

Возникает вопрос об исследовании свойств оценки (7). Учитывая представление (5), путем элементарных преобразований оценку (7) можно переписать в виде: [1]

$$\hat{f}(\lambda) = \tilde{P}_{j_0}(f) + \sum_{j=j_0}^{J-1} \tilde{Q}_j(f) = \tilde{P}_{j_0}(f) + (\tilde{P}_{j_0+1}(f) - \tilde{P}_{j_0}(f)) + (\tilde{P}_{j_0}(f) - \tilde{P}_{j_0-1}(f)) + \dots + (\tilde{P}_{J-1}(f) - \tilde{P}_{J-2}(f)) + \tilde{P}_J(f) - \tilde{P}_{J-1}(f).$$

Используя представление (6), окончательно получим:

$$\hat{f}(\lambda) = \tilde{P}_J(f) = \sum_{k=1}^{2^J} \hat{\alpha}_{J,k} \varphi_{J,k}(\lambda),$$

где $\hat{\alpha}_{J,k}$ задается соотношением (8), $\tilde{\psi}_{J,k}(\lambda)$ соотношением (4), $J \leq \log_2 CT^{1-\rho}$, где $0 < \rho < 1$, $0 < C < \infty$, произвольные, но фиксированные константы, $k = \overline{1, 2^J}$. Таким образом, в качестве оценки спектральной плотности $f(\lambda)$, $\lambda \in \Pi$ рассматривается ее проекция на пространство \tilde{V}_J .

Литература

1. Добеши И. Десять лекций по вейвлетам. – Ижевск: НИЦ “Регулярная и хаотическая динамика”, 2001. – 464с.
2. Neumann M.H. Spectral density estimation via nonlinear wavelet methods for stationary non-Gaussian time series. // J. Time Ser. Anal. 17 (1996), № 6, P. 601-633.
3. Н.В. Семенчук. Исследование моментов оценок вейвлет коэффициентов разложения спектральной плотности стационарного случайного процесса. / Материалы международной конференции DE&CAS'2005, г. Брест, 5 – 8 окт. 2005 г. Ч.2. С. 55-59.

МАТРИЧНЫЙ МЕТОД РЕАЛИЗАЦИИ АЛГОРИТМА КРИСТАЛЛИЗАЦИИ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

Смачек С.Н., БГТУ, Брест

Задача коммивояжера (Traveling salesman problem) относится к ряду сложнейших задач комбинаторной оптимизации и имеет на сегодняшний день множество приложений. Решение этой задачи сложно, что не дает возможности даже при современном развитии средств вычислительной техники находить решения при большой размерности исходных данных за приемлемое время. Поэтому для решения этой задачи создано множество приближенных методов, которые для своего исполнения используют приемлемое количество ресурсов (времени, оперативной памяти и других). Одним из таких методов является метод кристаллизации. Имея сложность $O(n^2)$, этот алгоритм показал средние результаты для качества находимых решений при довольно высокой скорости работы, что делает его применимым для задач большой размерности (десятки тысяч узлов).

Предлагаемая матричная реализация алгоритма кристаллизации служит для более эффективного решения задачи коммивояжера на ЭВМ. Первоначально составляется матрица, элементы которой состоят из двух полей: расстояние (dij) и флаг (fij) принадлежности к маршруту (1 - дуга принадлежит маршруту, 0 - не принадлежит, * - запрет на включение данной дуги). По этой матрице строится начальный маршрут коммивояжера, например, методом ближайшего соседа. Для этого в строке выбирается элемент (i, j) с наименьшим расстоянием, отмечается принадлежность этой дуги маршруту,

а для элемента (j, i) ставится запрет на включение в маршрут (это делается для того, чтобы избежать замыканий контура на двух узлах). В столбце j на все элементы ставится флаг запрета, чтобы j -узел был инцидентен только двум дугам. Далее действие повторяется для j -той строки матрицы. Таким образом, в маршрут включаются все узлы (рис. 1).

На втором этапе подобно методу потенциалов в транспортной задаче первоначальный контур разбивается на несколько «оптимальных в себе» контуров. Для этого высчитывается оценка уменьшения общей длины контуров и проводится изменение маршрута, если она положительна.

Введем обозначения:

$R(i)$ - номер строки, в которой i -тый столбец имеет флаг включения дуги в маршрут ($f_{R(i)}=1$), $C(i)$ - номер столбца, в котором i -тая строка имеет флаг включения дуги в маршрут ($f_{iC(i)}=1$).

Тогда, если $d_{R(i) i} - d_{ki} > d_{R(i) C(k)} - d_{k C(k)}$, проводим перемещение флага в i -том столбце на строку k , а в $C(k)$ столбце на строку $R(i)$ (рис. 2). Проведя такие действия над всей матрице, мы получим множество отдельных замкнутых контуров, которые в матрице путем пересортировки вершин всегда можно представить квадратными областями, расположенными по главной диагонали (рис. 3).

Теперь мы располагаем множеством отдельных маршрутов, которые предстоит склеить в наиболее оптимальный. Для этого необходимо предварительно вычислить расстояния между контурами.

Расстоянием между двумя контурами будем называть минимальное увеличение суммы длин контуров, которое достигается при самом экономном объединении этих двух контуров.

Высчитав по исходной матрице расстояния между контурами, можно составить новую матрицу расстояний уже для контуров и повторить все шаги алгоритма уже над этой матрицей, что в конечном итоге приведет к решению задачи коммивояжера небольшой размерности, где можно применить трудоемкие точные алгоритмы.

Полученный на конечном этапе маршрут разрывается по самому длинному ребру, «разворачивается» до размерности исходной задачи - все контуры имплементируются составляющими их вершинами - получается маршрут, приближенный к оптимальному.

После получения на первом этапе первичных контуров альтернативным путем является построение минимального остовного дерева, узлами которого будут являться найденные контуры, и дальнейшее склеивание производить по ребрам этого дерева. Еще одним более трудоемким вариантом может быть склеивание наиболее близкорасположенных контуров с последующим пересчетом расстояний между контурами (последовательность шагов приведена на рис. 4-7).

Результат можно улучшить, если над итоговым маршрутом провести процедуры эвристической оптимизации (рис. 8).

	1	2	3	4	5	6
1	∞	3	4	8	9	11
2	0	1	0	0	0	*
3	3	∞	3	4	8	9
4	*	0	1	0	0	0
5	4	3	∞	3	4	8
6	0	*	0	1	0	0
1	8	4	3	∞	3	4
2	0	0	*	0	1	0
3	9	8	4	3	∞	3
4	0	0	0	*	0	1
5	11	9	8	4	3	∞
6	1	0	0	0	*	0

Рис. 1. Исходная матрица расстояний и флагов

	1	2	3	4	5	6
1	∞	3	4	8	9	11
2	0	1	0	0	0	*
3	3	∞	3	4	8	9
4	*	0	1	0	0	0
5	4	3	∞	3	4	8
6	0	*	0	1	0	0
1	8	4	3	∞	3	4
2	0	0	*	0	1	0
3	9	8	4	3	∞	3
4	0	0	0	*	0	1
5	11	9	8	4	3	∞
6	1	0	0	0	*	0

Рис. 2. Процедура «кристаллизации»

	1	2	3	4	5	6
1	∞	3	4	8	9	11
2	0	1	*	0	0	0
3	3	∞	3	4	8	9
4	*	0	1	0	0	0
5	4	3	∞	3	4	8
6	1	*	0	0	0	0
1	8	4	3	∞	3	4
2	0	0	0	0	1	*
3	9	8	4	3	∞	3
4	0	0	0	*	0	1
5	11	9	8	4	3	∞
6	0	0	0	1	*	0

Рис. 3. Независимые контуры в матричном представлении

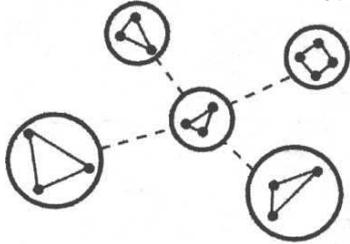


Рис. 4. Минимальное остовное дерево для первичных контуров

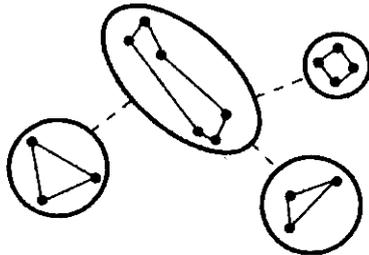


Рис. 5. Слияние двух ближайших контуров в один

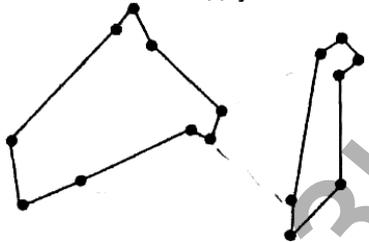


Рис. 6. Конечный этап образования единого маршрута - объединение двух контуров

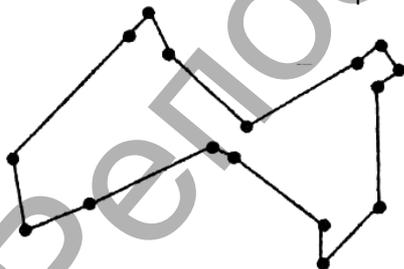


Рис. 7. Маршрут коммивояжера



Рис. 8. Эвристическое улучшение маршрута