



Рис. 1. Производительность поисковых систем

По графикам видно, что битовые автоматы всегда работают быстрее регистрового поиска, полубайтовые автоматы всегда быстрее битовых, а байтовые достигают очень высокой производительности при небольших объёмах таблиц, однако с ростом числа состояний их производительность падает настолько быстро, что они начинают проигрывать полубайтовым, а затем и битовым автоматам.

Можно заключить, что регистровый поиск следует использовать только в том случае, когда доступное количество памяти недостаточно даже для хранения таблиц битового автомата, битовый автомат — только если недостаточно памяти для полубайтового автомата. Выбор между байтовым и полубайтовым автоматами следует делать, исходя из измерений производительности обоих автоматов на целевом оборудовании.

#### ЛИТЕРАТУРА

1. Агibalов Г. П., Оранов А. М. Лекции по теории конечных автоматов. Томск: Изд-во Том. ун-та, 1984. 185 с.
2. Панкратов И. В. Одновременный поиск нескольких двоичных шаблонов в потоке с помощью конечного автомата // Прикладная дискретная математика. 2014. №2. С. 119–125.
3. Knuth D. E., Morris J. H. Jr., and Pratt V. R. Fast pattern matching in strings // SIAM J. Comput. 1977. No. 6(2). P. 323–350.
4. Aho A. V. and Corasick M. J. Efficient string matching: An aid to bibliographic search // Commun. ACM. 1975. No. 18(6). P. 333–340.

УДК 519.7

DOI 10.17223/2226308X/11/38

### АЛГОРИТМ ОПТИМАЛЬНОЙ МАРШРУТИЗАЦИИ В МУЛЬТИСЕРВИСНЫХ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЯХ

А. А. Солдатенко

Рассматривается NP-трудная задача поиска ресурсоограниченного кратчайшего пути (RCSP) в графе  $G = (V, E)$ . Задача RCSP является расширением извест-

ной задачи о кратчайшем пути в ориентированном графе, когда каждой дуге  $e \in E$  кроме основного веса  $w(e)$  дополнительно задаются несколько весовых функций  $w_r(e)$ , отражающих потребности в ресурсах, которые необходимы для передвижения по этой дуге. Задача RCSP позволяет моделировать мультисервисные телекоммуникационные сети и определять в них оптимальный маршрут передачи данных между двумя заданными узлами сети. Предлагаются два алгоритма для приближённого решения задач RCSP на сетях большой размерности. Первый алгоритм является расширением известного алгоритма Дейкстры, который присваивает дополнительные метки каждой вершине. Эти метки позволяют алгоритму Дейкстры находить ресурсоограниченный путь в графе. В отличие от известных модификаций, такая модификация не требует дополнительных знаний о графе. Второй алгоритм, помимо дополнительных меток, добавляет к алгоритму Дейкстры потенциальные функции и ориентиры. Потенциальные функции, вычисленные на основе ориентиров, дают значительное ускорение на графах большой размерности. Предлагаемые алгоритмы не превосходят по вычислительной сложности алгоритм Дейкстры. Приводятся результаты вычислительных экспериментов, подтверждающие эффективность предложенных алгоритмов.

**Ключевые слова:** *ресурсоограниченный кратчайший путь, графы большой размерности.*

Большинство современных телекоммуникационных сетей являются мультисервисными, поскольку предоставляют пользователям множество разнообразных услуг, касающихся проводной телефонии, сотовой связи, кабельного телевидения и передачи данных. Качество обслуживания (Quality of Service, QoS) в мультисервисных сетях определяется такими параметрами, как полоса пропускания, задержка, вариация задержки, стоимость и надёжность передачи данных [1]. Всякий запрос пользователя на оказание услуги предполагает определение некоторого (желательно, оптимального по затратам, например, по времени или стоимости) маршрута между двумя узлами с учётом ресурсных возможностей этой сети. Таким образом, задача оптимального обслуживания пользователя в мультисервисной сети может быть математически сформулирована как задача поиска ресурсоограниченного кратчайшего пути в графе. Данная задача известна в литературе под названием Resource Constrained Shortest Path (RCSP) [2, 3]. На практике задача RCSP возникает как при проектировании, так и при эксплуатации мультисервисных сетей. На этапе проектирования с помощью многократного решения задачи RCSP возможна оценка качества услуг, которую проектируемая сеть способна предоставить потенциальным пользователям. При эксплуатации сети RCSP решается в реальном масштабе времени для каждого отдельного запроса пользователя на какую-либо услугу. В настоящее время мультисервисные сети активно расширяются путем увеличения числа предоставляемых услуг и входящих в них узлов. Количество узлов в сети может достигать нескольких сотен тысяч. Именно для таких больших сетей, чаще всего, требуется находить решение задачи RCSP в реальных условиях.

Известны две формулировки задачи RCSP: в терминах теории графов и целочисленного линейного программирования. В теоретико-графовой формулировке мультисервисная сеть представляется ориентированным графом, вершины которого соответствуют узлам сети, а дуги — каналам связи. Предполагается, что всякая дуга обладает основным весом (это может быть стоимость или время прохождения по дуге), а также некоторыми дополнительными весами, отражающими потребности в ресурсах, которые нужны для передвижения по этой дуге. Требуется найти кратчайший путь между

двумя заданными узлами сети, отвечающий заданным ограничениям на итоговые ресурсные затраты, необходимые для прохождения этого пути. Показано, что даже при ограничении на один ресурс задача RCSP является NP-трудной [4].

В настоящее время выделяют три класса методов и алгоритмов, способных найти точное или приближенное решение задачи RCSP: методы ранжирования путей [4], методы маркировки вершин [1, 2, 5], методы лагранжевой релаксации [6–8]. Первые два класса методов основаны на теоретико-графовой постановке задачи, методы третьего класса исходят из постановки задачи RCSP на языке целочисленного линейного программирования. Большинство существующих алгоритмов не приспособлены к сетям большой размерности и, как следствие, не отвечают реальному масштабу времени, необходимому для обработки запросов при эксплуатации мультисервисных сетей. Поэтому актуальны различные приёмы и методы ускорения существующих алгоритмов, а также разработка новых подходов к решению задачи RCSP [1, 8].

Рассмотрим задачу RCSP в теоретико-графовой постановке, используя общепринятые в теории графов терминологию и обозначения [9]. Пусть мультисервисная сеть описана взвешенным ориентированным графом (далее просто графом)  $G = (V, E)$  без кратных дуг и петель, в котором каждая вершина  $v \in V$  представляет узел сети, каждая дуга  $e \in E$  — канал связи между соответствующими узлами сети. Считаем, что на множестве дуг графа  $G$  задана функция  $w(e): E \rightarrow \mathbb{R}^+$ , ставящая в соответствие каждой дуге  $e \in E$  её вес  $w(e) \geq 0$ . Пусть для вершин  $s, d \in V$  в графе  $G$  существует путь  $P$ , идущий от вершины  $s$  к вершине  $d$ . Полагаем, что вес этого  $(s, d)$ -пути  $P$  вычисляется как сумма весов всех входящих в него дуг:

$$w(P) = \sum_{e \in P} w(e), \quad (1)$$

т. е. функция  $w(P)$  является аддитивной. Если  $w(e)$  — стоимость передвижения по дуге  $e$ , то  $w(P)$  — стоимость прохождения  $(s, d)$ -пути  $P$  в графе  $G$ .

Пусть для каждой дуги графа  $G = (V, E)$ , кроме  $w(e)$ , заданы весовые функции  $w_r(e): E \rightarrow \mathbb{R}^+$ ,  $r = 1, \dots, k$ , отражающие потребности в ресурсах, которые необходимы для передвижения по этой дуге. Предполагается, что все эти функции аддитивные, т. е. для любого  $(s, d)$ -пути  $P$  верны равенства

$$w_r(P) = \sum_{e \in P} w_r(e), \quad r = 1, \dots, k.$$

Считаем, что заданы также величины  $W_r \in \mathbb{R}^+$ ,  $r = 1, \dots, k$ , определяющие ресурсные ограничения мультисервисной сети. Всякий  $(s, d)$ -путь  $P$  называется допустимым, если он удовлетворяет ресурсным ограничениям

$$\sum_{e \in P} w_r(e) \leq W_r, \quad r = 1, \dots, k. \quad (2)$$

Очевидно, что кратчайший  $(s, d)$ -путь в графе  $G$  не обязательно допустимый.

Задачу RCSP можно сформулировать следующим образом. Заданы граф  $G = (V, E)$ , на дугах которого определены неотрицательные вещественнозначные функции  $w(e)$  и  $w_r(e)$ ; величины  $W_r$ ,  $r = 1, \dots, k$ ;  $(s, d)$ -запрос. Требуется найти в  $G$  ресурсоограниченный кратчайший  $(s, d)$ -путь, т. е. такой  $(s, d)$ -путь  $P$ , который минимизирует  $w(P)$  в (1) и удовлетворяет ограничениям (2). Задача RCSP не имеет решения, если множество допустимых  $(s, d)$ -путей пустое.

Для решения задачи RCSP предлагается эвристический алгоритм  $A_1$ , являющийся модификацией алгоритма Дейкстры. Как известно, алгоритм Дейкстры — маркировочная процедура, осуществляющая поиск кратчайшего пути в графе на основе меток вершин за полиномиальное время [5]. В алгоритме  $A_1$  к меткам алгоритма Дейкстры добавлены ещё две метки: величина затраченных ресурсов при прохождении пути от стартовой вершины  $s$  до текущей вершины  $v$  и допустимость (да/нет) использования вершины  $v$  для дальнейшего построения искомого пути. Применение этих меток позволяет, прежде всего, находить допустимые пути и далее среди них кратчайший путь, не увеличивая существенно время поиска. В остальном алгоритм  $A_1$  полностью соответствует классическому алгоритму Дейкстры. Если множество допустимых решений задачи RCSP непустое, то решение, найденное алгоритмом  $A_1$ , всегда допустимое, но не обязательно оптимальное.

С целью применения алгоритма Дейкстры к графам большой размерности в настоящее время разрабатываются всевозможные приёмы ускорения [10, 11]. Одним из таких приёмов является применение потенциальных функций, которые могут определяться различными способами, в частности с помощью ориентиров. Предлагаемый двухфазный алгоритм  $A_2$  предназначен для ускорения алгоритма  $A_1$ . На первой фазе алгоритм  $A_2$  производит расстановку ориентиров в вершинах графа и вычисляет потенциальные функции, на второй фазе — осуществляет с помощью алгоритма  $A_1$  и вычисленных потенциальных функций поиск ресурсоограниченного оптимального маршрута в графе. Алгоритм  $A_2$  имеет следующие параметры: число ориентиров и стратегия их расстановки (случайным образом в вершинах графа, случайным образом в вершинах, образующих оболочку графа, и др.). Для вычисления потенциальных функций алгоритм  $A_2$  использует евклидову метрику.

Для оценки эффективности предложенных алгоритмов проведены серии вычислительных экспериментов. В табл. 1 представлены параметры графов, на которых проводились эксперименты. Первая серия экспериментов состояла в сравнении алгоритма  $A_1$  и пакета IBM ILOG CPLEX [12]. Эксперименты проводились на случайно сгенерированных графах с различным числом вершин и рёбер для последовательности  $\sigma$ , состоящей из 1000 случайно сгенерированных  $(s, d)$ -запросов. Сравнение осуществлялось по числу выполненных запросов, точности найденного решения и времени работы алгоритмов. Запрос считался выполненным, если для него получено точное или приближенное решение задачи RCSP. Точность приближённого решения, найденного алгоритмом  $A_1$ , оценивалась относительно результата, вычисленного с помощью CPLEX. В качестве результата пакет программ CPLEX всегда выдаёт оптимальное решение задачи RCSP, если множество допустимых решений не пусто.

Т а б л и ц а 1  
Параметры графов

Название графа	Число вершин	Число дуг
$G_1$	100	720
$G_2$	625	4800
$G_3$	2500	19600
$AK$	69082	157662
$CT$	153011	375310
$MA$	308401	771362

Результаты первой серии экспериментов представлены в табл. 2, где  $q_{\text{err}}$  обозначает количество запросов, для которых найдено приближённое решение;  $\text{err}$  — макси-

мальная относительная погрешность в процентах. По этим результатам можно сделать следующие выводы. Пакет CPLEX для рассматриваемых графов с заданными ресурсными ограничениями выполняет примерно половину запросов последовательности  $\sigma$ . Время нахождения точного решения существенно зависит от размеров графа и числа рассматриваемых ресурсов. Например, для исходного графа с числом вершин 2500, числом рёбер 19600 и количеством ресурсов 10 с помощью пакета CPLEX не удалось получить решение за 1000 с. Алгоритм  $A_1$  выполняет примерно такую же долю запросов, что пакет CPLEX, затрачивая на это на два порядка меньше времени. Эвристический характер алгоритма не исключает ситуации, когда построенное решение не является точным. Однако, как показывают эксперименты, это происходит сравнительно редко и с малой относительной погрешностью. Таким образом, алгоритм  $A_1$  можно применять к реальным мультисервисным сетям с числом узлов, не превосходящим 10 000, для которых приемлема небольшая потеря точности в решении.

Таблица 2

**Результаты сравнения алгоритма  $A_1$  с пакетом CPLEX**

Название графа	Количество ресурсов	CPLEX		$A_1$			err
		Время обработки $\sigma$ , с	Число выполненных запросов	Время обработки $\sigma$ , с	Число выполненных запросов		
					всего	$q_{err}$	
$G_1$	1	39,8	945	0,050	937	1	2
$G_2$	1	79,1	855	0,339	829	1	0,4
$G_3$	1	267,2	959	5,919	944	3	3
$G_1$	2	39,6	961	0,048	872	4	7
$G_2$	2	79,6	807	0,294	757	3	4
$G_3$	2	284,1	799	3,541	783	6	0,7
$G_1$	5	40,0	756	0,029	727	3	11
$G_2$	5	89,0	703	0,264	701	2	1
$G_3$	5	362,6	666	2,337	647	4	0,2
$G_1$	10	43,6	828	0,038	807	0	0
$G_2$	10	111,9	712	0,245	713	7	1
$G_3$	10	–	–	2,836	694	–	–

Цель второй серии экспериментов — сравнение алгоритмов  $A_1$  и  $A_2$  на графах большой размерности. Для алгоритма  $A_2$  выбраны следующие параметры: число ориентиров — 12, стратегия их расстановки — случайное размещение в вершинах графа. Сравнение производилось по времени работы и доле выполненных запросов. Результаты представлены в табл. 3. Видно, что время работы алгоритма  $A_2$  меньше времени работы алгоритма  $A_1$  в несколько раз, при этом число выполненных запросов совпадает. Это позволяет сделать следующий вывод: алгоритм  $A_2$  можно применять к реальным мультисервисным сетям с числом узлов в несколько сотен тысяч, для которых приемлема небольшая потеря точности в решении.

Во время экспериментов полагалось, что все поступающие запросы однородны и не конкурируют между собой. В реальных сетях возможны ситуации, когда несколько запросов конкурируют за ресурсы или имеют разный характер, например различные требования к ресурсам. Таким образом, требуется дальнейшее развитие алгоритмов для эффективной маршрутизации в подобных сетях.

Т а б л и ц а 3

Результаты сравнения алгоритмов  $A_1$  и  $A_2$

Название графа	Количество ресурсов	$A_1$		$A_2$	
		Число выполненных запросов	Время обработки $\sigma$ , с	Число выполненных запросов	Время обработки $\sigma$ , с
AK	1	531	511,171	531	434,538
CT	1	821	1100,88	821	117,283
MA	1	738	687,516	738	103,052
AK	2	587	548,222	587	397,847
CT	2	667	735,669	667	73,765
MA	2	802	776,492	802	158,573
AK	5	523	473,254	523	368,664
CT	5	813	1152,33	813	123,665
MA	5	806	756,533	806	137,998
AK	10	541	410,33	541	329,267
CT	10	679	688,384	679	74,905
MA	10	557	258,307	557	52,353

ЛИТЕРАТУРА

1. Van Mieghem P., Kuipers F. A., Korkmaz T., et al. Quality of service routing // LNCS. 2003. V. 2856. P. 80–117.
2. Joksch H. C. The shortest route problem with constraints // J. Math. Analysis Appl. 1966. V. 14. P. 191–197.
3. Dror M. Note on the complexity of the shortest path models for column generation in VRPTW // J. Operat. Res. 1994. V. 42. P. 977–978.
4. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.
5. Dumitrescu I. and Boland N. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem // J. Networks. 2003. V. 42. P. 135–153.
6. Mehlhorn K. and Ziegelmann M. Resource constrained shortest paths // LNCS. 2000. V. 1879. P. 326–337.
7. Jepsen M., Petersen B., Spoorendonk S., and Pisinger D. A branch-and-cut algorithm for the capacitated profitable tour problem // J. Discr. Optimization. 2014. V. 14. P. 78–96.
8. Horvath M. and Kis T. Solving resource constrained shortest path problems with LP-based methods // J. Computers & Operat. Res. 2016. V. 73. P. 150–164.
9. Емеличев В. А., Мельников О. И., Сарванов В. И., Тышкевич Р. И. Лекции по теории графов. М.: Книжный дом «Либроком», 2012. 390 с.
10. Быкова В. В., Солдатенко А. А. Оптимальная маршрутизация по ориентирам в нестационарных сетях // Прикладная дискретная математика. 2017. № 37. С. 114–123.
11. Быкова В. В., Солдатенко А. А. Адаптивное размещение ориентиров в задаче о кратчайшем пути для графов большой размерности // Программные продукты и системы. 2016. № 1. С. 60–67.
12. IBM ILOG CPLEX Optimizer Studio. CPLEX User's Manual. Version 12 Release 6, 2015.