УДК 004.7 : 681.324

С. Ю. СКОБЛИКОВ, аспирант, НТУ «ХПИ»

ПРИМЕНЕНИЕ АЛГОРИТМОВ РАСТРОВОЙ ГРАФИКИ ДЛЯ СОПОСТАВЛЕНИЯ ЭЛЕМЕНТОВ ЧИСЛОВЫХ МНОЖЕСТВ

В работе предложен метод сопоставления элементов множеств с использованием разностной матрицы, показано, что при некоторых условиях вычисления всей разностной матрицы не является необходимым, зато достаточно построить растровую линию, аппроксимирующей минимума разностной матрицы. Предложено вычислительно-эффективный алгоритм сопоставления множеств действительных чисел на базе алгоритма Брезенхема для окружности. Показано работоспособность и адекватность предложенного метода.

Ключевые слова: сравнение множеств, растровая графика, оптимизация вычислений.

Вступление

Задача сравнения числовых множеств возникает при анализе работы компьютерных алгоритмов, в частности, алгоритмов оценки параметров.

Представим, что есть некоторый набор медленно изменяющихся параметров (в моем случае это были направления прихода радиосигналов):

$$\mathrm{P} = \left[p_1, p_2, ..., p_N \right] \in \mathbb{R}^N . \qquad (1)$$

Параметры, в свою очередь, влияют на некий физический процесс (сответственно, электромагнитная волна):

$$s = f(\mathbf{P}). \qquad (2)$$

Сами значения параметров неизвестны, равно как и их количество N. Можно только зафиксировать суперпозицию процессов (2), зависящих от параметров (1), при помощи измерительного устройства:

$$s_{\text{ИУ}} = \sum_{i=1}^{N} s\left(p_i\right).$$

Далее исследуемый алгоритм должен вычислить значения исходных параметров:

$$s_{\text{ИУ}} \mapsto \hat{\mathbf{P}} \in \mathbb{R}^M .$$

Глобальная задача состоит в определении качества работы программы-оценщика. Для этого необходимо сравнивать между собой результаты работы программы при различный настройках. То есть, сравнивать множества вещественных чисел. Результат такого сравнения позволит находить так называемые «артефакты» – значения параметров, не имеющие физического источника, но добавленные программой для минимизации ошибки вычисления.

При сравнении результатов важно помнить, что оценки $\hat{\mathbf{P}}$ имеют некоторую погрешность, что делает маловероятным точное совпадение элементов разных множеств. Поэтому элементы могут считаться совпадающими даже если они не строго равны, но находятся на достаточно близком расстоянии (предел которого заранее неизвестен). Простыми словами, требуется умение определять, какие элементы из первого множества наиболее «похожи» на элементы из второго, и наоборот.

К сожалению, поиск по литературным источникам дал лишь алгоритмы сравнения множеств для случая, когда элементы считаются совпадающими лишь при строгом равенстве [1,2]:

$$a_i = b_j \Rightarrow a_i \rightleftharpoons b_j \,.$$

Постановка проблемы и нотация

Имеются два множества вещественных чисел, представленных в виде векторов:

$$\begin{aligned} \mathbf{a} &= \left[a_1, a_2, ..., a_M\right], \\ \mathbf{b} &= \left[b_1, b_2, ..., b_N\right]. \end{aligned} \tag{3}$$

Необходимо найти как можно больше пар взаимно ближайших элементов:

$$a_i \rightleftharpoons b_j \,, \tag{4}$$

то есть элементов, для которых выполняются условия:

– a_i – ближайший к b_j элемент вектора **a**;

– b_j ближайший к a_i элемент вектора **b**,

чтобы в конечном итоге построить таблицу соответствия элементов:

$$\begin{gathered} \mathbf{C} = f\left(\mathbf{a}, \mathbf{b}\right) \\ \left(\mathbb{R}^M, \mathbb{R}^N\right) \mapsto \mathbb{R}^{K\times 3} \\ K < \min\left(M, N\right) \end{gathered} \tag{5}$$

где каждая строка матрицы C описывает соотношение пары элементов (4):

$$c_k = (i \;\; j \;\; |a_i - b_j|). \tag{6}$$

Соотношение чисел M, N, K также способно показать соотношение между множествами:

$$\begin{aligned} M = 0 \quad &\Rightarrow \quad \mathbf{a} = \emptyset \\ N = 0 \quad &\Rightarrow \quad \mathbf{b} = \emptyset \\ K = 0 \quad &\Rightarrow \quad (\mathbf{a} \cap \mathbf{b}) = \emptyset \\ M = N = K \quad &\Rightarrow \quad \mathbf{a} = \mathbf{b} \\ M = K < N \quad &\Rightarrow \quad \mathbf{a} \subset \mathbf{b} \\ N = K < M \quad &\Rightarrow \quad \mathbf{a} \supset \mathbf{b} \\ 0 < K < \min(M, N) \quad &\Rightarrow \quad (\mathbf{a} \cap \mathbf{b}) \neq \emptyset \end{aligned}$$

Метод сопоставления числовых множеств

Для начала, построим разностную матрицу:

$$\mathbf{D} \in \mathbb{R}^{M\times N} : \quad d_{ij} = \left|a_i - b_j\right| . \tag{7}$$

Далее, для каждой строки $\mathbf{d}_i$ находится индекс минимального элемента:

$$\mathbf{g} = (g_1, g_2, ..., g_M)^T$$
$$g_i = \operatorname{index}(\min(\mathbf{d_i})), \tag{8}$$

то есть каждый элемент множества a находит для себя ближайший элемент множества b.

Наконец, нужно убедиться, что на любой элемент множества b ссылаются не более одного раза. Для этого множество a разбивается на подмножества $a(b_j)$ элементов, указывающих на один и тот же элемент множества b. Подмножеству $a(b_j)$ будут принадлежать все элементы $a_i \in \mathbf{a}$, для которых $g_i = j$

$$\mathbf{a} = \bigcap_{j=1}^{N} a(b_j); \tag{9}$$

$$\forall i, g_i = j: \quad a_i \in a(b_j). \tag{10}$$

Затем если в подмножестве содержится больше одного элемента, то необходимо исключить все, кроме ближайшего к b_j:

$$a(b_j) = \begin{cases} a_i|_{d_{ij}=min} & \text{для } |a(b_j)| > 1 \\ a(b_j) & \text{иначе.} \end{cases} \tag{11}$$

Теперь можно составить таблицу соответствия (5).

Однако предложенный алгоритм имеет высокую вычислительную сложность $O(M*N)$ в том виде, в котором он изложен в данном разделе. Далее будет показано, как можно значительно ее снизить.

Оптимизация алгоритма по скорости выполнения

Ускорение возможно в случае, если множества (3) представлены в виде сортированных векторов:

$$a_1 < a_2 < \cdots < a_M$$
$$b_1 < b_2 < \cdots < b_N \tag{12}$$

На рис. 1 приведен общий вид разностной матрицы сортированных векторов. Как видно из рисунка, минимальные элементы матрицы D будут выстраиваться вдоль главной диагонали. Другими словами, если в i-й строке минимум находится по индексу j, то в $(i+1)$-й строке он не может лежать левее этого индекса. Следовательно «ложбина минимумов» на матрице будет пролегать с северо-запада на юго-восток, и нет необходимости рассчитывать все элементы разностной матрицы.

Любую матрицу можно рассматривать как дискретное представление функции от двух аргументов

$$\mathbf{D} = f(\mathbf{a},\mathbf{b}). \tag{13}$$

Если бы функция была непрерывной, выражение

$$f(\mathbf{a},\mathbf{b}) = 0 \tag{14}$$

соответствовало бы линии – функции одной переменной $x = f(y)$. В нашем случае разностной функции, линия будет непрерывной, причем

$$0 < \frac{dx}{dy} < \infty \tag{15}$$

так что всегда будет существовать и обратная функция $y = f^{-1}(x)$. Тот факт, что двумерная функция является непрерывной, позволил бы однозначно сопоставлять элементы сравниваемых множеств.

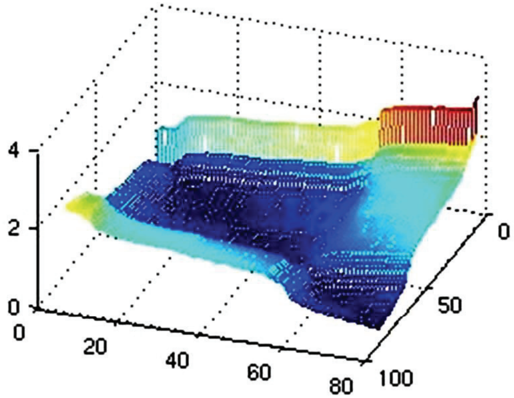

Рисунок 1 – Разностная матрица для упорядоченных векторов

К сожалению, разностная матрица содержит конечное количество элементов. Поэтому задача сопоставления двух множеств раскладывается на два этапа. Первым этапом является отрисовка на растровом поле линии, аппроксимирующей выражение (14). Координаты точек, лажещих на линии будут соответствовать элементам разностной матрицы, описывающим пары близко расположенных элементов. Однако не все точки, лежащие на этой линии будут соответствовать искомым парам элементов. Поэтому после отрисовки линии нужно будет убедиться, что в любом столбце или строке растрового поля присутствует не более, чем одна точка. Если точек было несколько, то остаться должна лишь та, которая аппроксимирует выражение (14) наилучшим образом.

Отрисовка линии методом Брезенхема

Алгоритм Брезенхема [4] для окружностей отрисовывает на растровом поле первый квадрант окружности.

В исходном состоянии курсор находится в левом верхнем углу поля. За исключением адаптации под целочисленные вычисления, идея алгоритма состоит в том, что каждая следующая точка может находиться либо правее, либо правее и ниже, либо ниже текущей. Из трех альтернатив выбирается та, которая имеет минимальную ошибку аппроксимации.

Если рассмотреть данный алгоритм в контексте вышеизложенного, то можно считать, что выражение

$$|x^2 + y^2 - R^2|, \quad x, y < R \in \mathbb{N} \tag{16}$$

соответствует разностной матрице для множеств

$$\mathbf{x} = \{0^2, 1^2, \ldots, (N-1)^2\}$$
$$\mathbf{y} = \{R^2-(N-1)^2, R^2-(N-2)^2, \ldots, R^2-0^2\} \tag{17}$$

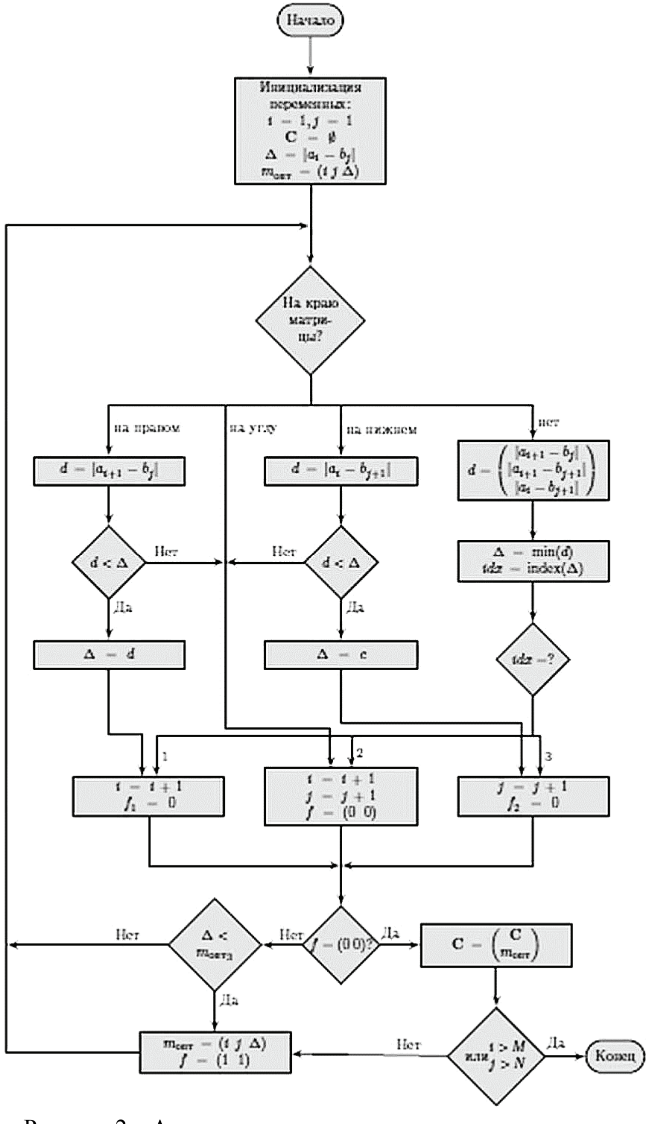

Рисунок 2 – Алгоритм сопоставления элементов множеств

Разностная матрица (7) обладает теми же свойствами, что и выражение (16): как видно из рис. 3, минимум пролегает вдоль главной диагонали. Следовательно, алгоритм Брезенхема пригоден для аппроксимации разностной матрицы для любой пары упорядоченных множеств.

Поиск минимумов в строках и столбцах

Выполнение условия (11) возможно обеспечить непрсредственно в процессе отрисовки линии. Для этого необходимо отслеживать наилучшего претендента $m_{опт}$ на включение в список (5). До тех пор, пока текущая отрисовываемая точка $m_{тек}$ находится на той же линии, что и $m_{опт}$, нужно при необходимости обновлять $m_{опт}$:

$$m_{опт} = \begin{cases} m_{тек} & \text{если} \quad \mathbf{D}(m_{опт}) > \mathbf{D}(m_{тек}) \\ m_{опт} & \text{иначе} \end{cases} \tag{18}$$

Как только $m_{тек}$ выйдет за пределы линии, содержащей $m_{опт}$, последнюю можно добавить в матрицу соответствий \eqref{eq:C}, после чего $m_{тек}$ будет считаться новой $m_{опт}$.

Полный алгоритм приведен на рис. 2. Обратите внимание, что особой обработки требуют случаи, когда текущая точка находится на нижнем или правом краю матрицы. В таких случаях вместо выбора между тремя альтернативными направлениями стоит лишь вопрос, нужно ли продолжать движение по последнему столбцу/строке дальше, или уже найдена точка с минимальной метрикой и выполнение алгоритма можно завершить. Если принимается решение о завершении работы алгоритма, то смещение по диагонали в каждом из случаев приведет к добавлению последней найденной оптимальной точки в таблицу соответствия и выходу из цикла.

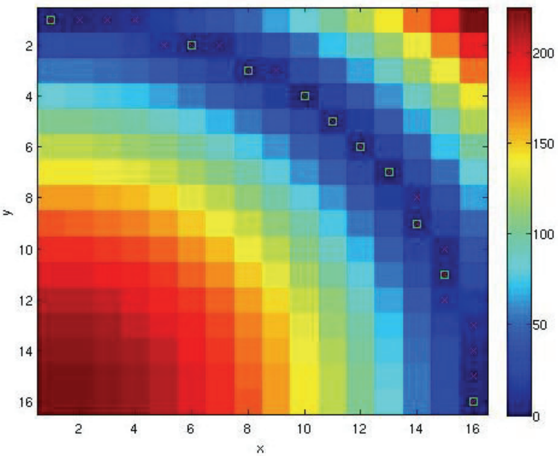

Рисунок 3 – Разностная матрица для множеств (17)

Проверка работоспособности предложенного метода

В качестве первого примера рассмотрим «родной» сценарий работы алгоритма Брезенхема. Сопоставляемые множества заданы выражением (17), при условии, что $R = 15$, $M = N = 16$.

На рис. 3, изображающем разностную матрицу, отмечен путь, проложенный алгоритмом, а также ячейки матрицы, соответствующие взаимно ближайшим парам элементов. Сами множества, а также связи, найденные алгоритмом показаны на рис. 4

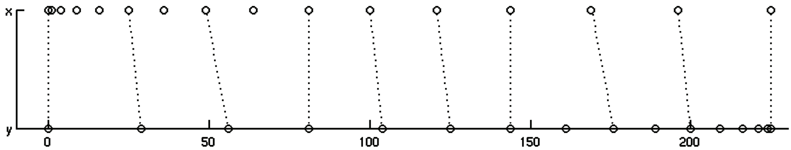

Рисунок 4 – Множества (17) и связи, найденные алгоритмом

На рис. 5 приведен результат работы алгоритма на примере прикладной задачи описанной во введении к данной статье. Обратите внимание, что алгоритм улавливает малые изменения в расстоянии между парами элементов, однако для одиноко стоящих элементов, хоть они находятся достаточно далеко друг от друга, алгоритм понимает, что они связаны, что является желаемым поведением.

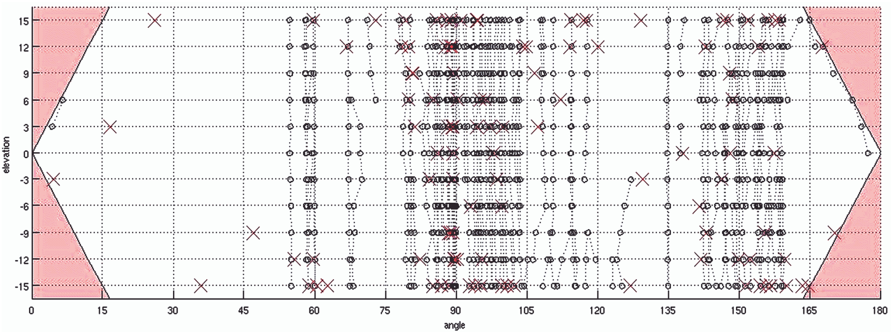

Рисунок 5 – Использование алгоритма в реальной прикладной задаче

Выводы. Разработан метод нахождения пар взаимно ближайших элементов числовых множеств. Произведена оптимизация разработанного метода по вычислительным затратам. Показано, что для случая сортированных векторов нет необходимости расчета всей разностной матрицы, достаточно лишь построить растровую апроксимацию нуля разностной матрицы. Обоснован выбор алгоритма Брезенхема для построения растровой линии. Показано, что вычислительная сложность задачи уменьшается с $O(M*N)$ до $O(M + N)$. Проверена работоспособность разработанного метода как на искусственном примере, так и на прикладной задаче. Показана адекватность и устойчивость алгоритма.

Список литературы: **1.** *Furtado A. L.*. Generalized set comparison / *A. L. Furtado* // Newsletter ACM SIGPLAN Notices. – 1984. – Vol. 19, no. 9. **2.** Efficient algorithm to compare similarity between sets of numbers? – http://stackoverflow.com/questions/1053821/efficientalgorithm-to-compare-similarity-between-sets-of-numbers. **3.** *Кормен Т.* Алгоритмы. Построение и анализ / *Т. Кормен.* – М.: МЦНМО, 2001. – 960 с. **4.** *Роджерс Д.* Алгоритмические основы компьютерной графики : Пер. с англ. / *Д. Роджерс.* – М.: Мир, 1989. – 512с.

Поступила в редколлегию 19.09.2012

УДК 004.7+681.324

Применение алгоритмов растровой графики для сопоставления элементов числовых множеств / С. Ю. Скобликов // Вісник НТУ «ХПІ». Серія: Техніка та електрофізика високих напруг. – Х.: НТУ «ХПІ», 2012. – № 52 (958). – С. 161-168. – Бібліогр.: 4 назв.

У роботі запропоновано метод зіставлення елементів множин з використанням різницевої матриці, показано, що за деяких умов обчислення всієї різницевої матриці не є необхідним, натомість достатньо побудувати растрову лінію, що апроксимує мінімуми різницевої матриці. Запропоновано обчислювально-ефективний алгоритм зіставлення множин дійсних чисел на базі алгоритму Брезенхема для окружності. Показано працездатність та адекватність запропонованого метода. Іл.: 5. Бібліогр.: 4 назви.

Ключові слова: порівняння множин, растрова графіка, оптимізація обчислень.

A method for matching between elements of sets of reals is proposed, which involves the computation of a difference matrix. It has been shown that under certain circumstances the computation of the whole difference matrix is not necessary and the aim can be achieved by plotting on raster a line that approximates the minimum of the matrix. A computationally effective algorithm for inter-sets matching which is based on the Circle Bresenham Algorithm is proposed. The proposed method is shown to be operable and adequate. Figs: 5. Refs: 4 titles.

Key words: inter-set matching, raster graphics, algorithm optimization.