DOI: 10.48542/mold/2022/parfenov.118-141

НАХОЖДЕНИЕ МАКСИМАЛЬНОГО ПОТОКА В СЕТИ С НЕПРЕРЫВНЫМИ КУСОЧНО-ЛИНЕЙНЫМИ УСИЛЕНИЯМИ

Парфенов Андрей Павлович

ФГБУН Институт проблем региональной экономики Российской академии наук, Санкт-Петербург (научный сотрудник, кандидат физико-математических наук)
e-mail: keldoor@gmail.com, ORCID: 0000-0002-1929-8412

Аннотация: *Рассмотрена задача нахождения максимального потока в сети с нелинейными функциями усиления в дугах. Она обобщает аналогичную задачу для сети с линейными функциями усиления, которая, в свою очередь, обобщает классическую задачу о максимальном потоке в сети.*

Для случая вогнутых кусочно-линейных усилений данная задача сведена к задаче нахождения максимального потока в сети с линейными усилениями в параллельных дугах, для которой есть эффективные алгоритмы решения.

Построен полупереборный алгоритм нахождения максимального потока в сети с непрерывными кусочно-линейными усилениями, основанный на рассмотрении непрерывной кусочно-линейной функции как максимума из вогнутых кусочно-линейных функций. Показано, что задача о рюкзаке эквивалентна частному случаю данной задачи. Метод ветвей и границ в задаче о рюкзаке обобщается на данную задачу.

Ключевые слова: *потоки в сетях, кусочно-линейное программирование, сетевая оптимизация.*

A MAXIMAL FLOW IN A NETWORK WITH CONTINUOUS PIECEWISE LINEAR GAINS

Parfenov Andrey Pavlovich

Institute for Regional Economic Studies Russian Academy of Sciences, Saint-Petersburg (Researcher, PhD in Physics and Mathematics)

Abstract: *Maximal flow searching problem is considered for a network with non-linear gains in arcs. This problem is a generalization of simular one for a network with linear gains in arcs. And linear problem is a generalization of a well-known problem: finding a maximal flow in a network.*

A case of concave piecewise linear gains is considered. For this case our task is reduced to a case of linear gains. For linear task there are known efficient algorithms.

Then any continuous piecewise linear gains are considered. They are defined as maximums of concave piecewise linear gains. A semi-iterative algorithm searching maximal *flow based on this representation is constructed. The knapsack problem is reduced to a non-linear maximal flow problem. The method of branches and boundaries in the knacksack problem is generalized to this problem.*

Keywords: *network flows, piecewise linear programming, network optimization.*

Введение

Важный класс математических задач сетевой оптимизации – потоковые задачи, обобщающие классическую задачу о максимальном потоке и минимальном сечении [1]. Эти задачи имеют разнообразные применения в оптимизации экономических и технологических систем.

Обобщение задачи о максимальном потоке проводилось в разных направлениях: задача о потоке минимальной стоимости, задача о перевозках, о многопродуктовом потоке, о потоке в сети с усилениями и т.п. [1].

В последнем случае предполагается, что в каждой дуге графа задан коэффициент усиления, определяющий, во сколько раз усиливается либо ослабляется поток, проходящий через эту дугу. Таким образом, значение потока на выходе из дуги зависит от значения потока на входе, причем зависимость – линейная функция.

Задача максимизации потока в такой сети также является задачей линейного программирования. Но ее комбинаторное решение сталкивается с рядом сложностей: в частности, даже если все коэффициенты в задаче целочисленные, не гарантирована целочисленность оптимального решения. Тем не менее, существует ряд эффективных алгоритмов как приближенного, так и точного решения данной задачи [4, 5 и т.п.].

В данной статье рассмотрены задачи нахождения максимального потока в сети с нелинейными функциями усиления в дугах. Эти задачи обобщает аналогичные задачи для сети с линейными функциями усиления, которые, в свою очередь, обобщают классическую задачу о максимальном потоке в сети.

Данные задачи применимы, например, к моделированию транспортных грузопотоков или потоков энергии при наличии потерь в каналах траспортировки. Функция потерь может нелинейно зависеть от количества траспортируемой продукции.

Более подробно рассмотрен случай непрерывных кусочно-линейных усилений в дугах. В этом случае задача максимизации потока является задачей кусочно-линейного программирования. В общем случае, задачи кусочно-линейного программирования являются NP-сложными, поэтому для них актуальны эвристические алгоритмы оптимизации – такие, как метод ветвей и границ [3]. К тому же, они не являются дискретными, а целочисленность решений не гарантирована, что делает невозможным полный перебор вариантов. Следовательно, необходимо построить алгоритмы частичного перебора, основанные на комбинаторных идеях, что и сделано в данной статье.

Доказана теорема о композиции параллельных дуг, позволяющая объединять и расщеплять параллельные дуги в сети, рассматривая усиление в объединенной дуге как решение вспомогательной оптимизационной задачи. Теорема о композиции дуг позволяет, во-первых, переходить от мультиграфа к простому графу, во-вторых, сводить функцию усиления в дуге к набору более простых функций для параллельных дуг.

Доказана теорема о достижении максимального значения оптимальным потоком, позволяющая свести задачу оптимизации потока в сети с непрерывными кусочно-линейными усилениями к семейству задач с вогнутыми кусочно-линейными усилениями, и теорема о композиции потоков, сводящая задачу с вогнутыми кусочно-линейными усилениями к задаче с линейными усилениями.

Таким образом, задача нахождения максимального потока в сети с вогнутыми кусочно-линейными усилениями сводится к задаче нахождения максимального потока в сети с линейными усилениями в параллельных дугах, для которой есть эффективные алгоритмы решения.

Построен полупереборный алгоритм нахождения максимального потока в сети с непрерывными кусочно-линейными усилениями, основанный на рассмотрении непрерывной кусочно-линейной функции как максимума из вогнутых кусочно-линейных функций.

Рассмотрены связи задачи о рюкзаке с задачей о максимальном потоке. Показано, что задача о рюкзаке сводится к частному случаю задачи о максимальном потоке в сети с непрерывными кусочно-линейными усилениями. Метод ветвей и границ для решения задачи о рюкзаке, а также жадный алгоритм для нахождения ее приближенного начального решения, обобщаются на задачу о максимальном потоке.

Некоторые результаты статьи ранее докладывались автором на двух конференциях [6, 7].

Обзор задач для сетей с линейными усилениями

Определение сети, потока и псевдопотока

Далее, поскольку мы часто будем рассматривать параллельные дуги, удобно формулировать задачи не для графа, а для мультиграфа. Будем называть *мультиграфом* набор (L, M, s, e), где:

- L – произвольное конечное множество, называемое *множеством вершин*;

– M – произвольное конечное множество, называемое *множеством дуг*;
– $s: M \to L$ – функция, сопоставляющая каждой дуге ее начальную вершину;
– $e: M \to L$ – функция, сопоставляющая каждой дуге ее конечную вершину.

Сеть с линейными усилениями – это мультиграф $N = (L, M, s, e)$, в котором имеется одна выделенная вершина $s'' \in L$, называемая *стоком*, и для каждой дуги которого $m \in M$ заданы:

– нижняя и верхняя пропускная способность $l_m, c_m \in \boldsymbol{R}_+$;
– усиление $g_m \in \boldsymbol{R}_+$ (рассматриваем только положительные усиления);
– а для каждой вершины $l \in L$ задан максимальный дефицит потока $v(l) \in \boldsymbol{R}$.

Потоком в сети называется функция на дугах $f: M \to \boldsymbol{R}$, удовлетворяющая следующим условиям:

1. $f(m) \in [l_m; c_m]$;
2. $V(x) = \sum_{m: s(m) = x} g_m f(m) - \sum_{m: e(m) = x} f(m) = v(x)$ для всех вершин, кроме стока s''.

Значение $V(x)$ будем называть *излишком* потока f в вершине x. То есть для каждой вершины задана фиксированная величина излишка. Таким образом, каждая вершина x, кроме стока, рассматривается как источник потока, генерирующий его в количестве $v(x)$.

Псевдопотоком называется функция, удовлетворяющая первому условию, во втором же равенство заменяется на неравенство: $V(x) \le v(x)$. Иначе говоря, в каждой вершине по-прежнему генерируется определенное количество потока, но это количество может быть меньше заданного – и вообще, каждая вершина может функционировать как сток, в который поток уходит.

Сеть с линейными усилениями и ценами дуг – это сеть с линейными усилениями, у которой для каждой дуги $m \in M$ задана цена этой дуги $p_m \in \boldsymbol{R}$.

Задача максимизации (псевдо)потока в сети с линейными усилениями: максимизировать (псевдо)поток, входящий в сток, то есть величину $-V_f(s)$.

Разложение потока на сумму элементарных потоков

Увеличивающий путь для (псевдо)потока f в сети N – это такой путь, через который можно дополнительно провести ненулевой поток.

Это эквивалентно тому, что все дуги пути ненасыщены.

Усиление пути – это произведение усилений его дуг. В частности, можно рассматривать усиление g цикла. Если $g > 1$, цикл называется *генерирующим*, если $g < 1$ – *поглощающим*, а если $g = 1$ – *нейтральным*.

Элементарный путь – это путь одного из следующих видов:

1. Путь из источника в сток.
2. Нейтральный цикл.
3. Петля – путь из источника в любую вершину, за которым следует поглощающий цикл.
4. Антипетля – генерирующий цикл, за которым следует путь в сток.
5. Двойная петля – генерирующий цикл, за которым следует путь и поглощающий цикл.

Теорема 1.1 (Гондрана-Майнакса)

Любой поток можно разложить в сумму *элементарных потоков*, т.е., потоков, которые не равны нулю только на элементарных путях (как еще говорят, потоков, проходящих через элементарные пути).

Доказательство

Состоит в последовательном рассмотрении элементарных путей и вычитании элементарных потоков так, чтобы на каждом шаге поток в одной из дуг стал равен 0. Поскольку количество дуг конечно, рано или поздно весь поток станет равен 0. □

Аналогичная теорема верна для псевдопотоков: каждый псевдопоток также можно разложить в сумму элементарных псевдопотоков, с той только разницей, что путь или антипетля могут заканчиваться не только в стоке, но и в любой вершине.

Теорема 1.2 (Онаги)

Существует максимальный поток, который можно разложить в сумму элементарных потоков вида 1 и 4 – через пути и антипетли.

Доказательство

Следует из теоремы Гондрана-Майнакса.□

Следствие (критерий максимальности потока)

Поток максимален тогда и только тогда, когда в сети нет увеличивающих путей и антипетель.

Для сети без генерирующих циклов можно не рассматривать антипетли и двойные петли, так что остаются только простые пути.

Вычитание потока и переразмечивание сети

Для любой сети N можно определить следующие операции, переводящие ее в другую сеть N':

1. Вычитание из сети допустимого потока f: $N' = N - f$. При этом из пропускной способности каждой дуги вычитается значение потока f. Значения усилений и стоимостей остаются теми же самыми.

Если поток был допустимым, очевидно, положительность пропускной способности сохраняется. При этом нулевой поток $f - f = 0$ является допустимым.

Также очевидно, что при вычитании из сети N с допустимым потоком f допустимого потока g поток f-g будет допустимым в сети $N - g$.

2. Переразмечивание с положительными метками вершин k: $N' = N \cdot k$.

Если каждой вершине l присвоена метка $k_l \in \mathbf{R}_+$, то переразмечивание – это преобразование

- усилений по закону $g'_m = g_m k(s(m))/k(e(m))$;
- пропускных способностей по закону $l'_m = l_m/k(s(m))$, $c'_m = c_m/k(s(m))$;
- дефицитов по закону $v'(a) = v(a)/k(a)$;
- потоков по закону $f'(m) = f(m)/k(s(m))$.

Как несложно видеть, множество потоков в переразмеченной сети эквивалентно множеству потоков в исходной сети, а значение потока в стоке s'' уменьшается в $k(s'')$ раз.

Данную операцию можно интерпретировать как увеличение единицы измерения в каждой вершине l в $k(l)$ раз.

Переразмечивание пути используется в алгоритмах нахождения максимального потока, вроде алгоритма Трумпера, использующих разложение потока на сумму элементарных. В этих алгоритмах сперва нужно пустить поток по путям, ведущим в сток и имеющим максимальное усиление. При этом каждой вершине присваивается метка, равная максимальному усилению пути из данной вершины в сток.

Правда, подобные алгоритмы работают лишь в сетях без генерирующих циклов, так что генерирующие циклы нужно сперва удалить – для этого есть алгоритмы, на рассмотрении которых мы не будем останавливаться.

Алгоритмы нахождения оптимального потока

Существуют приближенные алгоритмы полиномиальной временной сложности для задачи о максимальном потоке.

Наиболее эффективный – алгоритм Радзика, находящий ε-оптимальный поток за время $O(m^2 n \log(\varepsilon^{-1}))$ [4]. Примерная схема его работы такова. Это итеративный алгоритм, на каждом шаге которого усиления округляются вниз так, чтобы обеспечить удобное переразмечивание сети. После этого переразмечивания получается модифицированная сеть, максимальный поток в которой легко находится путем удаления генерирующих циклов и разложения на сумму элементарных потоков по алгоритму, аналогичному алгоритму Трумпера. Максимальный поток в модифицированной сети является δ-максимальным псевдопотоком в исходной сети, где $\delta^k = \varepsilon$ (он является допустимым псевдопотоком в исходной сети, т.к. мы округлили все усиления вниз).

На следующем шаге рассматривается остаточная сеть и уже в ней ищется максимальный поток и т.д. Все эти потоки складываются. Можно доказать, что в итоге на k-м шаге получаем δ^k-оптимальный, то есть ε-оптимальный псевдопоток, который затем преобразуется в поток.

Преобразование псевдопотоков в потоки

Утверждение 1.3. Для любого псевдопотока f существует поток f^*, дающий такое же значение в стоке.

Доказательство

Используем теорему Гондрана-Майнакса. Разложим псевдопоток в сумму элементарных псевдопотоков. После этого оставим только слагаемые, соответствующие псевдопотокам через пути и антипетли, ведущие в сток, а остальные отбросим.

Таким образом, останется сумма элементарных псевдопотоков через пути и антипетли. Каждый из них несложно преобразовать в поток.□

Сложность алгоритма преобразования псевдопотока в поток – $O(m^2n)$, где n – количество вершин, m – количество дуг.

Многие алгоритмы нахождения оптимального потока более удобно определить как алгоритмы нахождения оптимального псевдопотока. Затем он преобразуется в поток, который имеет ту же величину – то есть также является оптимальным. Поскольку все известные алгоритмы нахождения оптимального псевдопотока имеют сложность выше, чем $O(m^2n)$, это достаточно эффективно.

Сети с нелинейными усилениями в дугах

Определение сети, потока и псевдопотока

Итак, мы сделали краткий обзор задачи максимизации (псевдо)потока в сети с линейными усилениями в дугах. Обобщим ее теперь на нелинейные усиления.

Определение. *Конечная сеть с нелинейными усилениями* $(((L, M), s, e), s'', g, v, l, c)$ – это конечный ориентированный мультиграф $((L, M), s, e)$, для которого определена одна выделенная вершина $s'' \in L$, называемая *стоком*, а для каждой вершины, кроме стока, задан допустимый дефицит $v: L \to \boldsymbol{R}_+$, и для каждой дуги:

– неотрицательные *нижняя и верхняя пропускная способность* l, c: $M \to \boldsymbol{R}_+$;
– неотрицательная возрастающая непрерывная функция неотрицательного аргумента $g_m: \boldsymbol{R}_+ \to \boldsymbol{R}_+$ – *функция усиления*.

Определение. *Поток* в сети – это функция на дугах $f: M \to \boldsymbol{R}$, удовлетворяющая следующим условиям:

1. $f(m) \in [l_m; c_m]$
2. $V(x) = \sum_{m:\, e(m) = x} g_m(f(m)) - \sum_{m:\, s(m) = x} f(m) \in [0; v(x)]$ для всех вершин, кроме стока s''.

Значение $V(x)$ будем называть *излишком* потока f в вершине x.

Псевдопоток определяется аналогично, только излишек для псевдопотока удовлетворяет неравенству $V(x) \le v(x)$.

Любой поток является псевдопотоком, но не наоборот.

Аналогично сетям с линейными усилениями, рассмотрим задачу максимизации (псевдо)потока $\max_f(-V_f(s''))$. Заметим, что целевая функция в этой задаче непрерывна.

Если все пропускные способности конечны, то множество псевдопотоков F замкнуто и ограничено. Если все нижние пропускные способности $l_m = 0$, то оно еще и непусто, ведь нулевой псевдопоток всегда допустим.

Это значит, что задача нахождения максимального псевдопотока имеет решение – правда, оно может быть не единственным.

Кроме точной задачи, будем рассматривать и приближенную: задачу нахождения приближенно максимального (псевдо)потока с точностью ε.

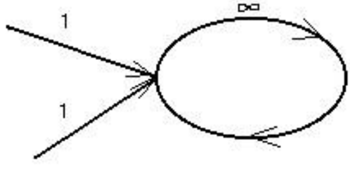
Задачи для сети с линейными усилениями являются частными случаями этих задач, когда усиления заданы линейными функциями:

$$g_m(x) = g_m x.$$

Основные отличия от линейного случая

В нелинейном случае теорема Гондрана-Майнакса не выполняется: псевдопотоки уже нельзя разложить на элементарные слагаемые, даже в том случае, когда имеет место обратимость дуг.

Пример. Рассмотрим следующий граф:



Здесь цифры над дугами означают пропускные способности, усиления в прямых дугах отсутствуют (равны 1), а усиление в петле равно

$$g(x) = \sqrt{x}.$$

Дальше будем называть эту петлю в графе циклом (поскольку это единственный цикл в нем), чтобы не путать с петлей, через которую идут элементарные потоки.

Заметим, что, если x – поток, протекающий через цикл, то при $x < 1$ цикл – генерирующий ($g(x) > x$), при $x = 1$ – нейтральный ($g(x) = x$), а при $x > 1$ – поглощающий ($g(x) < x$). При добавлении в вершину излишка потока величиной y он полностью поглощается при проходящем через цикл потоке x, если

$$y + \sqrt{x} = x,$$

то есть

$$y = x - \sqrt{x}$$

или (учитывая, что $x > 0$)

$$x^2 - (2y+1)x + y^2 = 0.$$

Рассмотрим теперь поток, равный $f = 1$ в прямых дугах – то есть $y = 1+1=2$ – и $f = x = 4$ в цикле. Несложно видеть, что при этом в вершине, из которой выходит цикл, нет излишков: величина входящего потока равна $\sqrt{4} + 1 + 1 = 4$, исходящего – 4. То есть это именно поток, а не псевдопоток.

Предположим, что его можно разложить на элементарные потоки. Очевидно, это разложение должно включать в себя хотя бы 2 петли плюс, возможно, нейтральный цикл. При этом, чтобы работало поглощение, в каждой петле должно быть $x > 1$, а значит, общее количество петель – 2 или 3 (ведь в сумме должно получиться 4). Значит, хотя бы в одной петле должно быть $y = 1$, то есть $x = (3 + \sqrt{5})/2 > 2,5$. Следовательно, 3 петли уже не получится, а получится только 2 (и никакого нейтрального цикла). Но тогда во второй петле тоже будет $x > 2,5$, а в сумме должно получиться больше 4 – противоречие.

Это противоречие вызвано тем, что из-за вогнутости функции усиления величина потока в цикле убывает медленнее, чем величины потоков во входящих в него дугах.

Таким образом, нелинейную задачу нельзя решать как линейную, последовательным добавлением потоков через элементарные пути. Для решения нелинейной задачи мы далее предложим другие методы.

Виртуальные сети и теорема о достижении максимума

Теорема 2.1 (о достижении экстремального значения оптимальным псевдопотоком). Если усиление в дуге m можно записать в виде

$$g_m = \max(g_{1m}, \ldots, g_{km}),$$

то существует максимальный псевдопоток, совпадающий с максимальным псевдопотоком в одной из k вспомогательных задач, в которых усиление в дуге m заменяется на g_{jm}.

Доказательство. Для максимального псевдопотока f в дуге m, где его значение равно $f(m)$, максимум достигается на одной из функций g_{jm}. Докажем, что f является допустимым и максимальным в j-й вспомогательной задаче. Допустимость следует из соотношений, определяющих псевдопоток: в них в j-й задаче стоит усиление g_{jm}. Максимальность – из того, что при переходе к j-й задаче функция усиления разве что уменьшается, а следовательно, множество псевдопотоков сужается. Поэтому, если псевдопоток был максимальным в исходной задаче, он будет максимальным и после сужения допустимого множества. □

Теорему о достижении экстремального значения максимальным псевдопотоком можно интерпретировать так. Рассматривается "виртуальная" сеть, в которой вместо одной дуги m – k "виртуальных" параллельных дуг $m_1, \ldots, m_k$, но использовать можно только одну из них. Тогда максимальный псевдопоток в исходной сети с усилением $g_m = \max(g_{1m}, \ldots, g_{km})$ равен максимальному потоку в «виртуальной» сети с усилениями $g_{1m}, \ldots, g_{km}$, в которой псевдопоток течет по одной из «виртуальных» дуг.

И наоборот, задачу максимизации псевдопотока в «виртуальной» сети можно аналогично свести к задаче в обычной сети, объединяя усиления с помощью функции максимума.

Об отбрасывании пропускной способности или ее уточнении

В задаче с линейными усилениями важную роль играют как усиления, так и пропускные способности. Но оказывается, что в задаче с нелинейными усилениями верхние пропускные способности можно отбросить:

Утверждение 2.2 (о роли пропускной способности). Множество максимальных (псевдо)потоков в сети N с пропускными способностями c_m совпадает с множеством максимальных (псевдо)потоков в сети N^* с функциями усиления

$$g^*_m(x) = g_m(\min(x, c_m))$$

и бесконечными верхними пропускными способностями.

Доказательство. Докажем, что каждый (псевдо)поток в сети N является (псевдо)потоком и в сети N^*. Действительно, пусть f – (псевдо)поток в сети N. Тогда для любой дуги m получаем

$$f(m) \leq c_m \Rightarrow g^*_m(f(m)) = g_m(f(m)).$$

Докажем теперь, что каждому (псевдо)потоку в сети N^* соответствует эквивалентный (псевдо)поток в сети N. Действительно, если f^* – (псевдо)поток в сети N^*, то (псевдо)поток $f = \min(f^*, c)$ корректен и дает не меньшее значение в стоке. □

Впрочем, пропускные способности играют важную роль в эвристических алгоритмах, позволяющих уменьшить число вариантов при переборе. Поэтому лучше не приписывать им бесконечные значения. Наоборот, важно уточнить пропускные способности – точнее, ограничения сверху на потоки в дугах – так, чтобы эти ограничения были как можно меньше. Тогда придется перебирать меньше вариантов.

Для уточнения пропускных способностей полезно также уточнить для каждой вершины x верхние и нижние ограничения на (псевдо)поток, выходящий из нее:

$$L_x \leq \sum_{m:\, s(m) = x} f(m) \leq C_x.$$

Верхние ограничения пропускных способностей дуг, входящих в вершину x, позволяют уточнить верхнее ограничение C_x:

(1) $$C_x \leftarrow \min(C_x,\ V(x) + \sum_{m:\, e(m) = x} g_m(c_m))),$$

где стрелочка ← означает, что верхнему ограничению присваивается новое значение (если изначально ему не было присвоено значение, можно считать, что это +∞).

Соответственно, нижние ограничения пропускных способностей дуг дают нижнее ограничение L_x на исходящий поток (но нижние ограничения на исходящий псевдопоток, разумеется, равны 0):

$$L_x \leftarrow \max(L_x,\ \sum_{m:\, e(m) = x} g_m(l_m)))$$

(если изначально ему не было присвоено значение, можно считать, что это -∞).

А дальше на основе ограничений L_x, C_x мы можем уже подкорректировать верхние и нижние пропускные способности в каждой дуге m: $s(m) = x$, выходящей из вершины x. Пусть

$$L'_x = \sum_{m:\, s(m) = x} l_x$$

$$C'_x = \sum_{m:\, s(m) = x} c_x.$$

Тогда в дугу m мы сможем переслать не больше $C_x - (L'_x - l_m) = C_x - L'_x + l_m$ единиц (псевдо)потока, поскольку как минимум $L'_x - l_m$ единиц будет отправлено в другие дуги. Таким образом, верхней пропускной способности c_m можно присвоить новое значение, так что задача останется эквивалентной:

(2) $$c_m \leftarrow \min(c_m, l_m + (C_x - L'_x)).$$

А при $C_x < L'_x$ сразу ясно, что задача не имеет решения.

Впрочем, если нижние пропускные способности в исходящих из x дугах равны 0, то $L'_x = 0$ и задача всегда имеет решение – но пересчет верхних пропускных способностей по-прежнему имеет смысл.

Если речь идет не о псевдопотоках, а о потоках, можно аналогично скорректировать нижние пропускные способности в исходящих дугах на основе того, что в них должно быть направлено общее количество потока не меньше L_x. Но для псевдопотоков нижнее ограничение L_x не имеет смысла, поэтому скорректировать нижние пропускные способности таким образом невозможно.

Таким образом, общий эвристический алгоритм корректировки верхних пропускных способностей для псевдопотоков выглядит так:

1. Для каждой вершины x посчитать верхние ограничения C_x на исходящие из нее псевдопотоки по формуле (1)
2. Для каждой вершины x пересчитать верхние пропускные способности c_m в исходящих из нее дугах по формуле (2). При этом, если в каких-то исходящих дугах есть нижние пропускные способности, может оказаться, что задача не имеет решения. Тогда на этом шаге алгоритм останавливается.
3. Поскольку мы пересчитали верхние пропускные способности в некоторых дугах, выполняем вторую итерацию: снова пересчитываем верхние ограничения C_x в вершинах, в которые входят эти дуги.
4. И так далее.

Если в графе нет контуров, то алгоритм можно остановить через n итераций, поскольку при этом каждая вершина повлияет на все вершины, в которые ведут пути из нее. Если есть контуры, возможно косвенное влияние вершины x на саму себя. При этом, поскольку функции усиления g_m нелинейны, оценить аналитически это влияние затруднительно. Поскольку алгоритм эвристический и предназначен для примерного уточнения множества вариантов, нет смысла прокручивать его много раз, добиваясь сходимости – достаточно, опять же, повторить процесс уточнения n итераций.

В итоге мы заодно получим и верхнюю оценку на количество псевдопотока, выходящего из стока $C_{s''}$. Она позволяет нам оценить сверху и величину псевдопотока в стоке, то есть целевую функцию основной задачи. Эту оценку мы далее будем использовать в методе ветвей и границ.

Теорема о композиции дуг и решение задачи для вогнутых кусочно-линейных усилений

Итак, в случае нелинейных усилений при нахождении оптимального потока нельзя воспользоваться методом его разложения на элементарные потоки. Мы воспользуемся другим методом: разложением дуг графа на параллельные с соответствующим упрощением функций усиления. При этом оказывается, что сети можно агрегировать, объединяя множества вершин и множества дуг.

Параллельное соединение дуг

Параллельное соединение дуг – это сеть с 2 вершинами s', s'' (источником и стоком), соединенными n дугами $M = \{1, ..., n\}$, в которых заданы нижние пропускные способности $l_1, ..., l_n$, верхние пропускные способности $c_1, ... , c_n$ и функции усиления $g_1, ..., g_n$. Также задано ограничение на дефицит потока в источнике:

$$V_f(s') \leq V.$$

Максимальный поток в такой сети – это решение задачи

$$\max (g_1(x_1) + ... + g_n(x_n)) \tag{3}$$
$$l_i \leq x_i \leq c_i$$
$$x_1 + ... + x_n \leq V$$

Очевидно, если V < c_1 + ... + c_n, второе неравенство можно заменить на равенство

x_1 + ... + x_n = V,

ведь от увеличения потока так, чтобы это равенство выполнилось, сумма усилений разве что вырастет.

Иначе, если V ≥ c_1 + ... + c_n, решение задачи тривиально:

x_1 = c_1

x_2 = c_2

.

.

.

x_n = c_n.

Эту задачу можно решать методом динамического программирования. Вводя функцию h_i для каждой дуги i, получим рекуррентные соотношения:

$h_n(x) = \max_{y \in [l(n);\, x]}\} g_n(\min(y, c_n)),\ x \ge l_n$
$h_{n-1}(x) = \max_{y \in [l(n-1);\, x-l(n)]}(g_{n-1}(\min(y, c_{n-1})) + h_n(x-y)),\ x - y \ge l_n \;\Rightarrow\; x \ge l_{n-1} + l_n$
.
.
.
$h_i(x) = \max_{y \in [l(i);\, x-l(n)-l(n-1)-\ldots-l(i+1)]}(g_i(\min(y, c_i)) + h_{i+1}(x-y)),\ x \ge l_i + l_{i+1} + \ldots + l_n$
.
.
.
$h_0 = h_1(V)$.

Разумеется, задача имеет решение, только если $V \ge l_1 + \ldots + l_n$. Величина h_0 – это и есть искомое максимальное значение.

При отсутствии пропускных способностей получаем просто

$$h_i = g_i \circ g_{i+1} \circ \ldots \circ g_n,$$

где ∘ – конволюция функций. Пропускные способности немного усложняют эту формулу, сужая отрезки, на которых ищутся максимумы.

Решение этой задачи для различных классов функций усиления g_i обладает рядом полезных свойств:

1. Если все g_i – вогнутые функции, то h – вогнутая функция. Это следует из того, что конволюция вогнутых функций вогнута.
2. Если все g_i – кусочно-линейные функции из $O(k)$ «кусочков», то h – также кусочно-линейная функция из $O(k^n)$ кусочков. Это следует из свойств интегрирования идемпотентных рациональных функций [2].
3. Если все g_i – вогнутые кусочно-линейные функции из $O(k)$ «кусочков», то h – также вогнутая кусочно-линейная функция из $O(kn)$ кусочков. Вогнутость следует из п.1, а количество «кусочков» – из того, что каждый из «кусочков» исходных функций сможет участвовать в сумме не более, чем 1 раз.

Теорема о композиции для параллельных дуг

Рассмотрим операцию *агрегирования параллельных дуг* сети, определяемую так. Дуги $m_1, \ldots, m_k$ между вершинами a и b объединяются в одну агрегированную дугу $m_{1,2,\ldots,k}$, а их усиления $g_{m1}, \ldots g_{mk}$ преобразуются в усиление h по правилу: h – это решение задачи (3).

Теорема 3.1 (о композиции потоков для параллельных дуг). Каждому максимальному псевдопотоку в исходной сети соответствует максимальный псевдопоток в агрегированной сети, и наоборот.

Доказательство. Пусть f – максимальный псевдопоток в исходной сети. Рассмотрим псевдопоток f' в агрегированной сети, определяемый так. Агрегированной дуге $m_{1, 2, \ldots, k}$ соответствует суммарный псевдопоток

$$f'(m_{1, 2, \ldots, k}) = f(m_1) + f(m_2) + \ldots + f(m_k),$$

а всем остальным дугам – такие же значения, как для потока f.

Псевдопоток f' окажется допустимым в агрегированной сети, если мы переопределим потоки $f(m_1), \ldots, f(m_k)$ так, чтобы суммарный псевдопоток, входящий в вершину b

$$g_1(f(m_1)) + g_2(f(m_2)) + \ldots + g_k(f(m_k))$$

достигал максимального значения $h(f'(m_{1, 2, \ldots, k}))$ – решение задачи (3). При этом для исходного псевдопотока f значение, входящее в вершину b, от перехода к максимуму в задаче (3) разве что увеличится, а значит, псевдопоток останется допустимым. А, поскольку это псевдопоток, его величину, выходящую из вершины b, можно оставить неизменной, и он по-прежнему будет максимальным.

Псевдопоток же f' в агрегированной сети, очевидно, невозможно сделать больше, поскольку ему соответствует только один поток f в исходной сети: с разложением в агрегированной дуге на такую сумму псевдопотоков $f(m_1) + f(m_k) + \ldots + f(m_k)$, на которой достигается максимум в задаче (3). □

Агрегирование линейных усилений

Теорему о композиции потоков для параллельных дуг можно применять двояко: во-первых, упростить граф, агрегировав его и уменьшив таким образом количество дуг. Во-вторых, упростить функции усиления, дезагрегировав граф – разложив каждую дугу на дуги с более простыми функциями усиления.

В частности, если функции усиления линейные: $g_i(x) = g_i x$, где $g_1, \ldots, g_n$ отсортированы по убыванию, то агрегированная функция при отсутствии пропускных способностей имеет вид

(4) $$h(x) = \min(g_1 x,\ g_2 x + (g_1\text{-}g_2)c_1,\ g_3 x + (g_1\text{-}g_3)c_1 + (g_2\text{-}g_3)c_2,\ \ldots,$$
$$g_n x + \textstyle\sum_{i=1}^{n-1}(g_i\text{-}g_n)c_i,\ \sum_{i=1}^{n} g_i c_i).$$

При их наличии она считается чуть сложнее, но также оказывается вогнутой кусочно-линейной.

Но можно и наоборот: дезагрегировать вогнутое кусочно-линейное усиление в линейные! Пусть в дуге задано усиление

$$g(x) = \min(g_1 x + b_1, \ldots\ g_n x + b_n, b_{n+1}),$$

где коэффициенты $g_1, \ldots, g_n$ отсортированы в порядке убывания. Коэффициент b_{n+1} введен для удобства и может быть равен $+\infty$.

Из условия неотрицательности усилений и возрастания функции $g(x)$ следует, что $b_{n+1} > b_n > b_{n-1} > \ldots > b_1 \geq 0$. Если $b_1 > 0$, то можно увеличить допустимый дефицит в вершине на b_1, заменить все b_i на $b_i - b_1$ и далее считать, что $b_1 = 0$. После этого дугу можно разложить на n параллельных дуг с усилениями $a_1, \ldots, a_n$ и пропускными способностями, которые находятся по рекуррентным формулам

$$
\begin{aligned}
& c_1 = b_1/(g_1 - g_2) \\
& c_2 = (b_2 - (g_1 - g_3)c_1)/(g_2 - g_3) \\
& . \\
& . \\
& . \\
& c_i = (b_i - (g_1 - g_{i+1})c_1 - \ldots - (g_{i-1} - g_{i+1})c_{i-1})/(g_i - g_{i+1}) \\
& . \\
& . \\
& . \\
& c_n = (b_{n+1} - g_1c_1 - \ldots - g_{n-1}c_{n-1})/g_n.
\end{aligned} \tag{5}
$$

Наличие нижней пропускной способности l меняет эту формулу незначительно: часть усилений (самых больших) $b_1, \ldots, b_k$ просто обрезается, поскольку по дугам с этими усилениями поток течет обязательно, чтобы достичь величины l. А для $(k+1)$-й дуги устанавливается нижняя пропускная способность $l - c_1 - \ldots - c_k$.

То, что все пропускные способности получатся положительными, следует из условий существенности слагаемых идемпотентного полинома, представляющего вогнутую кусочно-линейную функцию.

Решение задачи для вогнутых кусочно-линейных усилений

Мы выяснили, что агрегирование линейных усилений в параллельных дугах дает вогнутое кусочно-линейное усиление, и наоборот: любую вогнутую кусочно-линейную функцию $g(x)$ можно представить как решение задачи (3) с линейными усилениями $g_i(x_i) = g_ix_i$ и дефицитом в источнике $V = x$. Это делается по формулам (5).

Таким образом, при решении задачи о максимальном псевдопотоке дугу с вогнутым кусочно-линейным усилением можно превратить в набор эквивалентных дуг с линейными усилениями.

Аналогично, всю сеть с вогнутыми кусочно-линейными усилениями можно дезагрегировать в сеть с линейными усилениями, пользуясь формулами (5).

Аналогичный метод для задачи о потоке минимальной стоимости был описан еще в [1].

В этом случае можно использовать все упомянутые алгоритмы максимизации потока. Это потребует времени порядка $O(F(km, n))$, где $F(m, n)$ – сложность соответствующего алгоритма для линейных усилений, k – количество «кусочков» в функциях усиления. Сложность самой процедуры дезагрегирования равна всего лишь $O(km)$, так что она несущественно влияет на сложность всего алгоритма.

Решение задачи для непрерывных кусочно-линейных усилений

Полупереборный алгоритм для непрерывных кусочно-линейных усилений

Как известно, любая непрерывная кусочно-линейная функция $g(x)$ представима в виде

$$g(x) = \max(g^1(x), g^2(x), \ldots, g^k(x)), \tag{6}$$

где $g^i(x)$ – вогнутые кусочно-линейные функции.

Таким образом, воспользовавшись теоремой о достижении экстремального значения, мы получаем полупереборный метод нахождения максимального потока в сети с непрерывными кусочно-линейными усилениями. Метод состоит в том, что все усиления представляются как максимумы вогнутых кусочно-линейных усилений. После этого производится перебор по всем аргументам максимумов, и для каждого из них решается задача максимизации псевдопотока в сети с вогнутыми кусочно-линейными усилениями. Таким образом, мы получаем максимальный псевдопоток, который остается лишь преобразовать в поток (см. далее).

Сложность такого полупереборного алгоритма – $O(k^{2m}F(km, n))$, где k – максимальное количество «кусочков» в функциях усиления. В этом случае число аргументов минимума в представлении (6) также не больше k, и представление можно подобрать так, что суммарное число кусочков во всех функциях также не больше k.

На первый взгляд, алгоритм неэффективен. Но он включает в себя как частные случаи классические алгоритмы для сетей с линейными усилениями – следовательно, в этом направлении его эффективность вряд ли поддается существенному повышению.

Можно ли существенно сократить количество вариантов при переборе? Да, если использовать ряд эвристических алгоритмов.

Эвристические алгоритмы для сокращения числа вариантов при переборе

Перебор можно сократить, если при рассмотрении *j*-го варианта для дуги *m* принимать нижнюю пропускную способность l_m за минимальный *x* такой, что $g^j_m(x) = \max_l g^l_m(x)$, а верхнюю – за максимальный *x* такой, что $g^j_m(x) = \max_l g^l_m(x)$.

Для этого нужно уметь искать верхние и нижние ограничения на *x* в выражении (6). Они получаются из решения системы неравенств относительно *x*:

$g^j_m(x) \geq g^1_m(x)$

.

.

.

$g^j_m(x) \geq g^{j-1}_m(x)$

$g^j_m(x) \geq g^{j+1}_m(x)$

.

.

.

$g^j_m(x) \geq g^k_m(x)$

Пусть вогнутая кусочно-линейная функция g^l_m задана как минимум линейных функций

(7) $$g^l_m(x) = \min(a^{l1}_m x + b^{l1}_m, \ldots, a^{lk}_m x + b^{lk}_m).$$

Тогда в левой части каждого неравенства стоит минимум из набора функций, а если минимум больше чего-то, то и каждая из его компонент больше. Следовательно, получаем для каждого $i = 1, \ldots, k_{lm}$ систему неравенств

$a^{ji}_m x + b^{ji}_m \geq g^1_m(x)$

.

.

.

$a^{ji}_m x + b^{ji}_m \geq g^{j-1}_m(x)$

$a^{ji}_m x + b^{ji}_m \geq g^{j+1}_m(x)$

.

.

.

$a^{ji}_m x + b^{ji}_m \geq g^k_m(x)$

Неравенство же вида

$$a^{ji}_m x + b^{ji}_m \geq g^l_m(x)$$

то есть

$$a^{ji}_{m}x + b^{ji}_{m} \geq \min(a^{l1}_{m}x + b^{l1}_{m}, \dots, a^{lk}_{m}x + b^{lk}_{m})$$

решается просто. Оно верно для тех x, для которых выполняется хотя бы одно из линейных неравенств. Решение линейного неравенства – пустое множество, все множество вещественных чисел $\boldsymbol{R}$ или луч. Объединяя пустые множества, множества $\boldsymbol{R}$ или лучи, получаем в итоге либо множество $\boldsymbol{R}$, либо луч, либо пару лучей $(-\infty; a] \cup [b; +\infty)$. Впрочем, лучи, пары лучей или множества $\boldsymbol{R}$, если их пересечь с отрезком $[l_m; c_m]$, определяемым нижней и верхней пропускной способностью дуги, сразу превращаются в отрезки или пары отрезков.

Правда, при пересечении пар отрезков может получиться объединение отрезков, число которых будет расти в геометрической прогрессии. Но нам не нужно знать все эти отрезки – достаточно знать лишь их общую верхнюю и нижнюю границу. А она для каждого пересечения считается легко.

Таким образом, для j-го варианта для дуги m мы можем посчитать верхнее и нижнее ограничение на x

$$l^{j}_{m} \leq x \leq c^{j}_{m}$$

за время порядка $O(k^3)$, где k – максимальное число компонент в представлении вида (6) или (7).

Вообще, мы можем для каждого из вариантов для каждой из дуг посчитать эти ограничения еще в самом начале решения задачи. Это можно сделать за время порядка $O(k^4m)$.

Дальше мы на время рассмотрения данного (j-го) варианта устанавливаем это верхнее и нижнее ограничение в качестве новых пропускных способностей:

$$l_m \leftarrow l^{j}_{m};$$

$$c_m \leftarrow c^{j}_{m}.$$

Получив новые пропускные способности дуги, мы пересчитываем пропускные способности всех дуг, в которые можно добраться из нее, за время порядка $O(mn)$ (итеративный алгоритм уточнения пропускных способностей всех дуг повторяется n раз).

Наконец, новые пропускные способности дуг, в свою очередь, позволяют нам ограничить количество рассматриваемых вариантов для каждой дуги m'. Действительно, для каждого из вариантов j' для этой дуги надо просто посмотреть, пересекается ли отрезок $[l^{j'}_{m'}; c^{j'}_{m'}]$ с нашим текущим отрезком $[l_{m'}; c_{m'}]$. Если не пересекается, значит, этот вариант для дуги m' невозможен.

Таким образом, получаем метод улучшенного перебора путем обхода дерева вариантов, в котором возможные значения одних переменных ограничивают множество возможных значений других переменных.

При этом, если у нас есть оценка сверху для излишка потока в стоке - $V_f(s'')$, тем самым мы получаем верхние оценки для применения метода ветвей и границ. Дерево вариантов для этого метода строится так же.

Задача о рюкзаке как частный случай. Обобщение метода ветвей и границ на общую задачу

Рассмотрим классическую задачу о рюкзаке: максимизировать по k сумму

$$k_1w_1 + \ldots + k_mw_m,$$

где $k_i \in \boldsymbol{Z}_+$ (неотрицательные целые числа) и

$$k_1c_1 + \ldots + k_mc_m \le V.$$

Сделаем замену $x_i = k_ic_i$, тогда $k_i = x_i / c_i$. Получаем задачу максимизации по x суммы

$$w_1x_1 / c_1 + \ldots + w_mx_m / c_m$$

при условиях

$x_1 + \ldots + x_m \le V$

$x_i \ge 0$

$x_i / c_i \in \boldsymbol{Z}$.

Чтобы избавиться от последнего условия, заметим, что, подставив целую часть $[x_i / c_i]$ вместо x_i / c_i, мы получим при условии целочисленности x_i / c_i то же самое значение целевой функции. Если же x_i / c_i не целочисленно, то взятие целой части делает невыгодным использование дробного значения x_i / c_i – его выгоднее снизить до ближайшего целого. Следовательно, эквивалентная задача максимизации:

$$\max_x (w_1[x_1 / c_1] + \ldots + w_m[x_m / c_m])$$

при условиях

$$x_1 + \ldots + x_m \le V$$

$$x_i \ge 0.$$

Это задача максимизации потока в сети, в которой имеется 2 вершины, а между ними проходит m параллельных дуг. Первая вершина – источник с дефицитом V, вторая – сток. Дуга номер i имеет пропускную способность V. Можно было бы оставить и бесконечную пропускную способность – но тогда усиление в дуге состояло бы из бесконечного числа кусочков. Также в дуге i задана кусочно-линейная (точнее, даже кусочно-постоянная) функция усиления

$$g_i(x) = w_i[x_i / c_i],$$

которая на интервале $[(k-1)c_i; kc_i)$ равна $(k-1)w_i$.

Правда, в такой задаче функции усиления не являются непрерывными: функция g_i терпит разрыв первого рода в точках c_i, $2c_i$, …, V_ic_i, где $V_i = [V/c_i]$, совершая в точке kc_i скачок от $w_i(k-1)$ до w_ik. Но их можно с помощью малого преобразования свести к непрерывным кусочно-линейным, так что задача будет эквивалентной. Для этого достаточно на каждом отрезке $[kc_i - \varepsilon; kc_i]$ поменять функцию усиления на

$$g'_i(x) = w_ix/\varepsilon + w_ik - w_ikc_i/\varepsilon,$$

где $\varepsilon < \min_j c_j / m$ – достаточно маленькое число, чтобы было невыгодно снижать поток в i-й дуге на это значение (все такие снижения не сумеют повысить поток ни в какой другой j-й дуге на величину c_j, а значит, не изменят целевую функцию).

Таким образом, получаем задачу о максимальном потоке в сети с $n=2$ вершинами и m дугами, в которой i-я функция усиления состоит из порядка $2V_i = 2V/c_i$ кусочков и имеет порядка V_i точек выпуклости и V_i точек вогнутости. Каждую из них можно представить в виде

$$f_i(x) = \max(0, \min(w_i, w_ix/\varepsilon - w_ikc_i/\varepsilon), \min(2w_i, w_ix/\varepsilon + w_i - w_ikc_i/\varepsilon), \ldots,$$
$$\min(V_iw_i, w_ix/\varepsilon + (V_i-1)w_i - w_ikc_i/\varepsilon)).$$

Эту задачу можно решить, перебирая в каждой дуге V_i аргументов максимума, где $i=1, \ldots, m$, и решая для каждого случая задачу о максимальном потоке в сети с m дугами, в каждой из которых функция усиления состоит из 2 кусочков. Можно получить эквивалентную сеть с $2m$ дугами и линейными усилениями, в которой найти δ-оптимальный поток алгоритмом Радзика за время $O(m^2 \log(\delta^{-1}))$. По сложности это несущественно отличается от решения задачи о рюкзаке методом полного перебора значений $k_1, \ldots, k_m$.

Правда, для задачи о рюкзаке, как известно, есть более эффективные алгоритмы решения, чем полный перебор. Но многие из них легко обобщаются на задачу о максимальном потоке в сети с непрерывным кусочно-линейными усилениями.

Самое простое улучшение – перебор дерева вариантов, в котором возможные значения одних переменных ограничивают множество возможных значений других переменных. В частности, если уже известны значения $x_1 = k_1^*c_1$, …, $x_j = k_j^*c_j$, то в нашем распоряжении остается лишь величина потока

$$V - k_1^*c_1 - \ldots - k_j^*c_j,$$

а значит, величина k_l должна быть не больше $[(V - k_1^*c_1 - \ldots - k_j^*c_j)/c_l]$.

С точки зрения потоков, это означает, что, рассматривая для x_1 значение $k_1^*c_1$, мы задали в дуге 1 нижнюю пропускную способность $k_1^*c_1$, и аналогично для дуг $2, \ldots, j$. Тем самым, мы ограничили во всех остальных дугах верхнюю пропускную способность величиной $V - k_1^* c_1 - \ldots - k_j^* c_j$, что уменьшает множество возможных рассматриваемых «кусочков», которым может принадлежать значение потоков в этих дугах.

Но это в точности тот эвристический алгоритм ограничения числа вариантов при переборе «кусочков» функций, который мы уже рассмотрели!

Также задачу о рюкзаке можно решать методом ветвей и границ. Но такой же метод мы описали для максимизации потока – он ничем не отличается от метода для задачи о рюкзаке, поскольку дерево вариантов строится так же.

Хороший способ ускорения решения в методе ветвей и границ для задачи о рюкзаке – начать с самых перспективных вариантов. Они в задаче целочисленного линейного программирования, как известно, определяются соответствующей ЗЛП [1]. Приближенное решение задачи о рюкзаке путем решения ЗЛП и округления результата до целого с последующим решением «остаточной» задачи дает жадный алгоритм [1]. Он работает так. Сперва максимизируем такое k_i, для которого максимально отношение w_i / c_i. Затем смотрим возможные значения оставшихся k_j в дереве вариантов, снова выбираем максимальное отношение и т.п. Жадный алгоритм не всегда оптимален, но часто дает решение, близкое к оптимальному, что позволяет отсечь много вариантов в методе ветвей и границ.

С точки зрения потоков, этот метод выглядит так: в каждой дуге i мы аппроксимируем «ступенчатую функцию» усиления $g_i(x)$ линейной функцией:

$$g_i(x) \sim w_i x / c_i.$$

А дальше стараемся в первую очередь пускать больше потока через те дуги, в которых усиление больше.

Этот метод тоже легко обобщается на произвольную задачу о максимальном потоке в сети с непрерывными кусочно-линейными усилениями. Для этого в каждой дуге усиление также аппроксимируется линейным, а дальше мы в первую очередь стараемся пускать большее значение потока через дуги с большими усилениями – то есть рассматриваем для них такие «кусочки», для которых верхние и нижние пропускные способности больше.

Более целесообразно брать в качестве критерия выбора первоначального варианта даже не усиление каждой отдельной дуги m, а произведение усилений по путям, ведущим через эту дугу в сток – то есть метку вершины $s(m)$ в алгоритме Трумпера. Действительно, в этом алгоритме из вершины с наибольшей меткой в первую очередь пускается поток. Таким образом, можно сказать, что выбирая наиболее предпочтительные варианты, мы руководствуемся оптимальным решением в линеаризации первоначальной нелинейной задачи.

Преобразование псевдопотоков в потоки

До этого мы рассматривали задачу максимизации псевдовотоков. Попробуем теперь решить аналогичную задачу для потоков.

Утверждение 4.1. Для каждого псевдопотока существует поток, имеющий такую же величину.

Доказательство. Рассмотрим псевдопоток f. Построим для него поток f', имеющий ту же величину.

Не умаляя общности, будем считать, что все функции усиления удовлетворяют условию $g(0) = 0$. Действительно, если есть такие функции g_m, что $g_m(0) = g_0 > 0$, можно их модифицировать, введя дополнительную дугу, ведущую в $e(m)$, с пропускной способностью g_0, и перейти к функции $g^*_m(x) = g_m(x) - g_0$. Множество потоков и множество псевдопотоков от такой модификации не изменится.

Далее, используя теорему о достижении максимального значения, можно свести псевдопоток в исходной сети к псевдопотоку в виртуальной сети с вогнутыми кусочно-линейными усилениями. Надо просто посмотреть для каждой дуги, в каком именно «кусочке» достигается максимум соответствующей непрерывной кусочно-линейной функции.

Данный псевдопоток, в свою очередь, с помощью теоремы о композиции сводится к псевдопотоку в сети с линейными усилениями. Для последнего же, согласно теореме о разложении (псевдо)потока на элементарные, существует поток, имеющий такую же величину. Далее, используя теорему о композиции и теорему о достижении максимального значения, «сворачиваем» сеть, приводя ее в исходное состояние. Заметим, что всегда можно выбрать такой достаточно малый поток, чтобы при применении теоремы о композиции он остался потоком, а не превратился в псевдопоток. Таким образом, получаем требуемый поток в исходной сети.□

Замечание. Доказательство конструктивно и дает алгоритм преобразования псевдопотока в эквивалентный поток. Несмотря на применение теоремы о достижении максимального значения, этот алгоритм не требует перебора и имеет сложность $O(mnk)$.

Заключение

Итак, формализована задача нахождения максимального потока в сети с нелинейными функциями усиления в дугах. Для нее построен полупереборный алгоритм нахождения максимального потока в сети с непрерывными кусочно-линейными усилениями, основанный на рассмотрении непрерывной кусочно-линейной функции как максимума из вогнутых кусочно-линейных функций.

Показано, что задачу о рюкзаке можно рассматривать как частный случай данной задачи. Метод ветвей и границ для задачи о рюкзаке обобщается на произвольную задачу о максимальном потоке в сети с непрерывными кусочно-линейными усилениями. Жадный алгоритм решения задачи о рюкзаке, позволяющий находить близкие к оптимальным начальные решения, также обобщается на задачу о максимальном потоке.

Но куда больший интерес для практических приложений представляет не задача о максимальном потоке, а задача о потоке минимальной стоимости. В этом случае, нелинейными могут быть не только функции усиления, но и функции стоимости. Впрочем, возможен и важный частный случай: усиления отсутствуют, но имеются нелинейные функции стоимости.

Вероятно, многие приемы, используемые для нахождения максимального потока в сети с нелинейными функциями усиления, можно будет использовать и для нахождения потока минимальной стоимости в сети с нелинейными функциями стоимости. В частности, с непрерывными кусочно-линейными функциями стоимости.

Библиографический список

1. Ху Т. Целочисленное программирование и потоки в сетях. – М.: «Мир», 1973.
2. Bacelli Fr., Cohen G., Olsder J.G., Quadrat J.-P. Synchronization and linearity. Algebra for discrete event systems. – New-York: John Wiley & Sons, 1992.
3. Benchekroun B. A nonconvex piecewise linear optimization problem // Computers Math. Applic. 1991. V. 21. №6/7. P. 77-85.
4. Radzik T. Faster algorithms for the generalized network flow problem. // Mathematics of Operations Research, 23:69-100, 1998.
5. Wayne K.D. Generalized Maximum Flow Algorithms. – A Dissertation Presented to the Faculty of the Graduate School of Cornell University, 1999.
6. Парфенов, А.П. Оптимизация потока в сетях с кусочно-линейными функциями усиления и стоимости // Процессы управления и устойчивость: Труды 38 научной конференции аспирантов и студентов под редакцией Платонова А. В., Смирнова Н. В. – СПб., 2007. – С. 592-598.
7. Парфенов, А.П. Оптимизация потока в сети с нелинейными усилениями // Современные методы теории краевых задач: материалы Воронежской весенней математической школы «Понтрягинские чтения – XVIII». – Воронеж, 2007. – С. 125-126.