DOI:

АЛГОРИТМЫ ФОРМИРОВАНИЯ ОПТИМАЛЬНЫХ ДРЕВОВИДНЫХ СЕТЕЙ

Злотов А.В.

Федеральный исследовательский центр «Информатика и управление» РАН

Ул. Вавилова, д. 44

Zlot_a@mail.ru

Аннотация: В докладе рассматриваются точный и приближенный алгоритмы построения оптимальных сетей с разрывной функцией стоимости в зависимости от потока на ребрах. Установлены свойства оптимального решения задачи, описан алгоритм формирования всех деревьев и однокорневых поддеревьев графа, на базе которого построен алгоритм направленного перебора для поиска оптимального и приближенных решений задачи. Описаны алгоритмы получения приближенного решения и его корректировки.

Ключевые слова: алгоритм генерации деревьев разрывные функции стоимости ребер

Введение

В докладе рассматривается задача с разрывными функциями стоимостями на ребрах, приводится постановка задачи и ее частные случаи, описан специальный алгоритм полного перебора всех деревьев и промежуточных однокорневых поддеревьев полного графа, основанный на сформулированных ***Правиле расстановки пометок*** и ***Правиле подключения.*** Для формирования точного решения задачи в процессе работы алгоритма перебора деревьев и поддеревьев используются правила отбраковки, основанные на свойствах оптимального решения, заведомо неоптимальных решений, в результате чего перебор значительно сокращается.

1 Постановка задачи построения сети с разрывными функциями стоимости на ребрах

Пусть задано множество источников сырья $J=\{1,2,3,...,n\}$ с известными объемами сырья $b_j>0$ и сток q_0. На множестве $W=J\cup\{q_0\}$ задан полный граф возможных коммуникаций $U(W)$. Стоимость соединения двух произвольных вершин этого графа $c_{ij}(x_{ij})$ является разрывной функцией, зависящей от величины потока сырья x_{ij} между этими вершинами:

$c_{ij}(x_{ij})=(v_{ij}+u_{ij}*x_{ij}) * \mathrm{sign}(x_{ij})$ где v_{ij}, $u_{ij} \geq 0$ некоторые постоянные коэффициенты.

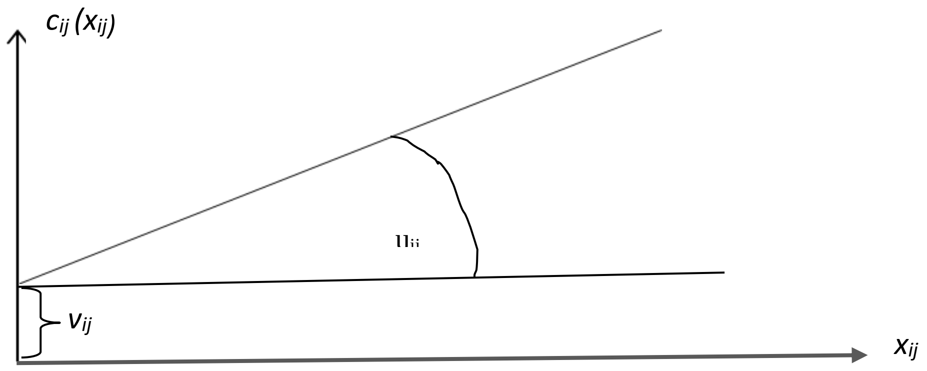

Рис. 1. Функция стоимости ребра сети в зависимости от потока по этому ребру

Требуется построить сеть минимальной стоимости, связывающую множество источников J со стоком q_0, при условиях полного перетока сырья из источников в сток. То есть, необходимо минимизировать величину:

$$(1)\sum_{i\in J}\sum_{j\in W}(\upsilon_{ij}+u_{ij}\cdot x_{ij})\cdot sign(x_{ij})\to\min$$

при условиях

$$(2)\sum_{i\in W}\left(x_{ij}-x_{ji}\right)=b_j \qquad j=1,2,3,...n$$

$$(3) x_{ij}\geq 0$$

Задача в данной постановке характерна для решения ряда прикладных задач построения различных коммуникационных сетей: трубопроводных, транспортных, сетей связи и других, когда стоимость коммуникации складывается из постоянных затрат v_{ij} на ее строительство и «эксплуатационных» затрат u_{ij}, зависящих от величины потока по данной коммуникации.

Если все $v_{ij}=0$ то Задача 1 сводится к известной задаче отыскания кратчайших путей по матрице $\|u_{ij}\|$ от множества источников J до стока q_0, для решения которой разработан ряд эффективных алгоритмов [1,2]. Если все $u_{ij}\equiv 0$, то Задача 1 представляет собой задачу построения кратчайшей связывающей сети (КСС) на множестве W, алгоритмы решения которой также хорошо разработаны [3,4].

В общем же случае, когда $v_{ij}\neq 0$ и $u_{ij}\neq 0$, данная задача представляет собой многоэкстремальную задачу дискретного программирования, которая является частным случаем сетевой постановки транспортной задачи с фиксированными доплатами.

Введем некоторые обозначения.

Пусть $J_s=(i_1,i_2,i_3,...,i_s)$ - некоторое подмножество источников $J_s\in J$ и на множестве $W_s = J_s\cup\{q_0\}$ построено поддерево R_S.

$P_s=W \setminus W_s$ - множество не присоединенных вершин;

μ_i^s - путь по поддереву R_S от вершины i до стока q_0;

$l_{i0}^s = \sum_{(i,j)\in\mu_i^s} u_{ij}$ - "длина пути" μ_i^s по поддереву R_S;

$\underline{u}_{ij}$ - "длина кратчайшего пути" между вершинами i и j по матрице $\|u_{ij}\|$;

$$\underline{u}_{j0} = \begin{cases} l_{j0}^s & при \quad j\in J_s \\ \tilde{u}_{jq_0} & при \quad j\in P_s \end{cases}$$

x_i - значение потока через i - *ую* вершину;

y_{ij} - значение потока по звену (ij)

$D_s=\{d_k^s\}$ - множество всех деревьев d_k^s, таких, что $R_s\in d_k^s$;

$\psi(d_k^s) = \sum_{(i,j)\in d_k^s}(v_{ij}+u_{ij}*y_{ij})$ - стоимость дерева d_k^s.

<u>Требуется определить</u> d_0^s - оптимальное решение Задачи 1 при зафиксированном поддереве $R_s\in d_0^s$: $\psi(d_0^s) = \min_{d_k^s\in D_s}\psi(d_k^s)$.

2 Анализ свойств оптимального решения

При анализе свойств оптимального решения предполагается, что известно некоторое зафиксированное поддерево $R_s \in d_0^s$ и необходимо достроить данное поддерево оптимальным образом. В качестве поддерева R_0 может выступать одиночная вершина-сток q_0.

2.1 Функция подключения

Определим функцию $F_i(x_i,w,R_s)$, как ***«функцию подключения»*** i-ой вершины к подмножеству вершин $w\subset W$.

$$(4)\quad F_i(x_i,w,R_s)= \begin{cases} v_{ij}+(u_{ij}+l_{j0}^s)\cdot x & при \quad ij\in R_s, i\in W_s \\ \min_{j\in w\setminus i}\left[v_{ij}+\left(u_{ij}+\tilde{u}_{j0}\right)\cdot x\right] & при \quad i\in P_s \end{cases}$$

Функция подключения $F_i(x_i,w,R_s)$ обладает следующими свойствами:

1. $F_i(x_i,w,R_s)$ - монотонно-возрастающая, кусочно-линейная функция по x.

2. $\sum_{i\in J} F_i(b_i,W,d_k)\equiv P(d_k)$, то есть $F_i(b_i,w,d_k)$ является членом разложения стоимости дерева d_k, соотнесенным к i-*ой* вершине.

3. Функция подключения является нижней оценкой стоимости всех деревьев, построенных при зафиксированном поддереве Rs:

$$(5)\quad \sum_{i\in J}F_i(b_i\mathrm{W},R_s)\le P(d_k) \quad при \quad R_s\in d_k$$

2.2 Теорема 1.   О недопустимом пути

Если для каких-либо вершин *i* и *j* $\in P_s$ выполняется условие $F_i(b_i, W_s, R_s) < \min\limits_{k\in W}(v_{ik}) + (\tilde{u}_{ij} + \tilde{u}_{j0})\cdot b_i$ (6) то в оптимальном решении, построенном на зафиксированном поддереве $R_s \in d_0^s$, путь из вершины *i* к стоку q_0 не может проходить через вершину *j*.

Следствие. На основании этого свойства можно оценить значение максимально возможного потока x_{jm}, который может проходить через *j-ую* вершину в решении d_0^s: $\underline{x}_{jm} = \sum_{i\in P_j'} b_i$, где P_j' - множество тех вершин $i \in P_s$ для которых не выполняется условие (6).

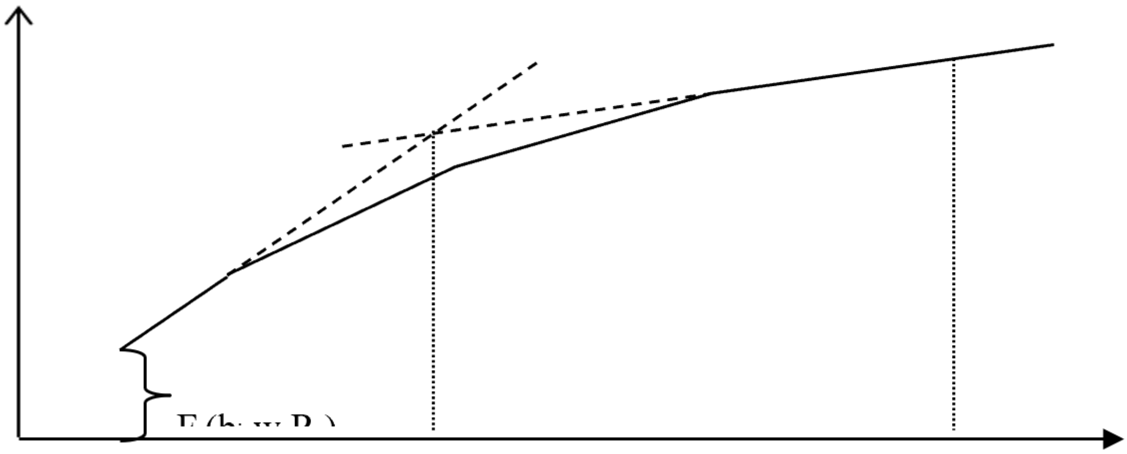

Рис. 2. Функция подключения i-ой вершины в зависимости от потока.

2.3 Теорема 2. Достаточные условия оптимальности подсоединения

Для того, чтобы ребро (i,j) $i\in P_s$, $j\in W_s$ входило в решение d_0^s достаточно выполнения условий

$$(7)\quad \begin{cases} F_i(b_i, W, R_s) = v_{ij} + (u_{ij} + l^s_{j0})\cdot b_i \\ F_i((\underline{x}_{im}, W, R_s) = v_{ij} + (u_{ij} + l^s_{j0})\cdot \underline{x}_{im} \end{cases}$$

2.4 Теорема 3. Необходимое условие вхождения подсоединения в оптимальное решение

Для того, чтобы ребро (i,j) $i\in P_s$, $j\in W_s$ могло входить в решение d_0^s, необходимо выполнение условия

$$v_{ij}+(u_{ij}+l^s_{j0})\cdot x^*_{ip} \le v_{i j1}+(u_{i j1}+l^s_{j1\,0})\cdot x^*_{ip}$$

$$(8)\quad где\ x^*_{ip} = \begin{cases} \dfrac{v_{i j_1} - v_{i j_2}}{u_{i j_2} + l^s_{j_2 0} - u_{i j_1} - l^s_{j_1 0}} & при \quad j_1 \neq j_2 \\ b_i & при \quad j_1 = j_2 \end{cases}$$

а j_1 и j_2 - вершины, на которых достигается минимум функций $F_i(b_i,W_s,R_s)$ и $F_i(\underline{x}_{im},W_s,R_s)$.

2.5 Теорема 4. Об оценке подсоединения

Погрешность оптимального решения с зафиксированным поддеревом $R_{s1} = R_s \cup (ij)$, где $i\in P_s$, $J \in W_s$ по сравнению с оптимальным решением при зафиксированном поддереве R_s не превосходит значения $\partial_{ij} = \max\{[v_{ij} + (u_{ij} + l^s_{j0})\cdot b_i - F_i(b_i, W, R_s)],[v_{ij} + (u_{ij} + l_{j0})\cdot \underline{x}_{im} - F_i(\underline{x}_{im}, W, R_s)]\}$ (9)

3 Алгоритм полного перебора всех деревьев и промежуточных однокорневых поддеревьев

Алгоритмы точного решения задачи строятся на базе алгоритма полного перебора всех деревьев и промежуточных однокорневых поддеревьев полного графа *U(W)*. Алгоритм перебора должен удовлетворять следующим требованиям:

1. Полный перебор без повторов не только всех деревьев, построенных на множестве вершин *W,* но и всех промежуточных поддеревьев с корнем в зафиксированной вершине q_0.
2. Конструктивность способа формирования деревьев и поддеревьев путем последовательного "наращивания" ребер.

Ниже описывается алгоритм перебора, удовлетворяющий перечисленным требованиям и реализованный в алгоритмах точного решения задачи.

В процессе перебора последовательно строятся поддеревья различных "уровней". На *S-ом* уровне $1 \le S \le n+1$ формируются деревья R_s, содержащие *S* вершин.

Процесс построения деревьев осуществляется последовательным движением вдоль ряда поддеревьев: производится переход от уровня к уровню (от 1 *-го* и до *N+1 -го*). Причем поддеревья верхнего уровня отличаются от образующего поддерева нижнего уровня добавлением только одного ребра. На каждом *S-ом* уровне образуются не сразу все поддеревья из *S* вершин, а лишь некоторая, однозначно определенная алгоритмом, часть их. После того, как будет получена часть деревьев *n+1 -го* уровня, происходит возврат к *n -му* уровню и из него образуются новые деревья *n+1 -го* уровня. Если образование новых деревьев *n+1 -го* уровня окажется невозможным, то производится возврат к *n-1-му* уровню и образование из него новой *n -ой*, а затем и *n+1 -ой* группы деревьев.

Процесс заканчивается, когда просмотрены все поддеревья 2-го уровня.

Опишем переход от *r-го* уровня к *r+1 -му* уровню.

Пусть из *r-1 -го* образован *r-ый* и из него необходимо образовать *r+1-ый* уровень.

На *r-ом* уровне имеется множество вершин $J_r=\{j_1^r,j_2^r,...,j_r^r\}$, образующих поддерево R_r. Этому J_r соответствует множество пометок $E_r=\{e_1^r,e_2^r,...,e_r^r\}$. Остальные вершины образуют множество не присоединенных вершин $P_r=\{p_1^r, p_2^r,...,p^r_{n+1-r}\}$.

Алгоритм перебора основывается на ***«Правиле подключения» и «Правиле расстановки пометок»***, которые формулируются следующим образом:

Правило подключения: Произвольная вершина $p_i^r \in P_r$ может быть подсоединена к вершине j_k^r поддерева R_r только в том случае, если ее номер больше пометки вершины j_k^r: $p_i^r > e_k^r$.

Правило расстановки пометок: При подключении произвольной вершины $p_i^r \in P_r$ к вершине j_k^r поддерева R_r она получает пометку, равную 0. Все вершины по пути от данной вершины к корню поддерева получают пометку, равную номеру вершины, подсоединенной к данной, по этому пути. Остальные вершины получают пометку, равную некоторому большому числу M > n.

Путь от последней подключенной вершины до корня дерева будем называть ***«генерирующим путем».***

В начальный момент работы алгоритма $r=1, J_1 \equiv q_0, P \equiv J, E_1=\{0\}$.

Пусть $r=n+1$. Это означает, что построено очередное дерево. В этом случае переходят к *n-ому* уровню и пытаются образовать из него новый *n+1-ый* уровень. Если это невозможно, то переходят к *n-1 -му* уровню для образования из него нового *n-го* уровня и т.д.

Опишем этот процесс в общем виде.

Пусть из *r+1 -го* уровня оказалось невозможным образование *r+2 -го* уровня. В этом случае производится переход к *r -му* уровню для образования из него нового *r+1 -го* уровня. На *r -ом* уровне для вершины p_i^r среди меток с индексом $k > k_r$ ищется метка $e > p_{i1}^r$.

Если такая метка найдена, то соответствующий ей номер фиксируется в качестве нового значения k_r и производится переход к *r+1 -му* уровню.

Если для $p^r_{i_i}$ не было найдено такой метки, то рассматривается следующая вершина $p^r_{i_i+1}$, значение k_r полагается равным нулю и опять ищется метка *e* с $e < p^r_{i_i+1}$.

Если i_1+1 станет равным *n - r,* то есть просмотрены все вершины множества p_r, производится возврат к *r-1 -му* уровню для образования из него нового *r -го* уровня.

Процесс заканчивается, когда из начального уровня *r=1* не удается построить следующий уровень.

Данный алгоритм реализует полный перебор без повторов всех деревьев и однокорневых поддеревьев на множестве вершин *W*.

Правило формирования следующего уровня путем подсоединения одного ребра служит конструктивным приемом построения ряда поддеревьев, упорядоченных по включениям $R_1 \subset R_2 \subset \ldots R_n \subset R_{n+1}$, что облегчает их анализ и отбраковку в алгоритме точного решения.

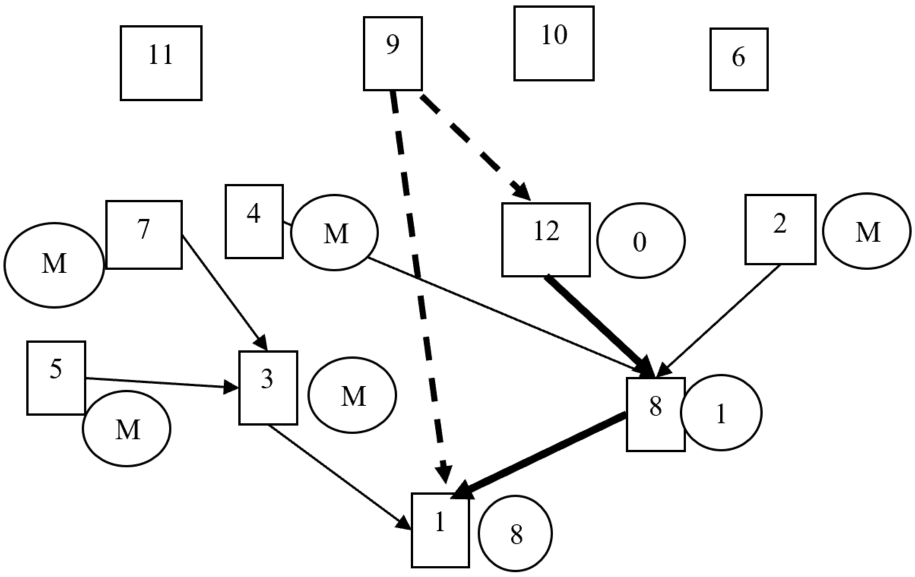

Рис. 3. Иллюстрация работы алгоритма перебора

Здесь: {1, 2, 3, 4, 5, 7, 8, 12} – вершины поддерева;

J_s ={1,2,3,4,5,7,,8,12}, P_s = {6,9,10,11}

{8,M,M,M,M,M,12,0}-пометки, где M–большое число M>12; {12} – последняя подсоединенная вершина; {1, 8, 12}- генерирующий путь;

Ребра (9,12), (9,1) – возможные подключения 9-*ой* вершины к генерирующему пути.

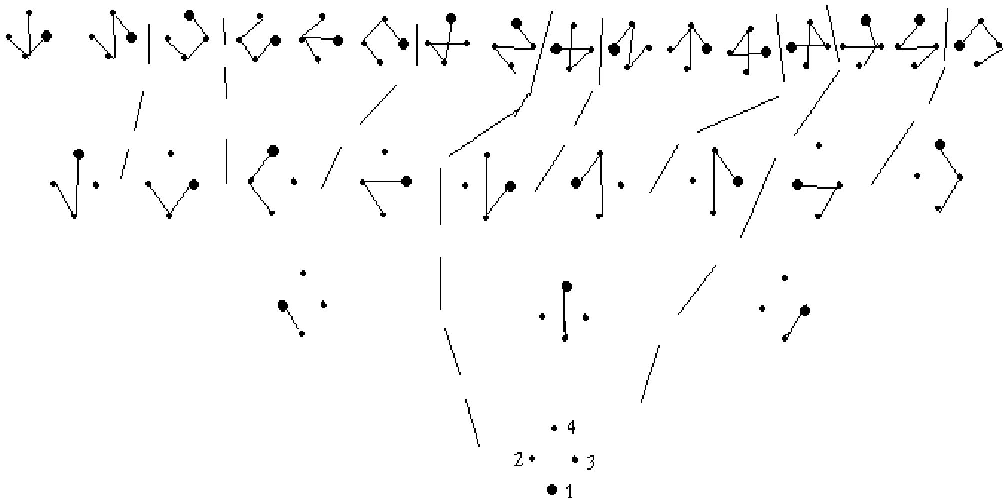

Рис 4. Все деревья и однокорневые поддеревья для четырех узлов

4 Алгоритмы построения оптимальных древовидных сетей с разрывными функциями стоимости на ребрах

Для решения этих задач разработаны следующие основные алгоритмы:

1. Алгоритм формирования начального поддерева.
2. Алгоритм полного перебора всех деревьев и промежуточных однокорневых поддеревьев *(описанный в предыдущем разделе)*.
3. Алгоритм точного решения задачи построения сети (1)-(3).

4. Модифицированный алгоритм отыскания точного решения.
5. Алгоритм получения начального приближения.
6. Алгоритм корректировки начального приближения.

4.1 Алгоритм формирования начального поддерева R_0

Этот алгоритм используется как в точном, так и в приближенном алгоритмах решения задачи и служит для формирования поддерева R_0 с корнем в стоке q_0 , которое заведомо входит в оптимальное решение задачи $d_0 : R_0 \subset d_0$. Для некоторых задач это алгоритм может сразу построить оптимальное решение $R_0 \equiv d_0$. В частности, точное решение получается в предельных случаях, - в задаче построения кратчайшей связывающей сети $(u_{ij} \equiv 0)$ и в задаче отыскания маршрутов наименьшей стоимости $(v_{ij} \equiv 0)$.

Алгоритм построения начального поддерева основывается на правилах, при помощи которых на каждом шаге алгоритма определяется очередное звено (i,j), добавляемое к уже построенной части поддерева.

Предварительно по матрице $||u_{ij}||$ вычисляется матрица «кратчайших расстояний» $||\widetilde{u}_{ij}||$. Далее на каждом шаге алгоритма *пока это возможно* для всех не подсоединенных вершин по правилам (6), определяется вектор максимально возможных потоков $||x^r_{im}||$ и в случае выполнения для какой-либо вершины условия оптимальности (7) она подключается к уже сформированному поддереву.

4.2 Алгоритмы точного решения задачи

Алгоритм точного решения задачи основан на направленном переборе всех деревьев полного графа. В процессе перебора используется ряд правил отбраковки, позволяющих не рассматривать большие группы деревьев и промежуточных поддеревьев, в результате чего перебор значительно сокращается. Правила отбраковки основываются на свойствах оптимального решения задачи, описанных в п. 2. В случае наличия ограничений по объему используемой оперативной памяти или времени решения задачи, необходимо применение алгоритмов приближенного решения. Эти алгоритмы описываются в следующем разделе.

При работе алгоритма точного решения задачи используется модификация метода перебора для того, чтобы просматривать все деревья и поддеревья при зафиксированном поддереве R_0, которое формируется по алгоритму построения начального поддерева R_0. Необходимая модификация состоит в том, что на каждом шаге алгоритма перебора при расстановке меток e_i^{r+1}, всем вершинам поддерева R_0 приписывается одинаковая метка, равная номеру вершины, подсоединяемой к поддереву R_0 по выделенному пути от вершины p_i до стока.

Основным аппаратом для отбраковки в алгоритме точного решения является правило отбраковки, основанное на условии допустимости подсоединения (8).

Кроме того, используется правило отбраковки тех поддеревьев R_s ,для которых нижняя оценка стоимости оптимального решения на данном поддереве - $\psi\left(R_s\right) = \sum_{i \in J} F_i\left(b_i, W, R_s\right)$ больше, чем стоимость временно оптимального решения d_1 - $\varphi(d_1)$. Для допустимых подсоединений $\varphi(d_1) \geq \psi(R_s)$.

Эффективность данного правила отбраковки зависит от того, насколько временно оптимальное решение d_1 близко по стоимости к оптимальному решению d_0. Поэтому в качестве начального значения для $\varphi(d_1)$ целесообразно брать решение, получаемое по алгоритму приближенного решения, которое затем может быть улучшено в процессе работы алгоритма точного решения.

Работу алгоритма точного решения можно представить в виде последовательности следующих этапов:

1. Вычисление матрицы кратчайших расстояний $||\widetilde{u}_{ij}||$ по матрице $||u_{ij}||$.
2. Построение начального поддерева R_0 по алгоритму построения начального поддерева.
3. Определение приближенного решения d_1 и его стоимости $\varphi(d_1)$ по алгоритму получения приближенного решения.
4. Получение оптимального решения.

4.3 Алгоритм приближенного решения задачи

Алгоритм приближенного решения можно представить, как последовательность алгоритмов формирования начального поддерева, построения начального приближения и корректировки начального приближения.

Использование совокупности этих алгоритмов позволяет получать хорошее приближенное решение с оценкой абсолютной погрешности получаемого решения.

4.4 Алгоритм построения начального приближения

Данный алгоритм работает после алгоритма формирования начального поддерева. Построение приближенного решения проводится последовательным «наращиванием» уже построенной части дерева путем добавления к нему такого звено (i,j), $i \in P_r$, $j \in W_r$, для которого минимальна величина δ_{ij} оценки погрешности на его включение в оптимальное решение, определяемая по формуле *(9)*.

4.5 Алгоритм корректировки начального приближения

В данном алгоритме для всех вершин множества J проверяется возможность их переключения с одновременным уменьшением стоимости решения. Переключение производится для вершины, дающей наибольшее уменьшение функционала. Работа алгоритма заканчивается, когда такого переключения нельзя сделать ни для одной вершины.

В результате последовательной реализации алгоритмов формирования начального поддерева, построения начального приближения и его корректировки будет найдено приближенное решение $d_1 \in D$. Погрешность полученного решения по сравнению с оптимальным решением не превосходит величины

$\Delta\Phi = \sum_{i \in J}\left[F_i(b_i, W, d_1) - F_i(b_i, W, R_0)\right]$, то есть равняется разности между величиной функционала полученного решения и нижней оценки оптимального решения после сформирования начального поддерева.

4.6 Система размещения объектов и коммуникаций

Описанные алгоритмы реализованы в Системе размещения объектов и коммуникаций, которая применяется для решения задач размещения совместно со связывающими коммуникациями. На рисунке 5. Показан пример применения этой системы для Тенгизского месторождения.

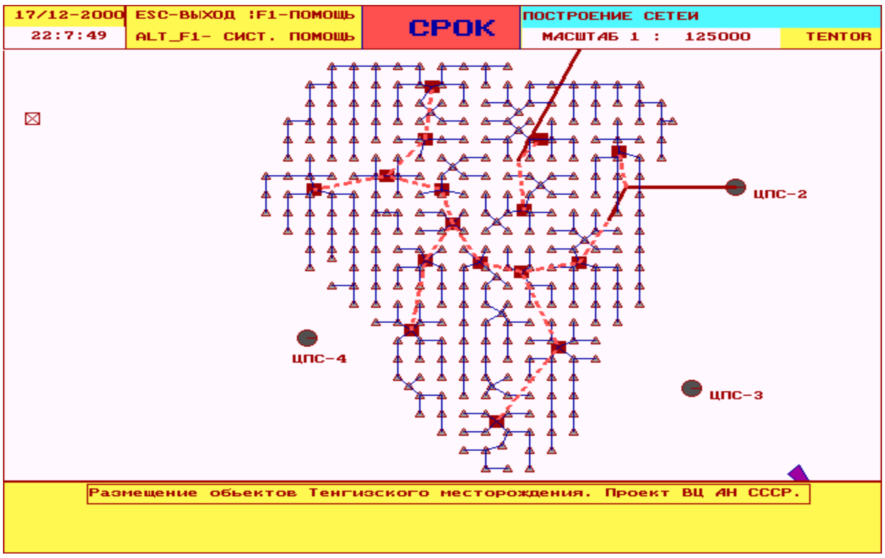

Рис. 5. Тенгизское нефтяное месторождение. Проект ВЦ РАН

Литература

1. *Берж.* Теория графов и ее применение – М: ИЛ., 1962.
2. *Ермольев Ю.М., Мельник Н.М.* Экстремальные задачи на графах.:- г. Киев, «Наукова думка», 1968.
3. *Прим Р.К.* Кратчайшие связывающие сети и некоторые обобщения. - М.: ИЛ. «Кибернетический сборник». 1961, №2
4. *Кельманс А.К.* О построении кратчайшей связывающей сети.
5. Сб. «Кибернетика и управление», М., 1967.
6. *Оре О.* Теория графов – М.:«Наука»., 1968