**Математическая модель**

Рассмотрим транспортную сеть (задача о кратчайшем маршруте) (рис. 1).

Пусть w[i,j] – расстояние (или стоимость переезда) от пункта i в пункт j (на рисунке заданы числами у каждой стрелки). Необходимо выбрать такой путь от пункта «Чел» до остальных пунктов, для которого его длина (или общая стоимость переезда) является минимальной.

Нахождение в пункте называется состоянием системы, переезд из пункта в пункт – процессом перехода из одного состояния в другое. Таким образом, переезд из пункта «Чел» в «Екб», например, есть одношаговый процесс, а из пункта «Чел» в другие пункты, не связанные с Челябинском, – многошаговый процесс перехода из состояния «Чел» в другое.

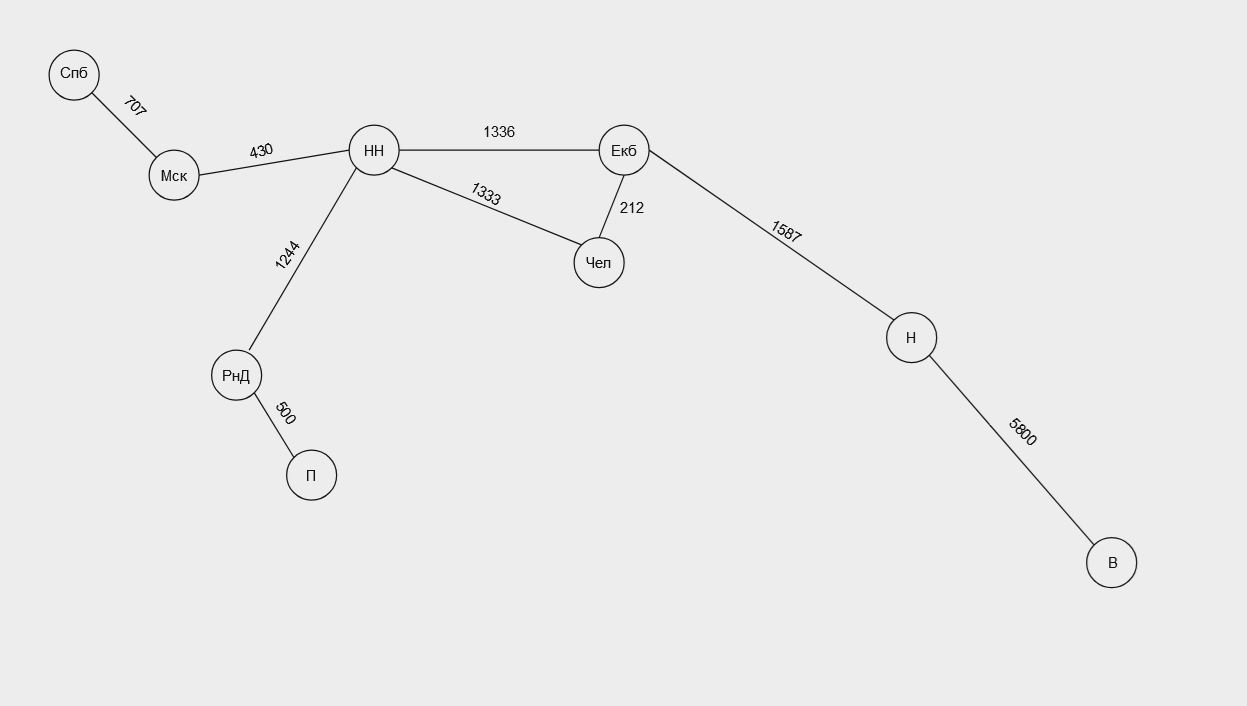
Выбор процесса перехода из состояния i в состояние j называется стратегией. Пусть найден оптимальный (в данном случае минимальный) маршрут и в настоящее время мы находимся в его промежуточном пункте, тогда, независимо от пути достижения этого пункта (состояния), оптимальный путь из данного пункта до конечного состояния есть часть общего оптимального пути.

Рисунок 1 – Схема графа транспортной сети

Используем принцип оптимальности, сформулированный Ричардом Беллманом: «каково бы ни было состояние системы перед очередным шагом, необходимо выбрать управление на данном шаге так, чтобы выигрыш на этом вместе с оптимальным доходом на всех последующих шагах был максимальным».

Для нахождения кратчайшего маршрута используем алгоритм Дейкстры.

выбрать начальную вершину – город Челябинск

создать последовательность из вершин, соединенных с начальной

while есть необработанные вершины в последовательности do

выбрать вершину последовательности с кратчайшим расстоянием до начальной

добавить эту вершину и ведущее в нее ребро в дерево пути

добавить в последовательность вершины, соединенные с добавленной

for для всех вершин последовательности do

добавить ребро, соединяющее ее с деревом и завершающее кратчайший путь к начальной вершине

end for

end while

**Обозначения**

* V V – множество вершин графа;
* E – множество рёбер графа;
* w[i,j] – вес (длина) ребра i, j;
* a – вершина, расстояния от которой ищутся;
* U – множество посещённых вершин;
* d[u] – по окончании работы алгоритма равно длине кратчайшего пути из a до вершины u;
* p[u] – по окончании работы алгоритма содержит кратчайший путь из a в u;
* v – текущая вершина, рассматриваемая алгоритмом.

**Псевдокод**

Присвоим d[a]←0, p[a]←0

Для всех u∈V отличных от a присвоим d[u]←∞

Пока ∃v∉ U

Пусть v∉ U – вершина с минимальным d[v] занесём v в U

Для всех u∉ U таких, что vu∈E

Если d[u]>d[v]+w[v,u] то

Изменим d[u]←d[v]+w[v,u]

Изменим p[u]←(p[v],u)p[u] ← (p[v], u)

Для поставленной задачи недостаточно применения динамического программирования в чистом виде, так как не учитываются следующие факторы:

1. Масса доставляемого груза. Возможна ситуация, что масса товара будет превышать грузоподъемность автомобиля и тогда возникнет необходимость отправить по маршруту несколько машин.
2. Учитывать, что каждая трасса имеет разную максимальную грузоподъемность (например, задать для каждой трассы логическую переменную, которая будет принимать значение 1, если транспорт может ехать по трассе или 0, если не может).
3. Расчёт нормы расхода топлива (необходимо учитывать возможную ситуацию, когда будет выгоднее легковому автомобилю вернуться в Челябинск, разгрузиться и отправиться в следующий город, чем отправлять более грузоподъёмный автомобиль).
4. Брак и возврат: у администратора должно быть поле для заявок на возврат + поле для жалоб на брак.

**Актуальность темы**

1. Эффективное управление логистикой.
2. Снижение затрат.
3. Рост конкурентной способности.