

УДК 519-161

А.В. Панишев¹, В.Л. Левковский², Г.В. Марчук³¹ ЖГТУ, г. Житомир, Украина, pzs.ztu@gmail.com² ЖГТУ, г. Житомир, Украина, levkivskyy@ztu.edu.ua³ ЖГТУ, г. Житомир, Украина, mgv.555.mgv@gmail.com

ПРЕОБРАЗОВАНИЯ В ПРИБЛИЖЕННЫХ АЛГОРИТМАХ РЕШЕНИЯ СИММЕТРИЧНОЙ ЗАДАЧИ КОММИВОЯЖЁРА (СЗК)

В статье приведено описание алгоритма нахождения гамильтонова цикла по i -дереву симметричной задачи коммивояжера. Алгоритм строит приближенное решение метрической задачи коммивояжера с оценкой погрешности меньшей, чем у большинства приближенных алгоритмов. Полученная оценка для данного алгоритма является не улучшаемой. По быстродействию предложенный алгоритм превосходит все известные алгоритмы метрической задачи коммивояжера.

СИММЕТРИЧНАЯ ЗАДАЧА КОММИВОЯЖЕРА (СЗК), I-ДЕРЕВО, ПРИБЛИЖЕННЫЕ АЛГОРИТМЫ, ГРАФ, ГАМИЛЬТОНОВ ЦИКЛ

Введение

Одна из самых известных задач комбинаторной оптимизации — это задача коммивояжера (англ. Travelling salesman problem). Задача коммивояжера (ЗК) состоит в том, чтобы найти кратчайший маршрут, проходящий через каждый город точно один раз, при заданных расстояниях между двумя городами [1].

ЗК является так называемой NP-трудной задачей, и при большом количестве вершин графа, точное решение может быть получено только за экспоненциальное время и решать ее переборными алгоритмами будет неэффективно.

В статье будет рассмотрен один из этапов построения приближенного решения СЗК, алгоритм нахождения гамильтонова цикла в i -дереве $T = (V, E_i)$ полного графа $H_n = (V, E_n)$ с невысокой погрешностью.

1. Постановка задачи

Суть задачи коммивояжера сводится к нахождению гамильтонова цикла минимальной стоимости. Частным случаем постановки задачи коммивояжера есть симметрическая задача коммивояжера (СЗК), это задача с симметричной матрицей весов, которая моделируется неориентированным графом.

Входными данными является i -дерево $T = (V, E_i)$ полного графа $H_n = (V, E_n)$.

Выходными данными является гамильтонов цикл полученный с помощью алгоритма преобразования i -дерево T .

2. Преобразование i -дерева в гамильтонов цикл

В СЗК для матрицы стоимостей $[d_{ij}]_n$, в которой $d_{ij} = d_{ji}$, $i \neq j$, $d_{ij} \geq 0$, $d_{ii} = \infty$ и $i, j = \overline{1, n}$, минимизируемый функционал определяется по формуле

$$D(\tau) = \sum_{i=1}^n d_{\tau[i]}, \quad (1)$$

где $\tau = (\tau[1], \tau[2], \dots, \tau[n])$ — циклическая перестановка или обход, которому соответствует маршрут коммивояжера $\tau = (\tau[1], \tau[2], \dots, \tau[n], \tau[1])$, а величину $D(\tau)$ называют стоимостью обхода [2].

Матрице $[d_{ij}]_n$ отвечает полный взвешенный граф

$$H_n = (V, E_n), |V| = n, |E_n| = C_n^2,$$

где ребру (i, j) приписан вес $d_{ij} = d_{ji}$. Допустимым решением СЗК является гамильтонов цикл, стоимость которого определяется из матрицы $[d_{ij}]_n$ и выражения (1).

В СЗК число всех допустимых решений равно $(1/2)(n-1)!$.

Быстродействие и точность приближенных методов решения СЗК заданной размерности n зависит от свойств матрицы стоимостей $[d_{ij}]_n$ и значений ее элементов. Для $[d_{ij}]_n$ число значений d_{ij} будет не больше $n(n-1)/2$, если же значения d_{ij} принадлежат промежутку $[a, b]$, то в случае выполнения неравенства

$$\frac{n(n-1)}{2} > b - a + 1$$

матрица $[d_{ij}]_n$ содержит несколько элементов одинакового веса, то есть $i < j$.

Решение СЗК точными методами не представляется возможным. Нахождение минимального обхода в СЗК гарантируют, как известно переборные методы, но применение, например алгоритма ветвей и границ, приводит к быстрому росту числа вершин дерева ветвлений и неэффективной работе правил отсева из-за большого количества чисел оценок, принимающих близкие значения. Единственной альтернативой точным методам являются приближенные алгоритмы решений.

Работоспособность приближенных алгоритмов комбинаторной оптимизации характеризуется двумя противоречивыми показателями: точностью и временем решения, полиномиально зависящих от размера входных данных задачи. При рассмотрении матриц стоимостей фиксированного размера

СЗК наблюдается характерная для приближенных алгоритмов тенденция роста времени и погрешности решения с увеличением числа близких значений d_{ij} .

Рассмотрим СЗК, где матрица стоимостей d_{ij} содержит достаточно большое число равных или близких друг к другу значений, образуя широкую область поиска в окрестностях оптимума (1). Рассматривая работу приближенных алгоритмов с подобными условиями СЗК можно заметить тенденцию уменьшения быстродействия работы алгоритмов.

Основная идея предлагаемого метода состоит в построении i -дерева, обобщающего понятие 1-дерева [2].

Под i -деревом $T=(V, E_i)$ будем понимать подграф полного графа, $H_n=(V, E_n)$ который содержит дерево, стягивающее множество вершин $V \setminus \{i\}$, и два ребра, инцидентные вершине $i, i \in \{1, 2, \dots, n\}, |V|=n$. Рассмотрим работу алгоритма, который использует i -дерево для уменьшения погрешности приближенных алгоритмов СЗК.

В i -дереве T обозначим $\delta_k, k=\overline{1, n}$, степень вершины k . Множество вершин V дерева представим разбиением на три подмножества V_1, V_2, V_3 , где V_1 – множество вершин с единичной степенью (конечных или висячих вершин), V_2 – множество вершин со степенью 2, а $V_3 = V \setminus (V_1 \cup V_2)$.

При построении алгоритма будем использовать Лемму [1], где для любого i -дерева графа $H_n=(V, E_n), |V|=n, i \in \{1, 2, \dots, n\}, n \geq 3$ справедливо равенство [1]:

$$|V_1| = \sum_{k=1}^n \max\{\delta_k - 2, 0\} \quad (2)$$

В i -дереве графа $H_n=(V, E_n)$, представленном гамильтоновым циклом, $V=V_2$. По определению любое i -дерево содержит один цикл, отсюда следует, что $V_2 \neq \emptyset$. Связный граф $T=(V, E_n), |V|=|E_n|$, в котором число вершин степени 2 меньше n , не является гамильтоновым циклом. Кроме того, из (2) следует, что если в i -дереве $T V_3 \neq \emptyset$, то и $V_1 \neq \emptyset$. Заметим, что для любого i -дерева графа $H_n=(V, E_n)$ выполняется условие $1 \leq |V_1| \leq |V| - 3$.

По теореме 2.1 [1] подграф полного графа $H_n=(V, E_n)$ представленный i -деревом $T=(V, E_i), |V|=|E_i|=n, i \in \{1, 2, \dots, n\}$, корректно преобразуется в гамильтонов цикл за время $O(n|V_i|)$, где $|V_i|$ - число висячих вершин в T .

i -дерево T полного графа $H_n=(V, E_n)$ преобразуется в гамильтонов цикл в результате следующих действий:

S0. По i -дереву $T=(V, E_i), |E_i|=n$, построить матрицу смежности. Взять i как начальную вершину и проделать проход по i -дереву по левой цепи или правой цепи, изменяя δ_k . В результате построить матрицу смежности, в которых $\delta_i = \delta_k = 2$.

Определить k как текущую вершину, $k = i$.

S1. Пока $V_1 > 0$ выполнять начало

S2. Пока $\delta_k = 2$ выполнять начало

запомнить текущую вершину как пройденную, $l = k$;

определить k как следующую вершину, смежную с пройденной

конец;

удалить из матрицы смежности ребро (l, k) и соответственно (k, l) ;

взять любую вершину f из V_1 ;

из V_1 удалить f ;

добавить в матрицу смежности ребро (l, f) и соответственно (f, l) ;

определить k как следующую вершину $k = f$;

конец

На шаге *S0* определяем вершину, которая будет начальной и направление прохождения по матрице смежности: по правой или левой цепи. В цикле *S1* удаляется ребро (l, k) , соответственно степень вершины уменьшается на единицу $\delta_k = \delta_k - 1$, если $\delta_k > 2$, и добавляем новое ребро (l, f) , где f , висячая вершина, соответственно степень вершины увеличивается на единицу $\delta_k = \delta_k + 1$, при этом из массива висячих вершин удаляем вершину f .

Пример. Преобразуем 10-дерево $T=(V, E_{10})$ полного графа $H_{11}=(V, E_{11})$ рис. 1, в гамильтонов цикл.

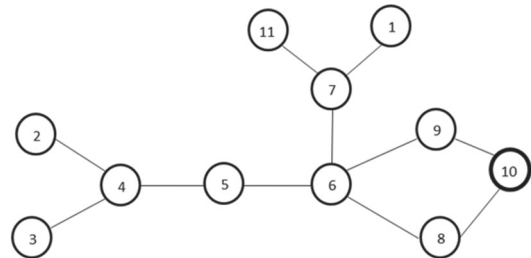


Рис.1. 10-дерево $T=(V, E_{10})$ полного графа H_{11}

Входная матрица смежности 10-дерево $T=(V, E_{10})$ полного графа H_{11} изображена на рис. 2.

	1	2	3	4	5	6	7	8	9	10	11
1	-	0	0	0	0	0	1	0	0	0	0
2	0	-	0	1	0	0	0	0	0	0	0
3	0	0	-	1	0	0	0	0	0	0	0
4	0	1	1	-	1	0	0	0	0	0	0
5	0	0	0	1	-	1	0	0	0	0	0
6	0	0	0	0	1	-	1	1	1	0	0
7	1	0	0	0	0	1	-	0	0	0	1
8	0	0	0	0	0	1	0	-	0	1	0
9	0	0	0	0	0	1	0	0	-	1	0
10	0	0	0	0	0	0	0	1	1	-	0
11	0	0	0	0	0	0	1	0	0	0	-

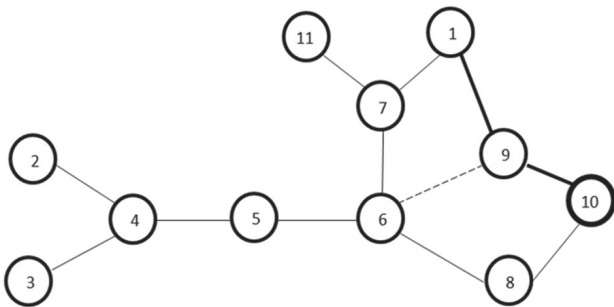
Рис. 2. Матрица смежности 10-дерево T

Перед началом работы определим массив висячих вершин $V_1 = \{1, 2, 3, 11\}$, начальную вершину $k = 10$, обход матрицы начинаем справа. Так как $\delta_{10} = 2$ запоминаем текущую вершину $l = k = 10$ и выбираем следующую вершину $k = 9$. Цикл повторяется до тех пор, пока $\delta_k = 2$.

При $k = 6, \delta_6 = 4, l = 9$, удаляем ребро $(6, 9)$ и соответственно $(9, 6)$ из матрицы смежности рис. 3 а), ребро изображено пунктирной линией на рис. 3 б).

	1	2	3	4	5	6	7	8	9	10	11
1	-	0	0	0	0	0	1	0	1	0	0
2	0	-	0	1	0	0	0	0	0	0	0
3	0	0	-	1	0	0	0	0	0	0	0
4	0	1	1	-	1	0	0	0	0	0	0
5	0	0	0	1	-	1	0	0	0	0	0
6	0	0	0	0	1	-	1	1	0	0	0
7	1	0	0	0	0	1	-	0	0	0	1
8	0	0	0	0	0	1	0	-	0	1	0
9	1	0	0	0	0	0	0	0	-	1	0
10	0	0	0	0	0	0	0	1	1	-	0
11	0	0	0	0	0	0	1	0	0	0	-

а



б

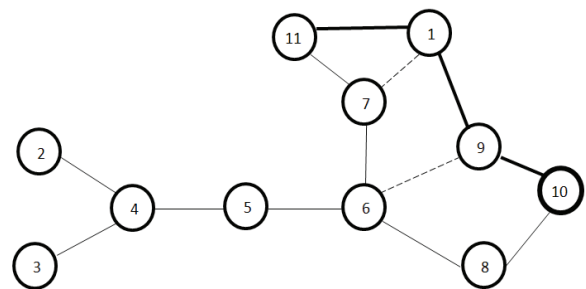
Рис. 3. а — Матрица смежности после выполненных операций, б — Граф, построенный после выполненных операций

Из массива висячих вершин выбираем любую вершину f , например $f = 1$ и добавляем ребро $(9, 1)$ и соответственно $(1, 9)$ в матрицу смежности рис. 4, а, на рис. 4, б, ребро изображено жирной линией, при этом степень текущей вершины уменьшается на 1 и $\delta_6 = 3$. Текущей вершиной становится вершина 1, $k = 1$ предыдущая вершина $l = 9$.

Рассмотрим вершину 1, степень вершины $\delta_1 = 2$, переходим к следующей вершине $l = 1, k = 7$. Вершина 7 имеет степень $\delta_7 = 3$, проделываем все предыдущие шаги: удаляем ребро $(7, 1)$ и соответственно $(1, 7)$ из матрицы смежности рис. 4, а, на рис. 4, б, ребро изображено пунктирной линией. Из массива висячих вершин выбираем любую вершину f , например $f = 11$ и добавляем ребро $(1, 11)$ и соответственно $(11, 1)$ в матрицу смежности рис. 4, а), а на рис. 4, б, ребро изображено жирной линией, при этом степень текущей вершины уменьшается на 1 и $\delta_7 = 2$. Текущей вершиной становится вершина 11, $k = 11$ предыдущая вершина $l = 1$.

	1	2	3	4	5	6	7	8	9	10	11
1	-	0	0	0	0	0	0	0	1	0	1
2	0	-	0	1	0	0	0	0	0	0	0
3	0	0	-	1	0	0	0	0	0	0	0
4	0	1	1	-	1	0	0	0	0	0	0
5	0	0	0	1	-	1	0	0	0	0	0
6	0	0	0	0	1	-	1	1	0	0	0
7	0	0	0	0	0	1	-	0	0	0	1
8	0	0	0	0	0	1	0	-	0	1	0
9	1	0	0	0	0	0	0	0	-	1	0
10	0	0	0	0	0	0	0	1	1	-	0
11	1	0	0	0	0	0	1	0	0	0	-

а

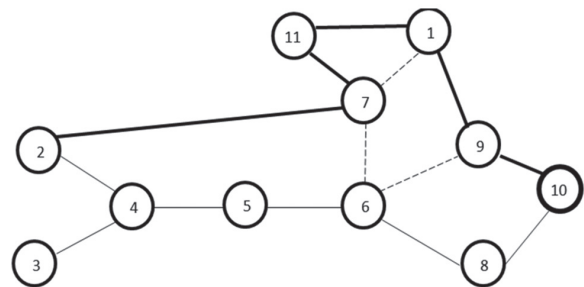


б

Рис. 4. а — Матрица смежности после выполненных операций, б — Граф, построенный после выполненных операций

	1	2	3	4	5	6	7	8	9	10	11
1	-	0	0	0	0	0	0	0	1	0	1
2	0	-	0	1	0	0	1	0	0	0	0
3	0	0	-	1	0	0	0	0	0	0	0
4	0	1	1	-	1	0	0	0	0	0	0
5	0	0	0	1	-	1	0	0	0	0	0
6	0	0	0	0	1	-	0	1	0	0	0
7	0	1	0	0	0	0	-	0	0	0	1
8	0	0	0	0	0	1	0	-	0	1	0
9	1	0	0	0	0	0	0	0	-	1	0
10	0	0	0	0	0	0	0	1	1	-	0
11	1	0	0	0	0	0	1	0	0	0	-

а



б

Рис. 5. а — Матрица смежности после выполненных операций, б — Граф, построенный после выполненных операций

Рассматриваем вершину 11 - $\delta_{11}=2$, переходим к следующей вершине $l=11, k=7$. Вершина 7 - $\delta_7=2$, переходим к следующей вершине $l=7, k=6$.

Вершина 6 имеет степень $\delta_6=3$, проделываем все предыдущие шаги: удаляем ребро (6, 7) и соответственно (7, 6) из матрицы смежности рис. 5, а, удаленное ребро изображено пунктирной линией на рис. 5, б. Из массива висячих вершин выбираем любую вершину f , например $f=2$ и добавляем ребро (7, 2) и соответственно (2, 7) в матрицу смежности рис. 5, а, ребро изображено жирной линией на рис. 5, б, при этом степень текущей вершины уменьшается на 1 и становится $\delta_6=2$. Текущей вершиной становится вершина 2, $k=2$ предыдущая вершина $l=7$.

Продолжаем с оставшимися вершинами все предыдущие действия и на выходе получаем матрицу смежности и граф, изображенные на рис. 6.

	1	2	3	4	5	6	7	8	9	10	11
1	-	0	0	0	0	0	0	0	1	0	1
2	0	-	1	0	0	0	1	0	0	0	0
3	0	1	-	1	0	0	0	0	0	0	0
4	0	0	1	-	1	0	0	0	0	0	0
5	0	0	0	1	-	1	0	0	0	0	0
6	0	0	0	0	1	-	0	1	0	0	0
7	0	1	0	0	0	0	-	0	0	0	1
8	0	0	0	0	0	1	0	-	0	1	0
9	1	0	0	0	0	0	0	0	-	1	0
10	0	0	0	0	0	0	0	1	1	-	0
11	1	0	0	0	0	0	1	0	0	0	-

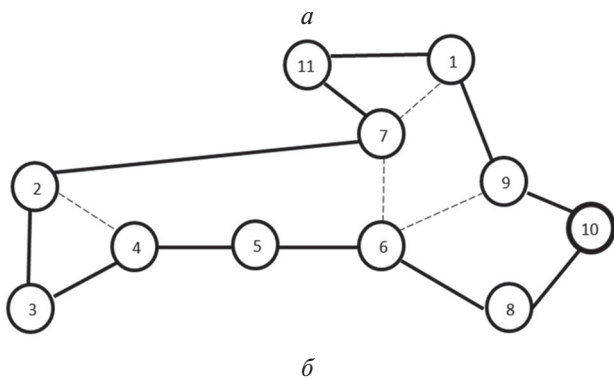


Рис.6. а – Матрица смежности, полученная после нахождения гамильтонова цикла, б – Гамильтонов цикл (10, 9, 1, 11, 7, 2, 3, 4, 5, 6, 8, 10), полученный при преобразовании 10-дерева $T=(V, E_{10})$

Предложенный алгоритм будем рассматривать как одну из стадий построения приближенного решения СЗК.

Обозначим i -дерево минимальной стоимости в полном взвешенном графе $H_n=(V, E_n), |V|=n$. Для его построения найдем в полном взвешенном графе с множеством вершин $V \setminus \{i\}$, минимальное остовное дерево (МОД) и добавим к нему пару

ребер с наименьшими весами, инцидентных вершине i . Стоимость $d(T_i)$ дерева T_i , равная сумме весов входящих в него ребер, является оценкой снизу стоимости оптимального решения τ^* СЗК:

$$d(T_i) \leq D(\tau^*), i = \overline{1, n}$$

Трудоёмкость нахождения дерева T_i оценивается временем работы градиентного алгоритма построения МОД. Известно, что МОД графа с множеством ребер E можно получить за время $O(|E| \log_2 |E|)$.

Сформулируем СЗК в терминах введенных понятий.

Пусть построено МОД $T_\vartheta, \vartheta = \{1, 2, \dots, n\}$ в полном взвешенном графе $H_n=(V, E_n)$ с весами (стоимостями) ребер $\{i, j\}, i, j = \overline{1, n}, d_{ij} \geq 0, i \neq j, d_{ii} = \infty$. Каждое ребро $\{k, l\}$ дерева T_ϑ представим парой (δ_k, δ_l) , где δ_k, δ_l – степени вершин k, l в $T_\vartheta, \delta_k \leq \delta_l, k, l \in \{1, 2, \dots, n\}$. Требуется найти в дереве T_ϑ все такие ребра, замена которых на ребро из H_n преобразует T_ϑ в маршрут коммивояжера минимальной стоимости, представленный n парами $(\delta_x, \delta_y), \delta_x = \delta_y = 2, x = \overline{1, n}, y \in \{1, 2, \dots, n\}, x \neq y$.

Ясно, что такая формулировка не приводит к простому и быстрому способу построения оптимального решения. Преобразования дерева T_ϑ в обход минимальной стоимости τ^* в силу NP-полноты рассматриваемой задачи остается в лучшем случае ограниченным перебором множества всех маршрутов коммивояжера. Однако с помощью конечного числа операций 1- замены, определяемого соотношением (2), можно получить приближенное решение τ_ϑ за время соизмеримое со временем построения T_ϑ .

Обозначим $n(\vartheta)$ число висячих вершин в дереве $T_\vartheta, \vartheta = \overline{1, n}$. Если в результате нахождения дерева T_ϑ построен маршрут коммивояжера, то он является решением СЗК, т.е. в случае $n(\vartheta)=0, \vartheta \in \{1, 2, \dots, n\}$, получим $d(T_\vartheta)=D(\tau^*), \tau^* = T_\vartheta$.

Алгоритм построения приближенного решения τ_ϑ СЗК состоит из двух стадий. На первой стадии строится дерево T_ϑ . Его стоимость:

$$d(T_\vartheta) = \sum_{\{i, j\} \in T_\vartheta} d_{ij}.$$

На второй стадии выполняется алгоритм преобразования – дерева $T=(V, E_\vartheta)$ в гамильтонов цикл полного взвешенного графа $H_n=(V, E_n)$. Стоимость полученного решения τ_ϑ равна сумме весов ребер, входящих в гамильтонов цикл графа H_n :

$$D(\tau_\vartheta) = d(T_\vartheta) = \sum_{\{i, j\} \in E_\vartheta^-} d_{ij} + \sum_{\{k, l\} \in E_\vartheta^+} d_{kl},$$

где E_ϑ^- множество ребер удаленных из $E_\vartheta, E_\vartheta^+$ – множество ребер графа H_n , добавленных к $E_\vartheta - E_\vartheta^-, |E_\vartheta^-| = |E_\vartheta^+| = n(\vartheta)$.

Так как в любом градиентном алгоритме построения МОД ребро с наименьшим весом d_{\min}

включается в T_ϑ , а в результате выполнения второй стадии оно может быть заменено на ребро с наибольшим весом d_{\max} в H_n , то в худшем случае

$$D(\tau_\vartheta) = d(T_\vartheta) + n(\vartheta)(d_{\max} - d_{\min}) \leq D(\tau^*) + n(\vartheta)(d_{\max} - d_{\min}). \quad (3)$$

Из (3) следует, что предложенный алгоритм обеспечивает невысокую погрешность СЗК в случае незначительного расхождения между элементами матрицы стоимостей, т.е. тогда когда применение метода ветвей и границ связано с предельными объемами вычислительных ресурсов.

Оценим стоимость обхода τ_ϑ , построенного для метрической задачи коммивояжера (МЗК), т.е. СЗК, удовлетворяющей неравенству:

$$d_{ij} \leq d_{ik} + d_{kj}, i \neq j, i, j, k = \overline{1, n}$$

Согласно Теореме 2.2. [1] в МЗК стоимость обхода τ_ϑ удовлетворяет равенству

$$D(\tau_\vartheta) \leq 2D(\tau^*) - 3d_{\min} \quad (4)$$

где τ^* — оптимальное решение, $d_{\min} = \min_{1 \leq i \neq j \leq n} d_{ij}$

Тот факт, что оценка (4) приближенного решения τ_ϑ не улучшаемая, следует из примера МЗК с матрицей стоимостей $[d_{ij}]_n, n = 2k + 1$, в которой,

$$d_{i(i+1)} = 1,$$

$$i = \overline{1, n-2},$$

$$d_{1(n-1)} = 1,$$

$$d_{im} = k - i + 1, d_{in} = i - k,$$

$$i = \overline{k+1, n-1},$$

а все остальные значения $d_{ij}, i \neq j$ равны 2. При $k = 3$

		1	2	3	4	5	6	7
$[d_{ij}]_7 =$	1	∞	1	2	2	2	1	3
	2	1	∞	1	2	2	2	2
	3	2	1	∞	1	2	2	1
	4	2	2	1	∞	1	2	1
	5	2	2	2	1	∞	1	2
	6	1	2	2	2	1	∞	3
	7	3	2	1	1	2	3	∞

На рис. 7, а утолщенными линиями представлено дерево $T_\vartheta, \vartheta = 7$, а на рис. 7, б изображен обход τ_ϑ , стоимость которого $D(\tau_\vartheta) = n - 2 + 2k = 4k - 1$. Но точное решение $\tau^* = (2, 3, 7, 5, 6, 1, 4)$, соответствующее гамильтонову циклу $(7, 4, 5, 6, 1, 2, 3, 7)$, имеет стоимость $D(\tau^*) = n = 2k + 1$ (рис. 7, в) и равенство (4) для приведенного примера выполняется при любом $k \geq 2$.

Алгоритм построения обхода τ_ϑ на стадии преобразования дерева T_ϑ в гамильтонов цикл располагает ресурсами, повышающими точность приближенного решения СЗК.

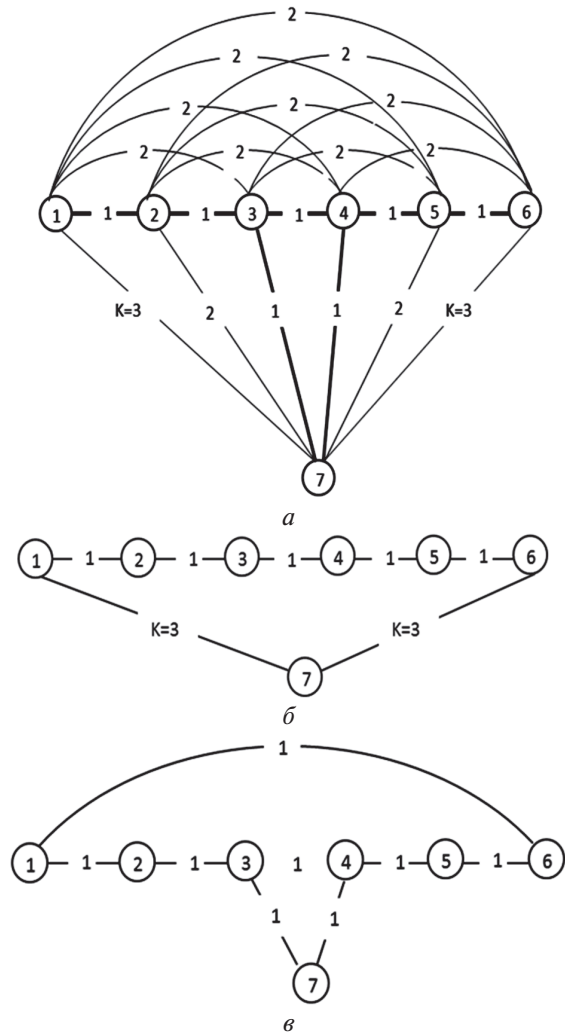


Рис. 7. а — Граф, соответствующий матрице $[d_{ij}]_7$, и дерево $T_\vartheta, \vartheta = 7$;

б — Обход τ_ϑ стоимостью $D(\tau_\vartheta) = 4k - 1$;

в — точное решение τ^* стоимостью $D(\tau^*) = 2k + 1$;

Выводы

В статье рассмотрен алгоритм нахождения гамильтонова цикла по i -дереву $T = (V, E_i)$ полного графа $H_n = (V, E_n)$, как один из этапов построения приближенного решения СЗК, с оценкой погрешности меньшей, чем у большинства приближенных алгоритмах.

Из приведенных расчетов следует, что полученная оценка для данного алгоритма является не улучшаемой, а алгоритм преобразования i -дерева в гамильтонов цикл располагает ресурсами, которые повышают точность и быстродействие приближенного решения симметричной задачи коммивояжера.

Список литературы:

1. Панишев А.В. Модели и методы оптимизации замкнутых маршрутов на транспортной сети / А.В. Панишев, А.В. Морозов. — Житомир: ЖГТУ, 2014. — 316 с.
2. Пападимитриу Х. Комбинаторная оптимизация. Алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. — М.: Мир, 1985. — 510 с.

Поступила до редколлегии 10.05.2017