ГОСУДАРСТВЕННЫЙ КОМИТЕТ РСФСР ПО ДЕЛАМ НАУКИ И ВЫСШЕЙ ШКОЛЫ
КАБАРДИНО - БАЛКАРСКИЙ ОРДЕНА ДРУЖБЫ НАРОДОВ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Байрактаров Бахтияр Риза-оглы
Bahtiyar BAYRAKTAROV

УДК 626.923.2.001.2+631.413.3

Моделирование и расчет оптимальных параметров сетей водоснабжения

Специальность 05.13.16 - Применение вычислительной техники, математического моделирования и математических методов в научных исследованиях (информатика, вычислительная техника и автоматизация)

Диссертация на соискание ученой степени
кандидата технических наук

Соискатель _____

Научный руководитель, д. ф. м. н.,
профессор _____
Шхануков Мухамед Хабалович

НАЛЬЧИК - 1992

С О Д Е Р Ж А Н И Е

ВВЕДЕНИЕ 5

1. ОПИСАНИЕ ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ СИСТЕМ ВОДОСНАБЖЕНИЯ

1.1. Вводная часть и обзор литературы.....15

1.2. Описание задачи технико-экономического и гидравлического расчета разветвленных водопроводных сетей 31

1.3. Описание задачи технико-экономического и гидравлического расчета кольцевых сетей.....34

2. ЗАДАЧА ТЕХНИКО-ЭКОНОМИЧЕСКОГО И ГИДРАВЛИЧЕСКОГО РАСЧЕТА РАЗВЕТВЛЕННЫХ ВОДОПРВОДНЫХ СЕТЕИ

2.1. Математическая модель задачи технико - экономического и гидравлического расчета закрытых оросительных сетей 37

2.2. Метод и алгоритм решения задачи технико-экономического и гидравлического расчета закрытых оросительных сетей 39

2.3. Описание программы задачи технико - экономического и гидравлического расчета закрытых оросительных сетей 48

2.4. Математическая модель задачи технико - экономического и гидравлического расчета магистральных водопроводов.....51

2.5. Метод и алгоритм решения задачи технико-эконо-

мического и гидравлического расчета магистральных водопроводов.....54
2.6. Описание программы задачи технико - экономического и гидравлического расчета магистральных водопроводов.....59
3. ЗАДАЧА ТЕХНИКО-ЭКОНОМИЧЕСКОГО И ГИДРАВЛИЧЕСКОГО РАСЧЕТА КОЛЬЦЕВЫХ СЕТЕЙ
3.1. Математическая модель задачи технико - экономического и гидравлического расчета кольцевых сетей.....65
3.2. Метод и алгоритм решения задачи технико -экономического и гидравлического расчета кольцевых сетей.....69
3.2.1. Определение узловых расходов сети.....69
3.2.2. Определение первоначальных значений потоков и потерь напоров на сети.....70
3.2.3. Минимизация функционала.....76
3.3. Описание программы задачи технико - экономического и гидравлического расчета кольцевых сетей.....78
ЗАКЛЮЧЕНИЕ.....83
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....85
ПРИЛОЖЕНИЕ 1. Акт приемки-сдачи научно-исследовательской работы по теме № 235...94

ПРИЛОЖЕНИЕ 2. Распечатки контрольных примеров программы задачи ТЭ и ГР ЗОС.....99

ПРИЛОЖЕНИЕ 3. Распечатки иисходных текстов программ ТЭ и ГР ЗОС.....113

ПРИЛОЖЕНИЕ 4. Распечатки контрольных примеров программы задачи ТЭ и ГР МВ.....142

ПРИЛОЖЕНИЕ 5. Распечатки исходных текстов программы ТЭ и ГР МВ.....150

ПРИЛОЖЕНИЕ 6. Распечатки контрольных примеров программы задачи ТЭ и ГР КС.....186

ПРИЛОЖЕНИЕ 7. Распечатки исходных текстов программы ТЭ и ГР КС.....207

ПРИЛОЖЕНИЕ 8. Таблица нормативно - справочной информации по трубам.....264

ВВЕДЕНИЕ

Мелиорация земель является одним из основных направлений научно-технического прогресса в сельском хозяйстве, важнейшим элементом комплекса мероприятий по интенсификации сельскохозяйственного производства.

Главной задачей в разработке мелиоративных систем является повышение эффективности капитальных вложений и качества строительства. Повышение эффективности строительства во многом зависит от организации проектной деятельности. Объем проектно-изыскательских работ увеличивается с каждым годом и такая тенденция сохранится и в будущем. В то же время, число занятых в проектировании специалистов не может расти пропорционально строительным объемам. Следовательно, выполнение дополнительных объемов должно происходить за счет повышения производительности труда. При этом качество и эффективность проектных решений должны резко возрастать, а сроки их разработки сокращаться. Это возможно при условии дальнейшего совершенствования проектного дела и использования современной вычислительной техники. Это, в свою очередь, предполагает математическое моделирование систем водоснабжения (либо отдельных ее компонентов) с целью получения оптимальных, технически совершенных мелиоративных и водохозяйственных систем.

Настоящая работа посвящена проблемам создания систем

автоматизированного проектирования работ (САПР) для проектирования систем сельхозводоснабжения.

При проектировании водопроводов полный технико-экономический расчет систем подачи и распределения воды осуществляется в три этапа:

1. Устанавливается наивыгоднейшая конфигурация трассы трубопроводов;
2. Выбирается наилучший вариант распределения потоков;
3. Определяется экономически наивыгоднейшие диаметры труб трубопроводов.

В работе рассматриваются задачи второго и третьего этапов, которые позволяют существенно (порядка до 60%) снижать стоимость системы водоснабжения в целом. Поэтому разработка и внедрение методов решения этих задач имеют большое народнохозяйственное значение.

Целью работы является разработка математических моделей, алгоритмов и программного обеспечения для решения этих задач.

Рассматриваемые в работе задачи решались в рамках хоздоговорной темы № 235 "Разработка систем автоматизированного проектирования работ(САПР I и II-ой очереди) для систем сельхозводоснабжения и магистральных водопроводов" в течении 1977-1985 гг, заключенной между СКГВХ (Северокавказский государственный институт по проектированию водохозяйственного и мелиоративного строительства) и Отраслевой научно-исследовательской лабораторией Министерства водного хо-

зяйства РСФСР при Кабардино-Балкарском государственном университете.

В диссертации рассматриваются основные оптимизационные задачи, возникающие в процессе проектирования разветвленных и кольцевых водопроводных сетей.

Проблемы оптимального проектирования, естественно, подразделяются на следующие задачи :

1. Задача оптимизации параметров разветвленной водопроводной сети с одним источником давления в начале сети, которая состоит в таком определении диаметров , толщин стенок, материала труб по участкам и напоров в узлах сети, чтобы выполнялись все технические требования и ограничения при минимальной приведенной стоимости сети ;
2. Задача оптимизации параметров разветвленной сети с выбором мест расположения насосных станций в узлах сети .

В этой задаче, дополнительно, требуется определить места размещения насосных станций и напоры на них таким образом, чтобы приведенные затраты на сеть были минимальными ;

3. Задача оптимизации параметров поселковой водопроводной сети .

Как известно, для бесперебойного обеспечения потребителей используют закольцевание трубопроводной сети. В связи с этим возникает задача такого определения направлений, величин потоков по участкам сети, технических параметров трубопровода (диаметров, толщин стенок и материала труб) и напоров в узлах, чтобы затраты на сеть были минимальными.

Необходимость в разработке этих задач возникла в связи с созданием единой САПР Минводхоза РСФСР по проектированию систем водоснабжения.

Постановщиком задач выступал головной проектной институт министерства - СКГВХ в городе Пятигорске.

В основе решения каждой из задач лежит метод решения задачи оптимизации параметров разветвленной водопроводной сети при фиксированном напоре. Доказано утверждение, что оптимальным является конструирование любого участка сети из труб не более, чем двух диаметров из задаваемого сортамента .

Так приходим к задаче кусочно-линейного выпуклого программирования размерности 2(n-1), где n - число узлов сети :

$$З = \alpha \sum_{i=2}^{n} C(h_i) l_i \longrightarrow \min \quad (1)$$

при ограничениях

$$\begin{cases} V_{min} \le V_i \le V_{max} \\ H_j = H^* - \sum_{i \in T_j} h_i l_i + Z_i - Z_j \ge H_j^{св} \end{cases}, \quad (2)$$

где $j = 1,2,\ldots, n$; α - нормативный коэффициент;

T_j - множество номеров участков , входящих в траекторию от насосной станции до вершины с номером j;

H^* -напор на насосной станции;

h_i -потери напора на единице длины участка;

l_i -длина i-го участка;

V_{min}, V_{max} – диапазон изменения скорости течения воды;

Z_j и $H_j^{св}$ -соответственно, высотная отметка и требуемый свободный напор на j-ом узле;

$C(h_i)$-стоимость единицы длины участка, причем

$$C(h_i) = \lambda_i^* c_p + (1-\lambda_i^*) c_m , \qquad (3)$$

где

$$\lambda_i^* \in [\ 0,1\ ] \text{ и } \lambda_i^* = \frac{h_i^m - h_i}{h_i^m - h_i^p} ;$$

p и m - номера диаметров труб из заданного сортамента для которых $h_i^p \le h_i \le h_i^m$;

c_p и c_m – стоимости единицы длины диаметров труб;

Такой подход дает возможность осуществить рабочее проектирование сетей, т.е. таблицы входных и выходных данных всех задач входят в состав конечной проектно -сметной документации объектов проектирования.

В первой главе диссертации приводится описание задач, их постановки на содержательном уровне и перечень входных данных и результатов по каждой из задач.

Во второй главе рассматриваются решения задач технико--экономического и гидравлического расчета закрытых оросительных сетей (ТЭ и ГР ЗОС) и магистральных водопроводов (ТЭ и ГР МВ): математические модели , методы и алгоритмы решения задач, программное обеспечение, реализующее эти алгортмы.

В основу решения задачи ТЭ и ГР ЗОС положен метод

покоординатного спуска к оптимуму. Решение задачи для сетей, содержащих около 100 вершин, составляет около одной минут на ЭВМ типа ЕС - 1045.

Математическая модель задачи ТЭ и ГР МВ имеет следующий вид:

$$З = \alpha \left[\sum_{i=2}^{n} C(h_i)\, l_i + \sum_{m \in M} A(H_m^*) \right] + \beta \sum_{m \in M} (q_m - Q_m)\, H_m^* \longrightarrow \min \qquad (4)$$

при ограничениях :

$$\begin{cases} H_j = H_m^* - \sum\limits_{i \in T_{mj}} h_i l_i + Z_i - Z_j \geq H_j^{CB} \\ V_{min} \leq V_i \leq V_{max} \end{cases}, \qquad (5)$$

где

q_i - расход воды на i-ом участке трубопровода;

Q_j - расход воды в j-ом узле ; H_j - напоры в узлах ;

H_m^* - напор на насосной станции на вершине m;

$M = \{ M_1, M_2, M_3 \}$ - множество задаваемых номеров вершин, на которых возможно поставить насосные станции подкачки, либо перекачки (M_1, M_2 и M_3 - соответственно, множества номеров вершин на которых возможно поставить только станции перекачки, только станции подкачки и либо перекачки, либо подкачки);

T_{mj} - множество номеров участков, входящих в траекторию от насосной станции на вершине m до вершины с номером j ;

где $i = 2,\ldots,n$, $j = 1,2,\ldots,n$, $m \in M$.

Для решения задачи ТЭ и ГР МВ применяется метод динамического программирования.

Конкретно, оптимизация по расстановке насосных станций каждый раз проводится на малой части всей сети, ограниченной тупиковыми узлами или узлами, являющимися корневыми вершинами кустов, оптимизированных на предыдущих итерациях.

Рекурентное соотношение Беллмана при этом зависит от типа устанавливаемой насосной станции (подкачка или перекачка, или отсутствует) и оптимального напора в граничных по отношению к рассматриваемому кусту сети узлах.

Если $S_m(H_m^k)$ - наименьшая стоимость куста m при условии, что на m-ой насосной станции напор равен H_m^k и на тупиковых вершинах (i1,i2,...,il) которого имеются вертикали (дискретные значения напоров на вершинах m) с известными стоимостными характеристиками, тогда

$$S_m(H_m^k) = \min \left\{ \sum_{k1=1}^{p} \alpha_{i1}^{k1} S_{i1}(H_{i1}^{k1}) + \sum_{k2=1}^{p} \alpha_{i2}^{k2} S_{i2}(H_{i2}^{k2}) + \ldots + \right.$$

$$\left. + \sum_{kl=1}^{p} \alpha_{il}^{kl} S_{il}(H_{il}^{kl}) + W_m(H_m^k, H_{i1}^{k1}, H_{i2}^{k2}, \ldots, H_{il}^{kl}) \right\}, \quad (6)$$

где

$$\sum_{k1=1}^{p} \alpha_{i1}^{k1} = 1, \quad \sum_{k2=1}^{p} \alpha_{i2}^{k2} = 1, \ldots, \sum_{kl=1}^{p} \alpha_{il}^{kl} = 1;$$

$$\alpha_{i1}^{k1}, \alpha_{i2}^{k2}, \ldots, \alpha_{il}^{kl} \in \{0,1\};$$

$$W_m(H_m^k) = \min_{\tau} \left\{ \alpha \left[\sum_{i \in J_m} C(h_i) l_i + A(H_m^{\tau} - H_m^k) \right] + \beta (q_m - Q_m)(H_m^{\tau} - H_m^k) \right\}, \quad (7)$$

где J_m - множество номеров вершин куста;

$$\text{если } m \in M_1, \text{то } \tau = \begin{cases} k, & \text{если } k > 0 \\ \overline{1,p}, & \text{если } k = 0 \end{cases}$$

$$\text{если } m \in M_2, \text{ то } \tau = k, k+1, \ldots, p;$$

$$\text{если } m \in M_3, \text{то } \tau = \begin{cases} \overline{1,p}, & \text{если } k = 0 \\ \overline{k,p}, & \text{если } k > 0 \end{cases}$$

p- количество точек на вертикали; $k = 1,2,\ldots,p$.

Стоимость насосных станций по желанию можно выбирать либо по аналогии с ранее запроектированными насосными станциями, либо по функции стоимости от расхода и подаваемого напора .

В третьей главе рассматривается оптимальное пректирование кольцевых трубопроводных сетей .

Задача ставится следующим образом :

$$З = \alpha \sum_{i=1} C(x_i, h_i)\, l_i + \beta H^* Q^* \longrightarrow \min \quad (8)$$

при ограничениях :

$$\begin{cases} \sum_i x_i = Q_j, \\ \sum_r h_i = 0, \\ H^* \geq \max_j \left\{ H_j^{CB} - (Z_i - Z_j) + \sum_{i \in T_j} h_i \right\}, \\ V_{min} \leq V_i \leq V_{max}, \end{cases} \quad (9)$$

где

первые два уравнения выражают I и II -ой законы Кирхгофа;

Q^* - общий расход сети (л/с);

Q_j - расход воды в j - ом узле сети;

x_i - расход воды (л/с), протекающей через i-ый участок.

С помощью декомпозиции переменных x_i и h_i итерационный процесс решения делится на две задачи :

1) при фиксированных направлениях и величинах потоков x_i проводится минимизация функционала (8);
2) при фиксированных гидравлических сопротивлениях на участках (диаметрах) проводится "увязка" сети, т.е. удовлетворение потоков и потерь напоров I-му и II-му законам Кирхгофа методом контурных расходов .

В соответствии расчетными значениями узловых расходов генерируется некоторое начальное распределение потоков по участкам сети и ориентируется начальное направление потоков таким образом, чтобы выполнялся I-й закон Кирхгофа. Затем, по значениям этих потоков, назначаются самые большие диаметры труб из заданного сортамента, удовлетворяющие условию

$$V_i \geq V_{min}.$$

При фиксированных значениях диаметров с помощью метода контурных расходов проводится "увязка " сети .

Далее, проводится "размыкание" сети, т.е. представление в виде дерева. Это позволяет применить к полученной сети

метод решения задачи ТЭ и ГР ЗОС с условием согласования напоров на "расщепленных" узлах .

Основные результаты, приведенные в диссертации, опубликованы в работах / 67 /-/ 69 / и включены в состав Технического и Рабочего проектов по разработке САПР - СКГВХ I и II - ой очередей / 70 / -/ 79 / .

Разработанное на основе методов, рассматриваемых в диссертации, программное обеспечение задач показало высокую эффективность в результате промышленной эксплуатации в зоне деятельности проектных институтов СКГВХ и ЮГВХ (просчитаны сотни объектов). Экономический эффект от внедрения этого программного обеспечения составил около 1.5 миллиона рублей (в Приложение 1 приводится акт об экономической эффективности разработанных подсистем САПР) .

1. ОПИСАНИЕ ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ СИСТЕМ ВОДОСНАБЖЕНИЯ

1.1. Вводная часть и обзор литературы

Проблема оптимального проектирования сложных распределенных систем трубопроводного транспорта является одной из самых актуальных в народном хозяйстве. На сооружение, развитие и реконструкцию этих систем ежегодно расходуется значительные денежные средства, они являются крупнейшими потребителями металла и энергосилового оборудования. Отсюда первостепенное значение количественного обоснования принимаемых решений, связанных с их оптимальным проектированием , что требует соответствующего математического, программного, информационного и организационного обеспечения.

В связи с этим принципиальным моментом является общность всего комплекса оптимизационных технико-экономических задач, составляющих проблему оптимального проектирования систем водоснабжения.

Основные задачи проектной практики в данной области заключаются в выборе: 1) мест размещения и параметров источников; 2) конфигурации сети, которая связывает множество потребителей с источниками; 3) физико-технических парамет-

ров всех элементов системы; 4) мероприятий по обеспечению и повышению структурной и технологической надежности системы .

Первые два типа задачи решаются отделами перспективного планирования (в основном на основе инженерного опыта и интуиции), которые обосновывают: необходимость создания конкретной системы, размещение источников, предельные радиусы снабжения от каждого из них и в итоге - структуру системы в целом, включая конфигурацию сети.

Следующий тип задач, который рассматривается для уже определенной структуры системы, заключается в выборе диаметров трубопроводов, сетевых сооружений (насосных станций) и расчетного потокораспределения для того или иного уровня нагрузок у потребителей. Выполняется это уже в других подразделениях проектных организаций.

Как уже было отмечено, проблемы, затронутые в работе, решались в рамках разработки САПР-СКГВХ. Эти разработки включены в состав основных проектирующих подсистем: Закрытых оросительных сетей(ЗОС), Магистральных водопроводов(МВ) и Поселковых водопроводных сетей (ПВС).

Под системой водоснабжения понимается комплекс гидротехнических сооружений и средств управления с помощью которой обеспечивается очистка, хранение, подача и распределение воды в нужные сроки и в нужном количестве на определенной территории.

В связи с тем, что применение математического моделиро-

вания при проектировании элементов подачи и распределения (в дальнейшем - система подачи и распределения) систем водоснабжения дает наибольший эффект, в работе рассматриваются именно задачи, связанные с определением оптимальных (или близких к оптимальным) параметров этих систем.

В зависимости от требований, предъявляемых к таким системам в смысле надежности (бесперебойного обеспечения водой), различают три типа водопроводных сетей: разветвленные, кольцевые и смешанные .

Известно, что разветвленные водопроводные сети (РВС) - это сети в которых вода поступает в заданную точку только с одной стороны, а в кольцевых сетях (КС) вода может поступать в заданную точку более чем с одной стороны.

Как правило, водопроводные сети проектируются в виде РВС для оросительных сетей и групповых магистральных водопроводов, а сети поселкового водоснабжения проектируются в виде КС, либо смешанные.

Систему подачи и распределения можно разбить на три составляющие :

1) источники давления и расхода (например, насосные или компрессорные станции, аккумулирующие емкости и др.), обеспечивающие притоки транспортируемой среды и дающие энергию в систему;

2) трубопроводные сети (в виде совокупности взаимосвязанных трубопроводов), соединяющие источники со множеством потребителей;

3) потребители.

Такое деление довольно условно и зависит от целей изучения реальной системы и характера решаемых задач, степени детализации, а также и от режимов ее работы. Например, в качестве потребителей могут рассматриваться как отдельные установки (дождевальные машины в оросительных сетях и др.), так и здания или даже поселки в целом. Сеть в одних случаях отображает и конкретизирует связи между источниками и потребителями, доводя эти связи до разводящих линий и фактических потребителей(это - оросительные и внутрипоселковые сети), а в других случаях включает лишь основные магистрали между источниками и укрупненными потребителями(магистральные водопроводы). Точно также и источники могут задаваться вместе со своей "начинкой" (оборудованием) или только выходными параметрами.

При математическом моделировании все эти элементы находят соответствующее отражение в расчетной схеме сети: участки сети, включающие арматуру и другие местные сопротивления; места расположения источников расхода и потребителей, а также соединение участков в виде узлов(вершин). Источники напора (а иногда и расхода) могут относиться как к узлам так и к участкам.

В качестве параметров участков и узлов будем различать: технические характеристики (диаметры трубопроводов, длины и гидравлические сопротивления); гидравлические параметры (расходы жидкости на участках или в узлах, скорость течения

жидкости, давление(напор) в узлах), описывающие состояние системы в любом из режимов работы; граничные условия - входные данные (величины притоков и нагрузок, допустимые диапазоны значений гидравлических параметров).

Выше сказанное относится к задачам технико-экономического и гидравлического расчета сетей :

- технико-экономический и гидравлический расчет закрытых оросительных сетей (ТЭ и ГР ЗОС);
- технико-экономический и гидравлический расчет магистральных водопроводов (ТЭ и ГР МВ);
- технико-экономический и гидравлический расчет кольцевых сетей (ТЭ и ГР КС).

В начальной стадии развития методов технико-экономического расчета водопроводных сетей решающее значение имело определение диаметров труб трубопроводов при интуитивном выборе наилучшего потокораспределения. Достаточно много работ посвящено именно этой проблеме .

В создании и усовершенствовании этих методов большой вклад внесли: Лобачев В.Г. /42/, Абрамов Н.Н. /1,2,3,4/, Мошнин Л.Ф. /49,50/, Андрияшев /11/,Кирсанов М.В. /33/, Хасилев В.Я. /60,62/, Койда Н.У. /34/, Сироткин В.П. /56/, Кикачейшвили Г.Е. /30,31/ и другие.

Экономический расчет водопроводных сетей был предложен В.Г. Лобачевым /41/,который при определении диаметров ввел в расчеты так называемый "ситуационный фактор", учитывающий взаимодействие участков сети.

В работе /32/ Кирсанов М.В. ввел гидравлические уклоны взамен диаметров и получил уравнение экономичности на основе применения метода Лагранжа и уравнения гидравлического равновесия, что дает в общем случае возможность постепенного подбора значений неизвестных гидравлических уклонов.

Аналитические расчеты систем водопроводов даны в работах Кузьмина В.Ю. /36/, Ильина В.Г. /26/, Леонтьева В.А. и других, при которых существенную роль играет так называемый коэффициент неравномерности расходования электроэнергии при подаче воды, зависящий от неравномерности потребления в течение расчетного периода. Этот коэффициент достаточно хорошо исследован в работах Абрамова Н. Н. и Поспеловой /3/.

Существует значительное количество методов, позволяющих подобрать экономически наивыгоднейшие диаметры, используя ЭВМ, которые позволяют значительно расширить постановку задачи и не прибегать к упрощениям различного рода.

Для достижения этих целей как у нас, так и за рубежом разработано несколько методов решения задачи по определению оптимальных диаметров разветвленных и кольцевых гидравлических цепей.

Наиболее общий и практически применяемый метод точного технико-экономического расчета водопроводных сетей впервые приводится в работах Мошнина Л.Ф. /50/. По этому методу определение диаметров производится отысканием минимума стоимости транзитного водовода с введением поправочных коэффи-

циентов, учитывающих взаимодействие линий. Эти коэффициенты, называемые "фиктивными расходами" участков сети, рассматриваются как поправочные множители, показывающие величину долевого участия данной линии в создании общего напора. В данной работе приводятся такие принципы и описание электрического моделирования для проведения технико-экономических расчетов систем водоснабжения.

Для выбора экономичных диаметров трубопроводов разветвленных сетей (применительно к тепловым сетям) Хасилевым В.Я. /60/ был использован метод динамического программирования, Меренковым А.П. /46/ были рассмотрены теоретические и вычислительные аспекты алгебраического и экстремального подхода к описанию и методам расчета потокораспределения в гидравлических сетях с сосредоточенными параметрами. Было рассмотрено объединение в одной общей задаче вопросов оптимального выбора диаметров и напоров в трубопроводах с наивыгоднейшим кольцеванием по условиям надежности.

В работах Ласиса А.Я. /37/ разработан алгоритм технико-экономического расчета водопроводных сетей с помощью ЭВМ, который в основном воспроизводит метод "фиктивных расходов", предложенный Мошниным Л.Ф. /49/ и основан на системе линейных уравнений, которые приближенно выражают и взаимодействие всех участков сети. Автором доказано, что данная система линейных уравнений имеет единственное решение.

В работе Койда Н.У. /34/ приводится известный в математическом программировании прием, заключающийся в минимиза-

ции некоторой вспомогательной функции, так называемый "метод штрафных функций". Задача заключается в минимизации функции

$$\Phi = \Pi + \lambda f , \qquad (1.1)$$

где Π - приведенная стоимость; λ-положительная величина;

$f(S,H)$ - рассматривается как функция сопротивления участков и напоров в точке питания, отображает условие гидравлической увязки сети. Минимизация функции проводится на дискретном множестве сопротивлений участков и напоров в точке питания, что позволяет после получения расчетных данных не прибегать к их корректировке.

Кикачейшвили Г.Е. в работе /31/ предложил метод, который сводит задачу нахождения наивыгоднейших диаметров труб водопроводных сетей к задаче линейного программирования. Для этого он предложил в качестве неизвестных ввести длины $x_{i,j}$, на которых в пределах каждого участка целесообразно уложить трубы того или иного стандартного диаметра d_i. Очевидно, что для каждого участка i

$$\sum_{j=1}^{m} x_{i,j} = l_i , \qquad (1.2)$$

где m - число стандартных диаметров.

Если $C_{i,j}$ - стоимость прокладки единицы длины j-го диаметра на i-ом участке, то общую строительную стоимость можно выразить так:

$$F = \sum_{i=1}^{n} \sum_{j=1}^{m} C_{i,j} \; x_{i,j} , \qquad (1.3)$$

Очевидно, что значение F будет тем меньше, чем меньше диаметры труб. Однако уменьшение диаметров труб влечет за собой увеличение потерь напора на участках сети, что соответственно приводит к увеличению подаваемого напора (H).

Если при заданном расходе q_i данному диаметру d_i соответствует гидравлический уклон $I_{i,j}$, то суммарные потери напора h_i на участке i окажутся равными:

$$h_i = \sum_{j=1}^{m} I_{i,j} x_{i,j} , \qquad (1.4)$$

Тогда, если H_k – требуемый напор в какой-либо точке, то

$$\sum_i h_i \leq H - H_k , \qquad (1.5)$$

где суммирование проводится по всем участкам сети от начала до рассматриваемой точки.

Приведенная стоимость сети записывается в виде :

$$S = \sum_{i=1}^{n} \sum_{j=1}^{m} C_{i,j} x_{i,j} + w H , \qquad (1.6)$$

где w - коэффициент удельной приведенной стоимости, зависящий от суммарного расхода Q, подаваемого напора H, срока окупаемости капитальных вложений, стоимости электроэнергии и других факторов.

Таким образом, задача сводится к отысканию минимума функционала (1.6) при выполнение ограничений (1.2) и (1.5).

Графические и аналитические методы занимают большое место в литературе, посвященной методам расчета систем водо-

снабжения. Этим методам посвящены работы Ф.Е. Максименко /44/ .

Обобщение и развитие графических методов, применительно к системам этого рода, дано в книге Н.Н. Абрамова /1/.

Для расчета напорных конденсатопроводов графический метод использовался в работе Б.Л. Шифринсона /63 / . В работе Л.А. Мелентьева /45 / этот метод применялся для иллюстрации переменных режимов в тепловых сетях. Имеются работы, в которых графо-аналитическая методика используется для расчета систем воздукообмена в зданиях, газовых сетей.

Ограниченные возможности графических методов для получения численных результатов, относящихся к сложным системам, очевидны. Однако они приносят большую пользу благодаря своей иллюстративности, способности показать на простых примерах качественный характер изменения режимов в гидравлических системах.

Среди гидравлических методов следует различать две группы построений. Первая имеет своей непосредственной целью выполнение расчетов потокораспределения, и ее отличительной особенностью является построение сопрягающихся кривых в системе координат "напоры-расходы". Принципиальная выполнимость таких построений определяется возможностью оценивать сопротивления отдельных ветвей и общее сопротивление гидравлической системы, отнесенное к тому или иному источнику действующего напора.

Вторая группа используется как в области водоснабжения,

так и (особенно широко) теплоснабжения. Это - построение так называемых "пьезометрических графиков", которые выполняются в системе координат "напоры-длины участков сети". Методическая и иллюстративная роль таких графиков несомненна. Во многих статьях, монографиях и учебниках "пьезометры" мастерски применяются для характеристики установившихся и динамических режимов сетей. Можно сказать, что несколько поколений специалистов воспитывалось, усваивая методику построения и анализ этих графиков. Однако по мере усложнения схем и режимов систем их полезность явно снизилась.

Для многих гидравлических систем особенно важны были наиболее простые схемы с последовательно-параллельным соединением ветвей при одном источнике питания. Поэтому в большинстве ранних работ предлагались и использовались методы расчета именно таких схем. К ним относятся: метод "эквивалентных отверстий", предложенный Д. Мюргом /65 /; метод "перемещения единиц объема" Каменева П.Н. /27/; метод суммирования проводимостей и сопротивлений , использованный Б.Л. Шифринсоном /63 / и другими авторами для расчета переменных режимов в тепловых сетях, а Н.Н. Гениевым /18/ - для простых водопроводов.

В алгебраическом отношении указанные методы эквивалентны и основаны на специальном свойстве матрицы соединений такого рода схем, которое позволяет в результате элементарных преобразований свертывать ее к единичной строке (или

контуру). Это формальное свойство матриц специального вида, справедливо для электрических и гидравлических цепей, определяет линейную природу методов расчета и оптимизации любых систем, имеющих подобные схемы соединений.

Метод, основанный на суммировании сопротивлений и проводимостей, удобен не только для ручных, но и для машинных расчетов потокораспределения, связанного с отключением и включением отдельных ветвей разветвленной (разомкнутой) сети. Например, он также использовался Меренковым А.П. и Хасилевым В.Я. в работе /47 / для анализа гидравлической устойчивости тепловых сетей.

Другие аналитические методы связаны со стремлением получить некоторую формулу, которая выражала бы зависимость между техническими параметрами системы и характеристиками ее режима, гидравлического и теплового; они также весьма широко использовались, особенно в области теплофикации и централизованного теплоснабжения. В этой связи следует указать на работы Н.М. Зингера /21 , 23/ и других. Располагая подобной формулой и варьируя значениями входящих в нее величин, можно исследовать характер изменения режима и получать необходимые выводы.

Такого рода формулы, получаемые в результате зачастую весьма громоздких, но простых по существу алгебраических преобразований, выражали решение задачи в радикалах и поэтому требовали, так или иначе, сведения к единичному контуру.

Таким образом, графические и аналитические методы сохраняют свое значение как вспомогательные и иллюстративные методы при расчете гидравлических систем простой структуры. Однако их ограниченность очевидна, и единственным методом ее преодоления является обращение к более общим и современным методам математического моделирования с применением ЭВМ.

В научно-технической и учебной литературе хорошо известны и до настоящего времени широко применяются на практике (в том числе и в качестве расчетных методов для ЭВМ) относительно простые методы последовательных приближений для расчета потокораспределения в кольцевых многоконтурных трубопроводных сетях. Эти методы органически учитывают сетевой характер задачи гидравлического расчета и сводятся к последовательному уравновешиванию (увязке) перепадов давлений на ветвях контуров или расходов в узлах сети, исходя из законов Кирхгофа.

Так называемый метод поконтурной увязки перепадов давлений (потерь напора) описывается в работах В.Г. Лобачева /42 ,43 /, Х. Кросса /64 /, М.М. Андрияшева /10,11/. Много статей и монографий посвящены этиму методу и его модификациям / 2,7,12,14,15,16,18, 38 ,53 ,59 ,61 и др./.

Исходной публикацией по методу поконтурной увязки перепадов давлений (потерь) считается вышедшая в 1932 году монография М.М. Андрияшева /10/, в которой впервые зафиксированы основные положения данного метода и предложена формула для увязочного контурного расхода.

В работе Лобачева В.Г. /42 / предлагается метод, который основан на применении математического анализа и позволяет решение квадратичных уравнений свести к решению линейных уравнений, и впервые предложен системный подход при рассмотрении совокупности колец .

Основное содержание его статьи сводится к построению и решению на примере "двухколечной сети" системы из двух линейных уравнений относительно "добавочных перекидных расходов" для этих колец, а также к обобщающему выводу: "Продолжая применение указанного метода к расчету многоколечных систем, мы будем каждый раз получать столько добавочных перекидных расходов, сколько колец".

В статье Х. Кросса /64 / "Анализ течений в сетях из трубопроводов или проводников" впервые в общем виде сформулированы основные положения для описания потокораспределения и его расчета на базе идей поконтурной и поузловой увязки потерь давления и расходов. Основные положения следующие :

а) общий расход притекающий к любому узлу, равен общему расходу покидающему его (непрерывность течения);

б) общие изменения потенциала вдоль любого замкнутого пути равно нулю (непрерывность потенциала).

Далее автором предлагаются два метода для расчета потокораспределения. Один исходит из того, что "потоки в трубах и проводниках сети всегда удовлетворяют условию равенства нулю их общего баланса в каждом узле и они последовательно

корректируются с тем, чтобы удовлетворить условию равенства нулю общего изменения давления вдоль каждого контура". Другой основан на том, что "общее изменение давления вдоль каждого контура всегда равно нулю, а потоки в трубах контура последовательно подбираются так, чтобы общий их баланс в каждом узле в конце концов, - приблизительно или точно, стал равным нулю". Первый метод назван им методом "балансирования (уравновешивания) давлений", а второй - методом "балансирования потоков (расходов)".

Для поконтурной увязки потерь давлений Х. Кросс выводит формулу

$$\Delta = -\frac{\sum h}{\sum r} = -\frac{\sum r\, Q_o^{n} \text{ (с учетом напр. потока)}}{\sum r\, Q_o^{n-1} \text{ (без учета напр. потока)}}, \quad (1.7)$$

где

Δ - увязочный расход, r - потеря давления в трубе для потока единичной величины,

Q_o - предыдущее приближение в значении расхода на каждом трубопроводе в данном контуре. Она получена из условия

$$\sum r\, Q^n = \sum r\, (Q_o + \Delta)^n = 0,$$

которое после отбрасывания нелинейных относительно Δ членов переходит в приближенное равенство

$$\sum r\, Q^n = -\Delta \sum n\, r\, Q_o^{n-1}, \quad (1.8)$$

откуда и следует (1.7).

Что касается второго метода, то он характеризуется им

следующим образом: "Мы можем задаться значениями давлений в системе и вычислить расход в каждой трубе, соответствующий разностям давления. Отсюда находится общий расход в каждом узле, включая входы и выходы. Распределим этот расход по трубам, сходящимся в данном узле, обратно пропорционально их сопротивлениям

$$R = n\,r\,Q^{n-1}$$

Это, конечно, вызовет избыток (или недостаток) расхода в последующих узлах, но путем последовательного распределения избыточные расходы в конце концов должны вытесняться через входы и выходы системы".

В статье оба метода иллюстрируются многочисленными примерами расчета, но не дается анализа их сходимости. В конце автор рассматривает общий случай зависимости между потерей давления и расходом в виде $h = f(Q)$ и дает соответствующую формулу для увязочного контурного расхода:

$$\Delta Q = \frac{\sum f(Q)}{\sum f(Q)} \qquad (1.9)$$

Наиболее полный обзор литературы и существующих разработок по данной теме приводится в книге А.П. Меренкова и В.Я. Хасилева /- 48 /.

1.2. Описание задачи ТЭ и ГР РВС

Как уже было отмечено, оросительные сети и групповые магистральные водопроводные сети проектируются в виде РВС.

Предполагается, что задано плановое положение сети (дерево сети- Рис 1.1) вершины, которого перенумерованы по порядку числами 1,2,...n, начиная с напорного сооружения (насосной станции), известны длины участков, высотные отметки вершин и величины потоков (расходы) на каждом участке сети, кроме того, задается набор стандартных диаметров труб и их стоимостей, которыми предполагается проектирование сети.

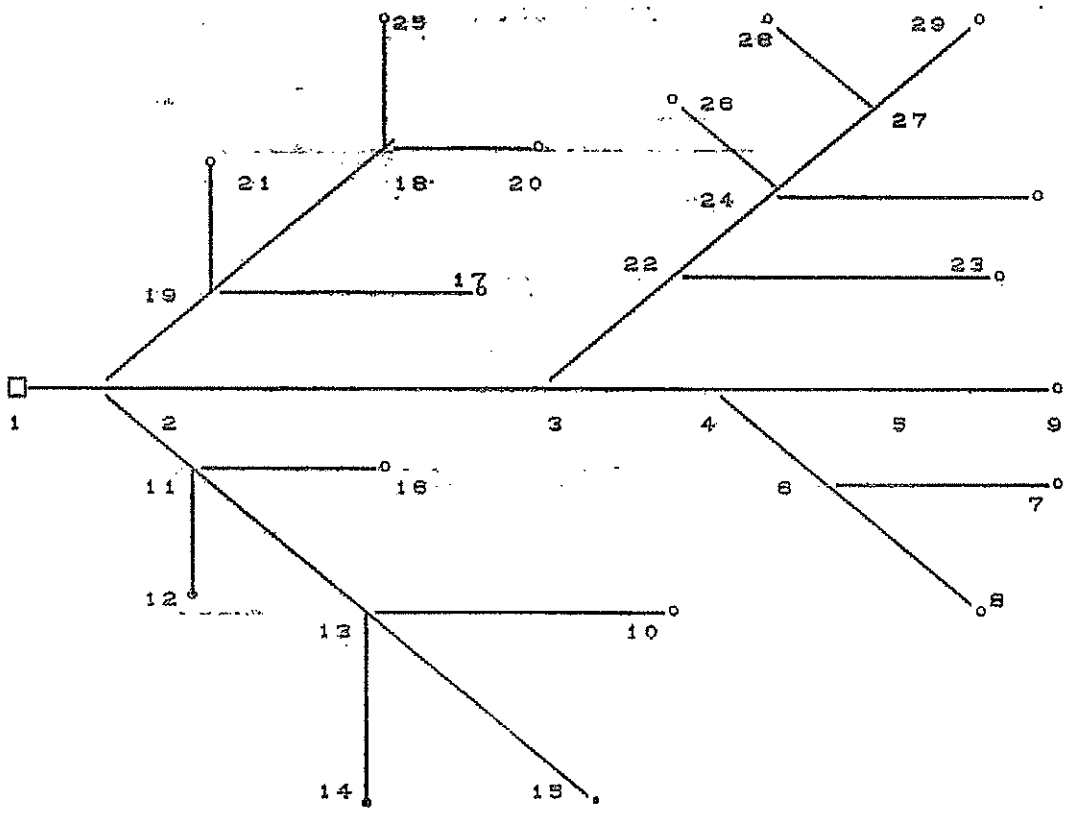
Задача ТЭ и ГР ЗОС заключается в следующем: при заданных технических требованиях и ограничениях определить диаметры труб участков и напор на насосной станции таким образом, чтобы приведенная стоимость сети была минимальной.

Приведенные затраты сети включают в себя капитальные (стоимость труб сети и насосной станции) и эксплуатационные затраты, основную часть,которых составляют затраты на электроэнергию для перекачки воды от насосной станции к потребителям

$$З = \alpha (C_T + C_{нс}) + Э, \qquad (1.10)$$

где α- нормативный коэффициент, учитывающий амортизационные отчисления в зависимости от срока окупаемости сети.

Результатом решения задачи ТЭ и ГР ЗОС является инфор-



Дерево сети.
Рис 1.1

мация, включающая в себя диаметры, толщины стенок, материал и стоимость труб, потери напора и скорость течения воды по участкам сети; величина напора в узлах сети, потребный напор на насосной станции; приведенная стоимость сети, капитальные и энергетические затраты .

В виду большой протяженности магистральных водопроводов на сети задаются дополнительно еще возможные места расположения насосных станций перекачки или подкачки (схема приведена на Рис 1.2).

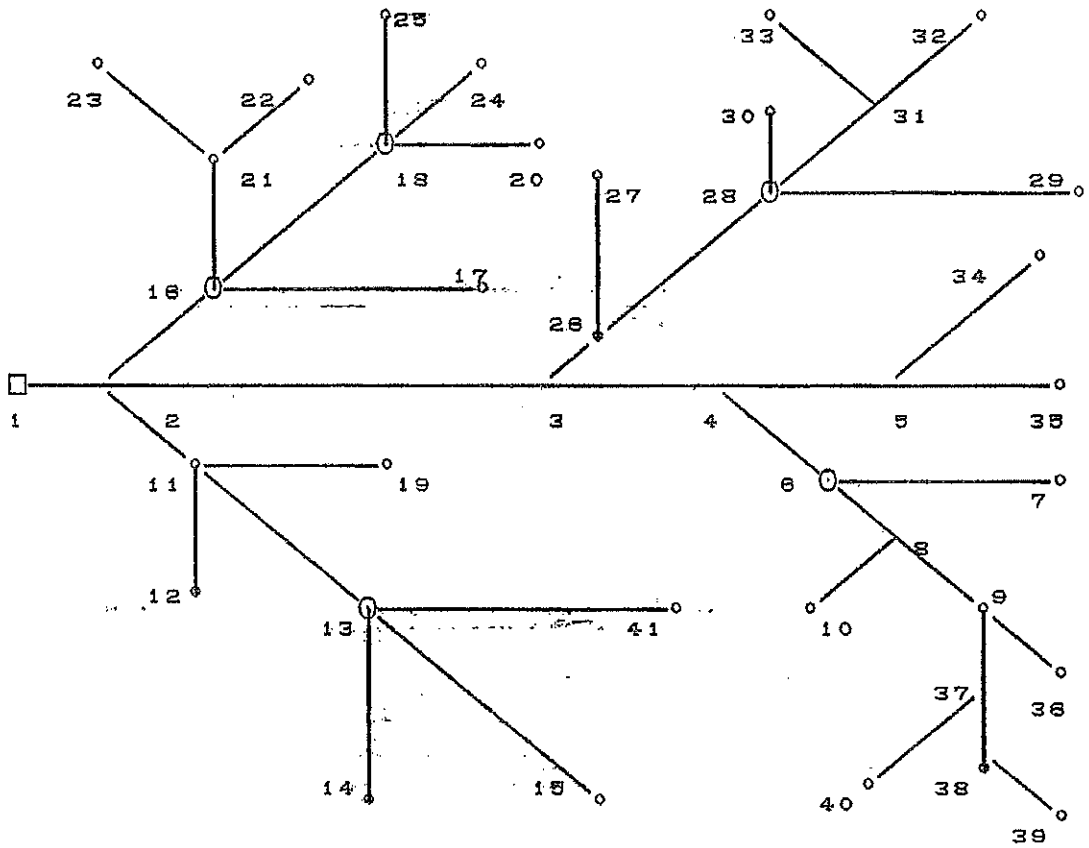


Схема магистрального трубопровода.

О - возможные места расположения насосных станций.

Рис 1.2

На содержательном уровне задача ТЭ и ГР МВ формулируется в следующем виде: при заданных технических требованиях и ограничениях определить диаметры труб участков сети, места расположения насосных станций перекачки или подкачки и напоры на них таким образом, чтобы приведенная стоимость сети была минимальной.

Приведенная стоимость (1.12) включает в себя капитальные затраты (стоимость труб и типовых насосных станций)

$$Э = \alpha (C_{т} + \sum C_{нс}) + \sum Э \qquad (1.12)$$

и затраты на электроэнергию для подкачки и перекачки воды к потребителям.

Результатом решения задачи ТЭ и ГР МВ является информация, включающая в себя выбранные места расположения насосных станций подкачки и перекачки, напоры на этих станциях; диаметры, толщины стенок, материал и стоимость труб, потери напора и скорость течения воды по участкам сети; величина напора в узлах сети, разность высотных отметок узлов; приведенная стоимость сети, капитальные и энергетические затраты.

1.3. Описание задачи ТЭ и ГР КС

Так как проектирование КС проводилось в составе подсистемы ПВС САПР, задача ТЭ и ГР КС рассматривается для сетей поселкового водоснабжения.

Известно, что проектирование водоснабжения поселка предусматривает три случая:

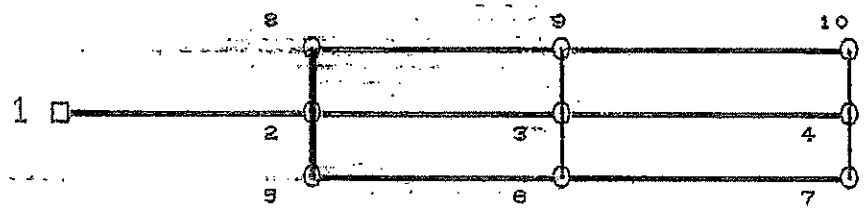
1) вода в поселок подается от группового водопровода с напором, достаточным для водоснабжения всех потребителей;

2) вода в поселок подается от группового водопровода с недостаточным напором;

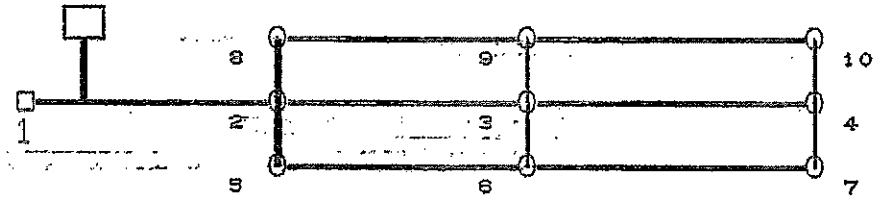
3) вода в поселок подается по локальной схеме(от водозабора).

В первом случае в состав рассчитываемых сооружений входит регулирующая емкость, в виде водонапорной башни, и разводящая сеть.

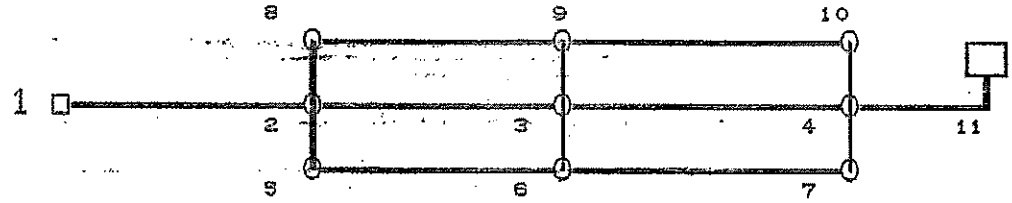
Во втором и в третьем случаях в состав рассчитываемых сооружений входят: резервуары, насосная станция поселка, водонапорная башня(при безбашенной схеме башня отсутствует) и сеть. При проектировании систем водоснабжения поселка могут рассматриваться три различные схемы (Рис 1.3) :



а) схема 1.



б) схема 2.



в) схема 3.

Схемы систем водоснабжения.

Рис 1.3.

1) безбашенная схема;

2) схема с башней (напорным резервуаром) в начале сети;

3) схема с башней в конце сети (с контррезервуаром).

Задача ТЭ и ГР КС формулируется следующим образом : при заданном плановом положении сети и при известной схеме системы водоснабжения (известны места расположения напорно-регулирующих сооружений), а также при заданных длинах участков, высотных отметок вершин и расхода сети, сортаментов труб, задаваемых на проектирование, определить технические и гидравлические параметры сети таким образом, чтобы приведенная стоимость сети была наименьшей.

Результатом решения задачи ТЭ и ГР КС является информация включающая напор на напорном сооружении, а также диаметры, толщины стенок, материал и стоимость труб, потери напора, скорость течения воды, величина и направление потоков по участкам сети ; свободный напор в узлах сети, пьезометры узлов; величины "невязок" по контурам и приведенная стоимость сети.

2. ЗАДАЧА ТЕХНИКО - ЭКОНОМИЧЕСКОГО И ГИДРАВЛИЧЕСКОГО РАСЧЕТА РАЗВЕТВЛЕННЫХ ВОДОПРОВОДНЫХ СЕТЕЙ

2.1. Математическая модель задачи ТЭ и ГР ЗОС

Как уже было отмечено, задача ТЭ и ГР ЗОС заключается в следующем: при заданном плановом положении сети и при заданных технических требованиях и ограничениях определить диаметры труб участков сети и напор на насосной станции таким образом, чтобы приведенная стоимость сети была минимальной.

Пусть задано плановое положение закрытой оросительной сети содержащей n узлов и пусть :

l_i - длина i-го участка сети;

q_i - расход воды на i-ом участке сети;

z_j - высотная отметка j-го узла сети;

$H_j^{св}$ - минимальный (свободный) напор в j-ом узле сети;

H_j - напор в j-ом узле сети;

V_{min} и V_{max} - допустимый диапазон скоростей течения воды на участках сети;

$D = \{d_k\}$ и $C = \{c_k\}$ - упорядоченное множество стандартных диаметров труб и соответствующее множество стоимостей одного погонного метра;

h_i - потери напора на единицу длины i-го участка;

V_i - скорость течения воды на i-ом участке;

H^* - напор на насосной станции,

где $i = 2,\ldots,n$, $j = 1,2,\ldots,n$, $k = 1,2,\ldots,r$,

r - количество труб в заданном множестве труб.

В дальнейшем, участки будем нумеровать номерами их концевых вершин, причем вершину соответствующую насосной станции нумеруем единицей .

Задача решается при ограничениях на скорость течения воды

$$V_{min} \le V_i \le V_{max} \qquad (2.1)$$

и при ограничениях на напоры в узлах : напоры в узлах сети должны быть не менее, чем заданные свободные напоры (II -ой закон Кирхгофа для разветвленных сетей) следовательно:

$$H_j = H^* - \sum_{i \in T_j} h_i l_i + Z_i - Z_j \ge H_j^{CB} , \qquad (2.2)$$

где $j = 1,2,\ldots,n$;

T_j -множество номеров участков, входящих в траекторию от насосной станции до вершины с номером j .

Приведенные затраты представляются в виде :

$$З = \alpha \sum_{i=2}^{n} C(h_i) l_i + \beta Q H^* + A_H(H^*,Q), \qquad (2.3)$$

где

α - нормативный коэффициент учитывающий, аморти-

зационные отчисления, в зависимости от срока окупаемости сети;

β - переводной коэффициент (в рубли);

$C(h_i)$ - функция, выражающая удельную стоимость i-го участка сети при потере напора на нем h_i.

$A_H(H^*, Q)$ - стоимость насосной станции .

Таким образом, математическая модель задачи заключается в следующем:

$$З = \alpha \sum_{i=2}^{n} C(h_i)\, l_i + \beta\, Q\, H^* + A_H(H^*, Q) \longrightarrow \min$$

при ограничениях :

$$\begin{cases} H_j = H^* + \sum\limits_{i \in T_j} h_i l_i + Z_i - Z_j \geq H_j^{CB}, \\ V_{min} \leq V_i \leq V_{max}. \end{cases}$$

2.2. Метод и алгоритм решения задачи ТЭ и ГР ЗОС

В отличие от линейной модели приведенной Кикачейшвили Г.Е. в работе /31/, размерность которой порядка r (n-1) (если учесть что средняя оросительная сеть содержит более 30 вершин, а используемый в проектировании сортамент труб со-

держит порядка 20 различных диаметров труб, то получаем более 600 неизвестных), зависимость $C(h_i)$ представляется в виде кусочно-линейной функции, что позволяет существенно снизить размерность задачи .

Рассмотрим на плоскости (C, h_i) дискретное множество точек, заданных координатами (c_k, h_i^k), причем таких, что $\forall k \in [1, r]$ (k-целое) выполняется неравенство (2.1) (в дальнейшем будем считать, что множество номеров k упорядочено, см. Рис 2.1).

Так как для любого значения h_i всегда будет выполняться неравенства

$$h_i^1 \le h_i \le h_i^k , \qquad (2.4)$$

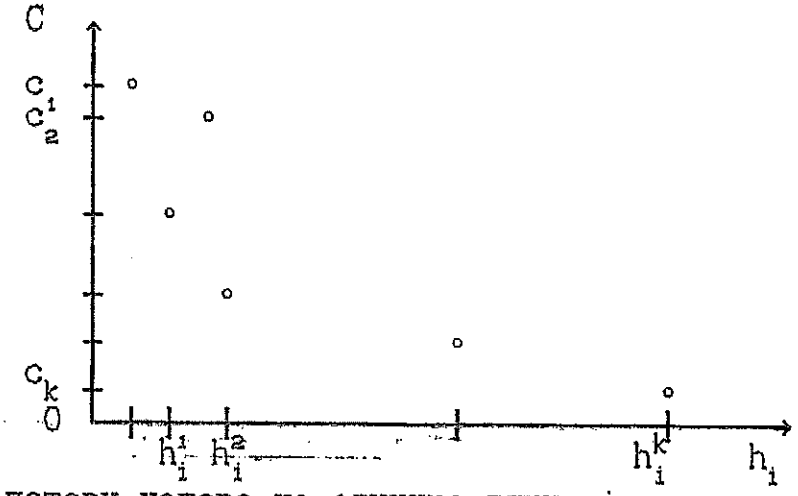
и соответствующее значение функции $C(h_i)$ будет удовлетворять неравенству:

$$c_k \le C(h_i) \le c_1 , \qquad (2.5)$$

тогда любое значение h_i и, соответствующее ему значение функции можно представить в виде линейной комбинации вершин этого многоугольника:

$$h_i = \sum_{m=1}^{k} \lambda_i^m h_i^m \quad \text{и} \quad C(h_i) = \sum_{m=1} \lambda_i^m c_m , \qquad (2.6)$$

причем $\sum_{m=1}^{k} \lambda_i^m = 1$.



h_i^k - потери напора на единице длины i-го участка при расходе воды q_i и при диаметре $d_k \in D$;
c_k - стоимость единицы длины трубы диаметра d_k.

Рис 2.1

Имеет место следующее

У Т В Е Р Ж Д Е Н И Е . При любом значении потери напора h_i удовлетворяющего (2.4) на i-ом участке можно оптимально уложить не более, чем два стандартных диаметра .

Действительно, в виду линейности функционала $C(h_i)$ он достигает своего минимального значения только на границе области, которая является выпуклым многоугольником с вершинами в точках (c_k, h_i^k), либо на его вершинах.

Таким образом, для i-го участка найдутся два таких номера p и m $\in [1, k]$ такие, что

$$h_i^p \le h_i \le h_i^m , \qquad (2.7)$$

и

$$h_i = \lambda_i^* h_i^p + (1 - \lambda_i^*) h_i^m , \qquad (2.8)$$

где $\lambda_i^* \in [0, 1]$.

Тогда, очевидно, имеет место

$$C(h_i) = \lambda_i^* c_p + (1 - \lambda_i^*) c_m \qquad (2.9)$$

Тем самым размерность задачи уменьшается до 2(n-1) неизвестных.

Рассмотрим случай, когда напор на насосной станции задан и ее стоимость постоянная, т. е. энергетические(β Q H^*) и капитальные затраты ($A_H(Q,H^*)$) функционала (2.3) фиксированы. Тогда задача сводится к определению минимума функционала

$$З = \alpha \sum_{i=2}^{n} C(h_i) l_i \qquad (2.10)$$

при ограничениях :

$$\begin{cases} H_j = H^* + \sum_{i \in T_j} h_i l_i + Z_i - Z_j \geq H_j^{CB} , \\ V_{min} \leq V_i \leq V_{max} \end{cases}$$

Для решения задачи разработан метод основанный на идее метода покоординатного спуска к минимуму (за i-ую координату принимается i -ый участок) с помощью, которого строится итерационный процесс

$$h_i^{(N+1)} = h_i^{(N)} + \Delta h_i^{(N+1)}, \qquad (2.11)$$

причем за первоначальное значение $h_i^{(0)}$ принимается значение h_i^1 , и $C(h_i) = C(h_i^1)$, тогда значение минимального напора H^*_{min} удовлетворяющего (2.2) будет

$$H^*_{min} = \max_j \left\{ \sum_{i \in T_j} h_i l_i + Z_j - Z_1 + H_j^{CB} \right\}, \qquad (2.12)$$

и при значении заданного напора $H^* \geq H^*_{min}$ мы можем вычислить значения напоров на j-ом узле по формуле

$$H_j = H^* - \sum_{i \in T_j}^{n} h_i l_i + Z_1 - Z_j . \qquad (2.13)$$

Для определения направления спуска вычисляется значение разности стоимости на каждом участке :

$$E_i = C(h_i) - C_i(h_i^\tau), \qquad (2.14)$$

где τ - целое и $1 \leq \tau \leq k$.

Выбор координаты i , дающей наискорейший спуск , осуществляется по максимальному значению :

$$\max_i \left\{ E_i \right\} , \qquad (2.15)$$

где $i = 2,3,\ldots,n$.

Для определения величины шага введем понятие ресурса участка:

ресурсом i-го участка назовем величину :

$$\Delta H_i = \min_{j \in T_i^*} (H_j - H_j^{CB}), \qquad (2.16)$$

где

T^*_i - множество номеров узлов, траектории, которых включают i-ый участок.

Величина шага спуска $\Delta h_i^{(N+1)}$ (увеличение h_i) определяется из условия:

$$\Delta h_i^{(N+1)} = \begin{cases} \Delta H_i \ , \text{ если } \Delta H_i \le \Delta h_i^T \\ \Delta h_i^T \ , \text{ в противном случае} \end{cases} , \qquad (2.17)$$

где

$\Delta h_i^T = h_i^T - h_i$ - значение, дающее максимум E_i в (2.14).

Процесс спуска продолжается до тех пор, пока имеются участки с положительными ресурсами. Очевидно, что после того, как будут исчерпаны ресурсы на всех участках будем иметь

$$h_i^{k_i+1} \le h_i \le h_i^{k_i} ,$$

тогда оптимальная стоимость труб участков может быть вычислена из системы уравнений относительно λ_i^* и $C(h_i)$:

$$\begin{cases} h_i = \lambda_i^* h_i^{k_i+1} + (1 - \lambda_i^*) h_i^{k_i} \\ C(h_i) = \lambda_i^* C_{k_i+1} + (1 - \lambda_i^*) C_{k_i} \end{cases} , \qquad (2.18)$$

где k_i - целое и $1 \le k_i \le k$.

В том случае, когда напор на насосной станции не задан и стоимость насосной станции постоянная, в качестве значения первого приближения напора принимается H^*_{min}, а оптимальный напор определяется из итерационного процесса

$$H^* = H^* + \Delta H^* , \qquad (2.19)$$

где

ΔH^*-наперед заданный шаг изменения напора.

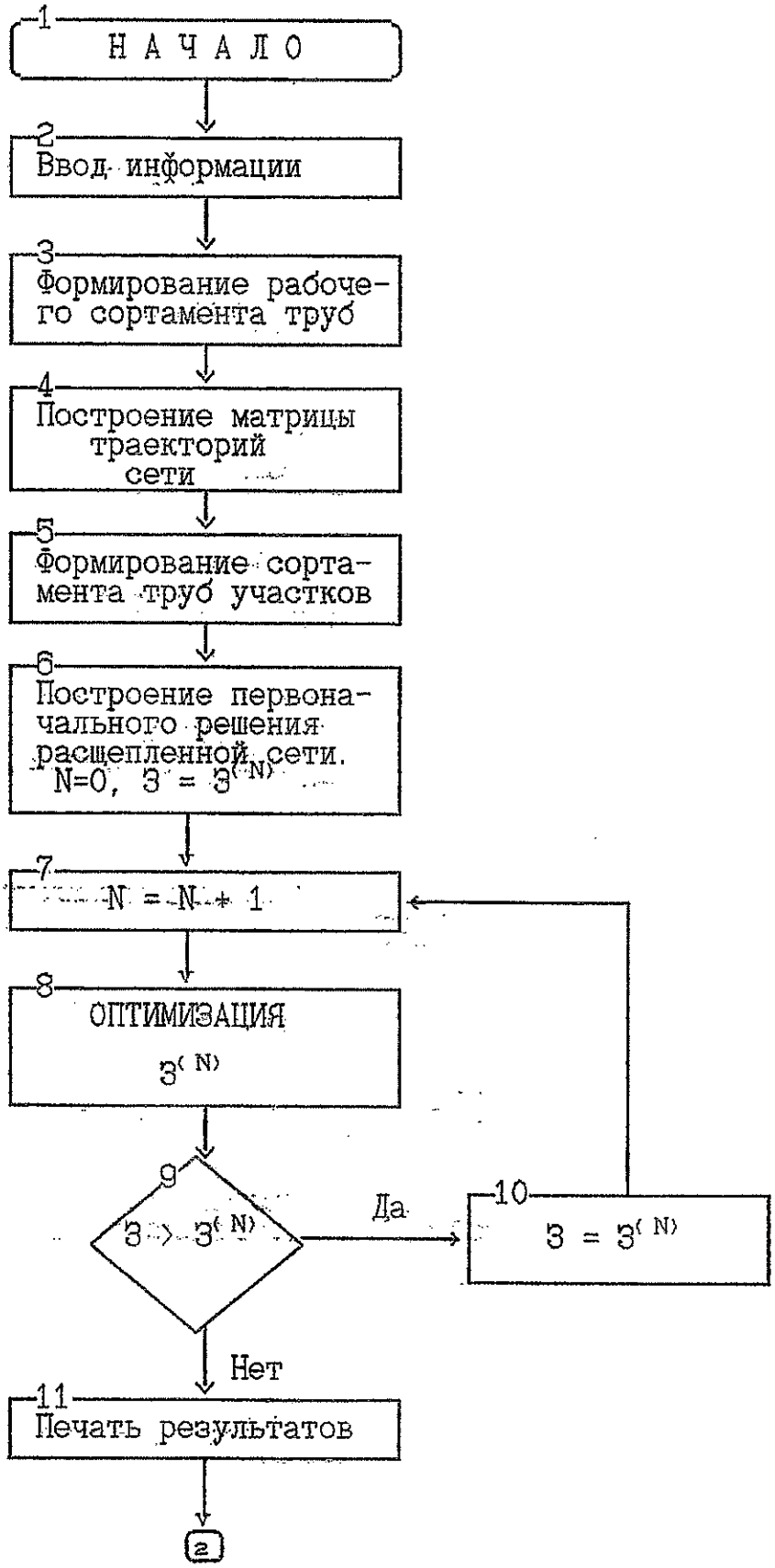
Ввиду выпуклости функционала (2.3), на каком-то приближении напора H^* мы получим $З^{(N^*+1)} > З^{(N^*)}$, тогда значение оптимального напора будет :

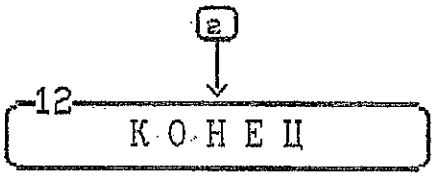
$$H^* = H^*_{min} + N^*\Delta H^* \qquad (2.20)$$

Очевидно, что, чем меньше шаг изменения напора ΔH^*, тем точнее полученное значение оптимального напора.

Блок-схема алгоритма приведена на Рис 2.2.

При решении задачи с выбором насосной станции формируется дискретная функция $A_H(H^*,Q)$ в зависимости от подаваемого напора и расхода из каталога типовых насосных


1
НАЧАЛО
2
Ввод информации
3
Формирование рабочего сортамента труб
4
Построение матрицы траекторий сети
5
Формирование сортамента труб участков
6
Построение первоначального решения расщепленной сети.
N=0, З = З(N)
7
N = N + 1
8
ОПТИМИЗАЦИЯ
З(N)
9
З > З(N)
Да
10
З = З(N)
Нет
11
Печать результатов
2



Блок-схема алгоритма решения задачи.

Рис 2.2

станций, используемых в проектировании закрытых сетей.

Каждая насосная станция имеет свою расходно-напорную характеристику (Q H), которая описывает ее рабочую область и, кроме того, еще имеет значения минимального и максимального расходов (Q_{min}, Q_{max}) и соответственно напоров (H_{max}, H_{min}).

Рабочие характеристики насосных станций с достаточной точностью аппроксимируются методом наименьших квадратов параболой :

$$H = a\,Q^2 + b\,Q\,, \qquad (2.21)$$

где

H и Q - напор и расход насосной станции ;

a и b - коэффициенты аппроксимации.

Задача решается полным перебором всех насосных станций, для которых выполняются условия :

$$\begin{cases} Q_{min} \le Q \le Q_{max} \\ H_{min} \le H^*_{min} \le H_{max} \end{cases}, \qquad (2.22)$$

где Q и H^*_{min} - соответственно расход и минимальный потребный напор сети.

Оптимизация проводится при напорах насосных станций вычисленных по формуле (2.21), а за значения функции S (H,Q)

принимается фактические значения стоимости типовых насосных станций .

2.3. Описание программы задачи ТЭ и ГР ЗОС

Программное обеспечение задачи разработано на двух алгоритмических языках (ПЛ/1-в среде операционной системы ЕС ЭВМ, и Си - для ПЭВМ, совместимых с IBM PC AT/XT) и реализует диалологовый режим. Объем оперативной памяти не более 200 кб и время счета средних сетей не более одной минуты . Данная программа находится в промышленной эксплуатации в составе подсистемы ЗОС САПР-СКГВХ с 1980 года.

Программное обеспечение содержит шесть основных модулей:

1) ввод и печать входной информации;
2) формирование рабочего сортамента диаметров труб из файла базы данных по сортаментам труб;
3) формирование первоначального решения;
4) формирование данных по насосным станциям;
5) минимизация функционала;
6) печать выходных данных.

Информационное обеспечение программы делится на нормативно-справочную (информация по всем сортаментам труб и каталог насосных станций используемых в проектировании) и оперативную информации. Нормативно-справочная включает:

-шифр сортамента, ГОСТ и марка;

-внешний диаметр трубы (м);

-толщина стенки(м) и стоимость единицы длины данного диаметра (руб.).

Структура каталога данных по насосным станциям включает:

-значения минимального и максимального расходов насосной станции ($м^3$ /с);

-значения минимального и максимального напоров (м);

- марка, индекс и габариты здания;
- количество агрегатов ;
- коэффициенты аппроксимации характеристик насосной станции;

-стоимость(тыс.руб).

Нормативно-справочная информация по трубам и насосным станциям приводится в Приложении 8.

Оперативная информация делится на входную и выходную информации.

Входная информация включает в себя:

- режим расчета сети (задан напор на насосной станции либо определение оптимального напора сети либо выбор насосной станции. При любом режиме производится оптимизация диаметров труб сети);
- шифры сортаментов труб, которыми предполагается проектирование сети;
- диапазон допустимых значений скорости течения воды по трубам (м/с);
- количество вершин сети;
- матрица связи вершин указывающая направление те-

чения воды;

- высотные отметки вершин (м);
- значение свободного напора на вершине (м);
- длина участка(м) и расход воды на участке($м^3$/с).

Выходная информация включает в себя :

- участок сети в виде номеров вершин(начало-конец);
- длина участка(м), расход воды на участке ($м^3$ /с);
- значение большего диаметра (м), толщина стенки (м) Гост, марка, длина (м) и потери напора по длине (м);
- значение меньшего диаметра (м), толщина стенки (м), ГОСТ, марка, длина (м) и потери напора по длине (м);
- потери напора по длине участка (м);
- фактический напор в начале участка (м);
- стоимость труб участка (руб);
- напор на насосной станции (м);
- данные выбранной насосной станции;
- приведенная стоимость сети(тыс.руб).

В приложении 2 приводятся распечатки контрольных примеров, а в приложении 3 - распечатки исходных текстов модулей программы.

2.4. Математическая модель задачи ТЭ и ГР МВ

На содержательном уровне задача ТЭ и ГР МВ формулируется в следующем виде: при заданном плановом положении сети и при заданных технических требованиях и ограничениях определить диаметры труб участков сети, места расположения насосных станций перекачки или подкачки и напоры на них таким образом, чтобы приведенная стоимость сети была минимальной.

Таким образом, задача ТЭ и ГР МВ, в сравнении с задачей ТЭ и ГР ЗОС, требует установить места расположения и параметры станций перекачки и подкачки.

Приведенная стоимость включает в себя капитальные затраты (стоимость труб и типовых насосных станций) и затраты на электроэнергию .

Пусть задано плановое положение магистрального трубопровода и пусть :

l_i - длина i-го участка трубопровода;

q_i - расход воды на i-ом участке трубопровода;

Z_j - высотные отметки узлов трубопровода;

$H_j^{св}$ - минимальные (свободные) напоры в узлах трубопровода;

Q_j - расход воды в j-ом узле ;

H_j - напоры в узлах ;

V_{min} и V_{max} - допустимый диапазон скоростей течения воды на участках;

$D = \{d_k\}$ и $C = \{c_k\}$ - упорядоченное множество стандартных диаметров труб и соответствующее

множество стоимостей одного погонного метра;

h_i - потери напора на единице длины i-го участка;

V_i - скорость течения воды на i-ом участке;

$M = \{ M_1, M_2, M_3 \}$ - множество задаваемых номеров вершин, на которых возможно поставить насосные станции подкачки, либо перекачки (M_1, M_2 и M_3 - множества номеров вершин на которых возможно поставить, соответственно: 1) только станции перекачки, 2) только станции подкачки и 3) либо перекачки, либо подкачки);

где

$i = 2,\ldots,n$, $j = 1,2,\ldots,n$ и $k = 1,2,\ldots,r$,

r - количество труб в сортаменте D.

В дальнейшем, примем за номер участка сети номер концевой вершины участка и за 1 - номер корневой вершины магистрального трубопровода .

Задача решается при ограничениях на скорость течения воды (2.1) и напоры, которые в узлах сети должны быть не менее, чем заданные свободные напоры :

$$H_j = H_m^* - \sum_{i \in T_{mj}} h_i l_i + Z_i - Z_j \geq H_j^{CB} , \qquad (2.23)$$

где

$j = 1,2,\ldots,n$, $m \in M$,

H_m^* - напор насосной станции на вершине m ,

T_{mj} -множество номеров участков, входящих в траекторию от насосной станции в точке m до вершины с номером j.

Если учесть, что насосные станции на сети ставятся

после забора воды в узле, тогда приведенные затраты можно представить в виде :

$$З = \alpha \left[\sum_{i=2}^{n} C(h_i)\, l_i + \sum_{m \in M} A(H_m^*) \right] + \beta \sum_{m \in M} (q_m - Q_m)\, H_m^* , \quad (2.24)$$

где

α - нормативный коэффициент, учитывающий амортизационные отчисления, в зависимости от срока окупаемости сети;

β - переводной коэффициент (в рубли);

$C(h_i)$ - выпуклая кусочно-линейная функция, выражающая стоимость единицы длины участка (см. подраздел 2.3.);

$A(H_m^*)$ - стоимость насосной станции подкачки или перекачки и имеет вид :

$$A(H_m^*) = \begin{cases} 0 , & \text{если } H_m^* = 0 \\ A_m^* H_m^* + B_m , & \text{при } H_m^* > 0 \end{cases} , \quad (2.25)$$

где

A_m^* , B_m -коэффициенты линеаризации функции стоимости насосной станции в зависимости от напора.

Таким образом, математическая модель задачи заключается в следующем :

$$З = \alpha \left[\sum_{i=2}^{n} C(h_i)\, l_i + \sum_{m \in M} A(H_m^*) \right] + \beta \sum_{m \in M} (q_m - Q_m)\, H_m^* \longrightarrow \min$$

при ограничениях :

$$\begin{cases} H_j = H_m^* - \sum_{i \in T_{mj}} h_i l_i + Z_m - Z_j \geq H_j^{CB} , \\ V_{min} \leq V_i \leq V_{max} \end{cases}$$

Целевая функция представляет собой выпуклую нелинейную функцию с разрывами первого рода в виду (2.25).

2.5. Метод и алгоритм решения задачи ТЭ и ГР МВ

Сложность решения задачи следует из ее большой размерности.

Действительно, если задачу решить прямым перебором, то потребуется рассмотреть 3^k вариантов расстановки насосных станций, где $k = |M|$ (мощность множества M). Ясно, что на каждом из этих вариантов решается задача оптимизации труб на сети (время счета этой задачи в среднем около минуты на ЭВМ ЕС-1045). Следовательно, при $k = 10$ получим около 1000 часов машинного времени.

Для численной реализации этой задачи разработан метод основанный на идеях динамического программирования.

Решающим здесь являются следующие две идеи :

а) дерево сети расчленяется на части удобные для проведения расчетов и направленного перебора;

б) на каждой такой части дискретно производится перебор напоров на насосной станции.

В связи с этим вводятся определения.

ОПРЕДЕЛЕНИЕ 1. Кустом (в отличие от принятой в теории графов) назовем часть сети, которая получается при движении, начиная с узла $m \in M$, по направлению движения потоков до следующих вершин из M, либо до тупиковых вершин (см. Рис 2.3).

ОПРЕДЕЛЕНИЕ 2. Вертикалью назовем дискретные значения напоров, заданных в промежутке $[0,\hat{H}]$, где $\hat{H}$-максимально допустимое значение напора на насосной станции.

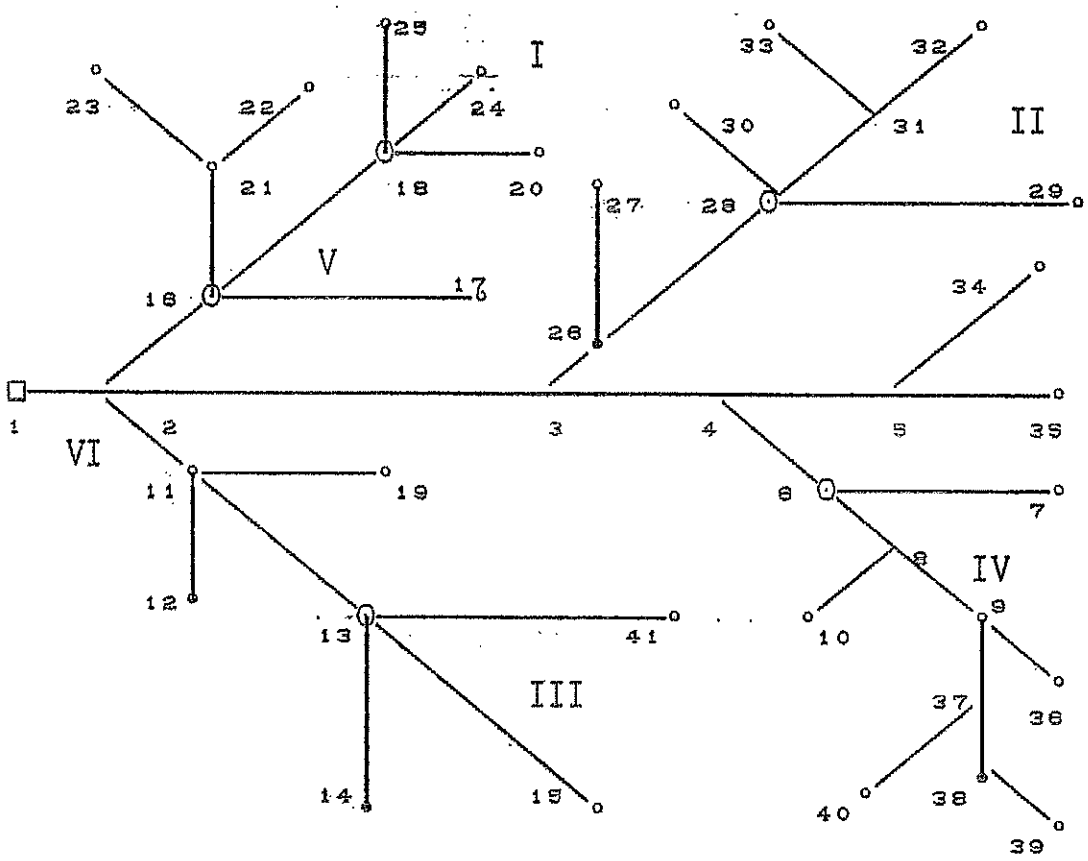


Схема магистрального трубопровода
I-VI- кусты на сети.
Рис 2.3

Напор H_m^* каждого m-го куста($m \in M$) может создаваться нижерасположенной (против течения) насосной станцией, либо насосной станцией в основании куста.

В первом случае $H_m^* = 0$ и стоимость станции в m-ой точке равна нулю. Во втором случае напор предыдущей насосной станции не удовлетворяет (2.23). Мы должны либо увеличить

его, поставив станцию подкачки, либо создать его от нуля, поставив станцию перекачки .

Рассмотрим пространственный граф, вершинами которого будут точки на вертикалях, восстановленных в основании каждого куста. Весами дуг такого графа будут являться оптимальные стоимости куста при заданном значении напора (точка на вертикали) в основании куста и при фиксированных напорах на его вершинах (при наличии вертикалей - это напоры на них).

Очевидно, что ряд дуг пространственного графа отпадут из рассмотрения потому, что напор на вершине куста не будет достаточным (нарушены условия (2.16)), чтобы обеспечить заданные свободные напоры на его конечных вершинах .

Дуге, соединяющей две точки на вертикали, соответствует стоимость станции подкачки, если напор на нижней точке больше нуля и стоимость станции перекачки, если напор на нижней точке равен нулю; в том и в другом случаях добавляется стоимость энергетических затрат и стоимость труб куста. На рисунке 2.4 изображен фрагмент соединения двух точек на вер-

а) станция перекачки; б) станция подкачки.

Рис 2.4

тикали. При этом, каждой точке вертикали соответствует дуга

с минимальным стоимостным весом, в том числе, рассматривается случай, когда данный напор (точка на вертикали) может создаваться насосной станцией, расположенной ниже против течения (вес дуги -стоимость труб куста).

Каждой насосной станции куста сопоставляется ряд стоимостей, соответствующих им напоров и соединений точек на вертикали (для случая б) связь 1 - 4). При этом, если $W_m(H_m^k)$ - минимальная стоимость m-го куста , соответствующая k-ой точке вертикали,при фиксированных значениях напоров на вершинах куста, то

$$W_m(H_m^k) = \min_{\tau} \left\{ \alpha \left[\sum_{i \in J_m} C(h_i) \, l_i + A(H_m^\tau - H_m^k) \right] + \beta(q_m - Q_m)(H_m^\tau - H_m^k) \right\}, \qquad (2.26)$$

где

J_m- множество номеров вершин куста;

если $m \in M_1$,то $\quad \tau = \begin{cases} k, & \text{если } k > 0, \\ \overline{1,p}, & \text{если } k = 0, \end{cases}$

если $m \in M_2$, то $\quad \tau = k, k+1, \ldots, p$;

если $m \in M_3$, то $\quad \tau = \begin{cases} \overline{1,p}, & \text{если } k = 0, \\ \overline{k,p}, & \text{если } k > 0, \end{cases}$

p- количество точек на вертикали;

$k = 1, 2, \ldots, p$.

Оптимизация проводится по стандартной схеме динамического программирования, проводя направленный перебор возможных вариантов соединений точек вертикалей на каждом кусте.

На первом шаге проводится оптимизация кустов вершины, которые являются тупиковыми и запоминаются стоимостные веса всех дуг на вертикалях этих кустов. На следующем шаге, проводится оптимизация тех кустов на тупиковых вершинах, на которых имеются вертикали с уже известными стоимостными весами дуг и эти веса включаются в оптимальную стоимость оптимизируемых кустов.

Конкретно, пусть $S_m(H_m^k)$ – наименьшая стоимость куста m при условии, что на m-ой насосной станции напор равен H_m^k и на тупиковых вершинах(i1,i2,...,il), которого имеются вертикали с известными стоимостными характеристиками. Тогда

$$S_m(H_m^k) = \min \left\{ \sum_{k1=1}^{p} \alpha_{i1}^{k1} S_{i1}(H_{i1}^{k1}) + \sum_{k2=1}^{p} \alpha_{i2}^{k2} S_{i2}(H_{i2}^{k2}) + \ldots + \right.$$

$$\left. + \sum_{kl=1}^{p} \alpha_{il}^{kl} S_{il}(H_{il}^{kl}) + W_m(H_m^k, H_{i1}^{k1}, H_{i2}^{k2}, \ldots, H_{il}^{kl}) \right\}, \quad (2.27)$$

где

$$\sum_{k1=1}^{p} \alpha_{i1}^{k1} = 1, \quad \sum_{k2=1}^{p} \alpha_{i2}^{k2} = 1, \ldots, \sum_{kl=1}^{p} \alpha_{il}^{kl} = 1;$$

$$\alpha_{i1}^{k1}, \alpha_{i2}^{k2}, \ldots, \alpha_{il}^{kl} \in \{0,1\}.$$

Этот процесс продолжается до куста с корневой вершиной 1 включительно т.е. процесс оптимизации дерева сети перемещается от его кроны к корню.

Затем, как и в общей схеме динамического программирова-

ния пройдя обратно, начиная с вертикали на вершине 1, последовательно, до остальных вертикалей определяются наилучшие точки на них, которые дают наличие или отсутствие соответствующих насосных станций на сети. После фиксирования мест расположения насосных станций на сети проводится оптимизация диаметров труб и напоров полученных кустов.

Укрупненная блок-схема алгоритма решения задачи приведена на Рис 2.5.

Для нахождения оптимальных стоимостей кустов при фиксированных значениях напоров на вершинах куста используется метод и алгоритм, описанные в подразделе 2.3.

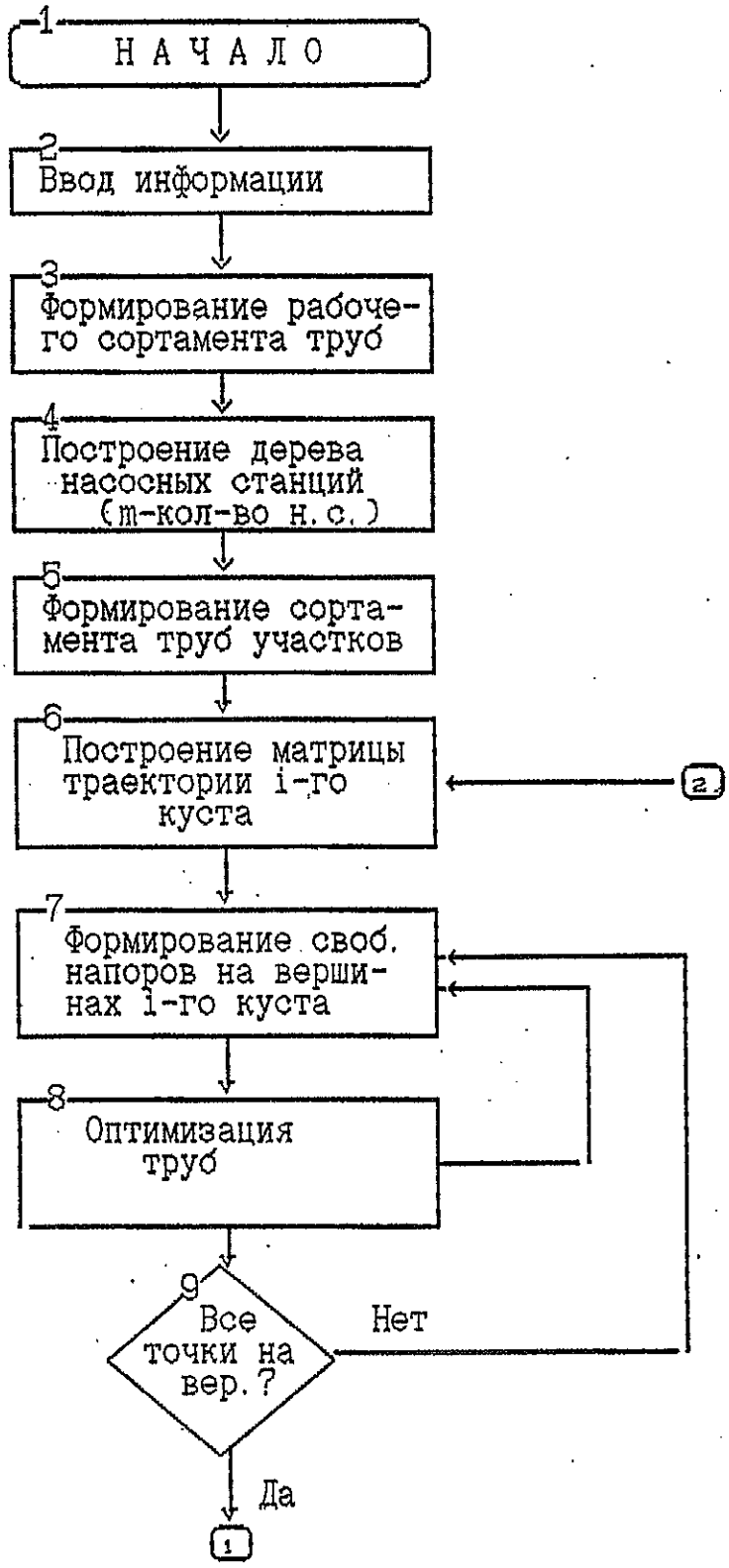
Очевидно, что количество переборов существенно зависит от количества точек на вертикалях (р), от взаиморасположения насосных станций и от количества заданных возможных мест расположений насосных станций. В этом случае предварительно решается задача при достаточно маленьком р, например, при р = 3 или 4, затем полученные места расположения насосных станций задаются в качестве возможных мест расположений и увеличивается количество точек на вертикалях.

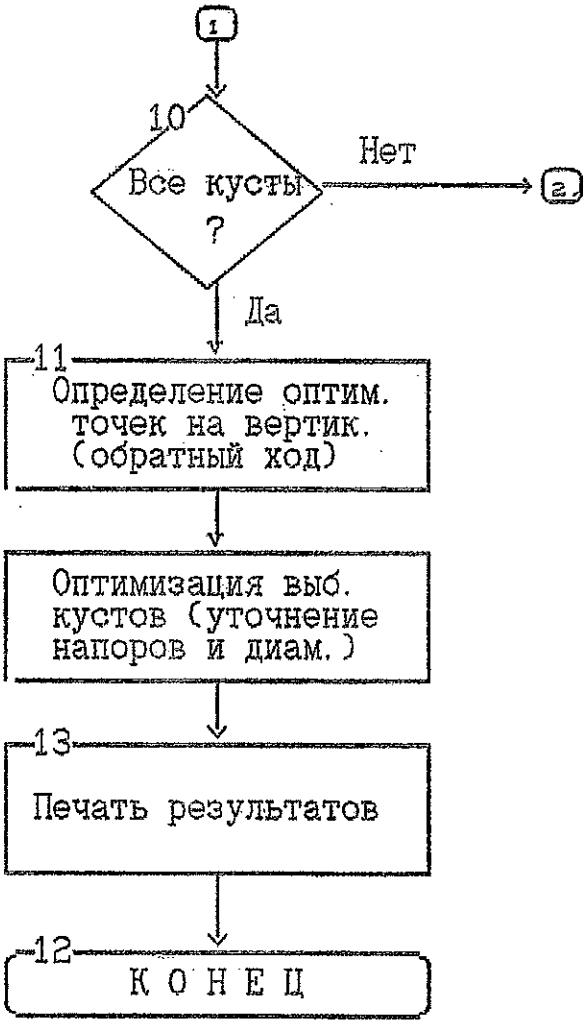
Задача ТЭ и ГР МВ реализована в составе подсистемы проектирования МВ САПР-СКГВХ .

2.6. Описание программы задачи ТЭ и ГР МВ

Программное обеспечение задачи разработано на алгоритмическом языке ПЛ/1 - в среде операционной системы ЕС ЭВМ

объем занимаемой оперативной не более 450 кб и время счета сетей содержащих около 100 вершин не более 40 минут. Прог-





Блок-схема алгоритма решения задачи.

Рис 2.5.

рамма может считать сети содержащие до 300 вершин.

Данная программа находится в промышленной эксплуатации в составе подсистемы МВ САПР-СКГВХ с 1982 года.

Программное обеспечение содержит 13 основных модулей:

1. Ввод и печать входной информации;
2. Формирование рабочего сортамента диаметров труб из файла базы данных по сортаментам труб;
3. Построение дерева куста;

4. Построение дерева насосных станций на сети;
5. Расчет стоимости насосных станций;
6. Определение дуг и их стоимостей;
7. Определение оптимального решения;
8. Формирование ограниченного сортамента труб;
9. Определение гидравлических параметров реконструируемых участков;
10. Формирование первоначального решения куста;
11. Формирование данных по насосным станциям;
12. Минимизация функционала по каждому кусту;
13. Печать выходных данных. ;

Информационное обеспечение программы делится на нормативно-справочную (информация по всем сортаментам труб и каталог насосных станций используемых в проектировании) и оперативную информации .

Структура данных по НСИ та же, что и описанная в подразделе 2.4. Оперативная информация делится на входную и выходную информации.

Входная информация включает в себя:

- количество точек на вертикалях;
- значение максимального значения напора на насосной станции;
- шифры сортаментов труб, которыми предполагается проектирование сети;
- диапазон допустимых значений скорости течения воды по трубам (м/сек);
- количество вершин сети;

- пара чисел, указывающая направление течения воды;
- высотные отметки вершин (м);
- значение свободного напора на вершине (м);
- длина участка(м) и расход воды на участке($м^3/с$);
- признак наличия насосной станции (0 -отсутствует, 1 - насосная станция перекачки, 2 - насосная станция подкачки и 3 -либо перекачка, либо подкачка).

При реконструкции сети задаются внешний диаметр (м) и шифр трубы на тех участках, на которых они известны .

Выходная информация по каждому кусту включает в себя :

- участок сети в виде номеров вершин(начало-конец);
- длина участка(м), расход воды на участке ($м^3/с$);
- разность высотных отметок начала и конца участка;
- значение большего диаметра (м), толщина стенки (м), Гост, марка, длина(м) и потери напора по длине (м);
- значение меньшего диаметра (м), толщина стенки (м), ГОСТ, марка, длина (м) и потери напора по длине (м);
- потери напора по длине участка (м);
- скорость течения воды (м/с);
- фактический напор в начале участка (м);
- стоимость труб участка (руб);
- номер вершины на которой выбрана насосная станция и ее тип(подкачки или перекачки);
- напор на насосной станции (м);

- приведенная стоимость куста;
- Суммарная приведенная стоимость сети(тыс.руб).

В приложении 4 приводятся распечатки контрольных примеров, а в приложении 5 -распечатки исходных текстов модулей программы.

3. ЗАДАЧА ТЕХНИКО - ЭКОНОМИЧЕСКОГО И ГИДРАВЛИЧЕСКОГО РАСЧЕТА КОЛЬЦЕВЫХ СЕТЕЙ

3.1. Математическая модель задачи ТЭ и ГР КС

Пусть задана водопроводная сеть, содержащая n участков и m узлов .

Введем следующие определения :

ОПРЕДЕЛЕНИЕ 1. Элементарным кольцом на сети назовем такую замкнутую последовательность участков на сети, которая ограничивает часть плоскости не содержащей в себе других ребер(участков).

ОПРЕДЕЛЕНИЕ 2. Контуром на сети назовем любую замкнутую последовательность участков на сети.

ОПРЕДЕЛЕНИЕ 3. Выбранную систему контуров на сети назовем линейно-независимой если количество контуров входящих в эту систему равно количеству элементарных колец.

ОПРЕДЕЛЕНИЕ 4. Выбранную линейно - независимую систему контуров назовем главной системой контуров если каждый из контуров содержит хотя бы один участок не входящий ни в какой другой контур системы. Такие участки назовем хордами

Пусть

k -количество элементарных колец на сети , причем $k = n - m + 1$;

Q^* – общий расход сети (л/с);

$M(n \times 2)$ – матрица связи узлов, элементами которого являются номера узлов и i-ая строка определяет участок сети;

l_i – длина i-го участка в метрах;

Z_j – высотные отметки узлов в метрах (поверхность земли относительно уровня моря);

Q_j – расход воды, потребляемой в j-ом узле (л/с);

$H_j^{св}$ – свободный напор в j-ой вершине в (м);

$D = \{d_k\}$ и $C = \{c_k\}$ – упорядоченное множество стандартных диаметров труб и соответствующее множество стоимостей одного погонного метра, $k=1,2,\ldots,r$ (r-количество труб в заданном сортаменте) ;

V_{max}, V_{min} – допустимые пределы изменения скорости (м/с).

Требуется определить:

x_i – расход воды (л/с), протекающей через i-ый участок;

h_i – потери напора воды на i-ом участке (м);

H^* – напор (м) на напорном сооружении.

Зависимость между потерями напора, диаметром, расходом воды и скоростями задается известными соотношениями из гидравлики .

Приведенные затраты представляются в виде:

$$З = \alpha \sum_{i=1}^{n} C(x_i, h_i) \cdot l_i + \beta H^* Q^* , \qquad (3.1)$$

где

$C(x_i, h_i)$ - выражает стоимость единицы длины i-го участка (руб);

α, β - нормативные коэффициенты .

Первое слагаемое выражает капитальные, а второе слагаемое эксплуатационные затраты.

Приведем систему ограничений .

Очевидно, что на сети необходимо выполнение I и II-го законов Кирхгофа:

$$\sum_{j} x_i = Q_j , \qquad (3.2)$$

где суммирование проводится по участкам, примыкающим к j-у узлу (I-ый закон Кирхгофа);

$$\sum_{r} h_i = 0, \qquad (3.3)$$

алгебраическая сумма потерь напоров по любому контуру равна нулю (II-ой закон Кирхгофа) (r=1,2,...,k).

Предположим, что напорное сооружение расположено в узле с номером 1. Тогда:

$$H^* \geq \max_{j} \left\{ H_j^{CB} - (Z_1 - Z_j) + \sum_{i \in T_j} h_i l_i \right\} , \qquad (3.4)$$

где

T_j - вектор - строка, элементами которого являются номера участков, входящие в одну из траекторий из j-го узла в узел 1.

Траектория - это путь поступления воды из корневой вершины в j-ую вершину, кроме того, должно быть соблюдено ограничение на скорость течения воды :

$$V_{min} \le V_i \le V_{max} , \qquad (3.5)$$

где

V_i - скорость течения воды на i-ом участке .

Таким образом, задача заключается в следующем : найти минимум функционала

$$З = \alpha \sum_{i=1}^{n} C(x_i, h_i)\, l_i + \beta \cdot H^* Q^*$$

при ограничениях :

$$\begin{cases} \sum_j x_i = Q_j , \\ \sum_r h_i = 0 , \\ H^* \ge \max_j \left\{ H_j^{CB} - (Z_1 - Z_j) + \sum_{i \in T_j} h_i l_i \right\} , \\ V_{min} \le V_i \le V_{max} , \end{cases}$$

3.2. Метод и алгоритм решения задачи ТЭ и ГР КС

Ввиду громоздкости задачи, для ее решения необходимо декомпозиция переменных. Наиболее естественной является декомпозиция по основным переменным : x, h . При этом сам процесс декомпозиции состоит из итераций на каждой из которых решается две задачи :

1) при фиксированных направлениях и величинах потоков x_i проводится минимизация функционала (3.1) ;
2) при фиксированных гидравлических сопротивлениях на участках (диаметрах) проводится "увязка" сети т.е. удовлетворение потоков и потерь напоров I-му и II-му законам Кирхгофа.

Алгоритм решения разбивается на четыре шага:

- определение узловых расходов Q ;
- определение первоначальных значений x ,h ;
- "увязка" сети ;
- минимизация функционала (3.1).

Блок - схема алгоритма решения задачи приводится на Рис 3.1.

3.2.1. Определение узловых расходов сети.

Очевидно, что отбор воды на сети производится вдоль участков сети и расход воды на том или ином участке зависит от характера жилой застройки улиц, от наличия промыш-

ленных предприятий и предприятий службы быта и др. (так называемые сосредоточенные водопотребители).

Условно принимается что отбор воды осуществляется только в узлах сети.

Для определения узловых расходов задается коэффициент (α_i) учитывающий интенсивность отбора воды на каждом участке. Имея общий расход сети Q и при известных длинах участков определяется удельный расход воды на участке по формуле:

$$\Delta q = \frac{Q}{\sum_{i=1}^{n} \alpha_i \, l_i}, \qquad (3.7)$$

а затем определяются путевые расходы по участкам сети:

$$q_i = \alpha_i \, l_i \Delta q \qquad (3.8)$$

и, наконец, расходы, приведенные к узлам :

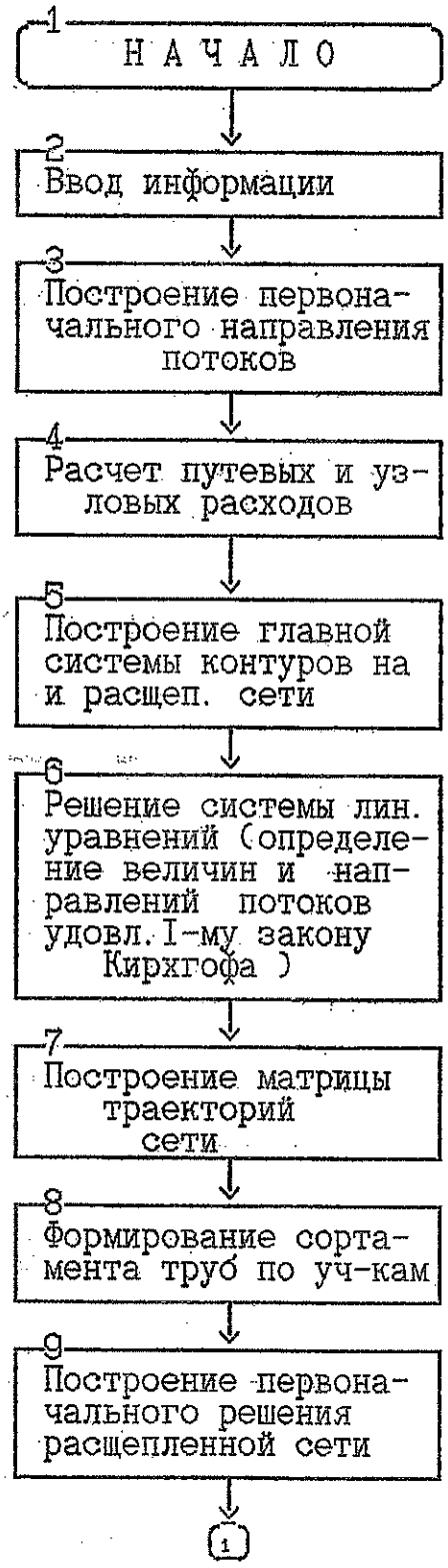
$$Q_j = 0.5 \sum_j q_i \, , \qquad (3.9)$$

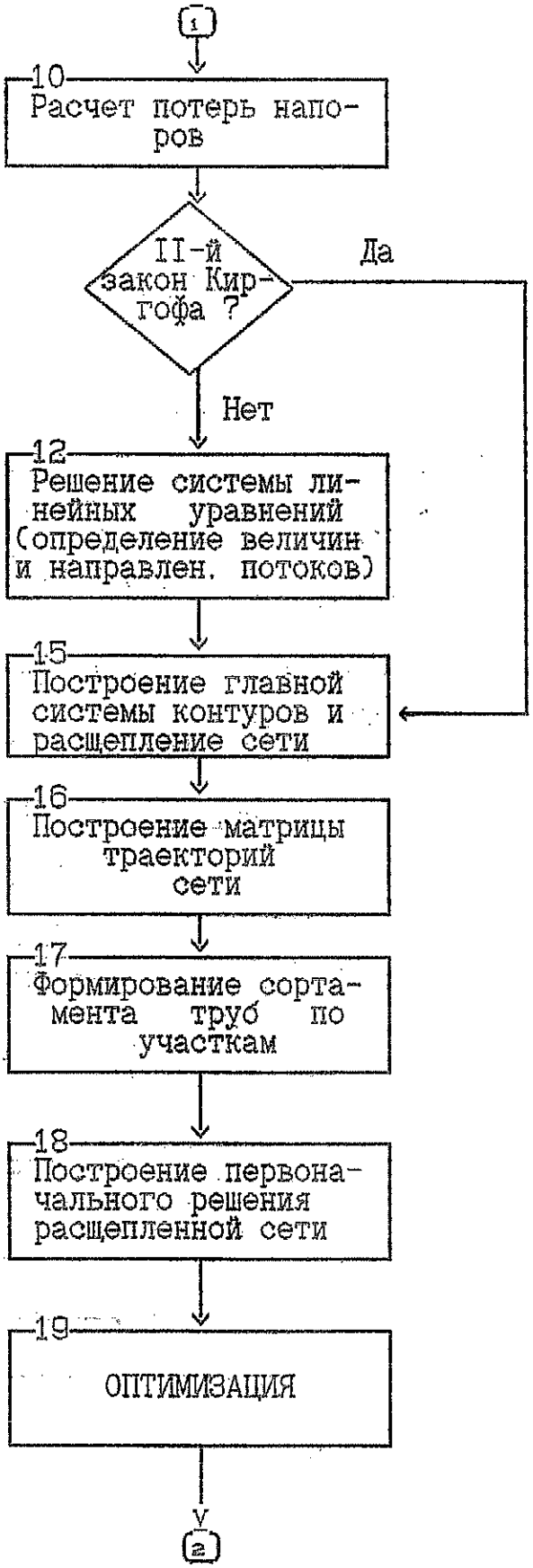
где суммирование проводится по всем участкам примыкающим к j-ому узлу.

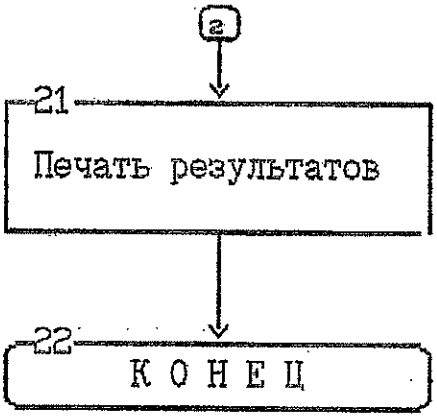
3.2.2. Определение первоначальных значений потоков и потерь напоров на сети.

Очевидно, что для определения значений потоков нам необходимо знать направление движения потоков по участкам.

Первоначальное направление потоков по сети определяется по принципу максимального растекания воды по сети.




1
10
Расчет потерь напоров
II-й закон Кирхгофа ?
Да
Нет
12
Решение системы линейных уравнений (определение величин и направлен. потоков)
15
Построение главной системы контуров и расщепление сети
16
Построение матрицы траекторий сети
17
Формирование сортамента труб по участкам
18
Построение первоначального решения расщепленной сети
19
ОПТИМИЗАЦИЯ
2



Блок-схема алгоритма решения задачи

Рис 3.1.

Имея направления потоков, мы можем выбрать на сети главную систему контуров, причем нумерацию участков сети построим таким образом, чтобы первые k участков являлись хордами.

Тогда, введя матрицу инцидентности A, отображающая связи узлов и направление потоков, первый закон Кирхгофа можно записать:

$$A\,x^T = Q^T\,, \qquad (3.10)$$

где m-1 уравнений и n неизвестных x_i;

Q^T - вектор-столбец узловых расходов;

x^T - вектор - столбец искомых потоков.

Причем:

$$a_{i,j} = \begin{cases} 1, & \text{если i-ый участок выходит из j-го узла;} \\ -1, & \text{если i-ый участок выходит из j-го узла;} \\ 0, & \text{если i- ый участок не связан с j-м узлом.} \end{cases}$$

Для определения значений x_i, удовлетворяющих (3.10) за значения первых k потоков x_i примем значения q_i (i=1,2,...,k), а остальные определим решив систему линей-

ных уравнений (3.10).

Таким образом, мы имеем значения и направления потоков по участкам сети, удовлетворяющие I-у закону Кирхгофа.

Используя ограничения

$$V_i \geq V_{min} ,$$

по известным эмпирическим зависимостям /1/ определим значения h_i соответствующие самым большим диаметрам из заданного сортамента .

Пусть B матрица размерности (k x n), причем:

$$b_{r,i} = \begin{cases} 1, \text{ если i-ый участок входит в r-ый контур его направление совпадает с направлением хорды,} \\ -1 \text{ - с противоположным направлением ,} \\ 0, \text{ если i-ый участок не входит r-ый контур ,} \end{cases}$$

где $r = 1,2,\ldots,k$.

Тогда, второй закон Кирхгофа можно записать в виде:

$$B\, h^T = 0 \qquad (3.11).$$

Для определения значений x ,удовлетворяющих второму закону Кирхгофа воспользуемся методом контурных расходов /48/, который является методом последовательных приближений .

Суть метода заключается в следующем:

N+1-ое приближение вычисляется по формуле :

$$x^{(N+1)} = x^{(N)} + \Delta x^{(N)} , \qquad (3.12)$$

где $\Delta x^{(N)}$ -приращения к N-ному приближению расходов.

Если учесть,что зависимость между потерями напора и гидравлическим сопротивлением на i-ом участке выражается

формулой :

$$h_i = s_i \; x_i \; |x_i| \; ,$$

где s_i - гидравлическое сопротивление участка, зависящее только от диаметра в области квадратичной зависимости потерь напоров от расходов, и ввести обозначения : S, X - диагональные матрицы с элементами s_i и $|x_i|$ уравнение (3.11) можно переписать в виде :

$$B \; S \; X \; x^T = 0 \qquad (3.13)$$

Приращение $\Delta x^{(N)}$ находим таким образом, чтобы:

$$B \; S \; (X^{(N)} + \Delta X^{(N)}) \; (x^{(N)} + \Delta x^{(N)}) = 0 \qquad (3.14)$$

Пренебрегая членами, содержащие $(\Delta x^{(N)})^2$ мы получим:

$$2 \; B \; S \; X^{(N)} \Delta x^{(N)} = - B \; S \; X^{(N)} \; x^{(N)} \qquad (3.15)$$

Решая систему линейных уравнений (3.15) мы находим приращения к первым k хордам $\Delta x_r^{(N)}$, (r=1,2,...,k).

Для определения приращения к остальным расходам из (3.8) получим :

$$A \; \Delta x^{(N)} = 0 \; , \qquad (3.16)$$

где

мы имеем m-1 уравнений и m-1 неизвестных x_i (i = k+1,k+2,...,n).

Таким образом процесс (3.12),(3.13) и (3.16) продолжа-

ется до тех пор, пока не выполнится

$$|x^{(N)} - x^{(N+1)}| < \varepsilon ,$$

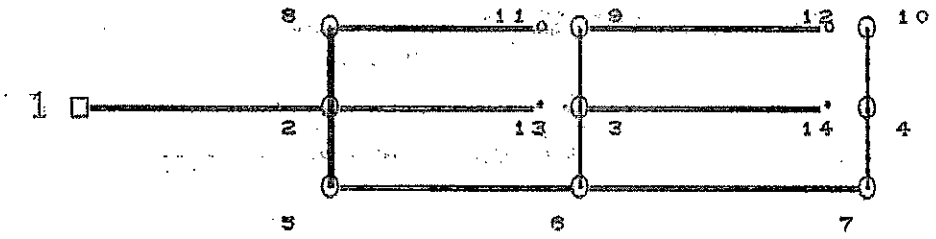
где ε - заданная точность "увязки".

Итак, x_i и h_i определены и удовлетворяют I и II -му законам Кирхгофа при условии $V_i \geq V_{min}, \forall\ i=1,2,\ldots,n$.

3.2.3. Минимизация функционала

При фиксированных значениях x_i решается задача минимизации функционала (3.1) при ограничениях (3.4)-(3.6).

Для решения задачи кольцевая сеть расщепляется в тупиковую (см. Рис 3.1.) введением дополнительных фиктивных вершин : $m+1, m+2, \ldots, m+k$ в конечных точках хорд. Очевидно для соблюдения второго закона Кирхгофа на фиктивных верши-



Расщепленная кольцевая сеть.

11-14 - дополнительные вершины

Рис 3.1.

нах необходимо выполнение условия согласования :

$$H^{CB}_{m+r} = H_{m_r}, \qquad (3.18)$$

где $r = 1,2,\ldots,k$;

m_r – номера фактических узлов, породивших узлы $m+r$;

H_{m_r} – текущий напор в узле m_r; $r = 1,2,\ldots,k$.

Таким образом, задача сведена к задаче технико-экономического и гидравлического расчета разветвленных сетей:

$$З = \alpha \sum_{i=1}^{n} C(x_i, h_i)\, l_i + \beta H^* Q^* \longrightarrow \min$$

при ограничениях :

$$\begin{cases} H_j = H^* + (Z_i - Z_j) - \sum\limits_{i \in I_j} h_i \geq H_j^{СВ}, \\ H_{m+r}^{СВ} = H_{m_r}, \\ V_{min} \leq V_i \leq V_{max}, \end{cases}$$

где $j = 1,2,\ldots,m,m+1,\ldots,m+k$, $r = 1,2,\ldots,k$ и $i = 1,2,\ldots,n$.

Условие (3.18) вносит некоторые изменения в алгоритм решения задачи.

Очевидно, введенные фиктивные вершины с номерами $m+r$ на сети являются тупиковыми. Поскольку в процессе оптимизации разветвленной сети на тупиковых вершинах, вообще говоря, обеспечивается задаваемый свободный напор, то применение алгоритма к расщепленной сети даст решение, в котором на фиктивных тупиковых вершинах обеспечивается задаваемый свободный напор, а на фактических вершинах m_r, не являющихся тупиковыми, будут обеспечиваться напоры больше задаваемого свободного напора и тем самым

$$H_{m_r} \geq H_{m+r} . \qquad (3.19)$$

Вследствие этого, выбор участков, на которых будут проводится изменение диаметров труб, разбивается на два шага: на первом шаге рассматриваются те участки сети, изменение потерь на которых влияет на напор фактических узлов т.е. тех участков, которые находятся на траекториях фактических узлов. Изменение потери напора по любому из этих участков приведет либо к изменению напора только фактической вершины либо в паре (фактической и фиктивной вершин). Таким образом, выполняется условие

$$H_{m_r} \leq H_{m+r} . \qquad (3.20)$$

На втором шаге, выбираются участки входящие в траекторию фиктивных вершин, но не входящих в траекторию соответствующих фактических вершин отбираются участки для изменения диаметров труб. Изменение диаметров труб по этим участкам осуществляется до тех пор, пока не будут соблюдены условия

$$H_{m_r} = H_{m+r} \qquad (3.21)$$

для всех фактических и фиктивных вершин.

3.3. Описание программы задачи ТЭ ГР и КС

Программное обеспечение задачи разработано на алгоритмических языках ПЛ/1 (в среде операционной системы ЕС ЭВМ)

и Си (для персональных ЭВМ совместимых с IBM PC XT/AT) объем занимаемой оперативной памяти не более 300 кб и время счета сетей содержащих около 50 вершин 20 колец и 70 участков не более 20 минут.

Программа может считать сети содержащие до 200 участков на ЭВМ типа ЕС и до 100 участков на персональных ЭВМ типа IBM .

В программе предусмотрены различные режимы расчета:

- расчет сети при заданных диаметрах на участках сети ("увязка");
- расчет сети на случай аварии на заданных участках сети;
- расчет сети на случай пожара в заданных точках(узлах);
- расчет сети при фиксированном напоре на вершине сети;
- расчет сети в оптимальном режиме (определение величин и направлений потоков по участкам диаметров труб и напора на напорном сооружении).

После расчета сети в оптимальном режиме программа автоматически проводит расчеты на случай аварии и на случай пожара если в исходных данных заданы эти режимы.

Данная программа находится в промышленной эксплуатации в составе подсистемы ПВС САПР-СКГВХ с 1984 года.

Программное обеспечение содержит 11 основных модулей:

- Ввод и печать входной информации;
- Формирование рабочего сортамента диаметров труб из

- файла базы данных по сортаментам труб;
- Определение первоначального направления потоков;
- Определение узловых расходов сети;
- Построение системы контуров;
- Расщепление кольцевой сети в тупиковую;
- Определение первоначальных значений диаметров труб участков и их гидравлических сопротивлений;
- "Увязка" сети ;
- Определение оптимального напора и диаметров труб участков сети;
- Расчет потерь напоров ;
- Печать выходных данных;

Информационное обеспечение программы делится на нормативно-справочную (информация по всем сортаментам труб) и оперативную информации .

Структура данных по НСИ та же, что приведена в подразделе 2.4. Оперативная информация делится на входную и выходную информации.

Входная информация включает в себя:

- наименование объекта;
- информационная карта, в которой задаются режимы расчета сети;
- точность "увязки" сети;
- шифры сортаментов труб, которыми предполагается проектировании сети;
- диапазон допустимых значений скорости течения воды по трубам (м/с);

- количество вершин и узлов сети;
- номер узла;
- высотная отметка узла (м);
- значение свободного напора в узле (м);
- значение сосредоточенного расхода в узле(л/с);
- значение расхода в узле на случай пожара(л/с);
- участки сети в виде пар чисел;
- длина участка(м);
- расход воды на участке($м^3$/с)(начальное значение и направление потоков могут быть заданы) ;
- значение диаметра трубы(м) участка и его шифр если выбран соответствующий режим;
- коэффициент путевого отбора;
- количество вариантов аварийных ситуаций и перечень аварийных участков;

Выходная информация включает :

- величины "невязок" по всем контурам выбранной системы контуров;
- участок сети в виде номеров вершин(начало-конец);
- длина участка(м), расход воды на участке (л/с);
- значение большего диаметра (м), толщина стенки, (м) Гост, марка, длина (м) и потери напора по длине (м);
- значение меньшего диаметра (м), толщина стенки, (м) ГОСТ, марка, длина (м) и потери напора по длине (м);
- потери напора по длине участка (м);
- стоимость труб участка (руб.);

- номер узла;
- фактический напор в узле (м);
- высотная отметка узла (м);
- пьезометр узла(м);
- напор на вершине сети (м);
- приведенная стоимость сети;
- результаты проверочных расчетов на случай аварии и пожара.

В приложении 6 приводятся распечатки контрольных примеров, а в приложении 7-распечатки исходных текстов модулей программы.

ЗАКЛЮЧЕНИЕ

1) Мелиорация земель является одним из основных направлений научно-технического прогресса в сельском хозяйстве, важнейшим элементом комплекса мероприятий по интенсификации сельскохозяйственного производства.

Главной задачей в разработке мелиоративных систем является повышение эффективности капитальных вложений и качества строительства. Это достигается путем дальнейшего совершенствования проектного дела и использования современной вычислительной техники. Это в свою очередь, предполагает математическое моделирование систем водоснабжения с целью получения оптимальных, технически совершенных мелиоративных и водохозяйственных систем с последующим включением их в состав автоматизированных систем.

2) Проведены исследования задач и построены модели систем водоснабжения включающие напорно -регулирующие сооружения и водопроводную сеть.

3) Разработаны эффективные методы решения описываемых моделей.

4) Разработаны программные обеспечения задач рассматриваемых в диссертации для ЭВМ серии ЕС-1045 и для персональных компьютеров совместимых с IBM PC AT/XT и включены в состав соответствующих подсистем САПР - СКГВХ .

5) Практическая реализация задач показала, что использование разработанных программных обеспечений в проектной практике способствует повышению надежности , сокращению сроков

проектирования систем и существенному снижению капитальных и эксплуатационных затрат.

6) Результаты диссертации используются в промышленной эксплуатации в проектных институтах Минводхоза РСФСР.

ЛИТЕРАТУРА

1. Абрамов Н.Н. Водоснабжение. -М.: Стройиздат, 1974. - 328 с.
2. Абрамов Н.Н. Обоснование и методы зонирования водопроводных систем. -Изд. МКХ РСФСР, М-Л., 1949. -152 с.
3. Абрамов Н.Н., Поспелова М.М. Расчет водопроводных сетей. - М.: Госстройиздат, 1962. - 311 с.
4. Абрамов Н.Н. Теория и методика расчета систем подачи и распределения воды. -М.: Стройиздат, 1972. - 288 с.
5. Абрамов Н.Н. Графические методы расчета водопроводных систем. - М.: 1946. - 136 с.
6. Азизов Г.Х. Исследование оптимальных параметров точек питания в групповых системах водоснабжения: Автореф. дис. канд. техн. наук. - Ташкент., 1974. -19 с. Ташкентский политехнический институт.
7. Алихашкин Я.И., Юшкин А.Р. Применение ЭВМ для гидравлических расчетов водопроводных сетей. -ГХМ, 1960. -178 с.
8. Альтшуль А.Д. Гидравлические сопротивления. - М.: Недра, 1970. -216 с.
9. Алферова З.В., Ежова В.П. Применение теории графов в экономических расчетах. - М.: Статистика, 1971. - 219 с.
10. Андрияшев М.М. Техника расчета водопроводной сети. - М.: Сов. законодательство, 1932. - 62 с.
11. Андрияшев М.М. Гидравлические расчеты водоводов и водопроводных сетей. - М.: Госстройиздат, 1964. - 107 с.
12. Белан А.Е. Универсальный метод гидравлического увязочного расчета кольцевых водопроводных сетей. // Изв. вузов. Строительство и архитектура, 1964, № 4, с. 69-73.

13. Белозеров Н.П.,Луговской М.В. Расчет систем водоснабжения с применением вычислительной техники.- М.: Колос, 1973.-213 с.

14. Берман Р.Я., Бобровский С.А., Галиуллин З.Т. Расчет режимов работы закольцованной системы газопроводов на ЭВМ. // Газовая промышленность, 1966,№ 12,с.14-16.

15. Васильченко М.П. Расчет кольцевых водопроводных сетей путем нахождения полных поправочных расходов.//Изв.вузов,Строительство и архитектура ,1964, № 6,с.80-90.

16. Виишневский К.П. Механизация расчета кольцевых водопроводных сетей.//Водоснабж. и сантехника,1961,№ 4,с.20-24.

17. Галустова Л.А.,Шор Н.З. Нахождение оптимальной конфигурации разомкнутых распределительных сетей.// Экономическая кибернетика и исследование операций. Вып.2., Киев, 1967.- 127 с.

18. Гениев Н.Н.Примеры расчета водопроводов.- М.;Л.: Госиздат, 1930.-57 с.

19. Гольштейн В.Г., Юдин Д.Б. Задачи линейного программирования транспортного типа.- М.:Наука,1969.- 223 с.

20. Данцинг Дж. Линейное программирование, его применение и обобщение.- М.:Прогресс, 1966.-600 с.

21. Зингер Н.М., Андреева К.С., Вульман Ф.А. Расчет многокольцевых гидравлических сетей на ЭВМ Урал.// Теплоэнергетика, 1960,№ 12 ,с.44-52.

22. Зингер М.Н. Гидравлические и тепловые режимы теплофикационных систем.- М.: Энергия, 1976.-336 с.

23. Зингер М.Н. Гидравлический расчет тепловых сетей с переменными расходами у абонентов на ЭВМ Урал.//Теплоэнергетика, 1962,№ 7,с.27-32.

24. Зуховицкий С.И., Авдеева А.И. Линейное и выпуклое программирование. - М.: Наука, 1967. -213 с.

25. Ибрагимов И.И., Азизов Г.Х. Оптимальное прикрепление водопотребителей к источникам водоснабжения. //Известия АН Узб.ССР Сер. Техническая наука, 1971, № 6, с.25-31.

26. Ильин В.Г. Расчет совместной работы насосов, водопроводных сетей и резервуаров. -Киев, Стройиздат, 1963. -145 с.

27. Каменев П.Н. Смешивание потоков. -М.; Л: ОНТИ 1936. -188 с.

28. Карамбиров Н.А., Барезин В.А. Сельскохозяйственные групповые системы водоснабжения. //Водоснабжение и санитарная техника, 1966, № 8, с. 17-22.

29. Кемелев А.А. Групповые системы сельскохозяйственного водоснабжения. -М., Колос, 1971. -156 с.

30. Кикачейшвили Г.Е., Начкебия В.П., Томарадзе Дж.С. Технико-экономический расчет водопроводных сетей и закрытых оросительных сетей с помощью ЭВМ. / Тезисы докладов XIX респ. конференции профессорско - препод. состава ГПИ им. В.И. Ленина и работников производства. Тбилиси, 1975, с. 12-23.

31. Кикачейшвили Г.Е. Технико-экономический расчет разветвленных водопроводных сетей методом линейного программирования. // Водоснабжение и санитарная техника, 1969, № 6. с. 7-8.

32. Кирсанов М.В. Зонные системы водоснабжения. //Труды МИЖДТ, вып. 88/9, 1967, с. 35-37.

33. Кирсанов М.В. Экономический расчет водопроводных сетей. - М.; Л.: изд-во МКХ РСФСР, 1949. -148 с.

34. Койда Н.У. Применение вычислительных машин при проектировании водопроводных сетей. Новые методы (обзор). - М.:, Высшая школа, 1969. - 36 с.

35. Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование.-М.:,Наука,1969.-368 с.

36. Кузьмин Ю.М., Небельсин Г.П. К вопросу аналитического выражения стоимости прокладки водопроводных труб.//Изв. вузов. Строительство и архитектура,1974,№ 6, с.37-40.

37. Лаеис А.Я. Технико-экономический расчет кольцевых водопроводных сетей с помощью ЭВМ./Математические методы в экономике. Рига, 1967, Вып. 1, с. 39-43.

38. Левин А.М., Смирнов В.А., Черкасова А.Я. Расчет многокольцевых городских газовых сетей на ЭВМ.// Газпром, 1961,№ 11, с.33-34.

39. Леонас В.Л., Моцкус И.Б. Об одной нелинейной задаче математического программирования. Математические методы решения экономических задач.- М., Наука, 1969,Сборник 1, (ЦЭМИ АН СССР).-187 с.

40. Леонтьев В.А.,Хорунжий П.Д. Аналитический метод расчета на ЭВМ систем водоснабжения с несколькими водоисточниками.Межведомственный научно-технический сборник.- Киев, Урожай,1969, вып.10.-127 с.

41. Лобачев В.Г. Примеры расчета водопроводных сетей.-М.: Издательство МКХ РСФСР, 1950.- 110 с.

42. Лобачев В.Г. Новый метод увязки колец при расчете водопроводных сетей.//Сан.техника, 1934,№ 2, с.8-12.

43. Лобачев В.Г. Вопросы рационализации расчетов водопроводных сетей.-М.:ОНТИ , 1936.- 148 с.

44. Максименко Ф.Е. Различные расчеты по курсу водопроводов.- М.:Типо-лит. Рихтер,1910.- 102 с.

45. Мелентьев Л.А. Теплофикация.- М.;Л.:АН СССР. ч.1,1944, с. 248; ч.2 1948.-с.276.

46. Меренков А.П. Математические модели и методы для анализа и оптимального проектирования трубопроводных систем. Автореф. на соис. ...д-ра физ.-мат. наук. Новосибирск: СО АН СССР, 1974. - 34 с.

47. Меренков А.П. и Хасилев В.Я. Расчет развлетвленных тепловых сетей на основе их оптимизации с использованием ЭВМ. //Изв. СО АН СССР. Сер. техн. наук, 1963, № 10, вып. 3, с. 42-48.

48. Меренков А.П., Хасилев В.Я. Теория гидравлических цепей. - М.: Наука, 1985. - 280 с.

49. Мошнин Л.Ф., Гальперин М.Е., Глазунов Е.М. Расчет систем водоснабжения с использованием вычислительной техники. //Водоснабжение и санитарная техника. 1966, № 4, с. 8-13.

50. Мошнин Л.Ф. Методы технико - экономического расчета водопроводных сетей. - М.: Стройиздат, 1950. -144 с.

51. Мошнин Л.Ф. Выбор диаметров водопроводных линий. //Водоснабжение и санитарная техника. 1940, № 2/3 с. 48-55.

52. Некрасова О.А., Хасилев В.Я. Оптимальное дерево трубопроводной системы. //Экономика мат. методы. 1970, т. 4, № 3, с. 427-432.

53. Пустовойт Б.В. Механика движения жидкостей в трубах. -Л.: Недра, 1971. - 144 с.

54. Пшеничный Б.Н. Расчет энергетических сетей на ЭВМ. //Выч. мат и мат. физика. 1962, № 5, с. 942-947.

55. Савельева Н.И. Трассировка сети групповых водопроводов. //Водоснабжение и санитарная техника, 1964, № 7. с. 2-7.

56. Сироткин В.П. Схемы и расчет водопроводов и водопроводных сетей. - М.: Высшая школа, 1968. - 157 с.

57. Ставровский Е.Р., Сухарев М.Г. Универсальная программа расчета газосборных сетей. //Газпром. 1965, № 7, с. 10-11.

58. Сумароков С.В. Применение динамического программирования оптимального проектирования расширяемых и реконструируемых разветвленных водопроводов. //Изв. вузов. Строительство и архитектура, 1975, № 11, с. 125-129.

59. Толмачева Н.И., Хасилев В.Я. Программа расчета многокольцевых гидравлических сетей увязочным методом. - М.: ГИПРОТИС Госстроя СССР, 1965, вып. 1-4. - 21с.

60. Хасилев В.Я., Меренков А.П., Сумароков С.В. О выборе диаметров труб разветвленных тепловых сетей с использованием ЭВМ. //Теплоэнергетика, 1966, № 6, с. 60-65.

61. Хасилев В.Я. Линейные и линеаризованные преобразования схем гидравлических цепей. // Изв. АН СССР, 1964, № 1, с. 69-88.

62. Хасилев В.Я. Гравитационные гидравлические цепи с распределенными параметрами и методика их расчета. - Иркутск, Вост.-Сиб. кн. изд. 1966, с. 349-362.

63. Шифринсон Б.Л. Основной расчет тепловых сетей. - М.; Л.: Госэнергоиздат. 1940. - 188 с.

64. Cross H. Analysis of flow in networks of conduits or conductors. Urbana, Illinois, Eng. Exp. Station of Univ. of Illinois, 1936, November, Bull. N 286, 29 p.

65. Murge D. Essai sur les machines d'aerage. -Bull. de la Sos. de L'Ind. Minerale, 1873, part 1, p. 464-472.

66. Щевелев Ф.А. Таблицы для гидравлического расчета стальных, чугунных, асбестоцементных, пластмассовых и стеклянных водопроводных труб. - М.:, Стройиздат, 1973. - 114 с.

67. Байрактаров Б.Р. Технико-экономический и гидравлический расчет магистральных водопроводов. Межвузовский сборник. Математические методы в мелиорации. Нальчик, 1981. с. 25-31.

68. Байрактаров Б.Р. и Крахмалев В.Ю. Задача выбора параметров кольцевой сети. Сборник научных трудов. Нелокальные задачи для уравнений в частных производных и их приложения к моделированию и автоматизации проектирования сложных систем. Нальчик, 1986. с. 120-122.

69. Аттаев А.Х. Байрактаров Б.Р. Кудаев В.Ч. Задача трассировки оросительной сети. Сборник докладов третьей Всесоюзной конференции по перспективным методам планирования и анализа экспериментов. Гродно, 1988. с. 7-13.

70. Математическое обеспечение системы автоматизированного проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения. Технический проект. Том 3. ТЛП закрытой оросительной сети . Кн. 3.2. Постановка задачи. Руководитель А.М. Нахушев. № ГР 79039202; Нальчик, 1978. с. 78.

71. Математическое обеспечение системы автоматизированного проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения Рабочий проект. Том 3. ТЛП закрытой оросительной сети Кн. 3.4. Специальное математическое обеспечение . Руководитель А.М. Нахушев. № ГР 79039202; Нальчик, 1978. с. 121.

72. Математическое обеспечение системы автоматизированного проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения. Технический проект. Том 3. ТЛП закрытой оросительной сети. Кн. 3.1. Общая часть. Руководитель А.М. Нахушев. № ГР 79039202; Нальчик, 1979. с. 154.

73. Математическое обеспечение системы автоматизированного

проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения. Рабочий проект. Том 3. ТЛП закрытой оросительной сети. Кн. 3.2. Специальное программное обеспечение. Руководитель А.М. Нахушев. № ГР 79039202; Нальчик, 1979. с. 178.

74. Математическое обеспечение системы автоматизированного проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения. Рабочий проект. Том 8. ТЛП магистральных водопроводов для сельскохозяйственного водоснабжения. Кн. 8.1. Общая часть. Руководитель А.М. Нахушев. № ГР 79039202; Нальчик, 1979. с. 233.

75. Математическое обеспечение системы автоматизированного проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения. Рабочий проект. Том 8. ТЛП магистральных водопроводов для сельскохозяйственного водоснабжения. Кн. 8.2. Специальное программное обеспечение. Руководитель А.М. Нахушев. № ГР 79039202; Нальчик, 1979. с. 215.

76. Система автоматизированного проектирования оросительных систем и систем магистральных водопроводов для Севкавгипроводхоза (САПР-СКГВХ). Технический проект (II очередь). Том 4. Подсистема поселковых водопроводных сетей. Кн. 4.1. Пояснительная записка. Руководитель А.М. Нахушев. № ГР 31055364; Нальчик, 1982. с. 67.

77. Система автоматизированного проектирования оросительных систем и систем магистральных водопроводов для Севкавгипроводхоза (САПР - СКГВХ). Технический проект

(II очередь). Том 4. Подсистема поселковых водопроводных сетей (ПВС). Научно-технический отчет. Руководитель А.М. Нахушев. № ГР 31055364; Нальчик, 1983. с. 97.

78. Система автоматизированного проектирования оросительных систем и систем магистральных водопроводов для Севкавгипроводхоза (САПР - СКГВХ). (II очередь). Подсистема поселковых водопроводных сетей (ПВС). Рабочий проект. Том 4. Кн. 4.1. Пояснительная записка. Руководитель А.М. Нахушев. № ГР 31055364; Нальчик, 1984. с. 257.

79. Научно-исследовательский отчет по подсистеме ПВС (заключительный). Руководитель А.М. Нахушев. № ГР 31055364; Нальчик, 1986. с. 138.

Приложение 1

Акт приемки-сдачи научно-исследовательской работы

по теме № 235

Кабардино-Балкарский
госуниверситет
Р/счет 14117 в Гор-
управлении Госбанка
г. Нальчика КБАССР

Приложение №
Минводхоз РСФСР
(наименование заказчика и
р/счет 387908 в Мосгорбюдбанке
его реквизиты)
г.Москвы

АКТ ПРИЕМКИ-СДАЧИ

научно-исследовательской работы по теме № 235 **от** I января 1983 г

Мы, нижеподписавшиеся, представители предприятия Заместитель начальника Главного управления проектных работ
(наименование предприятия)
Омельяненко Виктор Анатольевич
(фамилия, имя, отчество, должность)
с одной стороны, и представители Кабардино-Балкарского госуниверситета проректор по НИР Лесев Виталий Дмитриевич
(фамилия, имя, отчество, должность)
с другой стороны, составили настоящий акт о завершении и приемке работы Развить и ввести в действие в Секвавгипроводхозе систему автоматизированного проектирования оросительных систем и систем магистральных водопроводов для сельскохозяйственного водоснабжения (САПР-СКГВХ II очереди)
(наименование работы)
выполненной кафедрой ~~НИИ прикладной математики и механики~~ КБГУ в
период с I января 19 г. по 31 декабря 1985 г.

Результатом проведенной работы является:
Программный комплекс
САПР-СКГВХ II очереди

Результаты НИР внедрены на предприятии в следующем виде:
Принято к промышленному производству следующие подсистемы САПР-СКГВХ II очереди: развитие подсистемы "Закрытых оросительных сетей", разработка подсистем "Насосных станций", "Поседковых сетей водоснабжения", "Уровенного и содевого прогноза", развитие подсистемы "Магистральные водопроводы".

комендовано к опытно-промышленной проверке

рняты для использования в работах предприятия: программное обеспечение САПР-СКТВХ II очереди

етодика

екомендации

акет

роект

кспериментальные данные

ехнология

атериал

роекты стандартов

расчет

ри внедрении результатов НИР на предприятии получен годовой экономический эффект

1млн. 509 тыс. руб.

дается годовой экономический эффект руб.

ыв о работе (с указанием достигнутого научно-технического эффекта) Разработка програм-
ого обеспечения осуществлена на базе современных методов оптимизации
атематического аппарата. Так при реализации задачи гидравлического
счета закрытой оросительной сети применен впервые в отрасли метод
намического программирования, как наиболее точно моделирующий процесс
ижения жидкости в трубопроводах, задача трассировки осуществлялась по
году Хасидева-Меренкова, а при решении задач уровенного и солевого
огнозов были предложены совершенно новые методы, основанные на решений
эшанных дифференциальных уравнений и др.

ечания по работе:

а выполненную научно-исследовательскую работу в окончательный расчет с учетом перечислен-
авансов исполнителю в соответствии со сметной стоимостью работы причитается

(сумма

прописью)

Предлагается продолжение работ по указанной тематике и включение в план вуза на 19 —19 г.г.

оты:

Создать новые и развить действующие подсистемы САПР мелиоративных
и водохозяйственных систем (САПР-СКГВХ третья очередь)

Проректор КБГУ по НИР В.Д.Лесев

От предприятия

Минводхоза РСФСР

В.А.Омельяненко

г СКГВХ

С.П.Флоринский

От высшего учебного заведения

Руководитель НИС В.М.Кузамышев

Научный руководитель А.М.Нахушев

темы

Руководитель темы Ю.К.Назаров

хладненская гор. тип. Заказ № 5108—1 000.

Приложение к договору № 235

Расчет отчислений в фонды экономического стимулирования, включаемых в [illegible] стоимость темы

выполняемой по хоздоговору № 235 от 27.12.82 Минводхозом РСФСР

в зависимости от фактического экономического эффекта.

	Дата начала и окончания темы (этапа)	фактического экономического эффекта в целом по разработке в т. руб.	Плановый размер отчислений в ФП в зависимости от эф. по всей разраб. т. р		Доленое участие исполнителя (вуза) в ФП за разработку в целом, процент.	Сумма средств, включаемых в сметную стоимость темы на образование ФП в тыс. руб. в т. ч.:			Из общей суммы ФП направляются в т. р.		в ФСКМ и ЖС	Плановые накопления, включаемые в сметную стоимость темы на образ. ФРНИБ в тыс. руб.	Отношения ФМП по теме и план. фонду з-п исполнит
			процент	сумма в тыс. руб.		всего	в виде аванса (до 30%)	после окончан. темы	централиз. фонды поощрения объедин.	в ФМП организ. (вуза)			
А	Б	1	2	3	4	5	6	7	8	9	10	11	12
Всего по теме:	1.01.83	1509	3,32	50,0	100,0	50	–	50	2,5	28,5	19,0	–	8,6
в том числе:	31.12.85												

Развить и ввести в действие в Севкавгипроводхозе систему автоматизированного проектирования оросительных систем и магистральных водопроводов для сельхозводоснабжения (САПР-СКГВХ II очередь) по техническому заданию СКГВХ.

Исполнитель проректор по НИР, Лисев В.Д. (подпись, печать, дата) Заказчик: (подпись, печать, дата)

Прохладненская гор. Заказ № 5712—1000.

Приложение 2

Распечатка контрольного примера по задаче Технико-экономического расчета закрытых оросительных сетей

САПР - СКГВХ

ТЛП-30С

ВХОДНЫЕ ДАННЫЕ

Таблица П.2.1

Параметры расчета сети

Режим расчета	Минимальная скорость	Максимальная скорость	Коэффициенты капвложений	Коэффициенты эксплуатационных затрат	Количество сортаментов
0	0.30	3.51	1.14	7.56	1

Таблица П.1.2

Заданные шифры сорптаментов

ШИФР СОРТАМЕНТА
1200

Таблица П.2.3

Матрица связи узлов дерева сети и координаты вершин

Количество вершин сети (строк в таблице)- 27

Матрица вершин базового графа или дерева										Координаты вершин		
										X	У	Z
1										2	3	4
1	2	0								-747.7	-747.7	210.40
2	3	6	26	7	0					0.0	0.0	207.40
3	26	7	8	4	2	0				1020.0	0.0	193.40
4	7	8	9	5	3	0				2040.0	0.0	181.50
5	8	9	4	0						3060.0	0.0	170.80
6	7	11	10	26	2	0				0.0	1000.0	214.00
7	10	27	8	26	11	6	3	4	0	1020.0	1000.0	200.70
8	27	12	13	9	3	4	5	7	0	2040.0	1000.0	188.00
9	12	13	4	5	8	0				3060.0	1000.0	177.20
10	11	15	14	7	6	0				0.0	2000.0	223.70

продолжение таблицы П.1.3

1										2	3	4
11	12	16	15	27	14	7	6	0		1020.0	2000.0	213.00
12	13	17	16	27	15	8	9	11	0	2040.0	2000.0	201.00
13	16	17	9	8	12	0				3060.0	2000.0	188.50
14	15	19	18	10	11	0				0.0	3000.0	234.00
15	18	19	20	16	14	10	11	12	0	1020.0	3000.0	221.90
16	19	20	21	17	12	13	0			2040.0	3000.0	212.40
17	20	21	12	13	16	0				3060.0	3000.0	199.90
18	19	22	23	14	15	0				0.0	4000.0	234.80
19	22	20	24	23	14	15	16	18	0	1020.0	4000.0	231.70
20	23	24	25	17	19	15	16	21	0	2040.0	4000.0	223.90
21	24	25	16	17	20	0				3060.0	4000.0	213.90
22	23	18	19	0						0.0	5000.0	222.40
23	24	18	19	20	22	0				1020.0	5000.0	230.00
24	25	23	19	20	21	0				2040.0	5000.0	233.20
25	20	24	21	0						3060.0	5000.0	227.20
26	3	2	7	6	0					510.0	500.0	0.00

продолжение таблицы П.1.3

1	2	3	4
27 7 8 12 11 0	1530.0	1500.0	0.00

Таблица П.1.4

Исходные данные по гидрантам сети

Количество строк в таблице- 24

Номер узла-гидранта	Тип гидранта	Раасход на гидранте	Свободный напор	Количество поливов 4									
				номмера поливов									
				1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	0	92.0	85.0	1	0	1	0						
3	0	92.0	85.0	1	0	0	1						

Продолжение таблицы П.1.4

1	2	3	4	5	6	7	8	9	10	11	12	13	14
4	0	92.0	85.0	1	0	1	0						
5	0	92.0	85.0	1	0	0	1						
6	0	92.0	85.0	0	1	0	1						
7	0	92.0	85.0	0	1	1	0						
8	0	92.0	85.0	0	1	0	1						
9	0	92.0	85.0	0	1	1	0						
10	0	92.0	85.0	1	0	1	0						
11	0	92.0	85.0	1	0	0	1						
12	0	92.0	85.0	1	0	1	0						
13	0	92.0	85.0	1	0	0	1						
14	0	92.0	85.0	0	1	0	1						
15	0	92.0	85.0	0	1	1	0						
16	0	92.0	85.0	0	1	0	1						
17	0	92.0	85.0	0	1	1	0						
18	0	92.0	85.0	1	0	1	0						
19	0	92.0	85.0	1	0	0	1						

Продолжение таблицы П.1.4

1	2	3	4	5	6	7	8	9	10	11	12	13	14
20	0	92.0	85.0	1	0	1	0						
21	0	92.0	85.0	1	0	0	1						
22	0	92.0	85.0	0	1	0	1						
23	0	92.0	85.0	0	1	1	0						
24	0	92.0	85.0	0	1	0	1						
25	0	92.0	85.0	0	1	1	0						

В Ы Х О Д Н Ы Е Д А Н Н Ы Е

Таблица П.1.5

Результаты расчета задачи трассировки сети

Участок	Длина участка	Расход на участке	Высотная отметка конца	Координаты узлов сети: начала участка X	начала участка У	конца участка X	конца участка У
1	2	3	4	7	8	5	6
0 - 1	0.0	1.1040	210.40			-747.7	-747.7
1 - 2	1057.4	1.1040	207.40	-747.7	-747.7	0.0	0.0
2 - 3	1020.0	0.2760	193.40	0.0	0.0	1020.0	0.0
3 - 4	1020.0	0.0920	181.50	1020.0	0.0	2040.0	0.0
9 - 5	1000.0	0.0920	170.80	3060.0	1000.0	3060.0	0.0
2 - 6	1000.0	0.8280	214.00	0.0	0.0	0.0	1000.0
3 - 7	1000.0	0.0920	200.70	1020.0	0.0	1020.0	1000.0
3 - 8	1428.4	0.1840	188.00	1020.0	0.0	2040.0	1000.0
8 - 9	1020.0	0.0920	177.20	2040.0	1000.0	3060.0	1000.0

Продолжение таблицы П.1.5

1	2	3	4	7	8	5	6
6 -10	1000.0	0.0920	223.70	0.0	1000.0	0.0	2000.0
6 -11	1428.4	0.6440	213.00	0.0	1000.0	1020.0	2000.0
11 -12	1020.0	0.0920	201.00	1020.0	2000.0	2040.0	2000.0
17 -13	1000.0	0.0920	188.50	3060.0	3000.0	3060.0	2000.0
10 -14	1000.0	0.0920	234.00	0.0	2000.0	0.0	3000.0
11 -15	1000.0	0.4600	221.90	1020.0	2000.0	1020.0	3000.0
11 -16	1428.4	0.1840	212.40	1020.0	2000.0	2040.0	3000.0
16 -17	1020.0	0.0920	199.90	2040.0	3000.0	3060.0	3000.0
15 -18	1428.4	0.0920	234.80	1020.0	3000.0	0.0	4000.0
15 -19	1000.0	0.0920	231.70	1020.0	3000.0	1020.0	4000.0
15 -20	1428.4	0.1840	223.90	1020.0	3000.0	2040.0	4000.0
20 -21	1020.0	0.0920	213.90	2040.0	4000.0	3060.0	4000.0
18 -22	1000.0	0.0920	222.40	0.0	4000.0	0.0	5000.0
19 -23	1000.0	0.0920	230.00	1020.0	4000.0	1020.0	5000.0
20 -24	1000.0	0.0920	233.20	2040.0	4000.0	2040.0	5000.0
21 -25	1000.0	0.0920	227.20	3060.0	4000.0	3060.0	5000.0

Общая длина сети равна 26.32 км.

Таблицы П.1.6

Результаты гидравлического расчета сети

Участок	Длина участка	Расход на участке	Высотная отметка	Скорость воды	Диаметры	Толщина стенки	Код материала	Потери напора на участке	Напор в узле	Стоимость участка
1	2	3	4	5	6	7	8	9	10	11
0- 1			210.4						117.4	
1- 2	1057.4	1104	207.4	1.15	1120	8.0	1200	1.47	118.9	46.84
2- 3	1020.0	276	193.4	2.29	402	5.0	1200	21.22	111.7	8.61
3-200	916.6	92	182.7	2.12	245	5.0	1200	31.90	90.5	4.70
200- 4	103.4	92	181.5	2.68	219	5.0	1200	6.70	85.0	0.47
9- 5	1000.0	92	170.8	1.40	299	5.0	1200	11.63	85.0	6.28
2- 6	1000.0	828	214.0	1.28	920	7.0	1200	2.21	110.1	32.30
3-201	985.0	92	200.6	1.69	273	5.0	1200	18.88	85.6	5.64
201- 7	15.0	92	200.7	2.12	245	5.0	1200	0.52	85.0	0.08
3-202	293.6	184	192.3	1.74	377	5.0	1200	3.85	109.0	2.34

Продолжение таблицы П.2.6

1	2	3	4	5	6	7	8	9	10	11
202- 8	1134.9	184	188.0	2.02	351	5.0	1200	21.96	91.3	8.40
8- 9	1020.0	92	177.2	1.40	299	5.0	1200	11.86	90.2	6.41
6- 10	1000.0	92	223.7	0.76	402	5.0	1200	2.48	97.9	8.44
6- 11	1428.4	644	213.0	1.00	920	7.0	1200	1.96	109.1	46.14
11-203	998.4	92	201.3	2.12	245	5.0	1200	34.75	86.1	5.12
203- 12	21.6	92	201.0	2.68	219	5.0	1200	1.40	85.0	0.10
17- 13	1000.0	92	188.5	1.40	299	5.0	1200	11.63	85.0	6.28
10-204	836.9	92	232.3	0.76	402	5.0	1200	2.07	87.2	7.06
204- 14	163.1	92	234.0	0.87	377	5.0	1200	0.56	85.0	1.30
11-205	710.7	460	219.3	0.71	920	7.0	1200	0.52	102.3	22.96
205- 15	289.3	460	221.9	0.90	820	7.0	1200	0.38	99.3	8.33
11-206	464.1	184	212.8	1.74	377	5.0	1200	6.08	103.3	3.69
206- 16	964.4	184	212.4	2.02	351	5.0	1200	18.66	85.0	7.14
16-207	965.4	92	200.6	1.40	299	5.0	1200	11.23	85.6	6.06
207- 17	54.6	92	199.9	1.69	273	5.0	1200	1.05	85.2	0.31
15- 18	1428.4	92	234.8	0.53	480	5.0	1200	1.44	85.0	15.16

продолжение таблицы П.1.6

1	2	3	4	5	6	7	8	9	10	11
15-208	664.7	92	228.4	0.76	402	5.0	1200	1.65	91.2	5.61
208- 19	335.3	92	231.7	0.87	377	5.0	1200	1.15	86.7	2.67
15-209	1167.7	184	223.5	0.62	630	7.0	1200	1.09	96.6	26.97
209- 20	260.7	184	223.9	1.06	480	5.0	1200	0.94	95.3	2.77
20-210	981.7	92	214.3	0.87	377	5.0	1200	3.38	101.6	7.81
210- 21	38.3	92	213.9	1.01	351	5.0	1200	0.19	101.7	0.28
18-211	897.6	92	223.7	1.40	299	5.0	1200	10.44	85.7	5.64
211- 22	102.4	92	222.4	1.69	273	5.0	1200	1.96	85.0	0.59
19- 23	1000.0	92	230.0	0.87	377	5.0	1200	3.44	85.0	7.96
20- 24	1000.0	92	233.2	0.53	480	5.0	1200	1.01	85.0	10.61
21- 25	1000.0	92	227.2	0.87	377	5.0	1200	3.44	85.0	7.96

Примечание: узлы с номерами от 200 и выше в таблице П.1.6 это точки разбиения участка при укладке в ней двух различных диаметров.

Таблицы П.1.7

Распределение труб и материала по сети

По материалам всего			в том числе по диаметрам			
наименование материала	Марка	ГОСТ или ТУ	внешний диаметр	толщина стенки	длина диаметра	стоимость диаметра
1	2	3	4	5	6	7
С	ЭЛЕКТРОСВ.	10704-76	1120.0	8.00	1057.41	46.84
С	ЭЛЕКТРОСВ.	10704-76	402.0	5.00	3521.66	29.72
С	ЭЛЕКТРОСВ.	10704-76	245.0	5.00	1929.98	9.90
С	ЭЛЕКТРОСВ.	10704-76	219.0	5.00	125.03	0.57
С	ЭЛЕКТРОСВ.	10704-76	299.0	5.00	4883.02	30.67
С	ЭЛЕКТРОСВ.	10704-76	920.0	7.00	3139.15	101.39
С	ЭЛЕКТРОСВ.	10704-76	273.0	5.00	1141.97	6.54
С	ЭЛЕКТРОСВ.	10704-76	377.0	5.00	4237.74	33.73
С	ЭЛЕКТРОСВ.	10704-76	351.0	5.00	2137.46	15.82

Продолжение таблицы П.1.7

1	2	3	4	5	6	7
С	ЭЛЕКТРОСВ.	10704-76	820.0	7.00	289.28	8.33
С	ЭЛЕКТРОСВ.	10704-76	480.0	5.00	2689.13	28.53
С	ЭЛЕКТРОСВ.	10704-76	630.0	7.00	1167.73	26.97
					26319.56	339.03

Таблица П.1.8

Потребный напор на сети и затраты на сеть

Напор на насосной станции	Капиталь-затраты	Эксплуата-ционные затраты	Приведенные затраты
117.39	386.49	979.73	1366.23

Приложение 3

Распечатка исходных текстов модулей прогограммы задачи Технико-экономический и гидравлический расчета закрытых оросительных сетей

```
#include<conio.h>
#include<stdio.h>
#include<math.h>
/*****************************************************************/
/*                    Головная процедура                         */
/*****************************************************************/
void SORTAM   ();
void OPTIM1   ();
void NASOS    ();
void PECHAT   ();
void PROV     ();
float ALFA, BETA, CTZ1[C51], DELH, DLIN[C51], DVL[C51][21],
      FR1[C51], FR2[C51], FR3[C51], FR4[C51], FR5[C51],
      HAP[C51], HR[C51],HSB[C51], HZAD, NDT, NPT1, PN[C51],
      POTER[C51][CC51], PRIV,RASH[C51], SN, SNS,
      ST[C51][21], TR[C51][21], VIS[C51],VK, VMAX;
int A1[C51], AB[C51][CC51], IR1[C51], IR2[C51], IR3[C51],
    IR4[C51], IR5[C51], IR6[C51], IR7[C51], IT[C51],
    KHZAD, KODM[C51], KOLN[C51], KR, MTR[C51][31],
    MTR1[C51][31],N, N1, N2, ND, NOM, NP, NSOR, RR;
char c,MT[C51][11], GOST[C51][11], TRM[C51][6],
     TIPNS[C51][25],*A,*C, *CH,*RTN1, *G;
FILE   *printer,*RAB11,*RAB12,*RAB51,*RABTR;
main()
{int i,j,I,J,KL,PD,ID,KF,L,LL,II,JJ,N5,NT,
K1,L2,KLL,L1,IN,di;float P;char s;
 C="C";A="A";CH="Ч";RTN1="PTHC";G="Ж";
if((printer=fopen("vvod.","W"))==NULL)
   printf("\nOPEN printer ERROR");
```

```
ND=30;RR=1; VK=1.2;KR=0;
/*ПРИЗ.УТРОЕНИЯ СТОИМОСТИ СТ.ТР*/
DELH=0.5;N1=1;N2=2;HZAD=0;
for(i=1;i<C51;CTZ1[i]=HR[i]=DLIN[i]=VIS[i]=
            RASH[i]=IT[i]=HSB[i]=0,i++);
for(i=1;i<C51;i++)for(j=1;j<=ND;
                    MTR1[i][j]=MTR[i][j]=0,j++);
    if((RABTR = fopen("RABTR", "r")) != NULL)
    {       fscanf(RABTR,"\n%4d %4d %4d ",&N,&NP,&NSOR);
      for(i=1;i <= N;i++)
      {
     for(j=1;j < 31;j++)fscanf(RABTR,"%3d ", &MTR[i][j]);
            }
   for(i=1;i <=N;i++)
   {
  for(j=1;j < 31;j++) fscanf(RABTR,"%3d ", &MTR1[i][j]);
       }
                    for(i=1;i <= N;i++)
                       fscanf(RABTR,"\n %8g %8g %8g %8g",
                       &DLIN[i],&RASH[i],&VIS[i],&HSB[i]);
                    for(i=1;i <= NSOR;i++)
                    {
                      for(j=1;j <= NP;j++)
                        fscanf(RABTR,"%10g ", &TR[i][j]);
                    }
                    for(i=1;i <= NSOR;i++)
                    {
                       for(j=1;j <= NP;j++)
                       fscanf(RABTR,"%10g ", &ST[i][j]);
```

```
                    }
                    for(i=1;i <= NSOR;i++)
                    {
                      fscanf(RABTR,"\n%15s %15s %15s %10d",
                      &GOST[i],&MT[i],&TRM[i],&KODM[i]);
                      IR1[i]=KODM[i];
                    }
                    fprintf(RABTR,"\n");
                    close(RABTR);
                }
                else fprintf(printer,"\nERROR FILE
                                        OPEN RABTR\n");
        for(I=1;I<=N;I++)
        fprintf(printer,"\n |%3d -%3d|%8.2f|%8.4f|  %6.2f |",
         MTR1[I][1], MTR1[I][2], DLIN[I],  RASH[I],VIS[I]);
 if(( RAB11 = fopen("RAB11.", "r")) == NULL) fprintf(
 printer,"\nerror fopen RAB11\n");
  fscanf(RAB11, "%d %g %g %g", &KHZAD, &HZAD, &SN,&VMAX);
          fclose(RAB11);
  fprintf(printer,"\n\n\n\n");
  fprintf(printer,"\n%d %g %g %g", KHZAD,HZAD,SN,VMAX);
  I=0; for(i=1;i<=NSOR;i++) { IR1[NSOR+1]=IR1[i];
     if (IR1[i]>0) { I=I+1; for (j=i+1;j<=NSOR;j++)
      if(IR1[NSOR+1]==IR1[j]) IR1[j]=0;}
  }
 for(i=2;i<=I;i++) {
      for(j=2;j<=NSOR;j++) if (IR1[j]>0) { IR1[i]=IR1[j];
          IR1[j]=0;break;}
 }
```

```
for(i=1;i<=I;i++) printf("\n %d ",IR1[i]);
  if(KHZAD==2)
   { puts("\nУ Вас задан режим выбора насосной станции."
          "\nВыполнили ли Вы 9-ый пункт меню -"
              ""Проектирование ЗОС ОНС""  ?"
          "\n  1) Выпонили (Y);"
          "\n  2) Не выполнили (N).\0");
     c=getch();
     if( c=='2' || c=='N' || c=='n' || c=='H' || c=='т' ||
          c=='T' || c=='н')
       {
         puts("\nВыполните пункт 9 основного ""меню"
                " и затем выполните"
              "\nгидравлический расчет (пункт 3)\0");
         puts("\nПрочитав, нажмите любую клавишу.\0");
         getch(); goto MGL21;   }
    }
if(( RAB12 = fopen("RAB12.", "r")) == NULL)
       fprintf(printer,"\nerror fopen RAB12\n");
  fscanf(RAB12, "%g %g", &ALFA, &BETA);
          fclose(RAB12);if( KHZAD==2 ) BETA=0;RR=1;
for(i=1;i<=NSOR;i++)printf("\nTR[%2d][NP]=%5f  %d
%s\n",i,TR[i][NP],KODM[i],TRM[i]);
for (i=2;i<=N;i++) { MTR1[i][3]=MTR1[i][4]=0; }
  puts("\nВЫ МОЖЕТЕ ЗАДАТЬ НА ОТД.(ИЛИ НА ВСЕХ) УЧАСТКАХ
ШИФР СОРТАМЕНТА""\nЛИБО ВН.ДИАМЕТР И ШИФР ( ИЗ РАНЕЕ
ЗАДАННОГО СОРТАМЕНТА)."
       "\n (1-ДА / 0-НЕТ)");
G005: scanf("%d",&s);
```

```
            if ((s!=0) && (s!=1)) {
               printf("\nВЫ НЕ ПРАВИЛЬНО ЗАДАЛИ."
                              "\nЗАДАЙТЕ ЛИБО 0,ЛИБО 1.\n");
                                 goto G005; }
     if (s==1) {
       RAB51=fopen("RAB51.","w");
      for (i=2;i<=N;i++) {
        printf("\n РАНЕЕ ЗАДАННЫЕ ШИФРЫ :");
       for(j=1;IR1[j]>0;j++) printf("     %4d",IR1[j]);
         printf("\nНА УЧАСТКЕ %3d - %3d ",
                 MTR1[i][2],MTR1[i][1]);
puts("\nБУДЕТЕ ЗАДАВАТЬ ШИФР СОРТАМЕНТА НА ЭТОМ УЧАСТКЕ ?
(1-ДА /0-НЕТ)");
       G003:  scanf("%d",&s);  /*printf("\ns=%d",s);*/
            if ((s!=0) && (s!=1)) {
                        printf("\nВЫ НЕ ПРАВИЛЬНО ЗАДАЛИ."

                        "\nЗАДАЙТЕ ЛИБО 0,ЛИБО 1.\n");
                                   goto G003; }
    if (s==1) {
            puts("\nЗАДАЙТЕ ШИФР НА ЭТОМ УЧАСТКЕ.");
       G000: scanf("%d",&MTR1[i][4]);
                 printf("\nШИФР=%d",MTR1[i][4]);
            for (j=1;j<=NSOR;j++)
               if (KODM[j]==MTR1[i][4]) break;
            if (j > NSOR) {
            puts("\nЗАДАННЫЙ ШИФР ОТСУТСТВУЕТ В
                     ИСХ.СОРТАМЕНТЕ."
              "\nЗАДАЙТЕ ШИФР ПРАВИЛЬНО."); goto G000; }
```

```
      puts("\nБУДЕТЕ ЗАДАВАТЬ ДИМЕТР ?(1-ДА / 0-НЕТ)");
G004:   scanf("%d",&s);  printf("\ns=%d",s);
        if ((s!=0) && (s!=1))
          { printf("\nВЫ НЕ ПРАВИЛЬНО ЗАДАЛИ.",
                  "\nЗАДАЙТЕ ЛИБО 0,ЛИБО 1.\n");
                                        goto G004; }
        if (s==1) {
         puts("\nЗАДАЙТЕ ДИАМЕТР НА ЭТОМ УЧАСТКЕ.");
       G001:scanf("%d",&MTR1[i][3]);
                 printf("\nДИАМЕТР=%d",MTR1[i][3]);
            if (MTR1[i][3] >0 )  {
                for (j=1;j<=NSOR;j++)
                {  di = 1000*TR[j][NP];
                if ( 1000*TR[j][NP]-di>=0.5) di=di+1;
                 if ((KODM[j]==MTR1[i][4])
                        &&(abs(MTR1[i][3]-di)<1))
                 break;    }
                if (j > NSOR) {
         puts("\nЗАДАННЫЙ ВН.ДИАМЕТР ДАННОГО ШИФРА
               ОТСУТСТВУЕТ"
                 "\nВ РАНЕЕ ЗАДАННОМ СОРТАМЕНТЕ.");
        puts("\n ПОСМОТРИТЕ ДИАМЕТРЫ ТРУБ НСИ ЗАДАННЫХ
ШИФРОВ(нажмите ввод).");getch();
                 puts("\n--------------------"
                      "\n| ШИФР | ВН.ДИАМЕТР|"
                      "\n--------------------");
            for(j=1;j<=NSOR;j++) printf("\n %4d %5.0f",
              KODM[j],TR[j][NP]*1000);
              puts("\nЗАДАЙТЕ ПРАВИЛЬНО. ");goto G001;
```

```
                    }
                }
            }
        }
    fprintf (RAB51,"\n%3d - %3d   %4d  %4d",
      MTR1[i][2], MTR1[i][1], MTR1[i][3], MTR1[i][4]);
        }
}  SORTAM();fprintf(printer,"\nN=%6d  ND=%6d",N,ND);
if( RR==0 )  goto  MGL20;RR=1;
fputs("\nПеред входом в PROV\0",printer);
fprintf(printer,"\nHZAD=%f",HZAD);
 PROV();          /*ОБРАЩЕНИЕ К П/П */
fprintf(printer,"\nHZAD=%f",HZAD);
fprintf(printer,"\nN=%6d  ND=%6d",N,ND);
fputs("\nПосле выхода из PROV\0",printer);
fprintf(printer,"\nRR=%d",RR);
if( RR==0 )  goto  MGL20;
 if( KHZAD==2 )   NASOS() ; KR=0;
 for( I = 1 ; I <=  NSOR; I += 1) {
    if(  strcmp(TRM[I],C)!=0 ) KR=1;
 }KF=3; printf("\nKR=%d ",KR);
 /* KF- КОЭФФИЦИЕНТ УТРОЕНИЯ СТОМОСТИ СТАЛИ*/
 if( KR==1 )  for( I = 1 ; I <=  NSOR; I += 1) {
    if(  strcmp(TRM[I],C)==0 )
    for( J = 1 ;( J <=  NP-1)  && (TR[I][J]>0); J += 1) {
     ST[I][J]=ST[I][J]*KF;          }
fputs("\nMTR\0 ",printer);
for(i=1;i<=N;i++)
{ fprintf(printer,"\n");
```

```
  for(j=1;j<=ND ;j++)
  {     fprintf(printer,"%2d ",MTR[i][j]);  }
}
fputs("\nПеред входом в OPTIM1\O",printer);
fprintf(printer,"\nHZAD=%f",HZAD);
 OPTIM1();      /*      ОБРАЩЕНИЕ К П/П      */
fprintf(printer,"\nHZAD=%f",HZAD);
 if( RR==0 )  goto  MGL20;
  if( KR==1 )   for( I = 1 ; I <=  NSOR; I += 1) {
     if( strcmp(TRM[I],C)==0 )
     for( J = 1 ;( J <=  NP-1)  && (TR[I][J]>0); J += 1) {
          ST[I][J]=ST[I][J]/KF;
        }
    }
 PECHAT();      /*      ОБРАШЕНИЕ К П/П */
 printf("\nПРОГРАММА ГИДРАВ.РАСЧЕТА ЗАВЕРШИЛОСЬ НОРМАЛЬНО.
          Нажмите ввод."); getch();
  if( KHZAD==2 )  { IN=RR;fprintf(printer,"\n\n");
fprintf(printer,
 "\nДЛЯ ДЕТАЛЬНОГО ПРОЕКТИРОВАНИЯ РЕКОМЕНДУЕТСЯ НАСОСНАЯ
СТАНЦИЯ- %s ", TIPNS[IN]); } goto  MGL21;
 MGL20: puts("\nПРОГРАММА ЗАВЕРШИЛОСЬ НЕ НОРМАЛЬНО."
          "ПОСМОТРИТЕ ФАИЛ VVOD. . Нажмите ввод.");
       getch();
MGL21:  fprintf(printer,
 "\n\n\nАЛГОРИТМ И ПРОГРАММА РАЗРАБОТАНЫ В НИИ ПММ
КАБАРДИНО-БАЛКАРСК"
"ОГО ГОСУНИВЕРСИТЕТА В 1979 Г\n\n\n");
 puts("\n\nПРОГРАММА НА ПЭВМ РАЗРАБОТАНА В ОНИЛ
```

```
      КБГУ(г.НАЛЬЧИК) В 1990 г.\n");   /* END VVOD;*/
#include<stdio.h>
/*****************************************************************/
/*          Подпрограмма оптимизации                             */
/*        диаметров труб и напора на н.с.                        */
/*****************************************************************/
void OPTIM1(void)
{
extern float ALFA, BETA, CTZ1[C51], DELH, DLIN[C51],
      DVL[C51][21], FR1[C51],
      FR2[C51], FR3[C51], FR4[C51], FR5[C51], HAP[C51],
      HR[C51],HSB[C51], HZAD, NDT, NPT1, PN[C51],
      POTER[C51][CC51], PRIV,RASH[C51], SN, SNS,
      ST[C51][21], TR[C51][21], VIS[C51],
      VK, VMAX;
extern int A1[C51], AB[C51][CC51], IR1[C51], IR2[C51],
        IR3[C51],IR4[C51], IR5[C51], IR6[C51], IR7[C51],
        IT[C51],KHZAD, KODM[C51], KOLN[C51], KR,
        MTR[C51][31], MTR1[C51][31],
        N, N1, N2, ND, NOM, NP, NSOR, RR;
extern char MT[C51][11], GOST[C51][11], TRM[C51][6],
            TIPNS[C51][25],
            A[6], C[6], CH[6],RTN1[6], G[6];
extern FILE   *printer;
FILE *RABNS;
float PRIVV,HAPOR,VD,EFK1,PMM;
int MMT[C51][C51],MA1[C51],MA2[C51],KDELM,i,j,I,K,P3,P4,
    J1,J,II,JJ,REG,LK,J4,I2,J3,I3,J2,I1,I22,K1,K2,KK;
float  EFK[C51],PNN[C51],SP,PN2,EFK2,A2,PM,RES,PPN,HAPM,
```

```
        SOS,P1,P,X,P2,
        XL,STM,PRIV1,PC;
  fputs("\n OPTIM1\0",printer);
  printf("\nNSOR=%d  ND=%d",NSOR,ND);ND=30;
  printf("\nNSOR=%d  ND=%d",NSOR,ND);
  fputs("\nPOTER\0",printer);
  for(i=1;i<=N;i++){ fprintf(printer,"\n");
    for(j=1;(j<CC51) &&(POTER[i][j]>0);j++)
    { fprintf(printer,"%8f ",POTER[i][j]);  }
  }fputs("\nAB\0",printer);for(i=1;i<=N;i++)
  { fprintf(printer,"\n"); for(j=1;(j<CC51)
  &&(AB[i][j]>0);j++)
    {     fprintf(printer,"%2d ",AB[i][j]);  }
  }fputs("\nMTR\0",printer);for(i=1;i<=N;i++)
  { fprintf(printer,"\n");  for(j=1;(j<=ND) ;j++)
    {     fprintf(printer,"%2d ",MTR[i][j]);  }
  }PC=0; /*ХАРАКТЕРИСТИКИ  СЕТИ */
  for(i=1;i<C51;i++){ HAP[i]=A1[i]=0;if (HSB[i]>0) {
  if ( PC < ( VIS[i]-VIS[1]+HSB[i]))PC=VIS[i]-VIS[1]+HSB[i];}
    for(j=1;j<C51;j++)    MMT[i][j]=0;
  } PRIV=1e10;SNS=0;
    for( I = N2 ; I <=  N; I += 1) { K=0; J1=MTR[I][ND];
    for( J = 1 ;AB[I][J]!=J1; J +=  1 );PN[I]=POTER[I][J];
          for( II = N2 ; II <=  N; II += 1) {
            for( JJ = 2 ; JJ <=  ND-1; JJ += 1) {
              if( I==MTR[II][JJ] )  {K=K+1;  MMT[I][K]=II;}
            }
          }
  fprintf(printer,"\n");
```

```
for(K=1;MMT[I][K]>0;K++)fprintf(printer,"%d ",MMT[I][K]);}
 if( KHZAD==0 ) KDELM=100;
     if( KHZAD==1 )  {
       KDELM=1; if( NDT<=HZAD ) NDT=HZAD;
        else   {
         printf("\nЗАДАННЫЙ НАПОР МЕНЬШЕ ЧЕМ ДОПУСТИМЫЙ
         МИНИМАЛЬНЫЙ ПОТРЕБНЫЙ НА""ПОР НА СЕТИ");
         printf("\nМИНИМАЛЬНЫЙ ПОТРЕБ. НАПОР=%10f,A
           ЗАДАННЫЙ"" НАПОР РАВЕН %10f",NDT,HZAD);
         RR=0;  goto  MMML; }
      }
    HAPOR=NDT;REG=0;if( KHZAD==2 )  { REG=1;BETA=0;
         if((RABNS=fopen("RABNSS","w"))!=NULL) { ;}
         for( I = 1 ;HR[I]> 0; I +=  1 );  KDELM=I-1; }
      fprintf(printer,"\nKDELM=%d",KDELM);
 M1011:for( II = 1 ; II <=  KDELM; II += 1) {
 if( REG==1 )  {if( NDT>HR[II] ) goto M3666; NDT=HR[II];
         LK=KOLN[II]; SNS=CTZ1[LK]; goto  M1012; }
       NDT=HAPOR+DELH*(II-1);
 M1012:  for( I = N2 ; I <=  N; I += 1) { SP=0;
         MMT[I][N+1]=1;
          for( J = N1 ; J <=  ND-1; J += 1) {
          K=MTR[I][J]; if( K==1 )  goto  M3555;
           SP=SP+PN[K];  }
 M3555: HAP[I]=NDT-SP+VIS[1]-VIS[I];  }
       for( I = N1 ; I <=  N; I += 1) {
       EFK[I]=-1;K=0;A1[I]=J1=MTR[I][ND];J4=IT[I];
       for( K = 1 ; (K < CC51)  && (AB[I][K]!=J1);
         K += 1);PN[I]=POTER[I][K];I2=MTR[I][2];
```

```
        for( J = 1 ; J <=  K-1; J += 1) {
                J3=AB[I][J]; PN2=POTER[I][J]; EFK2=0;
                for( JJ = 1 ; (JJ <=  NP-1)  &&
                (TR[J3][JJ]>0); JJ += 1) {
                VD=4*RASH[I]/(3.14*TR[J3][JJ]*TR[J3][JJ]);
                if( (DVL[J3][JJ]>=HAP[I2]) && (VD<VMAX))
                {
                        A2=ST[J1][1]-ST[J3][1];
                        PM=PN2-PN[I];
                        if( PM>0 ) EFK2=DLIN[I]*A2/PM;
                        goto  S_M000;
                }
                }
S_M000:         if( EFK[I]<=EFK2 )
                { EFK[I]=EFK2;    A1[I]=AB[I][J];   }
        }
}
for( I = N2 ; I <=  N; I += 1) { SP=0;
        for( J = N1 ; J <=  ND-1; J += 1) {
         K=MTR[I][J];if( K==1 )  goto  M3555m;
         SP=SP+PN[K];   }
M3555m: HAP[I]=NDT-SP+VIS[1]-VIS[I];
  } fprintf(printer,"\n%7f %7f PN[%d]=%5f",
HAP[I],HSB[I],I,PN[I] );
M1020:EFK1=EFK[2];I2=2;
      for( I = N2+1 ; I <=  N; I += 1) {
        if( EFK1<=EFK[I] ) { EFK1=EFK[I];I2=I; }
      }
      J2=A1[I2];RES=HAP[I2]-HSB[I2];PPN=PN[I2];I3=I2;
```

```
        for( J = 1 ;MMT[I2][J]>0; J += 1 ) {
          I1=MMT[I2][J]; HAPM=HAP[I1]-HSB[I1];
          if( RES>HAPM )  {  RES=HAPM; I3=I1;  }
         }
        if ( RES < 0 )  RES=0;
        for( K = 1 ; (K <  CC51)  && (AB[I2][K]!=J2); K += 1);
        PM=POTER[I2][K]-PN[I2];
 if( RES>PM ) {
  J4=IT[I2]; EFK[I2]=-1;PN[I2]=POTER[I2][K];I22=MTR[I2][2];
  M3000: for( J = 1 ; J <=  K-1; J += 1) {
          J3=AB[I2][J]; PN2=POTER[I2][J];EFK2=0;
          for( JJ = 1 ; (JJ <=  NP-1)  && (TR[J3][JJ]>0);
          JJ += 1)
          { VD=4*RASH[I]/(3.14*TR[J3][JJ]*TR[J3][JJ]);
           if( (DVL[J3][JJ]>=HAP[I22]) && (VD<=VMAX ))  {
            A2=ST[J2][1]-ST[J3][ 1];PMM=PN2-PN[I2];
          if( PMM > 0 ) EFK2=DLIN[I2]*A2/PMM; gotoS_M001;
           }
         }
  S_M001:  if( EFK[I2]<=EFK2 ) { EFK[I2]=EFK2;
            A1[I2]=AB[I2][J];     }
    if( EFK[I2]> 0 )  MTR[I2][ND]=J2;  goto  M1018;
  }
 if( RES <= PM ) PN[I2]=PN[I2]+RES;
  M3100: for( J = 1 ; (J <=  ND-1)  && (MTR[I3][J]>0);
          J += 1) {  J3=MTR[I3][J];EFK[J3]=-1;  }
  M1018: P2=PN[I2]-PPN; HAP[I2]=HAP[I2]-P2;
  for( I = N1 ; (I <=  N)  && (MMT[I2][I]>0);I += 1) {
 K=MMT[I2][I]; HAP[K]=HAP[K]-P2; }
```

```
        for( I = N2 ; I <=  N; I += 1) {
         if( EFK[I]> -1 )  goto M1020; }
        SOS=0;
        for( J = N2 ; J <=  N; J += 1) {
          J1=A1[J]; J2=MTR[J][ND];I2=MTR[J][2];
          for( JJ = 1 ; (JJ < CC51)  && (AB[J][JJ]>0);
             JJ += 1) { if( J1==AB[J][JJ] ) K1=JJ;
            if( J2==AB[J][JJ] ) K2=JJ;  }
          P1=POTER[J][K1];P=POTER[J][K2];
          if( J2==J1 )  { X=0; goto M4000;  }
          X=(P1-PN[J])*DLIN[J]/(P1-P);
          fprintf(printer,"\n L=%8.2f X=%8.3f J2=%3d
                    P=%8.3f h=%8.3f J1=%3d P1=%8.3f",
            DLIN[J],X,J2,P,PN[J],J1,P1 );
M4000:     P3=P4=1;
          for (i=1;(i<=NP) && ( HAP[I2]<=DVL[J2][i]);i+=1)
                { IT[J]=i;break;}
          for (i=1;(i<=NP) && ( HAP[I2]<=DVL[J1][i]);i+=1)
                { MMT[J][N+1]=i;break;}
          XL=DLIN[J]-X;
          fprintf(printer," XL=%8.3f P3=%3d ST=%8.3f P4=%3d
                          ST=%8.3f IT=%3d %3d ",
          XL,P3,ST[J2][P3],P4,ST[J1][P4],IT[J],MMT[J][N+1]);
          STM=ST[J2][P3]*X+ST[J1][P4]*XL;   SOS=SOS+STM;
      } SOS=SOS/1000;PRIV1=ALFA*SOS+BETA*RASH[1]*NDT+SNS;
 if (KHZAD==2) {fprintf(RABNS,"\n
%-12s%8.2f%8.2f%8.2f%8.2f%8.2f%8.2f%8.2f%8.2f%8.2f",
TIPNS[LK],HR[II],FR1[LK],FR2[LK],FR3[LK], CTZ1[LK],SOS,
ALFA*SOS,PRIV1,(HR[II]-PC)/(RASH[1]*RASH[1]),PC);  }
```

```
fprintf(printer,"\nSOS=%10f  NDT=%10f PRIV1=%10f ALFA=%10f
                  BETA=%10f RASH[1]=%10f SNS=%10f",
                 SOS,NDT,PRIV1,ALFA,BETA,RASH[1],SNS);
        if( PRIV >PRIV1 )
        {  PRIV=PRIV1; NPT1=NDT; KK=II;
           for( I = N2 ; I <=  N; I += 1) {
               MA1[I]=MTR[I][ND];MA2[I]=A1[I];PNN[I]=PN[I];
               IR1[I]=IT[I]*100+MMT[I][N+1];  }
        }
        if( HR[II+1]> 0 )  goto  M3666;
        if( PRIV<PRIV1 )   goto M4555;
 M3666: ; }
NDT=NPT1;if( PRIV==1e10 )  goto  MGL10;
M4555:for( I = N1 ; I <=  N; I += 1) {  SP=0;
       IT[I]=IR1[I];A1[I]=MA2[I]; MTR[I][ND]=MA1[I];
       PN[I]=PNN[I];
       for( J = N1 ; J <=  ND-1; J += 1) {
                K=MTR[I][J];if( K==1 )  goto  MG000;
                SP=SP+PNN[K];
       }
 MG000: HAP[I]=NPT1-SP+VIS[1]-VIS[I];
 }
if (KHZAD==2) {  RR=LK=KOLN[KK]; HR[KK]=0;SOS=0;
               /*РЕКОМЕНДУЕМЫЙ ВАРИАНТ Н.С.*/
             fprintf(RABNS,"\n %-12s%8.2f%8.2f%8.2f%8.2f
                          %8.2f%8.2f%8.2f%8.2f%8.2f%8.2f",
             TIPNS[LK],HR[KK],FR1[LK],FR2[LK],FR3[LK],
             CTZ1[LK],SOS,
      ALFA*SOS,PRIV1,(HR[KK]-PC)/(RASH[1]*RASH[1]),PC); }
```

```
fclose(RABNS);  goto  MMML; MGL10:  RR=0;
 puts("\nB КАТАЛОГЕ ТИПОВЫХ Н.С. НЕТ(НАЖМИТЕ ВВОД.)");
 MMML:          ;
fprintf(printer,"\nSOS=%10f  HAP[1]=%10f  ",SOS,HAP[1]);
printf("\n/Капит.зат.=%10f,Приведенные затраты=%10f
          Напор на н.с.=%7.2f  ",SOS,PRIV,HAP[1]);
fputs("\nEND OPTIM1\0",printer);}
#include<stdio.h>
/*****************************************************************/
/*          Подпрограмма формирования рабочих массивов          */
/*                  по типовым насосным станциям  .             */
/*****************************************************************/
void NASOS(void)
{extern float ALFA, BETA, CTZ1[C51], DELH, DLIN[C51],
      DVL[C51][21], FR1[C51],FR2[C51], FR3[C51], FR4[C51],
      FR5[C51], HAP[C51], HR[C51],HSB[C51], HZAD, NDT,
      NPT1, PN[C51], POTER[C51][CC51], PRIV,
      RASH[C51], SN, SNS, ST[C51][21], TR[C51][21],
      VIS[C51],VK, VMAX;
extern int A1[C51], AB[C51][CC51], IR1[C51], IR2[C51],
      IR3[C51],IR4[C51], IR5[C51], IR6[C51], IR7[C51],
      IT[C51],KHZAD, KODM[C51], KOLN[C51], KR,
      MTR[C51][31], MTR1[C51][31],
extern char MT[C51][11], GOST[C51][11], TRM[C51][6],
    TIPNS[C51][25], A[6], C[6], CH[6],RTN1[6], G[6];
char    P3[13];float P1,P;FILE *RABNS;int I,J,i,j,P2,NN,I1;
for(i=1;i<C51;i++){CTZ1[i]=HR[i]=KOLN[i]=FR4[C51]=0;
strcmp(TIPNS[i][0],"-"); }
if((RABNS=fopen("RABNS","r"))!=NULL) {
```

```
   fscanf(RABNS,"%d",&NN);
   for( I = 1 ; I <=  NN; I += 1) {
      fscanf(RABNS,"%s %g %g %g %g %g ",
                    &TIPNS[I],&HR[I],&FR1[I],
                    &FR2[I],&FR3[I],&CTZ1[I]);
                       KOLN[I]=I;
   printf("\n %s  %g %g  %g  %g %g",TIPNS[I],FR1[I],
                          FR2[I],FR3[I],CTZ1[I],HR[I]);
   }
 }
  for( I = 1 ; I <=  NN; I += 1) { P=HR[I]; I1=I;
      for( J = I+1 ; J <=  NN; J += 1) {
          if( HR[J]==0 ) goto M200;
            if( HR[J]<P )  {   P=HR[J]; I1=J;  }
       }
 M200:  ;
   P1=HR[I1]; HR[I1]=HR[I]; HR[I]=P1;
   P2=KOLN[I1];KOLN[I1]=KOLN[I];KOLN[I]=P2;
 M160:  }   }
#include<stdio.h>
/*****************************************************************/
/*          Подпрограмма печати выходных                         */
/*                    таблиц                                     */
/*****************************************************************/
#include<string.h>
void    PECHAT(void) {
extern float ALFA, BETA, CTZ1[C51], DELH, DLIN[C51],
      DVL[C51][21], FR1[C51],FR2[C51], FR3[C51], FR4[C51],
      FR5[C51], HAP[C51], HR[C51],
```

```
      HSB[C51], HZAD, NDT, NPT1, PN[C51], POTER[C51][CC51],
      PRIV,RASH[C51], SN, SNS, ST[C51][21], TR[C51][21],
      VIS[C51],VK, VMAX;
extern int A1[C51], AB[C51][CC51], IR1[C51], IR2[C51],
      IR3[C51],IR4[C51], IR5[C51], IR6[C51], IR7[C51],
      IT[C51],KHZAD, KODM[C51], KOLN[C51], KR,
      MTR[C51][31], MTR1[C51][31],N, N1, N2, ND, NOM, NP,
      NSOR, RR;
extern char MT[C51][11], GOST[C51][11], TRM[C51][6], A[6],
      C[6], CH[6],RTN1[6], G[6];extern FILE *printer;
char MAT[6]="-\0";FILE *RAB61;
int IN,IK,KDOP,I,J1,J2,J,K1,K2,K,KOD,I1;
float D,DR,SOS,TCD,PN1,HS,STT,DL,R,V,P1,P2,PP,X1,Z1,H1,PKZ,PQH,T;
if((RAB61=fopen("RAB61.","w"))==NULL)
       fputs("\nОШИБКА ОТКРЫТИЯ RAB61",printer);
fprintf(printer,"\nПРИМЕЧАНИЕ:
           ВОЗМОЖНО В ТАБЛИЦЕ БУДУТ УЗЛЫ"
          " С НОМЕРАМИ 200 И ВЫШЕ,\nТОЧКИ РАЗБИЕНИЯ УЧАСТКА
           С РАЗЛИЧНЫМИ ДИАМЕТРАМИ");
    SOS=0;IN=MTR1[1][2];IK=MTR1[1][1];HAP[IK]=NPT1;
KDOP=200;DL=0;R=0;V=0;DR=0;TCD=0;PN1=0;HS=0;STT=0;KOD=0;
fprintf(RAB61,"\n%3d %3d %10g %10g %10g %10g %5g %10g %5d
                %5s %10g %10g %10g %10g  ",
   IN,IK,DL,R,VIS[IK],V,DR,TCD,KOD,MAT,PN1,HS,HAP[IK],STT);
fprintf(printer,"\n¦ %4d-%4d ¦ ¦ ¦%7.2f ¦ ¦ ¦ ¦ ¦ ¦
                  ¦%7.2f ¦        ¦",
 IN,IK,VIS[IK],HAP[IK]);
 C0001:for( I = 2 ; I <=  N; I += 1) {
        J1=MTR[I][ND]; printf("\nIT=%d",IT[I]);
```

```
        J2=A1[I];
          for( J = 1 ; (J < CC51)  && (AB[I][J]>0); J += 1)
           {if( J1==AB[I][J] ) K1=J;
             if( J2==AB[I][J] ) K2=J;   }
             P1=POTER[I][K1];P2=POTER[I][K2];
             if( J1!=J2 ) PP=(P2-PN[I])/(P2-P1);else  PP=1;
             X1=PP*DLIN[I];
             if( (X1==DLIN[I]) || (X1==0) ) DG001: {
             IN=MTR1[I][2];IK=MTR1[I][1];
             if( X1==DLIN[I] ) K=J1;else  K=J2;
             DR=TR[K][NP]*1000;KOD=KODM[K]; J=IT[I]/100;
             printf("\nIT=%d J=%d DR=%5g 1000*TR[K][NP]=%g"
             ,IT[I],J,DR, 1000*TR[K][NP]);
             TCD=(DR-TR[K][J]*1000)/2;strcpy(MAT,TRM[K]);
             V=4*RASH[I]/(3.14*TR[K][J]*TR[K][J]);
             STT=ST[K][J]*DLIN[I]/1000;SOS=SOS+STT;
             RASH[I]=RASH[I]*1000;
    fprintf(RAB61,"\n%3d %3d %10g %10g %10g %10g %5g %10g
                    %5d %5s %10g %10g %10g %10g  ",
    IN,IK,DLIN[I],RASH[I],VIS[IK],V,DR,TCD,KOD,MAT,PN[I],
    HSB[IK],HAP[IK],STT);
 fprintf(printer,"\n| %4d-%4d |%8.2f |%7.2f |%7.2f |%5g
|%5.1f |%5d |%4s   |%6.2f |%8.3f |%7.2f |%7.2f |",
IN,IK,DLIN[I],RASH[I],VIS[IK],  DR,  TCD,  KOD, MAT, V,
PN[I], HAP[IK],STT); goto M0002;  }
             if( DLIN[I]-X1>0 && X1>0 ) DG002: {
             IN=MTR1[I][2];IK=KDOP;J=IT[I]/100;
             K=J1;PN1=PP*P1;
             DR=TR[K][NP]*1000;TCD=(DR-TR[K][J]*1000)/2;
```

```
            strcpy(MAT,TRM[K]);
            KOD=KODM[K];V=4*RASH[I]/(3.14*TR[K][J]*TR[K][J]);
            STT=ST[K][J]*X1/1000;I1=MTR[I][1];SOS=SOS+STT;
            Z1=(VIS[I1]-VIS[IN])*X1/DLIN[I]+VIS[IN];
            HS=0;H1=HAP[IN]-PN1-(Z1-VIS[IN]);
            RASH[I]=RASH[I]*1000;
    fprintf(RAB61,"\n%3d %3d %10g %10g %10g %10g %5g %10g
                    %5d %5s %10g %10g %10g %10g  ",
    IN,IK,X1,RASH[I],Z1,V,DR,TCD,KOD,MAT,PN1,HS,H1,STT);
 fprintf(printer,
"\n| %4d-%4d |%8.2f |%7.2f |%7.2f |%5g |%5.1f |%5d |%4s
            |%6.2f |%8.3f |%7.2f |%7.2f |",
IN,IK,X1,RASH[I],Z1,DR,TCD,KOD,MAT,V,PN1,H1,STT);
            K=J2;J=IT[I]/100;J=IT[I]-J*100;
            IN=IK;IK=MTR1[I][1];X1=DLIN[I]-X1;PN1=(1-PP)*P2;
            DR=TR[K][NP]*1000;TCD=(DR-TR[K][J]*1000)/2;
            strcpy(MAT,TRM[K]); KOD=KODM[K];
            V=4*RASH[I]/(3.14*TR[K][J]*TR[K][J]*1000);
            STT=ST[K][J]*X1/1000;SOS=SOS+STT;
    fprintf(RAB61,"\n%3d %3d %10g %10g %10g %10g %5g %10g
                    %5d %5s %10g %10g %10g %10g  ",
    IN,IK,X1,RASH[I],VIS[IK],V,DR,TCD,KOD,MAT,PN1,HSB[IK],
    HAP[IK],STT); fprintf(printer,
"\n| %4d-%4d |%8.2f |%7.2f |%7.2f |%5g |%5.1f |%5d |%4s
|%6.2f |%8.3f |%7.2f |%7.2f |",
IN,IK,X1,RASH[I],VIS[IK],DR,TCD,KOD,MAT,V,PN1,HAP[IK],STT);
            KDOP=KDOP+1;   }
 M0002:;}     fprintf(printer,
"\n-----------------------------------------------------"
```

```
"-----------------------------\n");
 PKZ=ALFA*SOS;/*КАП. ЗАТРАТЫ  */
 PQH=BETA*RASH[1]*NPT1;
if( KHZAD==2 ) PQH=CTZ1[RR];/*  ЭКСП.ЗАТРАТЫ*/PRIV=PKZ+PQH;
      fputs("\n-----------------------------------------"
      "\n! НАПОР ! КАПИТ. ! ЭКСПЛУАТ. ! ПРИВЕДЕН. !"
      "\n!НА Н.С.!ЗАТРАТЫ ! ЗАТРАТЫ   !  ЗАТРАТЫ  !"
      "\n! (М)   ! (Т.РУБ)! (Т.РУБ.)  ! (Т.РУБ.)  !"
      "\n-----------------------------------------\0",
      printer);
 fprintf(printer,"\n¦%7.2f¦%8.3f¦ %9.3f ¦ %9.3f ¦",NPT1,
       PKZ,PQH,PRIV);
          fputs("\n-------------------------------"
                "-----------",printer);
printf("\n/Капит.зат.=%10f,Приведенные затраты=%10f
 Напор на н.с.=%7.2f  ",PKZ,PRIV,NPT1);
        IN=0;IK=0;DL=0;R=0;Z1=0;V=0; D=0;T=0;KOD=0;PN1=0;
strcpy(MAT,"-\0");
       fprintf(RAB61,"\n%3d %3d %10g %10g %10g %10g %5g
                  %10g %5d %5s  %10g %10g %10g %10g ",
    IN,IK,DL,R,Z1,V,D,T,KOD,MAT,NPT1,PKZ,PQH,PRIV);
      fclose(RAB61);    }
#include<math.h>
#include<string.h>
/*************************************************************/
/*          Подпрограмма формирования первоначального      */
/*              решения по условиям давления               */
/*************************************************************/
void   PROV(void) {
```

```
extern float ALFA, BETA, CTZ1[C51], DELH, DLIN[C51],
   DVL[C51][21], FR1[C51],FR2[C51], FR3[C51], FR4[C51],
   FR5[C51], HAP[C51], HR[C51],HSB[C51], HZAD, NDT,
   NPT1, PN[C51], POTER[C51][CC51], PRIV,
   RASH[C51], SN, SNS, ST[C51][21], TR[C51][21],
   VIS[C51],VK, VMAX;
extern int A1[C51], AB[C51][CC51], IR1[C51], IR2[C51],
   IR3[C51],IR4[C51], IR5[C51], IR6[C51], IR7[C51],IT[C51],
   KHZAD, KODM[C51], KOLN[C51], KR, MTR[C51][31],
   MTR1[C51][31],N, N1, N2, ND, NOM, NP, NSOR, RR;
extern char MT[C51][11], GOST[C51][11], TRM[C51][6], A[6],
   C[6], CH[6], RTN1[6], G[6];extern FILE *printer,*RAB11;
float PN2,POTN[C51],HMAX,SP;float P5,P6,P7;
int I,K,J,J1,I4,KLL,KL,MNOM,i,j;
 fprintf(printer,"\nRR=%d",RR);
fprintf(printer,"\nHZAD=%f",HZAD);
printf("\nNSOR=%d  ND=%d",NSOR,ND);ND=30;
for(i=1;i<C51;i++)for(j=1;j< 21;DVL[i][j]=0,j++);
for(i=1;i<C51;PN[i]=HAP[i]=POTN[i]=0,i++);
 C001:for( I = N2 ; I <=  N; I += 1) { K=0;
       for( J = 1 ;POTER[I][J] >0; J +=  1 ) ;
       K=J-1; PN[I]=POTER[I][K]; MTR[I][ND]=AB[I][K]; }
       for( I = 1 ; I <=  NSOR; I += 1) {
          DVL[I][1]=150; if( TRM[I][0]=='C' )  {
          if( TR[I][NP]<=1 ) P7=1;
          if( TR[I][NP]>1 ) if( TR[I][NP]<=1.2 ) P7=1.05;
          if( TR[I][NP]>1.2 ) P7=1.1;
          for( J = 1 ; (J <=  NP-1)  && (TR[I][J]>0);J+=1)
          {
```

```
            P5=TR[I][NP]-TR[I][J]; P5=100*P5; P5=P5/2;
            P6=P7*(100*TR[I][NP]-2*P5);
            DVL[I][J]=1558.442*P5/P6;
            DVL[I][J]=10*DVL[I][J];
           }    }
        if( strcmp(MT[I],"BT-6")==0 ) DVL[I][1]=60;
        if( strcmp(MT[I],"BT-9")==0 ) DVL[I][1]=90;
        if( strcmp(MT[I],"BT-12")==0 ) DVL[I][1]=120;
        if( strcmp(MT[I],"BT-15")==0 ) DVL[I][1]=150;
        if( strcmp(TRM[I],"Ч")==0 ) DVL[I][1]=160;
        if( strcmp(TRM[I],"Ж")==0 ) DVL[I][1]=150;
        printf("\nDVL[%2d][1]=%f",I,DVL[I][1]);
         } HMAX=0;MNOM=0;
  C002:for( I = N1 ; I <=  N; I += 1) {  SP=0;
           for( J = 1 ; J <=  ND-1; J += 1) {
              K=MTR[I][J]; if( K==1 )  goto  MM444;
              SP=SP+PN[K];  }
 MM444:POTN[I]=SP-VIS[1]+VIS[I]+HSB[I];
        if( HSB[I]>0 && HMAX-POTN[I]<=0 )
         {  HMAX=POTN[I];MNOM=I;    }
}
       for( I = N2 ; I <=  N; I += 1) {
         HAP[I]=HMAX-POTN[I]+HSB[I]; }
for(i=1;i<C51;IT[i]=1,i++);
 C003: for( I = N2 ; I <=  N; I += 1) {J1=MTR[I][ND];
           for( K = 1 ; (K < CC51)  && (AB[I][K]!=J1);K+=1);
           I4=MTR[I][2]; KLL=1;
           while (KLL>0)
           {   KL=0;
```

```
            for( J = 1 ; (J <=NP)  && (DVL[J1][J]>0);J+=1)
              {   if( HAP[I4]<DVL[J1][J] )
                  {    IT[I]=J; KL=1; KLL=0; goto  M000; }
             }
           if( KL==0 )  {
             K=K-1; if( K>0 )  {
              J1=MTR[I][ND]=AB[I][K]; PN[I]=POTER[I][K]; }
                fprintf(printer,"\nK=%d",K);
                if( K==0 )
{
printf("\n HAP[%d]=%g DVL[%d][%d]=%g",I4,HAP[I4],J1,J,
DVL[J1][J-1]);
puts("\nЗАДАЧА НЕ МОЖЕТ БЫТЬ РЕШЕНА НА ДАННОМ СОРТАМЕНТЕ ПО"
       " УСЛОВИЯМ ДАВЛЕНИЯ\0");
puts("\nПОМЕНЯЙТЕ СОРТАМЕНТ И ЗПУСТИТЕ ТРАССИРОВКУ ЗАНОВО.");
RR=0; getch();
pintf(printer,"\nRR=%d",RR);goto ENDPROV; }
   goto  C002; }  M000:  ;          } }
       for( I = N2 ; I <=  N; I += 1) {   SP=0;
           for( J = N1 ; J <=  ND-1; J += 1) {
              K=MTR[I][J];if( K==1 )  goto  MM400;
              SP=SP+PN[K];              }
 MM400:POTN[I]=HSB[I]+VIS[I]-VIS[1]+SP;          }
       for( I = N2 ; I <=  N; I += 1) {
         if( HSB[I]!=0 ) if( HMAX-POTN[I]<=0 )
            { HMAX=POTN[I]; MNOM=I; }
        }        NDT=HMAX;
printf("\nМИНИМАЛЬНЫЙ ПОТРЕБНЫЙ НАПОР=%f
ДИКТ.ВЕРШИНА-%d",HMAX,MNOM);ENDPROV:;}
```

```
#include<stdio.h>
#include<math.h>
/*****************************************************************/
/*          Подпрограмма формирования общего сортамента          */
/*            труб и сортамента по каждому участку               */
/*****************************************************************/
void SORTAM(void){
extern float ALFA, BETA, CTZ1[C51], DELH, DLIN[C51],
   DVL[C51][21], FR1[C51],FR2[C51], FR3[C51],
   FR4[C51], FR5[C51], HAP[C51], HR[C51],
   HSB[C51], HZAD, NDT, NPT1, PN[C51], POTER[C51][CC51],
   PRIV,RASH[C51], SN, SNS, ST[C51][21], TR[C51][21],
   VIS[C51],VK, VMAX;
extern int A1[C51], AB[C51][CC51], IR1[C51], IR2[C51],
   IR3[C51],IR4[C51], IR5[C51], IR6[C51], IR7[C51],
   IT[C51],KHZAD, KODM[C51], KOLN[C51], KR,MTR[C51][31],
MTR1[C51][31],N, N1, N2, ND, NOM, NP, NSOR, RR;
extern char MT[C51][11], GOST[C51][11], TRM[C51][6],
            A[6], C[6], CH[6],RTN1[6], G[6];
extern FILE *printer;FILE *RABNS;
float VD,P,P1,P2,P3,PN2,d;
int KKK,K,I,IJ,II,I1,J,i,j,di;ND=30;
fprintf(printer,"\nNSOR=%d  ND=%d",NSOR,ND);
for(i=1;i<=NSOR;i++)
 fprintf(printer,"\nTR[%2d][1]=%10f",i,TR[i][1]);
for(i=1;i<C51;i++)for(j=1;j<CC51;j++)
      POTER[i][j]=AB[i][j]=0;
 S_MKKKK:for( KKK = 2 ; KKK <=  N; KKK += 1) {
          for(i=1;i<C51;PN[i]=0,i++);P=4*RASH[KKK];K=0;
```

```
        for( I = 1 ; I <=  NSOR; I += 1) {
         if (MTR1[KKK][4] > 0 )  {
             if ( MTR1[KKK][3] == 0 ) {
               if (MTR1[KKK][4] == KODM[I] ) d=TR[I][1];
                   else goto M5500;      }
             else  { di=TR[I][NP]*1000;
                if  (fabs(MTR1[KKK][3] - di)<0.001 )
                    d=TR[I][1];  else goto M5500;  }
          }
        d=TR[I][1];if( d >0 )  {  VD=P/(3.14*d*d);
       if( VD>VMAX )  goto  M5500;
       if( VD<SN )  goto  M5500;
       PN[I]=1.1*DLIN[KKK];
       if(( KODM[I]>=900) && (KODM[I]<1400))  {
               if( VD<VK )  {
              P1=pow((1+0.867/VD),0.3);
              P2=0.00148*RASH[KKK]*RASH[KKK]/pow(d,5.3);
              PN[I]=P2*P1*PN[I]; goto  M5500;  }
              if( VD>=VK )  {
                 P1=RASH[KKK]*RASH[KKK]/pow(d,5.3);
                 PN[I]=0.001735*P1*PN[I]; goto  M5500;
              }        }
    if( KODM[I]>=100 && KODM[I]<200 )  {
L1: P1=0.00005/d;
    P2=pow((P1+ 0.000089/(VD*d)),0.25);P3=0.00561*P2;
    PN[I]=P3*(VD*VD/d)*PN[I];  goto  M5500;       }
    if( KODM[I]>=200 && KODM[I]<300 )  {
L2: P1=pow(RASH[KKK],1.85)/pow(d,4.89);
   PN[I]=0.001486*P1*PN[I]; goto  M5500; }
```

```
    if( KODM[I]>=300 && KODM[I]<400 )  {
L3: P1=RASH[KKK]*RASH[KKK]/pow(d,5.33);
   PN[I]=0.00263*P1*PN[I]; goto  M5500;      }
    if( KODM[I]>=400 && KODM[I]<500 )  {
L4: P1=pow(RASH[KKK],1.85)/pow(d,4.89);
   PN[I]=0.001486*P1*PN[I]; goto  M5500;      }
    if(( KODM[I]>=500) && (KODM[I]<900))
      { L5: P1=pow(RASH[KKK],1.774)/pow(d,4.774);
   PN[I]=0.00105*P1*PN[I]; goto  M5500;      }
    if(( KODM[I]>=1400) && (KODM[I]<1800))  {
L14: P1=pow((1+3.51/VD),0.19);
    P1=(RASH[KKK]*RASH[KKK]/pow(d,5.19))*P1;
    PN[I]=0.00091*P1*PN[I]; goto  M5500;      }
    if( KODM[I]>=1800 && KODM[I]<1900 )  {
L18: P1=pow(VD,1.774)/pow(d,1.226);
    PN[I]=0.000857*P1*PN[I]; goto  M5500;      }
   if( KODM[I]==2000 )  {
   P1=0.00561*pow((0.00005/d+0.000089/(VD*d)),0.25);
   PN[I]=PN[I]*VD*VD*P1/d; }
M5500: if( PN[I] > 0 ) K=K+1; }
       }       IJ=0;
MGL3:  for( II = 1 ; II <=  NSOR; II += 1) {
       if( PN[II]== 0 )  goto  MGL2;
       IJ=IJ+1; PN2=PN[II];I1=II;
       for( J = II+1 ; J <=  NSOR; J += 1) {
       if( PN2<PN[J] )  {  PN2=PN[J];I1=J; }      }
        AB[KKK][IJ]=I1; POTER[KKK][IJ]=PN[I1]; PN[I1]=0;
       if( IJ==K )  goto  MGLM;  goto  MGL3;
MGL2: ;  } MGLM: ;  }
```

```
        for( I = N2 ; I <=  N; I += 1) {
        if( POTER[I][1]==0 ) {
 fprintf(printer,
"\nДЛЯ %d-ГО УЧАСТКА НЕ ВЫДЕРЖИВАЕТСЯ ОГРАНИЧЕНИЕ"
" ПО СКОРОСТИ",I);
printf("\nДЛЯ %d-ГО УЧАСТКА НЕ ВЫДЕРЖИВАЕТСЯ"
        " ОГРАНИЧЕНИЕ ПО СКОРОСТИ" ,I);RR=0; }
         } MKONS: ;puts("        END SORTAM");}
```

Приложение 4

Распечатка контрольного примера по задаче
Технико-экономического и гидравлического расчета магистральных водопроводов.

С А П Р - С К Г В Х

Т Л П - М В

Решается задача технико-экономического и гидравлического расчета МВ

Количество точек на вертикали напоров	5
Минимальная скорость течения воды	0.500 (м/с)
Максимальная скорость течения воды	3.500 (м/с)
Верхнее ограничение на напор насосной станции	200.000 (м)
Коэффициент "АЛЬФА" целевой функции	1.140
Коэффициент "БЕТТА" целевой функции	16.700
Усредненная стоимость одной насосной станции	70000 (руб)

Таблица П.4.1

Исходные данные по сети

Исходные данные для гидравлического расчета МВ								
Наименование объекта - Чиркский групповой водопровод								
Составил					Наименование схемы		Стадия	
Проверил							Чертеж Шифр	
ГИП							Дата	
Участок	Длина участка сети	Высотная отметка узла	Расход участка сети	Свободный напор в узле	Шифр трубы	Место расположение	Примечание	
1	2	3	4	5	6	7	8	
0-1	0.00	82.5	0.18670	0.00	0	3		
1-2	300.0	92.0	0.18670	0.00	0	0		
2-3	600.0	91.5	0.14020	18.00	0	3		
3-4	3540.0	93.0	0.07691	0.00	0	0		
4-5	1520.0	99.4	0.07379	0.00	0	0		
5-6	6220.0	90.0	0.07379	0.00	0	3		
6-7	4200.0	105.0	0.06943	18.00	0	0		
7-8	4100.0	110.0	0.06866	18.00	0	0		

Продолжение таблицы П.4.1

1	2	3	4	5	6	7	8
8-9	2680.0	180.5	0.05746	0.00	0	0	
9-10	4280.0	118.0	0.03510	0.00	0	0	
10-11	5300.0	118.7	0.02890	0.00	0	0	
9 -12	3400.0	100.0	0.05000	18.00	0	0	
12-13	4560.0	99.0	0.04500	18.00	0	0	
13-14	5600.0	130.0	0.04000	10.00	0	0	
6 -15	4500.0	100.0	0.03200	18.00	0	0	
15-16	6700.0	150.0	0.02500	0.00	0	0	
16-17	7800.0	98.0	0.02300	0.00	0	0	
7 -18	2300.0	110.0	0.02400	0.00	0	0	
18-19	4670.0	97.0	0.02300	0.00	0	3	
19-20	5400.0	110.2	0.02000	0.00	0	0	
19-21	6780.0	113.0	0.02300	0.00	0	0	
21-22	9870.0	89.6	0.02100	0.00	0	0	

Таблица П.4.2

Заданный сортамент труб

Наличие ограничений на сортамент	0	Количество сортаментов труб	3

Шифр сортамента
1300 1500 1600

Таблица П.4.3

Таблица выходных данных по гидравлическому расчету водоводов

Участок	Длина участка	Расход воды на участке	Высотная отметка в начале	Высотная отметка в конце	Диаметр трубы Д1	Материал трубы Д1	ГОСТ трубы Д1	Длина участка с Д1	Скорость воды в Д1
1	2	3	4	5	6	7	8	9	10
6-15	4500	.0320	90.0	100.0	0.279	А,ВТ9	539-80	0.0	0.000
15-16	6700	.0250	100.0	150.0	0.189	А,ВТ9	539-80	6700.0	0.892
16-17	7800	.0230	150.0	98.0	0.189	А,ВТ9	539-80	3511.1	0.820
6-7	4200	.0694	90.0	105.0	0.368	А,ВТ9	539-80	0.0	0.000
7-8	4100	.0686	105.0	110.0	0.279	А,ВТ9	539-80	4100.0	1.124
8-9	2680	.0574	110.0	180.5	0.279	А,ВТ9	539-80	2680.0	0.940
9-10	4280	.0351	180.5	118.0	0.189	А,ВТ9	539-80	4280.0	1.252
10-11	5300	.0289	118.0	118.7	0.189	А,ВТ9	539-80	5279.8	1.030
9 -12	3400	.0500	180.5	100.0	0.279	А,ВТ9	539-80	3400.0	0.818
12-13	4560	.0450	100.0	99.0	0.279	А,ВТ9	539-80	800.6	0.736
13-14	5600	.0400	99.0	130.0	0.235	А,ВТ9	539-80	5600.0	0.923
7 -18	2300	.0240	105.0	110.0	0.189	А,ВТ9	539-80	2300.0	0.856
18-19	4670	.0230	110.0	97.0	0.189	А,ВТ9	539-80	4670.0	0.820
19-20	5400	.0200	97.0	110.2	0.141	А,ВТ9	539-80	5279.4	1.280
19-21	6780	.0230	97.0	113.0	0.189	А,ВТ9	539-80	6780.0	0.820
21-22	9870	.0210	113.0	89.6	0.189	А,ВТ9	539-80	6504.1	0.750

Окончание таблицы П.4.3

Участок	Диаметр трубы Д2	Материал трубы Д2	ГОСТ трубы Д2	Длина участка с Д2	Скорость воды в Д2	Потери напора на участке	Потери напора с учетом сопротивлений	Свободный напор в конце участка	Стоимость участка
11	12	13	14	15	16	17	18	19	20
6-15	0.279	A,BT9	539-80	4500.0	0.53	4.659	5.125	119.4	24.12
15-16	0.141	A,BT9	539-80	0.0	0.00	29.371	32.308	37.4	20.84
16-17	0.141	A,BT9	539-80	4288.9	1.47	80.948	89.042	0.0	19.24
6 -7	0.368	A,BT9	539-80	4200.0	0.65	4.693	5.163	114.3	37.88
7 -8	0.235	A,BT9	539-80	0.0	0.00	17.356	19.092	90.2	21.98
8 -9	0.235	A,BT9	539-80	0.0	0.00	8.156	8.972	10.8	14.37
9 -10	0.141	A,BT9	539-80	0.0	0.00	35.197	38.717	34.5	13.31
10-11	0.141	A,BT9	539-80	20.2	1.85	30.757	33.832	20.0	16.46
9 -12	0.235	A,BT9	539-80	0.0	0.00	8.002	8.803	82.5	18.22
12-13	0.235	A,BT9	539-80	3759.5	1.04	18.404	20.244	63.2	19.50
13-14	0.189	A,BT9	539-80	0.0	0.00	20.184	22.203	10.0	22.68
7 -18	0.141	A,BT9	539-80	0.0	0.00	9.350	10.285	99.0	7.15
18-19	0.141	A,BT9	539-80	0.0	0.00	17.549	19.304	92.7	14.52
19-20	0.100	A,BT9	539-80	120.6	2.55	72.294	79.523	0.0	10.38
19-21	0.141	A,BT9	539-80	0.0	0.00	25.479	28.027	48.7	21.09
21-22	0.141	A,BT9	539-80	3365.9	1.35	65.540	72.097	0.0	26.76

Напор на насосной станции равен 134.475

Стоимость труб сети равна	308.51
Стоимость насосной станции равна	70.00
Энергетические затраты равны	71.86
Приведенные затраты равны	493.561
Стоимости печатаются в тыс.руб.	
Станция перекачки в начале участка	15

Таблица П.4.4

Таблица выходных данных по гидравлическому расчету водоводов

Участок	Длина участка	Расход воды на участке	Высотная отметка в начале	Высотная отметка в конце	Диаметр трубы Д1	Материал трубы Д1	ГОСТ трубы Д1	Длина участка с Д1	Скорость воды в Д1
1	2	3	4	5	6	7	8	9	10
1-2	300	.1867	82.5	92.0	0.456	А,ВТ9	539-80	300.0	1.144
2-3	600	.1402	92.0	91.5	0.456	А,ВТ9	539-80	298.1	0.859
3-4	3540	.0769	91.5	93.0	0.368	А,ВТ9	539-80	3540.0	0.723
4-5	1520	.0738	93.0	99.4	0.368	А,ВТ9	539-80	1520.0	0.694
5-6	6220	.0738	99.4	90.0	0.368	А,ВТ9	539-80	5307.2	0.694

Окончание таблицы П.4.4

Участок	Диаметр трубы Д2	Материал трубы Д2	ГОСТ трубы Д2	Длина участка с Д2	Скорость воды в Д2	Потери напора на участке	Потери напора с учетом сопротивлений	Свободный напор в конце участка	Стоимость участка
11	12	13	14	15	16	17	18	19	20
1-2	0.368	А,ВТ9	539-80	0.0	0.000	0.732	0.805	19.33	3.96
2-3	0.368	А,ВТ9	539-80	301.9	1.319	1.666	1.832	18.00	6.69
3-4	0.279	А,ВТ9	539-80	0.0	0.000	4.776	5.253	11.20	31.9
4-5	0.279	А,ВТ9	539-80	0.0	0.000	1.900	2.090	2.76	13.7
5-6	0.279	А,ВТ9	539-80	912.8	1.208	11.052	12.157	0.00	52.7

Напор на насосной станции равен 29.637

Стоимость труб сети равна 109.02

Стоимость насосной станции равна 70.00

Энергетические затраты равны 92.4

Приведенные затраты равны 286.690

Стоимости печатаются в тыс.руб.

Станция перекачки в начале участка 2

Общая стоимость равна 780.251 тысяч рублей.

Приложение 5

Распечатка исходных текстов модулей прогограммы задачи Технико-экономического и гидравлического расчета магистральных водопроводов

```
/*****************************************************************/
/*                    Головная процедура                         */
/*****************************************************************/
 BETTPLI:GRIET:PROC(PCB,PCB1) OPTIONS(MAIN);
     DCL (PCB,PCB1) POINTER, KTV EXT,(CTVV1,CTVV2) EXT,
      MTR(300,50)  FIXED DEC(3),
      1 STRUCT1 EXT,
        2 TR(100,10),2 ST(100,10),2 DVL(100,10),
        2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
        2 GOST(100) CHAR(10),2 MT(100) CHAR(10),
        2 TRM(100) CHAR(1),
        2 AB(300,50) FIXED DEC(3),2 POTER(300,50),
        2 V(100) FIXED DEC(1), 2 HOPCTH(10,10),
        1 STRUCT2 EXT,
         2 N1,2 N2,2 NSOR,2 NP,2 SN,2 VK,2 VMAX,
         2 TRC,2 TCH,2 TG,2 TA,2 RTN,2 POLI,
         2 DVLG,2 DVCH,2 DVLR,2 DVLA,2 DVLP,2 RR,2 KR,
      (POLI1,C,A,G,CH,RTN1) CHAR(5)EXT,
      1 STRUCT(300) EXT,
          2 DLIN, 2 VIS, 2 RASH, 2 HSB,
          2 MATER FIXED DEC(2), 2 MPN FIXED DEC(1), 2 OGRH,
/*МАССИВЫ ДЛЯ ЗАДАНИЯ ДИАМЕТРОВ ПО УЧАСТКАМ СЕТИ          */
        (TRR,STT,KODD FIXED(4))(300) EXT;
        DCL (FAMNSX,FAMGIP,FAMISX,FAMPRO,FAMNOA,FAMPER,
           FAMIVM ) CHAR(12),
         (DATA,SHIFO,SHIFP,STAD)CHAR(8), CHERT CHAR(5),
         LIST DEC(4) EXT,
           1 OB EXT,
            2 IMOB CHAR(23),2 IMOB1 CHAR(23),2 IMOB2 CHAR(23),
```

```
          1 BOD EXT, 2 IMBOD CHAR(23),2 IMBOD1 CHAR(23),
           2 IMBOD2 CHAR(23);
        VK=1.2;VMAX=3.0;SN=0.5;LIST=1;
        GOST=' ';MT=' ';NKS=50;  TRR,STT,KODD=0;
       DVLA=9;DVLG=150;DVLCH=160;DVLR=150;STRUCT=0;
  PUT EDIT('С А П Р - С К Г В Х')(SKIP,X(29),A);
  PUT EDIT('Т Л П - М В')(SKIP(2),X(31),A);
  PUT EDIT('РЕШАЕТСЯ ЗАДАЧА ТЕХНИКО-ЭКОНОМИЧЕСКОГО'
           'И ГИДРАВЛИЧЕСКОГО РАСЧЕТА '
(SKIP(2),X(13),A);PUT SKIP;PUT SKIP;PUT SKIP;
        CTVV1=0;CTVV2=10**10;NP=9;DVCH=DVLCH;N1=1;N2=2;
   GET LIST(KTV);   /* КОЛИЧЕСТВО ТОЧЕК НА ВЕРТИКАЛИ */
  PUT EDIT('КОЛИЧЕСТВО ТОЧЕК НА ВЕРТИКАЛИ НАПОРОВ',KTV)
 (SKIP   ,A,X(8),F(3)); GET LIST(SN,VMAX);
  PUT EDIT('МИНИМАЛЬНАЯ СКОРОСТЬ ТЕЧЕНИЕ ВОДЫ',SN,'(М/СЕК)',
 'МАКСИМАЛЬНАЯ СКОРОСТЬ ТЕЧЕНИЯ ВОДЫ',VMAX,' ' (М/СЕК)')
  (SKIP   ,A,X(11),F(8,3),X(1),A,SKIP,A,X(10),F(8,3),A);
   GET LIST(POGRH);
  PUT EDIT('ВЕРХНЕЕ ОГРАНИЧЕНИЕ НА НАПОР НАСОСНОЙ СТАНЦИИ'
 ',POGRH,'(М)') (SKIP,A,F(8,3),X(1),A);OGRH=POGRH;
  GET LIST(ALFA,BETA,STNS);
  PUT EDIT('КОЭФФИЦИЕНТ <<АЛФА>> ЦЕЛЕВОЙ ФУНКЦИИ',ALFA)
          (SKIP   ,A,X(9),F(8,3));
  PUT EDIT('КОЭФФИЦИЕНТ <<БЕТА>> ЦЕЛЕВОЙ ФУНКЦИИ',BETA)
          (SKIP,A,X(9),F(8,3));
PUT EDIT('УСРЕДНЕННАЯ СТОИМОСТЬ ОДНОЙ НАСОСНОЙ СТАНЦИИ',
 STNS,  '(ТЫС.РУб)')(SKIP,A,X(1),F(8,3),X(1),A);PUT SKIP(3);
CALL VVOD1(MTR,KW);  KT=47;
/*--КОЛИЧЕСТВО СВОБО'НЫХ СТОЛБ. В ОСНОВ. МАССИВАХ--*/
```

```
  OGRH(KW+1)=ALFA;OGRH(KW+2)=BETA;OGRH(KW+3)=STNS;
  K1=1;DO I=1 TO KW;K=0;
  DO J=1 BY 1 WHILE(MTR(I,J)^=0);K=K+1;END;
  IF K>KT THEN DO;  K1=0;
  PUT EDIT('ТРАЕКТОРИЯ ВЕРШИНЫ',I,'
'СОДЕРЖИТ БОЛЬШЕ ЧЕМ ДОПУСТИМО'^
  KT ,'ВЕРШИН')(SKIP,2(A,F(3)),A);END;END;
IF K1=0 THEN GOTO MCON1;
        IF MTR(1,1)=-1 THEN GOTO MCON1;
  KA=0;DO I=1 TO KW;IF MATER(I)>0 THEN KA=1;END;
CALL SORTAM(PCB,KW);
/*  ОБРАЩЕНИЕ ВПРОС. ФОРМИР ПОТЕР*/
IF KA=1 THEN CALL POTUCH(KW); DVL=0;DVLA=DVLA*10;
  C1000:DO I=1 TO NSOR;IF TRM(I)=C THEN DO;
  IF TR(I,NP)<=1THEN P7=1;
IF TR(I,NP)>1&TR(I,NP)<=1.2 THEN P7=1.05;
  IF TR(I,NP)>1.2 THEN P7=1.1;
  DO J=1 TO NP-1;
  IF TR(I,J)^=0 THEN DO;P5=TR(I,NP)-TR(I,J);P5=100*P5/2;
  P6=P7*(100*TR(I,NP)-2*P5);
  DVL(I,J)=1558.442*P5/P6;DVL(I,J)=10*DVL(I,J);END;END;
  END;END C1000; ND=50;
        DO I=1 TO KW;
  IF AB(I,NKS-2)>0 THENPUT EDIT('НА УЧАСТКЕ ',I, ' БОЛЕЕ,ЧЕМ
',NKS-2,  ' ДИАМЕТРОВ,ЧТО НЕ ДОПУСТИМО','ИЗМЕНИТЕ
ОГРАНИЧЕНИЕ НА СКОРОСТЬ')  (SKIP(2),A,F(4),A,F(4),A,SKIP,A);
 AB(I,NKS)=MPN(I);/*--ЗАДАННЫЕ МЕСТОРАСПОЛОЖЕНИЯ Н.С.--*/END;
  MCON:  CTVV1=0;MOV=0;IND=0;
        DO I=1 TO KW; MPN(I)=AB(I,NKS);END;
```

```
 CTVV3=10**10;PR=ABS(CTVV3-CTVV1);KTV3=KTV-1;PR=CTVV3-CTVV1;
 C0001:DO KIGL=1 BY 1 WHILE(PR>5);
KMN=0;DELH=OGRH(1)-HSB(1);MOV=0;
   C0002:DO I=1 TO KW;
         IF KIGL=1&MPN(I)^=0 THEN
         DO;IF MPN(I)=3 THEN POTER(I,NKS)=OGRH(I);
            ELSE POTER(I,NKS)=OGRH(I)-HSB(I);
          POTER(I,NKS)=POTER(I,NKS)/KTV3;END;
        IF KIGL^=1&MPN(I)^=0 THEN DO;
        I1=MTR(I,2); KMN=KMN+1;
       DELH=POTER(I,NKS); IF MPN(I)=3 THEN
         POTER(I,NKS)=OGRH(I)/KTV3;
        ELSE POTER(I,NKS)=(OGRH(I)-HSB(I1))/KTV3;
         END; END C0002;
  KL=KMN*KTV-1;  DO I=1 TO KL;   DO J=1 TO 10;
   IND(I,J,1)=0;END;END; CALL OPPUT(MTR,PCB1,DELH,KW,KIGL);
    PR=ABS(CTVV3-CTVV1); PR=CTVV3-CTVV1; CTVV3=CTVV1;
   IF CTVV1>=10**10&KIGL=1 THEN DO;
  PUT EDIT('ЗАДАЧА НА ДАННОМ СОРТАМЕНТЕ НЕ РЕШАЕТСЯ')(SKIP,A);
  PUT EDIT('РАСЧЕТ ПОВТОРЯЕТСЯ С ИЗМЕННЫМ ШАГОМ')(SKIP,A);
  PUT EDIT('ИЛИ ИЗМЕНИТЕ СОРТАМЕНТ')(X(40),A);KTV=KTV+1;
         GOTO MCON;  END;  END C0001;
  MCON1: PUT SKIP;PUT SKIP;PUT SKIP;
  PUT EDIT('АЛГОРИТМ И ПРОГРАММА ЗАДАЧИ РАЗРАБОТАНЫ
В НИИ ПММ КАБАРДИНО-БАЛКАРСКОГО ГОСУНИВЕРСИТЕТА В 1980 Г.)
(SKIP,A);     END BETTPLI;
/*****************************************************************/
/*          Подпрограмма определения стоимости пути          */
/*****************************************************************/
```

```
CTOPUT:PROC(II,DELH,MTRN,KH,ITN,MTR);
 DCL  ( MTR(300,50), MTRN(50,10),ITN(50,2)) FIXED DEC(3),
 1 STRUCT1 EXT,
 2 TR(100,10),2 ST(100,10),2 DVL(100,10),
 2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
 2 GOST(100) CHAR(10),2 MT(100) CHAR(10),
 2 TRM(100) CHAR(1), 2 AB(300,50) FIXED DEC(3),
 2 POTER(300,50), 2 V(100) FIXED DEC(1), 2 HOPCTH(10,10),
      1 STRUCT(300) EXT,
           2 DLIN, 2 VIS, 2 RASH, 2 HSB, 2 MATER FIXED DEC(2),
           2 MPN FIXED DEC(1), 2 OGRH,
         KTV EXT;  NKS=50;  ND=50;
         DO I=1 BY 1 WHILE(MTR(I,1)^=0);
         MTR(I,ND-1)=0;END;
       K1=1; DO I=1 TO II; IF MTRN(I,2)^=0 THEN K1=0; END;
       IF K1=1 THEN III=II; ELSE III=II-1;
         /*MOV(30)-МАССИВ ОбРАбОТАННЫх Н.С.*/
         /*бЛОК СГМЕНИЯ ОГРАНИЧЕНИИ*/
   C0019:DO I=III BY -1 TO 1;
         K=MTRN(I,1);/*K-НОМЕР ПРЕДШЕСТВ&ВЕРТИКАЛИ*/KK=MTRN(I,2);
/*KK-НОМЕР ВЕРТИК.С КОТОРОИ СВЯЗ.К-АЯ*/
       IF KK=0 THEN KK=MTRN(I,1);
      KJ1=MOV(KK);/*KJ1-ТОЧКА ВЕРТИКАЛИ С ОРТИМ.СТОИМ.*/
         KV1=ITN(KK,2); DELH=POTER(KV1,NKS);
         KLJ=IND(KJ1,1,1); KLJ1=KLJ-KK*KTV+1;
         IF KLJ1=KTV THEN KLJ1=KLJ1-1;OGRH(KV1)=KLJ1*DELH;
   C0020:DO J1=2 BY 1 WHILE(IND(KJ1,J1,1)^=0);
 KI1=IND(KJ1,J1,1);/*KJ1-ТОЧКИ ВЕРТИК.С КОТОР.СОЕДЕН.*/
         KI2=KI1/KTV;     /*KI2-НОМЕР ЭТОИ ВЕРТИКАЛИ*/
```

```
        IF KI2=K THEN   DG021:DO;
        KI11=IND(KI1,1,1);/*KI11-ТОЧКА НА ВЕРТИКАЛИ ЧЕРЕЗ
        КОТОР.ПРОХОДИТ ПУТЬ*/
        KV1=ITN(K,2);/*--KV1--НОМЕР ВЕРШИНЫ СЕТИ ----*/
        KV2=MTR(KV1,2);/* KV2-ФАКТИЧЕСКАЯ ВЕРШИНА С Н.С.*/
    DELH=POTER(KV1,NKS);/* ШАГ ПО НАПОРУ НА ВЕРШИНЕ KV1 */
       KJK1=KI1-KI2*KTV;
        IF KJK1>=1 THEN   DG022:DO;
        IV2=KV2;  IF MPN(KV1)=3 THEN MTR(KV1,ND)=0 ;
        IF MTR(IV2,ND-1)^=IV2 THEN DO;P=HSB(IV2);
       IF (KH=1!KH=0)&MPN(KV1)^=3 THEN
        HSB(IV2)=P+(KJK1-1)*DELH;
  IF KH=1&MPN(KV1)=3 THEN HSB(IV2)=KJK1*DELH;   END;
        MTR(IV2,ND-1)=IV2; END DG022;
 KLJ2=KI11-KI2*KTV+1;IF KLJ2=KTV THEN KLJ2=KLJ2-1;
        OGRH(KV1)=KLJ2*DELH;  MOV(K)=KI1;
        END DG021;   END C0020; END C0019;
 C0022:DO I=1 TO II;K1=MTRN(I,1);KV2=ITN(K1,2);KV1=MTR(KV2,2);
 C0023:DO J=I+1 TO II;K11=MTRN(J,1);KV22=ITN(K11,2);
 KV11=MTR(KV22,2); IF KV1=KV11 THEN DO;
        IF OGRH(KV2)<OGRH(KV22) THEN OGRH(KV2)=OGRH(KV22);
        ELSE OGRH(KV22)=OGRH(KV2);     END;
  END C0023; END C0022; END CTOPUT;
/*****************************************************************/
/* Подпрограмма построения дерева насосных станций        */
/*****************************************************************/
 MATRNS:PROCEDURE(MTR,ITN,II,MTRN);
  DCL  (MTR(300,50), ITN(50,2),MTRN(50,10))  FIXED DEC(3),
     1 STRUCT(300) EXT,2 DLIN, 2 VIS, 2 RASH, 2 HSB,
```

```
          2 MATER FIXED DEC(2), 2 MPN FIXED DEC(1),
          2 OGRH      ;  ND=50;
        MTRN=0;/*МАТРИЦА ТРАЕКТ.Н.С.*/
   C0014:DO I=1 TO II;
        K1=ITN(I,2); MTRN(I,1)=I; JJ=1;
   C0015:DO J=2 TO ND-1 WHILE(MTR(K1,J)^=0); I1=MTR(K1,J);
        IF MPN(I1)^=0 THEN
   DG024:DO;JJ=JJ+1;
   C0016:DO J1=1 TO II;
        IF I1=ITN(J1,2) THEN  MTRN(I,JJ)=J1; END C0016;
        END DG024;  END C0015;  END C0014; END MATRNS;
/*****************************************************************/
/*   Подпрограмма определения оптимальных соединений     */
/*****************************************************************/
   OPPUT:PROCEDURE(MTR, PCB1,  DELH,KW,KIGL);
   DCL(  MTR(300,50), ITN(50,2), MSV(20,2),
MTRK(100,30) )FIXED  DEC(3), PCB1 POINTER, (NUT,KW,N),
      (CTV(500), HSB1(100),DELH), KTV EXT,
      1 STRUCT1 EXT, 2 TR(100,10),2 ST(100,10),
        2 DVL(100,10), 2 MOV(50) FIXED DEC(3),
        2 IND(500,10,2) FIXED DEC(3), 2 GOST(100) CHAR(10),
        2 MT(100) CHAR(10), 2 TRM(100) CHAR(1),
        2 AB(300,50) FIXED DEC(3),2 POTER(300,50),
        2 V(100) FIXED DEC(1), 2 HOPCTH(10,10),
      1 STRUCT(300) EXT,2 DLIN, 2 VIS, 2 RASH,2 HSB,
       2 MATER FIXED DEC(2), 2 MPN FIXED DEC(1),  2 OGRH;
         MTRK=0;HSB1=0;CTV=10**10;/*МАССИВ СТОИМ-ЕЙ ПУТИ */
         NKS=50; ND=50;
          IF KIGL=1 THEN   MD001:DO;
```

```
        /*БЛОК ПЕРЕСТРОЕНИЯ МЕСТОП&НАС&С&  */
  C0001:DO I=1 TO KW;  IF MPN(I)^=0 THEN
        C0002 :DO I1=1 TO KW;
        IF MTR(I1,2)=I THEN  D0001:DO;
            POTER(I1,NKS)=POTER(I,NKS); CTV(I1)=OGRH(I);
        IF MPN(I)=1 THEN MTR(I1,ND)=10;
        IF MPN(I)=2 THEN MTR(I1,ND)=20;
        IF MPN(I)=3 THEN MTR(I1,ND)=30;
        IF MPN(I)=4 THEN MTR(I1,ND)=40;
        END D0001; END C0002; END C0001; KNST=0;
        /*KNST-КОЛИЧ-ВО Н.С. НА СЕТИ*/
  C0003:DO I=1 TO KW;  MPN(I)=0;
        IF CTV(I)^=10**10 THEN OGRH(I)=CTV(I);
       CTV(I)=10**10;  IF MTR(I,ND)^=0 THEN KNST=KNST+1;
      IF MTR(I,ND)=10 THEN MPN(I)=1;
        IF MTR(I,ND)=20 THEN MPN(I)=2;
        IF MTR(I,ND)=30 THEN MPN(I)=3;
        IF MTR(I,ND )=40 THEN MPN(I)=4; MTR(I,ND)=0;
      END C0003;  END MD001;
        CTV=10**10; ITN=0;/*МАССИВ ПРОСЧИТАНЫХ Н.С.*/
      II=0; /*ПОРЯДКОВ НОМЕР ПО УДАЛЕННОСТИ*/
 KNST=0; DO I=1 TO KW; IF MPN(I)^=0 THEN KNST=KNST+1; END;
     IF KNST>50 THEN DO;
PUT EDIT('КОЛИЧЕСТВО ВОЗМОжНЫХ МЕСТ РАСПОЛОжЕНИИ  (', KNST,
')',  'НАСОСНЫх СТАНЦИИ ПОЛУЧИЛОСЬ(С УЧЕТОМ '
'РАЗДВОЕНИЯ ОДНОЙ НАСОСНОЙ СТАНЦИИ
   В ТОЧКАХ РАЗВЕТВЛЕНИЙ)','  БОЛЬШЕ ЧЕМ ДОПУСТИИМО'(НЕ БОЛЕЕ
50-ТИ)')  (SKIP(2),A,F(3),3 A);
  PUT SKIP LIST('УМЕНЬШИТЕ КОЛИЧЕСТВО ВОЗМОж.МЕСТ И '
```

```
'ПОВТОРИТЕ ЗАДАНИЕ');  P=0; K=1;R=K/P;END;
 NUT=0;/*УДАЛЕННАЯ Н.С.*/
 MSV=0;/*МАССИВ ПРОСЧИТ.ВЗАИМОСВЯЗ.Н.С.*/
   C0004:DO IGL=1 TO KNST;
         NUT=UDT(MTR,KW); II=II+1;
         ITN(II,1)=NUT;ITN(II,2)=NUT;
         CALL PMTRK(NUT,MTR, KW,N,MTRK);
         KS=0; /*БЛОК НАХОЖДЕНИЯ ВЗАИМОСВ.Н.С.*/
         IF II>1 THEN
    D0002 :DO; KS=0; MSV=0;/*MSV-МАССИВ ВЗАИМОСВ.Н.С.*/
    C0005:DO I=1 TO II-1;
         I1=ITN(I,1);/*I1-ВЕРШИНА С Н.С.*/
         IF I1=0  THEN GOTO M0000;
   C0006:DO J=1 TO ND-1 WHILE(MTR(I1,J)^=0);
   IF NUT=MTR(I1,J) THEN D0003:DO;
   KS=KS+1;/*KS-КОЛИЧЕСТВО ВЗАИМОСВ.Н.С.*/
   ITN(I,1)=0; MSV(KS,1)=I1;/*MSV(KS,1)-НОМЕРА ВЕРШИН*/
   MSV(KS,2)=I;/*MSV(KS,2)-НОМЕРА ВЕРТИКАЛЕЙ*/
 END D0003; END C0006;  M0000: END C0005; END D0002;
         /*БЛОК ЗАПОЛНЕНИЯ СВОБ.НАПОРОВ КУСТА*/
C0007:DO I=1 TO N; I1=MTRK(I,1); HSB1(I)=HSB(I1); END C0007;
 CALL VIBORI(HSB1,MTRK,MSV,KNST,PCB1,DELH,KIGL,IGL,
           MTR,ITN,KS,CTV); MPN(NUT)=0; END C0004;
         DO I=1 TO KW; MPN(I)=MTR(I,ND);END; END OPPUT;
/*****************************************************************/
/*      Подпрограмма оптимизации кустов                          */
/*****************************************************************/
   OPTIMM:PROCEDURE(NDT,PRIZ,HSB1,N,MTR);
      DCL (C,CH,A,G,RTN1,POLI1) CHAR(5) EXT,
```

```
    1 STRUCT2 EXT, 2 N1,2 N2,2 NSOR,2 NP,
    2 SN,2 VK,2 VMAX, 2 TRC,2 TCH,2 TG,2 TA,2 RTN,2 POLI,
    2 DVLG,2 DVCH,2 DVLR,2 DVLA,2 DVLP,2 RR,2 KR,
  1 STRUCT1 EXT, 2 TR(100,10),2 ST(100,10),2 DVL(100,10),
   2 MOV(50) FIXED DEC(3), 2 IND(500,10,2) FIXED DEC(3),
   2 GOST(100) CHAR(10),2 MT(100) CHAR(10),
   2 TRM(100) CHAR(1), 2 AB(300,50) FIXED DEC(3),
   2 POTER(300,50),2 V(100) FIXED DEC(1), 2 HOPCTH(10,10),
  1 STRUCT(300) EXT, 2 DLIN, 2 VIS,2 RASH,2 HSB,
    2 MATER FIXED DEC(2),2 MPN FIXED DEC(1),2 OGRH,
PN(300), NDT FLOAT DEC,A1(100)FIXED DEC(2),MTR(100,30)FIXED
DEC(3),(EFK(100),HAP(100),HSB1(100)),N,(TRR,STT)(300) EXT;
    DVLCH=DVCH;
  M0001:POLI1='П/Э';RTN1='P'; ND=30;ND1=50;
  A1=0;HAP=0;PN=0;KF=3;  KOLRAZ=0;
  IF KR=0 THEN DO; DO I=1 TO NSOR;IF TRM(I)=C THEN DO;
   DO J=1 TO NP-1;  ST(I,J)=KF*ST(I,J);END; END;END;END;
        DO I=N2 TO N; J2=MTR(I,1); IF PRIZ^=0 THEN
        MTR(I,ND-1)=MTR(I,ND);J1=MTR(I,ND);
        DO KK=1 BY 1 WHILE(AB(J2,KK)^=J1);END;
        PN(J2)=POTER(J2,KK);END;KK1=MTR(1,1);
        DO I=N2 TO N;KK2=MTR(I,1);SP=0;
        DO J=N1 TO ND-2 WHILE(MTR(I,J)^=KK1);
        K=MTR(I,J);SP=SP+PN(K); END;
  M3666:HAP(I)=NDT-SP+VIS(KK1)-VIS(KK2);END;
  M1122:HAP(1)=NDT;  DO I=N2 TO N;
IL1=MTR(I,2);DO KII=1 TO N WHILE(MTR(KII,1)^=IL1);END;
IL=KII;  K1=MTR(I,1); J1=MTR(I,ND);J4=V(J1);
 DO J=1 TO ND1 WHILE(AB(K1,J)^=0);
```

```
  IF J1=AB(K1,J) THEN K=J;END;
  IF K=1 THEN DO;  A1(I)=J1;EFK(I)=-1; PN(K1)=POTER(K1,K);
  GO TO M4001;END;  PN(K1)=POTER(K1,K);EFK(I)=0;
 DO J=1 TO K-1;  J3=AB(K1,J); IF TRM(J3)=A THEN PDAV=DVLA;
 IF TRM(J3)=G THEN PDAV=DVLG; IF TRM(J3)=CH THEN PDAV=DVCH;
 IF TRM(J3)=RTN1 THEN PDAV=DVLR; IF TRM(J3)=POLI1 THEN
PDAV=DVLP;
 IF TRM(J3)=C THEN DO JK=1 TO NP-1;
 IF HAP(IL)<=DVL(J3,JK) THEN DO; PDAV=DVL(J3,JK);
GOTO M3777;END; END;  M3777: IF HAP(IL)<=PDAV THEN DO;
         PN2=POTER(K1,J);IF PN2=PN(K1) THEN DO;
         IF ST(J3,1)<ST(J1,J4) THEN  DO;
         MTR(I,ND)=J3;GO TO M1122; END;ELSE GO TO M4444;END;
A2=ST(J1,J4)-ST(J3,1);P2=PN2-PN(K1);EFK2=DLIN(K1)*A2/P2;
IF EFK(I)<EFK2 THEN DO;EFK(I)=EFK2;A1(I)=J3;END; END;
M4444: END;
M4001:IF EFK(I)=0 THEN DO;A1(I)=MTR(I,ND);EFK(I)=-1;END;END;
  M1020:EFK1=EFK(2);I2=2; DO I=N2 TO N;KK2=MTR(I,1);SP=0;
        DO J=N1 TO ND-2 WHILE(MTR(I,J)^=KK1);
         K=MTR(I,J);SP=SP+PN(K);END;
         IF EFK1<=EFK(I) THEN DO; EFK1=EFK(I);I2=I;END;
  M3555:HAP(I)=NDT-SP+VIS(KK1)-VIS(KK2);END;K1=MTR(I2,1);
        J2=A1(I2);RES=1000; DO I=N1 TO N;
        DO J=N1 TO ND-2 WHILE(MTR(I,J)^=0);
         IF K1=MTR(I,J) THEN DO; HAPM=HAP(I)-HSB1(I);
         IF RES>HAPM THEN DO;RES=HAPM;I3=I;END;END; END;
        END; IF RES<0 THEN RES=0;
        DO J=1 TO ND1-3 WHILE(AB(K1,J)^=0);
         IF J2=AB(K1,J) THEN K=J;END;P10=POTER(K1,K)-PN(K1);
```

```
        IF K=1 THEN  IF RES>P10 THEN
        DO;PN(K1)=POTER(K1,K);EFK(I2)=-1;
        GO TO M1018; END;
         IF RES>P10 THEN  DG001:DO;J1=J2 ;I=I2;
PN(K1)=POTER(K1,K);J4=1;EFK(I)=0;
DO J=1 TO K-1;J3=AB(K1,J);PN2=POTER(K1,J);IL1=MTR(I,2);
DO KII=1 TO N WHILE(MTR(KII,1)^=IL1);END;IL=KII;
IF TRM(J3)=A THEN PDAV=DVLA;IF TRM(J3)=G THEN PDAV=DVLG;
IF TRM(J3)=CH THEN PDAV=DVCH;IF TRM(J3)=RTN1 THEN PDAV=DVLR;
 IF TRM(J3)=POLI1 THEN PDAV=DVLP;IF TRM(J3)=C THEN
DO JK=1 TO NP-1;IF HAP(IL)<=DVL(J3,JK) THEN DO;
PDAV=DVL(J3,JK);GOTO M3888;END;END;  M3888:;
        IF  HAP(IL)<=PDAV  THEN DO;
         IF PN2=PN(K1) THEN DO;IF ST(J3,1)<ST(J1,J4) THEN
        DO; K=K-1;J2=J3;A1(I)=J3; GO TO DG001;END;
         ELSE GO TO M3000; END;
         A2=ST(J1,J4)-ST(J3,1);
         P10=PN2-PN(K1);EFK2=DLIN(K1)*A2/P10;
        IF EFK(I)<EFK2 THEN DO;EFK(I)=EFK2;A1(I)=J3;END;
          END;  M3000:END;
IF EFK(I)^=0 THEN DO;MTR(I,ND)=J1;GO TO M1020;END;
IF EFK(I)=0 THEN DO;EFK(I)=-1;GO TO M1018;END;
 END DG001; ELSE PN(K1)=PN(K1)+RES;
  M3100: DO J=N1 TO ND-2 WHILE(MTR(I3,J)^=0);
        J3=MTR(I3,J); DO J1=1 TO N;
         IF J3=MTR(J1,1) THEN EFK(J1)=-1;END;END;
  M1018: DO I=N2 TO N; KK2=MTR(I,1);SP=0;
        DO J=N1 TO ND-2 WHILE(MTR(I,J)^=KK1);
         K=MTR(I,J);SP=SP+PN(K); END;
```

```
   M1013:HAP(I)=NDT-SP+VIS(KK1)-VIS(KK2);
 POTER(KK2,ND1-2)=HAP(I); END;
DO I=N2 TO N;IF EFK(I)^=-1 THENGO TO M1020;END;
KF=3;IF KOLRAZ=0 THEN DO;KOLRAZ=1;GO TO M1122;END;
IF KR=0 & PRIZ=0 THENDO I=1 TO NSOR;
IF TRM(I)=C THEN DO J=1 TO NP-1;ST(I,J)=ST(I,J)/KF;END;
END;  MC001:SOS=0;V=1;
DO J=N2 TO N;    /*--ВЫЧИСЛЕНИЕ СТОИМОСТИ КУСТА--------*/
 JIJ=MTR(J,1);IF A1(J)=0 THEN A1(J)=MTR(J,ND);
J1=A1(J); J2=MTR(J,ND); J4 =V(J2);J41=V(J1);
IF TRR(J)>0 THEN DO;STM=STT(J)*DLIN(J);GOTO MSOS;END;
DO JJ=1 TO ND1 WHILE(AB(JIJ,JJ)^=0);
 IF J1=AB(JIJ,JJ) THEN K1=JJ;
 IF J2=AB(JIJ,JJ) THEN K2=JJ;END;
 P1=POTER(JIJ,K1);P=POTER(JIJ,K2);
IF J2=J1 THEN DO; X=0; GO TO M4000; END;
 X=(P1-PN(JIJ))*DLIN(JIJ)/(P1-P);
  M4000:IF TRM(J2)=C THEN DO;DO LL=J4 TO NP-1;
        IF HAP(I5)<DVL(J2,LL) THEN DO;
        P3=LL;GOTO MM999;END;END; END; ELSE P3=1;
  MM999: IF TRM(J1)=C THEN DO; P1=-P*X/DLIN(JIJ);
        DO LL=J41 TO NP-1; PP=HAP(J)+P1;
        IF PP<DVL(J1,LL) THEN DO; P4=LL;GO TO MM500;END;END;
        END;ELSE P4=1; MM500: XL=DLIN(JIJ)-X;
        STM=ST(J2,P3)*X+ST(J1,P4)*XL;
  MSOS:SOS=SOS+STM; END; SOS=SOS/1000;
  IF KR=0 & PRIZ^=0 THEN DO I=1 TO NSOR;
  IF TRM(I)=C THEN    DO J=1 BY 1 WHILE(ST(I,J)^=0);
  ST(I,J)=ST(I,J)/KF;END;END; SOS1=0;RTN1='PTH';
```

```
  IF PRIZ=0 THEN CALL PECHAT(N,MTR,A1,NDT, PN,HAP);
 DO I=1 TO N;  MTR(I,ND)=MTR(I,ND-1);MTR(I,ND-1)=0;END;
KONOP:RETURN(SOS); END OPTIMM;
/*****************************************************************/
/*        Подпрограмма печати выходных таблиц                    */
/*****************************************************************/
 PECHAT:PROCEDURE(N,MTR,A1,NDT, PN,HAP);
    DCL N, MTR(100,30)FIXED DEC(3),PN(300),HAP(100),
   1 STRUCT2 EXT,2 N1,2 N2,2 NSOR,2 NP,2 SN,2 VK,2 VMAX,
    2 TRC,2 TCH,2 TG,2 TA,2 RTN,2 POLI,2 DVLG,2 DVCH,
    2 DVLR,2 DVLA,2 DVLP,2 RR,2 KR,
 1 STRUCT1 EXT,2 TR(100,10),2 ST(100,10),2 DVL(100,10),
2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
2 GOST(100) CHAR(10),2 MT(100) CHAR(10),
2 TRM(100) CHAR(1),2 AB(300,50) FIXED DEC(3),
2 POTER(300,50),2 V(100) FIXED DEC(1),
2 HOPCTH(10,10), 1 STRUCT(300) EXT,2  DLIN, 2 VIS, 2 RASH,
2 HSB, 2 MATER FIXED DEC(2),2 MPN FIXED DEC(1),
2 OGRH,  NDT FLOAT DECIMAL,(POLI1,C,G,A,CH,RTN1)CHAR(5)EXT,
(TRR,STT,KODD FIXED(4))(300) EXT,
A1(100) FIXED DEC(2), MATTR CHAR(5);ND=30; KNK=1;
   C0001:DO J=2 TO N;KVER1=MTR(J,1);KVER2=MTR(J,2);
IF TRR(KVER1)>0 THEN DO;A1(J)=J;
MTR(J,ND)=J;X1=DLIN(KVER1);P1=0;P2=PN(KVER1);GOTO ;M990;END;
J1=A1(J);J2=MTR(J,ND);
   C0002:DO JJ=1 BY 1 WHILE(AB(KVER1,JJ)^=0);
         IF J2=AB(KVER1,JJ) THEN K2=JJ;
         IF J1=AB(KVER1,JJ) THEN K1=JJ;END C0002;
         P1=POTER(KVER1,K1);P2=POTER(KVER1,K2);P=P2;
```

```
        IF J2=J1 THEN X1=0;    /*X-Д ИНА D1*/
        ELSE X1=(P1-PN(KVER1))*DLIN(KVER1)/(P1-P);
   M990:PN1=1000*P2/DLIN(KVER1);
        DO I=1 TO N WHILE(MTR(I,1)^=KVER2);END;I5=I;
        K4=1;K3=1;IF TRR(KVER1)=0&TRM(J2)=C THEN DO;
        DO LL=1 BY 1 WHILE(DVL(J2,LL)^=0);
       IF HAP(I5)<DVL(J2,LL) THEN DO;
       K3=LL;GOTO MM990;END;END;  END; ELSE K3=1;
  MM990:IF TRR(KVER1)=0&TRM(J1)=C THEN DO;
       P6=-P*X1/DLIN(KVER1);
       DO LL=1 BY 1 WHILE(DVL(J1,LL)^=0); PP=HAP(J)+P6;
       IF PP<DVL(J1,LL) THEN DO;K4=LL;GO TO MM700;END;END;
 END; ELSE K4=1;  MM700:X2=DLIN(KVER1)-X1;/*X2-ДЛИНА Д2*/
PN2=1000*P1/DLIN(KVER1);  /*PN2-ПОТЕРИ В ТР D2*/
IF TRR(KVER1)=0 THEN PTR1=TR(J1,K4);ELSE PTR1=TRR(KVER1);
V2=4*RASH(KVER1)/(3.14*PTR1**2);
IF TRR(KVER1)=0 THEN PTR2=TR(J2,K3);ELSE PTR2=TRR(KVER1);
V1=4*RASH(KVER1)/(3.14*PTR2**2);
IF X2=0 THEN V2=0;IF X1=0 THEN V1=0;
RAZN=VIS(KVER1)-VIS(KVER2);/*RAZN-РАЗНОСТЬ ОТМЕТОК*/
IF TRR(KVER1)>0 THEN DO;PST1=0;PST2=STT(KVER1);END;
ELSE DO;PST1=ST(J1,K3);PST2=ST(J2,K4);END;
/* STM=X1*ST(J2,K3)+X2*ST(J1,K4);*/
STM=X1*PST2+X2*PST1;
IF KNK=1 THEN   DG001:DO; IF J=2 THEN DO;
PUT EDIT('*',(52)'-','ЛИНИЯ ОТРЕЗА',(52)'-','*')(SKIP,5 A);
  PUT EDIT('ТАБЛИЦА  ВЫХОДНЫХ  ДАННЫХ  ПО  '
'ГИДРАВЛИЧЕСК')(SKIP(3), X(77),A);
 PUT EDIT((120)'-')(SKIP(2),A);
```

```
  PUT EDIT(':          : ДЛИНА   : РАСХОД  : '
'ОТМЕТКА : ОТМЕТКА :ГЕОМЕТР. :   ДИАМЕТР :МАТЕРИАЛ : Г О С Т :
ДЛИНА  : ','СКОРОСТЬ:  ПОТЕРИ:')(SKIP,A);
  PUT EDIT(':          :          :  ВОДЫ   :'
'ПОВЕРХНОС:ПОВЕРХНОС: РАЗНОСТЬ:ВОДОВОДА :  ТРУБ   :  ТРУБЫ
: УЧАСТКА : ','ПОТОКА  : НАПОРА :')(SKIP,A);
  PUT EDIT(': УЧАСТОК : УЧАСТКА :   НА    :','ТИ ЗЕМЛИ :ТИ
ЗЕМЛИ : ОТМЕТОК :     Д1    :    НА   :   Д1    : С Д1   :
В',' ТРУБЕ : В ТРУБЕ:')(SKIP,A);
 PUT EDIT(':          :          : УЧАСТКЕ :','В НАЧАЛЕ :В
КОНЦЕ  : КОНЦОВ  :            : УЧАСТКЕ :          :          :
  Д1    :   Д1   :')(SKIP,A);
  PUT EDIT(':          :  ( М )  :КУБ.М/СЕК: УЧАСТКА :
УЧАСТКА : УЧАСТКА :     (М)   :   Д1    :          :    (М)
:  М/СЕК  :(1000I) :')(SKIP,A);
 PUT EDIT(':          :          :          :  ( М )  : ( М )
:  ( М )  :            :          :          :          : '
'        :         :')(SKIP,A); PUT EDIT((120)'-')(SKIP,A);
  PUT EDIT(':    1    :    2    :    3    :    4    :    5
:    6    :    7    :    8    :    9    :   10    :   11
'  :   12   :')(SKIP,A);  PUT EDIT((120)'-')(SKIP,A);END;
PN1=PN1/1.1;IF X1<=0.05 THEN DO;V1=0;PN1=0;PTR2=0; END;
  IF TRR(KVER1)>0 THEN J2=NSOR+1;
IF TRM(J2)='P' THEN MATTR=RTN1;ELSE MATTR=TRM(J2);
PUT EDIT(':',KVER2,'-',KVER1,':',DLIN(KVER1),':',
RASH(KVER1),':',VIS(KVER2),':',VIS(KVER1),':',RAZN,':',
PTR2,':',MATTR,',',MT(J2),':',GOST(J2),':', X1,':',V1,
':',PN1,':')(SKIP,A,F(4),A,F(3),X(1),A,F(9,3),A, F(8,5),
X(1),A, X(2),F(6,2),X(1),A,X(2),F(6,2),X(1),A,X(2),F(6,2),
```

```
X(1),A,X(1),F(6,3),X(2),A,A(3),A,A(5),A,A(9),A,X(1),F(8,2),
A,X(1),F(6,3),X(2),A,F(8,3),A);  PUT EDIT((120)'-')(SKIP,A);
 END DG001; ELSE DG002:DO; IF J=2 THEN DO;
  PUT SKIP(3);PUT SKIP(3);
  PUT EDIT('*',(52)'-','ЛИНИЯ ОТРЕЗА',(52)'-','*')
(SKIP,5 A);
  PUT EDIT('ОМУ  РАСЧЕТУ  ВОДОВОДОВ')(SKIP(3),A);
  PUT EDIT((110)'-')(SKIP(2),A);
  PUT EDIT(  ':          : ДИАМЕТР : МАТЕРИАЛ: Г О С Т : ДЛИНА
 : СКОРОСТЬ: ПОТЕРИ  :   ПОТЕРИ :  ПОТЕРИ
:СВОБОДНЫЙ:СТОИМОСТЬ:',:          : ВОДОВОДА:ТРУБ  НА : ТРУБЫ
: УЧАСТКА : ПОТОКА  : НАПОРА  :   НАПОРА  :  НАПОРА :  НАПОР
:          :УЧАСТОК :   Д2    : УЧАСТКЕ :   Д2    :С ДИАМЕТР:
   НА    : В ТРУБЕ :    НА    :С УЧЕТОМ : В КОНЦЕ : УЧАСТКА
:         :   (М)  :   Д2    :         :   Д2    : УЧАСТКЕ
:   Д2    :   УЧАСТКЕ :СОПРОТИВ-: УЧАСТКА :         :',
':        :         :         :         :         :   Д2
: (1000I) :           : ЛЕНИИ   :  ( М )  : ( Т.РУБ):',
':        :         :         :         :       '
'  :  М/СЕК  :         :         :         :         :
:')(6(SKIP, A));  PUT EDIT((110)'-')(SKIP,A);
  PUT EDIT(  '   13    :   14    :   15    :   16    :   17
'  :   18    :   19    :    20    :   21    :   22    :
23    :')(SKIP,A);  PUT EDIT((110)'-')(SKIP,A);
   END; PN2=PN2/1.1;IF X2<=0.05 THEN DO;V2=0;PN2=0;END;
   PN3=PN(KVER1)/1.1; STM=STM/1000;IF TRR(KVER1)>0 THEN
J1=NSOR+1;IF TRM(J1)='P' THEN MATTR=RTN1;ELSE MATTR=TRM(J1);
PUT EDIT(':',KVER2,'-',KVER1,':',PTR1      ,':',
MATTR,',',MT(J1),':',GOST(J1),':',X2,':',V2,':',PN2,':'
```

```
,PN3,':' ,PN(KVER1),':', HAP(J),':',STM,':')
(SKIP,A,F(3),A,F(3),X(1),A,X(1),F(6,3),X(2),A,
A(3),A,A(5),A, A(9),A,F(9,2),A,X(1),F(6,3),X(2),A,
F(8,4),X(1),A,F(8,3),X(1),A,F(8,3),X(1),A,F(8,2),X(1),A,
F(8,3),X(1),A);  PUT EDIT((110)'-')(SKIP,A);END DG002;
END  C0001; IF KNK=1 THEN DO;KNK=0;GOTO C0001;END;
  PUT EDIT('НАПОР НА НАСОСНОЙ СТАНЦИИ РАВЕН',NDT)
(SKIP,X(65),A,F(9,3));  END  РЕСНАТ;
/*****************************************************************/
/*   Подпрограмма построения матрицы траектории кустов   */
/*****************************************************************/
  PMTRK:PROCEDURE(NUT,MTR, KW,N,MTRK);
 DCL( MTR(300,50),MTRK(100,30)) FIXED DEC(3),(N,NUT,KW);
          ND=50;MTRK=0; I1=MTR(NUT,2);MTRK(1,1)=I1;
          MTRK(2,1)=NUT;MTRK(2,2)=I1;  N=2;
   C0001:DO M1=1 TO KW;IF MTR(M1,ND-2)=0 THEN D0001:DO;
  C0002:DO M2=2 TO ND-3 WHILE(MTR(M1,M2)^=0);
          IF MTR(M1,M2)=NUT THEN D0002:DO; N=N+1;KW1=M2+1;
   IF N>100 THEN DO;
PUT SKIP LIST('ПРИМЕР НЕ МОЖЕТ БЫТЬ  ПРОСЧИТАН, ТАК
   КАК КОЛИЧЕСТВО УЧАСТКОВ НА КУСТЕ БОЛЬШЕ ЧЕМ '
'ДОПУСТИМО ПО ПРОГРАММЕ-100  '); P=0;R=1;R=R/P;END;
  C0003:DO M3=1 TO KW1;
          MTRK(N,M3)=MTR(M1,M3); END C0003;
 END D0002;  END C0002;  END D0001; END C0001;
          IF MTRK(1,1)=MTR(1,1) THEN D0003:DO;
  C0004:DO I=1 TO KW;MTR(I,ND-2)=0;END C0004;END D0003;
          IF MTRK(1,1)^=MTR(1,1) THEN D0004:DO;
  C0005:DO I=2 TO N;I2=MTRK(I,1);
```

```
      MTR(I2,ND-2)=I2;END C0005; END D0004;   END PMTRK;
/*****************************************************************/
/* Подпрограмма построения первоначального решения куста*/
/*****************************************************************/
    PROV:PROCEDURE(MTR,HSB1,N);
     DCL HDT,
        1 STRUCT2 EXT, 2 N1,2 N2,2 NSOR,2 NP,2 SN,2 VK,2 VMAX,
         2 TRC,2 TCH,2 TG,2 TA,2 RTN,2 POLI,
         2 DVLG,2 DVCH,2 DVLR,2 DVLA,2 DVLP,2 RR,2 KR,
      1 STRUCT1 EXT, 2 TR(100,10),2 ST(100,10),2 DVL(100,10),
        2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
        2 GOST(100) CHAR(10),2 MT(100) CHAR(10),
        2 TRM(100) CHAR(1), 2 AB(300,50) FIXED DEC(3),
        2 POTER(300,50),2 V(100) FIXED DEC(1),2 HOPCTH(10,10),
        NDT FLOAT DEC,
      1 STRUCT(300) EXT,2 DLIN, 2 VIS, 2 RASH, 2 HSB,
        2 MATER FIXED DEC(2),2 MPN FIXED DEC(1),2 OGRH,
      (C,G,CH,A,RTN1,POLI1)CHAR(5) EXT,
 (HSB1(100), PN(300), HAP(100)), MTR(100,30) FIXED DEC(3);
N1=1;N2=2;RTN1='P'; ND=30;ND1=50;
PN=0;POTN=0;HAP=0; DVLCH=DVCH;
   C0001:DO I=N2 TO N; I1=MTR(I,1);
       DO J=1 TO ND1-2 WHILE(POTER(I1,J)^=0); END;
         PN(I1)=POTER(I1,J-1);MTR(I,ND)=AB(I1,J-1);
         END C0001; /*ВЫЧИСЛЕНИЕ МИН.ПОТР.НАПОРОВ*/
         KK1=MTR(1,1);HMAX=0;MNOM=0;
   C0003:DO I=N2 TO N; I1=MTR(I,1); SP=0;
   C0004:DO J=N1 TO ND-2 WHILE(MTR(I,J)^=KK1);
         K1=MTR(I,J);SP=SP+PN(K1); END C0004;
```

```
        POTN   =HSB1(I)+VIS(I1)-VIS(KK1)+SP;
        IF HMAX<=POTN THEN DO;HMAX=POTN;MNOM=I;END;
        END C0003; NDT=HMAX; V=1;
   C0006:DO I=N2 TO N;SP=0; I1=MTR(I,1);
   C0007:DO J=1 TO ND-2 WHILE(MTR(I,J)^=KK1);
        K1=MTR(I,J);SP=SP+PN(K1); END C0007;
        HAP(I)=NDT-SP+VIS(KK1)-VIS(I1);
        END C0006; NETR=0;
   C0009:DO I=N2 TO N; K=0;I1=MTR(I,1);
   C0010:DO J=1 TO ND1 WHILE(AB(I1,J)^=0);
K=K+1;END C0010;
/*K-КОЛИЧЕСТВО ТРУб ИЗ СОРТАМЕНТА ДЛЯ УЧАСТКА I1*/
I40=MTR(I,2); /*I40-НОМЕР ВЕРШИНЫ */
C0011:DO I30=1 TO N WHILE(MTR(I30,1)^=I40);END C0011;
I4=I30;/*ИНДЕКС ВЕРШИНЫ СООТВЕТСТВУюШЕЕ I40 */
J1=MTR(I,ND);V(J1)=1;IK=J1;
   MMM10: K1=K ; IF TRM(J1)^=C THEN GOTO MMM7;
        DO J=1 TO NP-1;
        IF TR(J1,J)^=0 THEN IF HAP(I4)<DVL(J1,J) THEN DO;
        JL=J;GOTO MM222;END;END;
        IF K>1 THEN DO;K=K-1;IK=AB(I1,K);
        J1=IK; GO TO MMM10; END; ELSE NETR=1;
  MMM7: IF TRM(J1)=A&HAP(I4)>DVLA THEN
   D0001: DO; JL=1; K1=K-1;
  MM111: IK=AB(I1,K1); IF TRM(IK)=RTN1&HAP(I4)<=DVLR THEN
GO TO MM222; IF TRM(IK)=G THEN IF HAP(I4)<DVLG THEN GOTO
MM222; IF TRM(IK)=CH THEN IF HAP(I4)<DVLCH THEN GOTO MM222;
IF TRM(IK)=C THEN  DO JL=1 BY 1 WHILE(DVL(IK,JL)^=0);
        IF HAP(I4)<=DVL(IK,JL) THEN GO TO MM222; END;
```

```
      IF K1>1 THEN DO;K1=K1-1;GOTO MM111;END;
      NETR=1; GOTO MM200; END DO001;
MMM8: IF TRM(J1)=G & HAP(I4)>DVLG THEN
 DO002: DO; JL=1; K1=K-1;
MM110:  IK=AB(I1,K1);
 IF TRM(IK)=CH THEN IF HAP(I4)<DVLCH THEN GO TO MM222;
 IF TRM(IK)=C THEN DO JL=1 BY 1 WHILE(DVL(IK,JL) ^=0);
 IF HAP(I4)<=DVL(IK,JL) THEN GO TO MM222;  END;
      IF K1>1 THEN DO; K1=K1-1;GOTO MM110;END;
      NETR=1; GOTO MM200;  END DO002;
MMM9: IF TRM(J1)=RTN1 & HAP(I4)>DVLR THEN
 DO003: DO;  JL=1;  K1=K-1;
MM122:  IK=AB(I1,K1);
   IF TRM(IK)=CH THEN IF HAP(I4)<DVLCH THEN GOTO MM222;
   IF TRM(IK)=C THEN DO JL=1 BY 1 WHILE(DVL(IK,JL)^=0);
      IF HAP(I4)<=DVL(IK,JL) THEN GO TO MM222; END;
      IF K1>1 THEN DO; K1=K1-1;GOTO MM122;END;
      NETR=1; GO TO MM200;   END DO003;
      IF TRM(J1)=CH & HAP(I4)>DVLCH THEN
 DO004: DO; JL=1; K1=K-1;
MM109: IK=AB(I1,K1);
      IF TRM(IK)=C THEN
      DO JL=1 BY 1 WHILE(DVL(IK,JL)^=0);
      IF HAP(I4)<=DVL(IK,JL) THEN GO TO MM222;END;
      IF K1>1 THEN DO;K1=K1-1;GOTO MM109;END;
      NETR=1; GOTO MM200;
 END DO004; MM222:MTR(I,ND)=IK;PN(I)=POTER(I1,K1); V(IK)=JL;
MM200: END CO009; NDT=0;MNOM=0;
 CO012:DO I=N2 TO N; SP=0; I1=MTR(I,1);
```

```
   C0013:DO J=1 TO ND-1 WHILE(MTR(I,J)^=KK1);
         K1=MTR(I,J); SP=SP+PN(K1); END C0013;
         POTN   =HSB1(I)+VIS(I1)-VIS(KK1)+SP;
         IF  NDT<=POTN THEN DO; NDT=POTN;MNOM=I;END;
         END C0012;
IF NETR=1 THEN HDT=1000;IF NETR=1 THEN NDT=10**10;
         RETURN(NDT);  END PROV;
/*****************************************************************/
/* Подпрограмма построения оптимальных точек на вертикалях*/
/*****************************************************************/
   VIBORI:PROCEDURE(HSB1,MTRK,MSV, KNST, PCB1,
                    DELH,KIGL,IGL, MTR,ITN,KS,CTV);
 DCL ( NDT, HNS,HDT, DELH) FLOAT DEC, PCB1 POINTER,
(N,KW,KS), KTV EXT, (CTVV1,CTVV2)EXT,
1 STRUCT1 EXT,2 TR(100,10),2 ST(100,10),2 DVL(100,10),
2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
2 GOST(100) CHAR(10),2 MT(100) CHAR(10),2 TRM(100) CHAR(1),
2 AB(300,50) FIXED DEC(3),2 POTER(300,50),2 V(100) FIXED
DEC(1), 2 HOPCTH(10,10),  ( MTR(300,50),
MSV(20,2),MTRK(100,30),MTRN(50,10),ITN(50,2),NIP(10))FIXED
DEC(3),CTV(500),HSB1(100),
1 STRUCT(300) EXT, 2 DLIN,2 VIS,2 RASH,2 HSB,
 2 MATER FIXED DEC(2),2 MPN FIXED DEC(1),2 OGRH ;
DO I=1 BY 1 WHILE(MTR(I,1)^=0);KW=I;END;
DO I=1 BY 1 WHILE(MTRK(I,1)^=0);N=I;END;ND=50; II=IGL; NKS=50;
NKS=50;NIP=KTV;/*NIP-ПАРАМЕТРЫ КОНЦА ЦИКЛОВ*/KQ=0;KT1=9;
KN1=0;KN2=0;KN3=0;KN4=0;KN5=0;KN6=0;KN7=0;KN8=0;KN9=0;
CTV1=0;CTV2=0;CTV3=0;CTV4=0;CTV5=0;CTV6=0;
CTV7=0;CTV8=0;CTV9=0;/*3 ПЕРЕКАЧКА,2-ПОДКАЧКА,1-ОБЕ*/
```

```
KTS1=KTV;NUT=MTRK(2,1);KV=MTRK(1,1);IF KV=1 THEN HSB(KV)=0;
KST=MPN(NUT);IF KST=3 THEN KST=1;
IL1=0;IL2=0;IL3=0;IL4=0;IL5=0;IL6=0;IL7=0;IL8=0;IL9=0;
I6=0;I7=0;I8=0;I9=0;I1=0;I2=0;I3=0;I4=0;I5=0;
KS1=KS-8; /*КОЛИЧ.ВЕРТИК.=9*/
IF KS1>0 THEN DO;KL1=MSV(KS1,2);IL1=ITN(KL1,2);IL1=MTR(IL1,2);
DO I1=1 TO N WHILE(MTRK(I1,1)^=IL1);END;
END;KS2=KS-7;/*KS2-КОЛ-ВО ВЕРТ.8*/
IF KS2>0 THEN DO;IL2=ITN(KL2,2);IL2=MTR(IL2,2);
DO I2=1 TO N WHILE(MTRK(I2,1)^=IL2);END;END;KS3=KS-6;
IF KS3>0 THEN DO;KL3=MSV(KS3,2);IL3=ITN(KL3,2);IL3=MTR(IL3,2);
DO I3=1 TO N WHILE(MTRK(I3,1)^=IL3);END;END;KS4=KS-5;
IF  KS4>0 THEN DO;KL4=MSV(KS4,2);IL4=ITN(KL4,2);IL4=MTR(IL4,2);
DO I4=1 TO N WHILE(MTRK(I4,1)^=IL4);END;END;KS5=KS-4;
IF KS5>0 THEN DO;KL5=MSV(KS5,2);IL5=ITN(KL5,2);IL5=MTR(IL5,2);
DO I5=1 TO N WHILE(MTRK(I5,1)^=IL5);END;END;KS6=KS-3;
IF KS6>0 THEN DO;KL6=MSV(KS6,2);IL6=ITN(KL6,2);IL6=MTR(IL6,2);
DO I6=1 TO N WHILE(MTRK(I6,1)^=IL6);END;END;KS7=KS-2;
IF KS7>0 THEN DO;KL7=MSV(KS7,2);IL7=ITN(KL7,2);IL7=MTR(IL7,2);
DO I7=1 TO N WHILE(MTRK(I7,1)^=IL7);END;END;KS8=KS-1;
IF KS8>0 THEN DO;KL8=MSV(KS8,2);IL8=ITN(KL8,2);IL8=MTR(IL8,2);
DO I8=1 TO N WHILE(MTRK(I8,1)^=IL8);END;END;KS9=KS;
IF KS9>0 THEN DO;KL9=MSV(KS9,2);IL9=ITN(KL9,2);IL9=MTR(IL9,2);
DO I9=1 TO N WHILE(MTRK(I9,1)^=IL9);END;END;
NI=1;NJ=1;NK=1;NM=1;NL=1;NG=1;NE=1;NT=1;NC=1;
        IF KS1<0 THEN NIP(1)=1;
   CC01:DO NI1=1 TO NIP(1);
IF KS1>0 THEN DO;KL11=ITN(KL1,2);DELH=POTER(KL11,NKS);
IF IL1^=0 THEN HSB1(I1)=OGRH(KL11)-(KTV-NI1)*DELH;
```

```
KN1=KL1*KTV+NI-1;CTV1=CTV(KN1);END;
IF IL2=IL1 THEN DO;NJ=NI1;NIP(2)=NJ;END;
  CC02:DO NJ1=NJ TO NIP(2);
       IF KS2>0 THEN DO;
       KL21=ITN(KL2,2);DELH=POTER(KL21,NKS);
       IF IL2^=IL1 THEN HSB1(I2)=OGRH(KL21)-(KTV-NJ1)*DELH;
       KN2=KL2*KTV+NJ1-1;  CTV2=CTV(KN2);  END;
  IF IL3=IL2 THEN NK=NJ1;IF IL3=IL1 THEN NK=NI1;
  IF IL3=IL2!IL3=IL1 THEN NIP(3)=NK;
  CC03:DO NK1=NK TO NIP(3);
    IF KS3>0 THEN DO; KL31=ITN(KL3,2);DELH=POTER(KL31,NKS);
    IF IL3^=IL2 & IL3^=IL1 THEN
    HSB1(I3)=OGRH(KL31)-(KTV-NK1)*DELH;
KN3=KL3*KTV+NK1-1;CTV3=CTV(KN3);END;
IF IL4=IL3 THEN NM=NK1;IF IL4=IL2 THEN NM=NJ1;
IF IL4=IL1 THEN NM=NI1;
IF IL4=IL3!IL4=IL2!IL4=IL1 THEN NIP(4)=NM;
CC04:DO NM1=NM TO NIP(4);
IF KS4>0 THEN DO;KL41=ITN(KL4,2);DELH=POTER(KL41,NKS);
IF IL4^=IL3 & IL4^=IL2 & IL4^=IL1 THEN
 HSB1(I4)=OGRH(KL41)-(KTV-NM1)*DELH;
KN4=KL4*KTV+(NM1-1);CTV4=CTV(KN4);END;
IF IL5=IL4 THEN NL=NM1;IF IL5=IL3 THEN NL=NK1;
IF IL5=IL2 THEN NL=NJ1;IF IL5=IL1 THEN NL=NI1;
IF IL5=IL4!IL5=IL3!IL5=IL2!IL5=IL1 THEN NIP(5)=NL;
    CC05:DO NL1=NL TO NIP(5);
IF KS5>0 THEN DO;KL51=ITN(KL5,2);DELH=POTER(KL51,NKS);
IF IL5^=IL4&IL5^=IL3 & IL5^=IL2 & IL5^=IL1 THEN
HSB1(I5)=OGRH(KL51)-(KTV-NL1)*DELH;
```

```
KN5=KL5*KTV+NL1-1;CTV5=CTV(KN5);END;
IF IL6=IL5 THEN NG=NL1;IF IL6=IL4 THEN NG=NM1;
IF IL6=IL3 THEN NG=NK1;IF IL6=IL2 THEN NG=NJ1;
IF IL6=IL1 THEN NG=NI1;
IF IL6=IL5!IL6=IL4!IL6=IL3!IL6=IL2!IL6=IL1 THEN NIP(6)=NG;
    CC06:DO NG1=NG TO NIP(6);
IF KS6>0 THEN DO;
KL61=ITN(KL6,2);DELH=POTER(KL61,NKS);
IF IL6^=IL5&IL6^=IL4 THEN
IF IL6^=IL3 & IL6^=IL2 & IL6^=IL1 THEN
HSB1(I6)=OGRH(KL61)-(KTV-NG1)*DELH;
KN6=KL6*KTV+NG1-1;CTV6=CTV(KN6);END;
IF IL7=IL6 THEN NE=NG1;IF IL7=IL5 THEN NE=NL1;
IF IL7=IL4 THEN NE=NM1;IF IL7=IL3 THEN NE=NK1;
IF IL7=IL2 THEN NE=NJ1;IF IL7=IL1 THEN NE=NI1;
IF IL7=IL6!IL7=IL5!IL7=IL4!IL7=IL3!IL7=IL2!IL7=IL1
THEN NIP(7)=NE;
  CC07:DO NE1=NE TO NIP(7);
    IF KS7>0 THEN DO;KL71=ITN(KL7,2);DELH=POTER(KL71,NKS);
IF IL7^=IL6&IL7^=IL5&IL7^=IL4&IL7^=IL3&IL7^=IL2&IL7^=IL1
THEN HSB1(I7)=OGRH(KL71)-(KTV-NE1)*DELH;
KN7=KL7*KTV+NE1-1;CTV7=CTV(KN7);END;
IF IL8=IL7 THEN NT=NE1;IF IL8=IL6 THEN NT=NG1;
IF IL8=IL5 THEN NT=NL1;IF IL8=IL4 THEN NT=NM1;
IF IL8=IL3 THEN NT=NK1;IF IL8=IL2 THEN NT=NJ1;
IF IL8=IL1 THEN NT=NI1;
IF IL8=IL7!IL8=IL6!IL8=IL5!IL8=IL4!
IL8=IL3!IL8=IL2!IL8=IL1 THEN NIP(8)=NT;
    CC08:DO NT1=NT TO NIP(8);IF KS8>0 THEN DO;
```

```
KL81=ITN(KL8,2);DELH=POTER(KL81,NKS);
IF IL8^=IL7&IL8^=IL6&IL8^=IL5&IL8^=IL4 THEN
IF IL8^=IL3 & IL8^=IL2 & IL8^=IL1 THEN
HSB1(I8)=OGRH(KL81)-(KTV-NT1)*DELH;
KN8=KL8*KTV+NT1-1;CTV8=CTV(KN8);END;
IF IL9=IL8 THEN NC=NT1;IF IL9=IL7 THEN NC=NE1;
IF IL9=IL6 THEN NC=NG1;IF IL9=IL5 THEN NC=NL1;
IF IL9=IL4 THEN NC=NM1;IF IL9=IL3 THEN NC=NK1;
IF IL9=IL2 THEN NC=NJ1;IF IL9=IL1 THEN NC=NI1;
IF IL9=IL8!IL9=IL7!IL9=IL6!IL9=IL5 THEN NIP(9)=NC;
IF IL9=IL4!IL9=IL3!IL9=IL2!IL9=IL1 THEN NIP(9)=NC;
    CC09:DO NC1=NC TO NIP(9);
IF KS9>0 THEN DO;
KL91=ITN(KL9,2);DELH=POTER(KL91,NKS);
IF IL9^=IL8&IL9^=IL7&IL9^=IL6&IL9^=IL5&IL9^=IL4
THEN IF IL9^=IL3&IL9^=IL2&IL9^=IL1 THEN
HSB1(I9)=OGRH(KL91)-(KTV-NC1)*DELH;KN9=KL9*KTV+NC1-1;
 CTV9=CTV(KN9); END;
M0000: CTV11=CTV1+CTV2+CTV3+CTV4+CTV5+CTV6+CTV7+CTV8+CTV9;
       DELH=POTER(NUT,NKS); NDT=PROV(MTRK,HSB1,N);
    C0009:DO I=KTV TO KST BY -1; KSN=II*KTV+I-1;
          IF MPN(NUT)=3 THEN KTS1=I; IF I=1 THEN KTS1=KTV;
          IF MPN(NUT)^=3 THEN HDT=HSB(KV)+(I-1)*DELH;
          IF MPN(NUT)=3 THEN HDT=OGRH(NUT)-(KTV-I)*DELH;
          PRIZ=1; IF NDT<=HDT&HDT<=OGRH(NUT) THEN
          SOS=OPTIMM(HDT,PRIZ,HSB1, N,MTRK);ELSE SOS=10**20;
   C0010:DO J=I TO KTS1;
          KSS=II*KTV+J-1;HNS=(J-I)*DELH;RASH1=RASH(NUT);
/*--РАСХОД НА ВЕРШИНЕ КУСТА-----------------*/
```

```
IF MPN(NUT)=4 THEN ITIP=1;ELSE ITIP=2;
IF HNS^=0 THEN STNS=OGRH(KW+3)+RASH1*OGRH(KW+2)*HNS;
/*--СТОИМ.Н.С.И ЭНЕРГЕТ.*//*STNS=STOIMN(HNS,RASH1,ITIP);*/
 ELSE STNS=0;
IF HNS=0 THEN PRIV=SOS*OGRH(KW+1)+CTV11;
IF HNS>0 THEN PRIV=STNS+CTV(KSS);
PP=CTV(KSN);PRR=CTV(KSS);
          IF PRIV<CTV(KSN) THEN  DG017:DO;
          CTV(KSN)=PRIV; IF I=J THEN IND(KSN,1,1)=KSN;
          ELSE IND(KSN,1,1)=IND(KSS,1,1);
          IF KS>0 THEN  DG018:DO;
          IF KSN=KSS THEN  DG019:DO;
         IND(KSN,3,1)=KN8;IND(KSN,2,1)=KN9;
         IND(KSN,4,1)=KN7;IND(KSN,5,1)=KN6;IND(KSN,6,1)=KN5;
         IND(KSN,7,1)=KN4;IND(KSN,8,1)=KN3;IND(KSN,9,1)=KN2;
         IND(KSN,10,1)=KN1;END DG019;
         ELSE  CO011:DO KJ=2 BY 1
        WHILE(IND(KSS,KJ,1)^=0);IND(KSN,KJ,1)=IND(KSS,KJ,1);
         END CO011; END DG018;END DG017;END CO010;
END CO009;END CC09;
END CC08;END CC07;END CC06;END CC05;END CC04;END CC03;
          END CC02; END CC01;MTR(NUT,ND)=MPN(NUT);
  K1=(II+1)*KTV-1; IF IGL=KNST THEN  MD000:DO;
          /*KII-ПОРЯДКОВ.НОМЕР ТОЧКИ НА ПОСЛЕД.ВЕРТИК.*/
          KII=II*KTV; CTVV1=CTV(KII);
          IF KIGL=1&CTVV1>=10**10 THEN GOTO NCON;
          MOV(II)=KII;  /*БЛОК ПЕРЕПРИСВ.MPN*/
    CO013:DO I=1 TO KW;  MPN(I)=MTR(I,ND); END CO013;
          CALL MATRNS(MTR,ITN,II,MTRN);
```

```
        /*CTVV2-СТОИМОСТЬ СЕТИ ПРИ ПРЕДШЕСТВ.DELH*/
        /*CTVV1-СТОИМОСТЬ СЕТИ ПРИ ДАННОМ  DELH*/
        IF KIGL=1 THEN CTVV2=10**10;  RAZS=CTVV2-CTVV1;
        IF RAZS>5 THEN  DD000:DO;  DELH1=DELH;
      IF CTVV2>CTVV1 THEN DO;CTVV2=CTVV1;IKN=(II+1)*KTV-1;
   C0017:DO I=KTV TO IKN;
   C0018:DO J=1 BY 1 WHILE(IND(I,J,1)^=0);
        IND(I,J,2)=IND(I,J,1);END C0018; END C0017;
        DO I=1 TO KW;POTER(I,NKS-1)=POTER(I,NKS);END;END; KH=0;
     CALL CTOPUT(II,DELH,MTRN, KH,ITN,MTR);
        END DD000;  ELSE
   DD001:DO;  /*БЛОК ФИКСИРОВАНИЯ МЕСТОПОЛ.Н.С.*/
        IF CTVV2<CTVV1 THEN DELH=DELH1;
        IF CTVV2<CTVV1 THEN DO;
        DO I=1 TO KW;POTER(I,NKS)=POTER(I,NKS-1);END;END;KH=1;
   DO I=1 TO K1;   DO J=1 BY 1 WHILE(IND(I,J,2)^=0);
    IND(I,J,1)=IND(I,J,2);END;END;
        CALL CTOPUT(II,DELH,MTRN, KH,ITN,MTR);
       DCL MTRKZ(100,30) FIXED DEC(3);
/*ДЛЯ СОВМЕШ. 2-Х Н.С. С ОДНОЙ(.)*/
  GLAVC:DO KKK=0 BY 1 WHILE(KKK<2);
/*ГЛАВНЫЙ cИКЛ ПЕРЕРАСЧТА КУСТОВ   */CTVV4=0;MTRK=0;MTRKZ=0;
   C0021:DO I=1 TO II;K=MTRN(I,1);
KJ1=MOV(K);  /*KJ1-ТОЧКА К-ОЙ ВЕРТИКАЛИ*/
 IF CTVV2<CTVV1 THEN DO;KJ2=IND(KJ1,1,2);
/*KJ2-ТОЧКА ВЕРТИК.Ч/З КОТОР.ПРОХОД.ПУТЬ*/
KJ3=IND(KJ2,1,2);END;
ELSE DO;KJ2=IND(KJ1,1,1);KJ3=IND(KJ2,1,1);END;
IV=ITN(K,2); /*IV-НОМЕР ВЕРШИНЫ*/
```

```
IV1=MTR(IV,2); /*IV1-ФАКТИЧ.НОМЕР ВЕР.С Н.С.*/
DELH=POTER(IV,NKS);
IF MTR(IV1,ND-1)^=IV1 THEN MTR(IV1,ND-1)=IV1;
IF IV1^=MTR(1,1)&KJ1=KJ2 THEN MPN(IV)=0;
IF MPN(IV)^=0 THEN DO23:DO; NUT=IV;CALL PMTRK(NUT,MTR,KW,N,MTRK);
IF MPN(NUT)=3 THEN HDT=(KJ2-K*KTV)*DELH;
ELSE HDT=(KJ3-K*KTV)*DELH;
IF KKK=1 THEN DO;PH=POTER(NUT,NKS-1);
DO JR=1 TO KW;IF JR=NUT THEN GOTO M871504;
IV2=MTR(JR,2);IF IV1^=IV2 THEN GOTO M871504;PH1=POTER(JR,NKS-1);
IF PH<PH1 THEN OGRH(IV)=OGRH(JR); MPN(JR)=0;
CALL PMTRK(JR ,MTR,KW,N1,MTRKZ);
DO ILP=2 TO N1;MTRK(N+ILP-1,*)=MTRKZ(ILP,*);END;
N=N+N1-1; M871504:;END;  END;PQ=0;
   CO022:DO KI=1 TO N;
K1=MTRK(KI,1);HSB1(KI)=HSB(K1);
IF KI>1 & MTRK(KI,2)=MTRK(1,1) THEN PQ=PQ+RASH(K1);
END CO022;NDT=PROV(MTRK,HSB1,N);
IF MPN(NUT)=2 THEN PNDT=POTER(IV1,NKS-2);
ELSE PNDT=NDT;IF PNDT<NDT THEN PNDT=NDT;
DELH2=1.5;PRIV=10**10;HDT1=-1;
OGRH(IV)=OGRH(IV)+PNDT;
/*ОПТИМАЛЬНЫЙ СОРТАМЕНТ МОЖЕТ БЫТЬ НЕ ВЫПУКЛЫМ*/
   CD1:DO HDT=PNDT BY DELH2 WHILE(HDT<=OGRH(IV));
         P2=       PQ*OGRH(KW+2)*HDT;HNS=HDT;
         STNS=OGRH(KW+3)+P2;
         PRIZ=1;SOS=OPTIMM(HDT,PRIZ,HSB1,N,MTRK);
         PRIV1=SOS*OGRH(KW+1)+STNS;/*СТОИМОСТЬКУСТА*/
  IF PRIV>PRIV1 THEN DO;PRIV=PRIV1;HDT1=HDT;END;
```

```
END CD1;IF HDT1=-1 THEN DO;
PUT SKIP LIST('ПЕРЕРАСЧЕТ КУСТА НЕ СРАБОТАЛ');
PUT SKIP LIST(PNDT,DELH2,OGRH(IV));END;
DO1:DO;HDT=HDT1;PRIZ=1;
IF KKK=0 THEN SOS=OPTIMM(HDT,PRIZ,HSB1,N,MTRK);
POTER(NUT,NKS-1)=HDT;
IF KKK=1 THEN DO2:DO;PRIZ=0; PUT SKIP(3);
SOS=OPTIMM(HDT,PRIZ,HSB1,N,MTRK);
 PRIV=SOS*OGRH(KW+1)+HDT*OGRH(KW+2)*PQ+OGRH(KW+3);
P2=HDT*OGRH(KW+2)*PQ;CTVV4=CTVV4+PRIV;
  PNS=PRIV-SOS*OGRH(KW+1)-P2;/*-СТОИМ.Н.С.------*/
  PUT EDIT('СТОИМОСТЬ ТРУБ СЕТИ РАВНА',SOS)
(SKIP,X(63),A,X(7),F(9,2));
  PUT EDIT('СТОИМОСТЬ НАСОСНОЙ СТАНЦИИ РАВНА',PNS)
(SKIP,X(63),A,F(9,2));
  PUT EDIT(';НЕРГЕТИЧЕСКИЕ ЗАТРАТЫ РАВНЫ',P2)
(SKIP,X(63),A,X(4),F(9,2));
  PUT EDIT('ПРИВЕДЕННЫЕ ЗАТРАТЫ РАВНЫ',PRIV)
(SKIP,X(63),A,X(7),F(10,3));
  PUT EDIT('СТОИМОСТИ ПЕЧАТАЕТСЯ В ТЫС.РУБЛЯХ')
(SKIP,X(63),A); END DO2;END DO1;
IF KKK=1 THEN DO3:DO; /*ПЕЧАТЬ ВИДА НАС.СТ.*/IK=KJ1-K*KTV;
         IF KJ1^=KJ2&IK=0 THEN
  PUT EDIT('СТАНсИЯ ПЕРЕКАЧКИ В НАЧАЛЕ УЧАСТКА',IV)
(SKIP,X(63),A,F(4)); IF KJ1^=KJ2&IK^=0 THEN
  PUT EDIT('СТАНсИЯ ПОДКАЧКИ В НАЧАЛЕ УЧАСТКЕ',IV)
(SKIP,X(63),A,F(4));IF KJ1^=KJ2&KJ2^=KJ3&IK=0 THEN
  PUT EDIT('СТАНсИИ ПЕРЕКАЧКИ И ПОДКАЧКИ В НАЧАЛЕ'
' УЧАСТКА',IV)(SKIP,X(56) ,A,F(4)); IF ITIP=1 THEN
```

```
PUT EDIT('ПРИМЕЧАНИЕ: СТОИМОСТЬ НАСОСНОЙ СТАНсИИ
  СЧИТАЕТСЯ ПОСТОЯННОЙ')(SKIP,X(40),A);
  MPN(NUT)=0;  END D03;   END DG023;  END C0021;
 IF KKK=0 THEN D04:DO;
  C111111:DO I=1 TO II;K=MTRN(I,1);IV=ITN(K,2);
 IV1=MTR(IV,2);IF IV1^=MTR(1,1)&POTER(IV,NKS-1)<=
POTER(IV1,NKS-2) THEN MPN(IV)=0;  END C111111;END D04;
  END GLAVC  PUT EDIT('ОБЩАЯ СТОИМОСТЬ=',CTVV4,'ТЫС.РУБ.')
(SKIP,X(20),A,F(10,3),A);END DD001;  END MD000;
  NCON: END VIBORI;
/*****************************************************************/
/* Подлрограмма расчета стоимости насосных станций          */
/*****************************************************************/
   STOIMN:PROCEDURE(HNS,RASH1,ITIP);
          DCL HNS,RASH1,
      1 STRUCT1 EXT,2 TR(100,10),2 ST(100,10),2 DVL(100,10),
        2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
        2 (GOST(100) ,2 MT(100) ,2 TRM(100) )  CHAR(1),
        2 AB(300,50) FIXED DEC(3),2 POTER(300,50),
        2 V(100) FIXED DEC(1),  2 HOPCTH(10,10)  ;
   PK1=16.5;PH=HNS*RASH1; IF ITIP=2 THEN D0001:DO; I1=0;
   PH=HNS*RASH1;  C0001:DO I=2 TO 5;
          IF PH>HOPCTH(I-1,1) THEN IF PH<=HOPCTH(I,1)
          THEN I1=I; END C0001;IF I1=0 THEN I1=5;
  C0002:DO J=2 TO 6;
          IF HNS>HOPCTH(1,J-1)THEN IF HNS<=HOPCTH(1,J)
          THEN J1=J; END C0002;
          PK1=16.5;STNS1=PH*HOPCTH(I1,J1)+PK1*PH;
          END D0001; ELSE   D0002:DO;
```

```
        STNS1=120; STNS1=10;   STNS1=STNS1+PK1*PH;
        END D0002; RETURN(STNS1); END STOIMN;
/*****************************************************************/
/*  Подпрограмма ввода-вывода исходной информации          */
/*****************************************************************/
   VVOD1:PROCEDURE(MTR,KW);
 DCL  MAGV CHAR(70),KW,MTR(300,50) FIXED DEC(3),
    1 STRUCT2 EXT, 2 N1,2 N2,2 NSOR,2 NP,
    2 SN,2 VK,2 VMAX, 2 TRC,2 TCH,2 TG,2 TA,2 RTN,2 POLI,
    2 DVLG,2 DVCH,2 DVLR,2 DVLA,2 DVLP,2 RR,2 KR,
   1 STRUCT1 EXT, 2 TR(100,10),2 ST(100,10),2 DVL(100,10),
   2 MOV(50) FIXED DEC(3),2 IND(500,10,2) FIXED DEC(3),
   2 GOST(100) CHAR(10),2 MT(100) CHAR(10),
   2 TRM(100) CHAR(1),2 AB(300,50) FIXED DEC(3),
   2 POTER(300,50),2 V(100) FIXED DEC(1),2 HOPCTH(10,10),
  1 STRUCT(300) EXT,  2 DLIN, 2 VIS, 2 RASH, 2 HSB,2 MATER FIXED
    DEC(2),  2 MPN FIXED DEC(1), 2 OGRH;
 MTR=0; TRC=0; TCH=0; TG=0; RTN=0;POLI=0; TA=0;
  GET LIST(MAGV); PUT EDIT((90)'_')(SKIP,A);
  PUT EDIT(':','ИСХОДНЫЕ ДАННЫЕ ДЛЯ ''ГИДРАВЛИЧЕСКОГО РАСЧЕТА',
'И-13',':')(SKIP,A,X(25),A,X(10),A,X(6),A);
  PUT EDIT((90)'-')(SKIP,A);
  PUT EDIT(':НАИМЕНОВАНИЕ',':')(SKIP,A,X(76),A);
  PUT EDIT(':ОБЪЕКТА',':')(SKIP,A,X(81),A);
  PUT SKIP(0) EDIT(MAGV)(SKIP(0),X(16),A(70));
  PUT EDIT((90)'-')(SKIP,A);
  PUT EDIT(': СОСТАВИЛ',':',':',':НАИМЕНОВАНИЕ ''СХЕМЫ
:СТАДИЯ',':')  (SKIP,A,X(14),A,X(15),A,X(9),A,X(12),A);
  PUT EDIT((51)'-',(20)'-')(SKIP,A,X(19),A);
```

```
  PUT EDIT(': ПРОВЕРИЛ',':',':',':',':ЧЕРТЕЖ:ШИФР',':')
(SKIP,A,X(14),A,X(15),A,X(9),A,X(19),A,X(7),A);
PUT EDIT((51)'-',':',':ОБЬЕКТА',':')
(SKIP,A,X(19),A,X(6),A,X(4),A);
PUT EDIT(': ГЛ.ИНЖ.ПР.',':',':',':',(20)'-')
(SKIP,A,X(12),A,X(15),A,X(9),A,X(19),A);
  PUT EDIT((51)'-',':ДАТА',':')(SKIP,A,X(19),A,X(14),A);
  PUT EDIT(': НАЧ-К ОТДЕЛА',':',':',':',':',':')
          (SKIP,A,X(10),A,X(15),A,X(9),A,X(19),A,X(18),A);
  PUT EDIT((90)'-')(SKIP,A);
  PUT EDIT(':НОМЕР',':ДЛИНА',':ВЫСОТНАЯ',':РАСХОД НА:'
'СВОБОДН.',':МАТЕРИАЛ',':МЕСТОРАС-:','ПРИМЕЧАНИЕ',':')
(SKIP,A,X(4),A,X(4),A,X(1),A,X(1),A,X(1),A,X(2),A,X(6),A);
  PUT EDIT(':УЧАСТКА N:УЧАСТКА M:ОТМЕТКА B:УЧАСТКЕ',
  ':НАПОР B  : ТРУБ    :ПОЛОЖЕНИЕ:',':')(SKIP,A,X(2),A,X(18),A);
  PUT EDIT(':',':',':КОНсЕ M  :КУБ.M    :КОНсЕ    :',':'
'НАСОСНОЙ :',':')(SKIP,A,X(9),A,X(9),A,X(9),A,X(18),A);
PUT EDIT(':',':',':',':',':УЧАСТКА  :',':СТАНЦИИ ',
':',':') (SKIP,A,X(9),A,X(9),A,X(9),A,X(9),A,X(9),A,X(18),A);
  PUT EDIT((90)'-')(SKIP,A);
  PUT EDIT(':','1',':','2',':','3',':','4',':','5',':',
  '6',':','7',':','8',':')
(SKIP,A,X(4),A,X(4),A,X(4),A,X(4),A,X(4),A,X(4),A,X(4),A,
X(4),A,X(4),A,X(4),A,X(4),A,X(4),A,X(4),A,X(4),A,
X(9),A,X(8),A);  PUT EDIT((90)'-')(SKIP,A);
 RTN=TG;TG=0; ND=50;MTR(1,1)=1; /* ВВОД ИСХОДНЫХ ДАННЫХ */
 /* I1-НАЧАЛО УЧАСТКА,I2-КОНЕЦ УЧАСТКА,KW-КОЛИЧЕСТВО ВЕРШИН
СЕТИ*/ KW=0;  MMM01:GET LIST(I1,I2);
IF I2^=0 THEN   DO001:DO;KW=KW+1;MTR(KW,ND-1)=I2;
```

```
GET LIST(DLIN(I2),VIS(I2),RASH(I2),HSB(I2),MATER(I2),MPN(I2));
   PUT EDIT(':',I1,I2,':',DLIN(I2),':',VIS(I2),'!',
   RASH(I2),':',HSB(I2),':',MATER(I2),':',MPN(I2),':',':')
(SKIP,A,X(1),F(3),X(1),F(3),X(1),A,F(8,2),X(1),A,
F(9,3),A,F(8,5),X(1),A, X(2),F(6,2),X(1),A,X(3),
F(2),X(4),A,X(4),F(1),X(4),A,X(18),A);
  PUT EDIT((90)'-')(SKIP,A);
         MTR(I2,ND)=I1;   MTR(I2,1)=I2; END D0001;
         IF I1^=0 THEN  D01:D0;
        D0 J=1 BY 1 WHILE(MTR(I1,J)^=0);END;
         ND1=ND-2;IF J>ND1 THEN D0;
   PUT EDIT(I1,'-АЯ ТРАЕКТОРИЯ СОДЕРЖИТ БОЛЬШЕ ВЕРШИН,ЧЕМ'
' ДОПУСТИМО')(SKIP,F(3),A);
  PUT EDIT('РАЗБЕЙТЕ СЕТЬ НА ДВЕ ЧАСТИ')(SKIP,A);MTR(1,1)=-1;END;
  C0001:D0 J1=2 TO ND-2;
         MTR(I2,J1)=MTR(I1,J1-1); END C0001; END D01;
         IF I2^=0 THEN GOTO MMM01;
         D0 I=1 TO KW;  IF MTR(I,1)=0 THEN D0;
  PUT EDIT('ОШИБКА В ДАННЫХ,НЕТ ВЕРШИНЫ С НОМЕРОМ',I)
(SKIP,A,F(3)); MTR(1,1)=-1;END; IK=MTR(I,ND-1);
         D0 II=I+1 TO KW;IF IK=MTR(II,ND-1) THEN D0;
  PUT EDIT('ОШИБКА В ДАННЫХ, ВЕРШИНА',IK,'ПОВТОРЯЕТСЯ'
' НА 2-ОМ СТОЛБЦЕ')(SKIP,A,F(3),A);
         MTR(1,1)=-1;END; END;END;
         IF MTR(1,1)^=-1 THEN DD000:D0;  MMC:KL=0;
   C0003:D0 I=2 TO KW;
         D0 I3=1 TO ND WHILE(MTR(I,I3)^=0);END;
         I3=I3-1; IF MTR(I,I3)^=MTR(1,1) THEN
   D0003:D0;I1=MTR(I,ND);
```

```
C0004:DO J=2 TO ND-2;MTR(I,J)=MTR(I1,J-1); END C0004;
 END D0003;  END C0003;
 C0005:DO J1=2 TO KW;
       DO J3=1 BY 1 WHILE(MTR(J1,J3)^=0);END;J3=J3-1;
       IF MTR(J1,J3)^=MTR(1,1) THEN  KL=1;
 END C0005; IF KL=1 THEN GOTO MMC;
C0002:DO I=1 TO KW;MTR(I,ND-1)=0;MTR(I,ND)=0;END C0002;
  END DD003; END VVOD1;
```

Приложение 6

Листинг контрольного примера по задаче Технико-экономического и гидравлического расчета кольцевых сетей

С А П Р - С К Г В Х

(II-Очередь)

Решается задача выбора параметров кольцевых сетей

В Х О Д Н А Я И Н Ф О Р М А Ц И Я

Шифр обьекта - 001

0-диаметры на участках не заданы

0-узловые расходы не заданы

0-расходы на участках не заданы

0-промежуточная печать отключена

1-аврийные участки заданы

0-ограничение на напор не задан

0-напор на вершине не задан

0-коэффициент "Альфа" целевой функции не задан

0-коэффициент "Бета" целевой функции не задан

0-шаг по напору не задан

Таблица П.6.1

Исходная информация по участкам сети

Номер участка	Номер начала и конца	Длина участка	Диаметр трубы	Материал трубы или код	Расход на участке	Коэффициент путевого отбора
1	2	3	4	5	6	7
1	1- 2	2000.0				0.00
2	2- 3	3000.0				1.00
3	2- 4	5000.0				1.00
4	2- 5	5000.0				1.00
5	2- 6	6000.0				1.00
6	3- 4	4000.0				1.00
7	5- 4	5000.0				0.00
8	6- 5	3000.0				0.00

Таблица П.6.2

Невязка по расходам	0.1000
Невязка по потерям	0.5000
Минимальная скорость течения	0.1000
Максимальная скорость течения	4.5000

Продолжение таблицы П.6.2

Расход населенного пункта	120.000

Таблица П.6.3

Заданный сортамент труб

Количество сортаментов	Шифр сортамента	Количество труб в таблице ограничений
1	500	0

Таблица П.6.4

Исходная информация по узлам сети

Номер узла	Высотная отметка узла	Свободный напор в узле	Расход в узле	Сосредоточенный расход	Расход на пожар в узле
1	2	3	4	5	6
1	64.22	0.000		0.000	0.000
2	62.24	14.000		10.700	10.000
3	62.23	14.000		0.000	0.000
4	77.53	14.000		0.000	0.000
5	77.53	14.000		0.000	15.000
6	90.80	14.000		20.800	0.000

Таблица П 6.5

Исходная информация по вариантам аварий на сети

Номер варианта	Номера аварийных участков
1	2 3
2	3 6 8

ВЫХОДНАЯ ИНФОРМАЦИЯ

Таблица П.6.6

Количество участков	Количество узлов	Количество сосредоточенных расходов	Количество пожаров
8	6	2	2

Таблица П.6.7

Расчетные расходы воды по узлам и по участкам сети

Номер узла	Участки примыкающие к узлу	Длина участка	Удельный расход	Путевые расходы	Узловые расходы сосредоточенный	Узловые расходы приведенный к узлу	Начальное значение потоков
1	2	3	4	5	6	7	8
1	1- 2	2000.0	0.00276	0.00	0.000	0.0000	120.0
2	1- 2	2000.0	0.00276	0.00	10.700	46.9283	120.0
	2- 3	3000.0	0.00276	8.28			25.0
	2- 4	5000.0	0.00276	13.80			15.3
	2- 5	5000.0	0.00276	13.80			18.2
	2- 6	6000.0	0.00276	16.57			14.5
3	2- 3	3000.0	0.00276	8.28	0.000	9.6630	25.0
	3- 4	4000.0	0.00276	11.04			15.3
4	2- 4	5000.0	0.00276	13.80	0.000	12.4239	15.3
	3- 4	4000.0	0.00276	11.04			15.3
	4- 5	5000.0	0.00276	0.00			18.2
5	2- 5	5000.0	0.00276	13.80	0.000	21.9022	18.2
	4- 5	5000.0	0.00276	0.00			18.2
	5- 6	3000.0	0.00276	0.00			14.5

Продолжение таблицы П.6.7

1	2	3	4	5	6	7	8
6	2- 6	6000.0	0.00276	16.57	20.800	29.0826	14.5
	5- 6	3000.0	0.00276	0.00			14.5

1-ый контур

1	2-	4	16.093166
2	3-	4	-7.976425
3	2-	3	-8.116742

Невязка по потерям: 0.000001

2-ый контур

1	2-	5	26.184723
2	3-	4	-7.976425
3	4-	5	-10.091557
4	2-	3	-8.116742

Невязка по потерям: 0.000001

3-ый контур

1	2-	6	31.636175
2	3-	4	-7.976425
3	4-	5	-10.091557
4	5-	6	-5.451448
5	2-	3	-8.116742

Невязка по потерям -0.000003

Таблица П.6.7

Результаты гидравлического расчета сети

N/N	Участок	Длина участка	Расход на участке	Труба большего диаметра: диаметр	материал	длина	скорость	потери напора
1	2	3	4	5	6	7	8	10
1	2- 4	5000.0	15.3	0.1888	П, Л	342.6	0.548	0.68321
2	2- 5	5000.0	18.2	0.1698	П, Л	3908.7	0.805	17.58685
3	2- 6	6000.0	14.5	0.1698	П, Л	17.9	0.642	0.05393
4	3- 4	4000.0	15.3	0.1888	П, Л	4000.0	0.548	7.97643
5	4- 5	5000.0	18.2	0.2124	П, Л	2972.7	0.515	4.59400
6	5- 6	3000.0	14.5	0.1888	П, Л	3000.0	0.520	5.45145
7	1- 2	2000.0	120.0	0.3354	П, Л	2000.0	1.359	9.88648
8	2- 3	3000.0	25.0	0.2124	П, Л	3000.0	0.706	8.11674

Окончание таблицы П.6.7

Участок	Характеристика трубы меньшего диаметра: диаметр	материал	длина	скорость	потери напора	Потери напора на участке	Стоимость участка
11	12	13	14	15	16	17	18
2- 4	0.1698	П, Л	4657.4	0.68	15.4100	16.09317	13512.4

Продолжение таблицы П.6.7

11	12	13	14	15	16	17	18
2- 5	0.1510	П, Л	1091.3	1.02	8.5979	26.18472	12994.4
2- 6	0.1510	П, Л	5982.1	0.81	31.5822	31.63618	14285.0
3- 4	0.1698	П, Л	0.0	0.00	0.0000	7.97643	13120.0
4- 5	0.1888	П, Л	2027.3	0.65	5.4976	10.09156	18748.4
5- 6	0.1698	П, Л	0.0	0.00	0.0000	5.45145	9840.0
1- 2	0.2976	П, Л	0.0	0.00	0.0000	9.88648	19760.0
2- 3	0.1888	П, Л	0.0	0.00	0.0000	8.11674	12210.0

Таблица П.6.8

Результаты гидравлического расчета сети

Номер узла	Расход воды в узле	Высотная отметка узла	Напор в узле	Пьезометр узла
1	0.00	64.22	82.10	146.32
2	46.93	62.24	74.20	136.44
3	9.66	62.23	66.09	128.32
4	12.42	77.53	42.81	120.34
5	21.90	77.53	32.72	110.25
6	29.08	90.80	14.00	104.80

Напор на вершине сети равен 82.1027

Приведенные затраты на сеть равны 205.177

Производится расчет на максимально-хозяйственный режим водопотребления с выбранными диаметрами на сети.

Участки на которых были уложены два диаметра делятся на два участка введением дополнительных вершин.

Участки с дополнительными вершинами :

2 - 4 вершина 7

2 - 5 вершина 8

2 - 6 вершина 9

4 - 5 вершина 10

Таблица П.6.9

Расчетные расходы воды по узлам и по участкам сети

Номер узла	Участки примыкающие к узлу	Длина участка	Удельный расход	Путевые расходы	Узловые расходы сосредоточенный	Узловые расходы приведенный к узлу	Начальное значение потоков
1	2	3	4	5	6	7	8
1	1- 2	2000.0	0.00000	0.00	0.000	0.0000	95.0
2	2- 7	342.6	0.00000	0.00	0.000	36.9280	11.6
	2- 8	3908.7	0.00000	0.00			10.7
	2- 9	17.9	0.00000	0.00			14.5
	1- 2	2000.0	0.00000	0.00			95.0
	2- 3	3000.0	0.00000	0.00			21.2
3	3- 4	4000.0	0.00000	0.00	0.000	9.6630	11.6
	2- 3	3000.0	0.00000	0.00			21.2

Продолжение таблицы П.6.9

1	2	3	4	5	6	7	8
4	3- 4	4000.0	0.00000	0.00	0.000	12.4240	11.6
	4- 10	2972.7	0.00000	0.00			10.7
	7- 4	4657.4	0.00000	0.00			11.6
5	5- 6	3000.0	0.00000	0.00	0.000	6.9020	14.5
	8- 5	1091.3	0.00000	0.00			10.7
	10- 5	2027.3	0.00000	0.00			10.7
6	5- 6	3000.0	0.00000	0.00	0.000	29.0830	14.5
	9- 6	5982.1	0.00000	0.00			14.5
7	2- 7	342.6	0.00000	0.00	0.000	0.0000	11.6
	7- 4	4657.4	0.00000	0.00			11.6
8	2- 8	3908.7	0.00000	0.00	0.000	0.0000	10.7
	8- 5	1091.3	0.00000	0.00			10.7
9	2- 9	17.9	0.00000	0.00	0.000	0.0000	14.5
	9- 6	5982.1	0.00000	0.00			14.5
10	4- 10	2972.7	0.00000	0.00	0.000	0.0000	10.7
	10- 5	2027.3	0.00000	0.00			10.7

1-ый контур

1 7- 4 10.114175

2	2-	7	0.448418
3	3-	4	-4.637877
4	2-	3	-5.938378

Невязка по потерям: 0.013662

2-ый контур

1	9-	6	22.157143
2	2-	8	-9.832460
3	2-	9	0.037837
4	5-	6	-7.325525
5	8-	5	-4.806899

Невязка по потерям: -0.230096

3-ый контур

1	10-	5	2.226978
2	2-	8	-9.832460
3	3-	4	4.637877
4	4-	10	1.860959
5	2-	3	5.938378
6	8-	5	-4.806899

Невязка по потерям: -0.024832

Таблица П.6.10

Результаты гидравлического расчета сети

N/N	Участок	Длина участка	Расход на участке	Труба большего диаметра: диаметр	материал	длина	скорость	потери напора
1	2	3	4	5	6	7	8	10
1	2- 4	5000.0	12.1	0.1888	П	342.6	0.432	0.44842
2	2- 5	5000.0	13.1	0.1698	П	3908.7	0.580	9.83246
3	2- 6	6000.0	11.9	0.1698	П	17.9	0.526	0.03784
4	3- 4	4000.0	11.3	0.1888	П	4000.0	0.403	4.63788
5	4- 5	5000.0	10.9	0.2124	П	2972.7	0.309	1.86096
6	5- 6	3000.0	17.2	0.1888	П	3000.0	0.614	7.32552
7	1- 2	2000.0	95.0	0.3354	П	2000.0	1.076	6.53214
8	2- 3	3000.0	21.0	0.2124	П	3000.0	0.592	5.93838

Окончание таблицы П.6.10

Участок	Характеристика трубы меньшего диаметра: диаметр	материал	длина	скорость	потери напора	Потери напора на участке	Стоимость участка
11	12	13	14	15	16	17	18
2- 4	0.1698	П	4657.4	0.53	10.11417	10.56259	

Продолжение таблицы П .6.10

11	12	13	14	15	16	17	18
2- 5	0.1510	П	1091.3	0.73	4.80690	14.63936	
2- 6	0.1510	П	5982.1	0.67	22.15714	22.19498	
3- 4	0.0000		0.0	0.00	0.00000	4.63788	
4- 5	0.1888	П	2027.3	0.39	2.22698	4.08794	
5- 6	0.0000		0.0	0.00	0.00000	7.32552	
1- 2	0.0000		0.0	0.00	0.00000	6.53214	
2- 3	0.0000		0.0	0.00	0.00000	5.93838	

Таблица П.6.11

Результаты гидравлического расчета сети

Номер узла	Расход воды в узле	Высотная отметка узла	Напор в узле	Пьезометр узла
1	0.00	64.22	69.08	133.30
2	36.93	62.24	64.52	126.76
3	9.66	62.23	58.60	120.83
4	12.42	77.53	38.66	116.19
5	6.90	77.53	34.60	112.13
6	29.08	90.80	14.00	104.80

Минимальный потербный напор равна 69.077

Расчет сети на случай аварии : вариант 1

Исключаемые участки:

2- 3

2- 4

Таблица П.6.12

Расчетные расходы воды по узлам и по участкам сети

Номер узла	Участки примыкающие к узлу	Длина участка	Удельный расход	Путевые расходы	Узловые расходы сосредоточенный	Узловые расходы приведенный к узлу	Начальное значение потоков
1	2	3	4	5	6	7	8
1	1- 2	2000.0	0.00000	0.00	0.000	0.0000	66.5
2	2- 8	3908.7	0.00000	0.00	0.000	25.8496	30.5
	2- 7	17.9	0.00000	0.00			10.2
	1- 2	2000.0	0.00000	0.00			66.5
3	4- 3	4000.0	0.00000	0.00	0.000	6.7641	6.8
4	4- 3	4000.0	0.00000	0.00	0.000	8.6968	6.8
	9- 4	2972.7	0.00000	0.00			15.5
5	5- 9	2027.3	0.00000	0.00	0.000	4.8314	15.5
	5- 6	3000.0	0.00000	0.00			10.2
	8- 5	1091.3	0.00000	0.00			30.5
6	7- 6	5982.1	0.00000	0.00	0.000	20.3581	10.2
	5- 6	3000.0	0.00000	0.00			10.2

Продолжение таблицы П.6.12

1	2	3	4	5	6	7	8
7	7- 6	5982.1	0.00000	0.00	0.000	0.0000	10.2
	2- 7	17.9	0.00000	0.00			10.2
8	2- 8	3908.7	0.00000	0.00	0.000	0.0000	30.5
	8- 5	1091.3	0.00000	0.00			30.5
9	5- 9	2027.3	0.00000	0.00	0.000	0.0000	15.5
	9- 4	2972.7	0.00000	0.00			15.5

1-ый контур

1	5-	6	0.411366
2	7-	6	-41.538288
3	2-	8	27.993000
4	2-	7	-0.070933
5	8-	5	13.685234

Невязка по потерям: -0.480379

Таблица П.6.13

Результаты гидравлического расчета сети

N/N	Участок	Длина участка	Расход на участке	Труба большего диаметра: диаметр	материал	длина	скорость	потери напора
1	2	3	4	5	6	7	8	10
1	5- 6	3000.0	3.4	0.1888	П	3000.0	0.121	0.41137
2	5- 4	5000.0	15.5	0.1888	П	2027.3	0.553	4.10752
3	2- 5	5000.0	23.7	0.1698	П	3908.7	1.046	27.99300
4	2- 6	6000.0	17.0	0.1698	П	17.9	0.750	0.07093
5	4- 3	4000.0	6.8	0.1888	П	4000.0	0.242	1.86978
6	1- 2	2000.0	66.5	0.3354	П	2000.0	0.753	3.46944

Окончание таблицы П.6.13

Участок	Характеристика трубы меньшего диаметра: диаметр	материал	длина	скорость	потери напора	Потери напора на участке	Стоимость участка
11	12	13	14	15	16	17	18
5- 6	0.0000		0.0	0.00	0.00000	0.41137	
5- 4	0.2124	П	2972.7	0.44	3.43242	7.53995	
2- 5	0.1510	П	1091.3	1.32	13.68523	41.67823	

Продолжение таблицы П.6.13

1	2	3	4	5	6	7	8	10
2- 6	0.1510		П	5982.1	0.95	41.53829	41.60922	
4- 3	0.0000			0.0	0.00	0.00000	1.86978	
1- 2	0.0000			0.0	0.00	0.00000	3.46944	

Таблица П.6.14

Результаты гидравлического расчета сети

Номер узла	Расход воды в узле	Высотная отметка узла	Напор в узле	Пьезометр узла
1	2	3	4	5
1	0.00	64.22	85.66	149.88
2	25.85	62.24	84.17	146.41
3	6.76	62.23	33.09	95.32
4	8.70	77.53	19.66	97.19
5	4.83	77.53	27.20	104.73
6	20.36	90.80	14.00	104.80

Минимальный потербный напор равен 85.6587

Расчет сети на случай аварии : вариант 2

Исключаемые участки: 2- 4

3- 4

6- 5

Таблица П.6.15

Расчетные расходы воды по узлам и по участкам сети

Номер узла	Участки примыкающие к узлу	Длина участка	Удельный расход	Путевые расходы	Узловые расходы сосредоточенный	Узловые расходы приведенный к узлу	Начальное значение потоков
1	2	3	4	5	6	7	8
1	1- 2	2000.0	0.00000	0.00	0.000	0.0000	66.5
2	2- 8	17.9	0.00000	0.00	0.000	25.8500	13.5
	2- 7	4000.0	0.00000	0.00			20.4
	1- 2	2000.0	0.00000	0.00			66.5
	2- 3	3000.0	0.00000	0.00			6.8
3	2- 3	3000.0	0.00000	0.00	0.000	6.7640	6.8
4	9- 4	2000.0	0.00000	0.00	0.000	8.6970	8.7
5	5- 9	2027.3	0.00000	0.00	0.000	4.8310	8.7
	8- 5	1091.3	0.00000	0.00			13.5
6	7- 6	5982.1	0.00000	0.00	0.000	20.3580	20.4
7	7- 6	5982.1	0.00000	0.00	0.000	0.0000	20.4
	2- 7	4000.0	0.00000	0.00			20.4

Продолжение таблицы П.6.15

1	2	3	4	5	6	7	8
8	2- 8	17.9	0.00000	0.00	0.000	0.0000	13.5
	8- 5	1091.3	0.00000	0.00			13.5
9	5- 9	2027.3	0.00000	0.00	0.000	0.0000	8.7
	9- 4	2000.0	0.00000	0.00			8.7

Таблица П.6.16

Результаты гидравлического расчета сети

N/N	Участок	Длина участка	Расход на участке	Труба большего диаметра: диаметр	материал	длина	скорость	потери напора
1	2	3	4	5	6	7	8	10
1	5- 4	4027.3	8.7	0.1888	П	2027.3	0.311	1.48018
2	2- 5	1109.2	13.5	0.1698	П	17.9	0.598	0.04745
3	2- 6	9982.1	20.4	0.1698	П	4000.0	0.899	21.90864
4	1- 2	2000.0	66.5	0.3354	П	2000.0	0.753	3.46944
5	2- 3	3000.0	6.8	0.2124	П	3000.0	0.191	0.79919

Окончание таблицы П.6.16

Участок	Характеристика трубы меньшего диаметра: диаметр	материал	длина	скорость	потери напора	Потери напора на участке	Стоимость участка
11	12	13	14	15	16	17	18
5- 4	0.2124	П	2000.0	0.25	0.83218	2.31236	
2- 5	0.1510	П	1091.3	0.76	5.06876	5.11621	
2- 6	0.1510	П	5982.1	1.14	57.37159	79.28022	
1- 2	0.0000		0.0	0.00	0.00000	3.46944	
2- 3	0.0000		0.0	0.00	0.00000	0.79919	

Таблица П.6.17

Результаты гидравлического расчета сети

Номер узла	Расход воды в узле	Высотная отметка узла	Напор в узле	Пьезометр узла
1	0.00	64.22	123.33	187.55
2	25.85	62.24	121.84	184.08
3	6.76	62.23	121.05	183.28
4	8.70	77.53	99.12	176.65
5	4.83	77.53	101.43	178.96
6	20.36	90.80	14.00	104.80

Минимальный потербный напор равен 123.33

Приложение 7

Распечатки исходных текстов модулей пргограммы задачи
Технико-экономический и гидравлический расчет кольцевых сетей

```
/*********************************************/
/*          Головная программа               */
/*********************************************/
#include<stdlib.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
#include"params.h"
#include<process.h>
extern void shifrob();
extern void novaja1(int NUCH,int  NUZ);
extern void novaja2(int *KK,int NKON,int NUCH,float EPS1),
extern void novaja3(int NKON,int NUCH);
extern void wkoef(int IDH,int IALF,int IBET,int IHAP,
                  int IOGR,int IUZZ,float *QPOS,float *EPS,
                  float *EPS1,float *VMIN,float *VMAX);
extern void naprav(int *II,int *KK);
extern void wmdr(int *NUCH);
extern void derewo(int *NKON,int NUCH);
extern void vibps( int N1, int *N2, float *QNP,
                   int *IPOGAR, int IAV);
extern void wprizt();
extern void bpn(int NUCH);
extern void prov(int NUCH,float *NDT);
extern void sortam( float SN,float VMAX,int N);
extern void rashep(int NUZ,int NUCH);
extern void alfa(int NKON,int NUCH);
extern void pezom(int *NUZ,int NUCH);
```

```
extern void  optimm(int N,float *NDT,int NUZ,
                float *PRIV,float RASHV,float EPS1);
extern void bixod(int IPOGAR,int *NUCH);
extern void isavu(int *NUCH,int KUCH,int NZ,int *NUZ,int KOLAV);
extern void pechin(int *IDZ,int *IUZZ,int *INRAS,int *IOGR,
               int *IHAP,int *IPECH,
               int *IALF,int *IBET,int *IDH,int *IAV);
  int           MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],NSOR,
                ND,NP,KPT,KR,KOD[kuch],MTR[kuch][nd],A1[kuch],
                V[kuch],NAVU[20][10];
  float     POTER[kuch][kpt],DLIN[kuch],RASH[kuch],TRR[kuch],
            HSB[kuch],VIS[kuch],PN[kuch],DLIN1[kuch],
            RASH1[kuch],HAP[kuch],PN1[kuch],
                TR[ktrub][np],ST[ktrub][np],
                S[kuch],Q[kuz], QP[kuz],QS[kuz],DELTAH[kkon],
                REGP,DELHP,BETAB,ALFAF,PNDTZ,OGRH,
                SN,VK, RR;
   char         TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5],
                shifrobj[3];
  FILE         *print;
void main(void) {
 int     AM[kuch],IDZ,IUZZ,INRAS,IOGR,IHAP,
         IPECH,IALF,IBET,IDH,IAV,i,j,I,NUCH,NUZ,NKON,IPOGAR,
         KOLAV,J1,J2,J,JJ,II,KK,K,K3,KT,KOPT,
         K1,K2,JJJ,IK,KN,NZ, NCH,IP,KUCH1,KUZL,KFT,KUCH;
 float    PK,P,VMIN,VMAX, EPS,EPS1,QPOS,PNDT,SUM,RP,PRIW,
          PP,PRIV,PNDT1, P1,P2,PKE,RASHV; FILE *a1000;
      ND=nd-1; KPT=kpt-1;NP=np-1;
      for(i=1;i < kuz;i++) Q[i]=QS[i]=QP[i]=0;for(i=1;i<kuch;i++)
```

```
  {  AM[i]=0;
   for (j=1; j <3; j++)   MKIN[i][j]=0;
     strcpy(TRM[i]," ");
     A1[i]=KOD[i]=TRR[i]=0;
     VIS[i]=HAP[i]= HSB[i]=0;
     PN[i]=DLIN[i]=RASH[i]=0;
     PN1[i]=DLIN1[i]=RASH1[i]= S[i]=0;
    for(j=1;j <= ND;j++)   MTR[i][j]=0;
     for(j=1; j <= KPT; j++)  AB[i][j]=POTER[i][j]=0;
     for(j=1;j < ktrub; j++) KODM[j]=0;
  }
 for(i=1;i<kkon;i++) DELTAH[i]=0;
 shifrob();
 pechin(&IDZ,&IUZZ,&INRAS,&IOGR, &IHAP,&IPECH,
        &IALF,&IBET,&IDH,&IAV);
 DELHP=1.1;ALFAF=1.14;BETAB= 7.58;
if( INRAS==1 && IDZ==1 ) NUCH=-2;
if( INRAS==0 && IDZ==0 ) NUCH=-3;
if( INRAS==0 && IDZ==1 ) NUCH=-1;
if( INRAS==1 && IDZ==0 ) NUCH=-4;
if( IDZ==0 )   REGP=1;  else    REGP=0;
 wmdr(&NUCH);/*-ВВОД ИНФОРМАЦИИ ПО УЧАСТКАМ СЕТИ-*/
if( IUZZ==1 ) NUZ=-1; else  NUZ=1;
if( INRAS==1)  NUCH=-NUCH; OGRH=PNDTZ=0;
 wkoef(IDH,IALF,IBET,IHAP,IOGR,IUZZ,&QPOS,
       &EPS,&EPS1,&VMIN,&VMAX);
 if( REGP==1 )  wprizt();
 II=0;KK=0;          /*II,KK-pAPAMETPy KOHTPOlq */
 naprav(&II,&KK);
```

```
     if( II==1 )  exit(1);
        IPOGAR=0;VMIN=0;KOLAV=0;NKON=0;
        printf("\n\nВход в п/п VIBPS");
MVIBPS: vibps(NUCH,&NUZ,&QPOS,&IPOGAR,IAV);
        printf("\n\nВыход из п/п VIBPS");
MNACHAL:NKON=NUCH-NUZ+1; P=0;
        for( I = 1 ; I <=  NUZ; I += 1) P=P+Q[I]; P=-P;
        RASHV=-P/1000; /*--РАСХОД НА ВЕРШИНЕ-----*/
 derewo(&NKON,NUCH);
       spawnl(P_WAIT,"betamain.exe",NULL);
       a1000=fopen("rasxuu.","r");
       for( I = 1 ; I <=  NUCH; I += 1)
        {fscanf(a1000," %g",&RASH[I]);
         if(I<=NUZ) fscanf(a1000," %g",&Q[I]);
         }
        fclose(a1000);
       if( IPECH==1 )
         for(I=1;I<=NUCH;I++)
           fprintf(print,"\nRASH[%3d]=%8.3f  %8.4f",I,
                   RASH[I],Q[I]);
      if( REGP==1 )
DO0001: {
         rashep(NUZ,NUCH);
         PRIW=1000000000; KT=NUCH+1;
           sortam(VMIN,VMAX,KT);
           prov(KT,&PNDT);
           for( I = 1 ; I <=  NUCH; I += 1)
           {
            if( I<=NKON ) K1=NUZ+I; else  K1=MKIN[I][2];
```

```
            K=MTR[K1][ND];strcpy(TRM2[I],TRM[K]);
            KOD[I]=KODM[K];TRR[I]=TR[K][1];
            }
        } /* D00001*/   KK=0;KOPT=0;
C0062:for( II = 1 ;KK!=1; II += 1 )
      {
       for(i=1;i<kuch;i++) { PN[i]=0;RASH1[i]=RASH[i];}
       bpn(NUCH);
           for(I=1;I<=NUCH;I++)
          {
           S[I]=PN[I]/(RASH[I]*RASH[I]);
           PN[I]=S[I]*fabs(RASH[I])*RASH[I];
      if( IPECH==1 )
fprintf(print,"\nI=%3d %3d %3d DLIN=%8.3f TRR=%8.4f KOD=%4d
           RASH=%8.4f PN=%8.4f S=%8.3f ",I,MKIN[I][1],
          KIN[I][2],DLIN[I],TRR[I],KOD[I],RASH[I],PN[I],
           S[I]);
          }
       KK=1;
       novaja2(&KK,NKON,NUCH,EPS1);
       if( KK==0 )  alfa(NKON,NUCH);
  } /* C0062*/
     if( IPECH==1 )
     for(I=1;I<=NUCH;I++) fprintf(print," PN[%d]=%8.3f
                                %8.4f  ",I,PN[I],RASH[I]);
  if( KOPT==1 || IPECH==1 || REGP==0 )
     novaja3(NKON,NUCH);
       bpn(NUCH);
      if( REGP==1 ) NUZ=-NUZ;
```

```
        pezom(&NUZ,NUCH);
      if( REGP==1 )
DO0002: {
         if( KOPT==1 )
         {
          KOPT=2;NUCH=NUCH+1; goto  DC002;
         }
       derewo(&NKON,NUCH);  rashep(NUZ,NUCH);
      NUCH=NUCH+1;
      sortam(VMIN,VMAX,NUCH);
      prov(NUCH,&PNDT);
     if( PNDTZ<PNDT && IHAP==1 )
       {
        fprintf(print,"\n\nзаданный напор,меньше,чем мин.
                              потреб. напор сети %g", PNDT);
        goto MKON00;
       }
      if( OGRH <PNDT && IOGR==1 )
       {
        fprintf(print,"\n\nограничение на напор менше,чем
                              мин.потр.напо %g", PNDT);
        goto MKON00;
       }
     if( PNDTZ>=PNDT && IHAP==1 ) PNDT=PNDTZ;
     if( IHAP==1 ) NUZ=-NUZ;
     optimm(NUCH,&PNDT,NUZ,&PRIV,RASHV, EPS1);
     window(1,1,80,25); textbackground(BLACK); clrscr();
     if (IPECH==1) fprintf(print,"\nPRIV=%9.3f
                              PNDT=%8.3f",PRIV,PNDT);
```

```
      if( PRIW>=PRIV )
DC001: {
       PRIW=PRIV; KK=1;/* проверка 2-го закона Кирхгофа */
       PNDT1=PNDT;
       for( I = 1 ; I <=  NUCH-1; I += 1)
       {
        if( I<=NKON ) K=NUZ+I; else  K=MKIN[I][2];
        AM[K]=A1[K]+MTR[K][ND]*100;PN[I]=PN1[K];
        if (IPECH==1)
          fprintf(print,"\n PN1[%d]=%8.5f HAP[%3d]=%8.3f
             %8.3f .%8.3f",K,PN1[K],K,HAP[K],VIS[K],PNDT1);
        }
       if( IHAP==1 )  goto . DC002;
        } /* DC001*/
M00001: if( KK==0 )
       {
         NUCH=NUCH-1;
         novaja1(NUCH,NUZ);  goto  C0062;
  DC002:for(i=1;i<=NUCH;i++)
        {
         MTR[i][ND]=AM[i]/100;A1[i]=AM[i]-MTR[i][ND]*100;
         }
        for( I = 1 ; I <=  NUCH-1; I += 1)
        {
         if( I<=NKON ) K=I+NUZ; else  K=MKIN[I][2];
         DLIN1[K]=DLIN[I];RASH1[K]=RASH[I];
        if(IPECH==1)
        fprintf(print,"\n PN1[%d]=%8.5f HAP[%3d]=%8.3f
          %8.3f %8.3f",K,PN1[K],K,HAP[K],VIS[K],PNDT1);
```

```
  }
 if( KOPT==1 ) VMAX=2*VMAX;
 sortam(VMIN,VMAX,NUCH);
 if( KOPT==0 )
  {
   KOPT=1;HAP[1]=PNDT1;KK=0;VMAX=2*VMAX;
    goto  M00001; }  /*--11.12*/
 NUCH=NUCH-1;
} /* D00002*/
if( REGP==0 )
 { P=0;
   for( I = 1 ; I <=  NUZ; I += 1)   if( P<HAP[I] )
         P=HAP[I];
   if( IHAP==1 )
    {
      if ( P<PNDTZ ) P=PNDTZ;
   if( P>PNDTZ )     {
     fprintf(print,"\n\nЗаданный напор на вершине
            сети=%g меньше,чем мин.потр.напор=%g",
     PNDTZ,P);exit(1); }
    }
     for(i=1;i<=NUZ;i++)
     { P1=HSB[i]-HAP[i];HAP[i]=P+P1;
      if(IPECH==1)
      fprintf(print,"\nmkp HAP[%d]=%8.3f P=%8.3f",
                        i,HAP[i],P);
      }
     for(i=1;i<=NUCH;i++) strcpy(TRM[i],TRM2[i]);
   }
```

```
    if( REGP==1 )
       if( IOGR==1 )
         if( PNDT1>OGRH )
           { IHAP=1;PNDTZ=OGRH; }
        P=1000;/* HAP[1]=PNDT1;*/
        for( I = 1 ; I <=  NUZ; I += 1)
        {
         if ( P>HAP[I]-HSB[I])  P=HAP[I]-HSB[I];
         if (IPECH==1) fprintf(print,"\nHAP[%d]=%g
                    HSB[%d]=%g",I,HAP[I],I,HSB[I]);
        }
         if( IHAP==0 && REGP==1 )
            {
       PNDT1=PNDT1-P;
       for(i=1;i<kuch;i++) HAP[i]=HAP[i]-P;
            }
         if( IHAP==1 && REGP==1 )
SM19587: {
          if( fabs(P)<=0.0001 ) P=0;
          PNDT1=PNDT1-P;
          for(i=1;i<kuch;i++) HAP[i]=HAP[i]-P;
          if( IHAP==1 || IOGR==1 )
        fprintf(print,"\n\nВозможно наруш.ограничен. по
                          VMAX в начале сети");
          KFT=0;PKE=PNDT1-PNDTZ;HAP[1]=PNDTZ;
          PNDT1=PNDTZ;
    SMIF: if( fabs(PKE)>0.1 )
SM195870: for( I = 1 ; I <=  NUCH; I += 1)
        { if( MKIN[I][1]==MTR[1][1] )
```

```
            {
          if( PKE-PN[I]>0.1 )
          {
      for( JJJ = 1 ; JJJ <=  NSOR-1; JJJ += 1)
       for(j=1;j<=NP;j++)
        { TR[JJJ][j]=TR[JJJ+1][j];ST[JJJ][j]=ST[JJJ+1][j];}
          for(j=1;j<=NP;j++) {TR[NSOR][j]=0;ST[NSOR][j]=0;}
              NSOR=NSOR-1;
              for(i=1;i<kuch;i++) RASH[i]=RASH[i]*1000;
              goto  MNACHAL;
            }
         PN[I]=PN[I]-PKE;
         if( I<=NKON ) K=I+NUZ; else  K=MKIN[I][2];
         for( IK = 1 ; IK <=  KPT-2; IK += 1) {
           if( PN[I]>POTER[K][IK] )
             {  MTR[K][ND]=AB[K][IK];
              if( IK>1 ) A1[K]=AB[K][IK-1];
               else   {
                    if( KFT==1 ) goto SMBIX;
                    KFT=1;A1[K]=AB[K][IK];
                    }
              if( KFT==1 )
                {
                  PKE=-PKE;PNDT1=PNDTZ+PKE;
fprintf(print,"\n\nПопытка восстановить заданные условия
                  по напору на ");
fprintf(print,"\nнапору на вершине на данном сортаменте
                 не удалося");
fprintf(print,"\nесть возможность попытки увеличить  VMAX
```

```
                и выполнить расчет.");
                  goto  SMIF;
                      }
        SMBIX:  if( IK>1 ) A1[K]=AB[K][IK-1];
                else  A1[K]=AB[K][IK];
                MTR[K][ND]=AB[K][IK]; goto  SM195871;
                    }
            } /**/
      SM195871:; }
    } /* SM195870*/
 } /* SM19587*/
  bixod(IPOGAR,&NUCH);
if( REGP==0 )
 fprintf(print,"\n\nМинимальный потербный напор=%g",
                HAP[1]);
 else   {
     fprintf(print,"\n\nНапор на вершине сети=%g",PNDT1);
     PRIW=ALFAF*POTER[1][1]+BETAB*RASHV*PNDT1;
     fprintf(print,"\n\nПриведенные затраты на сеть=%g",
                    PRIW);
      }
if( REGP==0 && IPOGAR==1 )
      { IPOGAR=2;
        for(i=1;i<kuch;i++) strcpy(TRM2[i],TRM[i]);
        fprintf(print,"\n\nПроводится расчет на
                          хозяйственный режим."),
         goto  MVIBPS.
      }
if( IPOGAR==1 &&  REGP==1 ) {  IHAP=0;IOGR=0,
```

```
    fprintf(print,"\n\nПроизводится расчет на мак.хоз.
                  режим водопотребления с выбранными");
    fprintf(print,"\nдиаметрами на сети.");
    fprintf(print,"\n\nУчастки на которых были уложены
                  два диаметра делятся");
    fprintf(print,"\nна два участка введением
                              дополнительных вершин.");
    fprintf(print,"\n\nУчастки с дополнительными
                        вершинами :");
    NZ=MTR[1][6];NCH=MTR[1][5];KN=NZ;
    for( IP = 1 ; IP <=  NUCH; IP += 1)
     {
      if( MKIN[IP][2]>NZ )
       { KN=KN+1;
         NCH=NCH+1;
         fprintf(print,"\n
    %3d -%3d вершина %3d ",MKIN[IP][1],MKIN[NCH][2],KN);
          }
      }
     for(i=1;i<kuch;i++) strcpy(TRM2[i],TRM[i]);
     REGP=0;IPOGAR=2;
     for( I = NUZ+1 ; I <=  NUCH-NKON+1; I += 1)
       Q[I]=HSB[I]=VIS[I]=0;
     NUZ=NUCH-NKON+1;
      goto  MVIBPS;
}
 /*---блок расчета отключения аварийных участков------*/
  if( IAV==1 )
   {       if ( KOLAV==0 )
```

```
        {
          for(i=1;i<kuch;i++)
          {
            if(i<kuz) Q[i]=Q[i]*0.7;
           AB[i][1]=MKIN[i][1];AB[i][2]=MKIN[i][2];
          DLIN1[i]=DLIN[i];PN1[i]=TRR[i];strcpy(TRMM[i],
           TRM2[i]);
          AB[i][3]=KOD[i];
          }
           KUCH1=NUCH;KUZL=NUZ;
        }
SMKIN00:  for(i=1;i<kuch;i++)
          {
             MKIN[i][1]=AB[i][1];MKIN[i][2]=AB[i][2];
             DLIN[i]=DLIN1[i];TRR[i]=PN1[i];
             strcpy(TRM2[i],TRMM[i]);
             KOD[i]=AB[i][3];
           }
             NUCH=KUCH1;KUCH=KUCH1;
             NUZ=KUZL;NZ=MTR[1][6];
             for( I = NUZ+1 ; I <=  NUCH-NKON+1; I += 1)
             Q[I]=HSB[I]=VIS[I]=0;
        KOLAV=KOLAV+1;
        if( NAVU[KOLAV][1]==9999 )  goto  MKON00;
         isavu(&NUCH,KUCH,NZ,&NUZ,KOLAV);
        KK=0;
         naprav(&IAV,&KK);
    if(IPECH==1)
    {    fprintf(print,"\n после NAPRAV NUCH=%3d",NUCH);
```

```
    for(i=1;i<=NUCH;i++)
    fprintf(print,"\n%3d-%3d ",MKIN[i][1],MKIN[i][2]);
   }
        if( KK==1 )  goto  SMKIN00;
        IPOGAR=3;REGP=0;
        if( NAVU[KOLAV][1]>0 )  goto  MVIBPS;
      }
 MKON00:;
 fprintf(print,"\n\n\n\n\n\n\nАвтор программы ст.пр.
            отд.Системотехники и Управления КБГУ");
 fprintf(print," Байрактаров Б.Р.");
 exit(1);
}
#include<math.h>
#include<stdio.h>
#include "params.h"
void alfa(int NKON,int NUCH)
  {
/*****************************************************************/
/*        Подпрограмма решения систем уравнений          */
/*****************************************************************/
extern FILE *print;
extern  int   MKIN[kuch][3],KODM[ktrub],NSOR,ND,NP,KPT,KR,
              KOD[kuch],MTR[kuch][nd];
extern  float DLIN[kuch],RASH[kuch],TRR[kuch],
              HSB[kuch],VIS[kuch],PN[kuch],DLIN1[kuch],
              RASH1[kuch],HAP[kuch],PN1[kuch],
              S[kuch],Q[kuz], QP[kuz],QS[kuz],
               DELTAH[kkon];float P ,X,M[kkon][kkon+1];
```

```
int i,j,I,J,K,N,/*B[kkon][kuch];*/B[2][kuch];
FILE *matkon,*matkon1;N=NKON;
if(kkon==N+1)
{  printf("\nКоличество контуров на сети(%d) должно
            быть меньше,чем %d",N,kkon-1);
  getch();exit(1);}
          matkon=fopen("matkon.","r");
  C0065:for( I = 1 ; I <=  NKON; I += 1)
        {
          for(K=1;K<=NUCH;K++) fscanf(matkon," %d ",
                                  &B[0][K]); /*B[I][K]*/
          matkon1=fopen("matkon.","r");
    C0066:for( J = 1 ; J <=  NKON; J += 1)
          {
           M[I][J]=0;
      C0067:for( K = 1 ; K <=  NUCH; K += 1)
           {
              fscanf(matkon1," %d ",&B[1][K]); /*B[J][K]*/
            P=2*B[0][K]*S[K]*fabs(RASH[K]);
            M[I][J]=M[I][J]+P*B[1][K];
            } /* C0067*/
          } /* C0066*/
          fclose(matkon1);
        } /* C0065*/
        fclose(matkon);
  C0001: for( I = 1 ; I <=  N; I += 1)
          M[I][N+1]=DELTAH[I]; /* C0001*/
  C0002: for( K = 1 ; K <=  N; K += 1)
        {          if( M[K][K]==0 )
```

```
        {
C0009: for( J = K+1 ; (J <=  N)  && (M[J][K]==0);J += 1);
        /* C0009*/
       if( J==N+1 )  goto M0001;
C0010: for( I = K ; I <=  N+1; I += 1)
       {
        P=M[K][I]; M[K][I]=M[J][I]; M[J][I]=P;
       } /* C0010*/
     }
C0003: for( I = K+1 ; I <=  N; I += 1) {
      X=-M[I][K]/M[K][K];
C0004: for( J = K ; J <=  N+1; J += 1) {
      P=M[K][J]*X;M[I][J]=M[I][J]+P;
       } /* C0004*/
       } /* C0003*/
       } /* C0002*/
C0005: for( I = 1 ; I <=  N; I += 1)  DELTAH[I]=M[I][N+1];
     P=M[1][1];
C0006: for( K = 2 ; K <=  N; K += 1)  P=P*M[K][K];
      if( P==0 )  goto M0001; X=DELTAH[N]/M[N][N];
      DELTAH[N]=X;
C0007: for( I = N-1 ; I >=  1; I +=  -1 ) { X=DELTAH[I];
C0008: for( J = I+1 ; J <=  N; J += 1) {
      X=X-M[I][J]*DELTAH[J];  } /* C0008*/
       DELTAH[I]=X/M[I][I];
       } /* C0007*/ goto M0002;
M0001:printf("Система не разрешима,главный
                 определитель=0"),
getch();exit(1);M0002:  C0073:for( I = 1 ; I <=  NUCH; I += 1)
```

```
        {
           if(I<=NKON)  RASH[I]=RASH[I]+DELTAH[I];
           else
           { P=0; matkon=fopen("matkon.","r");
             for(J=1;J <= NKON;J+=1)
             {
              for(K=1;K<=NUCH;K++)
         fscanf(matkon," %d ",&B[0][K]); /*B[J][K]*/
              P=P+B[0][I]*DELTAH[J];
              }
              fclose(matkon); RASH[I]=RASH[I]+P;
            }
           PN[I]=S[I]*fabs(RASH[I])*RASH[I];
         } /* C0073*/
M0003:;
}
#include<math.h>
#include<string.h>
#include<stdio.h>
#include<process.h>
#include"params.h"
   /*****************************************************/
   /*     Подпрограмма построения дерева сети и    */
   /*        формирования матрицы контуров         */
   /*****************************************************/
void derewo(int *NKON,int NUCH)
{ extern int   MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],
               NSOR,ND,NP,KPT,KR,
               KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch];
```

```
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
              TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
              DLIN1[kuch],RASH1[kuch],HAP[kuch],PN1[kuch],
              Q[kuz],TR[ktrub][np],ST[ktrub][np],
              REGP, SN,VK, RR;
extern char  TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
extern FILE *print; FILE *matkon;
int I,J,i,j,KK,K,MKOD,JJ,II,NKON1;
float PKP,NN,MM,PN11;char MAT[5]; NKON1=0;
  C0040:for( I = 2 ; I <=  NUCH; I += 1)
       { KK= 0;
    C0041:for( J = 1 ; J <=  I-1; J += 1)
            if( MKIN[I][2]==MKIN[J][2] ) KK=1; /* C0041*/
        if( KK==1 )
  CD020:  { NKON1=NKON1+1;
          II=MKIN[I][1];MKOD=KOD[I];PN11=PN[I];PKP=RASH[I];
          JJ=MKIN[I][2]; MM=DLIN[I];NN=TRR[I];
          strcpy(MAT,TRM2[I]);
  C0042: for( J = I ; J >=  NKON1+1 ; J --)
        {
       MKIN[J][1]=MKIN[J-1][1];MKIN[J][2]=MKIN[J-1][2];
       PN[J]=PN[J-1];RASH[J]=RASH[J-1];
       DLIN[J]=DLIN[J-1];KOD[J]=KOD[J-1];
       strcpy(TRM2[J],TRM2[J-1]);TRR[J]=TRR[J-1];
        } /* C0042*/
         MKIN[NKON1][1]=II; strcpy(TRM2[NKON1],MAT);
       MKIN[NKON1][2]=JJ;PN[NKON1]=PN11;RASH[NKON1]=PKP;
       DLIN[NKON1]=MM;KOD[NKON1]=MKOD; TRR[NKON1]=NN;
   } /* CD020*/     } /* C0040*/
```

```
KK=NUCH-NKON1+1;*NKON=NKON1;matkon=fopen("mkin.","w");
fprintf(matkon," \n%d %d",NUCH,KK);
for( I = 1 ; I <=  NUCH; I += 1)
{ fprintf(matkon,"\n %g ",RASH[I]);
 if(I<=KK) fprintf(matkon," %g",Q[I]);
 for( J = 1 ; J <=  2; J += 1)
  {
   fprintf(matkon," %d ",MKIN[I][J]);
  }
}
fclose(matkon);
spawnl(P_WAIT,"matkon.exe",NULL);
}

#include<math.h>
#include<string.h>
#include<stdio.h>
#include"params.h"
   /*****************************************************/
   /* Подпрограмма формирования участков сети     */
   /*****************************************************/
void isavu(int *NUCH,int KUCH,int NZ,int *NUZ,int KOLAV)
 { extern int   MKIN[kuch][3],AB[kuch][kpt],
               KODM[ktrub],NSOR,ND,NP,KP,
               KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch],
               NAVU[20][10];
  extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
               TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
               DLIN1[kuch],RASH1[kuch],
```

```
                HAP[kuch],PN1[kuch],
                TR[ktrub][np],ST[ktrub][np],
                REGP, SN,VK, RR;
 extern char  TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
 int MKINZ[kuch][2],J,I,JP,j,II,JJ,K,NZ1,Nch1,L,JK,KI,JL;
extern FILE *print;
FILE *mkinz;
mkinz=fopen("mkinz","r");
Nch1=*NUCH;
for(I=1;I<=Nch1;I++)
 fscanf(mkinz,"%d %d",&MKINZ[I][0],&MKINZ[I][1]);
fprintf(print,"\n\n\nРасчет сети на случай аварии :
вариант  %3d",KOLAV);
fprintf(print,"\nИсключаемые участки:");
     CJ00:for( J = 1 ; (J <  10)&&( NAVU[KOLAV][J]>0);
              J += 1)
           {
             K=NAVU[KOLAV][J];
             fprintf(print,"\n
                     %3d-%3d",MKINZ[K][0],MKINZ[K][1]);
      CI00:for( II = 1 ; II <=  KUCH; II += 1)
            {
 if ( MKINZ[K][0]==MKIN[II][1] && MKINZ[K][1]==MKIN[II][2]
 || MKINZ[K][1]==MKIN[II][1] && MKINZ[K][0]==MKIN[II][2] )
             {
              Nch1=Nch1-1;
              for( JJ = II ; JJ <=  KUCH-1; JJ += 1)
               {MKIN[JJ][1]=MKIN[JJ+1][1];
                MKIN[JJ][2]=MKIN[JJ+1][2];
```

```
                DLIN[JJ]=DLIN[JJ+1];TRR[JJ]=TRR[JJ+1];
                KOD[JJ]=KOD[JJ+1];
                strcpy(TRM2[JJ],TRM2[JJ+1]);
               }
               MKIN[JJ][1]=0; MKIN[JJ][2]=0;
               DLIN[JJ]=0;TRR[JJ]=0;KOD[JJ]=0;KUCH=Nch1;
             }
           if( MKIN[II][2]>NZ )
           {
for( JK = 1;(JK<=KUCH) && (MKIN[II][2]!=MKIN[JK][1]);
            JK += 1);
if ( MKIN[II][1]==MKINZ[K][0] &&
MKIN[JK][2]==MKINZ[K][1]  ||
MKIN[II][1]==MKINZ[K][1] && MKIN[JK][2]==MKINZ[K][0] )
            {
                Nch1=Nch1-2;KI=II;if( JK>II ) KI=JK;
                for( JP = 1 ; JP <=  2; JP += 1)
           {
           for( JL = KI ; JL <=  KUCH-1; JL += 1)
            {
    for(j=1;j <=2; j++) MKIN[JL][j]=MKIN[JL+1][j];
             DLIN[JL]=DLIN[JL+1];
             TRR[JL]=TRR[JL+1];KOD[JL]=KOD[JL+1];
            }
            KI=II;
            if( JK<II ) KI=JK;
           }
              for( JP = Nch1+1 ; JP <=  KUCH; JP += 1)
              { for(j=1;j <=2; j++) MKIN[JP][j]=0;
```

```
            DLIN[JP]=0;TRR[JP]=0;KOD[JP]=0;
                }
              KUCH=Nch1;
            }
          }
        } /* CIOO*/
    } /* CJOO*/
     NZ1=NZ;
     for( I = 1 ; I <=  Nch1; I += 1)
      {
       if( MKIN[I][1]>NZ )
       { NZ1=NZ1+1;
         for( L = 1 ; L <=  Nch1; L += 1)
          if( MKIN[I][1]==MKIN[L][2] ) MKIN[L][2]=NZ1;
         MKIN[I][1]=NZ1;
        }
      }
       *NUCH=Nch1;*NUZ=NZ1;/*НОВОЕ КОЛИЧЕСТВО УЗЛОВ*/
}
#include<math.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
#include<stdlib.h>
#include"params.h"
#define n 50
   /*****************************************************/
   /* Подпрограмма оптимизации расщепленной сети  */
   /*****************************************************/
```

```
void  optimm(int N,float *NDT,int NUZ,
             float *PRIV,float RASHV,
            float EPS1)
{ extern int   MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],
             NSOR,ND,NP,KPT,KR,
             KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
             TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
             DLIN1[kuch],RASH1[kuch],HAP[kuch],PN1[kuch],
             TR[ktrub][np],ST[ktrub][np],
             OGRH,PNDTZ,REGP,DELHP, ALFAF, BETAB,
             SN,VK,RR;
extern char  TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
extern FILE *print;
float PRES,NPT1,KN[kuch],EFK[kuch],DVL[ktrub][np],
       SNS,HZAD,SP,PPP,
       P1,PP,PTT,QQ,P,X,XL,STM,PRIV1,
       SOS,PN2,EFK2,A2,PP2,P2,EFK1,RES,HAPM;
int   KOLN[50],/*MMT[kuch][kuch+1],*/
      KHZAD,J4,I1,KK,N1,J3,N2,LK,I2,I3,J2,
      i,j,I,J,J1,KF,REG,iii,kkk,
      K1,K2,I5,I4,J41,LL,P3,P4,K,II,JJ,KDELM;
char A[5];
FILE  *fil_dvl;
window(1,1,80,25); textbackground(0); clrscr();
PRES=EPS1;N1=1;N2=2;
KHZAD=0;SNS=0;HZAD=*NDT;
fil_dvl      = fopen ( "dvl.", "r" );
       for( I = 1 ; I <=  NSOR; I += 1)
```

```
        {
        for(J=1;J<=NP;J+=1)
          fscanf(fil_dvl," %g ",&DVL[I][J]);
        fscanf(fil_dvl,"\n");
        }
      fclose(fil_dvl);
  for( I = N2 ; I <=  N; I += 1)
  {
   J=MTR[I][ND];
   for( J1 = 1 ;AB[I][J1]!=J; J1 +=  1 );
   PN1[I]=POTER[I][J1];
   K=0;
   for( II = N2 ; II <=  N; II += 1)
   {
   /*     for( JJ = N2 ; (JJ <=  ND-1)  && (MTR[II][JJ]>0);
JJ += 1)
        if( I==MTR[II][JJ] )  { K=K+1;MMT[I][K]=II; }*/
   }
  }
  *NDT=0;
  for( I = N1 ; I <=  N; I += 1)
  {
    SP=0;
    for( J = 1 ; (J <=  ND-1) && (MTR[I][J]>1); J += 1)
    SP=SP+PN1[MTR[I][J]];
    PPP=HSB[I]+SP+VIS[I]-VIS[1];
    if( *NDT<PPP ) *NDT=PPP;
  }         *PRIV=10000000000;KDELM=500;
        if( NUZ<0 )  { KHZAD=1;NUZ=-NUZ; }
```

```
        for(i=1;i<kuch;i++) {MTR[i][0]=AB[i][0]=0;KN[i]=0;
A1[i]=0;HAP[i]=0;}
        if( KHZAD==1 )
         {  KDELM=1;
          if( *NDT<=HZAD ) *NDT=HZAD;
          else
            {
cprintf("\n\nЗаданный напор меньше,чем
минимальный потреб.напор" );
cprintf("\nсети %g  .Минимальный потреб.напо=%g",
HZAD,*NDT);
cprintf("\nНажмите на клавищу Enter.");getch();exit(1);
            }
         }
        KK=MTR[1][1];
        for( I = N1 ; I <=  N; I += 1) {
          SP=0;
          for(J= N1;(J<= ND) && (MTR[I][J]!=KK);J += 1)
         {
          K=MTR[I][J];
          SP=SP+PN1[K];
          }
        HAP[I]=*NDT-SP+VIS[KK]-VIS[I];
        if( I>NUZ ) HSB[I]=HAP[MKIN[I-NUZ][2]];
         }
  M1011:for( II = 1 ; II <=  KDELM; II += 1)
        {
        if( II==1 )  goto  M1012;
        *NDT=*NDT+DELHP; for( I = 1 ; I <=  N; I += 1)
```

```
        {
         HAP[I]=HAP[I]+DELHP;
         if( I>NUZ ) HSB[I]=HAP[MKIN[I-NUZ][2]];
         }
        if( *NDT > OGRH && OGRH>0 )  goto  M4555;
M1012:for( I = N2 ; I <=  N; I += 1)
        {
         EFK[I]=-1;A1[I]=MTR[I][ND];J2=A1[I];
         J4=V[J2];
         for( K = 1 ;AB[I][K]!=J2; K +=  1 ) ;
        PN1[I]=POTER[I][K];
        for( J = 1 ; J <=  K-1; J += 1)
       {
         J1=AB[I][J]; PN2=POTER[I][J]; EFK2=0;
         A2=ST[J2][J4]-ST[J1][1];  P2=PN2-PN1[I];
         if( P2>0 ) EFK2=DLIN1[I]*A2/P2;
         if( EFK[I]<EFK2 )
         {
           EFK[I]=EFK2;  A1[I]=AB[I][J];
          }
         }
        }
 M1020:for( I = N1 ; I <=  N; I += 1)
        {
         if( EFK[I]<-1.1 ) EFK[I]=-EFK[I];
         if( I>NUZ ) HSB[I]=HAP[MKIN[I-NUZ][2]];
        }
SML00:EFK1=-1000000000;I2=0;
        for( I = N2 ; I <=  N; I += 1)  {
```

```
        if(  EFK[I]!=-1 && EFK1<EFK[I] )
          { EFK1=EFK[I];I2=I; }
       }
         J2=A1[I2]; RES=1000;
         for( I = N2 ; I <=  N; I += 1)
         {
 for( J = N1 ;(J <= ND-1)&&(MTR[I][J]>0); J += 1)
         if( I2==MTR[I][J] )
            { I1=I ;
              HAPM=HAP[I1]-HSB[I1];
              if( RES>HAPM ) RES=HAPM;
              break; }
       }
       if( RES<0 ) RES=0;
       if( RES<PRES )  {
if( EFK1>0 )  { EFK[I2]=-EFK[I2]; goto  SML00; }
         else   { EFK[I2]=-1; goto  M1018; }
                                  }
       for( K = 1 ;AB[I2][K]!=J2; K +=  1 ) { }
       PP2=POTER[I2][K]-PN1[I2];
     if( RES>PP2 )
     DG0001: {
              PN1[I2]=POTER[I2][K];EFK[I2]=-1;J4=1;
              for( J = 1 ; J <=  K-1; J += 1) {
                PN2=POTER[I2][J];EFK2=0;J3=AB[I2][J];
         A2=ST[J2][J4]-ST[J3][1];P2=PN2-PN1[I2];
         if( P2>0 ) EFK2=DLIN1[I2]*A2/P2;
         if( EFK[I2]<=EFK2 )
                { EFK[I2]=EFK2;A1[I2]=AB[I2][J];
```

```
            } M3000: ; } if( EFK[I2]>0 ) MTR[I2][ND]=J2;
           } /* DG0001*/ else  PN1[I2]=PN1[I2]+RES;
M1018: RES=0; for( I = N1 ; I <=  N; I += 1) {
   SP=0;for( J = N1 ; J <=  ND-1; J += 1) {
   K=MTR[I][J]; if( K==1 ) goto M1013;  SP=SP+PN1[K]; }
M1013:HAP[I]=*NDT-SP+VIS[1]-VIS[I];
if( I>NUZ ) HSB[I]=HAP[MKIN[I-NUZ][2]];
if( RES>HAP[I]-HSB[I] ) RES=HAP[I]-HSB[I];
 } *NDT=*NDT-RES; for( I = N2 ; I <=  N; I += 1)
     { HAP[I]=HAP[I]-RES; if( EFK[I]!=-1 ) goto M1020; }
      SOS=0; for( J = N2 ; J <=  N; J += 1)
       {if( A1[J]==0 ) A1[J]=MTR[J][ND];
        J1=A1[J]; J2=MTR[J][ND]; if( V[J1]==0 ) V[J1]=1;
 I5=MTR[J][2];J4=V[J2];J41=V[J1];
for( JJ = 1 ; JJ <= KPT; JJ += 1)
 { if( J1==AB[J][JJ] ) K1=JJ; if( J2==AB[J][JJ] ) K2=JJ; }
 P1=POTER[J][K1];P=POTER[J][K2];if( J2==J1 )  { X=0; goto
M4000; }  X=(P1-PN1[J])*DLIN1[J]/(P1-P); M4000:  P3=1;
for( LL = J4 ; LL <=  NP-1; LL += 1)
 if( HAP[I5]<DVL[J2][LL] ) { P3=LL; goto  MM999;  }
 MM999: P4=1;  P1=-P*X/DLIN1[J];
    for( LL = J41 ; LL <=  NP-1; LL += 1)
 { PP=HAP[J]+P1; if( PP<DVL[J1][LL] ){  P4=LL;goto MM500; }
 }  MM500:XL=DLIN1[J]-X;
      STM=ST[J2][P3]*X+ST[J1][P4]*XL; SOS=SOS+STM;
 }  PRIV1=ALFAF*SOS/1000+BETAB*RASHV**NDT+SNS;
      gotoxy(II,25);cprintf("▓");  if( *PRIV>PRIV1 )
      {    *PRIV=PRIV1;NPT1=*NDT; KK=II;
         for( I = N2 ; I <=  N; I += 1)
```

```
        { MTR[I][0]=MTR[I][ND]; AB[I][0]=A1[I]; KN[I]=PN1[I]; }
      }
  if( *PRIV<PRIV1 ) goto M4555;  M3666:; }
 if( *PRIV >=100000000 )  goto  MGL10;  M4555:*NDT=NPT1;
      for( I = N1 ; I <=  N; I += 1)
       {SP=0; A1[I]=AB[I][0]; MTR[I][ND]=MTR[I][0];
        PN1[I]=KN[I];
          for( J = N1 ; J <=  ND-1; J += 1)
        { K=MTR[I][J]; if( K==1 )  goto  MG000;  SP=SP+KN[K]; }
 MG000:HAP[I]=NPT1-SP+VIS[1]-VIS[I];
      }
      for( I = N2 ; I <=  N; I += 1)
      {  J1=MTR[I][ND];I4=MTR[I][2];for(J=1;J<=NP;J+=1)
       if (HAP[I4] > DVL[J1][J]) goto MG202;
printf("\n%3d-ом участке не выдерживается условие по
давлению",I);printf("\nдавление=%9.3f,а напор в
начале=%8.3f",DVL[J1][1],HAP[I4]);
printf("\nНажмите на клавищу Enter.");
 MG202:; }  KF=3;  if( KR==0 )  {
      for( I = 1 ; I <=  NSOR; I += 1) {
       if( strcmp(TRM[I],"C")==0 )  {
        for( J = 1 ; J <=  NP-1; J += 1) {
         ST[I][J]=ST[I][J]/KF;
        }    }      }   }
 goto  MMML;  MGL10:RR=0;  MMML: ;
 } /* OPTIMM*/
```

```
#include<stdio.h>
#include "params.h"
   /*****************************************************/
   /*    Подпрограмма печати выходных таблиц      */
   /*****************************************************/
void bixod(int IPOGAR,int *NUCH)
{ extern int   MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],
 NSOR,ND,NP,KPT,KR,KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch];
char GOST[ktrub][10],MT[ktrub][10];extern FILE *print;
int I,J,NKON,NUZ,I1,IN1,IN2,II,NZ,NCH,NCH1,KI,J1,J2,L1,L2,i;
float D1,D2,VD,X1,X2,P,P1,P3,P4,P5,P6,R1,R2,PALF,PN11,
SOS;FILE *fil_gost_mt;if(REGP>0.1)
{  fil_gost_mt = fopen ( "gost_mt.", "r" );
for ( I=1 ; I<=NSOR ; I++ )fscanf(fil_gost_mt,"%s
%s",&GOST[I],&MT[I]);}for( I = 1 ;MKIN[I][1] > 0; I +=  1 );
NCH1=I-1;for (i=1;i <= NCH1; i++)RASH[i]=1000*RASH[i];  NUZ=1;
for( I = 1 ; I <=  NCH1; I += 1){ for( J = 1 ; J <=  2; J += 1)
if (NUZ < MKIN[I][J]) NUZ=MKIN[I][J];} NKON=NCH1-NUZ+1;
  if( IPOGAR>0 )  {  NCH=NCH1;NZ=NUZ; for(i=1;i < kuch; i++)
strcpy(TRM2[i]," ");    } fprintf(print,"\n\n\n
Т      А      Б      Л      И      Ц      А          ");
fprintf(print,"\n\nВЫХОДНОЙ ИНФОРМАЦИИ К РАСЧЕТУ КОЛЬЦЕВЫХ СЕТЕЙ
"); fprintf(print,"\n\n-----------------------------------
-----------------------------------------------");
 fprintf(print,"\n|      |        |        |        |              |");
 fprintf(print,"\n|номер|        | длина | расход|
характеристика трубы  большего  диаметра   |");
 fprintf(print,"\n|уч-ка|участок| уч-ка |  на   |
```

```
-----------------------------------------------------------");
 fprintf(print,"\n¦      ¦        ¦        ¦ уч-ке ¦
диаметр¦ материал ¦ длина ¦скорость ¦    потери ¦");
 fprintf(print,"\n¦      ¦        ¦  (м)  ¦ (л/с) ¦
(м)  ¦ и марка  ¦  (м)  ¦ (м/сек) ¦ напора(м)¦");
 fprintf(print,"\n¦-----------------------------------
-----------------------------------------------------");
 fprintf(print,"\n¦  1  ¦    2  ¦   3   ¦   4   ¦   5   ¦
6    ¦   7   ¦    8    ¦     10   ¦");
 fprintf(print,"\n-----------------------------------------------------");
IN1=MKIN[I][1];IN2=MKIN[I][2];J=I;if( IPOGAR==0 ¦¦ IPOGAR==1 )
I1=I; DLIN1[I]=DLIN[I];D1=DLIN1[I];P=fabs(RASH[I])/1000;
if( TRR[I]>0 ) VD=4*P/(3.14*TRR[I ]*TRR[I]);
if( IPOGAR>1 ) { NUZ1=MTR[1][6]; if(MKIN[I][2]>NUZ1 )
for( II = 1 ; II <=  NCH1; II += 1) { if(MKIN[I][2]==MKIN[II][1])
{ D1=DLIN[II]+D1;IN2=MKIN[II][2]; }} if( MKIN[I][1]>NUZ1 )  goto
MC0085;I1=I1+1;}fprintf(print,"\n¦ %3d ¦%3d-%3d¦%7.1f¦%7.2f¦
%6.3f¦ %6s  ¦%7.2f¦ %6.3f  ¦%9.6f ¦", I1,IN1,IN2,
D1,RASH[J],TRR[J ],TRM[J], DLIN1[J ], VD,  PN[J] );
  MC0085:; } /* C0085*/}        if( REGP >0.1 ) {
for( I = 1 ; I <=  NCH1; I += 1){ if( I<=NKON ) KI=NUZ+I; else
KI=MKIN[I][2]; L1=MTR[KI][ND];L2=A1[KI];
R1=POTER[KI][J1];R2=POTER[KI][J2];
if( R1!=R2 ){ if( PN[I]<0 )  { R2=-R2;R1=-R1; }
if( PN[I]>R2 ) PN[I]=R2; if( PN[I]<R1 ) PN[I]=R1;
PALF=(PN[I]-R2)/(R1-R2);if (PALF<-0.1)
if (PALF < 0.00000001) PALF=-PALF;/*10.01.92*/
X1=PALF*DLIN[I];   }else  X1=DLIN[I];PN11=X1*R1/DLIN[I];
D1=TR[L1][1];VD=4*RASH[I]/(3.14*D1*D1);if( X1==0 ) VD=0;
```

```
VD=VD/1000; if( IPOGAR>=1 )
{ if( fabs(X1-DLIN[I])>0.5 && X1>0.1    )
{  NCH=NCH+1;NZ=NZ+1;MTR[NCH][1]=NZ; MTR[NCH][2]=MKIN[I][2];
MTR[I][2]=NZ;  MTR[I][1]=MKIN[I][1];DLIN1[NCH]=DLIN[I]-X1;
DLIN1[I]=X1;strcpy(TRM2[I],TRM[L1]); KOD[I]=KODM[L1];
TRR[I]=TR[L1][1];TRR[NCH]=TR[L2][1];strcpy(TRM2[NCH],TRM[L2]);
KOD[NCH]=KODM[L2]; } if( fabs(X1-DLIN[I]) <=0.1 )
{TRR[I]=TR[L1][1];strcpy(TRM2[I],TRM[L1]);
DLIN1[I]=DLIN[I]; KOD[I]=KODM[L1]; } if( X1<0.5      )
{ TRR[I]=TR[L2][1];strcpy(TRM2[I],TRM[L2]),
            MTR[I][1]=MKIN[I][1];MTR[I][2]=MKIN[I][2];
            DLIN1[I]=DLIN[I];  KOD[I]=KODM[L2];   }  }
fprintf(print,"\n| %3d |%3d-%3d|%7.1f|%7.2f| %6.3f|%3s,%5s
          |%7.2f| %6.3f  |%9.6f |",I,MKIN[I][1],MKIN[I][2],
            DLIN[I],RASH[I],D1,TRM[L1],MT[L1], X1,VD,PN11);
      } /* pE~ATx ХАРАКТЕРiСТiК bOlx{OgO diAMETPA */ }
 fprintf(print,"\n|        |   характеристика  трубы  меньше
го  диаметра    | потери   |стоимость|");
 fprintf(print,"\n|        |-------------------------------
----------------|   на     |   на    |");
 fprintf(print,"\n|участок| диаметр |материал|  длина  |ско
рость|  потери  | участке  | участке |");
 fprintf(print,"\n|        |   (м)   |        |   (м)   | (м
/сек)|  напора  |   (м)    |  (руб)  |");
fprintf(print,"\n-----------------------------------------
-------------------------------------");
 fprintf(print,"\n|  11   |    12   |   13   |    14   |
15   |    16    |   17     |    18   |");
fprintf(print,"\n-----------------------------------------------");
```

```
if( REGP < 1 ) {  C0086:for(I=1;I<= NCH1;I+= 1) {P1=0;
P3=0;P4=0;strcpy(P2,"       ");        P5=PN[I ]; VD=0;
IN1=MKIN[I][1];IN2=MKIN[I][2];if(    IPOGAR>1 )     {     if(
MKIN[I][2]>NUZ1 )   { P5=PN[I];for( II = 1;II <= NCH1;II+=1)
    { if( MKIN[II][1]==MKIN[I][2] ) {
    P1=TRR[II];strcpy(P2,TRM[II]); P3=DLIN[II];IN2=MKIN[II][2];
if(P1>0 )VD=4*fabs(RASH[II])/(3.14*P1*P1*1000);
P4=PN[II];P5=P5+PN[II];  } } }
if( MKIN[I][1]>NUZ1 ) goto MC0086;}
fprintf(print,"\n¦%3d-%3d¦ %7.3f ¦ %5s ¦ %7.2f ¦  %6.2f  ¦%9.5f
¦%9.5f ¦ ¦", IN1,IN2,P1,P2,P3,VD,P4,P5);
MC0086:; } /* C0086*/}
if( REGP >0.1 )
{ SOS=0.for(I=1;I<= NCH1;I += 1)
{  if(  I<=NKON  )  KI=NUZ+I;  else
KI=MKIN[I][2];   L1=MTR[KI][ND];L2=A1[KI];
for(J1 = 1;AB[KI][J1]!=L1; J1 += 1 ) ;
for(J2=1 ;AB[KI][J2]!=L2; J2 +=1) ;
R1=POTER[KI][J1];R2=POTER[KI][J2];
if(  PN[I]<0  )   {R2=-R2;R1=-R1;}
if(R1!=R2  )
{PALF=(PN[I]-R2)/(R1-R2);X1=PALF*DLIN[I];}
else X1=DLIN[I];X2=DLIN[I]-X1;
STT=ST[L1][1]*X1+ST[L2][1]*X2;
SOS=SOS+STT; D2=TR[L2][1];VD=4*RASH[I]/(3.14*D2*D2);
if(X2==0) VD=0; VD=VD/1000;PN2=X2*R2/DLIN[I];
fprintf(print,"\n¦%3d-%3d¦ %7.3f ¦%3s,%4s¦ %7.2f ¦ %6.2f¦ %8.4f
¦%10.6f    ¦%8.1f¦",MKIN[I][1],MKIN[I][2],   D2,TRM[L2],MT[L2],
X2,VD,PN2,  PN[I],STT);  } SOS=SOS/1000;POTER[1][1]=SOS; }
```

```
fprintf(print,"\n-----------------------------------------------
-----------------------------------------");
 fprintf(print,"\n-----------------------------------------------");
 fprintf(print,"\n|номер| расход |высотная|  напор  |пьезом
етр|"); fprintf(print,"\n|узла |  воды  | отметка| в узле  |
узла   |");
 fprintf(print,"\n|      | в узле |  узла  |   (м)   |   (м)
|"); fprintf(print,"\n|      |  (л/с) |  (м)   |         |   |");
 fprintf(print,"\n-----------------------------------------------");
     if( REGP < 1 && IPOGAR>1 ) NUZ=NUZ1;
     for( J = 1 ; J <=  NUZ; J += 1) { P1=HAP[J]+VIS[J],
fprintf(print,"\n| %3d | %7.2f| %6.2f | %7.2f | %7.2f |",
           J,Q[J]*1000,VIS[J],HAP[J],P1); }
 fprintf(print,"\n-----------------------------------------------");
      MTR[1][6]=NUZ;MTR[1][5]=NCH1;
  if( IPOGAR>=1 && REGP==1 )
  { for(i=1;i <= NCH1; i++) strcpy(TRM[i],TRM2[i]);
      *NUCH=NCH;NUZ=NZ; for( I = 1 , I <=  NCH, I += 1)
      { DLIN[I]=DLIN1[I];
        MKIN[I][1]=MTR[I][1];MKIN[I][2]=MTR[I][2]; }
   }    }
#include<stdio.h>
#include"params.h"
   /*************************************************/
   /* Подпрограмма проверки выполнения 2-го      */
   /*        закона Кирхгоффа                    */
   /*************************************************/
void novaja2(int *KK,int NKON,int NUCH,float EPS1)
{extern  float RASH[kuch],S[kuch],DELTAH[kkon],
```

```
REGP,DELHP,BETAB,ALFAF,PNDTZ,OGRH, SN,VK, RR;extern FILE
*print;FILE  *matkon;int I,J,B[kuch];matkon=fopen("matkon.","r");
  C0068:for( I = 1 ; I <=  NKON; I += 1)
          { DELTAH[I]=0;
  C0069: for( J = 1 ; J <=  NUCH; J += 1)
          { fscanf(matkon," %d",&B[J]);
           DELTAH[I]=DELTAH[I]+B[J]*fabs(RASH[J])*RASH[J]*S[J]; }
DELTAH[I]=-DELTAH[I];
if( fabs(DELTAH[I])>EPS1 ) *KK=0; }
fclose(matkon);}

#include<stdio.h>
#include"params.h"
   /*****************************************/
   /* Подпрограмма печати контуров и невязок     */
   /*     по этим контурам                       */
   /*****************************************/
void novaja3(int NKON,int NUCH){extern  int   MKIN[kuch][3];
*matkon;int B[kuch],I,J,K;float PP;matkon=fopen("matkon.","r");
for( I = 1 ; I <=  NKON; I += 1)
{ fprintf(print,"\n\n%3d-ый контур", I); K=0;
for( J = 1 ; J <=  NUCH; J += 1){ fscanf(matkon," %d ",&B[J]);
if( B[J]!=0 )  {  K=K+1;PP=PN[J]*B[J];fprintf(print,"\n%2d
%3d-%3d %10.6f", K,MKIN[J][1],MKIN[J][2],PP); } }
fprintf(print,"\n--------------------");
fprintf(print,"\nНевязка по потерям: %10.6f",DELTAH[I]);
}fclose(matkon);}
#include<stdio.h>
#include"params.h"
```

```
   /*****************************************/
   /* Подпрограмма построения  первоначального   */
   /*        направления на сети                 */
   /*****************************************/
void naprav(int *II,int *KK)
{extern int MKIN[kuch][3];extern FILE *print;
 int  NN[kuch][3],J,J1,I,N,J2,i,j;
for(i=1;i<kuch;i++)for(j=1;j<3;j++) NN[i][j]=0;
J1=0;for( I = 1 ;MKIN[I][1]>0; I +=  1 ){ if( MKIN[I][1]==1 )
{ J1=J1+1;NN[J1][1]=1;NN[J1][2]=MKIN[I][2]; }}N=I-1; for( I = 1 ;
I <=  N; I += 1){for( J = 1 ; J <=  N;J+= 1)
{ for( J2 = 1 ; J2 <=  J1; J2 += 1){  if( NN[J2][1]==MKIN[J][1] )
if( NN[J2][2]==MKIN[J][2] )  goto M1;if( NN[J2][2]==MKIN[J][1] )
if( NN[J2][1]==MKIN[J][2] ) goto  M1; }if( NN[I][2]==MKIN[J][1])
{J1=J1+1;NN[J1][1]=MKIN[J][1];NN[J1][2]=MKIN[J][2];  }
if( NN[I][2]==MKIN[J][2] ){J1=J1+1;NN[J1][1]=MKIN[J][2];
NN[J1][2]=MKIN[J][1];} M1:; } }if(J1!=N) if( *II==0 )if( *KK==0)
       { printf("\n\nЗаданная сеть имеет разрыв,проверьте
данные по участкам");*II=1;printf("\nНажмите любую клавищу.");
getch();exit(); }
if( J1!=N )
   if( *II==1 )
    if( *KK==1 )
        { printf("\n\nПосле отключения аварийных участков
данного варианта сеть");   printf("\nполучилась разрывной.Расчет
проводится для следующего варианта.");  *KK=0; }
   for( I = 1 ; I <=  N; I  += 1) for( J = 1 ; J <=  N; J += 1)
     { if( MKIN[I][1]==NN[J][2] &&  MKIN[I][2]==NN[J][1] )
       { MKIN[I][1]=NN[J][1];MKIN[I][2]=NN[J][2]; }  }  }
```

```
#include<stdio.h>
#include"params.h"
    /*****************************************************/
   /*   Подпрограмма ввода информации по узлов    */
   /*****************************************************/
void woduz(int *NUZ)
{ extern intMKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],NSOR,ND,NP,
KPT,KR,KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
TPR[kuch],HSB[kuch],VIS[kuch],PN[kuch], DLIN1[kuch],RASH1[kuch],
HAP[kuch],PN1[kuch],TR[ktrub][np],ST[ktrub][np],REGP,
Q[kuz],QP[kuz],QS[kuz], SN,VK, RR; extern char  TRM[kuch][5],
TRMM[kuch][5], TRM2[kuch][5]; extern FILE *print;  FILE *dannie;
int i,j,K,I,Nuz1,KK; char file[15];
extern char shifrobj[3]; strcpy(file,"uz1.");
      strcat(file,shifrobj);Nuz1=*NUZ;
      if((dannie=fopen(file,"r"))==NULL)
       { cprintf("\nИзвените, я не могу открыть файл- %s",
file);  getch();exit(1);        }
fscanf(dannie,"%d ",&KK);
for(i=1;i<kuch;i++) VIS[i]=HSB[i]=0;
if( Nuz1 < 0 ) K=0; else  K=1;
  C001:for(I=1;I<=KK;I++)
     { fscanf(dannie,"%d ",&j);
       fscanf(dannie,"%g ",&VIS[I]);
       fscanf(dannie,"%g ",&HSB[I]);
       if( K==0 ) fscanf(dannie,"%g ",&Q[I]);
       else  fscanf(dannie,"%g  %g ",&QS[I],&QP[I]);
   } /* C001*/
```

```
 M001:Nuz1=I-1:
fprintf(print,"\n          ТАБЛИЦА ИСХОДНОЙ ИНФОРМАЦИИ  ");
fprintf(print,"\n             ПО УЗЛАМ СЕТИ              ");
fprintf(print,"\n-------------------------------------------
----------------------");
fprintf(print,"\n|номер| высотная | свободный|  расход  |
сосредот. |  расход  |");
fprintf(print,"\n|узла | отметка  |  напор   |  в узле  |
расход  | на пожар |");
fprintf(print,"\n|     |   (м)    |   (м)    |  (л/с)   |
(л/с)   |  (л/с)   |");
fprintf(print,"\n-------------------------------------------");
fprintf(print,"\n|  1  |     2    |     3    |     4    |
5     |     6    |");
fprintf(print,"\n-------------------------------------------");
 C002:for(I=1;I<=Nuz1;I++)  {     if( K==0 )
fprintf(print,"\n| %3d | %8.2f | %8.3f | %8.3f ||          |",
I,VIS[I],HSB[I],Q[I]); if( K==1 )
fprintf(print,"\n| %3d | %8.2f | %8.3f |
| %8.3f | %8.3f |", I,VIS[I],HSB[I],QS[I],QP[I]);  }
fprintf(print,"\n-------------------------------------------");
*NUZ=Nuz1;
}
#include<stdio.h>
#include"params.h"
   /*****************************************************/
   /*  Подпрограмма ввода параметров сети          */
   /*****************************************************/
void wkoef(int IDH,int IALF,int IBET,int IHAP,int IOGR,
```

```
            int IUZZ,float *QPOS,float *EPS,float *EPS1,
            float *VMIN.float *VMAX)
{extern FILE *print;  FILE *dannie;
extern float OGRH. PNDTZ,DELHP,BETAB,ALFAF;,
        int i;  float QPOS1;
extern float REGP;extern char shifrobj[3];
        float epsilon1,epsilon,Vmin1,Vmax1,char  file[15];
       strcpy(file,"param."); strcat(file,shifrobj);
       if((dannie=fopen(file,"r"))==NULL)
        {
cprintf("\nИзвините, я не могу открыть файл - %s",file);
          getch();exit(1);        }
       if( lOGR==1 )
         {  fscanf(dannie,"%g ",&OGRH);
  fprintf(print,"\n-");for(i=1;i<36;i++)
fprintf(print,"-");
  fprintf(print,"\n|Ограничение на напор  |%8.3f|",
OGRH);}f( IHAP==1 ) {
fscanf(dannie,"%g ",&PNDTZ);
fprintf(print,"\n-");for(i=1;i<36;i++) fprintf(print,"-");
fprintf(print,"\n|Заданный напор на вер. |%8.3f|",PNDTZ);
}
 if( lALF==1 ){
fscanf(dannie,"%g ",&ALFAF);
fprintf(print,"\n-");for(i=1;i<36;i++) fprintf(print,"-");
fprintf(print,"\n|Коэф-нт ц.ф.кап.вложен. |%8.3f|",ALFAF);
}
 if( IBET==1 ) {   fscanf(dannie,"%g ",&BETAB);
fprintf(print,"\n-");for(i=1;i<36;i++) fprintf(print,"-");
```

```
fprintf(print,"\n¦Коэф-нт ц.ф.  энергозат. ¦%8.3f¦",BETAB);
  } if( IDH==1 )  {  fscanf(dannie,"%g ",&DELHP);
fprintf(print,"\n-");for(i=1;i<36;i++) fprintf(print,"-");
fprintf(print,"\n¦Шаг по напору              ¦%8.3f¦",DELHP);}
fscanf(dannie,"%g %g ",&epsilon,&epsilon1);
if (REGP >0.1){fscanf(dannie,"%g %g ",&Vmin1,&Vmax1);
*VMIN=Vmin1;*VMAX=Vmax1;}*EPS=epsilon;*EPS1=epsilon1;
fprintf(print,"\n-");for(i=1;i<36;i++) fprintf(print,"-");
fprintf(print,"\n¦ Невязка по расходам
¦%7.4f ¦",*EPS);fprintf(print,"\n-");
for(i=1;i<36;i++) fprintf(print,"-");
fprintf(print,"\n¦ Невязка по потерям
¦%7.4f ¦",*EPS1);fprintf(print,"\n-");
for(i=1;i<36;i++) fprintf(print,"-");
 if (REGP >0.1) {fprintf(print,"\n¦ Минимальная скор.
теч-ия ¦%7.4f ¦",*VMIN);fprintf(print,"\n-");
for(i=1;i<36;i++) fprintf(print,"-");
fprintf(print,"\n¦ Максимальная скор.
теч-ия¦%7.4f ¦",*VMAX);fprintf(print,"\n-");
for(i=1;i<36;i++) fprintf(print,"-");  }
   if( IUZZ==0 ) { fscanf(dannie,"%g ",&QPOS1);*QPOS=QPOS1;
fprintf(print,"\n¦ РАСХОД Н/П (л/с)        ¦%8.3f¦",QPOS1);
fprintf(print,"\n-----------------------------------------"); } }
#include<stdio.h>
#include"params.h"
    /*****************************************************/
   /* Подпрограмма ввода информации по участкам  */
   /*****************************************************/
void wmdr(int *NUCH)
```

```
{ extern int   MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],
               NSOR,ND,NP,KPT,KR,KOD[kuch],MTR[kuch][nd],
               A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
               TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
               DLIN1[kuch],RASH1[kuch],HAP[kuch],PN1[kuch],
               TR[ktrub][np],ST[ktrub][np],REGP,SN,VK,RR;
extern char  TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
extern FILE *print; FILE *dannie,*mkinz;
       int KNUCH,i,j,PPP,I;char file[15];
extern char shifrobj[3];strcpy(file,"uchas.");
      strcat(file,shifrobj);
      if((dannie=fopen(file,"r"))==NULL)
       {cprintf("\nИзвените, я не могу открыть файл - %s",file);
         getch();exit(1);  }
if((mkinz=fopen("mkinz.","w"))==NULL)
 {   printf("\nЯ не могу открыть файл mkinz.");
   getch();exit(1); }
 fprintf(print,"\n\n\nТАБЛИЦА ИСХОДНОЙ ИНФОРМАЦИИ");
 fprintf(print,"\nПО УЧАСТКАМ ");
      for(i=1;i<kuch;i++) {
       DLIN[i]=0;RASH[i]=0;KOD[i]=0;
       TRR[i]=0;strcpy(TRM[i]," ");
       for(j=1;j<3;j++)  MKIN[i][j]=0;      }
/*-если  NUCH=-1 то диаметры заданы^расходы нет,
   если NUCH=-2 диамеъры и расходы заданы  ,
   если NUCH=-3 диаметры и расходы не заданы,
   если NUCH--4 диаметры заданы ^расходы не заданы----*/
 fprintf(print,"\n-"); for(i=1;i<68;i++) fprintf(print,"-");
```

```
fprintf(print,
"\n¦Номер¦  Номер  ¦  Длина  ¦ Диаметр¦Материал¦ Расход
¦Коэффициент¦");fprintf(print,
"\n¦уч-ка¦ начала  ¦ участка ¦  трубы  ¦
трубы  ¦ участка ¦ наличия   ¦");fprintf(print,
"\n¦      ¦ и конца ¦   (м)   ¦   (м)   ¦
или код ¦   л/с   ¦пут.отбора ¦"); fprintf(print,"\n-");
 for(i=1;i<68;i++) fprintf(print,"-");
fprintf(print,"\n¦  1  ¦    2    ¦    3    ¦    4    ¦
5   ¦    6    ¦     7     ¦"); fprintf(print,"\n-");
 fscanf(dannie,"%d ",&PPP);if (PPP==kuch-1)
{fprintf(print,"\nУчастков на сети больше,
чем допустимо(%3d)",kuch-1);exit(1);getch();}
 for(i=1;i<68;i++) fprintf(print,"-");
  C001:for(I=1;I<=PPP;I++){fscanf(dannie,"%d ",&MKIN[I][1]);
fscanf(dannie,"%d ",&MKIN[I][1]);
fscanf(dannie,"%d %g ",&MKIN[I][2],&DLIN[I]);
fprintf(mkinz,"\n%d %d",MKIN[I][1],MKIN[I][2]);
if( *NUCH==-1 || *NUCH==-2 )
   fscanf(dannie,"%g %d" ,&TRR[I],&KOD[I]);
if( *NUCH==-2 || *NUCH==-4 ) fscanf(dannie,"%g ",&RASH[I]);
if( *NUCH==-1 || *NUCH==-3 ) fscanf(dannie,"%g ",&RASH[I]);
/*-BB0d K0¦f.p0TPEb. HA u~ACTKE--*/
  if( *NUCH==-1 ){
fprintf(print,"\n¦ %3d ¦ %3d-%3d ¦%8.2f ¦ %7.3f ¦  %4d  ¦
¦   %5.2f   ¦",  I,MKIN[I][1],MKIN[I][2],DLIN[I],
       TRR[I],KOD[I],RASH[I]);
       if( RASH[I]<0.000001 ) DLIN[I]=-DLIN[I];     }
   if( *NUCH==-2 )fprintf(print,"\n¦ %3d ¦ %3d-%3d ¦%8.2f ¦ %7.3f
```

```
¦  %4d  ¦%8.3f¦            ¦",I,MKIN[I][1],MKIN[I][2],DLIN[I],
TRR[I],KOD[I],RASH[I]);
   if( *NUCH==-3 ) { fprintf(print,"\n¦ %3d ¦ %3d-%3d ¦%8.2f ¦
¦         ¦          ¦   %5.2f   ¦",
I,MKIN[I][1],MKIN[I][2],DLIN[I], RASH[I]);
      if( RASH[I]<0.000001 ) DLIN[I]=-DLIN[I];
   }   if( *NUCH==-4 ) fprintf(print,"\n¦ %3d ¦ %3d-%3d ¦%8.2f ¦
¦           ¦ %8.3f¦            ¦",I,MKIN[I][1],MKIN[I][2],DLIN[I],
             RASH[I]);  }  /* C001*/
 M001:KNUCH=I-1;   *NUCH=KNUCH; fprintf(print,"\n-");
 for(i=1;i<68;i++) fprintf(print,"-");
fprintf(print,"\n\n\n");fclose(mkinz);}
```

```
/*************************************************/
/*   Подпрограмма расчета свободных напоров   */
/*                 в узлах сети               */
/*************************************************/
void pezom(int *NUZ,int NUCH)
{
 extern int   MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],
              NSOR,ND,NP,KPT,KR,
              KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
              TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
              DLIN1[kuch],RASH1[kuch],HAP[kuch],PN1[kuch],
              TR[ktrub][np],ST[ktrub][np],
              REGP, SN,VK,  RR;
 extern char  TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
int KK1,I,KK,KVER,N,IN,K,J,NUZ1;
 float SP;
extern FILE *print;
NUZ1=*NUZ;
/*fprintf(print,"pezom NUZ1=%d NUCH=%d",NUZ1,NUCH);
printf("pezom NUZ1=%d NUCH=%d",NUZ1,NUCH);
          for( I = 1 ; I <=  NUCH; I += 1)
            fprintf(print,"\n%3d--%3d ",MKIN[I][1],
                          MKIN[I][2]);*/
       KK1=1;
       if( NUZ1<0 )
         {
          NUZ1=-NUZ1;KK1=0;
         }
```

```
          KK=NUCH-NUZ1+1;N=KK+NUZ1;
         for( I = 1 ; I <=  NUCH; I += 1)
         {
           if  ( RASH[I]<0 )
        {
         RASH[I]=-RASH[I];
         K=MKIN[I][1];
         if (K!=1) { MKIN[I][1]=MKIN[I][2];MKIN[I][2]=K;}
        }
          }
         KVER=1;
C0050:for( I = 1 ; I <=   NUZ1; I += 1)
      {
       IN=I;  SP=0;
       for( K = 1 ;IN!=1; K +=  1 )
       {
         /* printf("K=%d IN=%d ",K,IN);
          fprintf(print,"K=%d IN=%d ",K,IN);*/
    for( J = 1 ;(J <=  NUCH)&&(MKIN[J][2]!=IN); J += 1);
          SP=SP+PN[J];IN=MKIN[J][1];
        }
    if( KK1==0 ) HAP[I]=HAP[KVER]-SP+VIS[KVER]-VIS[I];
    if( KK1==1 ) HAP[I]=HSB[I]+SP+VIS[I]-VIS[KVER];
/*  fprintf(print,"\npezom HAP[%d]=%8.3f ",I,HAP[I]);*/
      } /* C0050*/
*NUZ=NUZ1;
}
```

```
#include<math.h>
#include<string.h>
#include<stdio.h>
#include"params.h"
void rashep(int NUZ,int NUCH)
{
   /*****************************************************/
   /* Подпрограмма формирует матрицу траекторий  */
   /*     из любой вершины в корневую            */
   /*****************************************************/
 extern int   MKIN[kuch][3],AB[kuch][kpt],KODM[ktrub],
              NSOR,ND,NP,KPT,KR,KOD[kuch],MTR[kuch][nd],
              A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
              TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
              DLIN1[kuch],RASH1[kuch],HAP[kuch],
              PN1[kuch],TR[ktrub][np],ST[ktrub][np],
              REGP,SN,VK, RR;
 extern char  TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
extern FILE *print;
int N,i,j,IK,J2,KP[kkon],K,I,J,KK,I1,J1,KI,NKON;
NKON=NUCH-NUZ+1;
for(i=1;i<kuch;i++)
  for(j=1;j<nd;j++) MTR[i][j]=0;
        K=NUCH-NUZ+1;
        N=K+NUZ;  /*--KOli~ECTBO uzlOB-----------------*/
        for( I = 1 ; I <=  K; I += 1) {KP[I]=MKIN[I][2]; }
   C0001:for( I = 1 ; I <=  K; I += 1)
    {          MKIN[I][2]=NUZ+I;
```

```
  } /* C0001*/
        MTR[1][1]=1;
C0002: for( I = 1 ; I <=  NUCH; I += 1) {
        IK=MKIN[I][2];MTR[IK][1]=IK;
        MTR[IK][2]=MKIN[I][1];
  } /* C0002*/          KK=0;
C0003:for( I = 2 ; I <=  N; I += 1) {
 for( J = 1 ; J <=  ND-1  && MTR[I][J]!=0; J += 1) { }
       if( J==ND )
        {
      printf("\7\n Произошло переполнение матрицы MTR");
      printf("\nНажмите на кл. Enter");getch();exit();
        }
      I1=MTR[I][J-1];J1=J;
C0004:for( J = 2 ; J <=  ND  && MTR[I1][J]!=0; J += 1) {
       if( J==ND )
        {printf("\n\n Произошло переполнение матрицы MTR");
         printf("\nНажмите на кл. Enter");getch();exit();}
      MTR[I][J1]=MTR[I1][J];J1=J1+1;
  } /* C0004*/
} /* C0003*/
      for( I = 1 ; I <=  N; I += 1)
      {
       for( J = 1 ;MTR[I][J]!=0; J +=  1 );
        if( MTR[I][J-1]!=MTR[1][1] )
D0001: {
         KK=KK+1;
         if( KK<=NUCH )  goto  C0003;
         else    {
```

```
            for( KI = 1 ; KI <=  N; KI += 1)
            { printf("\n");
             for(J2=1;J2<3;J2++)
              fprintf(print,"I=%3d %3d ",KI,MKIN[KI][J2]);
             }
 printf("\n\n\nЗадание снимается!!!
                 Требуется корректировка п/п rashep" );
 printf("\nп/п не может построить матрицу траектории");
 fprintf(print,"\n\n\nЗадание снимается!!!
                    Требуется корректировка п/п rashep" );
fprintf(print,"\nп/п не может построить матрицу
                  траектории");
    printf("\nНажмите на кл. Enter");getch();exit(1);
          for( KI = 1 ; KI <=  N; KI += 1)
          { printf("\n");fprintf(print,"\n");
           for(J2=1;MTR[KI][J2]>0;J2++)
            printf(" %3d ",MTR[KI][J2]);
            fprintf(print," %3d ",MTR[KI][J2]);
           }
         }
      } /* D0001*/
      }
        /*_KOHEc pOCTPOEHiq MTR__*/
    for( I = 1 ; I <=  K; I += 1)  MKIN[I][2]=KP[I];

    for( I = 1 ; I <=  NUCH; I += 1)
      {
      if( I<=NKON ) K=NUZ+I; else  K=MKIN[I][2];
      if( I<=NKON )   {
```

```
          KK=MKIN[I][2];HSB[K]=HSB[KK];VIS[K]=VIS[KK];
          }
         DLIN1[K]=DLIN[I];RASH1[K]=RASH[I];
         }
}
#include<math.h>
#include<string.h>
#include<stdio.h>
#include"params.h"
   /*****************************************************/
   /* Подпрограмма формирования начального нап-  */
   /*        равления потоков                    */
   /*****************************************************/
void raspr(int NUCH,int NUZ)
{
 extern int    MKIN[kuch][3],AB[kuch][kpt],
               KODM[ktrub],NSOR,ND,NP,KPT,KR,
               KOD[kuch],MTR[kuch][nd],A1[kuch],V[kuch];
 extern float POTER[kuch][kpt],DLIN[kuch],RASH[kuch],
               TRR[kuch],HSB[kuch],VIS[kuch],PN[kuch],
               DLIN1[kuch],RASH1[kuch],HAP[kuch],PN1[kuch],
               TR[ktrub][np],ST[ktrub][np],Q[kuz],REGP,SN,
               VK, RR;
 extern char   TRM[kuch][5],TRMM[kuch][5],TRM2[kuch][5];
int i,j,JJ,K2,K1,J,I,KT,ITS[kkon],KRR,KK,I1,
float P;
        KRR=0;
        for(i=1;i<kkon;i++) ITS[i]=0;
        for(i=1;i<kuch;i++) RASH1[i]=0; MOOC:K1=0;
```

```
        for( J = 1 ; J <=  NUZ; J += 1)
        {
          KT=0;
         for( I = 1 ; I <=  NUCH; I += 1)
if( J==MKIN[I][1] ) KT=1;
         if( KT==0 )
           {
          K1=K1+1;ITS[K1]=J;
           }
         }
        for( I = 1 ; I <=  K1; I += 1) {
          J=ITS[I];KK=0;
          for( I1 = 1 ; I1 <=  NUCH; I1 += 1) {
           if( J==MKIN[I1][2] ) KK=KK+1;
           }
          for( I1 = 1 ; I1 <=  NUCH; I1 += 1) {
            if( J==MKIN[I1][2] )
             {
              RASH1[I1]=Q[J]/KK;
              if( RASH1[I1]==0 )
               {
printf("\nПервоначальное направление в узле %3d ",J);
printf("\nзадано не верно,так как расход в этом узле=0");
          .     exit(1);      }
             }
           }
        }
  C000:for( KRR = 1 ; KRR <=  2*NUCH; KRR += 1) {
  D001:for( I = 1 ; I <=  NUZ; I += 1) {
```

```
        KR=0;P=0;
        for( J = 1 ; J <=  K1; J += 1) {
         if( I==ITS[J] ) KR=1;
         }
        if( KR==0 )  { K2=0;
        for( JJ = 1 ; JJ <=  NUCH; JJ += 1) {
         if( I==MKIN[JJ][2] ) K2=K2+1;
         if( I==MKIN[JJ][1] )
           { if( RASH1[JJ]==0 )  goto  MK0;
             P=P+RASH1[JJ];
           }
         }
        P=P+Q[I];
        for( JJ = 1 ; JJ <=  NUCH; JJ += 1)
        {
         if( I==MKIN[JJ][2] )   RASH1[JJ]=P/K2;
        }
      }
 MK0: ; } /* D001*/
         KR=0;
   C0014:for( I = 1 ; I <=  NUCH; I += 1) {
          if( RASH1[I]==0 ) KR=1;
          } /* C0014*/          /*if( KR==0 ) KRR=1;*/
 } /* C000*/
     if( KR==1 )
    {  printf("\nНа сети обнаружен контур
не имеющий точки схода потоков");
     printf("\nпоэтому начальное распределение
потоков в некоторых участках");
```

```
    printf("\nправна нулю.Проверьте результаты расчета.");
    printf("\nПри программном прерывании исправьте
направление потоков и выполните расчет");
    }
}
#include<stdio.h>
#include"params.h"
  /*****************************************************/
  /* Подпрограмма ввода и печати информацион-   */
  /*       ной карты                            */
  /*****************************************************/
void pechin(int *IDZ,int *IUZZ,int *INRAS,
            int *IOGR,int *IHAP,
int *IPECH,int *IALF,int *IBET,int *IDH,int *IAV)
{
  extern FILE *print;
         FILE *dannie;
  int IDZ1,IUZZ1,INRAS1,IOGR1,IHAP1,
      IPECH1,IALF1,IBET1,IDH1,IAV1;
  char filerez[10],file[15];
  extern char shifrobj[3];
      strcpy(file,"begin.");
      strcat(file,shifrobj);
      if((dannie=fopen(file,"r"))==NULL)
       {
         cprintf("\nИзвините, я не могу открыть файл-
%s",file);
         getch();exit(1);
       }      strcpy(filerez,"rezul.");
```

```
      strcat(filerez,shifrobj);
       if((print=fopen(filerez,"w"))==NULL)
       {
         cprintf("\nИзвините, я не могу открыть файл  %s",
filerez);
         getch();exit(1);
       }
   fprintf(print,"\n\n\n
С А П Р - С К Г В Х ");
   fprintf(print,"\n\n  (II-Очередь)    ");
   fprintf(print,"\n\n
Решается задача гидравлического расчета кольцевых сетей");
fprintf(print,"\n\n\n
В Х О Д Н А Я   И Н Ф О Р М А Ц И Я\n\n");
      fprintf(print,"\n Шифр объекта-%s ",shifrobj);
      fscanf(dannie,"%d %d %d %d %d %d %d %d %d %d ",
               &IDZ1,&IUZZ1,&INRAS1, &IPECH1,&IAV1,
               &IOGR1,&IHAP1, &IALF1,&IBET1,&IDH1);
  if( IDZ1==0 )
fprintf(print,"\n   %d-диаметры на участках не заданы",
IDZ1);
  if( IDZ1==1 )
fprintf(print,"\n   %d-диаметры на участках  заданы",IDZ1);
if( IUZZ1==0 )
fprintf(print,"\n   %d-узловые расходы не заданы",IUZZ1);
  if( IUZZ1==1 )
fprintf(print,"\n   %d-узловые расходы  заданы",IUZZ1);
  if( INRAS1==0 )
fprintf(print,"\n   %d-расходы на участках не заданы",
```

```
INRAS1);  if( INRAS1==1 ) fprintf(print,"\n
%d-расходы на участках  заданы",INRAS1);
  if( IPECH1==0 ) fprintf(print,"\n
%d-промежуточная печать отключена",IPECH1);
  if( IPECH1==1 ) fprintf(print,"\n
%d-промежуточная печать включена",IPECH1);
  if( IAV1==0 )   fprintf(print,"\n
%d-аврийные участки не заданы",IAV1);
  if( IAV1==1 )   fprintf(print,"\n
%d-аврийные участки  заданы",IAV1);
  if( IOGR1==0 )  fprintf(print,"\n
%d-ограничение на напор не задан",IOGR1);
  if( IOGR1==1 )  fprintf(print,"\n
%d-ограничение на напор  заданы",IOGR1);
  if( IHAP1==0 )  fprintf(print,"\n
%d-напор на вершине не задан",IHAP1);
  if( IHAP1==1 )  fprintf(print,"\n
%d-напор на вершине задан",IHAP1);
  if( IALF1==0 )  fprintf(print,"\n
%d-коэффициент Алфа целевой функции не задан",IALF1);
  if( IALF1==1 )  fprintf(print,"\n
%d-коэффициент Алфа целевой функции задан",IALF1);
  if( IBET1==0 )  fprintf(print,"\n
%d-коэффициент Бета целевой функции не задан",IBET1);
  if( IBET1==1 )  fprintf(print,"\n
%d-коэффициент Бета целевой функции задан",IBET1);
  if( IDH1==0 )   fprintf(print,"\n
%d-шаг по напору не задан",IDH1);
  if( IDH1==1 )   fprintf(print,"\n
```

```
%d-шаг по напору  задан",IDH1);
  *IDZ=IDZ1;*IUZZ=IUZZ1;*INRAS=INRAS1;
*IOGR=IOGR1;*IHAP=IHAP1;
  *IPECH=IPECH1;*IALF=IALF1;
*IBET=IBET1;*IDH=IDH1;*IAV=IAV1;
}
#include<string.h>
#include"params.h"
   /*****************************************************/
   /*   Подпрограмма ввода и печати информации   */
   /*            по аварийным участкам            */
   /*****************************************************/
void pnavu(void)
{
extern int NAVU[20][10];
extern FILE*print;
FILE  *dannie;
 int i,j,I,J,K,KK;
      if((dannie=fopen("avar.dat","r"))==NULL)
       {
cprintf("\nИзвените, я не могу открыть файл avar.dat");
         getch();exit(1);
       }
for(i=0;i<20;i++)
  for(j=0;j<10;j++) NAVU[i][j]=0;
fprintf(print,"\n\n\n-----------
-------------------------------------------------------");
fprintf(print,"\n¦НОМЕР ¦
¦");fprintf(print,"\n¦ВАР-ТА¦
```

```
НОМЕРА АВАРИЙНЫХ УЧАСТКОВ                    ¦");
fprintf(print,"\n¦АВАРИИ¦
¦");
 for(I=1;I <=KK;I+=1)
 {   fscanf(dannie," %d",&NAVU[I][1]);
     fscanf(dannie," %d",&NAVU[I][1]);
     for( J = 2 ; J <  10; J += 1)
     {fscanf(dannie," %d",&NAVU[I][J]);
      if( NAVU[I][J]==0 )  goto  M001;}
   M001:fprintf(print,"\n¦  %3d ¦",I);
      for(K=1;(K< 10)&&(NAVU[I][K]>0);K++)
        fprintf(print,"%4d", NAVU[I][K]);
      for(i=1;i<58-4*K;i++) fprintf(print," ");
      fprintf(print,"¦");
}
M000: fprintf(print,"\n-----------------------------------------
   -----------------");
}
```

Приложение 8

Нормативно - справочная информация по сортаментам труб и насосным станциям

Таблица П.8.1

Нормативно-справочная информация по сортаментам труб и насосным станциям

Количество сортаментов 18

Шифр сортамента	Внешний диаметр	Толщина стенки	Стоимость 1 п.м.	ГОСТ	Марка	Материал
1	2	3	4	5	6	7
100	0.168	0.0020	0.57	33-17-82	-	С
		0.0022	0.64	33-17-82	-	С
		0.0027	0.85	33-17-82	-	С
		0.0032	1.08	33-17-82	-	С
		0.0037	1.33	33-17-82	-	С
	0.219	0.0020	0.85	33-17-82	-	С
		0.0022	0.96	33-17-82	-	С
		0.0027	1.29	33-17-82	-	С
		0.0032	1.66	33-17-82	-	С
		0.0037	2.06	33-17-82	-	С
	0.257	0.0020	1.13	33-17-82	-	С
		0.0022	1.24	33-17-82	-	С
		0.0027	1.65	33-17-82	-	С
		0.0032	2.16	33-17-82	-	С
		0.0037	2.68	33-17-82	-	С
	0.273	0.0027	1.93	33-17-82	-	С
		0.0032	2.52	33-17-82	-	С
		0.0037	3.14	33-17-82	-	С

Продолжение таблицы П.8.1

1	2	3	4	5	6	7
		0.0042	3.85	33-17-82	-	С
	0.325	0.0032	3.39	33-17-82	-	С
		0.0037	4.22	33-17-82	-	С
		0.0042	5.20	33-17-82	-	С
	0.377	0.0042	7.00	33-17-82	-	С
		0.0047	8.36	33-17-82	-	С
	0.426	0.0042	8.85	33-17-82	-	С
		0.0047	10.68	33-17-82	-	С
200	0.262	0.0180	9.79	31-7-77	ТСЦ	С
	0.325	0.0180	11.60	31-7-77	ТСЦ	С
300	0.273	0.0100	9.79	33-95-78	ТСА	С
	0.299	0.0100	10.10	33-95-78	ТСА	С
	0.325	0.0105	11.60	33-95-78	ТСА	С
400	0.313	0.0435	13.60	33-6-82	РТНС	С
	0.375	0.0435	14.60	33-6-82	РТНС	С
	0.478	0.0445	19.10	33-6-82	РТНС	С
	0.582	0.0485	24.10	33-6-82	РТНС	С
	0.686	0.0505	26.70	33-6-82	РТНС	С
500	0.160	0.0045	2.38	18599-83	Л	П
	0.180	0.0051	2.66	18599-83	Л	П
	0.200	0.0056	3.28	18599-83	Л	П
	0.225	0.0063	4.07	18599-83	Л	П
	0.250	0.0070	5.00	18599-83	Л	П
	0.280	0.0078	6.29	18599-83	Л	П
	0.315	0.0087	7.84	18599-83	Л	П

Продолжение таблицы П.8.1

1	2	3	4	5	6	7
	0.355	0.0098	9.88	18599-83	Л	П
	0.400	0.0110	12.40	18599-83	Л	П
	0.450	0.0123	15.60	18599-83	Л	П
	0.500	0.0137	19.30	18599-83	Л	П
	0.560	0.0153	24.20	18599-83	Л	П
	0.630	0.0172	30.00	18599-83	Л	П
	0.900	0.0244	61.49	18599-83	Л	П
	1.000	0.0271	75.57	18599-83	Л	П
	1.200	0.0328	107.83	18599-83	Л	П
600	0.160	0.0071	3.78	18599-83	СЛ	П
	0.180	0.0079	4.32	18599-83	СЛ	П
	0.200	0.0087	5.06	18599-83	СЛ	П
	0.225	0.0098	6.40	18599-83	СЛ	П
	0.250	0.0109	8.16	18599-83	СЛ	П
	0.280	0.0121	9.88	18599-83	СЛ	П
	0.315	0.0137	12.50	18599-83	СЛ	П
	0.355	0.0153	15.80	18599-83	СЛ	П
	0.400	0.0172	19.90	18599-83	СЛ	П
	0.450	0.0194	25.20	18599-83	СЛ	П
	0.500	0.0215	31.10	18599-83	СЛ	П
	0.560	0.0240	39.00	18599-83	СЛ	П
	0.630	0.0270	49.20	18599-83	СЛ	П
	0.710	0.0304	59.80	18599-83	СЛ	П
	0.800	0.0341	76.60	18599-83	СЛ	П
	0.900	0.0384	99.07	18599-83	СЛ	П

Продолжение таблицы П.8.1

1	2	3	4	5	6	7
	1.000	0.0426	120.89	18599-83	СЛ	П
	1.200	0.0502	174.05	18599-83	СЛ	П
700	0.160	0.0103	4.97	18599-83	С	П
	0.180	0.0115	6.20	18599-83	С	П
	0.200	0.0128	7.70	18599-83	С	П
	0.225	0.0144	9.70	18599-83	С	П
	0.250	0.0159	12.00	18599-83	С	П
	0.280	0.0168	15.00	18599-83	С	П
	0.315	0.0190	19.00	18599-83	С	П
	0.355	0.0224	23.80	18599-83	С	П
	0.400	0.0252	30.30	18599-83	С	П
	0.450	0.0283	38.00	18599-83	С	П
	0.500	0.0314	43.70	18599-83	С	П
	0.630	0.0395	69.27	18599-83	С	П
	0.710	0.0445	91.17	18599-83	С	П
	0.800	0.0501	109.51	18599-83	С	П
800	0.160	0.0163	7.50	18599-83	Т	П
	0.180	0.0183	9.35	18599-83	Т	П
	0.200	0.0203	11.50	18599-83	Т	П
	0.225	0.0230	14.60	18599-83	Т	П
	0.250	0.0253	18.00	18599-83	Т	П
	0.280	0.0283	22.60	18599-83	Т	П
	0.315	0.0318	27.01	18599-83	Т	П
	0.400	0.0401	43.21	18599-83	Т	П
	0.500	0.0503	66.98	18599-83	Т	П

Продолжение таблицы П.8.1

1	2	3	4	5	6	7
900	0.118	0.0075	4.25	9583-75	ЛА	Ч
	0.170	0.0083	6.54	9583-75	ЛА	Ч
	0.222	0.0092	9.38	9583-75	ЛА	Ч
	0.274	0.0100	12.26	9583-75	ЛА	Ч
	0.326	0.0108	15.85	9583-75	ЛА	Ч
	0.378	0.0117	19.43	9583-75	ЛА	Ч
	0.429	0.0125	23.79	9583-75	ЛА	Ч
	0.532	0.0142	32.92	9583-75	ЛА	Ч
	0.635	0.0158	42.74	9583-75	ЛА	Ч
	0.738	0.0175	53.85	9583-75	ЛА	Ч
	0.842	0.0192	65.35	9583-75	ЛА	Ч
	0.945	0.0206	78.44	9583-75	ЛА	Ч
1000	0.118	0.0083	4.25	9583-75	А	Ч
	0.170	0.0092	6.52	9583-75	А	Ч
	0.222	0.0101	9.30	9583-75	А	Ч
	0.274	0.0110	12.10	9583-75	А	Ч
	0.326	0.0119	15.60	9583-75	А	Ч
	0.378	0.0128	19.10	9583-75	А	Ч
	0.429	0.0138	23.60	9583-75	А	Ч
	0.532	0.0156	32.70	9583-75	А	Ч
	0.635	0.0174	43.40	9583-75	А	Ч
	0.738	0.0193	53.40	9583-75	А	Ч
	0.842	0.0211	64.90	9583-75	А	Ч
	0.945	0.0223	78.10	9583-75	А	Ч
1100	0.118	0.0090	4.25	9583-75	Б	Ч

Продолжение таблцы П.8.1

1	2	3	4	5	6	7
	0.170	0.0100	6.54	9583-75	Б	Ч
	0.222	0.0110	9.33	9583-75	Б	Ч
	0.274	0.0120	12.21	9583-75	Б	Ч
	0.326	0.0130	15.82	9583-75	Б	Ч
	0.378	0.0140	19.38	9583-75	Б	Ч
	0.429	0.0150	23.82	9583-75	Б	Ч
	0.532	0.0170	33.14	9583-75	Б	Ч
	0.635	0.0190	43.62	9583-75	Б	Ч
	0.738	0.0210	54.29	9583-75	Б	Ч
	0.842	0.0230	66.02	9583-75	Б	Ч
	0.945	0.0250	79.22	9583-75	Б	Ч
1200	0.108	0.0030	1.43	10704-76	ЭЛЕКТРОСВ.	С
	0.168	0.0050	3.49	10704-76	ЭЛЕКТРОСВ.	С
		0.0055	3.84	10704-76	ЭЛЕКТРОСВ.	С
		0.0060	4.10	10704-76	ЭЛЕКТРОСВ.	С
		0.0070	4.71	10704-76	ЭЛЕКТРОСВ.	С
		0.0080	5.35	10704-76	ЭЛЕКТРОСВ.	С
	0.180	0.0050	3.74	10704-76	ЭЛЕКТРОСВ.	С
		0.0055	4.11	10704-76	ЭЛЕКТРОСВ.	С
		0.0060	4.39	10704-76	ЭЛЕКТРОСВ.	С
		0.0070	5.05	10704-76	ЭЛЕКТРОСВ.	С
		0.0080	5.26	10704-76	ЭЛЕКТРОСВ.	С
	0.194	0.0050	4.04	10704-76	ЭЛЕКТРОСВ.	С
		0.0055	4.43	10704-76	ЭЛЕКТРОСВ.	С
		0.0060	4.74	10704-76	ЭЛЕКТРОСВ.	С

Продолжение таблицы П.8.1

1	2	3	4	5	6	7
	0.203	0.0050	4.24	–	–	–
		0.0055	4.64	–	–	–
	0.219	0.0050	4.57	–	–	–
	0.245	0.0050	5.13	–	–	–
		0.0055	5.61	–	–	–
	0.273	0.0050	5.73	–	–	–
		0.0055	6.29	–	–	–
	0.299	0.0050	6.28	–	–	–
		0.0055	6.87	–	–	–
	0.325	0.0050	6.84	–	–	–
		0.0055	7.48	–	–	–
	0.351	0.0050	7.40	–	–	–
		0.0055	8.08	–	–	–
		0.0060	8.64	–	–	–
	0.377	0.0050	7.96	–	–	–
		0.0060	9.29	–	–	–
		0.0070	10.70	–	–	–
	0.402	0.0050	8.44	–	–	–
		0.0055	9.22	–	–	–
		0.0060	9.88	–	–	–
	0.426	0.0050	9.71	–	–	–
		0.0055	9.78	–	–	–
	0.480	0.0050	10.61	–	–	–
		0.0080	15.99	–	–	–
		0.0090	17.58	–	–	–

Продолжение таблицы П.8.1

1	2	3	4	5	6	7
	0.530	0.0070	20.20		-	-
		0.0080	22.50	-	-	-
	0.630	0.0070	23.10	-	-	-
1300	0.600	0.0650	39.90	125860-83	ТН	Ж
	1.000	0.0750	84.80	-	ТН	Ж
	1.200	0.0850	121.00	-	ТН	Ж
1400	0.122	0.0090	0.98	539-80	ВТ-6	А
	0.168	0.0110	1.88	-	ВТ-6	А
	0.427	0.0230	7.71	-	ВТ-6	А
1500	0.122	0.0110	1.10	-	ВТ-9	А
	0.168	0.0135	1.94	-	ВТ-9	А
	0.274	0.0195	4.05	-	ВТ-9	А
	0.324	0.0225	5.38	-	ВТ-9	А
	0.528	0.0360	13.20	-	ВТ-9	А
1600	0.168	0.0135	2.28	-	ВТ-12	А
	0.224	0.0215	3.58	-	ВТ-12	А
	0.274	0.0200	4.62	-	ВТ-12	А
	0.324	0.0270	6.23	-	ВТ-12	А
1700	0.224	0.0220	6.62	-	ВТ-15	А
	0.274	0.0240	9.50	-	ВТ-15	А
	0.528	0.0460	29.98	-	ВТ-15	А
1800	0.292	0.0044	17.60	33РСФСР-80	ТЭР	Э

Таблица П.8.2

Нормативно-справочная информация по насосным станциям

a	b	H_{min}	H_{max}	Индекс насосной станции	Стои-мость	C_{min}	Q_{max}	K	Габа-риты	Марка
1	2	3	4	5	6	7	8	9	10	11
-421.990	486.386	76.0	94.8	201-0.8-114	105.12	0.48	0.88	2	18Х9	14Д-6
-394.222	427.995	76.0	94.1	202-0.72-84	103.07	0.44	0.80	2	18Х9	14Д-6
-254.307	275.297	88.0	93.0	203-0.68-65	95.58	0.40	0.82	2	18Х9	300Д-90
-261.833	261.413	87.0	92.5	204-0.71-42	94.14	0.36	0.76	2	18Х9	300Д-90
-255.603	237.500	86.0	92.0	205-0.56-48	94.54	0.32	0.72	2	18Х9	14НДС
-170.547	255.420	98.0	94.1	206-0.92-80	100.26	0.68	0.98	2	18Х9	14НДС
-178.338	233.237	87.0	92.0	207-0.80-70	96.01	0.60	0.92	2	18Х9	14Д-6
-192.308	328.673	76.0	94.8	301-1.20-114	123.61	0.72	1.32	2	24Х9	14Д-6
-175.209	285.330	76.0	94.8	302-1.08-98	122.11	0.66	1.26	3	24Х9	300Д-90
-113.257	183.581	88.0	93.0	303-1.02-65	112.00	0.60	1.23	3	24Х9	300Д-90
-116.370	174.279	87.0	92.5	304-1.96-56	109.84	0.54	1.14	3	24Х9	300Д-90

Продолжение таблицы П.8.2

1	2	3	4	5	6	7	8	9	10	11
-113.601	158.333	87.0	93.0	305-0.85-48	108.26	0.48	1.08	3	24Х9	200Д-60
-388.450	384.141	86.0	92.5	306-0.5-98	108.26	0.42	0.54	3	24Х9	200Д-60
-425.454	358.338	80.0	92.0	307-0.50-72	107.47	0.38	0.54	3	24Х9	14НДС
-84.555	180.440	80.0	94.1	308-1.38-90	118.44	1.03	1.40	3	24Х9	14НДС
-79.261	155.491	89.0	92.5	309-1.20-70	111.95	0.90	1.38	3	24Х9	14Д-6
-105.497	243.193	87.0	94.8	401-1.60-114	138.00	0.96	1.76	3	24Х9	300Д-90
-98.555	213.997	76.0	94.1	402-1.44-8	136.09	0.88	1.68	4	24Х9	300Д-90
-63.701	137.648	80.0	93.0	40-1.36-65	123.68	0.90	1.64	4	24Х9	300Д-90
-65.450	130.709	88.0	92.5	404-1.28-65	120.82	0.72	1.52	5	24Х9	200Д-60
-63.591	118.333	88.0	92.0	405-1.12-48	121.80	0.64	1.44	4	24Х9	200Д-60
-268.498	317.902	80.0	92.5	406-0.67-92	116.03	0.56	0.72	4	24Х9	14НДС
-238.512	268.277	80.0	92.0	407-0.67-72	117.15	0.51	0.72	4	24Х9	14НДС
-118.696	271.567	89.0	94.1	408-1.84-0	132.04	1.86	1.90	4	24Х9	14НДС
-44.584	116.618	87.0	92.5	409-1.60-70	123.41	1.20	1.84	4	24Х9	3В-200Х2
-475.999	474.379	77.0	94.3	501-0.58-108	154.65	0.37	0.80	4	24Х9	3В-200Х2

Продолжение таблицы П.8.2

1	2	3	4	5	6	7	8	9	10	11
-503.846	463.343	77.0	93.4	502-0.52-96	136.77	0.35	0.75	5	24К9	3В-200Х2
-473.834	417.424	77.0	93.0	503-0.45-85	136.26	0.34	0.72	5	24К9	3В-200Х2

a и b -коэффициенты аппроксимации характеристик типовых насосных станций параболой : $H = a Q^2 + b Q$, где Q - подаваемый расход воды;

H_{min} и H_{max} - минимальный и максимальный напор насосной станции;

Q_{min} и Q_{max} - минимальный и максимальный расход насосной станции;

K - количество агрегатов ;