



СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

P5-97-258

С.В.Горбунов¹, И.В.Кисель, Е.В.Конотопская,
Р.Т.Файзуллин¹

СРАВНЕНИЕ МЕТОДОВ
ГАРАНТИРОВАННОЙ ГЛАДКОСТИ
И ЭЛАСТИЧНОЙ СЕТИ
ДЛЯ ЗАДАЧИ КОММИВОЯЖЕРА НА ПЛОСКОСТИ

¹Омский государственный университет

1997

Введение

Задача коммивояжера — классический пример трудно-решаемой задачи из области комбинаторной оптимизации, имеющей дело с эффективными методами максимизации или минимизации функций многих переменных. В упрощенном виде она звучит так:

Даны положения N городов, необходимо найти кратчайший замкнутый маршрут, проходящий через каждый город, причем только один раз.

Алгоритмы нахождения решения можно условно разделить на две группы: точные и приближенные. В точных методах на каждом этапе поиска путь проходит точно через города, в то время как в приближенных методах эволюция развивается во всей плоскости городов¹.

Все известные точные методы определения оптимального пути требуют затрат компьютерного времени, экспоненциально возрастающего с ростом городов, так что на практике оптимальное решение можно пытаться найти только для задач с не более чем несколько сотен городов. Задача коммивояжера принадлежит к большому классу так называемых NP-полных² задач, т.е. не может быть решена за полиномиальное время, и к ней сводится любая другая задача из класса NP [1].

Существующие точные алгоритмы сводятся, в сущности, к организации полного перебора вариантов. Несмотря на то, что многие из них находят решение достаточно быстро, для каждого такого алгоритма существует задача, на которой приходится перебирать все $N!$ возможных путей. Среди алгоритмов этого типа наиболее распространен метод ветвей и границ в различных модификациях.

Во многих практических приложениях не требуется нахождение именно оптимального решения. Необходимо получить более или менее хороший результат за короткое время. Кроме того, иногда важен общий вид пути, а не его длина. Для этих целей применяются быстрые точные алгоритмы. Из быстрых алгоритмов наиболее известен алгоритм иди к ближайшему. Суть его такова: начинаем обход из произвольного города и на каждом шаге выбираем ближайший непройденный город. Для задачи с неравенством треугольника результат превышает точное решение не более чем в $\frac{1}{2}(\lceil \log_2 N \rceil + 1)$ раза [1], причем эта оценка достижима при любом числе точек. Для геометрической задачи алгоритм иди к ближайшему является асимптотически сходящимся, т.е. он сходится к оптимальному решению по вероятности с ростом числа точек. Известны точные алгоритмы с лучшим

¹Заметим, что результатом точного метода будет не обязательно наилучшее решение. В дальнейшем наилучшее решение будем называть оптимальным.

²Nondeterministic polynomial time complete.

качеством решения для общей задачи, например, двойной обход минимального остового дерева [1]. Его решение отличается от оптимального не более чем в два раза.

Исследования в этой области были направлены также на поиск решений, близких к оптимальному, за время, растущее как полином малой степени. В последние годы было предложено несколько новых методов для решения задачи коммивояжера (и оптимизационных задач в целом), такие как имитационный отжиг [2], генетические алгоритмы [3, 4] и нейронные сети [5, 6, 7, 8, 9].

Кроме универсальных алгоритмов решения, разработано также множество точных и приближенных алгоритмов, рассчитанных на конкретные типы практических задач. Например, были предложены модификации метода эластичной сети для обработки данных в физике высоких энергий [10, 11, 12, 13].

1. Алгоритм с гарантированной гладкостью решения

Пусть даны N точек плоскости $(x^C_i, y^C_i), i = \overline{1, N}$. Требуется найти замкнутый путь минимальной длины, проходящий через эти точки.

Пусть (i_1, \dots, i_N) — путь, последовательно проходящий через точки i_1, \dots, i_N . Его можно представить в виде последовательности координат этих точек $((x^C_{i_1}, y^C_{i_1}), \dots, (x^C_{i_N}, y^C_{i_N}))$. Таким образом, каждому возможному пути соответствует вектор в R^{2N} .

Длина произвольного пути в плоскости городов равна

$$L((x_1, y_1), \dots, (x_N, y_N)) = \sum_{l=1}^N \sqrt{(x_l - x_{l-1})^2 + (y_l - y_{l-1})^2}. \quad (1)$$

Ясно, что выбор пути можно представить как выбор одного из $N!$ векторов, соответствующих возможным путям, на котором функционал L достигает минимального значения.

Каким образом это сделать, чтобы избежать полного перебора? Можно выделить небольшую область в R^{2N} , содержащую пути, на которых L достигает близкое к минимуму значение. Затем найти точку минимума L в этой области, решая непрерывную задачу минимизации, и взять ближайший к ней путь, проходящий через города.

Для этого необходим критерий, позволяющий выделить близкие к оптимальному пути. Один из возможных подходов — найти свойства, присущие большинству оптимальных решений, и искать пути, наделенные этими свойствами.

Производные L по x_l, y_l в точке экстремума должны быть близки к 0.

$$-\frac{\partial L}{\partial x_l} = \frac{x_{l+1} - x_l}{\sqrt{(x_{l+1} - x_l)^2 + (y_{l+1} - y_l)^2}} - \frac{x_l - x_{l-1}}{\sqrt{(x_l - x_{l-1})^2 + (y_l - y_{l-1})^2}}. \quad (2)$$

Проведем через $(x_1, y_1), \dots, (x_N, y_N), (x_1, y_1)$ кривую $(X(t), Y(t))$, параметризованную длиной дуги. Тогда выражение (2) можно рассматривать как аппроксимацию второй производной функции $X(t)$ по t в точке (x_l, y_l) .

Таким образом, нам необходимо минимизировать вторые производные функций $X(t)$ и $Y(t)$ в узлах неравномерной сетки, определяемой точками $(x_1, y_1), \dots, (x_N, y_N)$. Т.к. для нашей задачи алгоритм иди к ближайшему асимптотически сходится, то с ростом числа точек оптимальный путь будет проходить через близкие друг к другу точки, следовательно, представляя искомый путь в виде непрерывной замкнутой кривой, мы получаем требование ее гладкости.

Алгоритм с гарантированной гладкостью решения непосредственно реализует полученный результат. Его идея состоит в следующем:

Строим заведомо гладкую кривую, как можно более приближенную к исходным точкам. Затем строим маршрут вдоль этой кривой, выбирая локально наилучший путь.

Кривая будет задаваться функциями $X(t), Y(t)$, представляемыми в виде частичных сумм соответствующих рядов Фурье.

Пусть

$$X(t) = \sum_{k=0}^m (a_k^X \cos(kt) + b_k^X \sin(kt)), \quad Y(t) = \sum_{k=0}^m (a_k^Y \cos(kt) + b_k^Y \sin(kt)), \quad (3)$$

где $t \in [0, 2\pi]$. Тогда

$$\frac{\partial^2 X}{\partial t^2} = - \sum_{k=0}^m (k^2 a_k^X \cos(kt) + k^2 b_k^X \sin(kt)). \quad (4)$$

Мы рассматриваем $X(t)$ и $Y(t)$ в функциональном пространстве с базисом $\sin(kt)$ и $\cos(kt)$. Норма второй производной

$$\left\| \frac{\partial^2 X}{\partial t^2} \right\|^2 = \sum_{k=0}^m k^2 (a_k^{X^2} + b_k^{X^2}). \quad (5)$$

Выражение для $Y(t)$ аналогично.

Получаем

$$\left\| \frac{\partial^2 X}{\partial t^2} \right\|^2 + \left\| \frac{\partial^2 Y}{\partial t^2} \right\|^2 = \sum_{k=1}^m k^2 (a_k^{X^2} + b_k^{X^2} + a_k^{Y^2} + b_k^{Y^2}). \quad (6)$$

Для минимизации (6) необходимо, чтобы коэффициенты как можно быстрее убывали с ростом k .

Будем строить кривую следующим образом:

1. На первом шаге полагаем все коэффициенты нулевыми.
2. В качестве a_0^X, a_0^Y возьмем координаты центра масс системы точек.
3. Рассчитываем $a_1^X, b_1^X, a_1^Y, b_1^Y$ так, чтобы среднеквадратичное расстояние от точек до кривой было минимальным, затем переходим к следующему коэффициенту.

Заметим, что на каждом шаге меняются лишь четыре коэффициента с соответствующим индексом, а все предыдущие остаются неизменными. При таком построении в рядах Фурье будут преобладать низкие частоты, что и требовалось.

Алгоритм останавливается, когда при добавлении следующего коэффициента длина волны становится меньше половины среднего расстояния между соседними точками. Это условие следует из предела Найквиста [14], в противном случае более высокие частоты будут неотличимы от низких.

Для локальной оптимизации используется алгоритм, напоминающий сортировку пузырьком:

Берем две соседние точки и переставляем их, если это приводит к уменьшению длины пути. Затем переходим к следующим, до тех пор, пока эти перестановки дают результат.

Еще раз заметим, что этот алгоритм является локальным и общий вид решения определяется именно построенной гладкой кривой.

2. Метод эластичной сети

В методе эластичной сети [7] путь представляется в виде набора "узлов" с эластичным соединением между ними, образуя замкнутый маршрут. Такая эластичная сеть одновременно и притягивается к городам, и сжимается под действием сил упругости. Тем самым алгоритмически реализуется сама формулировка задачи. Привлекательностью эластичной сети является главным образом ее простота и универсальность. Главным ее недостатком была низкая скорость, которая в основном обуславливалась большим числом узлов сети (например, в оригинальной работе предлагалось использовать узлов в 2.5 раза больше, чем городов).

В настоящей работе, полностью сохраняя идею метода, предлагается его модификация, которая приводит к значительному (примерно на 2 по-

рядка) увеличению скорости при сохранении качества решения. Это делает метод эластичной сети конкурентоспособным по отношению к хорошо разработанным классическим методам глобальной оптимизации.

Имеем набор N городов (точек) на плоскости. Каждой точке присваивается некоторая "масса" (вес). В полученном "гравитационном поле" размещается непрерывная кривая, состоящая из M узлов, которые также имеют "массы", и начинает двигаться под действием сил притяжения как городов, так и узлов.

Итак, кривая представляется в виде эластичной сети из M узлов с координатами (x_i^E, y_i^E) , $i = \overline{1, M}$ и первоначально задается в виде окружности.

Веса точек пропорциональны их удаленности от соседних. Это обеспечивает рассмотрение удаленных точек наравне с возможными скоплениями³, что, очевидно, соответствует задаче

$$w_i^C = \frac{\sum_{k=1}^N \|x_i^C - x_k^C, y_i^C - y_k^C\|}{\max_j \sum_{k=1}^N \|x_j^C - x_k^C, y_j^C - y_k^C\|}. \quad (7)$$

Воздействие города на сеть будем определять в каждый момент времени таким образом, чтобы по возможности притягивать только ближнюю часть сети. Вес города i по отношению к сети характеризуется величиной

$$v_i^C = \min_k \left\{ ((x_i^C - x_k^E)^2 + (y_i^C - y_k^E)^2)^4 \right\}, \quad (8)$$

а действие точки i на узел j величиной

$$c_{ij} = w_i^C \cdot v_i^C \cdot ((x_i^C - x_j^E)^2 + (y_i^C - y_j^E)^2)^{-4}. \quad (9)$$

И, наконец, полное смещение узла j по оси x

$$\Delta x_j^E = \alpha \cdot \frac{\sum_{i=1}^N c_{ij} \cdot (x_i^C - x_j^E)}{\sum_{i=1}^N c_{ij}} + \beta \cdot (x_{j+1}^E - 2x_j^E + x_{j-1}^E), \quad (10)$$

$\alpha = 0.5$ и $\beta = 0.1$ — константы, а выражение при β обеспечивает "эластичность" кривой, минимизируя первую производную вдоль нее.

Алгоритм устроен следующим образом:

³Это аналогично процедуре кластеризации экспериментальных точек при обработке данных в физике высоких энергий.

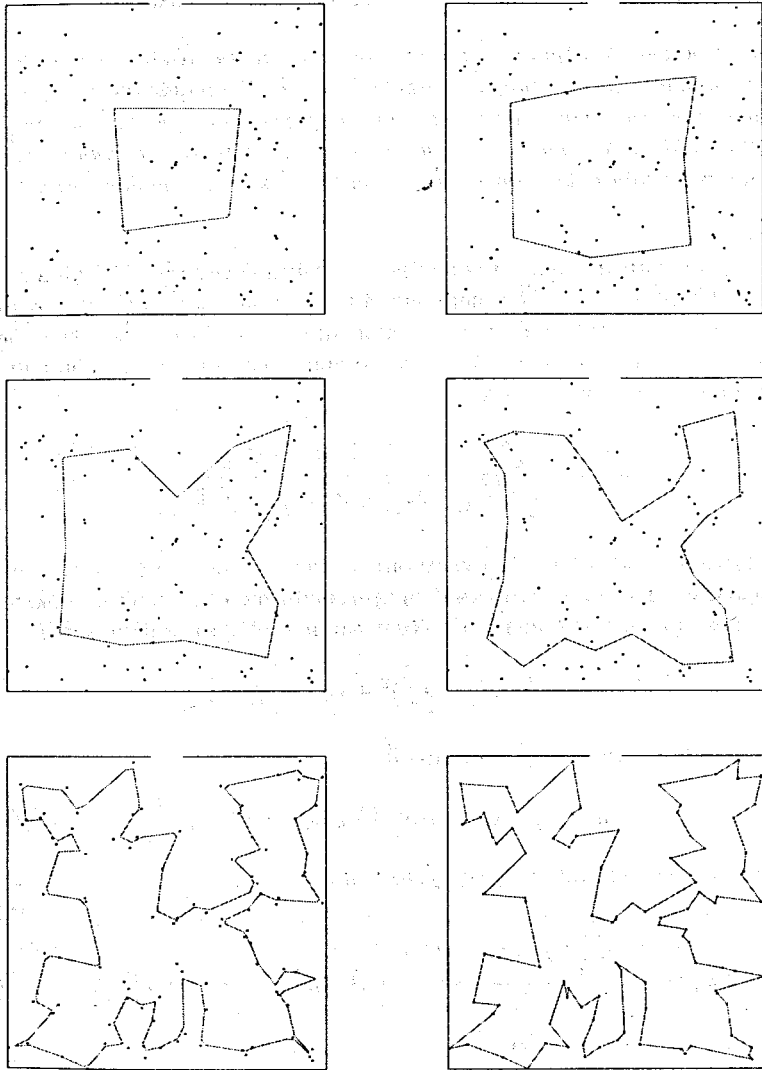


Рис. 1. Пример итераций метода эластичной сети для 100 городов

1. Берется малое число узлов, например, равное 4.
2. Производится несколько итераций, до тех пор, пока изменения кривой не станут достаточно малы.
3. Города проецируются на кривую, и проверяется простейший критерий выхода: расстояние между точками кривой меньше расстояния между городами вдоль маршрута.
4. Увеличивается число узлов кривой. Просчитанные узлы соединяются отрезками, на которых и располагаются новые узлы.
5. Производится следующая итерация с пункта 2.

Заметим, что алгоритм стартует с гладкой кривой, а точки гладко воздействуют на нее. Кроме того, постепенное увеличение числа узлов аналогично пошаговому добавлению частот. Т.е. можно утверждать, что решение получается достаточно гладким. Поскольку результирующая кривая может не проходить непосредственно через точки, то необходима локальная оптимизация, например при помощи алгоритма, описанного в методе “гарантированной гладкости”.

3. Результаты тестирования

Описанные выше алгоритмы были протестированы при различных количествах городов (точек). Точки равномерно распределялись в квадрате, для каждого числа точек производилось 1000 опытов. Расчеты велись на компьютере РС 386. Результаты анализа приведены в таблице.

Под относительной разностью норм коэффициентов понимается

$$\frac{|C^1_k - C^2_k|}{C^1_k + C^2_k}, \quad (11)$$

где

$$C^1_k = \sqrt{a_k^2 X_k^2 + b_k^2 X_k^2 + a_k^2 Y_k^2 + b_k^2 Y_k^2}, \quad (12)$$

a и b — коэффициенты кривой алгоритма с гарантированной гладкостью, а C^2_k — аналогичное выражение для кривой с таким же числом коэффициентов Фурье, проведенной через маршрут, полученный в эластичной сети в первом случае, и в идею к ближайшему во втором. Выражение (11) характеризует степень гладкости второй кривой по отношению к первой.

Т.к. алгоритм идею к ближайшему сходится к решению по вероятности, на конкретных задачах он может давать плохой результат. Наши же алгоритмы дают стабильно лучшее решение. Это отражено в статистике в строках “лучший/худший результат”.

Мы сравнили алгоритмы на задаче со 101 городом⁴, для которой известно точное решение. Результаты в процентах от оптимального:

Иди к ближайшему	-	128.4%
Гарантированная гладкость	-	114.7%
Эластичная сеть	-	110.2%

Заключение

Интерес к исследованию задачи коммивояжера с помощью методов гарантированной гладкости и эластичной сети объясняется активным использованием, в частности, метода эластичной сети для обработки данных в физике высоких энергий.

Привлекательностью эластичной сети является главным образом ее простота и универсальность. Главным ее недостатком была низкая скорость, которая в основном обуславливалась большим числом узлов сети (например, в оригинальной работе предлагалось использовать узлов в 2.5 раза больше, чем городов).

В настоящей работе исследованне свойств гладкости путем сравнения с методом гарантированной гладкости позволило предложить модификацию метода эластичной сети, которая приводит к значительному (примерно на 2 порядка) увеличению скорости при сохранении качества решения. Это делает метод эластичной сети конкурентоспособным по отношению к хорошо разработанным классическим методам глобальной оптимизации. Так, например, скорость метода эластичной сети для задачи с 1000 городами повышена до 2 мин. 35 с. на PC-386 и до 30 с. на компьютере типа Sun.

Список литературы

- [1] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, San Francisco, 1987.
- [2] P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*. D. Reidel, Dordrecht, 1987.
- [3] R.M. Brady, *Optimization strategies gleaned from biological evolution*. *Nature* 317 (1985) 804.
- [4] H. Mühlhain, m. Gorges-Schleuter and O. Krämer, *Evolution algorithms in combinatorial optimization*. *Parallel Comput.* 7, 65 (1988) 85.

⁴Пример eil101 из TSP библиотеки университета г. Гайдельберг.

Длина пути, % от иди к ближ.		Число точек					
		50	100	200	400	600	1000
без локальной оптимизации	Гар. гладкость	90	94	100	113	121	142
	Эласт. сеть	89	90	90	92	92	90
с локальной оптимизацией	Гар. гладкость	88	90	94	101	105	118
	Эласт. сеть	88	88	88	90	90	89
лучший результат	Гар. гладкость	68	73	80	92	95	114
	Эласт. сеть	70	72	77	85	81	86
худший результат	Гар. гладкость	112	107	110	113	115	125
	Эласт. сеть	118	111	106	97	96	92
Время работы (мин. сек.)	Иди к ближ.	.1	.1	.1	.1	.1	.2
	Гар. гладкость	.1	.1	.7	.37	2.12	6.53
	Эласт. сеть	.1	.1	.4	.17	.50	2.35
Относ. разность норм для эласт. сети	1 коэфф.	0.01	0.01	0.01	0.21		
	2 коэфф.	0.28	0.45	0.65	7.61		
	3 коэфф.	0.18	0.27	0.34	3.88		
	4 коэфф.	0.31	0.50	0.73	8.41		
	5 коэфф.	0.29	0.42	0.53	6.09		
Относ. разность норм для иди к ближайшему	1 коэфф.	0.26	0.46	0.70	8.0		
	2 коэфф.	1.01	1.63	2.31	26.4		
	3 коэфф.	0.42	0.70	0.96	10.9		
	4 коэфф.	0.46	0.79	1.24	14.3		
	5 коэфф.	0.39	0.57	0.75	8.4		

- [5] J.J. Hopfield and D.W. Tank, "Neural" computation of decisions in optimization problems. *Biol. Cybern.* 52 (1985) 141.
- [6] G.V. Wilson and G.S. Pawley, *On the stability of the travelling salesman problem algorithm of Hopfield and Tank.* *Biol. Cybern.* 58 (1988) 63.
- [7] R. Durbin and D. Willshaw, *An analogue approach to the travelling salesman problem using an elastic net method.* *Nature* 326 (1987) 689.
- [8] B. Angéniol, G. de la Croix Vaubois and J-Y. le Texier, *Self-organizing feature maps and the travelling salesman problem.* *Neural Networks* 1 (1988) 289.
- [9] C. Peterson and B. Söderberg, *A new method for mapping optimization problems onto neural networks.* *Int. J. Neural Syst.* 1 (1989) 3.
- [10] I. Kisel, V. Neskromnyi and G. Ososkov, *Applications of Neural Networks in Experimental Physics.* *Phys. Part. Nucl.* 24 (6), November-December 1993, p. 657.
- [11] M. Gyulassy and M. Harlander, *Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition.* *Comp. Phys. Commun.* 66 (1991) 31.
- [12] M. Ohlsson, C. Peterson and A.L. Yuille, *Track Finding with Deformable Templates — the Elastic Arms Approach.* *Comput. Phys. Commun.* 71 (1992) 77.
- [13] I. Kisel and V. Kovalenko, *Elastic net for broken multiple scattered tracks.* *Comp. Phys. Commun.* 98 (1996) 45.
- [14] J.S. Bendat and A.G. Piersol, *Random Data: Analysis and Measurement Procedures.* John Wiley & Sons, 1986.

Рукопись поступила в издательский отдел
22 августа 1997 года.