: більша «шкірка» зменшує частоту перебудов, але збільшує число пар, які перевіряються на кожному кроці [4]. Відповідну процедуру оновлення подано в алгоритмі 2, а внутрішнє влаштування обох списків як структур даних – на рис. 2.

Вхід: {r_i}; r_c; r_skin; збережені позиції {r_i⁰} з останньої перебудови.
Параметр: r_list ← r_c + r_skin.
1. d_max ← max_i |r_i − r_i⁰| // макс. зміщення з перебудови
2. if 2·d_max > r_skin then // критерій перебудови
3. перебудувати список Верле:
4. for each i: L(i) ← { j : |r_i − r_j| ≤ r_list } // через алгоритм 1
5. r_i⁰ ← r_i для всіх i
6. for i ← 1 to N do // крок інтегрування
7. for each j ∈ L(i) do
8. if |r_i − r_j| ≤ r_c then обчислити взаємодію (i, j)

Алгоритм 2. Оновлення списку Верле зі «шкіркою» [4].
Algorithm 2. Update of the Verlet list with a skin radius.

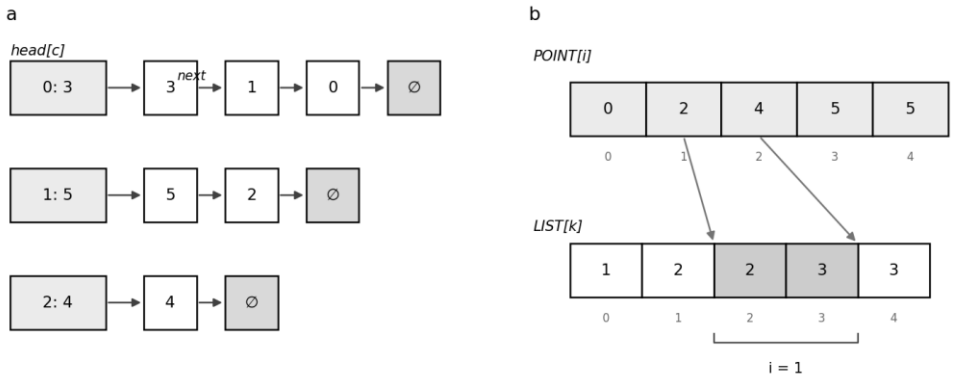

Рис. 2. Структури даних списків сусідів [3, 4]: а – список комірок як зв'язні ланцюжки head[c] → next; б – список Верле як масиви POINT/LIST (перелік сусідів частинки i).
Fig. 2. Data structures of neighbour lists: a – cell list as head[c] → next linked chains; b – Verlet list as POINT/LIST arrays (neighbour list of particle i).

3.3. KD-дерева та octree

KD-дерево – це збалансоване бінарне дерево, яке рекурсивно розбиває простір гіперплощинами, перпендикулярними до координатних осей [5, 6]. Його побудова потребує O(N log N) операцій, а запит на пошук у радіусі або k найближчих сусідів у низьковимірному просторі виконується в середньому за O(log N). На відміну від рівномірних комірок, деревоподібні структури адаптуються до локальної густини, тому є ефективнішими для систем із суттєво неоднорідним розподілом частинок (поверхні, пори, кластери). Споріднена структура – octree – лежить в основі деревних кодів типу Barnes–Hut, що застосовуються для далекодіючих взаємодій. Приклад розбиття простору KD-деревом та відповідне бінарне дерево показано на рис. 3. Зведене порівняння розглянутих структур наведено в таблиці 1.

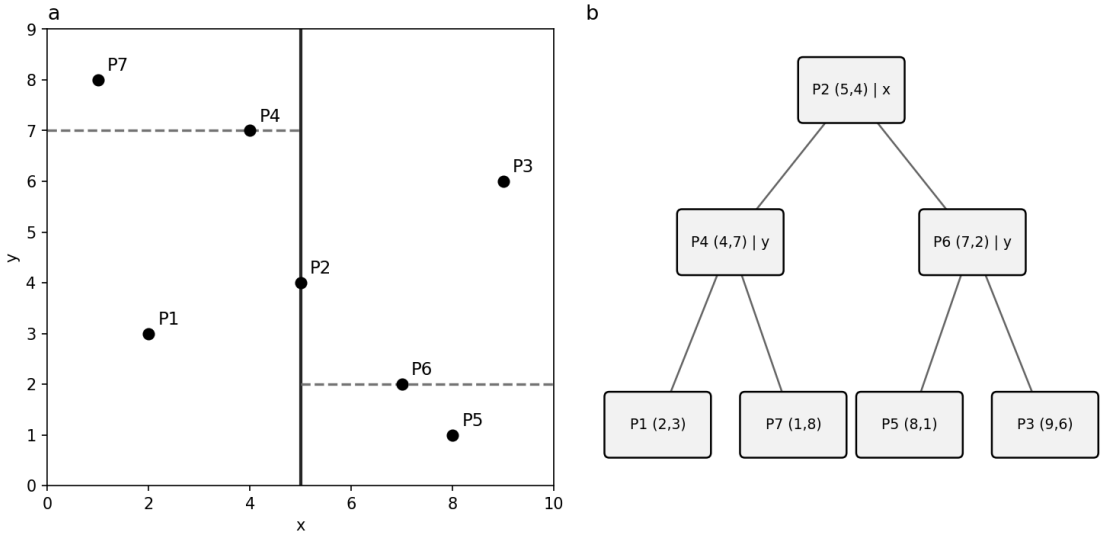

Рис. 3. KD-дерево [5, 6]: а – рекурсивне розбиття площини; б – відповідне бінарне дерево (у вузлах – точка та вісь поділу).
Fig. 3. KD-tree: a – recursive partition of the plane; b – the corresponding binary tree (nodes show the point and the splitting axis).

Таблиця 1
Table 1
Порівняння структур даних для пошуку найближчих сусідів у реальному просторі
Comparison of real-space nearest neighbour search data structures

Структура даних	Побудова	Запит	Пам'ять	Примітка
Повний перебір	–	$O(N^2)$	O(1)	базовий варіант; лише малі N
Список комірок	O(N)	O(N)	O(N)	оптимальний за однорідної густини
Список Верле	O(N) (перебудова)	O(N) аморт.	$O(N\cdot\bar{n})$	потребує налаштування «шкірки»
KD-дерево	O(N log N)	O(log N) сер.	O(N)	стійке до неоднорідної густини
octree / деревні коди	O(N log N)	O(log N)	O(N)	основа методів Barnes–Hut

4. Високовимірний режим пошуку у просторі дескрипторів

У машинно-навчених міжатомних потенціалах локальне оточення кожного атома в межах радіуса обрізання подається інваріантним відносно трансляцій, обертань і перестановок вектором-дескриптором – набором симетрійних функцій Белера–Паррінелло, дескриптором SOAP або розкладом ACE [9, 10, 15]. Унаслідок цього атомне оточення стає точкою у просторі ознак розмірністю в десятки й сотні, і задача пошуку найближчих сусідів переноситься з тривимірного фізичного простору в цей високовимірний простір. Там вона вже не обслуговує розрахунок сил, а керує іншими процедурами: активним навчанням потенціалів і відбором максимально інформативних конфігурацій (зокрема методом найвіддаленішої точки, farthest-point sampling), оцінюванням невизначеності прогнозу та аналізом подібності атомних оточень [14].

У цьому режимі ефективність точних деревоподібних структур стрімко падає: зі зростанням розмірності d KD-дерево вироджується, наближаючись за вартістю запиту до повного перебору – явище, відоме як «прокляття розмірності» [7]. Тому замість точного пошуку застосовують наближені методи: метричні дерева (Ball-Trees), стійкіші до багатовимірності; локально-чутливе хешування (LSH) на базі p-стабільних розподілів, що групує близькі вектори в спільні «кошики» [8]; та графові структури типу HNSW, які наразі забезпечують найкращий баланс повноти й швидкодії в задачах наближеного пошуку [13]. Систематичне експериментальне дослідження цього високовимірного режиму на реальному матеріалі та його застосування до машинного навчання потенціалів виходять за межі даної роботи і є предметом окремого дослідження.

5. Методика обчислювального експерименту

Для кількісного зіставлення структур даних реального простору обрано канонічну модельну систему молекулярної динаміки – леннард-джонсівський (ЛД) флюїд. Усі величини подаються у зведених одиницях (відстань – у σ, енергія – у ε, температура $T^* = k_B\cdot T/\varepsilon$, густина $\rho^* = \rho\cdot\sigma^3$). Парний потенціал ЛД з обрізанням на $r_c = 2.5\sigma$ має вигляд:

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \quad r \le r\_c \tag{2}$$

Конфігурації згенеровано в кубічній комірці з періодичними граничними умовами як гранецентровану кубічну (ГЦК) ґратку з невеликим випадковим збуренням положень частинок за густини $\rho^* = 0.8$; такий невпорядкований розподіл є достатнім для оцінювання вартості пошуку сусідів, яка визначається передусім розміром системи, густиною та радіусом обрізання. Для строгіших фізичних постановок зрівноважені знімки можна отримати стандартними пакетами молекулярної динаміки (наприклад, LAMMPS [11]). Розміри систем обираються логарифмічно рознесеними в діапазоні від $N \approx 5\cdot10^2$ до $N \approx 10^6$ (чотирнадцять значень); повний перебір $O(N^2)$ оцінюється лише до $N \approx 10^4$.

Порівнюються чотири способи побудови списку сусідів у радіусі r_c: (1) повний перебір; (2) список комірок із розміром комірки, не меншим за r_c + r_skin; (3) список Верле зі «шкіркою» r_skin = 0.3σ і критерієм перебудови за зміщенням понад r_skin/2; (4) KD-дерево з періодичними образами частинок (ghost particles). Вимірюваними величинами є: час побудови списку сусідів як функція N (у подвійному логарифмічному масштабі з підгонкою емпіричного показника складності); обсяг пам’яті як функція N; амортизована вартість на крок МД як функція r_skin (для визначення оптимальної «шкірки»); а також середнє число сусідів на частинку для контролю фізичної коректності. Обчислення виконуються однопотоково на одному процесорному вузлі; кожен вимір подається як медіана з кількох запусків за фіксованого початкового стану генератора випадкових чисел.

Для коректності порівняння всі чотири методи реалізовано однією мовою (C) на однаковому рівні абстракції: це дозволяє зіставляти саме обчислювальну поведінку структур даних, а не накладні витрати різних мов програмування чи готових бібліотек. Високорівневу частину – генерування конфігурацій, керування прогонами, підгонку показників складності та побудову рисунків – винесено в окремий скрипт (Python), що не бере участі у вимірюваних ділянках коду. Час фіксується монотонним годинником високої роздільності лише на фазі пошуку сусідів (без вводу-виводу та виділення пам’яті), із фіксованими прапорами компілятора. Коректність усіх реалізацій перевірено зіставленням однакової сумарної кількості знайдених сусідів, отриманої кожним методом.

6. Результати та обговорення

Обчислювальний експеримент проведено для конфігурацій ЛД-флюїду густини $\rho^* = 0.8$ за розмірів системи від $N = 5\cdot10^2$ до $N \approx 10^6$. У всіх точках усі чотири методи повернули однакову сумарну кількість знайдених сусідів, що підтверджує коректність реалізацій. Виміряний час побудови списку сусідів наведено в таблиці 2, а його залежність від N у подвійному логарифмічному масштабі – на рис. 4, а.

Підгонка емпіричних показників складності дала такі нахили: для повного перебору – 2.01, для списку комірок – 0.96, для списку Верле – 0.99, для KD-дерева – 1.05. Ці значення з високою точністю відтворюють теоретичні оцінки таблиці 1: квадратичну складність $O(N^2)$ перебору, лінійну $O(N)$ списків комірок і Верле та $O(N \log N)$ KD-дерева (нахил, дещо більший за одиницю, відповідає логарифмічному множнику, що проявляється саме на великих N).

Криві демонструють очікуване перехрестя: повний перебір є найшвидшим лише для малих систем ($N \lesssim 10^3$) завдяки мінімальному константному множнику, проте вже від $N \approx 1.4\cdot10^3$ його випереджає список комірок. На великих N квадратична складність унеможливлює застосування перебору: за $N \approx 10^6$ його час сягає $\approx 1.6\cdot10^3$ с (близько 26

хв) проти ≈ 1.6 с для списку комірок – різниця майже на три порядки, що наочно ілюструє «обчислювальний бар'єр» класичних методів.

Серед масштабованих методів найефективнішим є список комірок: за $N \approx 10^6$ він приблизно у 3.7 раза швидший за KD-дерево. Більший константний множник списків Верле та KD-дерева пояснюється відповідно побудовою переліку в розширеному радіусі r_list = r_c + r_skin за два проходи та витратами на побудову дерева й запити з періодичними образами. Оскільки досліджувана система є просторово однорідною, перевага деревоподібних структур за неоднорідного розподілу густини тут не виявляється.

Залежність амортизованої вартості на крок від радіуса «шкірки» (рис. 4, б) має характерний мінімум поблизу r_skin ≈ 0.4σ: за надто малої «шкірки» зростає частота перебудов списку, за надто великої – кількість зайвих кандидатів, що перевіряються на кожному кроці. Отримане значення узгоджується з класичною рекомендацією r_skin ≈ 0.3σ; його точне положення залежить від інтенсивності руху частинок і є орієнтовним.

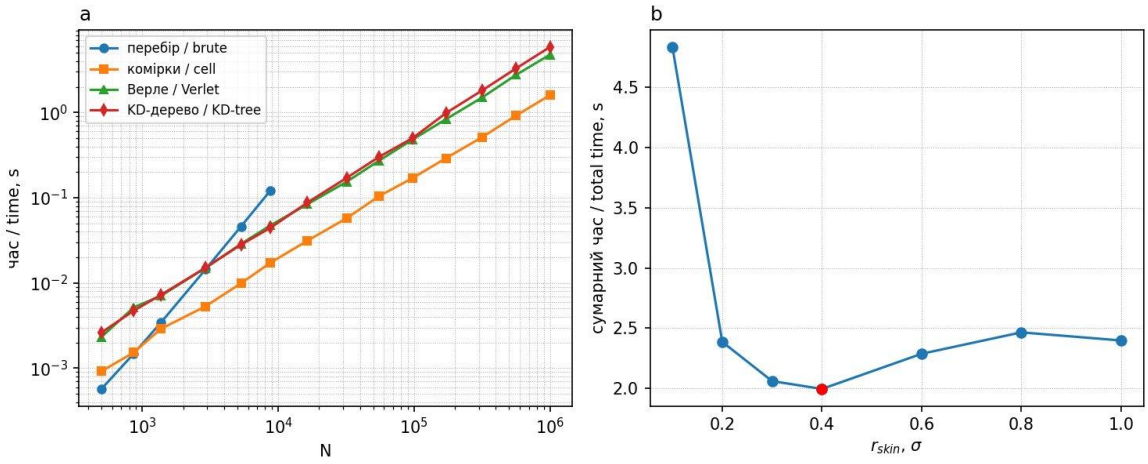

Рис. 4. Масштабування часу пошуку сусідів: а – час побудови списку від N; б – амортизована вартість на крок від радіуса «шкірки».

Fig. 4. Scaling of neighbour search time: a – list construction time vs N; b – amortised cost per step vs skin radius.

Таблиця 2
Table 2

Час побудови списку сусідів (с) для різних структур даних
Neighbour-list construction time (s) for different data structures

N	Перебір / Brute force	Комірки / Cell list	Верле / Verlet	KD-дерево / KD-tree
500	0.000572	0.000923	0.00233	0.00262
2916	0.0144	0.00532	0.0150	0.0151
8788	0.122	0.0173	0.0475	0.0444
32000	1.57	0.0577	0.154	0.171
97556	14.5	0.170	0.481	0.500
318028	156	0.515	1.50	1.82
1000188	1574	1.60	4.77	5.85

7. Висновки

Проведена систематизація дозволяє розмежувати два принципово різні режими пошуку найближчих сусідів у молекулярній динаміці – низьковимірний у реальному просторі та високовимірний у просторі дескрипторів атомного оточення, – кожен зі своїм оптимальним класом структур даних. У режимі реального простору структури на основі комірок (cell lists, списки Верле) та просторових дерев (KD-дерева, octree) знижують обчислювальну складність пошуку від квадратичної до майже лінійної або O(N log N) і є близькими до оптимальних; при цьому списки комірок найефективніші за однорідної густини, тоді як деревоподібні структури мають перевагу за суттєво неоднорідного розподілу частинок.

Обчислювальний експеримент на ЛД-флюїді кількісно підтвердив цю ієрархію: емпіричні показники складності (2.01 для перебору та 0.96–1.05 для масштабованих методів) з високою точністю відтворюють теоретичні оцінки. Квадратична складність робить повний перебір непридатним уже за $N \approx 10^6$ ($\approx$ 26 хв проти $\approx$ 1.6 с у списку комірок), тоді як найефективнішим методом для просторово однорідної системи виявився список комірок, а оптимальний радіус «шкірки» списку Верле для досліджуваних умов становить близько 0.4σ.

Перспективи подальших досліджень. Встановлений у даній роботі низьковимірний режим слугує базовою лінією для дослідження високовимірного режиму пошуку у просторі дескрипторів, де точні структури зазнають «прокляття розмірності». Експериментальне порівняння точних і наближених методів (Ball-Trees, LSH, HNSW) на реальному матеріалі з використанням потенціалу зануреного атома для міді та дескрипторів атомного оточення є предметом окремої роботи, що становить безпосереднє продовження даного дослідження.

Список використаної літератури

1. Allen, M. P., & Tildesley, D. J. (2017). Computer Simulation of Liquids (2nd ed.). Oxford University Press. Режим доступу https://doi.org/10.1093/oso/9780198803195.001.0001
2. Frenkel, D., & Smit, B. (2002). Understanding Molecular Simulation: From Algorithms to Applications (2nd ed.). Academic Press. Режим доступу https://shop.elsevier.com/books/understanding-molecular-simulation/frenkel/978-0-12-267351-1
3. Verlet, L. (1967). Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Physical Review, 159(1), 98–103. Режим доступу https://doi.org/10.1103/PhysRev.159.98
4. Hockney, R. W., & Eastwood, J. W. (1988). Computer Simulation Using Particles. Adam Hilger. Режим доступу https://doi.org/10.1201/9780367806934
5. Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509–517. Режим доступу https://doi.org/10.1145/361002.361007
6. Friedman, J. H., Bentley, J. L., & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software, 3(3), 209–226. Режим доступу https://doi.org/10.1145/355744.355745
7. Weber, R., Schek, H.-J., & Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. Proceedings of the 24th VLDB Conference, 194–205. Режим доступу https://dblp.org/rec/conf/vldb/WeberSB98.html
8. Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the 20th Annual

Symposium on Computational Geometry, 253–262. Режим доступу https://doi.org/10.1145/997817.997857
9. Behler, J., & Parrinello, M. (2007). Generalized neural-network representation of high-dimensional potential-energy surfaces. Physical Review Letters, 98(14), 146401. Режим доступу https://doi.org/10.1103/PhysRevLett.98.146401
10. Bartók, A. P., Payne, M. C., Kondor, R., & Csányi, G. (2010). Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. Physical Review Letters, 104(13), 136403. Режим доступу https://doi.org/10.1103/PhysRevLett.104.136403
11. Thompson, A. P., et al. (2022). LAMMPS – a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. Computer Physics Communications, 271, 108171. Режим доступу https://doi.org/10.1016/j.cpc.2021.108171
12. Winkler, D., Rezavand, M., & Rauch, W. (2018). Neighbour lists for smoothed particle hydrodynamics on GPUs. Computer Physics Communications, 225, 140–148. Режим доступу https://doi.org/10.1016/j.cpc.2017.12.014
13. Malkov, Yu. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824–836. Режим доступу https://doi.org/10.1109/TPAMI.2018.2889473
14. Podryabinkin, E. V., & Shapeev, A. V. (2017). Active learning of linearly parametrized interatomic potentials. Computational Materials Science, 140, 171–180. Режим доступу https://doi.org/10.1016/j.commatsci.2017.08.031
15. Musil, F., Grisafi, A., Bartók, A. P., Ortner, C., Csányi, G., & Ceriotti, M. (2021). Physics-inspired structural representations for molecules and materials. Chemical Reviews, 121(16), 9759–9815. Режим доступу https://doi.org/10.1021/acs.chemrev.1c00021

References

1. Allen, M. P., & Tildesley, D. J. (2017). Computer Simulation of Liquids (2nd ed.). Oxford University Press. Retrieved from https://doi.org/10.1093/oso/9780198803195.001.0001
2. Frenkel, D., & Smit, B. (2002). Understanding Molecular Simulation: From Algorithms to Applications (2nd ed.). Academic Press. Retrieved from https://shop.elsevier.com/books/understanding-molecular-simulation/frenkel/978-0-12-267351-1
3. Verlet, L. (1967). Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Physical Review, 159(1), 98–103. Retrieved from https://doi.org/10.1103/PhysRev.159.98
4. Hockney, R. W., & Eastwood, J. W. (1988). Computer Simulation Using Particles. Adam Hilger. Retrieved from https://doi.org/10.1201/9780367806934
5. Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9), 509–517. Retrieved from https://doi.org/10.1145/361002.361007
6. Friedman, J. H., Bentley, J. L., & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software, 3(3), 209–226. Retrieved from https://doi.org/10.1145/355744.355745
7. Weber, R., Schek, H.-J., & Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. Proceedings of the 24th

VLDB Conference, 194–205. Retrieved from https://dblp.org/rec/conf/vldb/WeberSB98.html
8. Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the 20th Annual Symposium on Computational Geometry, 253–262. Retrieved from https://doi.org/10.1145/997817.997857
9. Behler, J., & Parrinello, M. (2007). Generalized neural-network representation of high-dimensional potential-energy surfaces. Physical Review Letters, 98(14), 146401. Retrieved from https://doi.org/10.1103/PhysRevLett.98.146401
10. Bartók, A. P., Payne, M. C., Kondor, R., & Csányi, G. (2010). Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. Physical Review Letters, 104(13), 136403. Retrieved from https://doi.org/10.1103/PhysRevLett.104.136403
11. Thompson, A. P., et al. (2022). LAMMPS – a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. Computer Physics Communications, 271, 108171. Retrieved from https://doi.org/10.1016/j.cpc.2021.108171
12. Winkler, D., Rezavand, M., & Rauch, W. (2018). Neighbour lists for smoothed particle hydrodynamics on GPUs. Computer Physics Communications, 225, 140–148. Retrieved from https://doi.org/10.1016/j.cpc.2017.12.014
13. Malkov, Yu. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824–836. Retrieved from https://doi.org/10.1109/TPAMI.2018.2889473
14. Podryabinkin, E. V., & Shapeev, A. V. (2017). Active learning of linearly parametrized interatomic potentials. Computational Materials Science, 140, 171–180. Retrieved from https://doi.org/10.1016/j.commatsci.2017.08.031
15. Musil, F., Grisafi, A., Bartók, A. P., Ortner, C., Csányi, G., & Ceriotti, M. (2021). Physics-inspired structural representations for molecules and materials. Chemical Reviews, 121(16), 9759–9815. Retrieved from https://doi.org/10.1021/acs.chemrev.1c00021

O. O. BOGATYREV
Candidate of Physical and Mathematical Sciences, Associate Professor,
Educational-Scientific Institute of Informational and Educational Technologies,
Bohdan Khmelnytsky National University at Cherkasy, Cherkasy, Ukraine,
a.o.bogatyrev@gmail.com

ALGORITHMS AND DATA STRUCTURES FOR NEAREST NEIGHBOUR SEARCH IN MOLECULAR DYNAMICS: EXACT METHODS IN REAL SPACE

DOI: 10.31651/2076-5851-2025-199-210 PACS 02.70.Ns, 02.70.-c, 07.05.Tp

Efficient nearest neighbour search is a decisive factor in the performance of computational molecular dynamics (MD), since the evaluation of local interactions within the potential cutoff radius accounts for the dominant share of the computational cost of a simulation, while the direct evaluation of all pairwise distances scales quadratically with the number of particles and becomes intractable for systems of realistic size. The aim of this article is to systematise and comparatively analyse the algorithms and data structures used for nearest

neighbour search in MD and to assess the relationship between their computational efficiency and scalability. The study distinguishes two fundamentally different regimes of nearest neighbour search in contemporary MD. The first concerns the three-dimensional real space in which interatomic forces are computed; here the classical data structures are analysed in detail, namely cell (linked) lists, Verlet neighbour lists with a skin radius, and spatial trees such as KD-trees and octrees, which reduce the algorithmic complexity from quadratic to near-linear or O(N log N) order and are, in low dimensionality, close to optimal. The second regime has emerged with the proliferation of machine-learning interatomic potentials, in which each local atomic environment is encoded as a high-dimensional descriptor vector; in this regime nearest neighbour search underpins active-learning workflows, uncertainty quantification, farthest-point sampling of representative configurations, and the assessment of environment similarity, and exact tree-based structures lose efficiency owing to the curse of dimensionality, motivating approximate methods such as metric trees (Ball-Trees), locality-sensitive hashing based on p-stable distributions, and graph-based structures (HNSW). As the methodological contribution, a computational experiment on the canonical Lennard-Jones fluid is designed to quantitatively compare brute-force search, cell lists, Verlet lists, and a KD-tree as functions of system size, skin radius, and number density. The measured complexity exponents (2.01 for brute force and 0.96-1.05 for the scalable methods) closely reproduce the theoretical estimates, with a gap of nearly three orders of magnitude between brute force and cell lists at the largest system size (N about 10^6). The study concludes that, in the low-dimensional real-space regime, cell- and tree-based structures are close to optimal and provide a baseline against which the high-dimensional regime, the subject of a companion study on a real material, is to be assessed.

Keywords: molecular dynamics; nearest neighbour search; data structures; Verlet list; cell list; KD-trees; computational complexity; Lennard-Jones fluid; scalability; algorithms.

Одержано редакцією 09.11.2025
Прийнято до друку 17.12.2025

Опубліковано 24.12.2025