

УДК 681.3

Э.И. Ватутин, канд. техн. наук, доцент, кафедра вычислительной техники, ЮЗГУ (e-mail: evatutin@rambler.ru)

А.Д. Журавлев, программист, интернет-портал BOINC.ru (e-mail: alexone07@mail.ru)

О.С. Заикин, канд. техн. наук, н.с., ИДСТУ СО РАН, лаборатория дискретного анализа и прикладной логики (e-mail: zaikin.icc@gmail.com)

В.С. Титов, докт. техн. наук, профессор, зав. кафедрой вычислительной техники, ЮЗГУ (e-mail: titov-kstu@rambler.ru)

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ВЗВЕШИВАЮЩИХ ЭВРИСТИК В ЗАДАЧЕ ПОИСКА ДИАГОНАЛЬНЫХ ЛАТИНСКИХ КВАДРАТОВ

Приведено описание стратегии использования стохастических методов на базе взвешивающих эвристик в задаче поиска диагональных латинских квадратов. Показано, что исходная задача, относящаяся к классу задач на существование решения, может быть эффективно сведена к задаче на оптимальность, являющейся задачей дискретной комбинаторной оптимизации, и, следовательно, допускающей применение при ее решении эвристических подходов, соответствующих методам случайного и взвешенного случайного перебора, а также алгоритма муравьиной колонии. В статье приведены соответствующие взвешивающие эвристики, рассмотрены особенности метаоптимизации и даны результаты применения указанных эвристических подходов при поиске диагональных латинских квадратов различного порядка. С использованием метода полного перебора с ограничением на число анализируемых ветвей показано, что в пространстве поиска решения расположены неравномерно, что позволяет использовать эвристические методы с целью выполнения разведки пространства с последующим уточнением его перспективных областей.

Ключевые слова: латинские квадраты, дискретная комбинаторная оптимизация, эвристические методы, метод взвешенного случайного перебора, алгоритм муравьиной колонии, метаоптимизация.

Одной из интересных комбинаторных задач является задача построения диагональных латинских квадратов заданного порядка N . В общем случае неизвестен полиномиальный алгоритм для перечисления всех возможных диагональных латинских квадратов для произвольного N . В частности, если точное количество латинских квадратов порядка 10 известно [1], то для диагональных латинских квадратов порядка 10 этот факт еще не установлен. Из этого следует, что задача построения диагональных латинских квадратов порядка 10 в достаточной степени интересна с позиции возможности применения для ее решения различных комбинаторных подходов, основанных на использовании эвристик. В данной статье приведены результаты использования эвристик, соответствующих случайному перебору [2–3], взвешенному случайному перебору [4] и алгоритму муравьиной колонии [5–7] в указанной задаче. Данная группа эвристик может быть эффективно применена при последовательном итеративном заполнении элементов латинского квадрата без каких-либо модификаций уже полученных решений (например, с использованием метода имитации отжига и др. [8–9]), комбинаторных возвратов [10] и т.п. приемов.

Латинский квадрат (ЛК) [11] порядка N является частным случаем магического квадрата и представляет собой матрицу $A = \|a_{ij}\|$, $i, j = \overline{1, N}$, элементами a_{ij} которой являются элементы некоторого множества S мощности $N = |S|$ (для определенности далее будем полагать, что $S = \{0, 1, 2, \dots, N-1\}$). По определению в каждой строке и в каждом столбце латинского квадрата каждый из элементов множества S встречается в точности один раз:

$$\forall i, j, k = \overline{1, N}, j \neq k : (a_{ij} \neq a_{ik}) \wedge (a_{ji} \neq a_{ki}). \quad (1)$$

Для диагональных латинских квадратов (ДЛК), являющихся специальным видом латинских квадратов, по определению дополнительно вводятся требования на отсутствие совпадающих элементов на главной и побочной диагонали латинского квадрата:

$$\forall i, j = \overline{1, N}, i \neq j : (a_{ii} \neq a_{jj}) \wedge (a_{N-i+1, N-i+1} \neq a_{N-j+1, N-j+1}). \quad (2)$$

Латинский квадрат называется нормализованным, если элементы его первой строки и его первого столбца упорядочены по возрастанию. Несложно показать, что путем биективной перенумерации (перестановки) элементов множества S любого корректного латинского квадрата и последующей перестановки строк можно добиться его нормализации (указанное множество квадратов представляет собой класс эквивалентности из $N! \times (N-1)!$ латинских квадратов (множитель $N!$ соответствует перестановкам элементов множества S , а множитель $(N-1)!$ – перестановкам строк квадрата со второй по N -ю). Для случая ДЛК процедура нормализации (представляющая собой последовательность перестановок строк и столбцов) может нарушить условие диагональности. При этом упорядочивание по возрастанию любой строки или столбца ДЛК может быть сделано с помощью последовательности перестановок элементов, т.к. эта процедура не нарушает условие диагональности. Далее мы будем рассматривать ДЛК с упорядоченной первой строкой, т.е. строго говоря, такие ДЛК в общем случае не будут являться нормализованными. Обладая одним корректным ДЛК такого вида, можно тривиально получить остальные ДЛК, входящие с ним в один класс эквивалентности и эквивалентных ему с точностью до перенумерации элементов. Примеры соответственно нормализованного ЛК порядка 10 и ДЛК порядка 10 с упорядоченной первой строкой приведены ниже:

$$\left(\begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 9 & 4 & 3 & 6 & 7 & 8 & 0 & 8 \\ 2 & 9 & 3 & 1 & 7 & 0 & 5 & 8 & 4 & 6 \\ 3 & 4 & 1 & 2 & 8 & 7 & 9 & 6 & 5 & 0 \\ 4 & 3 & 5 & 9 & 2 & 1 & 8 & 0 & 6 & 7 \\ 5 & 6 & 4 & 8 & 1 & 2 & 0 & 9 & 7 & 3 \\ 6 & 5 & 8 & 7 & 0 & 3 & 2 & 1 & 9 & 4 \\ 7 & 8 & 6 & 0 & 9 & 4 & 1 & 2 & 3 & 5 \\ 8 & 7 & 0 & 5 & 6 & 9 & 3 & 4 & 1 & 2 \\ 9 & 0 & 7 & 6 & 5 & 8 & 4 & 3 & 2 & 1 \end{array} \right), \left(\begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 7 & 2 & 4 & 9 & 0 & 6 & 5 & 1 & 3 & 8 \\ 8 & 3 & 6 & 7 & 5 & 9 & 0 & 2 & 4 & 1 \\ 2 & 6 & 8 & 5 & 1 & 7 & 4 & 0 & 9 & 3 \\ 5 & 8 & 9 & 1 & 7 & 0 & 3 & 4 & 6 & 2 \\ 9 & 4 & 1 & 2 & 8 & 3 & 7 & 6 & 0 & 5 \\ 4 & 7 & 5 & 6 & 9 & 1 & 8 & 3 & 2 & 0 \\ 3 & 0 & 7 & 8 & 2 & 4 & 1 & 9 & 5 & 6 \\ 6 & 5 & 0 & 4 & 3 & 2 & 9 & 8 & 1 & 7 \\ 1 & 9 & 3 & 0 & 6 & 8 & 2 & 5 & 7 & 4 \end{array} \right).$$

Несложно заметить, что рассматриваемые задачи построения ЛК и ДЛК имеют ярко выраженную комбинаторную природу, следствием чего является возможность использования для ее решения известных подходов, основанных на применении эвристик. Указанная задача относится к категории задач на существование (либо на подсчет, если по условию требуется определить число ЛК с заданными свойствами), для решения которых взвешивающие эвристики обычно не применяются, вместо чего используются вариации перебора с ограничениями [12–14] либо полиномиальное сведение данной задачи к другим задачам (например, к раскраске графов [15] или SAT-задачам [16]). Однако указанная комбинаторная задача также может быть эффективно сведена к оптимизационной задаче путем минимизации числа нарушений условий (1) и (2) (схожий принцип может быть применен, например, при эвристической проверке пары графов на изоморфизм [17]):

$$C(A) \rightarrow \min,$$

что, в свою очередь, позволяет использовать для решения рассматриваемой задачи различные взвешивающие эвристики. Если в ходе применения какого-либо из эвристических подходов на их базе выполняется условие $C(A) = 0$, это является свидетельством нахождения решения без нарушений, соответствующего всем ограничениям, в противном случае (при $C(A) > 0$) – об отсутствии решений без нарушений или о неэффективности выбранного подхода.

При построении решения будем использовать принцип последовательного заполнения ячеек квадрата в направлении слева направо и затем сверху вниз $(a_{11}, a_{12}, \dots, a_{1N}, a_{21}, a_{22}, \dots, a_{2N}, \dots, a_{NN})$, причем первую строку будем задавать в виде $[012 \dots N-1]$ с целью получения ДЛК указанного выше вида. Для каждого очередного элемента квадрата a_{ij} будем производить определение множества $S_{ij} \subseteq S$ допустимых элементов, не нарушающих условий (1) и (2), с последующим выбором одного из них с

использованием какой-либо эвристики. Если $S_{ij} = \emptyset$, то выбор очередного элемента квадрата без нарушения условий (1) и (2) невозможен, что будем отмечать символом «←» и инкрементом значения $C(A)$. Пример подобной ситуации для ЛК приведен ниже:

$$A = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 9 & 0 & 6 & 1 & 5 & 3 & 8 & 2 & 7 & 4 \\ 8 & 3 & 0 & 5 & 7 & 1 & 2 & 6 & 4 & - \\ 2 & 4 & 5 & 9 & 1 & 0 & 7 & 8 & 6 & 3 \\ 1 & 2 & 7 & 6 & 9 & 4 & 3 & 0 & 5 & 8 \\ 7 & 8 & 4 & 2 & 0 & 9 & 5 & 1 & 3 & 6 \\ 6 & 9 & 8 & 4 & 3 & 7 & 1 & 5 & 0 & 2 \\ 4 & 6 & 9 & 8 & 2 & - & 0 & 3 & 1 & 5 \\ 3 & 5 & 1 & 7 & 6 & 8 & 9 & 4 & 2 & 0 \\ 5 & 7 & 3 & 0 & 8 & 2 & 4 & 9 & - & 1 \end{pmatrix}, C(A) = 3.$$

В приведенном примере при последовательном заполнении элементов квадрата возникновение нарушений обеспечивается элементами $a_{3,10}$, $a_{8,6}$ и $a_{10,9}$ благодаря нарушению условия (1). Так, например, для элемента $a_{3,10}$ множество стоящих левее элементов строки $S_{3,10}^L = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$, стоящих выше – $S_{3,10}^U = \{4, 9\}$, и, соответственно, множество доступных элементов $S_{3,10} = S \setminus (S_{3,10}^L \cup S_{3,10}^U) = \emptyset$.

Наиболее простой эвристический подход – случайный перебор (англ. Random Search, сокр. RS) – базируется на идее равновероятного выбора одного из элементов множества $S_{ij} = \{s_{ij}^1, s_{ij}^2, \dots, s_{ij}^M\}$, $M(S_{ij}) = |S_{ij}|$, а соответствующая ему эвристика может быть записана в виде

$$f_{RS}(s_{ij}^l) = r_k, l = \overline{1, M(S_{ij})}, \quad (3)$$

где r_k – очередное псевдослучайное число. С использованием данной эвристики был организован вычислительный эксперимент, в ходе которого с использованием разработанной однопоточной программной реализации был осуществлен поиск ДЛК различного порядка с ограничением на время эксперимента равным 30 с (здесь и далее для экспериментов был использован процессор Intel Atom N270 @ 1,6 ГГц, ядро Diamondville).

Таблица 1

Число решений с $C(A) \leq 1$, найденных методом случайного перебора, в зависимости от размерности задачи (порядка квадрата N)

N	Число найденных решений
-----	----------------------------

6	68
7	4
8	1
9	0
10	0

Результаты эксперимента показывают, что с ростом размерности задачи эффективность случайного перебора существенно падает и для поиска решений, удовлетворяющих условиям (1) и (2), могут потребоваться существенные затраты машинного времени. С целью повышения эффективности необходимо какое-либо эвристическое ранжирование элементов множеств S_{ij} с целью выбора наиболее перспективных из них ввиду стремления к минимизации числа потенциальных нарушений для последующих элементов формируемого квадрата без выполнения время затратных проверок (которые в теории могут быть выполнены, например, методом полного перебора, однако на практике, как правило, неприемлемо трудоемки). Для этого попытаемся оценить число возможных вариантов выбора различных элементов для еще не заполненных ячеек квадрата, что для ЛК может быть представлено в следующем виде:

$$f_{ij}^{(x)} = \underbrace{\sum_{k=j+1}^N |S_{ik}|}_{\text{по строке}} + \underbrace{\sum_{k=i+1}^N |S_{kj}|}_{\text{по столбцу}} = \sum_{k=j+1}^N |S_{ik}| + (N-i)|S_{i+1,j}| \rightarrow \max, \quad (4)$$

где $x \in S_{ij}$ – один из элементов множества S , помещение которого в (i, j) -ю ячейку не приводит к нарушению условия (1) и, соответственно, к появлению нарушения. Пример вычисления значения функции (4) показан на рис. 1.

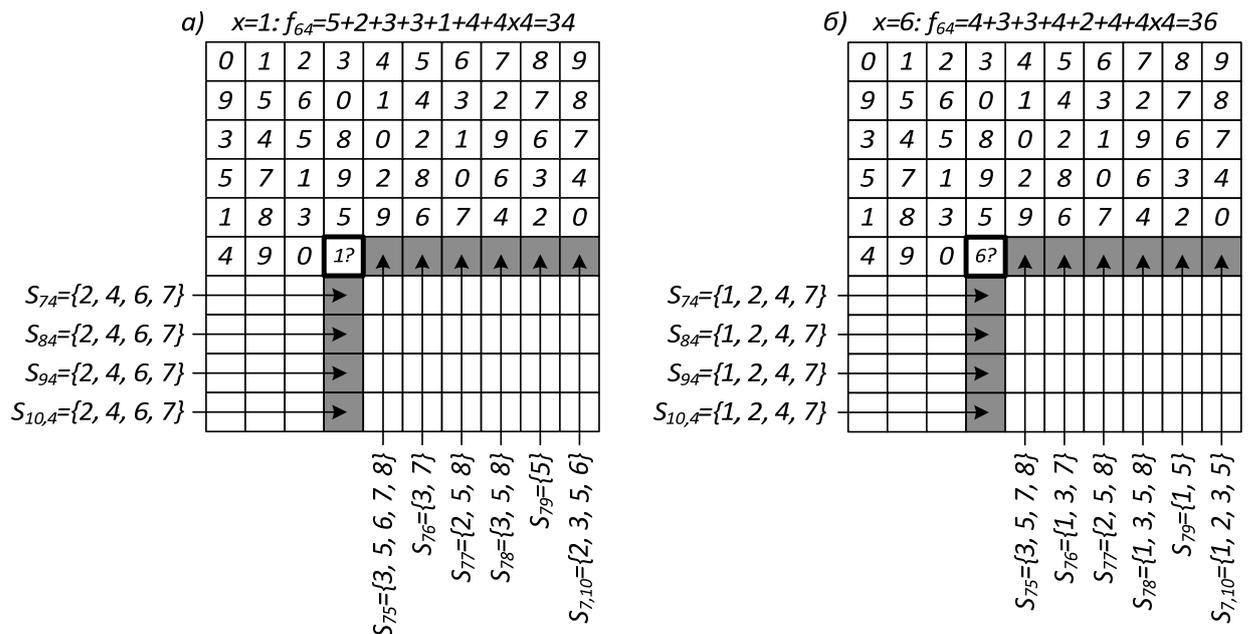


Рис. 1. Пример, поясняющий вычисления эвристики (4) для ЛК на примере выбора значения ячейки a_{64} . Цифрами указаны заполненные ранее ячейки;

жирным выделена текущая ячейка, для которой производится выбор элемента x ; выбор значений в белых пустых ячейках не зависит от текущего значения x , в отличие от серых пустых ячеек

При поиске ДЛК необходим учет возможных диагональных элементов, а число возможных вариантов выбора различных элементов для еще не заполненных ячеек квадрата может быть представлено в следующем виде:

$$g_{ij}^{(x)} = f_{ij}^{(x)} + \alpha_{ij} \underbrace{\sum_{k=i+1}^N |S_{kk}|}_{\text{главная}} + \beta_{ij} \underbrace{\sum_{k=i+1}^N |S_{k, N-k}|}_{\text{побочная}} \rightarrow \max, \quad (5)$$

где $\alpha_{ij} = \begin{cases} 0, i \neq j \\ 1, i = j \end{cases}$ – двоичный признак того, что (i, j) -я ячейка расположена

на главной диагонали, $\beta_{ij} = \begin{cases} 0, i + j \neq N \\ 1, i + j = N \end{cases}$ – на побочной диагонали.

При рассмотрении данной задачи с позиции дерева комбинаторного перебора использование эвристик (4) и (5) приводит к выбору элементов с максимальным числом ветвей в поддеревьях (максимальной арностью узлов на нижележащих уровнях), что теоретически должно обеспечивать большую вероятность получения корректного решения. Простейшим способом практического применения эвристик является их использование в рамках жадного подхода (англ. Greedy, сокр. G) путем использования формул (4) или (5) в зависимости от задачи в неизменном виде:

$$f_G(s_{ij}^l) = \overline{g_{ij}^{(s_{ij}^l)}} \rightarrow \max, l = 1, M(s_{ij}). \quad (6)$$

Данный подход был опробован на практике при поиске ДЛК различного порядка, результаты представлены в табл. 2.

Таблица 2

Число нарушений в решении, найденном с использованием жадного подхода, в зависимости от размерности задачи (порядка квадрата N) при поиске ДЛК

N	Число нарушений
3	2
4	3
5	4
6	5
7	5
8	8
9	12
10	12

Найденные решения для случаев $N = 3, 4, 5$ и 10 приведены ниже:

$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & - & 0 \\ 1 & 0 & - \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 3 \\ 3 & 2 & 0 & 1 \\ 2 & - & 3 & 0 \\ 1 & 0 & - & - \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 2 & 0 & 1 & 3 \\ 3 & 0 & - & 2 & 1 \\ 2 & 3 & 4 & - & 0 \\ - & 4 & 3 & 0 & - \end{pmatrix} \text{ и}$$

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 9 & 7 & 0 & 6 & 5 & 8 & 4 & 3 & 2 & 1 \\ 8 & 0 & 6 & 7 & 9 & 4 & 5 & 1 & 3 & 2 \\ 7 & 8 & 5 & 4 & 0 & 9 & 3 & 6 & 1 & - \\ 6 & 5 & 8 & 0 & 3 & 7 & 9 & 4 & - & - \\ 5 & 6 & 7 & 8 & - & 2 & 0 & 9 & 4 & 3 \\ 4 & 3 & 9 & 5 & 6 & 0 & 8 & 2 & 7 & - \\ 3 & 4 & - & 9 & 7 & 6 & 2 & 5 & 0 & 8 \\ 2 & - & 4 & 1 & 8 & 3 & 7 & 0 & 9 & 6 \\ - & 2 & 3 & - & 1 & - & - & 8 & 5 & - \end{pmatrix}.$$

Найденные решения далеки от корректных, поэтому в чистом виде эвристики (4) и (5) или им подобные использовать нельзя (в противном случае можно было бы утверждать, что найден полиномиальный алгоритм нахождения ДЛК на их базе). Однако они допускают использование совместно со стратегией взвешенного случайного перебора (англ. Wighted Random Search, сокр. WRS) [4], эвристика для которой может быть записана как

$$f_{WRS}(s_{ij}^l) = g_{ij}^{(s_{ij}^l)} \cdot (1 + 2d(r_k - 0,5)) \rightarrow \max, l = 1, \overline{M(S_{ij})}, \quad (7)$$

где d – настроечный параметр, определяющий степень разброса относительно жадного значения $g_{ij}^{(s_{ij}^l)}$. При ее использовании прежде всего необходимо выполнить метаоптимизацию, в ходе которой необходимо найти оптимальное значение степени разброса d^* . Для этого был организован вычислительный эксперимент, в рамках которого определялось число n ДЛК порядка 6, найденное за 30 с, в зависимости от различных значений d (рис. 2).

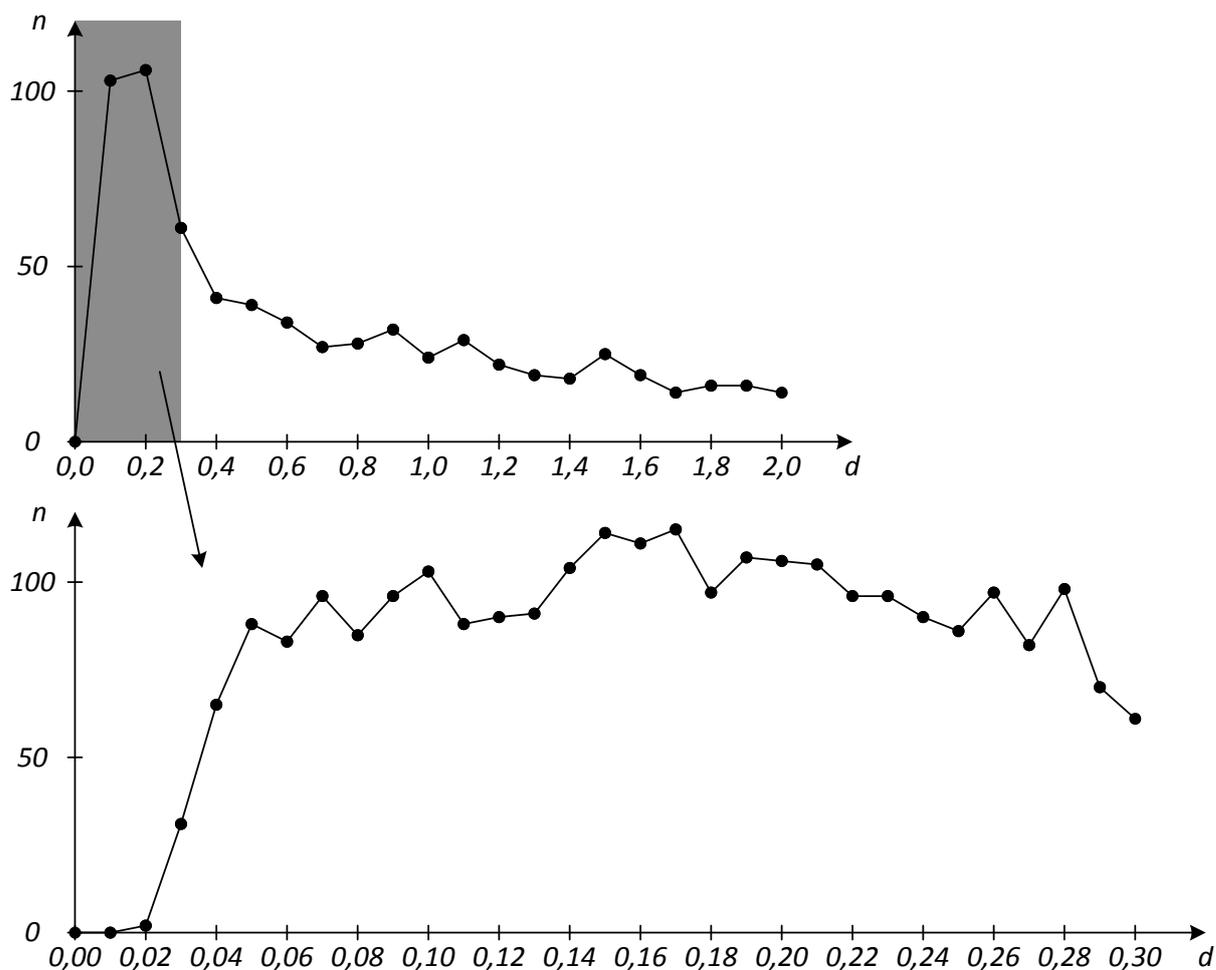


Рис. 2. Зависимость числа n найденных ДЛК от степени разброса d

Полученные зависимости $n(d)$ позволяют сделать вывод о том, что оптимальное значение степени разброса лежит в пределах $0,15 < d < 0,21$, для дальнейших экспериментов было выбрано значение $d^* = 0,17$. С его использованием был организован вычислительный эксперимент, аналогичный рассмотренному выше эксперименту с применением метода случайного перебора, число найденных ДЛК приведено в табл. 3.

Таблица 3
 Число решений с $C(A) \leq 1$, найденных методами случайного и взвешенного случайного перебора, в зависимости от размерности задачи (порядка квадрата N) при поиске ДЛК

N	RS	WRS
6	68	839
7	4	10
8	1	4
9	0	3
10	0	0

Полученные результаты позволяют сделать вывод о том, что использование эвристик вида (4) и (5) совместно со стохастическими методами оправдано: несмотря на большие вычислительные затраты, необходимые на построение одного решения, в целом число решений, полученных в единицу времени, также оказывается большим. Однако при построении ДЛК порядка 10 отведенных затрат вычислительного времени оказывается недостаточно для нахождения хотя бы одного решения. При этом опыт расположения элементов прошлых решений, найденных на предыдущих итерациях, никак не используется при построении последующих решений. Данный недостаток можно исправить путем использования эвристики, соответствующей муравьиной [5–7] стратегии построения решения (англ. Ant Colony, сокр. AC), при которой, следуя работе [7], производится пометка феромоном соответствий в двудольном графе, одной доле которого соответствуют $(N-1) \times N$ ячеек квадрата (первая строка фиксирована!), а другой – N элементов множества S :

$$p_{ij}^{(x)} = \left[g_{ij}^{(x)} \right]^{\alpha} \left[\tau_{ij}^{(x)} \right]^{\beta} r_k \rightarrow \max, \quad (8)$$

$$f_{AC}(s_{ij}^l) = p_{ij}^{(s_{ij}^l)} \rightarrow \max, l = 1, M(S_{ij}) \quad (9)$$

где $\tau_{ij}^{(x)}$ – количество феромона на ребре двудольного графа между (i, j) -м элементом квадрата и x -м элементом множества S , α и β – настроечные параметры, определяющие реакцию муравьев на приращение качества решения (эвристика $g_{ij}^{(x)}$) и количество феромона от предыдущих поколений $\tau_{ij}^{(x)}$. Как уже было отмечено в работе [6], использование эвристики в виде (8) и (9) позволяет обойтись без нормирования вероятностей $p_{ij}^{(x)}$ выбора элементов x множества S_{ij} , что незначительно экономит вычислительные ресурсы. Начальное значение феромона на ребрах будем выбирать равным τ_0 (настроечный параметр), после выполнения t -й итерации движения муравьев и получения в ее результате квадрата A_t с числом нарушений $C(A_t)$ будем производить испарение феромона $\left[\tau_{ij}^{(x)} \right]^{(t)} = \gamma \left[\tau_{ij}^{(x)} \right]^{(t-1)}$, где настроечный параметр $0 < \gamma \leq 1$ регулирует темп испарения, и добавление новой порции феромона $\Delta\tau = \frac{Q}{C(A_t) + 1}$ для ребер, соответствующих выбранным элементам квадрата, где Q – количество феромона в распоряжении одного муравья (настроечный параметр), единица в знаменателе добавляется к функции $C(A_t)$ с целью избегания деления на ноль при нахождении корректного решения.

Перед использованием данной «муравьиной» эвристики на практике, как и в случае взвешенного случайного перебора, необходимо выполнение метаоптимизации, которая в данном случае является более времязатратной

ввиду необходимости поиска экстремума в пространстве, образованном пятью настроечными параметрами $(\alpha, \beta, \tau_0, \gamma, Q)$. В качестве метода оптимизации был использован метод покоординатного спуска, в результате чего был найден следующий набор оптимальных значений настроечных параметров: $\alpha^* = 100, \beta^* = 1,1, \tau_0^* = 1,0, \gamma^* = 0,99999, Q = 1,0$. Соответствующие зависимости числа решений, найденных за 10 000 итераций работы алгоритма, подтверждающие правильность выбора значений настроечных параметров, приведены на рис. 3.

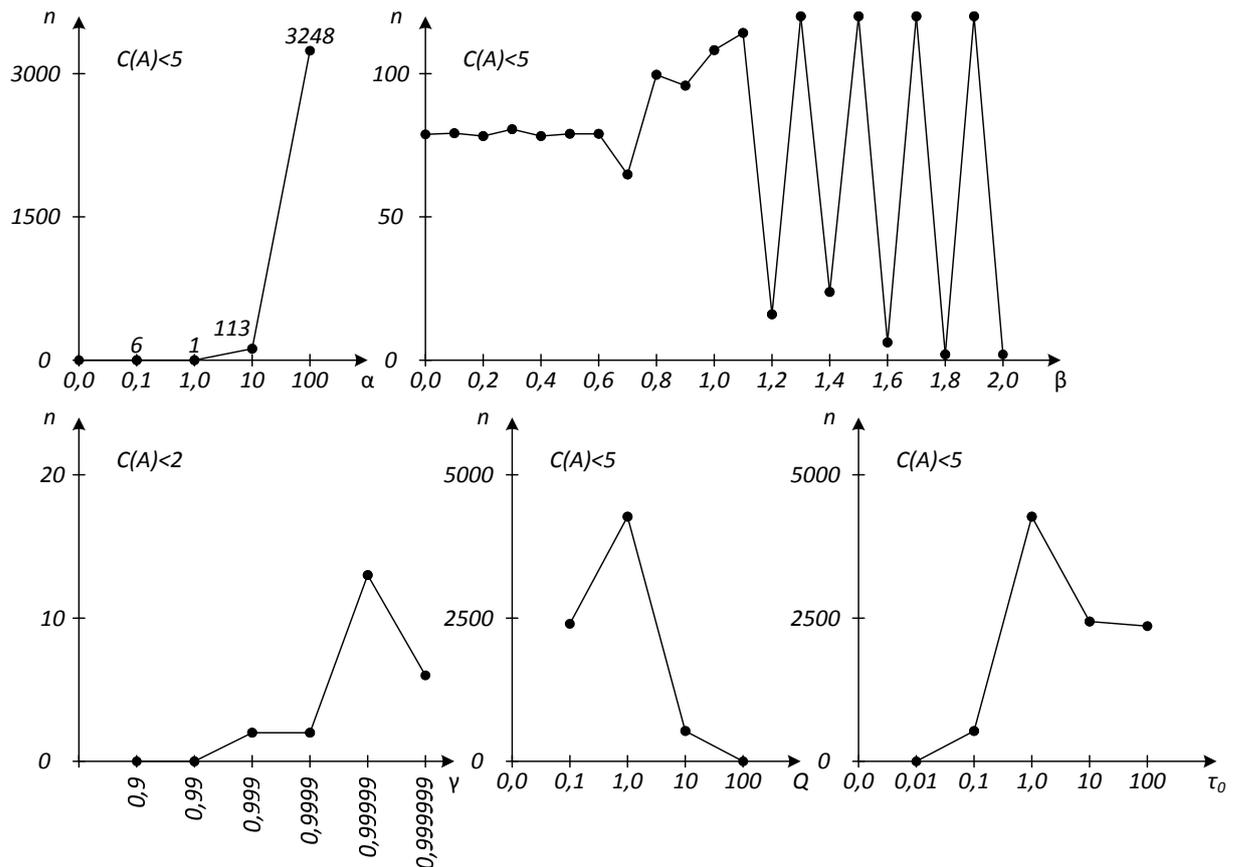


Рис. 3. Результаты метаоптимизации подхода на базе «муравьиной» эвристики

В выполненных экспериментах размер колонии был выбран равным 1 ввиду того, что полученные результаты даже с учетом данного упрощения базового алгоритма [5] оказываются весьма неплохими. Приведенные результаты метаоптимизации позволяют сделать ряд интересных выводов. Прежде всего, в указанной задаче баланс между важностью эвристики (5) (параметр α) и опытом прошлых поколений (параметр β) явно смещен в пользу первого компонента, что эмпирически подтверждает правильность идеи, лежащей в основе эвристики (5). Значение параметра α по результатам вычислительных экспериментов хотелось бы дополнительно увеличить, однако при этом начинают появляться ошибки, связанные с возникновением переполнения при выполнении расчетов с плавающей точкой (англ. floating

point overflow). Параметр β оказывает гораздо меньшее значение на результат поиска, однако полное его исключение ($\beta = 0$) нецелесообразно ввиду уменьшения числа найденных решений (в организованном эксперименте с 3 428 при $\beta = 1,1$ до 2 363 при $\beta = 0$). С другой стороны, при $\beta > 1$ возможны ситуации, когда число решений значительно снижается, что связано с попаданием алгоритма в локальные экстремумы, ввиду чего в качестве результата оптимизации выбрано компромиссное значение $\beta^* = 1,1$. То же самое поведение наблюдается для параметра γ , значение которого выбирались близкими к единице, но не превосходящими ее. При значениях $0,9 \leq \gamma \leq 0,99$ большинство решений далеки от корректных по числу нарушений ($C(A) \geq 2$) ввиду попадания в локальные экстремумы, в то время как дополнительное снижение темпа испарения феромона вызывает рост числа различных решений, ввиду чего было выбрано значение $\gamma^* = 0,99999$. Для зависимостей числа решений n от значений Q и τ_0 имеют место вполне определенные максимумы (см. рис. 3), позволяющие однозначно выбрать их оптимальные значения.

С использованием найденных значений настроечных параметров алгоритма муравьиной колонии был организован вычислительный эксперимент по поиску ДЛК различного порядка, его результаты приведены в табл. 4 (затраты вычислительного времени и процессор соответствуют рассмотренным выше экспериментам, результаты которых приведены в табл. 1 и 3).

Таблица 4

Число решений с $C(A) \leq 1$, найденных методами случайного, взвешенного случайного перебора и алгоритмом муравьиной колонии, в зависимости от размерности задачи (порядка квадрата N) при поиске ДЛК

N	RS	WRS	AC
6	68	839	9 027
7	4	10	10 050
8	1	4	64
9	0	3	20
10	0	0	6

Анализ результатов показывает высокую эффективность алгоритма муравьиной колонии в рассматриваемой задаче поиска ДЛК по сравнению с алгоритмами случайного и взвешенного случайного перебора. К его недостаткам можно отнести особенность, связанную с возможностью нахождения на разных итерациях работы совпадающих квадратов, вероятность чего при использовании случайного и взвешенного случайного перебора близка к нулю.

С целью сопоставления качества решений эвристических методов с результатами переборных методов также был реализован метод полного перебора с ограничением на число анализируемых ветвей дерева комбинаторного перебора (англ. Limited Brute Force, сокр. LBF). Ограничение было реализовано ввиду невозможности перебора всех ветвей дерева за разумное время. Эксперименты с его программной реализацией показали, что решения (ДЛК) расположены в пространстве поиска существенно неравномерно и во многом определяются, например, начальными значениями элементов квадрата. Так при выборе в качестве начальных значений второй строки следующих элементов (9 7 0 6 5 ? ? ? ? ?) среднее время между нахождением различных квадратов составило 132 мс, а при выборе диапазона (2 3 1 0 5 ? ? ? ? ?) – около 300 мс (для нахождения первого решения при этом необходимо около 1 минуты, далее решения начинают находиться более менее равномерно). Анализ решений, полученных с использованием алгоритма муравьиной колонии показывает, что данный подход по видимому «чувствует» данную особенность задачи, т.к. большинство решений, найденных с его использованием при различных начальных значениях генератора псевдослучайных чисел, имеют схожую структуру – например, элементы второй строки квадрата для них имеют следующий вид:

(9 2 0 7 8 6 4 1 3 5),
 (9 5 0 6 8 7 3 2 1 4),
 (9 2 0 6 8 7 3 1 4 5),
 (9 6 0 8 7 3 1 4 2 5),
 (9 3 0 8 7 6 2 1 4 5).

Не сложно заметить, что элемент a_{21} всегда принимает значение 9, a_{23} – всегда 0, элемент $a_{2,10}$ часто имеет значение 5 и т.д. Эта особенность задачи может быть использована следующим образом: эвристические методы (RS, WRS, AC и пр.) могут применяться для нахождения перспективных областей в пространстве поиска, в окрестности которых может быть организовано уточнение решений с использованием стратегии LBF.

С целью более масштабного сопоставления рассмотренных выше эвристических методов был организован вычислительный эксперимент, в ходе которого оценивалось количество ДЛК порядка 10, найденных за $t = 16$ часов на процессоре Intel Core i7 4770 @ 3,9 ГГц (ядро Haswell) при использовании однопоточной программной реализации.

Таблица 5

Результаты нахождения ДЛК порядка 10, выраженные в числе найденных квадратов

Метод	$C(A) \leq 1$	$C(A) = 0$
RS	7 934	124
WRS	13 341	352
AC	179 387	9 422
LBF	– (не оценивалось)	346 572

Таблица 6

Примеры ДЛК порядка 10, найденные в ходе эксперимента

RS:	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
	7 2 4 9 0 6 5 1 3 8	6 9 8 0 3 4 1 5 2 7	6 7 5 8 9 2 0 4 1 3
	8 3 6 7 5 9 0 2 4 1	2 7 5 6 9 1 3 4 0 8	5 0 4 2 7 3 1 6 9 8
	2 6 8 5 1 7 4 0 9 3	5 3 1 7 8 9 0 6 4 2	8 4 7 6 2 9 3 1 0 5
	5 8 9 1 7 0 3 4 6 2	1 2 7 9 6 8 4 0 3 5	1 6 3 4 8 7 5 9 2 0
	9 4 1 2 8 3 7 6 0 5	9 0 4 5 7 3 8 2 6 1	2 3 6 9 0 1 8 5 7 4
	4 7 5 6 9 1 8 3 2 0	4 8 3 1 0 7 2 9 5 6	3 8 0 5 1 6 9 2 4 7
	3 0 7 8 2 4 1 9 5 6	7 4 6 2 1 0 5 8 9 3	7 9 8 0 5 4 2 3 6 1
	6 5 0 4 3 2 9 8 1 7	8 5 9 4 2 6 7 3 1 0	9 2 1 7 3 8 4 0 5 6
	1 9 3 0 6 8 2 5 7 4	3 6 0 8 5 2 9 1 7 4	4 5 9 1 6 0 7 8 3 2
WRS:	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
	4 6 9 2 5 3 0 8 7 1	2 9 8 0 1 6 7 3 4 5	3 8 0 2 7 4 1 9 6 5
	2 3 4 9 0 6 7 5 1 8	1 2 6 9 7 0 4 5 3 8	5 4 9 8 0 6 7 2 1 3
	9 5 3 8 6 1 2 4 0 7	9 6 5 4 0 3 8 1 7 2	8 2 5 6 9 3 4 1 7 0
	8 7 5 1 9 0 3 2 4 6	4 5 7 2 8 1 3 9 6 0	2 9 7 1 5 8 3 0 4 6
	6 8 0 5 3 7 4 1 9 2	5 8 3 1 2 7 9 4 0 6	1 5 8 4 3 7 9 6 0 2
	3 0 7 6 2 8 1 9 5 4	8 0 4 7 3 9 5 6 2 1	6 7 3 5 1 0 2 8 9 4
	5 2 8 7 1 4 9 3 6 0	3 4 0 6 9 8 1 2 5 7	9 3 1 7 6 2 0 4 5 8
	7 4 1 0 8 9 5 6 2 3	7 3 9 5 6 2 0 8 1 4	4 0 6 9 2 1 8 5 3 7
	1 9 6 4 7 2 8 0 3 5	6 7 1 8 5 4 2 0 9 3	7 6 4 0 8 9 5 3 2 1
AC:	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
	9 7 0 5 6 8 1 4 2 3	9 7 0 6 8 4 1 2 5 3	9 3 0 6 7 8 1 5 2 4
	7 2 6 0 9 1 3 5 4 8	7 5 8 0 9 1 2 6 3 4	2 9 5 0 8 1 7 3 4 6
	2 6 3 8 7 9 0 1 5 4	8 6 3 4 7 9 0 5 1 2	5 4 8 7 3 9 0 6 1 2
	6 0 8 4 5 7 2 9 3 1	6 0 4 1 5 7 3 9 2 8	3 0 7 1 2 6 8 4 9 5
	8 3 5 9 1 4 7 2 0 6	1 8 6 9 3 2 5 4 0 7	7 6 1 9 5 4 2 8 0 3
	4 5 1 6 3 2 9 8 7 0	5 2 7 8 1 3 9 0 4 6	4 5 3 8 6 2 9 0 7 1
	1 9 4 7 2 0 8 3 6 5	4 9 1 2 6 0 8 3 7 5	6 8 4 2 9 0 5 1 3 7
	5 8 9 2 0 3 4 6 1 7	3 4 9 5 2 8 7 1 6 0	8 7 9 5 1 3 4 2 6 0
	3 4 7 1 8 6 5 0 9 2	2 3 5 7 0 6 4 8 9 1	1 2 6 4 0 7 3 9 5 8
LBF:	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9
	9 7 0 6 5 8 4 3 2 1	9 7 0 6 5 8 4 3 2 1	9 7 0 6 5 8 4 3 2 1
	8 0 6 7 9 4 5 1 3 2	8 0 6 7 9 4 5 1 3 2	8 0 6 7 9 4 5 1 3 2
	7 8 5 4 0 9 3 2 1 6	7 8 5 4 0 9 3 2 1 6	7 8 5 4 0 9 3 2 1 6
	6 5 8 1 2 0 9 4 7 3	6 5 8 1 2 0 9 4 7 3	6 5 8 1 2 0 9 4 7 3
	5 9 1 0 8 3 2 6 4 7	5 9 1 0 8 3 2 6 4 7	5 9 1 0 8 3 2 6 4 7
	2 4 3 5 6 7 1 9 0 8	2 4 3 5 6 7 1 9 0 8	2 4 3 5 6 7 1 9 0 8
	1 2 7 9 3 6 0 8 5 4	1 2 7 9 3 6 0 8 5 4	3 2 7 9 1 6 0 8 5 4

3 6 4 8 1 2 7 5 9 0	3 6 4 2 7 1 8 5 9 0	1 6 4 8 3 2 7 5 9 0
4 3 9 2 7 1 8 0 6 5	4 3 9 8 1 2 7 0 6 5	4 3 9 2 7 1 8 0 6 5

Полученные результаты позволяют сделать ряд выводов. Прежде всего необходимо отметить, что поставленная задача в достаточной степени удобна для метода полного перебора с ограничениями (LBF), что подтверждается большим числом найденных решений. В то же время на ее примере можно сделать вывод о целесообразности ее сведения к оптимизационной задаче и последующего применения взвешивающих эвристик совместно со стохастическими методами (RS, WRS, AC), что увеличивает найденное число решений при использовании эвристических методов, базирующихся на последовательном выборе элементов формируемого квадрата. Как уже было отмечено выше, возможно совместное использование эвристических методов для поиска перспективных областей пространства поиска и метода полного перебора с ограничениями для уточнения решений в указанной области. Полученные выводы и рекомендации могут быть полезны при решении более сложных комбинаторных задач, например, связанных с поиском пар или троек попарно ортогональных ДЛК [18].

Авторы статьи выражают благодарность Григору В.В. и Носову В.А. за ряд ценных замечаний, полученных в ходе подготовки рукописи к изданию.

Работа выполнена в рамках выполнения государственного задания для ЮЗГУ на 2014–2017 гг., номер НИР 2246, а также в рамках научной школы НШ-2357.2014.8. Работа получила частичную поддержку РФФИ (гранты 14-07-00403-А и 15-07-07891-А) и Совета по грантам для молодых ученых при Президенте РФ (стипендия СП-1184.2015.5, грант ВНШ-5007.2014.9).

Список литературы

1. McKay B.D., Rogoyski E. Latin Squares of Order 10 // *The Electronic Journal of Combinatorics*. Vol. 2. 1995.
2. Ватутин Э.И., Колясников Д.В., Мартынов И.А., Титов В.С. Метод случайного перебора в задаче построения разбиений граф-схем параллельных алгоритмов // *Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов*. Барнаул: Барнаул, 2014. С. 115–125.
3. Ватутин Э.И., Колясников Д.В., Титов В.С. Анализ результатов применения метода случайного перебора в задаче поиска разбиений граф-схем параллельных алгоритмов // *Известия Южного федерального университета. Технические науки*. 2014. № 12 (161). С. 102–110.
4. Ватутин Э.И., Дремов Е.Н., Мартынов И.А., Титов В.С. Метод взвешенного случайного перебора для решения задач дискретной комбинаторной оптимизации // *Известия ВолГТУ. Серия: Электроника, измерительная техника, радиотехника и связь*. № 10 (137). Вып. 9. 2014. С. 59–64.
5. Dorigo M. *Optimization, Learning and Natural Algorithms* // PhD thesis. Politecnico di Milano, Italie, 1992.

6. Ватутин Э.И., Титов В.С. Анализ результатов применения алгоритма муравьиной колонии в задаче поиска пути в графе при наличии ограничений // Известия Южного федерального университета. Технические науки. 2014. № 12 (161). С. 111–120.
7. Ватутин Э.И., Титов В.С. Об одном подходе к использованию алгоритма муравьиной колонии при решении задач дискретной комбинаторной оптимизации // Интеллект – 2015. Тула, 2015. Принята к опубликованию
8. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by Simulated Annealing // Science. Vol. 220 No. 4598. 1983. pp. 671–680. DOI: 10.1126/science.220.4598.671
9. Ватутин Э.И., Титов В.С. Параметрическая оптимизация алгоритма имитации отжига на примере решения задачи поиска кратчайшего пути в графе // Вестник ЧГУ. Череповец, 2015. Принята к опубликованию
10. Ватутин Э.И., Мартынов И.А., Титов В.С. Способ обхода тупиков при решении задач дискретной оптимизации с ограничениями // Перспективные информационные технологии (ПИТ-2014). Самара: изд-во Самарского научного центра РАН. С. 313–317.
11. Colbourn C.J., Dinitz J.H. Handbook of Combinatorial Designs, 2nd ed. (Discrete Mathematics and Its Applications). Chapman & Hall/CRC. 2006.
12. Посыпкин М.А., Сигал И.Х. Комбинированный параллельный алгоритм решения задачи о ранце // Известия Российской академии наук. Теория и системы управления. 2008. № 4. С. 50–58.
13. Колпаков Р.М., Посыпкин М.А. Верхняя и нижняя оценки трудоемкости метода ветвей и границ для задачи о ранце // Дискретная математика. 2010. Т. 22. № 1. С. 58–73.
14. Колпаков Р.М., Посыпкин М.А. Об оценках вычислительной сложности варианта параллельной реализации метода ветвей и границ для задачи о ранце // Известия Российской академии наук. Теория и системы управления. 2011. № 5. С. 74–82.
15. DIMACS Graphs: Benchmark Instances and Best Upper Bounds // Электронный ресурс: <http://www.info.univ-angers.fr/pub/porumbel/graphs/>
16. Prestwich S.D. CNF Encodings // Handbook of Satisfiability. 2009. pp.75–97.
17. Ватутин Э.И. Эвристический подход к распознаванию изоморфизма графов // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание – 2015). Курск, 2015. С. 80–83.
18. Заикин О.С., Кочемазов С.Е. Поиск пар ортогональных диагональных латинских квадратов порядка 10 в проекте добровольных распределенных вычислений SAT@home // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2015. Т. 4. № 3. С. 95–108.