

Параметрическая оптимизация алгоритма имитации отжига на примере решения задачи поиска кратчайшего пути в графе

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 [1], что позволяет ее использование в качестве тестового полигона для анализа качества решений различных эвристических методов [2–6], одним из которых является метод имитации отжига [7].

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 атомов может совершать достаточно большие колебания в пространстве, которые могут приводит 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.

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 задачи. В поставленной задаче оно может быть представлено в следующем виде. В качестве начального решения будем выбирать путь $P = [a_{нач}, a_{кон}]$, который в ходе работы алгоритма подвергается следующим случайным модификациям:

1. Добавление случайной вершины $a_j \notin P$ в случайную позицию R текущего пути:

$$P' = [a_{i_1}, \dots, a_{i_R}, a_j, a_{i_{R+1}}, \dots, a_{i_Q}], 1 < R < Q, P' - \text{путь после модификации.}$$
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 $P := [a_{нач}, a_{кон}]$, лучший путь $P^+ := [a_{нач}, a_{кон}]$.
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 [3–6], 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 средневыборочных 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}_{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 [8], 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 зависимости средней длины пути от параметров алгоритма приведены на рис. 1.

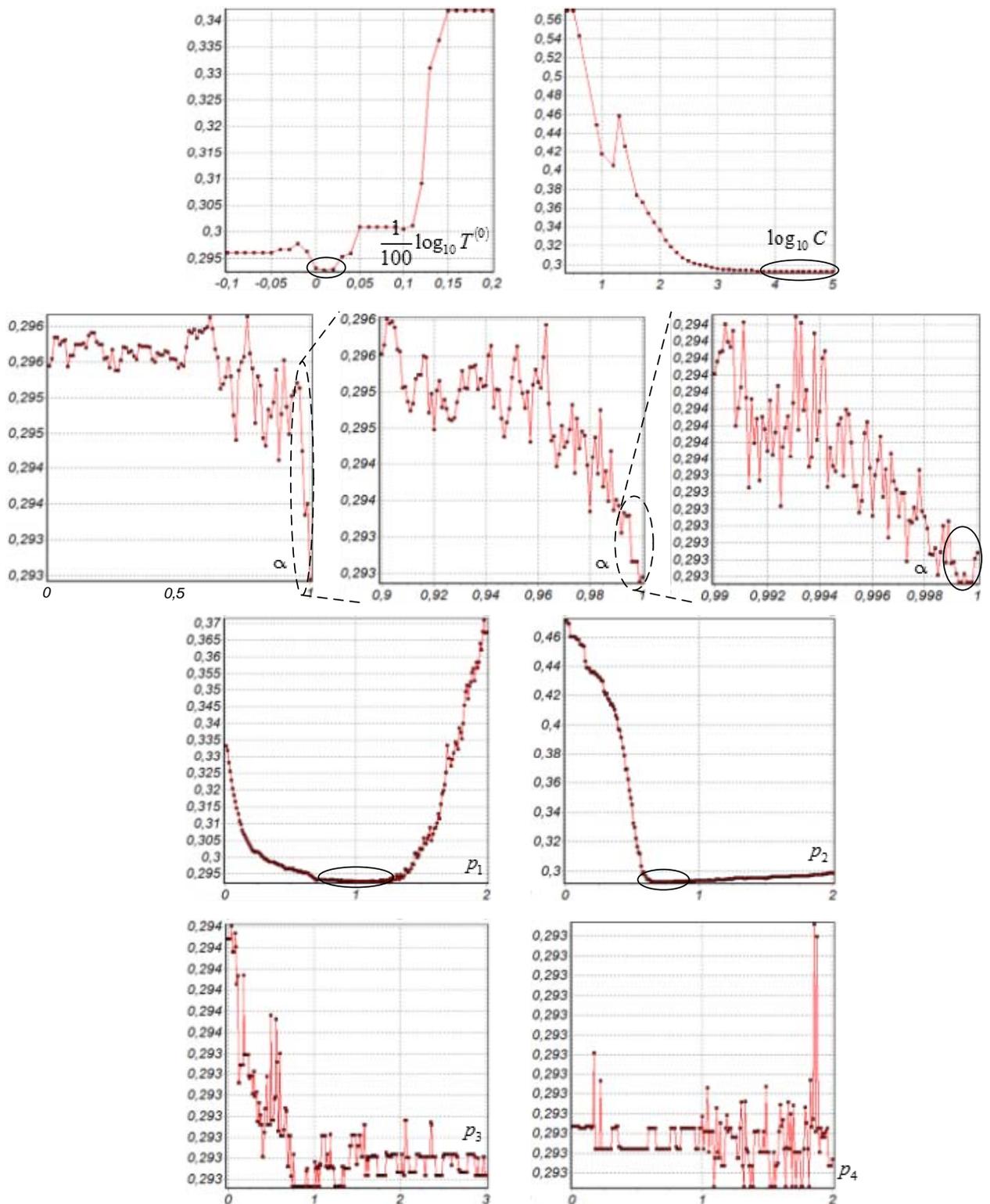


Рис. 1. Зависимости средней длины пути \bar{L} от настроечных параметров алгоритма

Число итераций, при котором качество решений практически перестает изменяться, составляет величину порядка $C \simeq 10^4 \div 10^5$, что может потребовать дополнительного увеличения числа итераций с ростом размерности задачи N .

С использованием полученных оптимальных параметров алгоритма можно произвести дополнительную серию вычислительных экспериментов с целью сопоставления качества получаемых решений с оптимумом (табл. 1–3).

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	40 мкс
	$C = 10^4$	0,4894	0,0126	0,987	0,925	420 мкс
	$C = 10^5$	0,4768*	0,0000	0,987	0,986	4 мс

* Замечание. 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	0,07 мс
	$C = 10^4$	0,2461	0,1493	1	0,157	0,69 мс
	$C = 10^5$	0,1773	0,0805	1	0,226	6,40 мс

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	0,57 мс
	$C = 10^4$	0,8357	0,1361	0,351	0,214	5,23 мс
	$C = 10^5$	0,7745	0,0749	0,351	0,250	57,11 мс

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

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.

Библиографический список

1. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. V. 1 (1959), PP. 269–271.
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. 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.
6. Ватутин Э.И., Мартынов И.А., Титов В.С. Анализ результатов использования метода перебора с ограничением глубины в задаче поиска кратчайшего пути в графе // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов (МППОС'15). Барнаул, 2015. С. 120–128.
7. 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
8. Ватутин Э.И., Валяев С.Ю., Дремов Е.Н., Мартынов И.А., Титов В.С. Расчетный модуль для тестирования комбинаторных оптимизационных алгоритмов в задаче поиска кратчайшего пути в графе с использованием добровольных распределенных вычислений // Свидетельство о государственной регистрации программы для ЭВМ № 2014619797 от 22.09.14.