

ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК 519.688/544

**А. В. МЫШЛЯВЦЕВ
П. В. СТИШЕНКО**

Омский государственный
технический университет

МОДИФИКАЦИЯ АЛГОРИТМА МЕТРОПОЛИСА ДЛЯ МОДЕЛИРОВАНИЯ МЕТАЛЛИЧЕСКИХ НАНОЧАСТИЦ

В данной работе мы предлагаем ряд модификаций алгоритма Метрополиса, повышающих его эффективность применительно к задаче о поиске равновесной формы металлических наночастиц. Наша реализация предлагаемого алгоритма, позволила найти равновесную форму наночастицы из 4033 атомов за 2 часа вычислений на современном микропроцессоре при использовании неаддитивных потенциалов.

Ключевые слова: алгоритм Метрополиса, металлические наночастицы, неаддитивный потенциал.

Введение

Уникальные свойства наночастиц (далее НЧ) известны достаточно давно, но обнаруженная в 1989 году повышенная каталитическая активность НЧ золота [1] вызвала ещё больший интерес к исследованиям в данной области. Благодаря своим особым, по сравнению со сплошными поверхностями, свойствам, НЧ нашли применение в фотонике, электронике, медицине, разработке химических сенсоров [2]. В силу малого размера их исследование экспериментальным

путем весьма сложно и дорого, и не всегда возможно напрямую измерить представляющие интерес показатели и свойства процессов, протекающих с участием НЧ. Поэтому особенно важной частью исследований в этой области является компьютерное моделирование. Огромное количество работ, в которых с помощью компьютерного моделирования изучались НЧ различных типов, подтверждает наличие потребности в таком виде исследований [3]. Обзор этих работ показывает, что большой интерес представляет форма НЧ в состоянии термодинамического

равновесия, в значительной степени определяющая их особые свойства. При моделировании НЧ применяются методы молекулярной динамики, теории функционала плотности (далее DFT), генетические алгоритмы, а также алгоритмы типа Метрополиса, основанные на использовании метода Монте-Карло, как для расчета равновесного состояния, так и кинетики исследуемых систем. Несмотря на потенциальную мощность методов молекулярной динамики, их применение ограничено системами, быстро достигающими состояния термодинамического равновесия, поскольку время моделирования напрямую зависит от времени протекания процесса в реальности. Поиск состояния с минимальной энергией для НЧ, состоящих из десятков или сотен атомов, выполняется в рамках DFT. Но интерес представляют НЧ размером до тысяч атомов. В этом случае чаще всего применяется алгоритм Метрополиса и его модификации, иногда в комбинации с генетическими алгоритмами.

С момента опубликования оригинального алгоритма Метрополиса [4] появилось множество его модификаций для различных молекулярных систем. Описание многих из них можно найти в учебниках [5–7]. Применение той или иной модификации может в разы изменить скорость моделирования, сделав его доступным для интересующих исследователя систем. Несмотря на большую практику применения алгоритмов типа Метрополиса для моделирования металлических НЧ и других подобных объектов (например, нанопроволок), насколько нам известно, на сегодня нет общепринятого способа реализации алгоритма, который бы использовался для данного типа молекулярных систем. Также нельзя сказать, что существует какой-то программный код, который используется в большинстве таких работ. Обычно приводится лишь беглое описание примененного метода, указывается используемый потенциал межатомного взаимодействия и, иногда, ограничение предельного радиуса взаимодействий. Наблюдается явный недостаток работ с описанием алгоритма достаточно детальным, чтобы можно было оценить его производительность. Это затрудняет оценку практической применимости других модификаций, поскольку неясно, в комбинации с какими структурами данных и алгоритмическими приемами их предстоит использовать.

В данной работе мы приводим описание модификации алгоритма Метрополиса для НЧ с детальным описанием ключевых особенностей алгоритма, оказывающих наибольшее влияние на его производительность. В предлагаемом алгоритме используются как известные подходы к ускорению алгоритма, так и разработанные нами приёмы, которые, насколько нам известно, ранее не описывались в литературе.

Алгоритм Метрополиса для металлических наночастиц

Классический алгоритм Метрополиса состоит из следующих шагов:

1. Взять систему в некотором случайном состоянии.
2. Обозначить текущее состояние системы как i . В соответствии с базовой матрицей марковского процесса выбрать некоторое состояние j .
3. Вычислить разницу энергии системы ΔE_{ij} в состояниях i и j .
4. С вероятностью $\exp(-\Delta E_{ij}/k_B T)$, где k_B — постоянная Больцмана, а T — температура системы, перевести систему в состояние j . Перейти к пункту 2.

Каждый шаг этого алгоритма можно реализовать различными способами, в зависимости от требований, предъявляемых к алгоритму, и характеристик исследуемых систем и процессов. Очевидно, что на первом шаге не много возможностей по модификации алгоритма. Второй шаг алгоритма полностью определяется базовой матрицей марковского процесса. Далее попытку перехода из текущего состояния в новое будем называть пробным шагом.

В решеточной модели возможны три варианта пробных шагов для моделирования диффузии атомов: прыжок в случайную точку, прыжок в соседнюю ячейку (в ближайшую или находящуюся в пределах определенной координационной сферы) и прыжки между случайными элементами списков активных атомов и вакансий. В первом случае велика вероятность попытки выполнить переход в ячейку решетке не имеющую рядом с собой никаких атомов. Поскольку вероятность успеха такой попытки очень мала, это сильно снижает долю успешных шагов. В случае прыжков в пределах некоторой сферы шанс на заведомо неудачную попытку гораздо меньше, но ограничение максимального расстояния перемещения атома может осложнить обход фазового пространства из-за «застреваний» в областях локальных минимумов энергии или метастабильных состояний. Используя динамические списки активных атомов и вакансий можно избежать недостатков обоих подходов. При таком подходе нужно учитывать, что ведение этих списков требует дополнительных вычислительных затрат, которые тем чувствительнее, чем выше процент успешных шагов, поскольку актуализация списков должна производиться после каждого успешного шага. Кроме того, в этом случае важно следить за соблюдением условия детального баланса. Поскольку число активных атомов и вакансий меняется после каждого успешного шага, базовая матрица теряет свою симметричность. Чтобы компенсировать это нарушение детального баланса, на шаге 4 используется смещенная вероятность принятия пробного шага. Например, в работе [8] для этого применялось следующее выражение:

$$p = \min\left(1, \left(\frac{N_i^a N_i^b}{N_r^a N_r^b}\right)^2 \exp\left(\frac{-\Delta E}{k_B T}\right)\right), \quad (1)$$

где $N_{i(f)}^a$ — количество поверхностных атомов до (i) и после (f) пробного шага, $N_{i(f)}^b$ — соответствующее количество активных вакансий, ΔE — изменение энергии системы, вызываемое пробным шагом, k_B — постоянная Больцмана, T — температура системы.

Насколько нам известно, сегодня нет общепринятого и описанного в литературе способа ведения списков активных атомов и вакансий. В тоже время при моделировании ведется активная работа с этими списками и выборка эффективного контейнера для этих целей значительно влияет на скорость моделирования. При моделировании выполняются следующие операции со списками активных атомов и вакансий:

- выбор случайного элемента на каждом пробном шаге;
- вставка и удаление элементов после каждого успешного шага.

Массив, дерево или список имеют сложность не меньше $O(\log N)$ как минимум для одной из этих операций. Для решеточной модели возможно обеспечить константную сложность всех операций. В нашей реализации алгоритма этот контейнер состоит из двух массивов R и L и индекса TAIL. Массив R

имеет заведомо достаточную длину, чтобы вместить все активные атомы (вакансии). Массив L является развёрткой кристаллической решётки. Каждый элемент массива L содержит либо индекс элемента в массиве R , либо -1 , означающее, что в данной ячейке решётки нет активного атома. Массив R содержит индексы элементов массива L , которые могут быть легко преобразованы в декартовы координаты. При этом соблюдается условие, что $R[i]=j$, если и только если $L[j]=i$. В массиве R элементы, ссылающиеся на активные атомы (вакансии) решётки, расположены непрерывно от начала массива и до элемента, на который указывает TAIL. Остальные элементы массива не используются. Когда в ходе моделирования число активных атомов меняется, TAIL сдвигается в ту или другую сторону.

В предлагаемой структуре данных выбор случайного элемента списка осуществляется генерацией случайного числа от 1 до TAIL, а удаление и вставка по заданным координатам выполняется сначала в массиве L , с последующей коррекцией массива R и индекса TAIL. Очевидно, что все требуемые операции в таком контейнере имеют сложность $O(1)$.

На следующем шаге алгоритма Метрополиса, вычисляется изменение энергии системы, вызываемое выбранным ранее пробным шагом. Расчет изменения потенциальной энергии в результате пробного шага это самый или, один из самых затратных с вычислительной точки зрения шагов алгоритма, поэтому ему необходимо уделять особое внимание.

Вычисление энергии системы может осуществляться с использованием различных эмпирических потенциалов межатомного взаимодействия, которые выбираются в зависимости от характеристик моделируемой системы, требуемой точности, от исследуемых явлений, и доступных вычислительных ресурсов. В некоторых случаях можно использовать простые парные потенциалы, такие как потенциал Леннарда-Джонса. Несмотря на свою простоту, такие потенциалы успешно используются для моделирования благородных газов, и даже некоторых свойств металлических НЧ [9].

С точки зрения компьютерного моделирования, важной особенностью таких потенциалов является их аддитивный характер. Это дает возможность легко получить изменение энергии системы, вызванное пробным шагом. Выражение полной энергии в общем виде для таких потенциалов выглядит так:

$$E_T = \sum_{i \in \Psi} \sum_{j \in \Psi_i} E(r_{ij}), \quad (2)$$

где Ψ — множество всех атомов системы, Ψ_i — множество атомов в окрестности атома i , r_{ij} — расстояние между атомами i и j , E — функция энергии парного взаимодействия. Здесь и далее под окрестностью атома понимается пространство в пределах некоторого расстояния называемого радиусом отсечки. Очевидно, что изменение E_T можно вычислить как разницу между $E(r_{ij})$ для тех i и j , для которых изменилось расстояние r_{ij} .

Присущая аддитивным потенциалам модель парных взаимодействий плохо описывает такие свойства металлов как константы упругости, энергия формирования вакансии, реконструкция и релаксация поверхности. Для исследования в этих областях применяются многочастичные потенциалы, теряющие свойство аддитивности: метод погруженного атома (EAM/MEAM), tight-binding (TB) потенциалы (Гупты [10], TB-SMA [11], Саттона-Чена [12]). Они учиты-

вают непарный характер взаимодействий в металлах и в настоящее время широко применяются как при моделировании методом Монте-Карло, так и с методами молекулярной динамики. TB потенциалы хорошо зарекомендовали себя именно в моделировании наноразмерных структур, а именно: НЧ, нанопроволок, длинных атомных цепочек (LAC).

Общая энергия системы для TB модели вычисляется по формуле

$$E = \sum_{i \in \Psi} \left(\sum_{j \in \Psi_i} R_{ij} - \sqrt{\sum_{j \in \Psi_i} B_{ij}} \right). \quad (3)$$

Здесь R_{ij} и B_{ij} — вклад отталкивания и притяжения в общую энергию взаимодействия между атомами i и j . Выражения для R_{ij} и B_{ij} у разных потенциалов отличаются. В случае TB-SMA они выглядят следующим образом:

$$R_{ij} = A_{ij} e^{-p_{ij}(r_{ij}/r_0-1)}, \quad (4)$$

$$B_{ij} = \xi_{ij}^2 e^{-2q_{ij}(r_{ij}/r_0-1)}, \quad (5)$$

где A , p , q , ξ — параметры потенциала, определяемые типом атомов i и j . В решеточной модели удобно задавать потенциал в виде таблицы значений R_{ij} и B_{ij} для различных типов атомов и расстояний r_{ij} , а не пользоваться формулами (4) и (5) напрямую.

Формулу (3) можно переписать в следующем виде

$$E = \sum_{i \in \Psi} \sum_{j \in \Psi_i} (R(r_{ij})) - \sum_{i \in \Psi} \sqrt{\sum_{j \in \Psi_i} B(r_{ij})}. \quad (6)$$

Отсюда видно, что для составляющей притяжения при перемещении l -го атома необходимо пересчитать не только корень $\sqrt{\sum_{j \in \Psi_i} B(r_{ij})}$, но и корни $\sqrt{\sum_{j \in \Psi_i} B(r_{ij})}$

для всех i таких, что $l \in \Psi_i$. Если $|\Psi_i| = M$, то таких корней также будет M . Это означает, что если для парных потенциалов расчет изменения энергии имеет сложность $O(M)$, то для TB потенциалов — $O(M^2)$.

Насколько нам известно, ранее в литературе эта особенность многочастичных потенциалов явно не обсуждалась, хотя в целом считается, что их вычислительная стоимость выше, чем у парных [13].

Если при моделировании металлических НЧ не применяются специальные техники для повышения процента успешных шагов, а также если используется решеточная модель расположения атомов, то этот процент будет очень низким на протяжении большей части времени моделирования. Благодаря этому имеет смысл хранить результаты промежуточных вычислений энергии системы для ускорения последующих вычислений. Такой прием называется мемоизацией. Если для каждого i -го атома хранить актуальное значение суммы $\sum_{j \in \Psi_i} B(r_{ij})$, то при переме-

щении l -го атома можно вычислить её изменение как разницу $B(r_{il}) - B(r_{ij})$. Очевидно, что в отличие от полного пересчета суммы, эта операция имеет сложность не $O(M)$, а $O(1)$. Это делает общую сложность вычисления изменения энергии не $O(M^2)$, а $O(M)$, то есть такой же, как и у аддитивных потенциалов. Конечно, необходимо поддерживать актуальность суммы $\sum_{j \in \Psi_i} B(r_{ij})$ для всех атомов частицы, что требует

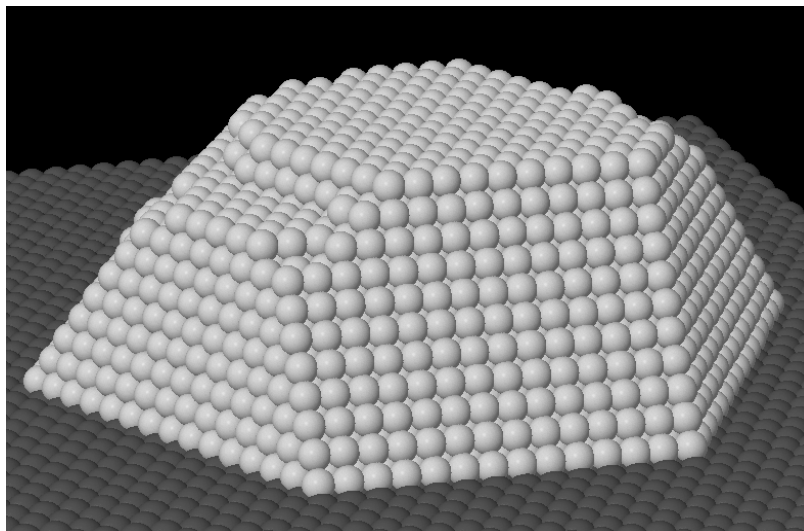


Рис. 1. Нанесенная наночастица платины из 4033 атомов, полученная с помощью предлагаемой модификации алгоритма Метрополиса (светлые сферы обозначают атомы платины, тёмные — атомы подложки)

дополнительных затрат. Поэтому не следует ожидать ускорения вычислений в M раз. В разработанной нами реализации алгоритма Метрополиса при моделировании в рамках решеточной модели платиновой НЧ применение такой мемоизации позволило ускорить процесс моделирования примерно в 15 раз (для $M=78$).

Реализация алгоритма

С применением описанных выше оптимизаций и техник реализация алгоритма Метрополиса для моделирования нанесенных НЧ выглядит следующим образом:

1. Заполнить модельную решётку атомами частицы и подложки; рассчитать R_i и V_i для $i \in \Psi$; составить списки активных атомов и вакансий.
 2. Выбрать случайные элементы из списков активных атомов и вакансий, посчитать суммарную энергию E_s атомов, находящихся в окрестностях этих двух точек.
 3. Энергия взаимодействия выбранного атома вычитается из аддитивных составляющих энергии в ячейках его соседей.
 4. Атом перемещается в выбранную вакансию, а значения аддитивных составляющих энергии его новых соседей увеличиваются в соответствии с формулами (4) и (5).
 5. Вычисляется суммарная энергия E_d атомов, находящихся в окрестностях начальной и конечной точек расположения атома.
 6. Вычисляется разница $\Delta E = E_d - E_s$, которая используется для принятия или отклонения данного перехода в соответствии с распределением Больцмана.
 7. Если пробный переход был принят, то ближайшие соседи исходной и конечной ячеек данного перехода, при необходимости, добавляются или удаляются из списков активных атомов и вакансий. Иначе — положение атома и значения аддитивных составляющих энергии вокруг атома и вакансии возвращаются в исходные состояния.
 8. Если не достигнуто состояние термодинамического равновесия, то алгоритм возвращается к пункту 2.
- С использованием данной реализации мы определили равновесную форму свободных и нанесенных

частиц платины размером из 4033 атомов при температуре 1162К. На процессоре Intel Xeon X5472 3 GHz реализованный алгоритм выполнял около 90 000 пробных шагов в секунду, из них около 500 успешных. Полученная нанесённая НЧ изображена на рис. 1.

Для достижения состояния термодинамического равновесия частиц потребовалось до 2000 шагов Монте-Карло, на каждом из которых предпринималось 72000 попыток (определяется размером решетки) выполнить акт диффузии. Чтобы убедиться, что частица пришла в состояние термодинамического равновесия, моделирование продолжалось до 5000 шагов. Весь процесс моделирования занимал не более двух часов, что очевидно можно считать приемлемыми вычислительными затратами при проведении исследований. Приведенный алгоритм успешно использовался нами в работе [14] для вычисления равновесной структуры адсорбирующей поверхности.

Заключение

Предлагаемая реализация алгоритма Метрополиса для моделирования свободных и нанесенных НЧ расширяет границы применимости алгоритма, позволяя за несколько часов процессорного времени решить задачу о поиске равновесной формы частиц из нескольких тысяч атомов. Требуемая производительность алгоритма была достигнута за счёт разработанной структуры данных для ведения списков активных атомов и вакансий с константной сложностью всех требуемых операций, а также благодаря предложенному приёму мемоизации при вычислении изменения энергии системы с помощью неаддитивных потенциалов.

Библиографический список

1. Haruta M., Yamada N., Kobayashi T. and Iijima S. Gold catalysts prepared by coprecipitation for low-temperature oxidation of hydrogen and of carbon monoxide // Journal of Catalysis. 1989, Vol. 115, pp. 301–309.
2. Metal nanoparticles: synthesis, characterization, and applications / ed. by Feldheim D.L., Foss C.A. - New York: Marcel Dekker, 2001. — 338 p.

3. D. J. Wales, J. P. K. Doye, A. Dullweber, M. P. Hodges, F. Y. Naumkin F. Calvo, J. Hernández-Rojas and T. F. Middleton. The Cambridge Cluster Database, URL: <http://www-wales.ch.cam.ac.uk/CCD.html>.
4. Metropolis N., Ulam S. The Monte Carlo method // Journal of the American Statistical Association. 1949, Vol. 44, pp. 335–341.
5. Frenkel D. Understanding molecular simulation: from algorithms to applications / Frenkel D., Smit B. - San Diego: Academic Press, 2002. — 638 p.
6. Landau D.P., and Binder K. A Guide to Monte Carlo Simulations in Statistical Physics. — 2nd ed. — Cambridge: Cambridge University Press, 2000. — 449 p.
7. Leach A.R. Molecular Modelling: Principles and Applications. — 2nd ed. — Prentice Hall, 2001. — 773 p.
8. McKenna K.P., Sushko P.V. and Shluge A.L. Transient Atomic Configurations of Supported Gold Nanocrystallites at Finite Temperature // Journal of Physical Chemistry C, 2007, Vol. 111, pp. 2823–2826.
9. Müller M., Albe K. Lattice Monte Carlo simulations of FePt nanoparticles: Influence of size, composition, and surface segregation on order-disorder phenomena // Physical Review B, 2005, Vol. 72, pp. 094203.
10. Gupta R. P. Lattice relaxation at a metal surface // Physical Review B, 1981, Vol. 23, pp. 6265–6270.
11. Cleri F., Rosatto V. Tight-binding potentials for transition metals and alloys // Physical Review B, 1993, Vol. 48, pp. 22–33.
12. Sutton A.P., Chen J. Long-range Finnis–Sinclair potentials // Philosophical Magazine Letters, 1990, Vol. 61, pp. 139–164.
13. Germann T. C., Kadam K., Lomdahl P. S. 25 Tflop/s Multibillion-atom Molecular Dynamics Simulations and Visualization/Analysis on BlueGene/L // Proceedings of the 2005 ACM/IEEE Conference on Supercomputing. 2005.
14. Myshlyavtsev A., Stishenko P. Monte Carlo model of CO adsorption on supported Pt nanoparticle // Applied Surface Science. 2010. Vol. 256, no. 17. Pp. 5376–5380.

МЫШЛЯВЦЕВ Александр Владимирович, доктор химических наук, профессор кафедры «Химическая технология переработки углеводородов», проректор по учебной работе.
СТИШЕНКО Павел Викторович, младший научный сотрудник кафедры высшей математики.
 Адрес для переписки: pvstishenko@omgtu.ru

Статья поступила в редакцию 30.08.2011 г.

© А. В. Мышлявцев, П. В. Стищенко

УДК 519.87

М. А. БАРЫШНИКОВ
А. А. КОЛОКОЛОВ

Омский государственный
 технический университет

Омский филиал Института
 математики им. С. Л. Соболева
 СО РАН

РЕШЕНИЕ НЕКОТОРЫХ ЗАДАЧ РАЗВОЗКИ НЕФТЕПРОДУКТОВ С ИСПОЛЬЗОВАНИЕМ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

В работе рассматриваются задачи оптимальной развозки нефтепродуктов. Для их решения построены модель дискретной оптимизации на графе и модель целочисленного линейного программирования, разработаны и реализованы алгоритмы муравьиной колонии, а также гибридный алгоритм с применением процедур локального поиска. Приведены результаты вычислительного эксперимента с реальными исходными данными, которые показали эффективность предложенного подхода к решению указанных задач.

Ключевые слова: задача оптимальной развозки, дискретная оптимизация, целочисленное программирование, дискретная оптимизация, алгоритм муравьиной колонии, локальный поиск.

Введение

Решение задач оптимальной развозки продукции [1, 2] необходимо для повышения эффективности работы транспортных структур крупных предприятий. Данная проблема особенно актуальна для организаций, в которых транспортные перевозки являются одним из основных видов деятельности [3]. Многие из указанных задач относятся к области развозки нефтепродуктов [4].

Рассматриваемые задачи, как правило, являются NP-трудными и на практике имеют большую размерность, вследствие чего нахождение точного решения представляется весьма трудоемким [5]. Использование эвристических алгоритмов позволяет получать приближенные решения с относительно небольшой погрешностью за приемлемое время [6].

В данной работе построены и реализованы эвристические алгоритмы, основанные на алгоритме муравьиной колонии [7, 8]. Приводятся результаты вы-

числительного эксперимента с использованием реальных данных, которые показали эффективность предложенного подхода к решению задач оптимальной развозки нефтепродуктов.

1. Постановка задачи и модель оптимизации на графе

Рассматривается следующая постановка задачи оптимальной развозки нефтепродуктов, учитывающая особенности отрасли. Имеется нефтебаза, обеспечивающая автозаправочные станции (АЗС) определенным видом топлива. Известна потребность каждой станции в топливе. Определен единый для всех потребителей директивный срок доставки.

Задана транспортная сеть, связывающая нефтебазу с АЗС и все станции между собой. Известны расстояния между узлами транспортной сети. Для развозки нефтепродуктов имеется несколько бензовозов с цистернами одинакового объема, разделенными на 3 отсека. Каждое транспортное средство выполняет один или несколько рейсов, удовлетворяя спрос некоторых АЗС, причем каждая из них обслуживается одним бензовозом и только один раз.

Количество топлива, необходимое для станций, входящих в один рейс, не должно превышать вместимости цистерны бензовоза. Требуется обеспечить всех потребителей продукцией в указанный директивный срок с наименьшими транспортными затратами.

Для построения математической модели введем граф $G=(W,E)$ с множеством вершин W и множеством ребер E , с помощью которого описывается транспортная сеть рассматриваемой задачи. Вершинам графа соответствуют потребители и база снабжения, а ребрам — соединяющие их участки пути. Совокупность потребителей — пункты с номерами $i=1, \dots, N$ (где $N=|W|-1$), а базе соответствует пункт с номером $i=0$.

Каждому ребру из множества E приписан вес, равный расстоянию между соответствующими вершинами, а для каждой вершины, соответствующей потребителю, известен показатель, равный его спросу. Рейсом транспортного средства будем называть цикл в указанном графе, начинающийся и заканчивающийся на базе и проходящий через некоторые вершины. Маршрутом назовем совокупность рейсов, совершаемых одним транспортным средством. Рейс является допустимым, если удовлетворен спрос всех входящих в него потребителей, а общее количество доставляемой продукции не превышает указанной вместимости. Маршрут называется допустимым, если он состоит из допустимых рейсов и их суммарное время не превосходит директивный срок. Каждый потребитель обслуживается одним транспортным средством и только один раз.

Необходимо найти совокупность допустимых маршрутов минимальной суммарной длины, обеспечивающих удовлетворение спроса всех потребителей.

2. Модель целочисленного программирования

Рассмотрим модель целочисленного линейного программирования (ЦЛП) для одной из постановок исследуемых задач. Для ее построения введем следующие обозначения.

Пусть N — количество АЗС, K — количество бензовозов, L — возможное количество рейсов, c_{ij} — затраты на перемещение из пункта i в пункт j , x_{ij}^{kl} — булевы переменные ($i, j=0, \dots, N; i \neq j; k=1, \dots, K; l=1, \dots, L$), V — объем бензовоза, V_i — спрос на

топливо АЗС с номером i , γ — коэффициент, обратно пропорциональный средней скорости бензовоза, δ — коэффициент, пропорциональный сумме скоростей налива топлива на нефтебазе и слива топлива на АЗС, T — директивный срок, r — количество отсеков бензовоза.

$$\min \sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K \sum_{l=1}^L c_{ij} \cdot x_{ij}^{kl}, \quad (1)$$

при условиях:

$$\sum_{j=1}^N \sum_{k=1}^K \sum_{l=1}^L x_{ij}^{kl} = 1, \quad i = 1, \mathbf{K}, N; \quad (2)$$

$$\sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^L x_{ij}^{kl} = 1, \quad j = 1, \mathbf{K}, N; \quad (3)$$

$$\sum_{j=0}^N x_{ij}^{kl} = \sum_{j=0}^N x_{ji}^{kl}, \quad i = 0, \mathbf{K}, N, \\ k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (4)$$

$$\sum_{j=0}^N x_{ij}^{kl} \leq 1, \quad i = 0, \mathbf{K}, N, \quad k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (5)$$

$$\sum_{i=0}^N x_{ij}^{kl} \leq 1, \quad j = 0, \mathbf{K}, N, \quad k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (6)$$

$$\sum_{i=0}^N \sum_{j=0}^N x_{ij}^{kl} \leq r + 1, \quad k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (7)$$

$$\sum_{i=1}^N \sum_{j=1}^N x_{ij}^{kl} \cdot V_i \leq V, \quad k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (8)$$

$$\sum_{i=0}^N \sum_{j=0}^N \sum_{l=1}^L x_{ij}^{kl} \cdot (\gamma \cdot c_{ij} + \delta \cdot V_i) \leq T, \quad k = 1, \mathbf{K}, K; \quad (9)$$

$$\sum_{j=1}^N x_{0j}^{kl} \cdot (r + 1) \geq \sum_{i=1}^N \sum_{j=1}^N x_{ij}^{kl}, \quad k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (10)$$

$$x_{ij}^{kl} + x_{ji}^{kl} \leq 1, \quad i, j = 0, \mathbf{K}, N, \\ k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L; \quad (11)$$

$$x_{ij}^{kl} = \begin{cases} 1, & \text{если ребро } (i, j) \text{ входит} \\ & \text{в рейс } l \text{ бензовоза } k, \\ i, j = 0, \mathbf{K}, N, \quad k = 1, \mathbf{K}, K, \quad l = 1, \mathbf{K}, L. \\ 0, & \text{иначе,} \end{cases} \quad (12)$$

Выражение (1) — целевая функция, минимизирующая суммарные затраты. Ограничения (2) и (3) означают, что из каждой вершины, кроме базы, выходит ровно одно ребро, и в каждую такую вершину входит также одно ребро; (4) показывает, что в рамках одного рейса, число ребер, входящих в каждую вершину, равно числу ребер, выходящих из нее; выражения (5) и (6) означают, что для фиксированного рейса число ребер, входящих в каждую вершину не более одного, число выходящих также не более одного; (7) — для фиксированного рейса число вершин, включая базу, не превышает $(r + 1)$; (8) — для одного рейса сумма спроса АЗС не превышает объема цистерны бензовоза; ограничение (9) показывает, что для каждого транспортного средства время выполнения всех рейсов с учетом сервисных остановок не превосходит директивный срок; (10) —

в каждый рейс обязательно включается база; (11) — ограничение, обеспечивающее замкнутость каждого рейса с отсутствием внутренних петель. Отметим, что в выражениях (2) — (12) для всех пар (i, j) предполагается, что $i \neq j$.

Разработанная модель целочисленного линейного программирования с помощью пакетов ЦЛП позволяет получать оптимальные решения задач развозки, а при анализе эвристических алгоритмов — оценить их точность. При ее создании использовалась модель ЦЛП задачи коммивояжера, которая была расширена дополнительными условиями, присущими специфике отрасли.

3. Алгоритм муравьиной колонии

Идея данного алгоритма основана на поведении муравьев в процессе поиска пути от муравейника до источника пищи. Каждый муравей при движении оставляет за собой специальное вещество — феромон. Другие члены колонии используют этот след при поиске источника пищи. Вероятность выбора направления движения повышается с увеличением концентрации феромона.

Рассмотрим разработанный алгоритм муравьиной колонии [7, 8] для исследуемой задачи. Перед началом первой итерации концентрация феромона на всех ребрах одинакова. На каждой итерации t алгоритма m искусственных муравьев строят решения задачи развозки, двигаясь по графу задачи от одного потребителя к другому согласно некоторому вероятностному правилу, периодически посещая базу. Если, находясь в вершине i , муравей выбирает для перехода вершину j , то дуга (i, j) добавляется в решение. Таким образом, найденное муравьем решение представляет собой список пройденных дуг. Для того, чтобы в построенном решении все пункты были различны, каждому муравью сопоставляется список запретов (*tabu list*). В этом списке запоминаются потребители, пройденные к текущему моменту.

При выходе из базы каждому муравью сопоставляется количество продукции V , которое уменьшается на величину V_i при посещении потребителя i . Когда оставшееся количество не позволяет удовлетворить спрос ни одного из потребителей, муравей возвращается на базу.

Вероятность перехода из пункта i в пункт j для муравья l на итерации t алгоритма вычисляется следующим образом:

$$p_{ij}^l = \begin{cases} \frac{(\tau_{ij}(t))^\alpha (\eta_{ij})^\beta}{\sum_{s \in R^l} (\tau_{is}(t))^\alpha (\eta_{is})^\beta}, & \text{если } j \in R^l, \\ 0, & \text{в противном случае,} \end{cases} \quad (13)$$

где τ_{ij} — уровень феромона, $\eta_{ij} = 1/c_{ij}$ является «видимостью» пункта j из пункта i , α и β — управляющие параметры, R^l — множество пунктов, которые муравей l еще не посетил.

После того, как муравей оказался на базе, и все пункты были посещены, начинается движение следующей муравей.

После завершения движения всех муравьев рассчитывается вклад каждого муравья l в изменение уровня феромона для каждого ребра (i, j) по формуле:

$$\Delta\tau_{ij}^l = \begin{cases} Q_0 / P^l(t), & \text{если } (i, j) \in T^l(t), \\ 0, & \text{в противном случае,} \end{cases} \quad (14)$$

где $T^l(t)$ — общий маршрут, построенный муравьем l на итерации t , $P^l(t)$ — суммарные затраты на данный маршрут, Q_0 — некоторая положительная константа.

Далее вычисляется суммарное изменение феромона для каждого ребра (i, j) :

$$\Delta\tau_{ij}(t) = \sum_{l=1}^m \Delta\tau_{ij}^l(t). \quad (15)$$

Концентрация феромона для каждого ребра (i, j) на следующей итерации рассчитывается по формуле:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \quad (16)$$

где $\rho \in (0, 1]$ — коэффициент испарения феромона.

Алгоритм завершает работу после выполнения заданного максимального числа итераций или по истечении определенного времени. Наилучшими считаются решения, соответствующие текущему рекорду [3].

4. Система «элитных» муравьев и гибридный алгоритм

Одним из усовершенствований классической версии алгоритма муравьиной колонии является введение в алгоритм так называемых «элитных» муравьев [7]. Элитой называется совокупность муравьев, чьи маршруты по значению целевой функции лучше остальных. Опыт показывает, что, двигаясь по ребрам, входящим в короткие пути, муравьи с большей вероятностью будут находить пути меньшей длины. Таким образом, эффективной стратегией является искусственное увеличение уровня феромона на самых удачных маршрутах.

В связи с этим нами была разработана следующая модификация вышеописанного алгоритма муравьиной колонии, основанная на знаниях об «элитных» муравьях [4]. Обозначим через $L(t)$ длину лучшего маршрута на момент времени t , а через e — количество элитных муравьев. Тогда уровень феромона на ребрах лучших маршрутов увеличится на величину:

$$\Delta\tau_e(t) = A^e \cdot \frac{Q}{L(t)}, \quad (17)$$

где A^e — «авторитет» множества элитных муравьев, Q — некоторая положительная константа. Таким образом, мы можем регулировать влияние «элитных» муравьев с помощью коэффициента A^e . Оптимальное значение A^e зависит от размерности графа и численности колонии муравьев.

Исследования в области дискретной оптимизации показали эффективность использования комбинированных методов. Для повышения точности алгоритмов муравьиной колонии предложен гибридный вариант алгоритма с применением процедур локального поиска.

Для рассматриваемой задачи был построен следующий алгоритм локального поиска. Лучшее решение, полученное на каждой итерации алгоритма муравьиной колонии, берется в качестве начального для процедуры локального поиска. К этому решению применялись следующие преобразования:

- 1) перестановка двух произвольных вершин внутри одного рейса;
- 2) перестановка двух произвольных вершин внутри одного маршрута.

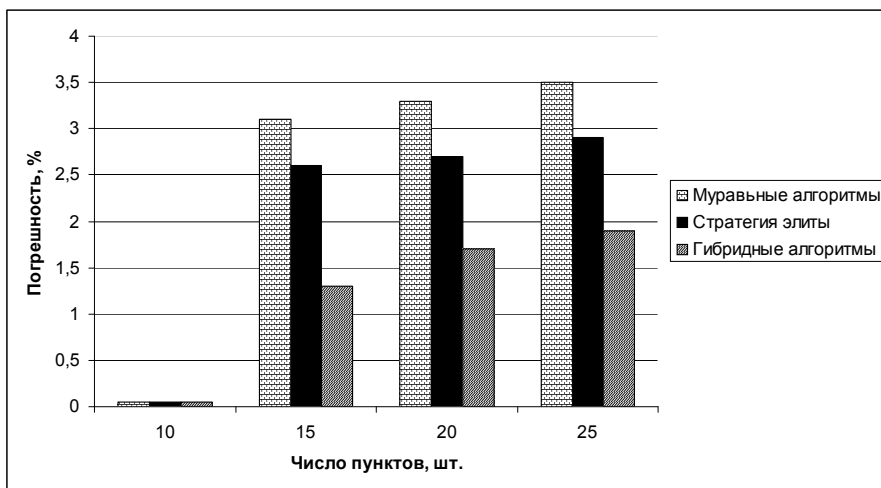


Рис. 1. Сравнение погрешности алгоритмов

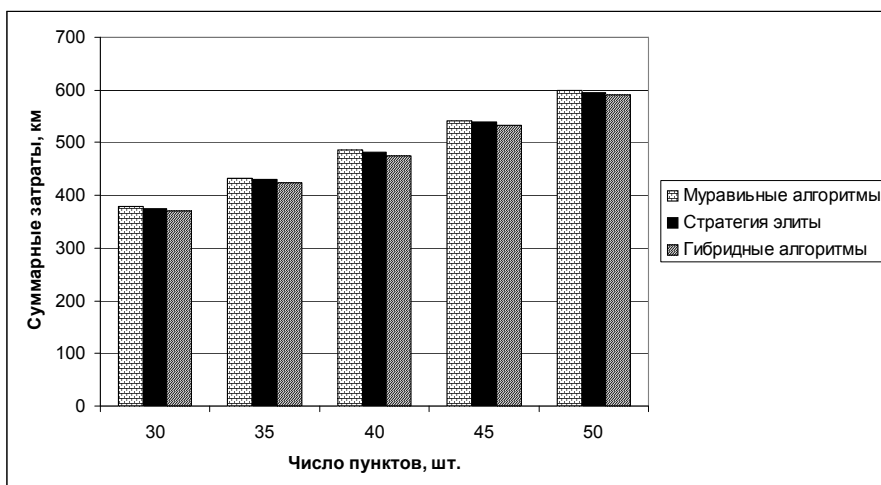


Рис. 2. Сравнение алгоритмов при размерности свыше 25

Указанные преобразования выполняются до тех пор, пока не будет получено лучшее решение. Далее лучшее решение передается алгоритму муравьиной колонии.

5. Результаты экспериментальных исследований

Указанные алгоритмы реализованы, проведены их экспериментальные исследования на задачах с реальными исходными данными одного из омских предприятий, осуществляющего развозку нефтепродуктов. Было решено более 100 задач размерности до 50 станций и получены приближенные решения. Для построения точных решений исследуемой задачи на основе разработанной модели ЦАП проведены расчеты с использованием пакета GAMS. Выполнен анализ алгоритмов по точности и времени работы.

Для получения оценок погрешности работы эвристических алгоритмов были протестированы задачи относительно небольшой размерности (не более 25 станций), чтобы они решались за приемлемое время в пакете GAMS. Результаты показали, что при размерности 10 в серии запусков алгоритмы почти всегда находят точное решение, при размерности от 10 до 25 лучшие результаты показали гибридные алгоритмы с применением процедур локального

поиска (рис. 1). На задачах большей размерности (свыше 25 станций) получение точного решения весьма трудоемко, однако возможно сравнить результаты работы алгоритмов между собой. В этом случае также гибридные алгоритмы показали лучшие результаты (рис. 2). При оценке работы алгоритмов было замечено небольшое увеличение времени решения у гибридных алгоритмов, но оно незначительно, для задач размерности 50 станций время работы менее 6 секунд можно считать приемлемым (рис. 3). Экспериментальным путем найдены оптимальные значения количества искусственных муравьев — 200 и количества итераций — 20.

На основе результатов вычислительного эксперимента сделан вывод о том, что алгоритмы муравьиной колонии позволяют получать достаточно хорошие приближенные решения за приемлемое время. Применение стратегии «элитных» муравьев и процедур локального поиска позволяет увеличить точность решения с незначительным повышением времени работы. Лучшие решения были получены с помощью гибридного алгоритма, их погрешность составила 1–2 %.

Заключение

Для исследуемых задач оптимальной развозки нефтепродуктов построены математические модели,

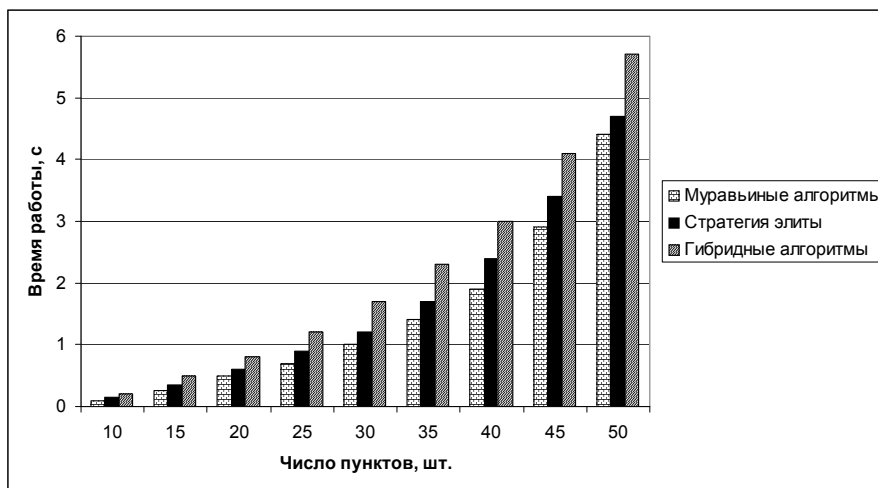


Рис. 3. Сравнение времени работы алгоритмов

разработаны и реализованы алгоритмы муравьиной колонии, а также гибридный алгоритм с применением процедур локального поиска. Проведенные экспериментальные исследования с реальными исходными данными показали эффективность предложенного подхода для решения рассматриваемых задач развозки.

Библиографический список

1. Aksen D. Open vehicle routing problem with driver nodes and time deadlines / D. Aksen, Z. Ozyurt, N. Aras // Journal of the Operational Research Society. – 2006. – Vol. 58, № 9. – P. 1223–1234.
2. Pisinger D. A general heuristic for vehicle routing problems / D. Pisinger, S. Ropke // Computers & Operations Research. – 2007. – Vol. 34, № 8. – P. 2403–2435.
3. Барышников, М. А. Приближенное решение некоторых задач оптимальной развозки продукции / М. А. Барышников, Д. А. Уляшев // Теоретические знания – в практические дела : сб. науч. статей Междунар. конф. (Омск, 7–15 апр. 2011 г.). В 2 ч. Ч. 2. – Омск : Изд-во ГОУ ВПО «РосЗИТАП», 2011. – С. 111–114;
4. Барышников, М. А. Разработка и реализация алгоритмов муравьиной колонии для некоторых задач развозки нефтепродуктов / М. А. Барышников // Статистика. Моделирование. Оптимизация : сб. тр. Всерос. конф. (Челябинск, 28 нояб.– 2 дек. 2011 г.). – Челябинск : Изд-во ЮУрГУ, 2011. – С. 274–279.
5. Костюк, Ю. Л. Сбалансированная эвристика для решения задачи маршрутизации транспорта с учетом грузоподъемности / Ю. Л. Костюк, М. С. Пожидаев // Вестник ТГУ. – 2010. – № 3. – С. 65–72.
6. Меламед, И. И. Задача коммивояжера. Приближенные алгоритмы / И. И. Меламед, С. И. Сергеев, И. Х. Сигал // Автоматика и телемеханика. – 1989. – № 11. – С. 3–26.
7. Колоколов, А. А. Алгоритмы муравьиной колонии для задач дискретной оптимизации : учеб. пособие / А. А. Колоколов, Т. В. Леванова, М. А. Лореш – Омск : Изд-во ОмГУ, 2008. – 32 с.
8. Dorigo, M. Ant Colony Optimization. Artificial Ants as a Computational Intelligence Technique / M. Dorigo, M. Birattari, T. Stutzle // IRIDA – Technical report series: TR/IRIDA/2006-023.

БАРЫШНИКОВ Михаил Алексеевич, аспирант кафедры «Математические методы и информационные технологии в экономике» Омского государственного технического университета.

КОЛОКОЛОВ Александр Александрович, доктор физико-математических наук, профессор, Омский филиал института математики им. С. Л. Соболева СО РАН, заведующий лабораторией дискретной оптимизации.

Адрес для переписки: kolo@ofim.oscsbras.ru

Статья поступила в редакцию 26.09.2011 г.

© М. А. Барышников, А. А. Колоколов

Книжная полка

Юша, В. Л. Методы и средства исследований : конспект лекций / В. Л. Юша, Н. А. Райковский ; ОмГТУ. – Омск : Изд-во ОмГТУ, 2011. – 95 с. – ISBN 978-5-8149-1108-7.

В конспекте лекций приведены математические методы планирования и обработки результатов экспериментальных исследований, рассмотрены основные принципы организации эксперимента, принципы работы с полученной выборкой значений результатов экспериментальных исследований, методы статистического анализа результатов исследования, а также необходимый минимум по теории вероятностей и математической статистике.