

ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ АЛГОРИТМА ИМИТАЦИИ ОТЖИГА
НА ПРИМЕРЕ РЕШЕНИЯ ЗАДАЧИ ПОИСКА КРАТЧАЙШЕГО ПУТИ В ГРАФЕ*Работа выполнена в рамках выполнения государственного задания
для Юго-Западного государственного университета на 2014–2017 гг., номер НИР 2246,
а также в рамках научной школы НШ-2357.2014.8*

В статье рассматривается вопрос применимости эвристических итерационных методов применительно к решению задач дискретной комбинаторной оптимизации на примере задачи поиска кратчайшего пути с использованием метода имитации отжига. Дано детальное описание алгоритма применительно к решению рассматриваемой задачи, базирующееся на использовании множества модифицирующих операций, к которым относятся вставка вершины в текущий путь, удаление вершин, замена одной из вершин пути на еще не посещенную вершину и перестановка пары вершин пути. Приведено описание результатов метаоптимизации метода, в результате которой получены оптимальные значения настроечных параметров. С их использованием проведен ряд вычислительных экспериментов на выборках случайных графов с различным числом вершин и плотностью, в результате которых установлено, что метод работает эффективно на плотных графах.

Дискретная комбинаторная оптимизация, метод имитации отжига, метаоптимизация, теория графов, поиск кратчайшего пути, эвристические методы.

The article considers the applicability of heuristic iterative methods applied to solving discrete combinatorial optimization problems on the example of finding the shortest path using the method of simulated annealing. The detailed description of the algorithm applied to the solution of the problem is given, based on the using of the set of modifying operations, which include the insert of the vertex to the current path, removing vertex from the current path, replacing one of the vertices on the path by not yet visited vertex and a perform permutation of pair vertices within the current path. The description of meta-optimization results is given; the resulting optimal values of tuning parameters are received. Using the obtained optimal values a number of computational experiments with samples of random graphs with a different number of vertices and different density was organized; as a result it was found out that the method worked effectively on dense graphs.

Discrete combinatorial optimization, simulated annealing, meta-optimization, graph theory, shortest path problem, heuristic methods.

Введение.

Одной из широко известных задач в области дискретной комбинаторной оптимизации является задача поиска кратчайшего пути в графе. Она имеет достаточно простую постановку: в заданном графе $G = \langle A, V \rangle$, где $A = \{a_1, a_2, \dots, a_N\}$ – множество вершин, $N = |A|$ – число вершин, $V = \{v_1, v_2, \dots, v_M\} \subseteq A \times A$ – множество дуг, взвешенных значением длины $l(v_i) = l_{a_j, a_k}$, $M = |V|$ – число дуг, требуется найти кратчайший путь $P = [a_{i_1}, a_{i_2}, \dots, a_{i_Q}]$ между заданной парой вершин $a_{нач.} = a_{i_1}$ и $a_{кон.} = a_{i_Q}$, такой,

что $L(P) = \sum_{j=1}^{Q-1} l_{a_{i_j}, a_{i_{j+1}}} \rightarrow \min$, и допускает получение

оптимального решения за время $O(N^2)$ с использованием алгоритма Дейкстры [7], что позволяет ее использование в качестве тестового полигона для анализа качества решений различных эвристических методов [1]–[5], одним из которых является метод имитации отжига [8].

В его основе лежит имитация процесса отжига металла, в результате которой его атомы образуют кристаллическую решетку, стараясь минимизировать суммарную потенциальную энергию. При достаточно большой начальной температуре каждый из ато-

мов может совершать достаточно большие колебания в пространстве, которые могут приводить в т. ч. и к локальному увеличению суммарной энергии. По мере планомерного снижения температуры возможность колебаний заметно уменьшается и атомы производят все меньшие колебания около состояния с минимальной энергией, которая достигается при достаточно низком конечном значении температуры.

Основная часть.

Аналогией описанного процесса можно воспользоваться при решении дискретных оптимизационных задач. При этом энергии сопоставляется минимизируемая целевая функция $Q(X)$, прыжки атомов соответствуют случайным изменениям решения (вектора параметров X), приводящим к изменению значения целевой функции ΔQ . Если после внесения изменения $\Delta Q < 0$ (полученное решение лучше предыдущего), оно замещает текущее. Если же после внесенного случайного изменения качество решения ухудшилось ($\Delta Q > 0$), производится вычисление

значения вероятности $p = e^{-\frac{\Delta Q}{T}}$, где T – текущая температура, и в случае, если $r_k > p$, где $r_k \in [0; 1]$ – очередное псевдослучайное число, получаемое с использованием генератора псевдослучайных чисел,

решение замещает предыдущее. При этом от итерации к итерации производится уменьшение температуры: $T^{(i)} = \alpha T^{(i-1)}$, где $\alpha \in [0; 1]$ – настроечный параметр алгоритма, регулирующий темп охлаждения, начиная с некоторого начального значения $T^{(0)}$, что приводит ко все меньшим разрешенным ухудшениям качества текущего решения ΔQ . В ходе итерационного процесса производится оценка качества всех промежуточных решений, наилучшее из которых является результатом работы алгоритма.

Приведенное общее описание метода может быть детализировано для конкретной области применения. В поставленной задаче оно может быть представлено в следующем виде. В качестве начального решения будем выбирать путь $P = [a_{\text{нач}}, a_{\text{кон}}]$, который в ходе работы алгоритма подвергается следующим случайным модификациям:

1. Добавление случайной вершины $a_j \notin P$ в случайную позицию R текущего пути: $P' = [a_{i_1}, \dots, a_{i_R}, a_j, a_{i_{R+1}}, \dots, a_{i_Q}]$, $1 < R < Q$, P' – путь после модификации.

2. Удаление случайной вершины $a_j = a_{i_R} \in P$ из текущего пути: $P' = [a_{i_1}, \dots, a_{i_{R-1}}, a_{i_{R+1}}, \dots, a_{i_Q}]$, $1 < R < Q$.

3. Замена случайной вершины a_{i_R} в текущем пути на случайную еще не посещенную вершину $a_j \notin P$: $P' = [a_{i_1}, \dots, a_{i_{R-1}}, a_j, a_{i_{R+1}}, \dots, a_{i_Q}]$, $1 < R < Q$.

4. Перестановка пары выбранных случайно вершин $a_{i_{R_1}}$ и $a_{i_{R_2}}$ текущего пути:

$$P' = [a_{i_1}, \dots, a_{i_{R_1-1}}, a_{i_{R_2}}, a_{i_{R_1+1}}, \dots, a_{i_{R_2-1}}, a_{i_{R_1}}, a_{i_{R_2+1}}, \dots, a_{i_Q}],$$

$$1 < R_1 < Q, 1 < R_2 < Q, R_1 \neq R_2.$$

Несложно заметить, что добавление и замену вершины (модификации 1 и 3) можно выполнять только в случае наличия еще не посещенных вершин ($|P| < N$), удаление вершины (модификация 2) только при $|P| > 2$, перестановку пары вершин (модификация 4) – при $|P| > 3$. При этом по условию задачи первая и последняя вершины пути не должны подвергаться изменению в ходе выполнения модификаций с целью получения на выходе корректных решений. Выбор одной из рассмотренных модификаций текущего пути будем производить случайным образом пропорционально соответствующим значениям параметров p_1, p_2, p_3, p_4 (значение $\frac{p_i}{\sum_{j=1}^4 p_j}$ можно

воспринимать как вероятность применения i -й модификации). Следует ожидать, что от их значений может зависеть как качество результирующих решений, так и скорость сходимости.

Алгоритм решения поставленной задачи, соответствующий приведенному выше описанию метода

имитации отжига, может быть представлен в следующем виде.

1. (Инициализация). Задать начальное значение температуры $T := T^{(0)}$, номер итерации $i := 1$, текущий путь $P := [a_{\text{нач}}, a_{\text{кон}}]$, лучший путь $P^+ := [a_{\text{нач}}, a_{\text{кон}}]$.

2. Произвести случайную модификацию текущего пути P путем применения к нему одной из описанных выше модификаций.

3. Оценить полученное изменение качества решения – длины пути: $\Delta L = L(P') - L(P)$, где P' – путь после случайной модификации. Рассчитать значение $p = e^{-\frac{\Delta L}{T}}$.

4. Если $\Delta L < 0$ или $(\Delta L > 0) \wedge (r_k > p)$, положить $P := P'$.

5. Если $L(P) < L(P^+)$, положить $P^+ := P$.

6. Произвести модификацию температуры $T := \alpha T$ и номера итерации $i := i + 1$.

7. Если $i < C$, где C – заданное число итераций, перейти к п. 2.

8. Конец алгоритма.

Для работы алгоритма с максимальной эффективностью требуется задание начальных значений настроечных параметров метода $T^{(0)}, \alpha, C$, а также значений $p_1, \dots, p_4$. Определение их оптимальных значений возможно осуществить в ходе параметрической оптимизации путем выполнения серии вычислительных экспериментов. Следуя работам [1]–[4], будем осуществлять формирование выборки $\Lambda = \{G_1, G_2, \dots, G_K\}$ из K псевдослучайных графов с заданными параметрами, к которым относятся число вершин N и плотность графа $0 \leq d = \frac{M}{N(N-1)} \leq 1$,

дуги которых взвешены псевдослучайными значениями длины $l(v_i)$, имеющими равномерное распределение в диапазоне $[0; 1]$. Для оценки качества получаемых решений будем производить определение следующих средневыбросных параметров: средняя длина пути $\bar{L}$; среднее отклонение $\Delta \bar{L}$ длины пути от минимальной $L_{\min}(G_i)$, получаемой с использованием алгоритма Дейкстры; вероятность нахождения решения $\bar{p}$ (отсутствия в составе пути запрещенных переходов); вероятность нахождения оптимального решения $\bar{p}_{\text{opt}}$ и среднее время получения решения t , для оценки которого использован компьютер с процессором Intel Core i7 4770 @ 3,4 ГГц (Haswell).

В ходе серии вычислительных экспериментов с использованием разработанной программной реализации [6], выполненных при $N = 10$, $d = 0,9$, $K = 1000$ и реализующих стратегию покоординатного спуска, были получены следующие оптимальные значения настроечных параметров алгоритма:

$T^{(0)*} = 10 \div 100$, $\alpha^* = 0,999 \div 0,99999$, $p_1^* = 1$,
 $p_2^* = 0,8$, $p_3^* = 2$, $p_4^* = 0$. Соответствующие зави-
 симости средней длины пути от параметров алго-
 ритма приведены на рисунке.

Число итераций, при котором качество решений
 практически перестает изменяться, составляет вели-
 чину порядка $C \simeq 10^4 \div 10^5$ и может потребовать до-

полнительного увеличения с ростом размерности
 задачи N .

С использованием полученных оптимальных па-
 раметров алгоритма можно произвести дополни-
 тельную серию вычислительных экспериментов с
 целью сопоставления качества получаемых решений
 с оптимумом (табл. 1–3).

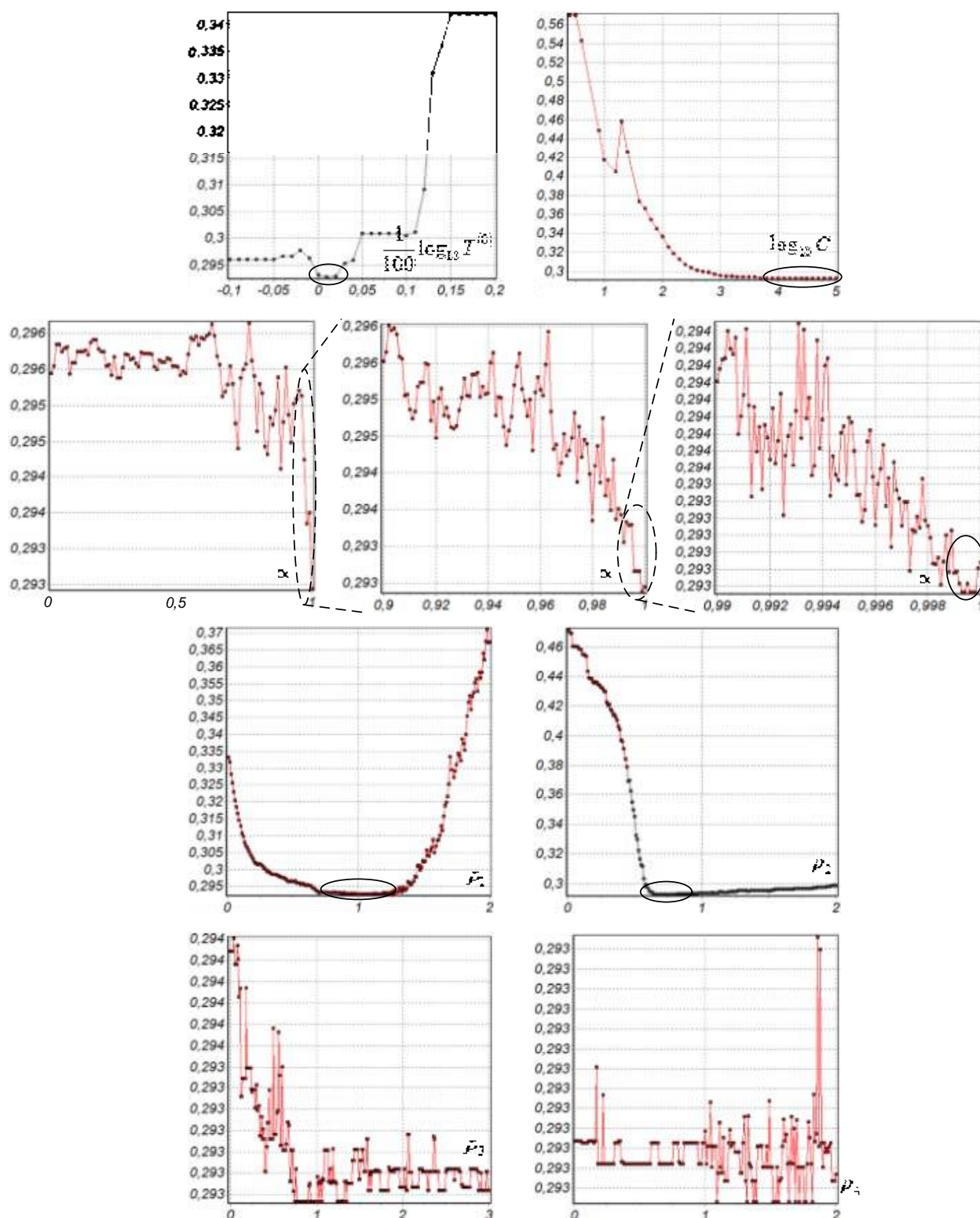


Рисунок. Зависимости средней длины пути $\bar{L}$ от настроечных параметров алгоритма

Таблица 1

Результаты вычислительного эксперимента
 $N = 10$, $d = 0,5$, $K = 1000$

Метод	$\bar{L}$	$\Delta\bar{L}$	$\bar{P}$	$\bar{P}_{opt}$	t
Дейкстры	0,4804	0	1	1	3,7 мкс
Имитации отжига	$C = 10^3$	0,5756	0,0988	0,987	0,709
	$C = 10^4$	0,4894	0,0126	0,987	0,925
	$C = 10^5$	0,4768*	0,0000	0,987	0,986

* З а м е ч а н и е . Полученное среднее значение меньше оптимального в связи с тем, что не для всех графов были найдены кратчайшие пути [4]

Таблица 2

Результаты вычислительного эксперимента
 $N = 100$, $d = 0,5$, $K = 1000$

Метод	$\bar{L}$	$\Delta\bar{L}$	$\bar{P}$	$\bar{P}_{opt}$	t
Дейкстры	0,0967	0	1	1	0,16 мс
Имитации отжига	$C = 10^3$	0,4063	0,3096	1	0,073
	$C = 10^4$	0,2461	0,1493	1	0,157
	$C = 10^5$	0,1773	0,0805	1	0,226

Таблица 3

Результаты вычислительного эксперимента
 $N = 100$, $d = 0,01$, $K = 1000$

Метод	$\bar{L}$	$\Delta\bar{L}$	$\bar{P}$	$\bar{P}_{opt}$	t
Дейкстры	0,9225	0	1	1	0,10 мс
Имитации отжига	$C = 10^3$	0,8566	0,1570	0,351	0,202
	$C = 10^4$	0,8357	0,1361	0,351	0,214
	$C = 10^5$	0,7745	0,0749	0,351	0,250

Выводы.

Полученные данные позволяют сделать вывод о том, что для плотных графов ($d > 0,5$) метод имитации отжига при достаточном числе итераций C обес-

печивает получение решений, близких к оптимуму. С ростом размерности задачи требуемое число итераций и, как следствие, необходимые затраты вычислительного времени существенно возрастают. Для графов малой плотности, в которых значительная часть путей между вершинами оказывается запрещенной, метод демонстрирует значительное снижение вероятности отыскания решения.

Литература

1. Ватулин, Э. И. Анализ результатов использования метода перебора с ограничением глубины в задаче поиска кратчайшего пути в графе / Э. И. Ватулин, И. А. Мартынов, В. С. Титов // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов (МППОС'15). – Барнаул, 2015. – С. 120–128.
2. Ватулин, Э. И. Анализ результатов применения алгоритма муравьиной колонии в задаче поиска пути в графе при наличии ограничений / Э. И. Ватулин, В. С. Титов // Известия Южного федерального университета. Технические науки. 2014. № 12 (161). С. 111–120.
3. Ватулин, Э. И. Анализ результатов применения метода случайного перебора в задаче поиска разбиений граф-схем параллельных алгоритмов / Э. И. Ватулин, Д. В. Колясников, В. С. Титов // Известия Южного федерального университета. Технические науки. – 2014. – № 12 (161). – С. 102–110.
4. Ватулин, Э. И. Метод взвешенного случайного перебора для решения задач дискретной комбинаторной оптимизации / [Э. И. Ватулин и др.] // Известия ВолГТУ. Сер.: Электроника, измерительная техника, радиотехника и связь. – 2014. – № 10 (137). – Вып. 9. – С. 59–64.
5. Ватулин, Э. И. Метод случайного перебора в задаче построения разбиений граф-схем параллельных алгоритмов / [Э. И. Ватулин и др.] // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов. – Барнаул, 2014. – С. 115–125.
6. Ватулин, Э. И. Расчетный модуль для тестирования комбинаторных оптимизационных алгоритмов в задаче поиска кратчайшего пути в графе с использованием добровольных распределенных вычислений / Э. И. Ватулин, С. Ю. Валяев, Е. Н. Дремов, И. А. Мартынов, В. С. Титов // Свидетельство о государственной регистрации программы для ЭВМ № 2014619797. – 2014. – 22 сентября.
7. Dijkstra, E. W. A note on two problems in connexion with graphs / E. W. Dijkstra // Numerische Mathematik. – 1959. – V. 1. – P. 269–271.
8. Kirkpatrick, S. Optimization by Simulated Annealing / S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi // Science. – 1983. – Vol. 220. – № 4598. – P. 671–680.

К. С. Калугин

Научный руководитель: доктор технических наук, профессор В. С. Титов
 Юго-Западный государственный университет,

В. С. Панищев

Центр информационных технологий в проектировании РАН

АЛГОРИТМ ПОСТРОЕНИЯ ПАНОРАМНЫХ ИЗОБРАЖЕНИЙ

Представлен алгоритм программы, позволяющей из набора отдельных изображений строить панорамные изображения и производить последующую их стабилизацию.