

УДК 536.75, 538.9

## ПРОГРАММА MEL АТОМИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ ФУНКЦИОНАЛЬНЫХ СЛОЕВ СОЛНЕЧНЫХ ЭЛЕМЕНТОВ

Ф. В. Григорьев<sup>1</sup>, А. Н. Романов<sup>1</sup>, И. В. Офёркин<sup>2</sup>, А. В. Сулимов<sup>2</sup>, В. Б. Сулимов<sup>1,2</sup>

Предложен и реализован новый метод атомистического моделирования периодических структур, формируемых органическими и гибридными молекулами. Такие структуры являются основой функциональных слоев органических и гибридных солнечных элементов. В рамках этого метода энергия межатомного взаимодействия рассчитывается в приближении классического силового поля, поиск наиболее стабильных периодических структур ведется с использованием генетического алгоритма глобальной оптимизации. Описана структура программы MEL, реализующей метод, в том числе ее параллельная версия, предложена оценка эффективности параллелизации. Выполнен тестовый расчет периодической структуры на основе молекулы фуллерена, проводится сравнение с экспериментальными данными.

**Ключевые слова:** моделирование структуры солнечных элементов, глобальная оптимизация структуры, моделирование периодических структур.

**1. Введение.** В традиционных солнечных элементах (СЭ) генерация носителей заряда происходит в результате поглощения света в функциональных слоях из неорганических материалов. Такие СЭ обладают высокой эффективностью, но относительно дороги вследствие высокой стоимости материала для их изготовления. В последние 10–20 лет интенсивно разрабатываются СЭ на основе органических и гибридных (сочетание органических и неорганических) материалов, что существенно снижает стоимость изготовления СЭ. Однако к настоящему времени их эффективность в разы меньше, чем у традиционных (неорганических) СЭ.

Одной из ключевых проблем при разработке гибридных и органических СЭ является выбор молекул, на основе которых будут создаваться функциональные слои, где и происходит генерация и транспорт носителей заряда. Эти слои состоят из граничащих друг с другом кластеров с упорядоченной структурой; каждый кластер образован либо молекулами донора, в которых при поглощении света возбуждается электрон, либо молекулами акцептора, способными принять из донора возбужденный светом электрон [1]. Параметры функциональных слоев, важные с точки зрения их практического использования в СЭ (полоса поглощения, транспортные характеристики, время рекомбинации электрон-дырочных пар и др.), определяются свойствами молекул, размерами кластеров, их структурой, а также структурой границы раздела между ними. Знание структуры функционального слоя необходимо для расчета его параметров и поиска молекул, способных обеспечить высокую эффективность СЭ.

Экспериментальное исследование структуры функциональных слоев из гибридных и неорганических материалов затруднено из-за относительной сложности структуры образующих их молекул, поэтому использование методов молекулярного моделирования для этой цели является оправданным. В настоящей статье представлен один из возможных подходов к такому моделированию на атомистическом уровне, описан его алгоритм, реализованный в программе MEL (Modeling of the solar cELs), и приведены первые результаты моделирования.

**2. Алгоритм моделирования.** Для изучения структуры функционального слоя необходимо научиться определять на атомистическом уровне структуру донорных и акцепторных кластеров, из которых состоит слой. Поскольку предполагается [2, 3], что сами кластеры имеют структуру, близкую к кристаллической, цель алгоритма, реализованного в программе MEL, заключается в поиске наиболее стабильных молекулярных кристаллов, построенных на основе молекул с известной трехмерной структурой. Предполагается, что молекулярный кристалл формируется на основе невалентных взаимодействий, т.е.

<sup>1</sup> Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, 119992, Ленинские горы, д. 1, стр. 4, Москва; Ф. В. Григорьев, ст. науч. сотр., e-mail: fedor.grigoriev@gmail.com; А. Н. Романов, вед. науч. сотр., e-mail: alexey.romanov@list.ru; В. Б. Сулимов, зав. лаб., e-mail: vladimir.sulimov@gmail.com

<sup>2</sup> ООО "Димонта", ул. Нагорная, д. 15, корп. 8, 117186, Москва; И. В. Офёркин, программист, e-mail: io@dimonta.com; А. В. Сулимов, системный программист, e-mail: sulimovv@mail.ru

химические связи между атомами, относящимися к разным ячейкам, отсутствуют. В элементарной ячейке могут быть как одна, так и несколько молекул — одинаковых или отличающихся по составу и структуре.

Наиболее стабильные кристаллы обладают наименьшей энергией  $U_{\text{cell}}$  в расчете на элементарную ячейку. Величина  $U_{\text{cell}}$  зависит от  $(3N-6)$  координат атомов молекул, входящих в состав ячейки ( $N > 2$  — число атомов в ней), и векторов трансляции (в общем случае девять переменных), а для определения переменных, отвечающих стабильным кристаллам, необходимо найти глобальный минимум  $U_{\text{cell}}$ . Точное решение задачи о глобальной оптимизации возможно только при небольшом числе переменных (например, для кластеров, образованных несколькими инертными атомами). В рассматриваемом случае молекулярные кристаллы формируются молекулами, которые содержат десятки атомов, так что их число в элементарной ячейке может быть  $\sim 10^2$ .

Чтобы понизить вычислительную сложность задачи, мы используем один из эвристических алгоритмов глобальной оптимизации — генетический алгоритм (ГА) [4–6] и уменьшаем число переменных, от которых зависит  $U_{\text{cell}}$ . В рамках нашей модели предполагается, что при формировании кристалла за счет невалентных взаимодействий длины связей и валентные углы не меняются, их величины близки к тем, которые соответствуют молекулам в газовой фазе, в то время как углы вращения вокруг связей, определяющие конформацию молекулы, могут меняться. Таким образом, в случае одной молекулы в составе ячейки  $(3N-6)$  декартовых координат атомов заменяются на  $N_{\text{rot}}$  углов вращения вокруг связей, что позволяет существенно редуцировать вычислительную сложность задачи. В общем случае, если в элементарной ячейке  $N_{\text{mol}}$  молекул, число переменных при глобальной оптимизации определяется выражением

$$N_{\text{tot}} = \sum_{i=1}^{N_{\text{mol}}} N_{\text{rot}}^i + 6(N_{\text{mol}} - 1) + 9. \quad (1)$$

Первое слагаемое в (1) — число углов вращения вокруг связей, второе — число переменных, необходимых для задания положения молекул в элементарной ячейке относительно друг друга, последнее — девять координат трех векторов трансляции.

Алгоритм поиска кристаллической структуры состоит из следующих основных этапов (рис. 1).

1) Выбор молекул, входящих в элементарную ячейку, задание их числа и геометрии. Поскольку углы вращения вокруг связей будут меняться в ходе моделирования, необходимая информация — это длины связей, валентные углы и матрица связности, содержащая для каждого атома информацию о типе химической связи (одинарная, двойная, тройная) и номерах атомов, с которыми данный атом образует связь. Однако на практике проще задать декартовы координаты атомов и матрицу связности, по которым можно рассчитать длины связей и валентные углы.

2) Для каждой молекулы, входящей в состав элементарной ячейки, определение списка углов вращения вокруг связи. Величины углов будут меняться в ходе моделирования.

3) Выполнение основного цикла моделирования: с использованием генетического алгоритма поиск значений  $N_{\text{tot}}$  координат, отвечающих минимуму энергии ячейки  $U_{\text{cell}}$ .

4) По окончании работы генетического алгоритма вывод значений координат, соответствующих минимальной  $U_{\text{cell}}$ .

В следующем разделе мы получим выражение для расчета энергии ячейки  $U_{\text{cell}}$  в зависимости от координат.

**3. Расчет энергии ячейки.** В рамках приближения классического силового поля энергия кристалла  $U_{\text{cr}}$  определяется выражением

$$U_{\text{cr}} = \sum_{i,j} U_{ij} + \sum_{i,j,k} U_{ijk} + \sum_{i,j,k,l} U_{ijkl}, \quad (2)$$

где  $U_{ij}$ ,  $U_{ijk}$ ,  $U_{ijkl}$  — энергия парных, тройных и четверных взаимодействий; суммирование ведется по

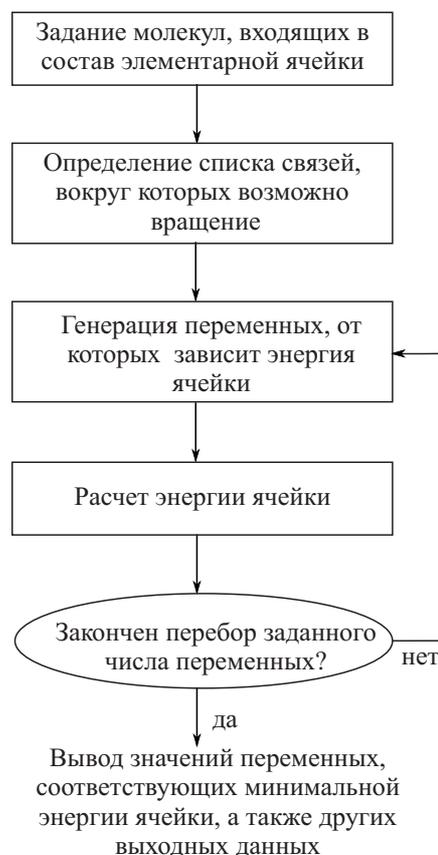


Рис. 1. Алгоритм поиска кристаллической структуры, формирующейся за счет невалентных взаимодействий молекул, используемых в функциональных слоях гибридных и органических солнечных батарей

всем парам, тройкам и четверкам атомов всего кристалла соответственно, второй индекс больше первого. Для неполяризуемых силовых полей к парным взаимодействиям относят невалентные, которые представляются в виде суммы электростатического взаимодействия и взаимодействия Ван-дер-Ваальса, а также вклад в энергию от изменения длин связей; слагаемое  $U_{ijk}$  описывает вклад в  $U_{cr}$  от изменения валентных углов, образуемых тройками атомов,  $U_{ijkl}$  — от изменения углов вращения вокруг связей и углов выхода атома из плоскости (последнее — для плоских, в основном ароматических структур).

Если кристалл образован за счет невалентных взаимодействий, то последние два слагаемых в (2) можно переписать в форме

$$\sum_{i,j,k} U_{ijk} + \sum_{i,j,k,l} U_{ijkl} = \sum_{\text{cell}} \sum_{i,j,k}^{N_3} U_{ijk} + \sum_{\text{cell}} \sum_{i,j,k,l}^{N_4} U_{ijkl} = N_{\text{cell}} \left( \sum_{i,j,k}^{N_3} U_{ijk} + \sum_{i,j,k,l}^{N_4} U_{ijkl} \right), \quad (3)$$

где  $N_{\text{cell}}$  — число ячеек,  $N_3$  — число троек атомов с ненулевым потенциалом тройного взаимодействия в пределах одной ячейки,  $N_4$  — число четверок атомов с ненулевым потенциалом четверного взаимодействия в пределах одной ячейки. Первое слагаемое в (2) может быть представлено следующим образом:

$$\sum_{i,j} U_{ij} = \frac{1}{2} \sum_i \left( \sum_{j=1}^{N_1} U_{ij} + \sum_{j=N_1+1}^{2N_1} U_{ij} + \dots \right) = \frac{1}{2} N_{\text{cell}} \sum_i \sum_{j=1}^{N_1} U_{ij}. \quad (4)$$

Здесь при суммировании по второму индексу мы явно выделили атомы, принадлежащие к одной ячейке,  $N_1$  — число атомов в ячейке. Первое суммирование в правой части (4) выполняется по всем атомам кристалла, второе — по атомам ячейки.

Учитывая (3) и (4), энергию кристалла запишем в виде

$$U_{cr} = N_{\text{cell}} U_{\text{cell}},$$

где  $U_{\text{cell}}$  — энергия одной ячейки:

$$U_{\text{cell}} = \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij} + \sum_{i,j,k,l}^{N_4} U_{ijkl} + \sum_{i,j,k}^{N_3} U_{ijk}. \quad (5)$$

Парные взаимодействия (первое слагаемое в (5)) представим в форме

$$\sum_i \sum_{j=1}^{N_1} U_{ij} = \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij}^{\text{el}} + \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij}^{\text{vdw}} + \sum_{i,j}^{N_2} U_{ij}^b, \quad (6)$$

где  $U_{ij}^{\text{el}}$  и  $U_{ij}^{\text{vdw}}$  — энергия электростатического взаимодействия и взаимодействия Ван-дер-Ваальса пары атомов;  $U_{ij}^b$  — энергия пары атомов, между которыми имеется ковалентная связь;  $N_2$  — число таких пар. В (6) явным образом учтено, что между атомами, относящимися к разным ячейкам, нет ковалентных связей. Подставив (6) в (5), получим

$$U_{\text{cell}} = \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij}^{\text{el}} + \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij}^{\text{vdw}} + \sum_{i,j}^{N_2} U_{ij}^b + \sum_{i,j,k,l}^{N_4} U_{ijkl} + \sum_{i,j,k}^{N_3} U_{ijk}. \quad (7)$$

Выражение (7) зависит как от координат атомов ячейки, так и от векторов трансляции, так как в первых двух слагаемых в (7) первое суммирование ведется по всем атомам кристалла. В последних трех слагаемых в (7) при суммировании значение второго индекса больше первого.

Задача, решаемая в программе MEL, заключается в поиске глобального минимума выражения (7) при соблюдении некоторых ограничений, а именно: длины связей (слагаемое  $U_{ij}^b$ ) и валентные углы (слагаемое  $\sum_{i,j,k}^{N_3} U_{ijk}$ ) не меняются. Тогда (7) можно записать так:

$$U_{\text{cell}} = \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij}^{\text{el}} + \frac{1}{2} \sum_i \sum_{j=1}^{N_1} U_{ij}^{\text{vdw}} + \sum_{i,j,k,l}^{N_4} U_{ijkl} + \text{const}. \quad (8)$$

Взаимодействие атомов, относящихся к различным ячейкам, можно целиком учесть в рамках парного потенциала  $U_{ij}$ .

Если  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  — векторы трансляции кристаллической решетки, то первые два слагаемых в (8) можно переписать в виде

$$U_{\text{cell}} = \frac{1}{2} \sum_{i=-na}^{na} \sum_{j=-nb}^{nb} \sum_{k=-nc}^{nc} \sum_{l,m=1}^{N_1} U^{\text{el}} |\mathbf{r}_l - (\mathbf{r}_m + i\mathbf{a} + j\mathbf{b} + k\mathbf{c})| + \\ + U^{\text{vdw}} |\mathbf{r}_l - (\mathbf{r}_m + i\mathbf{a} + j\mathbf{b} + k\mathbf{c})| + \sum_{i,j,k,l}^{N_4} U_{ijkl}, \quad (9)$$

где  $na$ ,  $nb$ ,  $nc$  — число трансляций по прямому и обратному направлениям векторов  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  соответственно; слагаемое  $\text{const}$  (см. (8)), не меняющееся в ходе моделирования, опущено. При  $i = j = k = 0$  в первой сумме учитываются только те слагаемые, для которых  $l \neq m$ .

Для удобства анализа перепишем (9) следующим образом:

$$U_{\text{cell}} = U_{\text{in}} + \frac{1}{2} \sum_{n=1}^{N_{\text{tr}}} U_n^{\text{tr}}. \quad (10)$$

Здесь  $N_{\text{tr}}$  — число трансляций;  $U_{\text{in}}$  — внутренняя энергия ячейки без учета ее взаимодействия с другими ячейками, полученными в результате трансляций:

$$U_{\text{in}} = \frac{1}{2} \sum_{\substack{l,m=1; \\ l \neq m}}^{N_1} U^{\text{el}} |\mathbf{r}_l - \mathbf{r}_m| + U^{\text{vdw}} |\mathbf{r}_l - \mathbf{r}_m| + \sum_{i,j,k,l}^{N_4} U_{ijkl}.$$

Заметим, что  $U_{\text{in}}$  не зависит от векторов трансляции.

Физический смысл каждого из слагаемых суммы в (10) — энергия взаимодействия ячейки с другими, полученными в результате трансляций, причем  $n$ -е слагаемое определяется следующим образом:

$$U_n^{\text{tr}} = \sum_{j=-n}^n \sum_{k=-n}^n \sum_{l,m=1}^{N_1} U^{\text{el}} |\mathbf{r}_l - (\mathbf{r}_m + n\mathbf{a} + j\mathbf{b} + k\mathbf{c})| + U^{\text{vdw}} |\mathbf{r}_l - (\mathbf{r}_m + n\mathbf{a} + j\mathbf{b} + k\mathbf{c})| + \\ + \sum_{i=-n}^n \sum_{k=-n}^n \sum_{l,m=1}^{N_1} U^{\text{el}} |\mathbf{r}_l - (\mathbf{r}_m + i\mathbf{a} + n\mathbf{b} + k\mathbf{c})| + U^{\text{vdw}} |\mathbf{r}_l - (\mathbf{r}_m + i\mathbf{a} + n\mathbf{b} + k\mathbf{c})| + \\ + \sum_{i=-n}^n \sum_{j=-n}^n \sum_{l,m=1}^{N_1} U^{\text{el}} |\mathbf{r}_l - (\mathbf{r}_m + i\mathbf{a} + j\mathbf{b} + n\mathbf{c})| + U^{\text{vdw}} |\mathbf{r}_l - (\mathbf{r}_m + i\mathbf{a} + j\mathbf{b} + n\mathbf{c})|. \quad (11)$$

Вычисление  $U_n^{\text{tr}}$  для  $n = 2$  в соответствии с (11) для плоского случая иллюстрируется на рис 2.

На рис. 3 иллюстрируется вычисление энергии  $U_{\text{cell}}$  для случая одной молекулы (с одним углом вращения вокруг связи) в элементарной ячейке.

Расчет энергии ячейки по (10) проводится в рамках силового поля MMFF94 [7], которое обладает универсальной параметризацией для широкого класса соединений, что позволит решать задачу об оптимизации состава функциональных слоев СЭ на основе единого подхода. Кроме того, способ расстановки парциальных зарядов на атомах, принятых в рамках MMFF94, позволяет автоматически обеспечить электронейтральность элементарной ячейки.

Потенциальная энергия молекулы в рамках силового поля MMFF94 представляется в виде суммы семи слагаемых, из которых в рамках нашего подхода учитываются три:

— слагаемое, описывающее электростатическое взаимодействие точечных зарядов  $q_i$ ,  $q_j$ , центрированных на атомах  $U^{\text{el}} = 332.0716 \frac{q_i q_j}{r_{ij}}$ ;

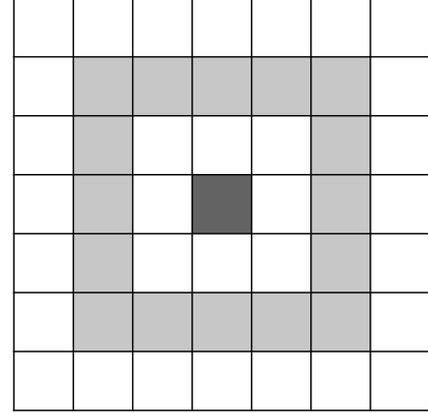


Рис. 2. Светло-серым показаны ячейки, взаимодействие которых с центральной ячейкой (показана темно-серым) учитывается при расчете  $U_n^{\text{tr}}$  для случая  $n = 2$

— взаимодействие Ван-дер-Ваальса

$$U^{vdw} = \epsilon_{ij} \left( \frac{1.07R_{ij}^*}{r_{ij} + 0.07R_{ij}^*} \right)^7 \left( \frac{1.12(R_{ij}^*)^7}{r_{ij}^7 + 0.12(R_{ij}^*)^7} - 2 \right),$$

где  $\epsilon_{ij}$ ,  $R_{ij}^*$  — параметры взаимодействия; силовыми параметрами для расчета  $U^{vdw}$  являются величины  $\alpha_i$ ,  $N_i$ ,  $A_i$ ,  $G_i$ ; с помощью этих величин вычисляются радиусы Ван-дер-Ваальса  $R_{ij}^*$  и глубина ям  $\epsilon_{ij}$ :

$$R_{ii}^* = A_i \alpha_i^{1/4}; \quad R_{ij}^* = 0.5(R_{ii}^* + R_{jj}^*) \left( 1.0 + B \left( 1 - \exp(-12\gamma_{ij}^2) \right) \right);$$

$$\gamma_{ij} = \frac{R_{ii}^* - R_{jj}^*}{R_{ii}^* + R_{jj}^*}; \quad \epsilon_{ij} = \frac{181.16 G_i G_j \alpha_i \alpha_j}{(\alpha_i/N_i)^{1/2} + (\alpha_j/N_j)^{1/2}} \frac{1}{R_{ij}^{*6}};$$

— энергия, обусловленная деформацией угла вращения  $\varphi$ :

$$U_{ijkl} = 0.5(V_1(1 + \cos \varphi) + V_2(1 + \cos 2\varphi) + V_3(1 + \cos 2\varphi)).$$

Расстановка параметров осуществляется программой MEL по правилам, описанным в [7].

$$U_{cell} = U_{cell}(\varphi, a_x, a_y, a_z, b_x, b_y, b_z, c_x, c_y, c_z)$$

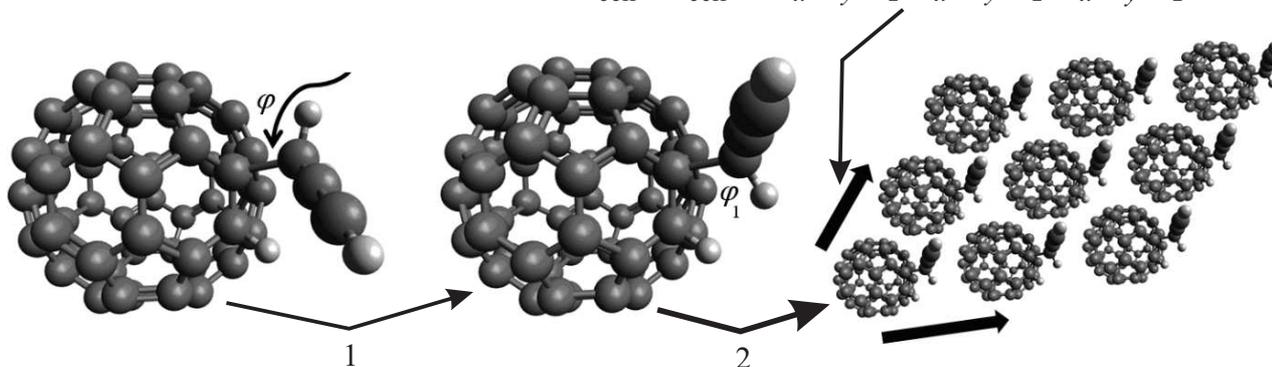


Рис. 3. Схема расчета энергии ячейки  $U_{cell}$  как функции угла вращения и трех векторов трансляций  $a, b, c$ : 1) по заданному значению угла вращения осуществляется вращение меньшей части молекулы относительно большей; 2) по заданным векторам трансляции генерируется кристаллическое окружение (трансляции) центральной ячейки; 3) для полученной геометрии рассчитывается  $U_{cell}$  по (10)

#### 4. Генетический алгоритм в программе MEL и особенности его параллельной реализации.

Для отыскания переменных, соответствующих глобальному минимуму энергии кристаллической решетки  $U_{cell}$ , используется генетический алгоритм (ГА). В качестве допустимых вариантов кристаллической решетки рассматриваются все возможные варианты триклинной сингонии, т.е. решетки, задаваемые тремя независимыми некопланарными векторами трансляций, что позволяет учитывать и другие возможные варианты решеток. В качестве допустимых конформаций молекул в ячейке рассматриваются все ее преобразования, сохраняющие неизменными длины валентных связей и валентные углы, т.е. преобразования, меняющие только углы вращения вокруг связей.

Основная идея ГА заключается в обходе многомерного пространства переменных минимизируемой функции при помощи некоторого набора пробных точек, из которого периодически выбрасываются точки с самыми максимальными значениями функции, а вместо них берутся точки, полученные при помощи некоторой вероятностной комбинации и возмущения точек с минимальными значениями функции. Таким образом, ГА является стохастическим алгоритмом, поэтому в программе MEL выполняются несколько независимых запусков ГА, использующих разные последовательности случайных чисел, и анализируется совпадение результатов их работы.

Генетический алгоритм, используемый в программе MEL, во многом основан на ГА, реализованном нами ранее в программе SOL [8].

При изложении реализации ГА будут использоваться следующие понятия:

- 1) особь — вариант кристаллической решетки с заданными векторами трансляции;
- 2) генотип [особи] — набор переменных, определяющих кристаллическую решетку;

- 3) фитнес [особи] — взятая с противоположным знаком энергия ячейки; чем больше конечный найденный фитнес, тем точнее решена задача;
- 4) популяция — совокупность особей одного шага эволюции после процедуры размножения выжившей популяции;
- 5) выжившая популяция — совокупность особей одного шага эволюции после отсева из популяции особей с низким фитнесом;
- 6) шаг эволюции, или поколение — порядковый номер итерации развития популяции;
- 7) мутация — случайное изменение (возмущение) в генотипе особи;
- 8) прямое наследование — перенос особи из выжившей популяции в популяцию без изменений генотипа;
- 9) комбинированное наследование — получение генотипа новой особи при помощи случайной комбинации генотипов двух особей из выжившей популяции;
- 10) родитель (родители) особи — особь (особи) из выжившей популяции предыдущего поколения, из генотипа (генотипов) которых получен генотип данной особи;
- 11) отбор — удаление особей с низким фитнесом из популяции для получения выжившей популяции следующего поколения;
- 12) элита — небольшое количество особей с наилучшим фитнесом, которые переносятся в следующее поколение без всяких изменений.

Каждый независимый запуск ГА — это последовательное выполнение заданного количества шагов эволюции. Каждый шаг эволюции заключается в

- построении новой популяции по выжившей при помощи: прямого наследования + внесения мутаций; комбинированного наследования + внесения мутаций; равномерно случайного задания генотипов особей (для самого первого шага эволюции, когда еще не существует выжившей популяции);
- переносе элиты из выжившей популяции в популяцию;
- расчете фитнеса особей в популяции;
- отборе максимально лучших и максимально отличающихся по генотипу особей в выжившую популяцию следующего поколения.

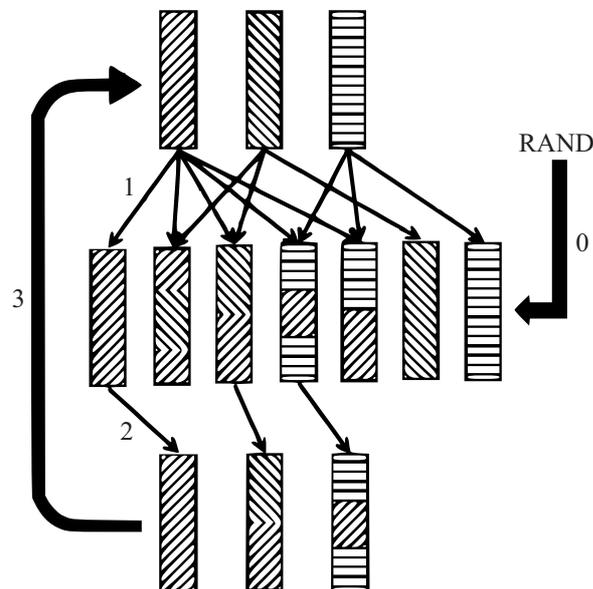


Рис. 4. Иллюстрация выполнения ГА

Схематически один “эволюционный акт” показан на рис. 4, на котором введены следующие обозначения:

- 0) в самом начале ГА популяция инициализируется равномерно случайными особями;
- 1) особи выжившей популяции переносятся в новую популяцию при помощи прямого и комбинированного наследования, затем в них вносятся мутации, элита переносится без изменений;
- 2) для особей в популяции рассчитывается фитнес, затем отбираются хорошие и разнообразные (диверсифицированные) особи в выжившую популяцию следующего поколения;
- 3) переход к следующему шагу эволюции.

Параллельная версия программы MEL реализована с использованием технологий MPI и OpenMP [9, 10]. Такое сочетание позволяет задействовать вычислительные возможности типичного кластерного суперкомпьютера: в пределах одного вычислительного узла параллельная работа программы осуществляется на основе OpenMP, что использует преимущества совместного доступа к общей памяти и быстрой синхронизации, а параллельная работа программы на разных узлах осуществляется при помощи MPI.

В программе MEL параллельным образом: выполняются независимые запуски ГА (MPI-распараллеливание); создаются особи новой популяции и рассчитываются значения фитнеса этих особей (MPI-распараллеливание); вычисляется фитнес одной особи (OpenMP-распараллеливание).

Один MPI-процесс является выделенным (управляющим) и занимается только синхронизацией вычислений, а все остальные MPI-процессы нагружены вычислениями в рамках ГА примерно в одинаковой

степени.

В вырожденном случае количество OpenMP-нитей у одного MPI-процесса может равняться одному. Количество MPI-процессов должно быть больше или равно двум, так как один MPI-процесс является управляющим.

Все MPI-процессы, кроме управляющего, объединены в непересекающиеся фиксированные группы. Каждая такая группа (“рабочая группа”) в каждый момент времени занимается расчетами в пределах одного запуска ГА. Однако время от времени рабочая группа может переключаться на расчет другого запуска ГА. В каждой рабочей группе есть главный процесс, который осуществляет коммуникации с управляющим процессом по принципу точка–точка. Прочие процессы рабочей группы не взаимодействуют напрямую с управляющим процессом. Процессы рабочей группы взаимодействуют между собой при помощи коллективных MPI-операций. Рабочая группа может состоять из одного или нескольких процессов. Рабочих групп может быть одна или несколько.

Управляющий MPI-процесс работает на одном (отдельном) узле, его задача состоит в координации работы остальных MPI-процессов и в обеспечении равномерного выполнения всех независимых запусков ГА. Для этого управляющий MPI-процесс для каждого независимого запуска ГА хранит его текущее состояние: выжившую популяцию и количество уже проделанных шагов эволюции.

Остальные MPI-процессы объединены в группы (“рабочие группы”), каждая рабочая группа получает от управляющего MPI-процесса выжившую популяцию, выполняет несколько шагов эволюции, затем высылает обратно новую выжившую популяцию. Количество таких рабочих групп составляет примерно 80% от количества независимых запусков ГА. Таким образом, в каждый момент времени примерно 80% независимых запусков ГА выполняются, а 20% ждут своей очереди. Кроме того, управляющий MPI-процесс собирает статистику, сколько каждая рабочая группа выполнила шагов эволюции к текущему моменту. Это позволяет перераспределять нагрузку между рабочими группами так, чтобы все независимые запуски ГА завершились примерно одновременно. Для этого:

- оценивается время завершения всех вычислений;
- для каждого свободного запуска ГА оценивается время его завершения на текущей рабочей группе;
- выбирается тот свободный запуск ГА, чье предполагаемое время завершения близко ко времени завершения всех вычислений.

Подобная схема организации параллельной работы позволяет эффективно работать в случае, когда рабочие группы имеют различное быстродействие, например когда из-за некратности количества MPI-процессов и количества независимых запусков ГА в одной рабочей группе один MPI-процесс, а в другой — два MPI-процесса.

В каждой рабочей группе есть главный процесс, ответственный за коммуникации с управляющим процессом, и прочие процессы, не взаимодействующие с управляющим процессом напрямую. Все процессы рабочей группы в равной степени создают особи для новой популяции. Затем они рассчитывают фитнес этих новых особей. Поскольку расчет фитнеса представляет собой суммирование большого числа независимых слагаемых (парных и зависящих от углов вращения), то это суммирование осуществляется при помощи OpenMP-распараллеливания. Затем вся новая популяция с рассчитанными значениями фитнеса собирается целиком в главном процессе рабочей группы, который набирает из них новую выжившую популяцию и отправляет ее управляющему процессу. Таким образом, еще один шаг эволюции оказывается выполненным.

В целях сокращения объема MPI-коммуникаций рабочая группа может делать несколько шагов эволюции, не отправляя результаты управляющему процессу.

В пределах одной рабочей группы коммуникация осуществляется при помощи коллективных MPI-процедур. Схема взаимодействия параллельных процессов в программе MEL показана на рис. 5.

На рис. 5 введены следующие обозначения.

MPI-proc#0 — управляющий процесс; занимается исключительно синхронизацией вычислений.

MPI-proc#1 и MPI-proc#2 — процессы, объединенные в рабочую группу MPI-group#0, из них MPI-proc#2 является главным.

MP — выжившая популяция; управляющий процесс хранит выжившие популяции для всех независимых запусков ГА.

Population — новая популяция для следующего шага эволюции.

Energy — массив значений фитнеса для особей из новой популяции.

Цифры на рисунке означают:

- 1) управляющий MPI-процесс передает главному процессу выжившую популяцию;
- 2) главный процесс раздает остальным процессам рабочей группы выжившую популяцию;

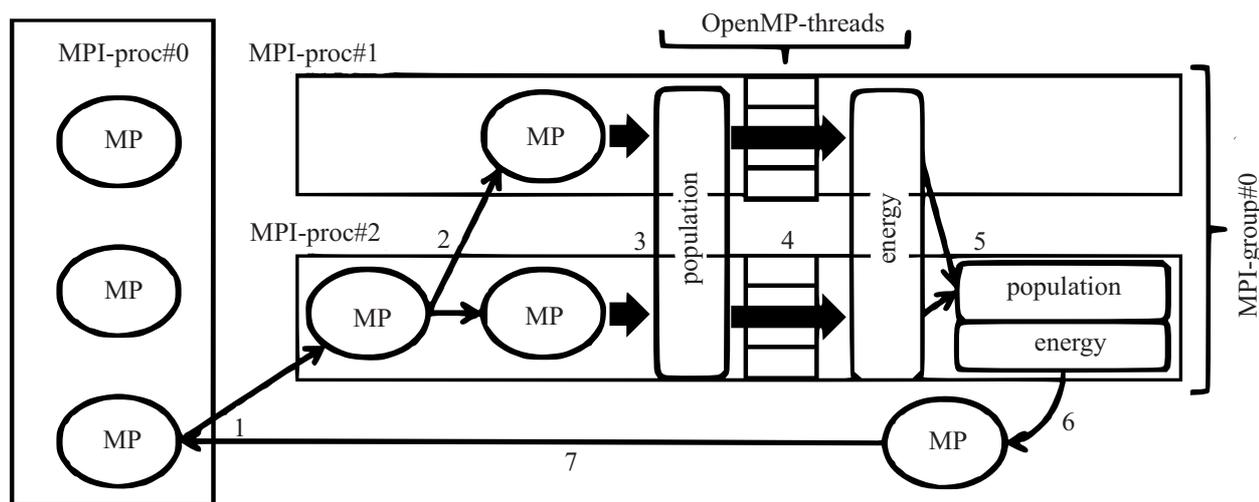


Рис. 5. Иллюстрация взаимодействий между параллельными процессами в программе MEL

- 3) процессы рабочей группы создают новую популяцию;
- 4) процессы рабочей группы рассчитывают фитнес особей из новой популяции, порождая для этого OpenMP-нити (OpenMP-threads);
- 5) новая популяция и рассчитанные значения фитнеса собираются целиком в главном процессе;
- 6) главный процесс набирает особей в новую выжившую популяцию;
- 7) главный процесс отправляет назад управляющему процессу новую выжившую популяцию.

**5. Масштабируемость параллельной программы MEL. Расчет кристалла фуллерена.** Испытания программы MEL проводились на суперкомпьютере “Ломоносов” [11] с использованием узлов на основе двух четырехъядерных процессоров Intel Xeon X5570 2.93 ГГц, соединенных коммуникационной сетью QDR InfiniBand 40 Гбит/сек для MPI-сообщений.

Для компиляции последовательной версии программы MEL использовался ifort версии 12.0.3 20110309. Для компиляции параллельной версии программы MEL использовался ifort версии 12.0.3 20110309 и MPI-библиотека intel mpi версии 4.0.1

При запуске программы на каждом узле выполнялся один MPI-процесс, поэтому каждый MPI-процесс (кроме управляющего) порождал внутри себя 8 OpenMP-нитей. На узле с управляющим процессом выполнялась только одна нить (а не 8). В принципе можно было бы нагрузить полезными вычислениями 7 простаивающих ядер на узле с управляющим процессом, однако:

- это могло бы замедлить работу управляющего процесса из-за конкурентного доступа к памяти;
- целью работы являлось распараллеливание программы на большое количество процессов, на котором простой семи ядер становится незаметным.

Для тестирования масштабируемости параллельной программы MEL была использована задача получения кристалла фуллерена  $C_{60}$ , перспективного для использования в гибридных СЭ в качестве акцептора [12]. Были заданы следующие параметры расчетов: количество независимых запусков ГА 99; размер популяции 16384; число шагов эволюции 499; размер выжившей популяции 13; количество особей в элите 4; соотношение комбинированного наследования к прямому наследованию при создании новой популяции 3:1; количество учитываемых соседей при расчете энергии кристаллической решетки 26, что соответствует учету взаимодействия центральной ячейки с ее отражениями, полученными в результате одной трансляции вдоль каждого вектора ячейки в прямом и обратном направлении.

Основное время работы программы занимает выполнение генетического алгоритма. Ориентировочное время работы последовательной версии программы MEL для использованных параметров составило бы около двух недель. Поэтому при запуске последовательной версии программы MEL было задано не 99 независимых запусков ГА, а 1, 2, 3, 4 и 5 независимых запусков ГА. Так как время работы последовательной программы практически линейно зависит от количества независимых запусков ГА, то по полученным значениям времени работы можно рассчитать время работы для 99 независимых запусков ГА, помножив их на 99, 99/2, 99/3, 99/4 и 99/5 соответственно.

В табл. 1 приведены полученные значения времени работы (time) последовательной программы MEL для разных значений количества независимых запусков ГА (NUMBER\_OF\_RUNS) и их экстраполяция (time\_99) для 99 независимых запусков.

Таблица 1

NUMBER_OF_RUNS	time, c	time_99, c
1	12810.111	1268200.989
2	25813.376	1277762.112
3	38733.548	1278207.084
4	51211.151	1267475.987
5	64027.049	1267735.570

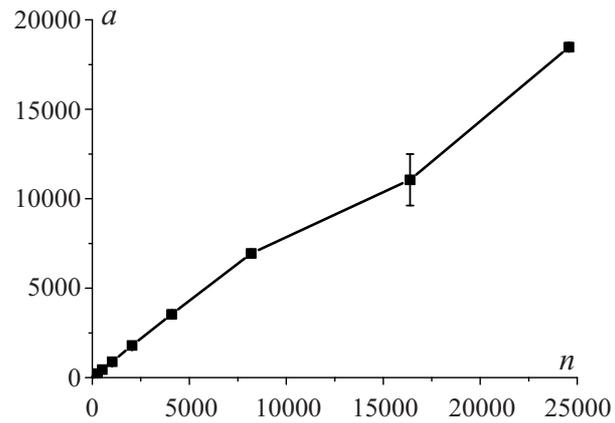


Рис. 6. Зависимость ускорения  $a$  параллельной программы MEL от количества задействованных ядер  $n$

Как видно из табл. 1, экстраполированные времена работы последовательной версии программы MEL для 99 независимых запусков ГА отличаются в третьей значащей цифре. Поэтому в качестве итогового времени работы последовательной версии программы MEL для 99 независимых запусков ГА будем использовать среднюю величину для экстраполированных времен работы и стандартное отклонение среднего значения для оценки погрешности этой величины. Полученное значение времени работы последовательной версии программы MEL  $T_{\text{serial}}$  (с) для 99 независимых запусков ГА рассчитывается по формуле

$$T_{\text{serial}} = (1.272 \pm 0.003) \times 10^6 . \tag{12}$$

При испытаниях параллельной программы MEL было задействовано от 2 до 3072 узлов (с учетом управляющего процесса) или, в пересчете на элементарные ядра, от 16 до 24576 ядер (с учетом управляющего процесса и его 7 простаивающих ядер). Для каждого значения количества узлов запуск был проведен три раза для накопления статистики. Затем по этим трем временам работы программы, полученным для одного и того же количества задействованных ядер, были рассчитаны средние значения со стандартными отклонениями среднего значения. Эти значения приведены в табл. 2.

Для анализа работы параллельной версии программы далее используются две характеристики: ускорение и эффективность. Ускорение показывает, во сколько раз параллельная программа выполняется быстрее, чем последовательная программа. Это отражает непосредственно видимую на практике экономию времени ожидания результатов работы программы. Ускорение работы программы  $a(n)$  на  $n$  ядрах вычисляется по формуле  $a(n) = T_{\text{serial}}/T(n)$ , где  $T(n)$  — время работы параллельной программы на  $n$  ядрах,  $T_{\text{serial}}$  — время работы последовательной программы (на одном ядре).

Эффективность работы программы показывает, какую часть времени работающие ядра занимают полезными вычислениями, а какая часть времени тратится на организацию параллельных вычислений. Если эффективность работы программы низкая, то следует запускать программу на меньшем числе ядер, а на освободившихся ядрах запускать другие программы. Эффективность работы программы  $e(n)$  на  $n$  ядрах вычисляется по формуле  $e(n) = \frac{a(n)}{n-8}$ , где  $a(n)$  — ускорение работы программы на  $n$  ядрах.

Ускорение  $a(n)$  делится именно на  $n-8$ , а не на  $n$ , так как в данном случае один узел с управляющим процессом (т.е. 8 ядер) не занимается полезными вычислениями. Поэтому насыщение производительности будет нагляднее показано именно для указанной формулы эффективности.

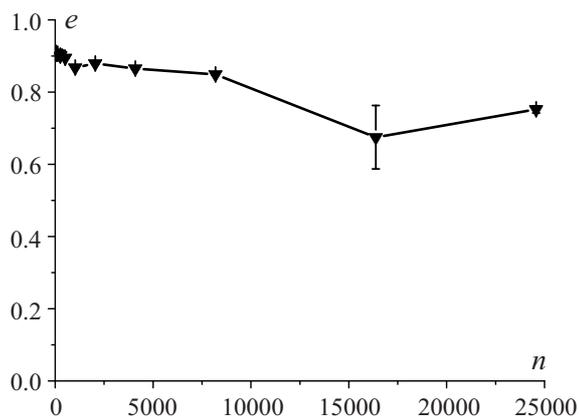
Рассчитанные средние времена  $\langle \text{time} \rangle$  с относительными стандартными отклонениями среднего значения  $\frac{\sigma_{\langle \text{time} \rangle}}{\langle \text{time} \rangle}$ , ускорение  $a$  и эффективность  $e$  работы параллельной программы MEL для разных количеств задействованных ядер  $n$  из формулы (12) и табл. 1 приведены в табл. 2. По этим данным построены график зависимости ускорения программы от числа задействованных ядер (рис. 6) и график зависимости эффективности программы от числа задействованных ядер (рис. 7).

Как видно из рис. 6 и 7, насыщение ускорения работы параллельной программы MEL не было достигнуто на проверенных количествах ядер 16...24576 (2...3072 узлов) и масштабирование довольно близко к линейному.

Использование, например, 8192 ядер (1024 узлов) позволяет сократить полное время работы программы с двух недель (последовательная версия) до четырех минут.

Таблица 2

$n$	$\langle \text{time} \rangle, \text{c}$	$\frac{\sigma_{\langle \text{time} \rangle}}{\langle \text{time} \rangle}$	$a$	$e$
16	175061.0	0.0013	7.265	0.908
32	58299.95	0.0002	21.816	0.909
64	25035.38	0.0004	50.803	0.907
128	11696.23	0.0005	108.742	0.906
256	5693.121	0.0006	223.406	0.901
512	2820.467	0.0003	450.945	0.895
1024	1441.896	0.0017	882.086	0.868
2048	709.627	0.0004	1792.317	0.879
4096	359.490	0.0001	3538.002	0.865
8192	183.138	0.0030	6944.907	0.849
16384	115.043	0.1300	11055.660	0.675
24576	68.874	0.0128	18466.712	0.752

Рис. 7. Зависимость эффективности  $e$  параллельной программы MEL от количества задействованных ядер  $n$ 

Во всех проведенных запусках последовательной и параллельной программы MEL была получена кристаллическая решетка фуллерена, близкая к гексагональной плотноупакованной с расстояниями между центрами молекул около  $10.0 \text{ \AA} - 10.1 \text{ \AA}$  (она показана на рис. 8). Разброс параметров объясняется тем, что последовательность случайных чисел при каждом запуске ГА различна.

Плотность полученной структуры ( $1.6 \text{ г/см}^3$ ) близка к экспериментальному значению  $1.7 \text{ г/см}^3$ , разность объясняется погрешностью силового поля при расчете энергии межмолекулярного взаимодействия [13, 14]. Было проведено исследование зависимости энергии взаимодействия ячейки с ее первыми трансляциями  $U_{\text{tr}}^1$  (подробнее см. (10) и (11) и поясняющий текст) от числа конфигураций  $N$  (рис. 9), которая рассчитывалась как произведение трех основных параметров генетического алгоритма:

$$(\text{количество независимых запусков ГА}) \times (\text{размер популяции ГА}) \times (\text{число шагов эволюции}).$$

Из рис. 9 видно, что энергия слабо меняется при  $N$  порядка  $10^5$ . Такое относительно небольшое число конфигураций объясняется сравнительно простой структурой молекула фуллерена (отсутствие степеней свободы, связанных с вращением вокруг связей), а также тем, что мы ограничились построением структур с одной молекулой в элементарной ячейке. Кроме того, было проведено сравнение  $U_{\text{tr}}^1$  и  $U_{\text{tr}}^2$ , (см. (11) и пояснения в тексте). Получено, что  $U_{\text{tr}}^2$  для молекулы фуллерена составляет менее  $0.01 \text{ ккал/моль}$ . Таким образом, для рассмотренного случая при расчете энергии взаимодействия ячейки с кристаллом достаточно учитывать первые трансляции (величина  $U_{\text{tr}}^1$ ).

**6. Заключение.** В настоящей статье представлен алгоритм построения молекулярных кристаллов, образованных молекулами среднего размера с известной структурой. Моделирование проводится на атомистическом уровне с расчетом энергии взаимодействия между атомами в рамках классического силового поля. Решение этой задачи необходимо для поиска молекул, пригодных для изготовления функциональных слоев гибридных и органических солнечных батарей.

Описана реализация параллельной версии алгоритма в рамках программы MEL, анализируется эффективность параллелизации. Приводятся результаты моделирования кристаллической структуры фуллерена, производные которого используются в качестве материала для функциональных слоев солнечных элементов.

Получены структуры кристаллов фуллерена, близкие к экспериментальным. Показана высокая эффективность распараллеливания использованного алгоритма построения молекулярного кристалла, и продемонстрировано линейное ускорение выполнения задачи с ростом числа вычислительных ядер. Полученные результаты дают основания полагать, что линейная зависимость ускорения от числа вычислительных ядер и высокая эффективность сохраняется при любом числе вычислительных ядер, что делает возможным считать программу MEL сверхмасштабируемой.

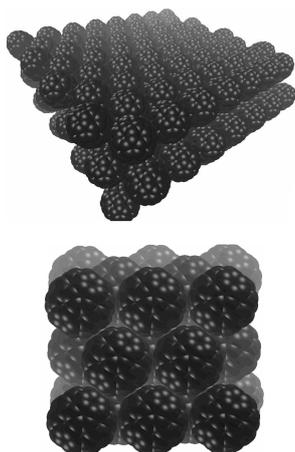


Рис. 8. Пример найденной кристаллической решетки фуллерена, отвечающей минимуму  $U_{\text{cell}}$

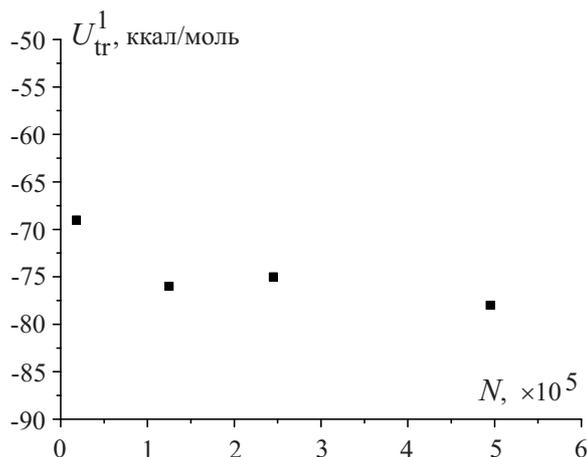


Рис. 9. Зависимость энергии взаимодействия ячейки с ее первыми трансляциями  $U_{\text{tr}}^1$  от числа конфигураций  $N$ , которые были перебраны в рамках моделирования с использованием генетического алгоритма глобальной оптимизации

Работа выполнена при финансовой поддержке Минобрнауки России по государственному контракту от 19.08.2011 г. № 07.514.11.4009 в рамках ФЦП “Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы”.

СПИСОК ЛИТЕРАТУРЫ

1. *de Falco C., Sacco R., Verrì M.* Analytical and numerical study of photocurrent transients in organic polymer solar cells // *Computer Methods in Applied Mechanics and Engineering*. 2010. **199**. 1722–1732.
2. *Cheung D.L., Troisi A.* Modelling charge transport in organic semiconductors: from quantum dynamics to soft matter // *Phys. Chem. Chem. Phys.* 2008. **10**. 5941–5952.
3. *Northrup J.E.* Atomic and electronic structure of polymer organic semiconductors: P3HT, PQT, and PBTTT // *Phys. Rev. B*. 2007. **76**. 245202(1–6).
4. *Goldberg D.E.* Genetic algorithms in search, optimization, and machine learning. Reading: Addison–Wesley, 1989.
5. *Goldberg D.E.* Real-coded genetic algorithm, virtual alphabets, and blocking // *Complex Systems*. 1991. **5**. 139–167.
6. *Goldberg D.E., Deb K.* A comparative analysis of selection schemes used in genetic algorithms // *Foundations of genetic algorithm*. Rawlins G.J.E. (Ed.). San Mateo: Morgan Kaufmann. 1991. 69–93.
7. *Halgren T.A.* Merck molecular force field // *J. Comp. Chem.* 1996. **17**. 490–519.
8. *Романов А.Н., Кондакова О.А., Григорьев Ф.В., Сулимов А.В., Луцкежина С.В., Мартынов Я.Б., Сулимов В.Б.* Компьютерный дизайн лекарственных средств: программа докинга SOL // *Вычислительные методы и программирование*. 2008. **9**, № 2. 64–84.
9. *Антонов А.С.* Параллельное программирование с использованием технологии MPI. М.: Изд-во Моск. ун-та, 2004.
10. *Антонов А.С.* Параллельное программирование с использованием технологии OpenMP. М.: Изд-во Моск. ун-та, 2009.
11. <http://parallel.ru/cluster/lomonosov.html>
12. *Singh T.B., Sariciftci N.S., Yang H., Yang L., Plochberger B., Sitter H.* Correlation of crystalline and structural properties of C-60 thin films grown at various temperature with charge carrier mobility // *Appl. Phys. Lett.* 2007. **90**. 213512.
13. *Керл Р.Ф., Смолли Р.Э.* Фуллерены // *В мире науки*. 1991. **12**. 14–24.
14. *Елецкий А.В., Смирнов Б.М.* Фуллерены и структура углерода // *Успехи физич. наук*. 1995. **165**, № 9. 977–1009.

Поступила в редакцию  
15.08.2012