

ОПТИМИЗАЦИЯ СЕТЕВЫХ МОДЕЛЕЙ

Сети:

Транспортные, электрические,
коммуникационные

Сетевое представление:

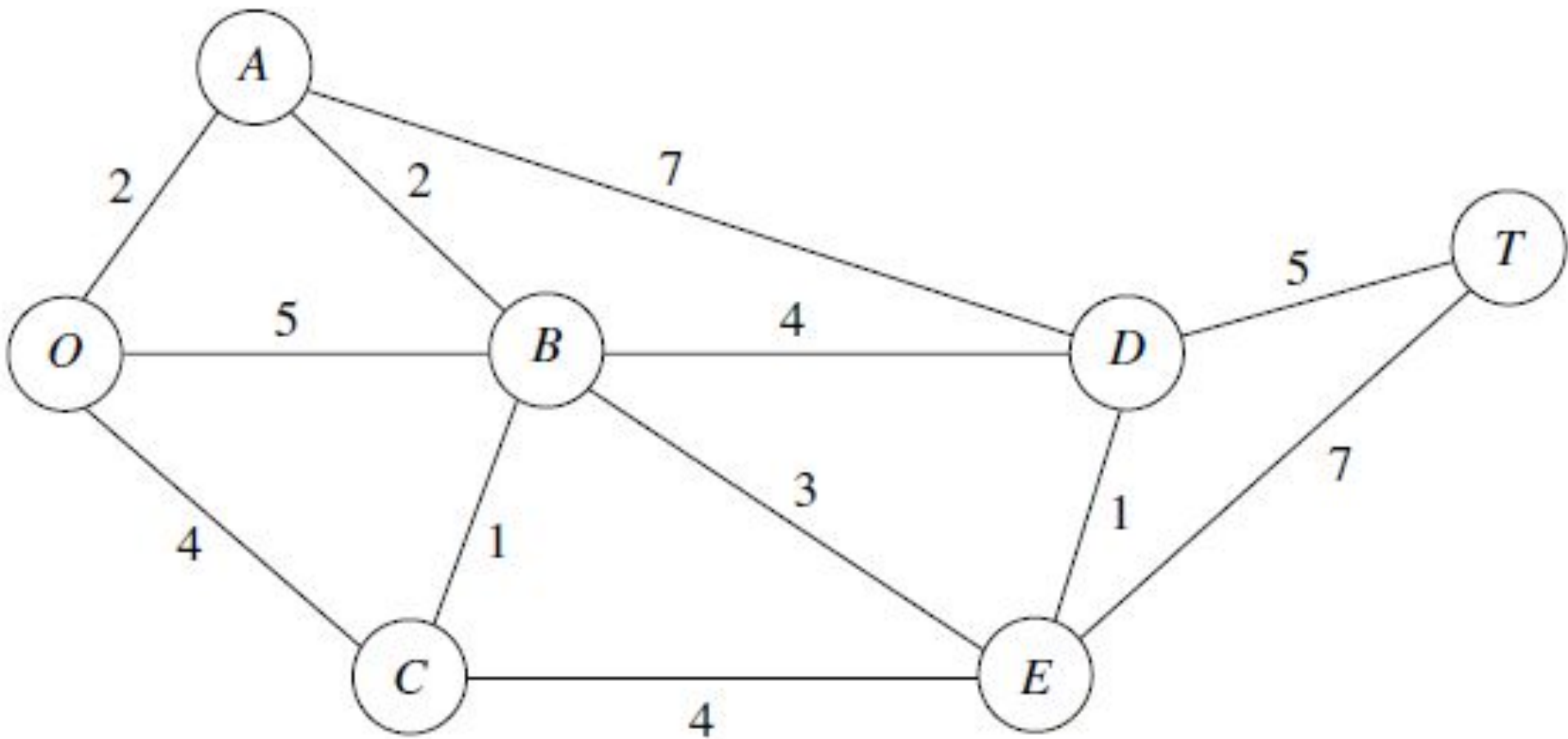
мощное визуальное и
концептуальное средство для
представления отношений компонентов
системы

Используется в

(производство, распределение,
планирование проекта, согласование
места размещения объектов,
управление ресурсами, финансовое

Пример:

Парк Кирова предназначен для экскурсий и пикников. Использование машин запрещено. Предусмотрена система дорог (для велосипедов и др). Точка O (вход), другие буквы - условные положение станций и других объектов. Числа - длина дорог в милях. Самый популярный объект - на станции T . Предположим, что организован транспортный проезд (парковые джипы и др.) для провоза посетителей от входа до



Система дорог для парка
Кирова

Администрация парка сталкивается с тремя проблемами:

- Определить маршрут от входа до станции Т с *наименьшим расстоянием*. (задача нахождения кратчайшего пути)
- Установить телефонную линию под дорогами для обеспечения связи между станциями с *минимальной длиной линии*. (задача минимального связующего дерева)
- *Максимизировать количество поездов / день, не нарушая предельной вместимости* на любой дороге. (задача о максимальном потоке).

(В пик сезона число единиц транспорта, едущего от входа на станцию Т *увеличивается*, поэтому чтобы увеличить количество рейсов / день маршруты

Сеть = *точки, соединенные линиями.*

Точки = **узлы** (вершины) – показаны окружностями.

Линии = **дуги** (звенья, ребра, ветки) - стрелками.

Поток сети:

Если в одном направлении (**ориентированные дуги, $A \rightarrow B$**).

Ориентированная сеть

Если в обоих направлениях (**неориентированные дуги, звенья**).

Неориентированная сеть

Поток сети: разница в двух направлениях потока

Путь = последовательность **дуг**, соединяющих два *узла*

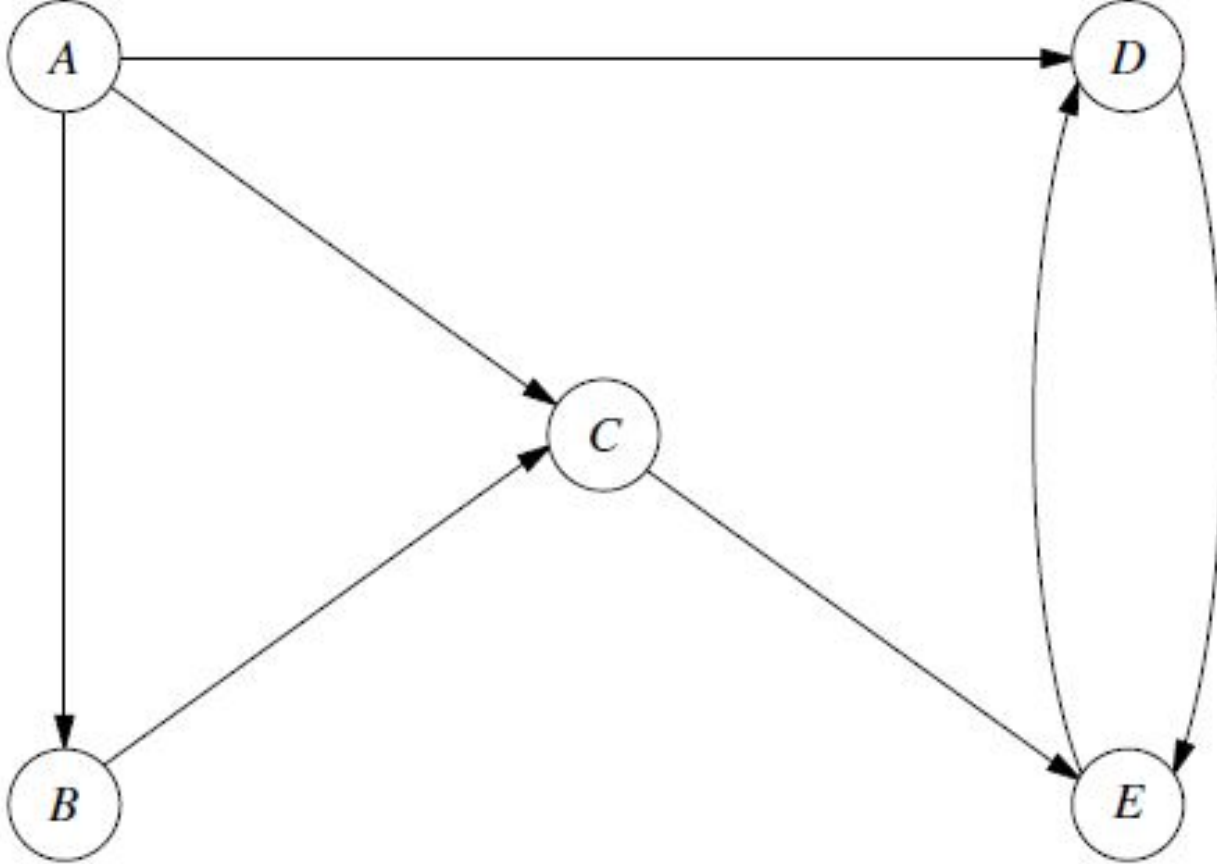
Ориентированный путь=

Последовательность дуг из узла i в узел j с направлением в узел j .

Неориентированный путь=

Последовательность дуг из узла i в узел j с направлением в или из узла j .

Цикл = путь начинается и заканчивается в одном и том же узле.



$(A \rightarrow B \rightarrow C \rightarrow E)$ = ориентированный путь (поток к E допустим). $(B \rightarrow C \rightarrow A \rightarrow D)$ = неориентированный путь $DE-ED$ (ориентированный цикл), $AB-BC-AC$ (неориентированный цикл), $OB-BO$ (не цикл, так как OB и BO не являются четкими дугами как $DE-ED$).

Два узла = связаны, если сеть содержит по крайней мере один *неориентированный* путь между ними.

Связанные сети = связана каждая пара узлов.
 n узлов сети (дуги удалены): растим (создаем) дерево, добавляя одну дугу за раз (между связанными узлами и новым узлом, не связанным ни с чем).

Число связанных узлов = число дуг + 1. Каждая новая дуга создает большее дерево (связанную сеть) *в которой нет неориентированных циклов*. После того, как добавлена дуга ($n - 1$), процесс останавливается, так как дерево соединяет все

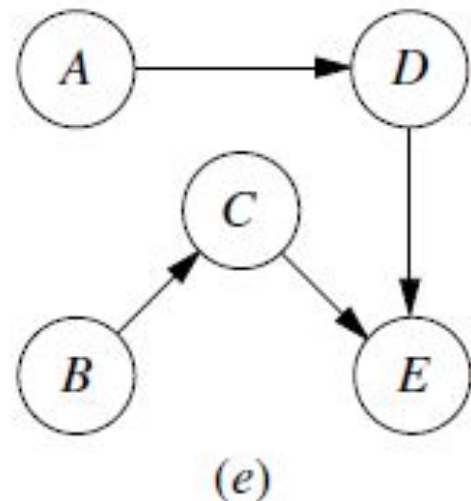
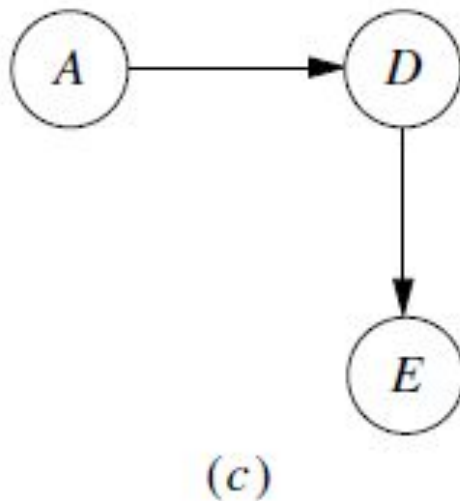
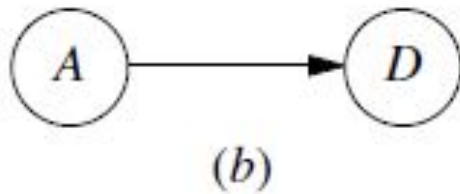
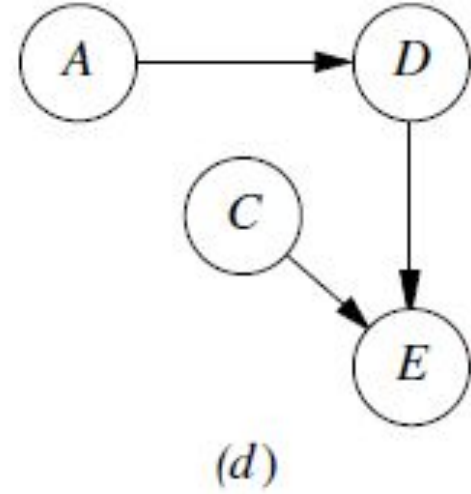
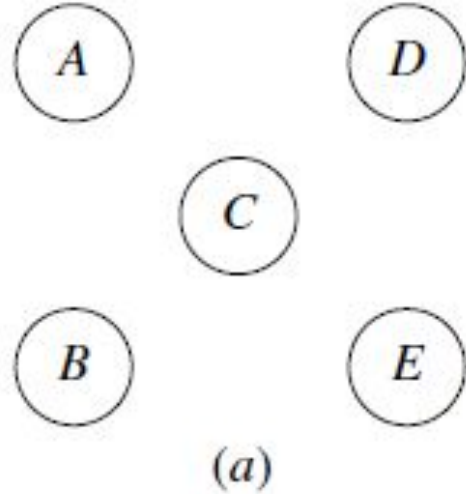
Пропускная способность дуги = *максимальное* значение потока на ориентированной дуге.

Узлы, либо сетевые генераторы потока, либо сетевые поглотители потока, либо ни то, ни другое.

Узел - поставщик(источник) = величина потока из узла превышает величину потока в узел.

Узел-потребитель (приемник) = величина потока в узел превышает величину потока из узла.

Передающий узел (промежуточный узел) =



Пример
связующего
дерева

ЗАДАЧА НАХОЖДЕНИЯ КРАТЧАЙШЕГО ПУТИ

В неориентированной и связанной сети с двумя особыми узлами (отправления и назначения). Ассоциируется с неотрицательным значением каждого звена (расстояние). Цель состоит в том, чтобы найти кратчайший путь (с минимальным общим расстоянием) от пункта отправления до пункта назначения.

Процедура начинается с исходной точки, последовательно определяя кратчайший путь к каждому узлу в порядке возрастания их расстояния от начальной точки, находя решения, когда узел назначения будет достигнут .

Алгоритм нахождения кратчайшего пути:

Цель n -ной итерации: Найти n -ный ближайший к началу узел (повторяем для $n = 1, 2, \dots$. Пока ближайший n -ный узел не будет являться точкой назначения.

Ввод n -ной итерации: $n - 1$ ближайших узлов к началу (найденных на предыдущих итерациях), включая их кратчайший путь и расстояние от начала. (Узлы + начало, называем решенные

Кандидаты для n -го ближайшего узла: Каждый решенный узел непосредственно соединенный звеном с одним или более нерешенным узлом обеспечивает одного кандидата - нерешенный узел с кратчайшим связующим звеном. (Связи обеспечивают дополнительных кандидатов.)

Расчет n -го ближайшего узла: Для каждого решенного узла и его кандидата, сложить расстояние между ними и расстояние кратчайшего пути от пункта отправления до этого решенного узла. Кандидат с таким наименьшим общим расстоянием является n -ным ближайшим узлом (связи обеспечивают дополнительные решенные узлы), и его кратчайшим путем является тот, кто обеспечивает это расстояние.

Решение задачи кратчайшего пути на примере о парке Кирова:

Администрации парка Кирова необходимо найти кратчайший путь от входа в парк (узел O) к основной достопримечательности (узел T) через показанную систему дорог. Применяем вышеупомянутый алгоритм □ результаты приведены в таблице (где связь для второго ближайшего узла позволяет сразу перейти к поиску четвёртого ближайшего следующего узла). Первый столбец (*число итераций*). Второй столбец (*решенные узлы*) для начала текущей итерации после удаления неподходящих значений (те, которые не связаны напрямую с любым нерешенным узлом).

Третья колонка (кандидаты на n -й узел)
(нерешенные узлы с кратчайшим
связующим звеном до решенного узла).
Четвертая колонка (расстояние
кратчайшего пути от пункта отправления
до каждого из этих кандидатов)
(расстояние до решенного узла +
расстояние по звену до кандидата). Пятая
колонка (кандидат с наименьшим таким
расстоянием = n -ный ближайший узел к
началу). Последние две колонки
(информация для этого нового решенного
узла, необходимая для перехода к
последующей итерации) (расстояние
кратчайшего пути от пункта отправления
до этого узла и последняя звено на этом
кратчайшем пути).

<i>n</i>	Solved Nodes Directly Connected to Unsolved Nodes	Closest Connected Unsolved Node	Total Distance Involved	<i>n</i> th Nearest Node	Minimum Distance	Last Connection
1	O	A	2	A	2	OA
2, 3	O A	C B	4 $2 + 2 = 4$	C B	4 4	OC AB
4	A B C	D E E	$2 + 7 = 9$ $4 + 3 = 7$ $4 + 4 = 8$	E	7	BE
5	A B E	D D D	$2 + 7 = 9$ $4 + 4 = 8$ $7 + 1 = 8$	D D	8 8	BD ED
6	D E	T T	$8 + 5 = 13$ $7 + 7 = 14$	T	13	DT

Алгоритм нахождения кратчайшего пути на примере парка Кирова

Ввод для n -ной итерации: даны 5-й и 6-й столбцы для предыдущих итераций, где решенные узлы в 5-м столбце указаны во 2-м столбце для текущей итерации после удаления тех, которые больше не связаны напрямую с нерешенными узлами.

Кандидаты n -го ближайшего узла затем перечислены в третьем столбце для текущей итерации. Расчет n -го ближайшего узла произведен в 4-м столбце, результаты в последних трех столбцах для текущей итерации.

Кратчайший путь от пункта назначения до начала можно отследить по последнему столбцу, $T \square D \square E \square B \square A \square O$ или

$T \square D \square B \square A \square O$. Таким образом, двумя альтернативами кратчайшего пути от пункта отправления до пункта назначения являются $O \square A \square B \square E \square D \square T$ и $O \square A \square B \square D \square T$, с общей протяженностью 13 км по обоим путям.

ЗАДАЧА МИНИМАЛЬНОГО СВЯЗУЮЩЕГО ДЕРЕВА

Задача минимального связующего дерева имеет некоторое сходство с основной версией задачи нахождения кратчайшего пути. В обоих случаях рассматривается неориентированная и связанная сеть, а данная информация включает в себя некоторые меры *положительной* величины (расстояние, стоимость, время и т.д.) заданные для каждого звена. В обеих задачах надо найти сочетание звеньев с *общей длиной, которая будет кратчайшей среди всех сочетаний звеньев*. Для задачи кратчайшего пути (выбранные звенья должны обеспечить путь между пунктами отправления и назначения). Для задачи минимального связующего дерева (выбранные звенья должны обеспечить путь между *каждой парой* узлов).

Минимальное связующее дерево:

1. *Даны узлы сети, но не звенья.* Вместо этого, учитываем потенциальные звенья и положительную длину для каждого (*если она имеется в сети*). (*Альтернативные меры для звеньев включают расстояние, стоимость и время*).

2. Создаем сеть, вставляя достаточно звеньев, чтобы удовлетворить условию, *что между каждой парой узлов должен быть путь*.

3. Цель заключается в удовлетворении этого условия таким образом, чтобы *свести к минимуму общую длину звеньев, вставленных в сеть*.

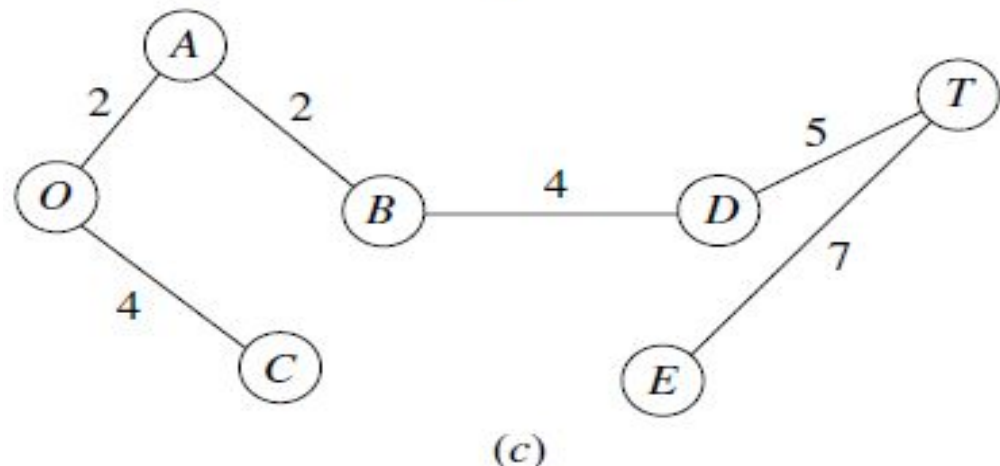
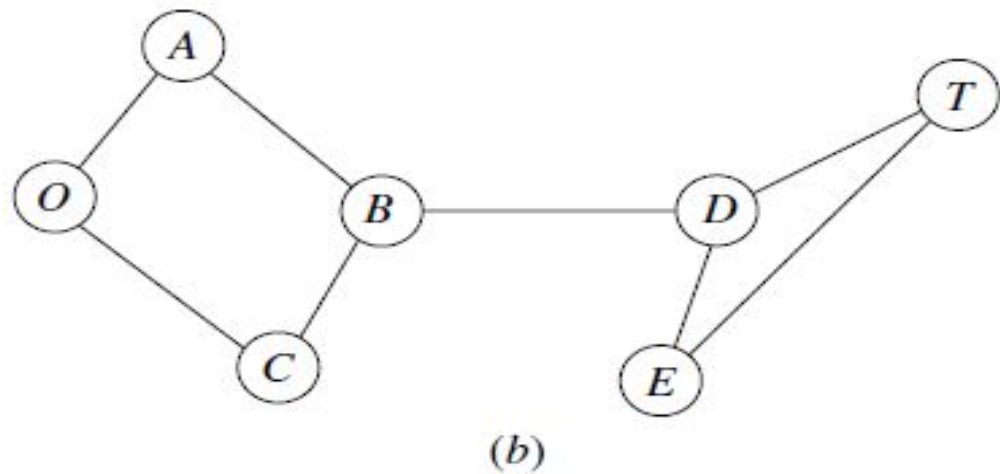
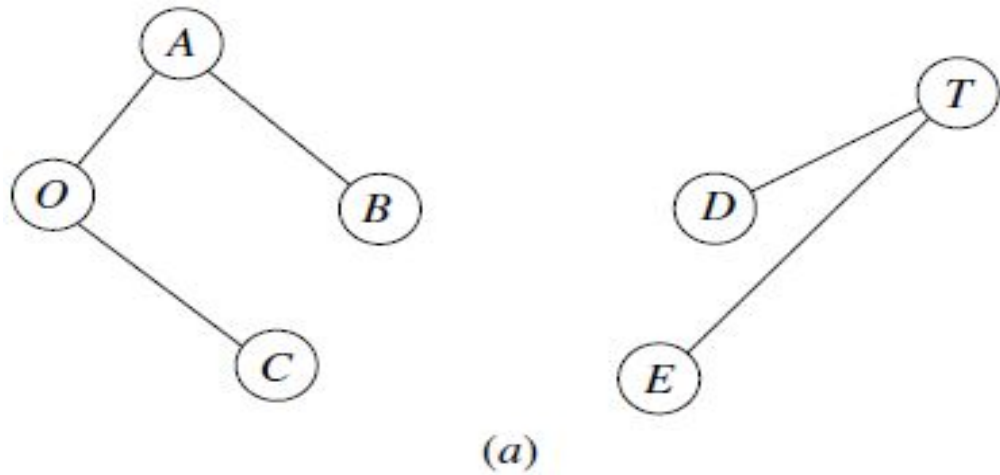
Для сети с n узлами требуется только $(n - 1)$ звено для обеспечения пути между каждой парой узлов. Не должны быть использованы дополнительные звенья, так как это приведет к ненужному увеличению общей длины выбранных звеньев. Нужно выбрать $(n - 1)$ звено таким образом, чтобы получившаяся сеть (только с выбранными звеньями) образует связующее дерево \square задача состоит в нахождении связующего дерева с минимальной общей длиной звеньев.

Показана концепция связующего дерева для задачи о парке Кирова, (а) - *не является покрывающим деревом, поскольку узлы O, A, B, C не связаны с узлами D, E и T. Чтобы их связать, необходимо еще одно звено. Эта сеть состоит из двух деревьев, по одному для каждого из этих двух наборов узлов.*

Звенья в (b) охватывают сеть (т.е., сеть связана), но это не дерево, потому что есть два цикла ($O-A-B-C-O$ и $D-T-E-D$). Она имеет *слишком много* связей. Поскольку задача о парке Кирова имеет $n = 7$ узлов, сеть должна иметь ровно $n - 1 = 6$ звеньев, без циклов, чтобы можно было поставить задачу о связующем дереве. Это условие выполняется в (c), так что *эта сеть является допустимым решением* (со значением общей длины звеньев 24 км) для задачи минимального связующего дерева. (Это решение не является оптимальным, потому что можно построить связующее дерево с общей длиной звеньев только 14 км .)

Применение:

1. Проектирование телекоммуникационных сетей (волоконно-оптические сети, компьютерные сети, выделенные линии телефонных сетей, сети кабельного телевидения).
2. Проектирование легко-нагруженных транспортных сетей, чтобы минимизировать общую величину звеньев (железнодорожные линии, дороги).
3. Проектирование сетей высоковольтных электрических линий электропередачи.
4. Проектирование сетей проводов на электрическом оборудовании (цифровые компьютерные системы), чтобы минимизировать общую длину провода.
5. Проектирование сети трубопроводов для подключения нескольких местоположений



Связующее
дерево для
парка Кирова –
только (с) –
связующее
дерево

Первый этап: выбор самого короткого звена к другому узлу, не думая о влиянии этого выбора на последующие решения.

Второй этап: определение несвязанных узлов, ближайших к любому из этих связанных узлов, а затем добавление соответствующих звеньев в сеть.

Процесс повторяется, пока не связаны все узлы.

Алгоритм минимального связующего дерева :

1. Произвольно выбираем любой узел, соединяем его с ближайшим отдельным узлом.

2. Определяем несвязанные узлы, ближайшие к связанному узлу, соединяем эти два узла.

Повторяем этот шаг до тех пор, пока не связаны все узлы.

3. В случае произвольного образования связей до ближайшего отдельного узла (шаг 1) или ближайшего несвязанного узла (шаг 2), алгоритм все равно должен давать оптимальное решение.

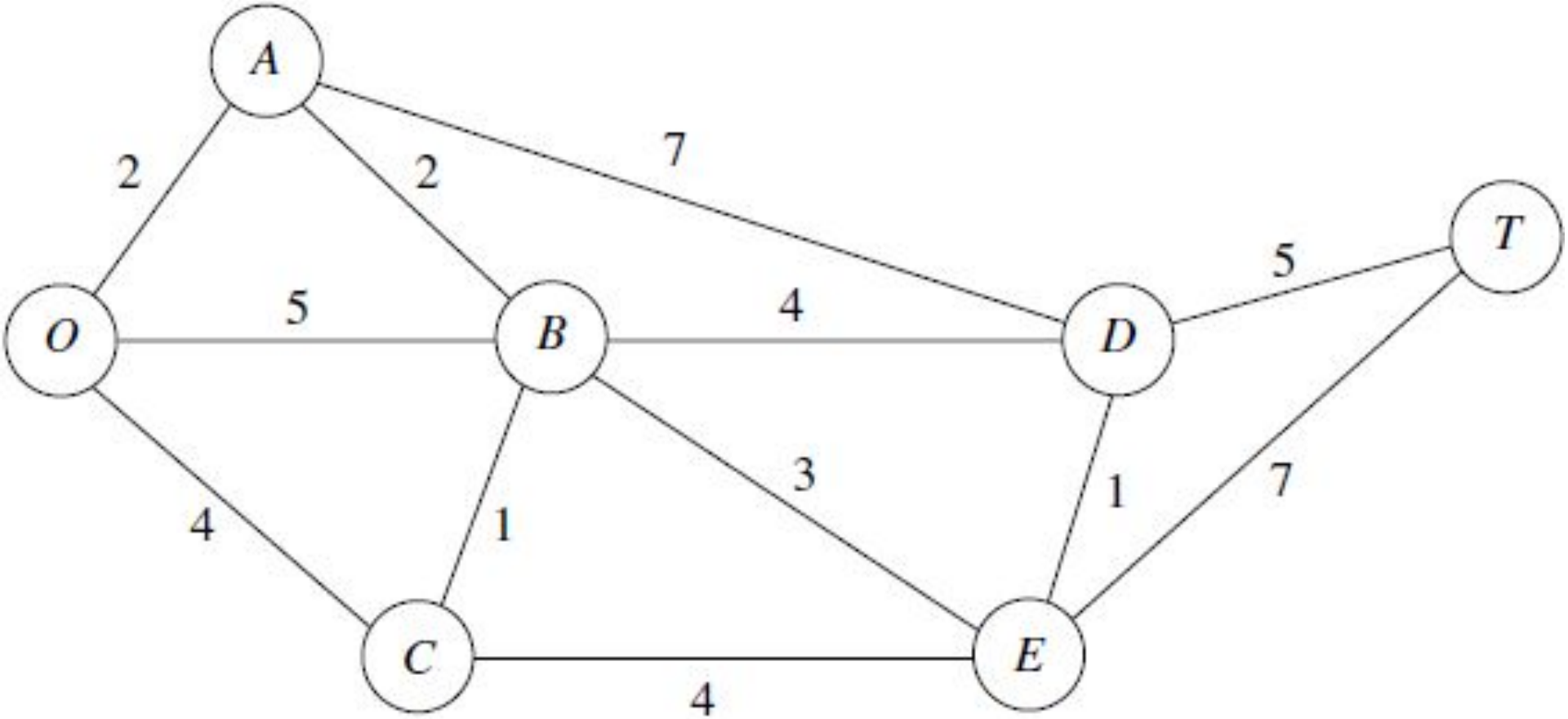
Применение минимального связующего дерева на примере парка Кирова:

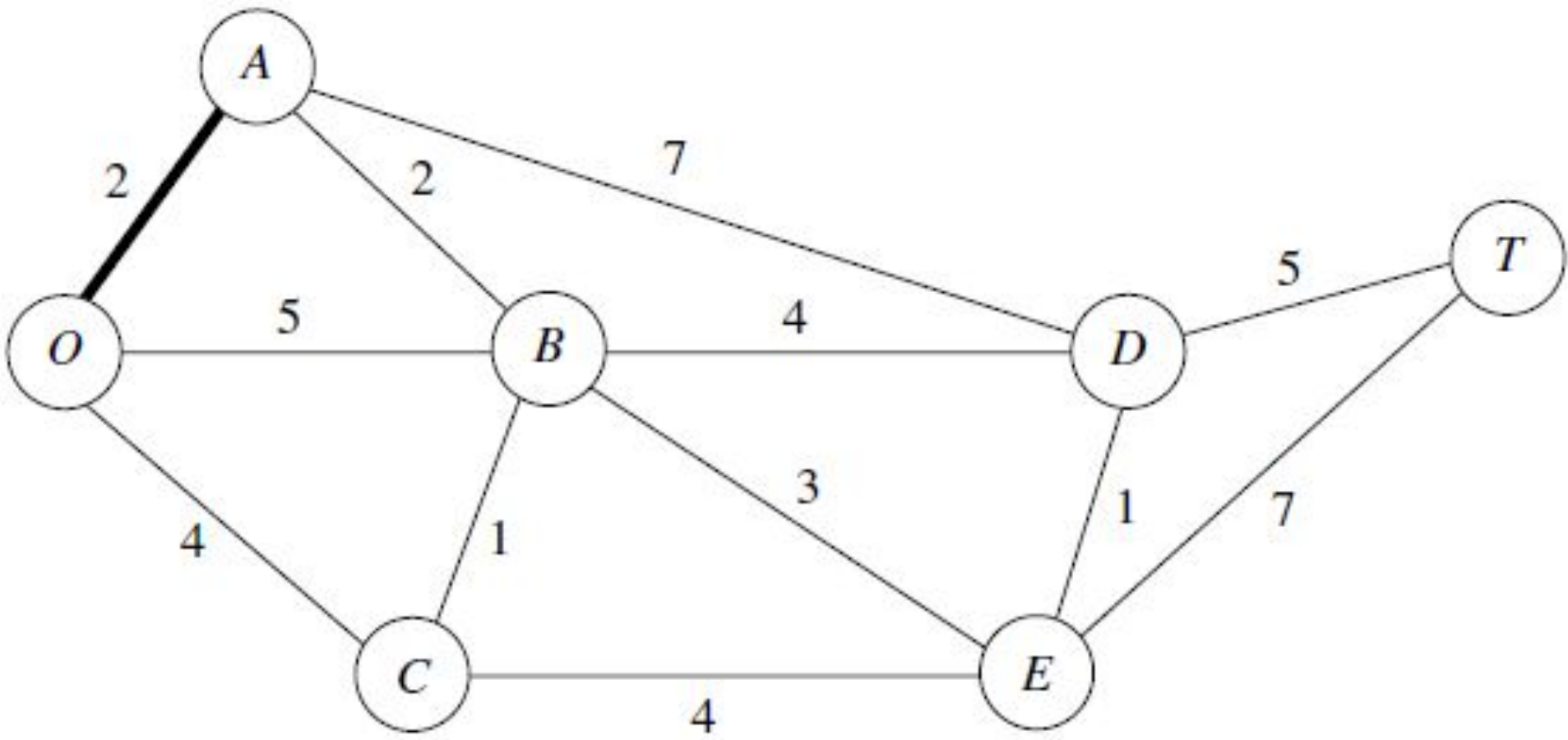
Руководство должно определить, под какими дорогами должны быть проложены телефонные линии для подключения всех станций с минимальной общей длиной линии. Произвольно выберем узел O . Несвязанным узлом, который ближе всего к O является A . Соединить A с O . Несвязанным узлом, который ближе всего к O , или A является B (ближайший к A). Соединить B с A . Несвязанным узлом, который ближе всего к O , A или B является C (ближайший к B). Соединить C с B . Несвязанным узлом, который ближе всего к O , A , B или C является E (ближайший к B). Соединить E с B . Несвязанным узлом, который ближе всего к O , A , B , C или E является D (ближайший к E). Соединить D с E . Единственный

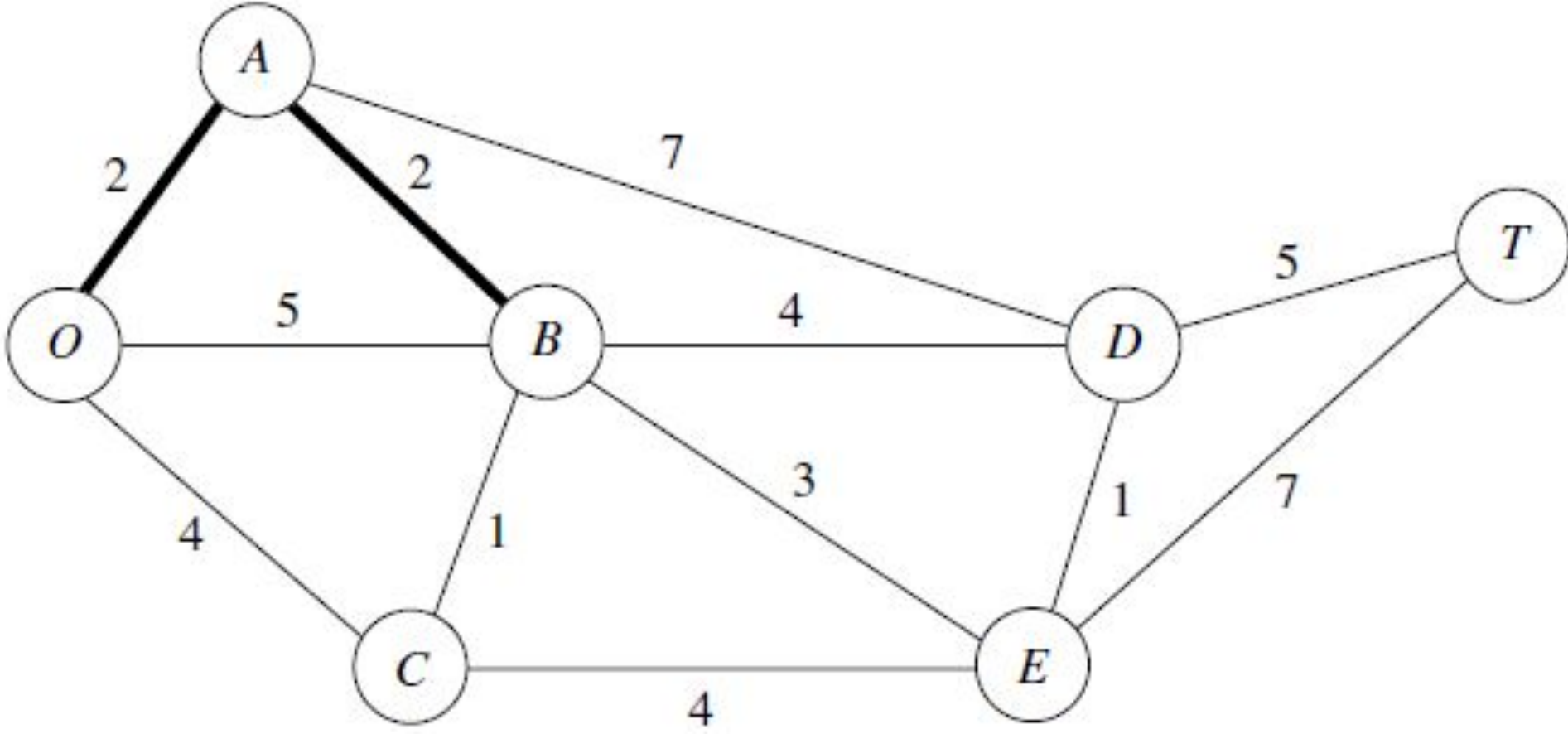
Все узлы теперь связаны, так что это решение является оптимальным. Общая длина звеньев составляет 14 км.

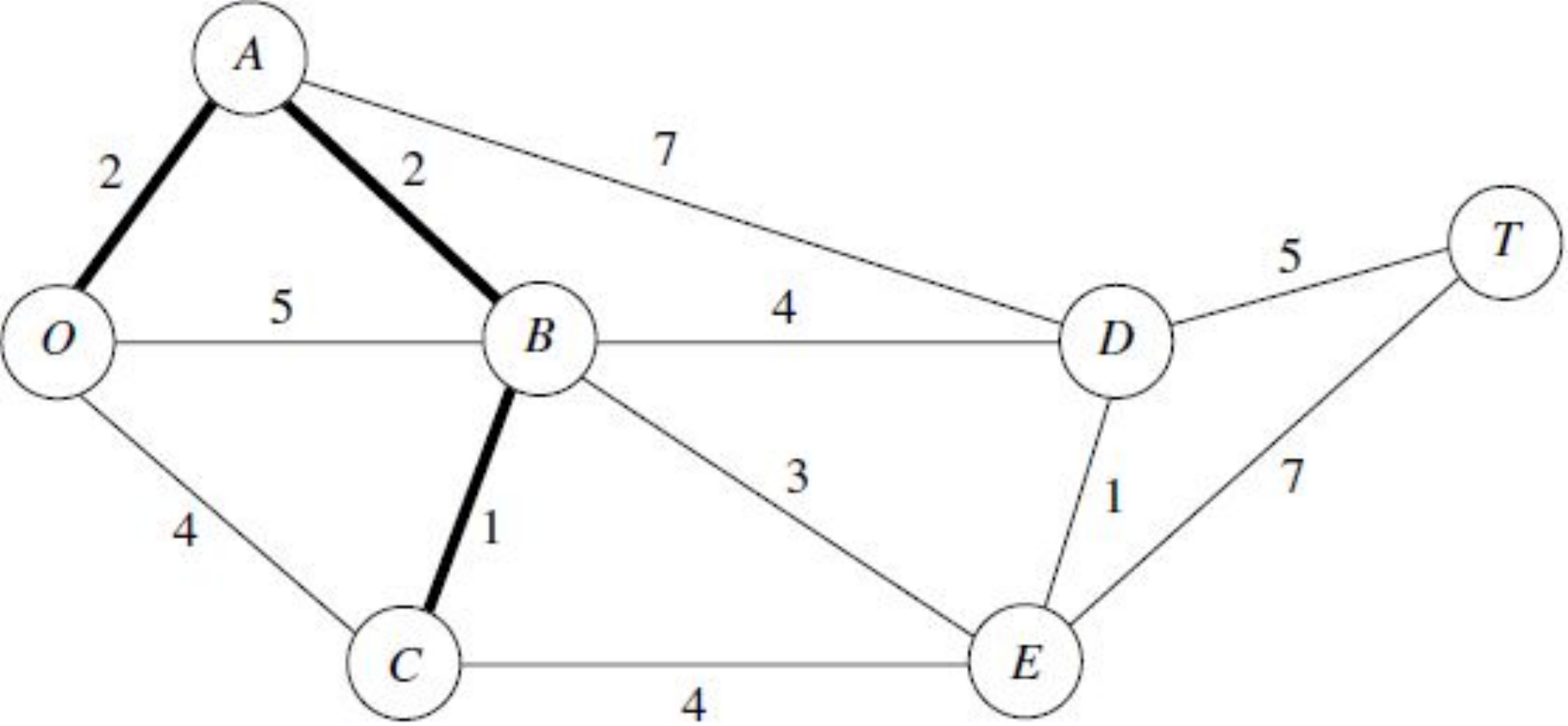
Выбор начального узла не влияет на окончательное решение (общая длина звеньев) при этой процедуре. Задача минимального связующего дерева подходит под широкую категорию задач *проектирования сетей*.

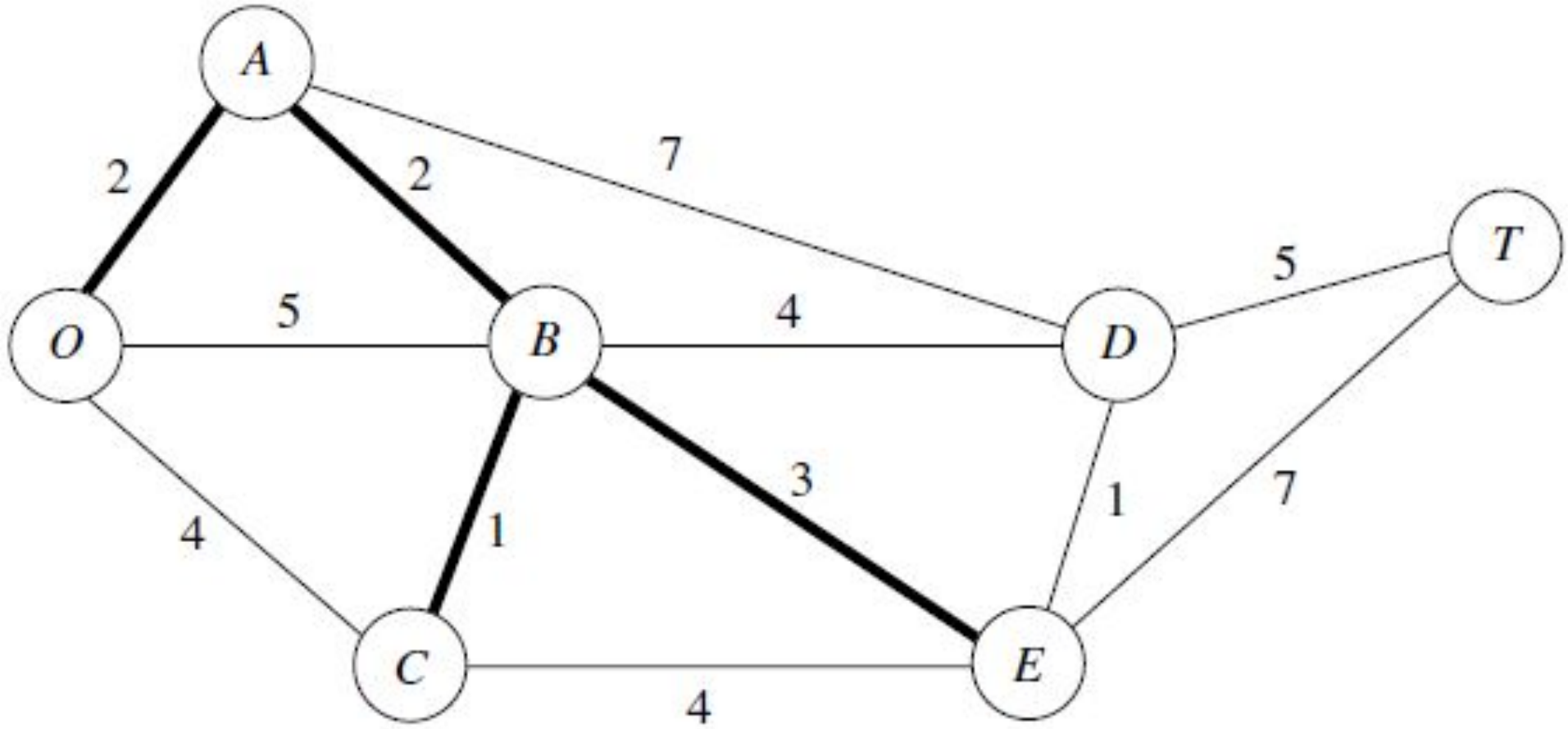
В этой категории цель заключается в *разработке* наиболее подходящей сети для данного приложения (включая транспортные системы), а не в анализе уже разработанной сети.

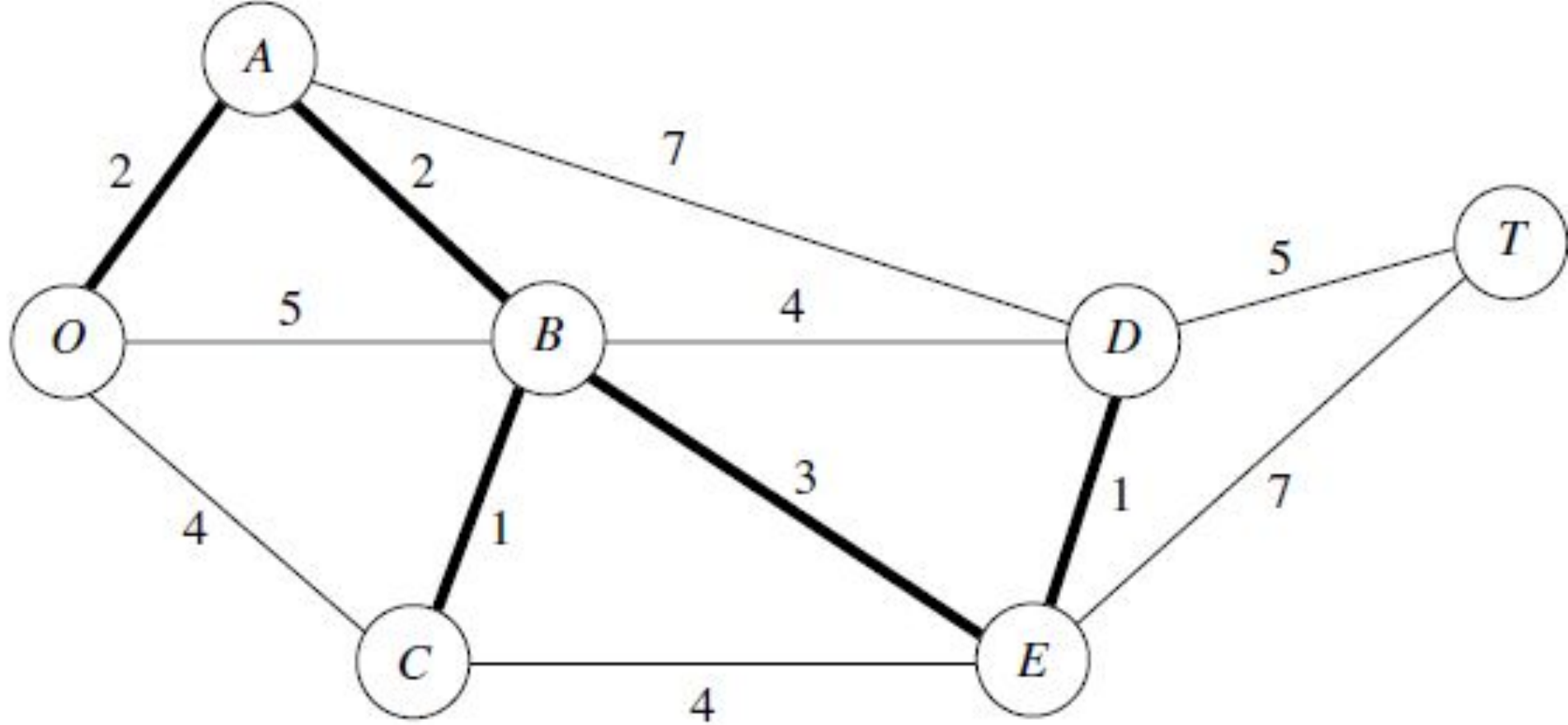


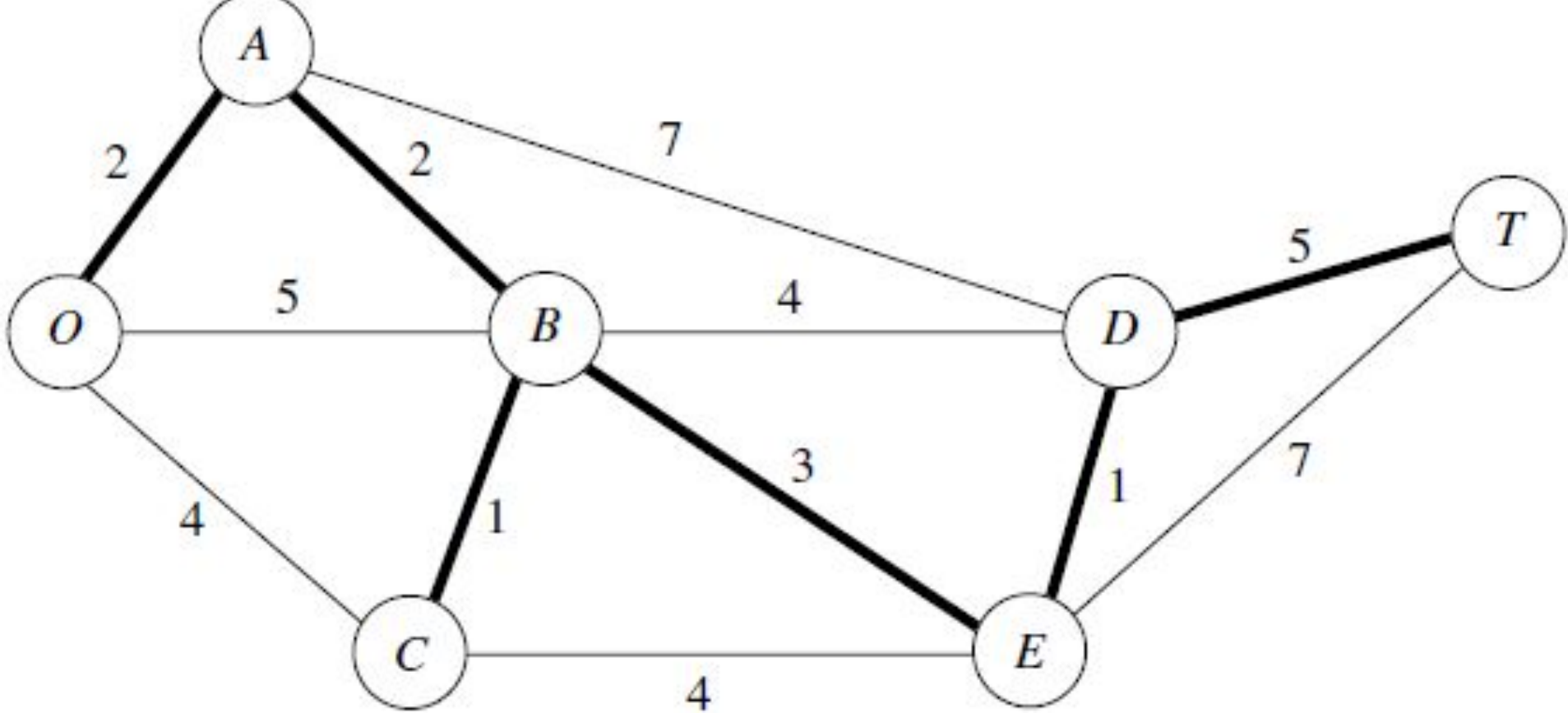












ЗАДАЧА МАКСИМАЛЬНОГО ПОТОКА

Третьей задачей, стоящей перед администрацией парка в пик сезона является составление маршрутов поездок к основной достопримечательности, так чтобы максимизировать количество поездок / день. Транспорт возвращается по тому же маршруту поездки (анализ фокусируется только на исходящих поездках). Чтобы не нарушать экологические требования, существует строгий верхний предел на исходящие поездки, разрешенные в день в исходящем направлении на каждой дороге (указано стрелкой). Число на стрелке указывает верхний предел на исходящие поездки - разрешено / день.

С учетом ограничений, одним из допустимых решений является отправка

7 трамваев / день, таким образом

5 используя маршрут $O \square B \square E \square T$,

1 используя $O \square B \square C \square E \square T$,

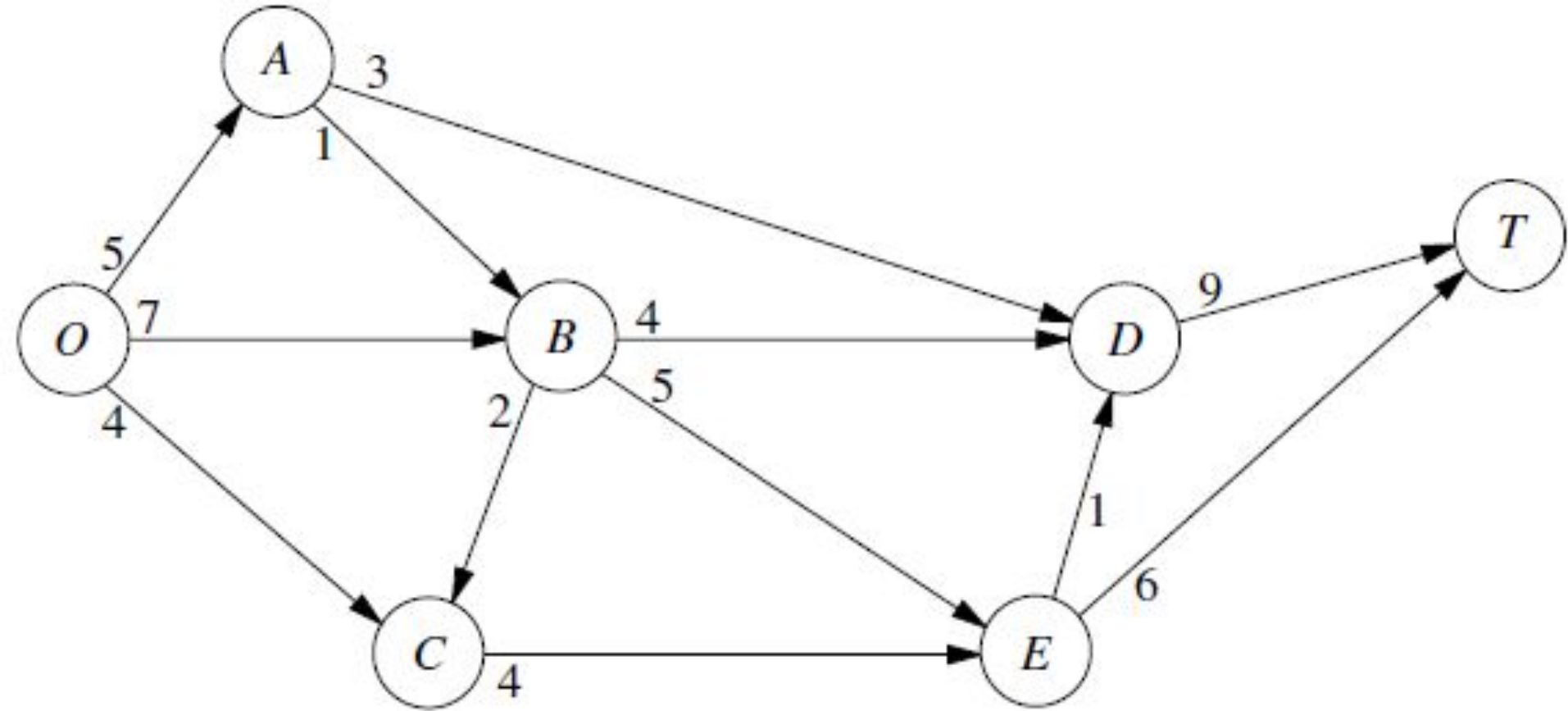
1 используя $O \square B \square C \square E \square D \square T$.

Это решение блокирует использование любых маршрутов, выезжающих из $O \square C$ (так как $E \square T$ и $E \square D$ возможности полностью использованы), поэтому легко найти лучшие возможные решения.

Необходимо учитывать несколько сочетаний маршрутов (и количество поездок для каждого), чтобы найти тот, при котором количество поездок / день максимально (задача о максимальном потоке).

Задача максимального потока описывается следующим образом:

1. Поток через ориентированные и связанные сети начинается в исходном узле и заканчивается в узле-приемнике [вход в парк (узел O) и основная достопримечательность (узел T)].
2. Остальные узлы – передающие узлы. (*узлы A, B, C, D, E*).
3. Поток через дугу разрешается только в направлении, указанном стрелкой, максимальное количество потока задается пропускной способностью дуги. В источнике, *дуги направлены от узла*. В приемнике, *дуги направлены к узлу*.
4. Цель состоит в максимизации общего количества потока от источника до приемника, измеряется либо *количеством, выходящим из источника* или *количеством, входящим в приемник*.



Задача о максимальном потоке на примере парка Кирова.

Применение:

1. Увеличить поток дистрибьюторской сети компании от своих заводов до заказчиков.
2. Увеличить поток сети поставок компании от ее поставщиков до своих заводов.
3. Увеличить поток нефти через систему трубопроводов.
4. Увеличить поток воды через систему акведуков.
5. Увеличить поток транспортных средств через транспортную сеть.

Для некоторых приложений, потоки сети начинаются в нескольких узлах и заканчиваются в нескольких узлах, но задача о максимальном потоке предполагает только один источник и один приемник. Например, дистрибьюторская сеть компании имеет несколько заводов и клиентов. Чтобы эта ситуация подходила под задачу максимального потока, необходима переформулировка (включает в себя расширение исходной сети, чтобы включить *фиктивные источники, фиктивные приемники и новые дуги*).

Фиктивный источник = в узле начинается поток, который на самом деле начинается в другом узле. Для каждого из этих других узлов, вставляется новая дуга, которая идет от фиктивных источников к этому узлу. Пропускная способность этой дуги = максимальному потоку, который, в действительности, может исходить из данного узла.

Фиктивный приемник = узел, поглощающий поток, который, в действительности, заканчивается в некотором другом узле. В каждом из этих других узлов вставляется новая дуга к фиктивному источнику. Пропускная способность этой дуги = максимальный поток, который, в действительности, может завершиться в этом узле.

Теперь исходные узлы сети будут являться передающими узлами, поэтому расширенная сеть требует один источник (фиктивный источник) и приемник (фиктивный приемник), чтобы

Алгоритм:

Задача о максимальном потоке = задача линейного программирования, решается симплекс-методом.

Используется более эффективный алгоритм добавляемого пути (на основе двух концепций: *остаточная сеть + увеличение пути*). После назначения потоков дуги, остаточная сеть показывает оставшиеся пропускные способности дуг (остаточную пропускную способность) для назначения *добавляемых потоков*. Например, дуга $O \square B$, имеет пропускную способность = 7.

Назначенные потоки включают поток 5 через эту дугу, которая оставляет остаточную емкость $7 - 5 = 2$ для назначения добавляемого потока через $O \square B$. Это состояние изображается в остаточной сети следующим:



Число дуг рядом с узлом дает остаточную пропускную способность для потока из этого узла в другой узел. В дополнение к остаточной пропускной способности 2 для потока от O до B, 5 справа показывает остаточную пропускную способность 5 для назначения потока от B до O (то есть, для отмены некоторых ранее назначенных потоков от O до B). Перед назначением потоков в остаточной сети (изменение дуг в исходной сети из ориентированной дуги на неориентированную дугу). Исходное направление пропускной способности сети остается таким же, пропускная способность дуги в обратном направлении = 0, так что ограничения на потоки остаются неизменными. Количество потока, назначенного для дуги вычитается из остаточной пропускной способности в том же направлении и добавляется к остаточной пропускной способности в противоположном направлении. Добавляемый путь = ориентированный путь от источника к приемнику в остаточной сети, так что каждая дуга на этом пути имеет строго положительную остаточную пропускную способность

Минимальная остаточная пропускная способность = остаточная пропускная способность добавляемого пути(она представляет собой величину потока, которая допустимо добавить ко всему пути) □ каждый добавляемый путь предоставляет возможность для дальнейшего увеличения потока через исходную сеть. Алгоритм добавляемого пути неоднократно выбирает некоторые увеличения пути и добавляет поток = его остаточную пропускную способность к исходной сети. Процесс продолжается, пока больше не останется путей, которые можно добавить - так что поток от источника к приемнику не может быть увеличен в дальнейшем. Обеспечение того, что окончательное решение является оптимальным - то, что увеличение пути может отменить некоторые ранее назначенные потоки в исходной сети, поэтому произвольный выбор пути для назначенных потоков не может предотвратить использование лучшей комбинации заданного потока. Каждая итерация алгоритма состоит из следующих трех этапов:

Алгоритм нахождения увеличивающего пути для задачи максимального потока:

1. Определить увеличивающий путь – находим в остаточной сети некоторый ориентированный путь от **источника** к принимающему узлу (**стоку**), так чтобы каждая дуга на этом пути имела строго положительную остаточную пропускную способность. (Если увеличивающий путь невозможен \square уже назначенные потоки сети представляют оптимальную модель потоков.)

2. Определить остаточную пропускную способность c^* этого увеличивающего пути – находим минимальную остаточную пропускную способность дуг на этом пути. Увеличиваем поток на этом пути на величину c^* .

3. Уменьшаем на величину c^* остаточную пропускную способность каждой дуги на этом увеличивающем пути. Увеличиваем на величину c^* остаточную пропускную способность каждой дуги в противоположном направлении на этом увеличивающем пути.

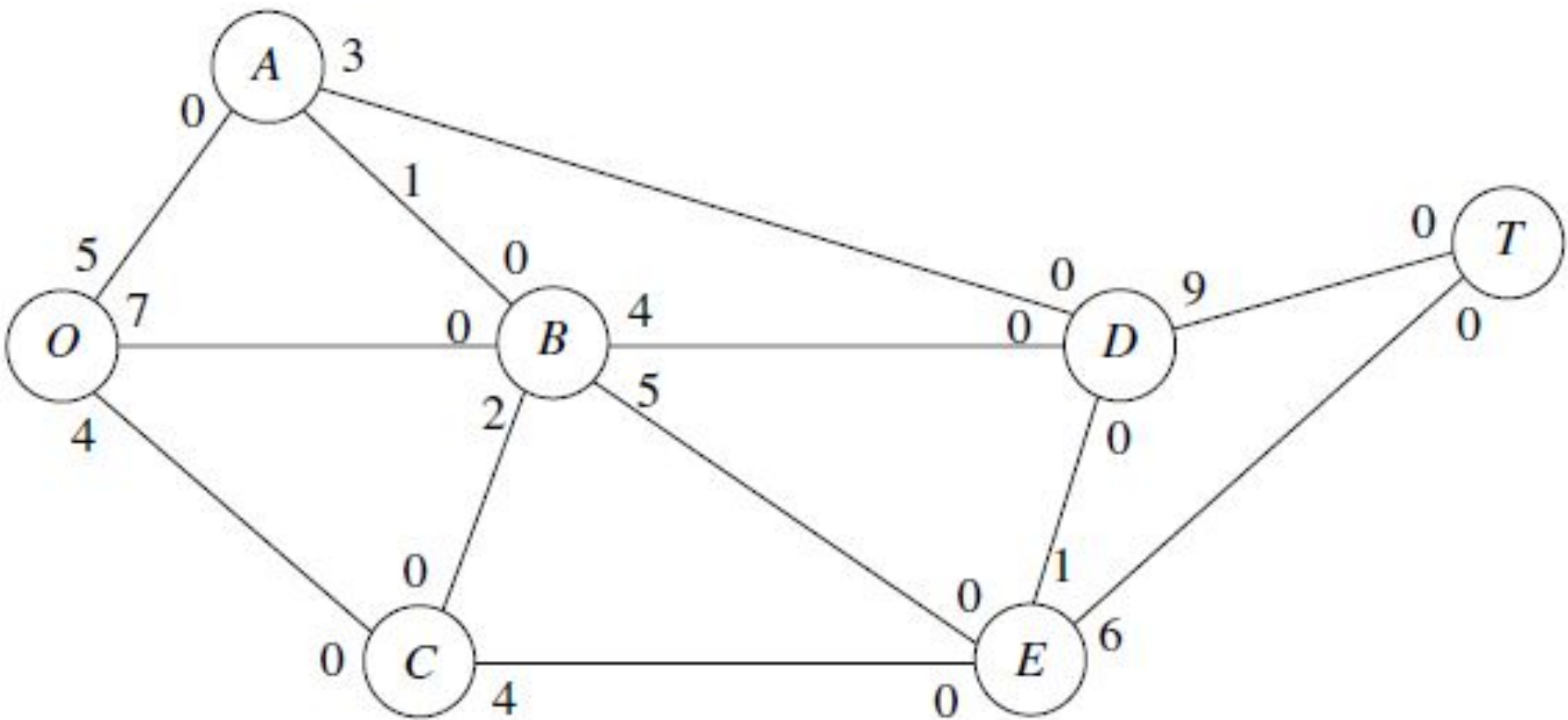
При выполнении шага 1 часто будет присутствовать число альтернативных увеличивающих путей, из которых нужно выбрать один. Стратегия алгоритма выбора важна для эффективности применения в больших масштабах.

Применение алгоритма для задачи максимального потока парка

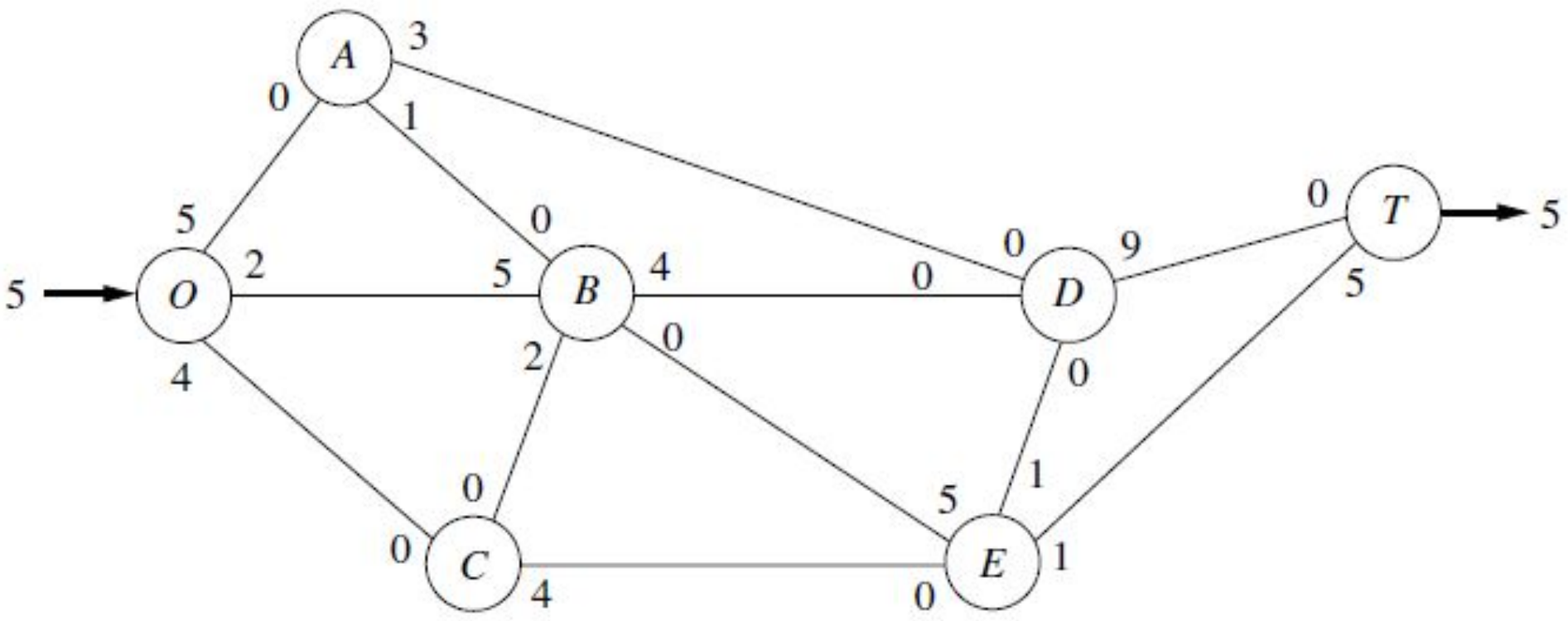
Начинаем с исходной остаточной сети, назначаем новую остаточную сеть после каждой итерации (или после каждых двух), где общая величина потока от O к T , достигнутая таким образом показано *жирным шрифтом* (около узлов O и T).

Итерация 1: Один увеличивающий путь $O \square B \square E \square T$, имеет остаточную пропускную способность минимум $\{7, 5, 6\} = 5$.

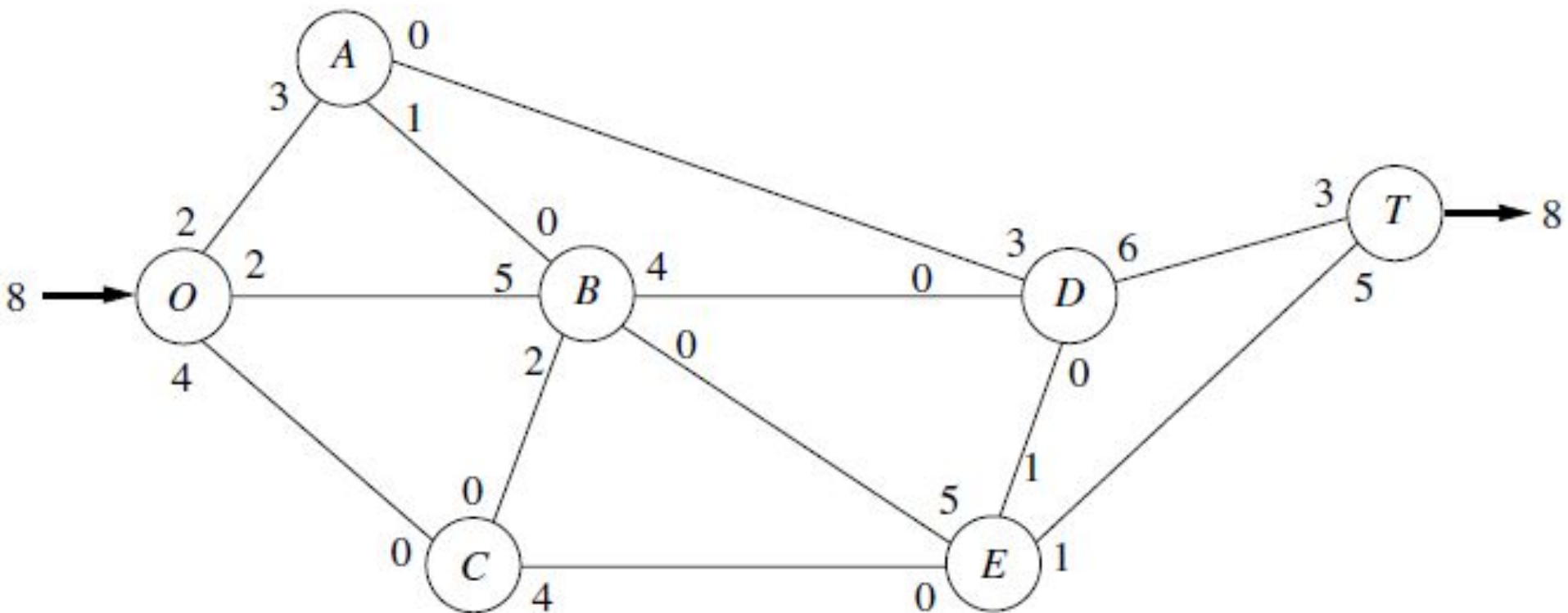
Назначим поток 5 для этого пути, получившаяся остаточная сеть имеет вид:



Исходная остаточная сеть для задачи максимального потока парка Кирова.



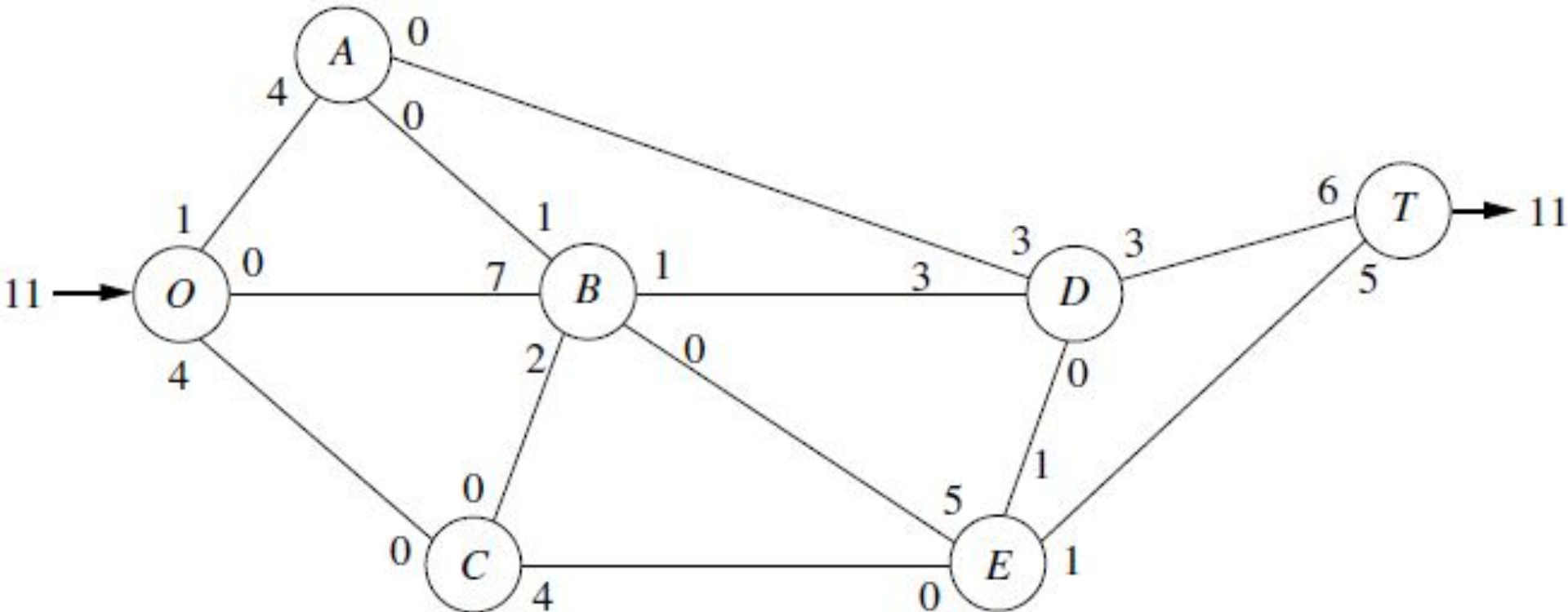
*Итерация 2: Назначим поток 3 для
увеличивающего пути $O \square A \square D \square T$.
Получившаяся остаточная сеть имеет вид*



Итерация 3: Назначим поток 1 для
увеличивающего пути $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$.

Итерация 4: Назначим поток 2 для
увеличивающего пути $O \rightarrow B \rightarrow D \rightarrow T$.

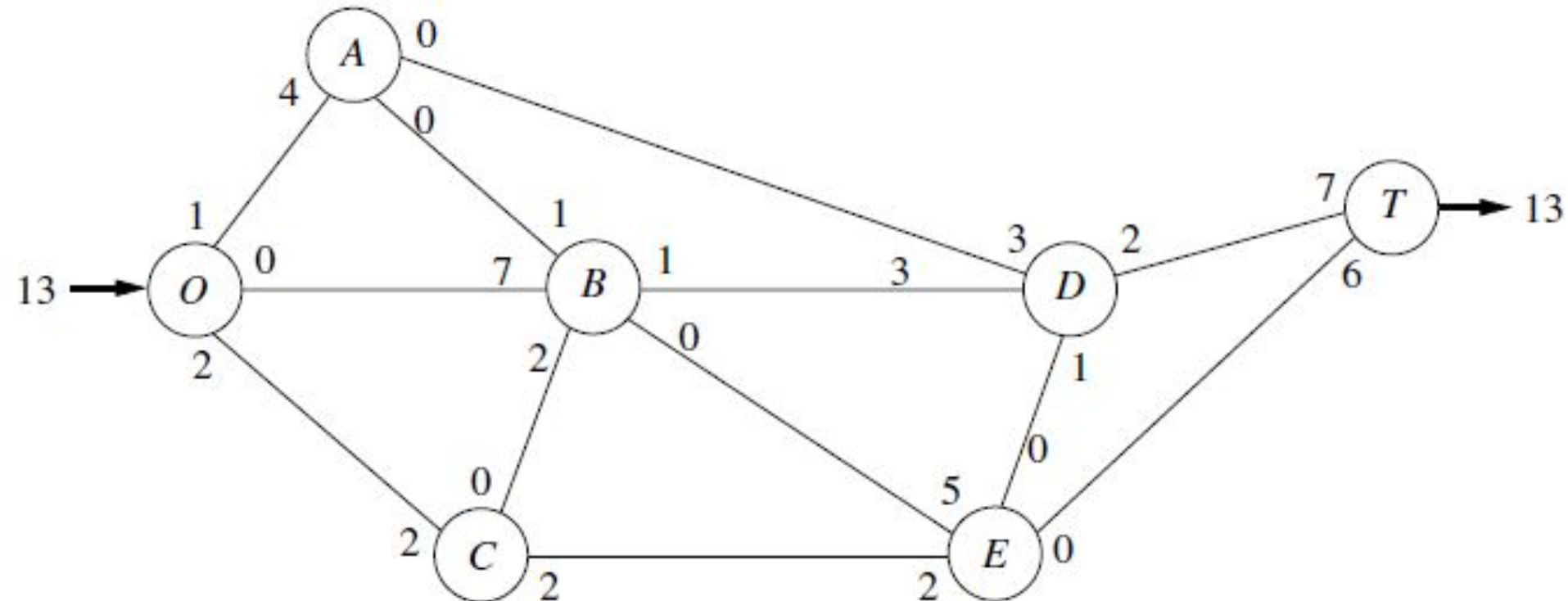
Получившаяся остаточная сеть имеет
вид



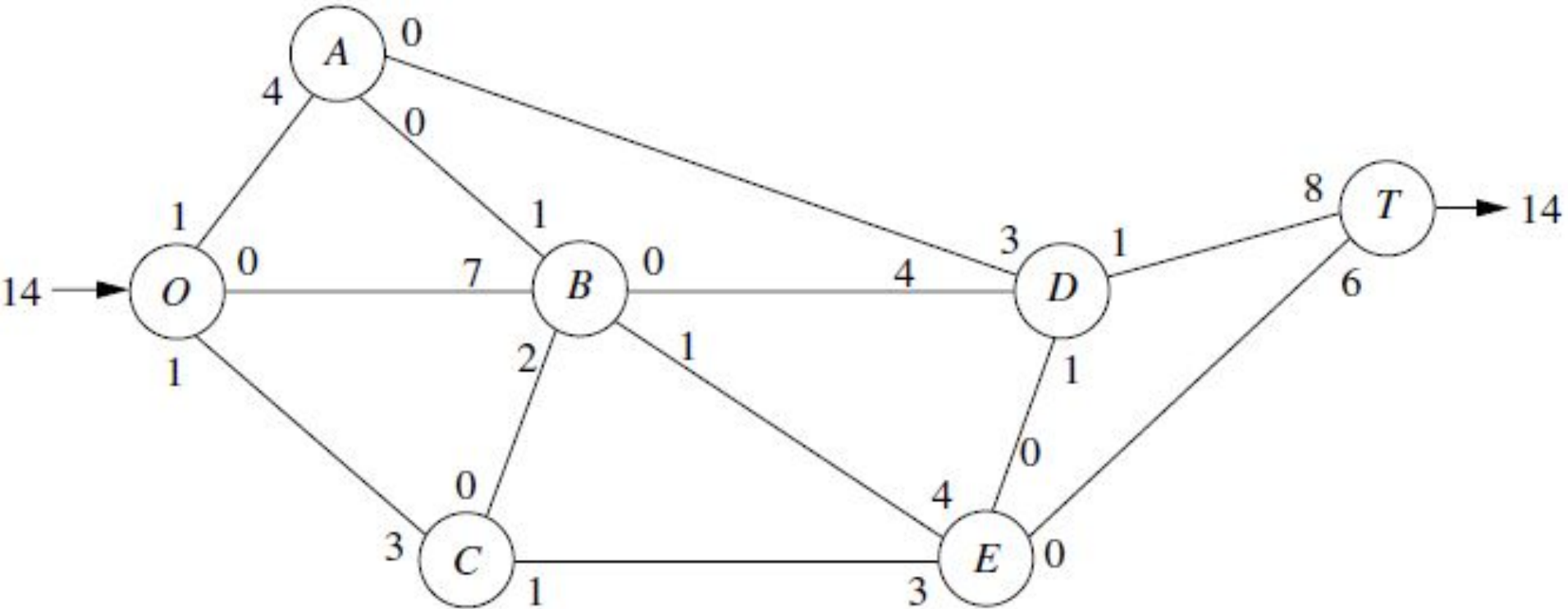
*Итерация 5: Назначим поток 1 для
увеличивающего пути $O \square C \square E \square D \square T$.*

*Iteration 6: Назначим поток 1 для
увеличивающего пути $O \square C \square E \square T$.*

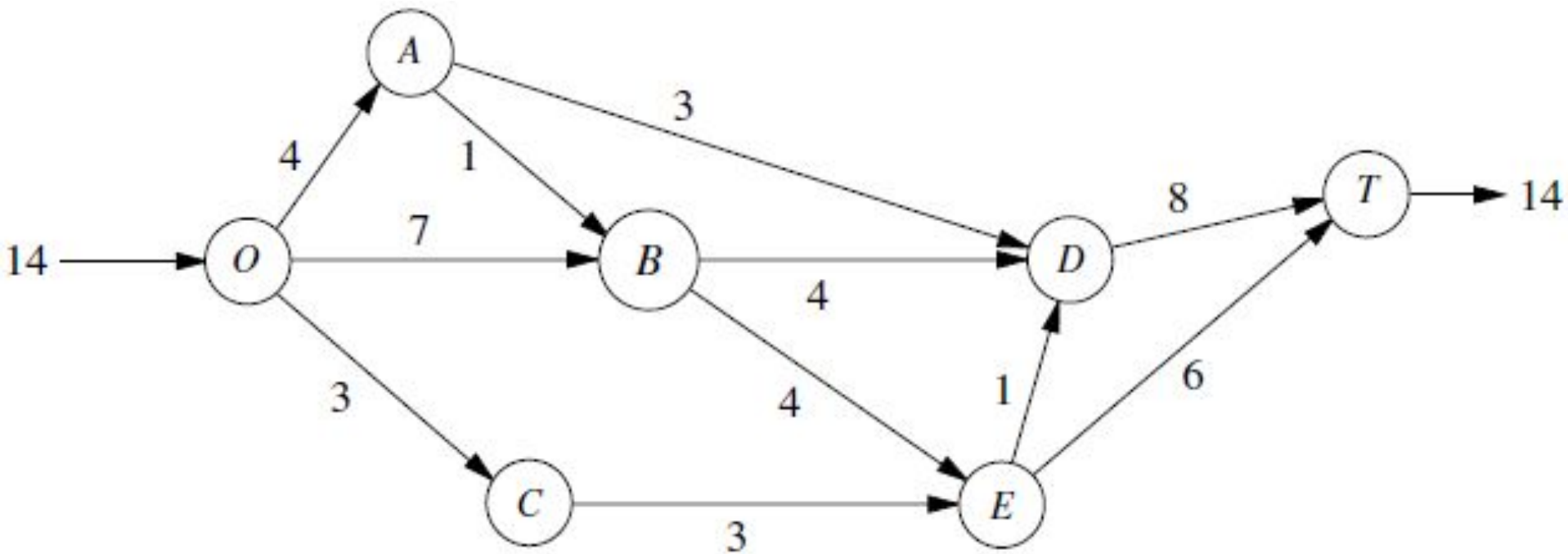
Получившаяся остаточная сеть имеет



Итерация 7: Назначим поток 1 для увеличивающего пути $O \square C \square E \square B \square D \square T$. Получившаяся остаточная сеть имеет



Больше не существует возможности увеличивающих путей, поэтому текущая модель потоков оптимальна.



Оптимальное решение для задачи максимальных потоков парка Кирова.

Текущая модель потоков: определяется

- Накопленными заданными потоками, или
- Сравнением конечных пропускных способностей с исходными пропускными способностями дуг (поток по дуге существует, если конечная остаточная пропускная способность $<$ исходной пропускной способности, величина потока = разница в этих пропускных способностях). Заменяя каждую ориентированную дугу $i \rightarrow j$ в исходной сети на неориентированную дугу в остаточной сети и увеличивая остаточную пропускную способность для $j \rightarrow i$ на величину c^* где поток c^* назначен для $i \rightarrow j$ (без этого изменения, первые 6 итераций будут неизменными, не останется увеличивающих путей потому что реальная неиспользованная пропускная способность дуги для $E \rightarrow B = 0$). Эта операция позволяет добавлять поток = 1 for $O \rightarrow C \rightarrow E \rightarrow B \rightarrow D \rightarrow T$ в итерации 7, это отменяет 1 единицу потока, назначенную на итерации 1.

$(O \rightarrow B \rightarrow E \rightarrow T)$ заменяет его назначением 1 единицы потока для $O \rightarrow B \rightarrow D \rightarrow T$ и $O \rightarrow C \rightarrow E \rightarrow T$.

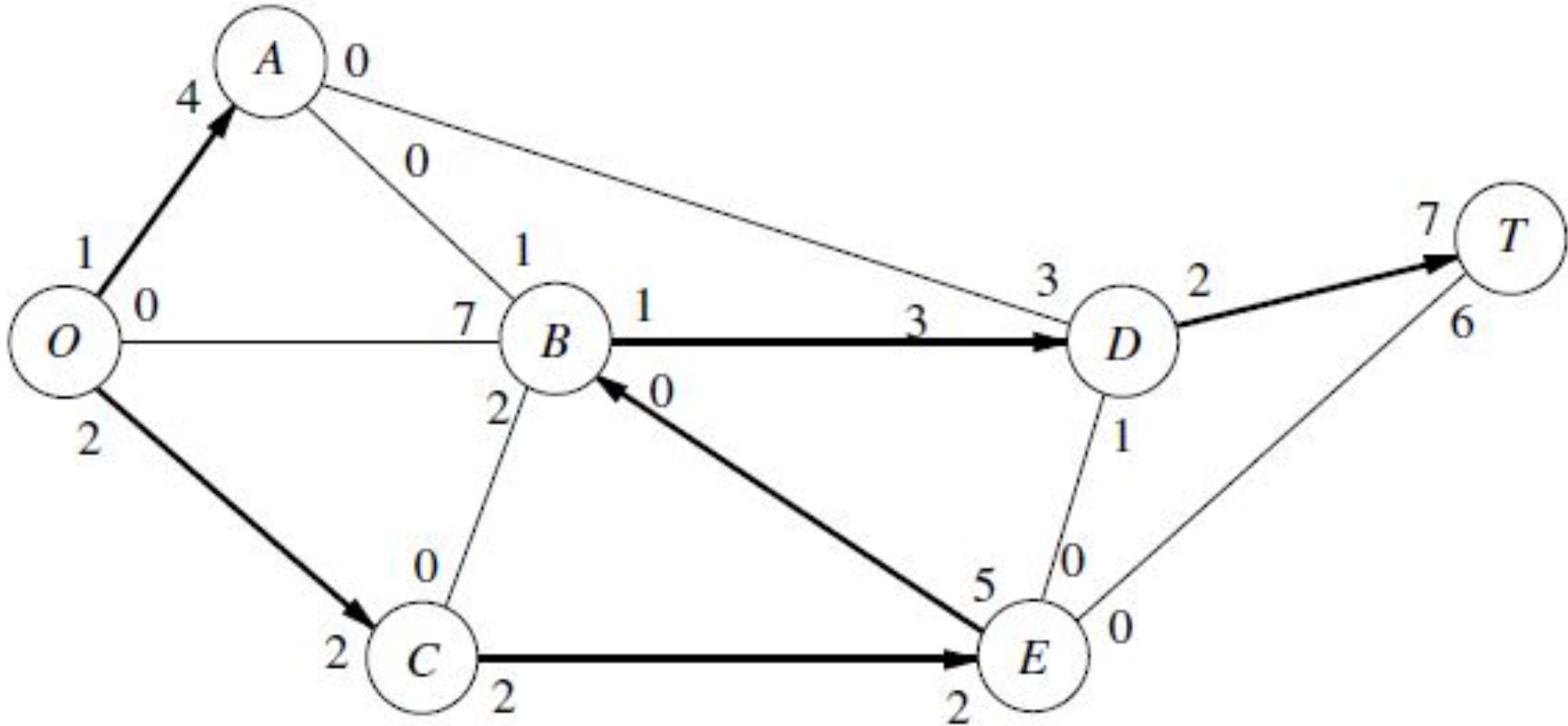
Нахождение увеличивающего пути:

Определяем узлы, которые могут быть связаны с источником при помощи одной единственной дуги со строго положительной остаточной пропускной способностью, или определяем новые узлы (еще не достигнутые), *которые могут быть связаны с этим узлом через дугу со строго положительной пропускной способностью.*

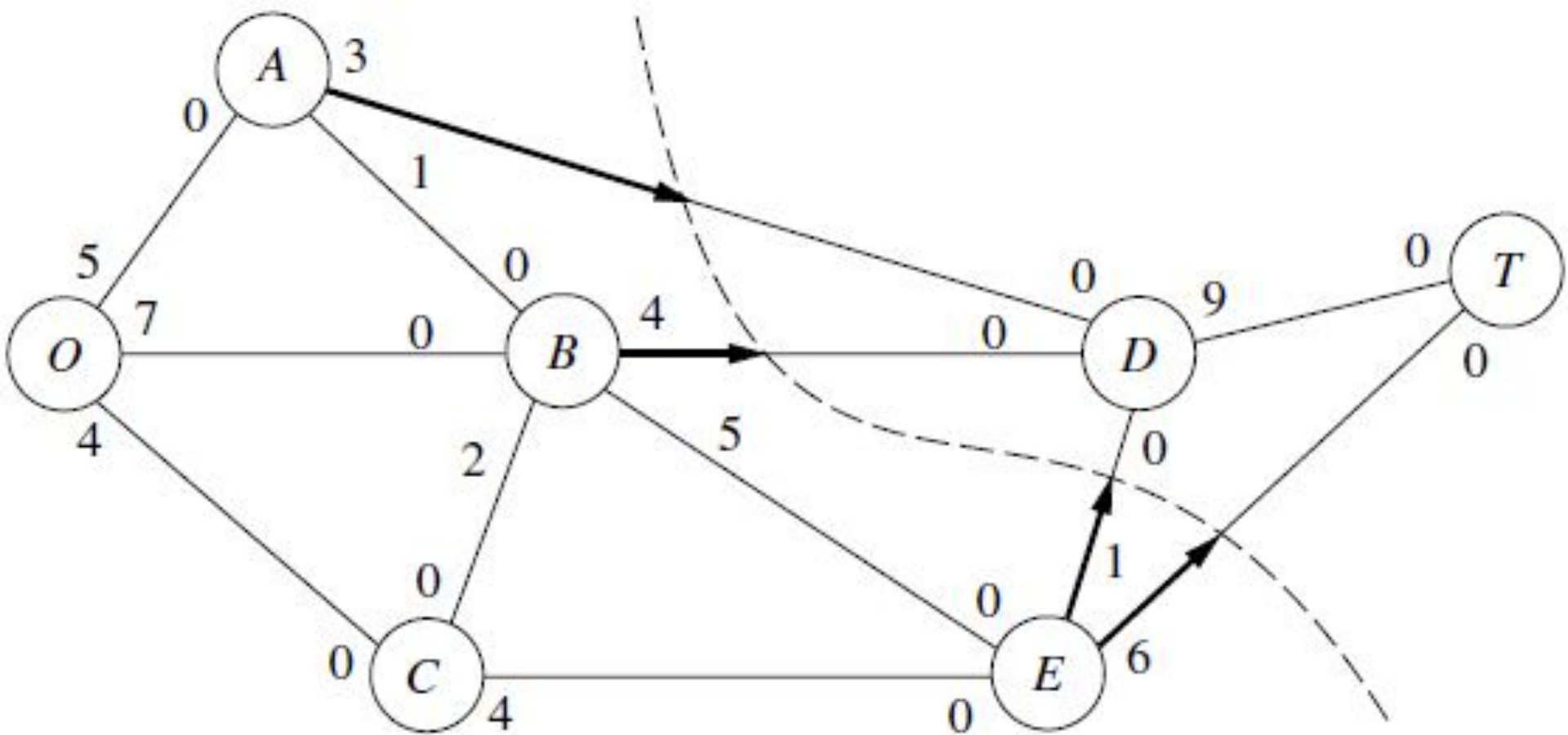
Последовательность действий повторяется при достижении новых узлов. В результате получится дерево из узлов, которые могут быть связаны с источником через путь со строго положительной остаточной пропускной способностью. Эта процедура определяет увеличивающий путь, если он существует. *Оптимальность проверяется теоремой минимального разреза максимального потока.*

Разрез = любое множество направленных дуг, содержащее минимум одну дугу из каждого ориентированного пути от источника к стоку (принимающей вершине).

Величина разреза = сумма пропускных способностей разрезанных ребер (в заданном направлении). Теорема минимального разреза максимального потока: для любой сети с единственным источником и стоком, *максимальный допустимый поток от источника к стоку = минимальному значению разреза для всех разрезов сети.* F = значение потока от источника к стоку для любой допустимой модели потока, тогда значение любого разреза представляет верхнюю границу F . *Наименьшие значения разреза = максимальное значение F □ если значение разреза = значению F полученного в процедуре текущего решения найденного в исходной сети, текущая модель потока должна быть оптимальной. Оптимальность достигается независимо от того, существует ли разрез = 0 в остаточной сети. Значение разреза $3 + 4 + 1 + 6 = 14$, - максимальное значение F , поэтому это минимальный разрез (разрез с минимальной пропускной способностью.) В остаточной сети, получившейся из итерации 7, где $F=14$, соответствующий разрез = 0. Как уже было замечено, нет необходимости искать дополнительные увеличивающие пути*



Процедура нахождения увеличивающего пути в итерации 7 задачи максимального потока парка Кирова.



Минимальный разрез для задачи максимального потока парка Кирова.