

Автоматизация оптимального планирования доставки материальных ресурсов до производственных центров предприятия

Р.А. Авхадиев

Аннотация. Одной из актуальных задач в автоматизации планирования производства является задача оптимизации перемещения ресурсов (деталей, комплектующих, материалов) между различными производственными объектами (источниками и производственными центрами - потребителями). Считаются заданными: множество потребителей ресурсов, график потребления (определяющий, какой ресурс к какому моменту и куда нужно доставить), множество источников ресурсов, график поставки, множество погрузчиков (объектов, обеспечивающих транспортировку ресурсов) и их характеристики (скорость, грузоподъемность, объем кузова). В работе представлен метод формирования оптимального графика движения погрузчиков при условиях: все ресурсы должны быть доставлены не позже регламентируемого срока и общая транспортная работа всех погрузчиков (пройденные тонно-километры) должна быть минимальной.

Ключевые слова: производственное планирование, логистика, график доставки.

Введение

Современное машиностроительное предприятие работает в условиях постоянно меняющегося спроса на выпускаемую продукцию, высоких требований покупателя к качеству и ассортименту продукции, а также коротких сроков поставки готовой продукции.

Все это делает крайне сложным управление производственным процессом, и обойтись без автоматизированной системы управления и планирования становится невозможным.

В настоящее время известно множество различных алгоритмов формирования плана доставки, начиная с классических, таких как метод ветвей и границ (https://ru.wikipedia.org/wiki/Метод_ветвей_и_границ), метод эластичной сети (<http://www.intuit.ru/>

[studies/courses/2255/139/lecture/20567?page=3](https://www.intuit.ru/studies/courses/2255/139/lecture/20567?page=3)), а также их многочисленные вариации, ориентированные на решение определенных практических задач. Однако, как правило, эти алгоритмы характеризуются следующими ограничениями:

- не учитываются конкретные сроки доставки ресурсов потребителям (определяются лишь большие временные интервалы);
- не учитывается возможность загрузки/выгрузки ресурсов в процессе выполнения задания. Обычно предполагается, что загрузка ресурсов происходит одновременно (на центральном складе), а затем развозится по потребителям. Это ограничение существенно снижает эффективность использования транспорта;
- время работы алгоритмов растет экспоненциально с ростом количества вершин графа. Это

затрудняет их использование в условиях большого объема данных реального производства.

В данной работе представлена возможность устранить вышеуказанные ограничения известных алгоритмов на основе гибкой коррекции плана с использованием метода кластеризации.

1. Обобщенная постановка задачи

Пусть имеются следующие входные данные.

- План производства на определенный период (производственный заказ). Он представляет собой перечень видов готовой продукции с указанием их количества и момента времени, к которому необходимо эту продукцию выпустить.

- Спецификация каждой единицы выпускаемой продукции. Она представляет собой перечень материалов и узлов (комплектующих), необходимых и достаточных для производства единицы готовой продукции. Узел, в свою очередь, также имеет определенную спецификацию. Раскрытие вложенных друг в друга спецификаций и приведение их к элементарным (не раскрываемым) материалам называется *“разузлованием”*.

- Технологические карты спецификаций. Они представляют собой перечень Технологических операций, которые нужно выполнить над составными частями спецификаций для *“превращения”* составных частей спецификации в результирующий продукт спецификации.

- Рабочие центры. Рабочий центр представляет собой совокупность физических объектов (станки, оборудование, рабочие, инструменты и т.д.), которая может выполнять определенный перечень технологических операций за определенное время. Рабочий центр также характеризуется затратами, которые необходимо понести для выполнения каждой технологической операции (электроэнергия, зарплата рабочих и т.д.).

- Склады ТМЦ (материалы, комплектующие, готовая продукция). При этом Рабочий центр также является складом, на котором хранятся ресурсы для производства, выпущенная продукция и НЗП (незавершенное производство).

Требуется: сформировать План выполнения производственного заказа. При этом производственный заказ должен выполняться в минимальный срок и/или с минимальной себестои-

мостью. Также требуется контролировать и учитывать его фактическое выполнение.

Указанный План состоит из следующих взаимосвязанных частей:

1. План обеспечения ресурсами. Он представляет собой перечень материалов и узлов (входящих в спецификации требуемых к выпуску единиц готовой продукции) с указанием способа их получения (закупка, производство, резерв склада) и момента времени, к которому каждый ресурс должен оказаться на Рабочем центре, которому он требуется.

2. План выполнения Технологических операций. Он представляет собой перечень Технологических операций с указанием Рабочего центра, на котором она будет выполнена, и момента времени выполнения.

3. План доставки ресурсов до рабочих центров (со складов и других рабочих центров)

2. Формирование плана доставки ресурсов

Рассмотрим подробно формирование Плана доставки ресурсов до рабочих центров

В математической модели считаются заданными:

1. множество Ресурсов и их характеристик (вес, объем, время, требуемое на погрузку/выгрузку);

2. множество источников Ресурсов (Склады и Рабочие центры);

3. множество потребителей Ресурсов (Рабочие центры);

4. множество Погрузчиков (объектов, обеспечивающих транспортировку ресурсов) и их характеристики (скорость, грузоподъемность, объем кузова); если перемещение ресурсов происходит посредством человека с тележкой - его тоже будем считать погрузчиком;

5. остатки Ресурсов в источниках на момент начала планирования;

6. график появления Ресурсов в источниках;

7. график потребности Ресурсов у потребителей;

8. граф проходимости местности, на которой расположены источники и потребители.

Требуется составить оптимальный график движения погрузчиков в виде таблицы с полями:

Табл. 1. График движения погрузчиков

Поле	Комментарий
№ задачи	Порядковый номер задачи (пункта графика)
Координаты	Координаты (X, Y) в которые нужно переместиться
Время	Момент времени, к которому необходимо достичь указанных координат
Ресурс	Код ресурса, который будет погружен или выгружен из Погрузчика в момент выполнения задачи
Количество Ресурса	Количество, на которое будет изменено содержание Ресурса в кузове Погрузчика (положительно - погрузка, отрицательное - выгрузка)

Так как все планирование происходит “сверху вниз” - от конечного результата (выпущенной продукции) к условиям, обеспечивающим этот результат, - будем отталкиваться от графика потребностей Ресурсов у потребителей.

Итак, у нас есть Рабочий центр, Ресурс, который ему требуется, и момент времени, в который ему требуется этот ресурс.

Следующий шаг - определить источник Ресурсов. Самый простой способ решения этой задачи - определить все Склады ТМЦ, на которых этот Ресурс имеется в наличии (на момент времени потребности минус запас времени на доставку) и выбрать из них один или несколько Складов с минимальной ценой перемещения ресурса между Складом и Рабочим центром.

На практике такой алгоритм на больших объемах данных будет работать неприемлемо долго даже на современных серверах. Это связано с тем, что для составления полного плана необходимо перебрать:

- Количество Рабочих центров
- × Количество требуемых видов Ресурсов
- × Количество Складов ТМЦ

Затем построить маршруты и выбрать из них оптимальные.

С первыми двумя элементами производства ничего сделать нельзя, но перебор по послед-

нему можно существенно оптимизировать. С учетом его значения (более 10 000 для крупного машиностроительного предприятия), скорость работы алгоритма сильно возрастет.

Оптимизация подбора Складов ТМЦ, имеющих требуемый Ресурс

Для оптимизации применим бинарные деревья и пространственное разбиение. Введем двумерную систему координат (X, Y) с началом координат в Складе ТМЦ с минимальными значениями географической широты и долготы.

Интересующая нас область пространства - это прямоугольник, ограниченный началом координат и Складом ТМЦ с максимальными значениями географической широты и долготы. Для крупного машиностроительного предприятия получится прямоугольник со сторонами в несколько километров.

Разобьем рассматриваемый прямоугольник на ячейки размером 10 метров на 10 метров.

Присвоим каждой ячейке целочисленные координаты $[X, Y]$. Ячейка в начале координат будет иметь координаты $[0, 0]$, остальные ячейки будут иметь определенные координаты

$$\left[\begin{array}{c} \text{дистанция до начальной ячейки по оси } X \\ \hline 10 \text{ метров} \\ \text{дистанция до начальной ячейки по оси } Y \\ \hline 10 \text{ метров} \end{array} \right]$$

«Привяжем» все Рабочие центры и Склады ТМЦ к ячейкам.

Введем понятие «Производственный объект» - это совокупность Рабочих центров и Складов ТМЦ, расположенных с достаточной плотностью (завод, корпус и т.д.).

Привяжем все ячейки к Производственным объектам; если ячейка расположена вне Производственного объекта, привяжем ее к пустому объекту.

Каждая ячейка также будет вершиной Графа проходимости местности (п. 6).

Далее, можно перебрать все ячейки в порядке удаления от рассматриваемого Рабочего центра (от ближних - к дальним) и найти Склады ТМЦ с нужным Ресурсом. Сначала искать в “своем” Производственном объекте, затем, если поиск был неуспешен, - расширить область поиска на другие Производственные объекты, также в порядке удаления от “своего” Производственного объекта.

Это существенно ускорит работу алгоритма, но недостаточно. Много времени будет затрачено на поиск в ячейках, вообще не имеющих Складов ТМЦ.

Построение бинарного дерева для поиска Складов ТМЦ

Для решения этой проблемы используем бинарные деревья. Каждому узлу дерева будет соответствовать некий прямоугольник и список ячеек, содержащих Рабочие центры / Склады ТМЦ и размещенных внутри этого прямоугольника.

Правила построения дерева.

1. На первой итерации исходный прямоугольник делится медианой на 2 равные части по оси X (по горизонтали). Каждой части сопоставляется узел дерева (левой - А и правой - В). Каждому узлу сопоставляется список ячеек, содержащих Рабочие центры / Склады ТМЦ. Ячейки с координатой X < координаты X медианы попадают в левый узел, ячейки с координатой X ≥ координаты X медианы попадают в правый узел. Если оба узла получили 1 или менее ячеек, построение дерева завершается.

2. Прямоугольники каждого узла, имеющие более 1 ячейки, делится медианой на 2 равные части (по вертикали), каждой из получившихся частей также сопоставляется узел (С, D,

Е, F) и список ячеек, содержащих Рабочие центры / Склады ТМЦ. Ячейки с координатой Y < координаты Y медианы переносятся из родительского узла в левый узел-потомок, ячейки с координатой Y ≥ координаты Y медианы переносятся из родительского узла в правый узел-потомок.

3. Итерация 2 повторяется (с чередованием оси разбиения (горизонталь/вертикаль)) до тех пор, пока не останется узлов, содержащих более 1 ячейки. [1]

Пример разбиения показан на Рис.1.

В результате получится бинарное дерево, показанное на Рис.2

Полученное дерево будет обладать следующими свойствами:

- каждому уровню дерева будет соответствовать ось координат (X или Y), использованная при построении дерева для разбиения пространства;
- каждый Рабочий центр / Склад ТМЦ, сопоставленный со всеми потомками левого поддерева, будет иметь координату соответствующей оси < координаты медианы этой же оси;
- каждый Рабочий центр / Склад ТМЦ, сопоставленный со всеми потомками правого поддерева, будет иметь координату соответствующей оси ≥ координаты медианы этой же оси.

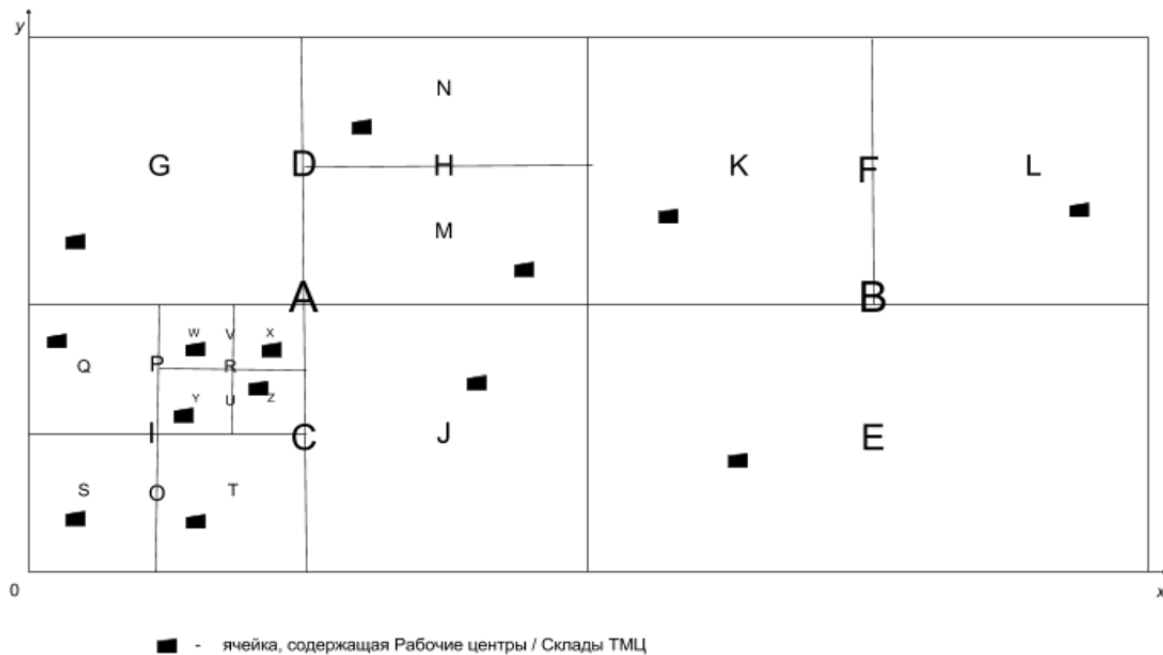


Рис. 1. Пространственное разбиение производственной площадки

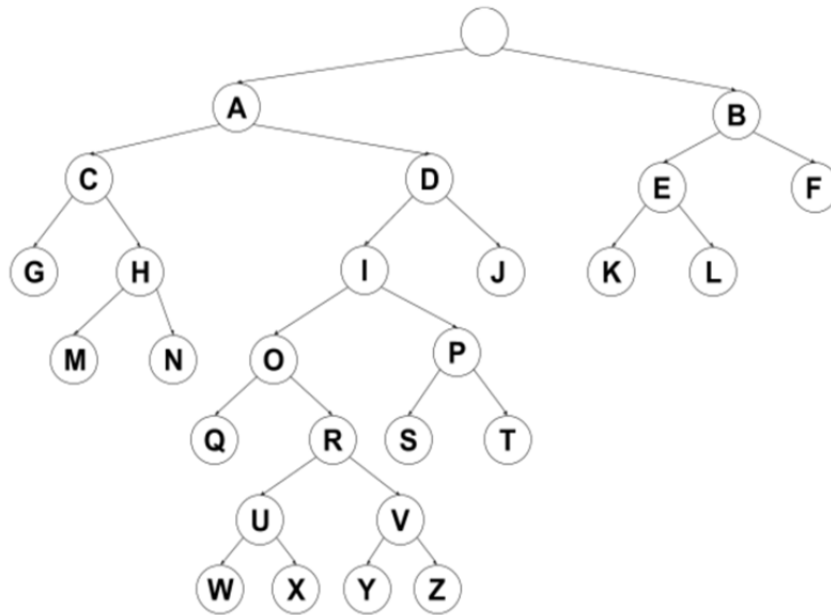


Рис. 2. Бинарное дерево поиска

Алгоритм поиска по построенному бинарному дереву

Благодаря свойствам построенного бинарного дерева, возможна реализация следующего высокоэффективного алгоритма поиска Складов ТМЦ, находящихся в определенном радиусе от Рабочего центра.

1. Определим исходную координату поиска $[X_0, Y_0]$, равную координате ячейки, содержащей Рабочий центр.
2. Определим радиус поиска R (целое число). Принцип его подбора будет описан ниже.
3. Определим область поиска. Это будет квадрат, заданный координатами

$$[X_1, X_2, Y_1, Y_2], \text{ где}$$

$$X_1 = X_0 - R$$

$$X_2 = X_0 + R$$

$$Y_1 = Y_0 - R$$

$$Y_2 = Y_0 + R$$

Таким образом, строго говоря, поиск будет осуществляться не внутри окружности радиусом R с центром в $[X_0, Y_0]$, а в квадрате, в который вписана эта окружность. Для целей алгоритма это несущественно.

4. Начнем обход дерева, начиная с корня. Определим взаимное расположение медианы корня (разделяющей исходный прямоугольник

всей рассматриваемой области пространства пополам по оси X) и области поиска. Координату медианы по оси X обозначим как X_m .

В расположении медианы и области поиска возможны 3 варианта:

- 4.1. Область поиска находится слева от медианы: $(X_2 < X_m)$. В этом случае исключим правого потомка из последующего поиска и будем производить поиск только в поддереве левого потомка. Добавим левого потомка в список узлов для поиска.
- 4.2. Область поиска находится справа от медианы: $(X_1 \geq X_m)$. В этом случае исключим левого потомка из последующего поиска и будем производить поиск только в поддереве правого потомка. Добавим правого потомка в список узлов для поиска.
- 4.3. Медиана находится внутри области поиска: $(X_2 \geq X_m)$ И $(X_1 < X_m)$. В этом случае поиск будем производить по обоим потомкам. Добавим левого и правого потомков в список узлов для поиска. [2]
5. Рассмотрим левого потомка корневого узла (если он не отброшен на этапе 4). Аналогично этапу 4 определим взаимное расположе-

ние его медианы и области поиска. Так как этот уровень дерева получен разбиением пространства по оси Y , взаимное расположение медианы и области поиска будем определять по оси Y . Здесь возможны те же 3 варианта, что и на этапе 4, Работа этой части этапа алгоритма аналогична этапу 4.

Удалим левый узел из списка узлов для поиска.

Далее, если у данного узла (левого потомка корневого узла) нет потомков, добавим все соответствующие ему Склады ТМЦ в список результата поиска.

6. Рассмотрим правого потомка корневого узла (если он не отброшен на этапе 4). Поступим с ним аналогично этапу 5

7. Если список узлов для поиска пуст - завершаем работу алгоритма. Если нет - выберем первый элемент.

8. Произведем с выбранным узлом те же действия, что и на этапе 5, учитывая ось разбиения выбранного узла (X или Y)

9. Перейдем на этап 7.

Среднее время работы данного алгоритма нотации $\text{big } O$ (https://en.wikipedia.org/wiki/Big_O_notation) составит:

$O(\log_2 n)$, где n - количество Складов ТМЦ [3]

По сравнению с методом перебора, имеющего сложность $O(n)$, результат значительно лучше.

В результате работы данного алгоритма получим список ближайших Складов ТМЦ, с которых будем пытаться получить требуемый Ресурс. Если суммарный запас ресурса на всех Складах ТМЦ из списка меньше, чем требуется Рабочему центру - увеличим радиус поиска R в 2 раза и повторим выполнение алгоритма. Если R стал больше максимального значения (просмотрена вся доступная площадь), а достаточное количество ресурса так и не найдено - записать ошибку в лог, затем попробовать повторить поиск позже (в надежде, что требуемое количество ресурса появится).

Определение начального радиуса поиска R

От значения начального радиуса поиска R сильно зависит эффективность работы алгоритма. Если сделать его слишком малым, то начальный поиск будет неудачным и процессорное время будет потрачено зря. Если, напротив, сделать его слишком большим - ал-

горитм подбора узлов будет работать долго и в выборку попадет слишком много Складов ТМЦ, на проверку которых опять же уйдет процессорное время. Вместе с тем, вычисление оптимального начального R - задача, требующая столько же ресурсов процессора, что и сам поиск. Поэтому, опытным путем был выбран следующий механизм:

Изначально $R = 1$. После выполнения успешного поиска, значение R запоминается (в разрезе Рабочего центра и Ресурса), и при следующем поиске в качестве начального R используется сохраненное значение. При этом один раз в определенный период (один раз в день) при очередном поиске начальное значение R делится на 2. Это сделано для того, чтобы начальное значение R имело небольшую тенденцию к уменьшению своего значения, т.к. в противном случае, при разовом поиске в неблагоприятных условиях (например, если требуемый Ресурс имеется только на одном удаленном Складе ТМЦ), R примет большое начальное значение, и последующий поиск всегда будет вестись по избыточной площади.

Алгоритм подбора Погрузчика для перемещения Ресурса

Итак, мы получили список Складов ТМЦ, имеющих Ресурс, который требуется Рабочему центру. Следующий этап составления Плана доставки ресурсов - выбор Погрузчиков и внесение задачи по перемещению Ресурса в его график. При этом, должно выполняться 2 условия.

1. Ресурс должен быть доставлен вовремя.

2. Доставка должна добавлять как можно меньше Транспортной работы Погрузчикам.

Транспортная работа - это произведение массы и расстояния, на которое масса была перемещена. Как правило, измеряется в тонно-километрах. Вопреки классическим правилам логистики, в массу перевозимого груза включим массу самого погрузчика. Это необходимо для того, чтобы сделать нежелательной ситуацию перевозки небольшого груза на большое расстояние пустым Погрузчиком.

Условие 1 - это просто ограничитель, по которому будем отсекаать неподходящие варианты перемещения. В алгоритме минимизировать будем транспортную работу из условия 2.

Идеальный вариант, к которому будем стремиться - найти Погрузчик, который итак перемещается от точки, близкой к Складу ТМЦ к точке, близкой к Рабочему центру. В этом случае, Погрузчик должен просто “захватить с собой” дополнительно еще и Ресурс, который требуется переместить.

На первом этапе работы алгоритма сформируем список Погрузчиков, находящихся в определенном радиусе R от Рабочего центра. R определим как:

(Расстояние от Рабочего центра до самого удаленного от него Склада ТМЦ) \times 1.1

Коэффициент “1.1” используется для того, чтобы включить в список Погрузчики, которые находятся относительно Рабочего центра “за” Складом ТМЦ, но при этом они недалеко от Склада ТМЦ и, возможно, направляются в сторону Рабочего центра.

Для формирования списка ближайших Погрузчиков применим пространственное разбиение и бинарное дерево, аналогично Складам ТМЦ.

При этом необходимо учесть следующую специфику:

Погрузчики, в отличие от Складов ТМЦ, постоянно перемещаются. Соответственно, нужно предусмотреть постоянное обновление бинарного дерева без его полного перестроения. Для этого используем следующий алгоритм:

1. При перемещении Погрузчика запомним его старые координаты (X_0, Y_0) и новые координаты (X_1, Y_1) . Если они все остались внутри одной ячейки - обновление дерева не требуется, завершаем работу алгоритма.

2. Если ячейка местонахождения сменилась - удалим ссылку на Погрузчик из старой ячейки.

3. Если в ячейке не осталось Погрузчиков - удалим соответствующий этой ячейке узел дерева. Так как ссылки на ячейки могут быть только у узлов типа “лист” (не имеет своих потомков), перестроения потомков не потребуется.

Родительский узел удаляемого узла заменим на единственного оставшегося потомка этого родительского узла

4. Добавим ссылку на Погрузчик в ячейку, соответствующую новым координатам Погрузчика. Если узла с такой ячейкой в дереве еще

нет - добавим его. Для этого начнем сканирование дерева, в поисках подходящего “листа”, начиная от корня.

4.1. Если $X_1 < X_m$, перейдем к левому потомку (назначаем его “текущим узлом”), если $X_1 \geq X_m$, перейдем к правому потомку (назначаем его “текущим узлом”). X_m - это координата медианы корневого узла

4.2. Если текущий узел является “листом”, то сканирование завершаем, искомым “лист” найден, если нет - переходим к следующему шагу

4.3. Если $Y_1 < Y_m$, перейдем к левому потомку текущего узла (назначаем его новым “текущим узлом”), если $Y_1 \geq Y_m$, перейдем к правому потомку текущего узла (назначаем его новым “текущим узлом”). Y_m - это координата медианы текущего узла

4.4. Таким образом, спускаемся вниз по дереву (учитывая ось разбиения текущего уровня дерева) до тех пор, пока не будет найден узел типа “лист”

4.5. Разобьем найденный узел медианой на 2 равные части (используя ось разбиения, соответствующую уровню дерева, на котором находится этот узел)

4.6. Поместим ячейку разбитого узла и только что добавленную (новую) ячейку в левый, либо в правый узел разбитого узла согласно правилу построения дерева: если координата по оси разбиения меньше медианы - в левый узел, если больше, либо равна - в правый узел

4.7. Если ячейки попали в разные узлы - завершаем алгоритм. Если ячейки попали в один и тот же узел - делим этот узел аналогично шагу 4.5. [4]

Итак, мы вплотную подошли к формированию графика движения Погрузчиков.

Имеем список Погрузчиков, находящихся в определенном радиусе R от целевого Рабочего центра и, предположительно, подходящих для перемещения требуемого Ресурса в Рабочий центр. Требуется выбрать из списка тот Погрузчик, который переместит Ресурс со Склада ТМЦ в Рабочий центр с минимальным увеличением своей Транспортной работы.

Рассмотрим сначала самый простой вариант: в списке всего 1 Погрузчик, его текущий график движения пустой, и грузоподъемности Погрузчика достаточно для перемещения всего требуемого количества Ресурса за 1 раз. В этом случае необходимо добавить в его график следующие задачи:

1. перемещение от текущих координат к Складу ТМЦ (опустим пока задачу нахождения кратчайшего пути по графу проходимости местности);

2. погрузка Ресурса;

3. перемещение к Рабочему центру;

4. выгрузка Ресурса.

Для учета времени, затраченного на погрузку/выгрузку/ожидание очереди, введем специальную координату Входа на Склад ТМЦ и Рабочий центр. Время на перемещение из координаты Входа в координату Склада ТМЦ/Рабочего центра будет включать время ожидания очереди (если Склад ТМЦ/Рабочий центр в данный момент занят другим Погрузчиком). Время на обратное перемещение (из координаты Склада ТМЦ/Рабочего центра) будет включать время, затраченное на погрузку/выгрузку.

Например:

Координаты Рабочего центра: (14, 8), вход (14, 9)

Координаты Склада ТМЦ: (25, 16), вход (25, 17)

Координаты Погрузчика: (1, 1), скорость 1 метр в секунду

Требуется перевести Ресурс с кодом 00019184 в объеме 5 шт.

Время на погрузку/выгрузку 1 шт.: 12 секунд

Препятствия на пути отсутствуют, очереди отсутствуют

Текущее время планирования: 0:00:00

Размер 1 ячейки: 10 метров на 10 метров

В этом случае, график Погрузчика будет сформирован следующим образом (Табл.2).

Теперь рассмотрим алгоритм выбора подходящего Погрузчика для общего случая.

0. Выбрать первый Погрузчик из списка подходящих Погрузчиков, назначить его текущим.

1. Если график текущего Погрузчика не пустой - найти координату в графике, с минимальным расстоянием до Склада ТМЦ.

Табл. 2. Пример построенного графика движения погрузчика

№	Координаты	Время	Ресурс	Кол-во
1	(1, 1)	0:00:00		
2	(25, 17)	0:04:48		
3	(25, 16)	0:04:58		
4	(25, 17)	0:06:08	00019184	5
5	(14, 9)	0:08:24		
6	(14, 8)	0:08:34		
7	(14, 9)	0:09:44	00019184	-5

1.1. Если график текущего Погрузчика пустой - использовать текущую координату Погрузчика

2. Вставить в график (после выбранной координаты) задачу забора Ресурса со Склада ТМЦ. Количество забираемого ресурса определяется по формуле:

MIN (Остаток Ресурса на Складе ТМЦ, Остаток места в Погрузчике в момент подъезда к Складу ТМЦ, Количество Ресурса требующееся на Рабочем центре)

3. Найти в графике (после только что вставленной задачи координату), максимально близкую к Рабочему центру. Если только что вставленная задача последняя - использовать координату этой задачи.

4. Вставить в график (после выбранной координаты) задачу выгрузки Ресурса в Рабочем центре.

5. Проверить, не оказался ли превышен лимит места в Погрузчике на задачах погрузки, расположенных после задачи, вставленной на Этапе 2. Если превышен - уменьшить количество погружаемого на Этапе 2 ресурса на объем превышения. Если количество стало равно 0 - отбросить текущий погрузчик, перейти к Этапу 8

6. Пересчитать время во всех задачах, расположенных после вставленной задачи на Этапе 2. Если хотя бы в одной задаче выгрузки в

Рабочем центре Ресурс будет доставлен позже необходимого срока - отбросить текущий Погрузчик, перейти к Этапу 8.

7. Пересчитать объем Транспортной работы, выполняемой по всему графику после задачи, вставленной на Этапе 2. Выделить объем добавленной Транспортной работы как разность между ее новым и старым значениями.

8. Выбрать следующий Погрузчик из списка, перейти к Этапу 1. Если его нет - перейти к Этапу 9.

9. Выбрать Погрузчик с минимальным объемом добавленной Транспортной работы, зафиксировать изменения в его графике. Если он может перевезти весь необходимый объем Ресурса - завершить алгоритм. Если не может - упорядочить Погрузчики по возрастанию добавленной Транспортной работы, подобрать по порядку Погрузчики для перевозки оставшегося объема требуемого Ресурса, зафиксировать изменения в их графиках. Если перевозка необходимого объема Ресурса обеспечена - завершить алгоритм, если нет - перейти к Этапу 10.

10. Увеличить радиус поиска Погрузчиков (R) в 2 раза, сформировать новый список Погрузчиков, исключив из него только что обработанные, повторить выполнение алгоритма с Этапа 0. Если радиус R оказался больше максимально допустимого (обработаны все имеющиеся в распоряжении Погрузчики) - записать несоответствие в лог, завершить алгоритм..

Учет очередей на погрузку/выгрузку

После формирования всего Плана доставки Ресурсов до Рабочих центров необходимо учесть очереди из Погрузчиков, которые могут возникнуть при погрузке Ресурсов в Складах ТМЦ или выгрузке в Рабочих центрах. В идеале, их нужно учитывать на этапе формирования Плана, но это вызовет постоянный рекурсивный пересчет графиков большого числа Погрузчиков и огромные затраты процессорного времени.

Выделим возникающие очереди (пересекающиеся временные промежутки в графиках разных Погрузчиков у одних и тех же Складов ТМЦ и Рабочих центров). Пересчитаем время “входа” в Склад ТМЦ / Рабочий центр и время всех следующих задач.

Если график того или иного Погрузчика перестал удовлетворять условию “доставка вовремя” -

пересчитаем весь его график, исключив текущий Склад ТМЦ (с очередью) из списка допустимых источников ресурсов. После этого повторим выделение и обработку очередей.

Заключение

Представлен метод формирования валидного (при отсутствии в логе несоответствий) Плана доставки Ресурсов машиностроительного предприятия до Рабочих центров.

Разработанный метод позволяет получать решения, по качеству лишь немного уступающие алгоритму полного перебора всех вариантов.

Так, в результате тестирования на данных в объеме крупного машиностроительного предприятия в объеме:

- Рабочих центров - 10 000 шт.
- Складов ТМЦ - 7 000 шт.
- Погрузчиков - 1 000 шт.
- Видов ТМЦ - 50 000 шт.
- Период планирования - 30 минут (на

большее не хватило вычислительных мощностей для алгоритма полного перебора).

общее количество затраченных тонно-километров рассмотренного алгоритма оказалось больше всего на 1.27% чем у алгоритма полного перебора.

В то же время, разработанный метод обладает следующими преимуществами, по сравнению с известными алгоритмами:

1. Время расчета достаточно мало для формирования/пересчета в реальном времени.

2. Даже если по каким-то причинам формирование законченного Плана невозможно (нестыковки в данных, недостаточное обеспечение технологического процесса, принципиальная невозможность произвести требуемый объем продукции в заданные сроки и т.д.), рабочая версия Плана все равно будет сформирована, и может быть использована на практике с результатом, близким к оптимальному.

3. При практическом выполнении Плана неизбежны определенные отклонения. «Отклонения от оптимальности» сформированного с помощью предлагаемого метода Плана (лишние затраты времени) будут значительно меньше возникших отклонений факта и “растворятся” в них.

Литература

1. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2009) [1990]. Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4.
2. Mehlhorn, Kurt; Sanders, Peter (2008). Algorithms and Data Structures: The Basic Toolbox
3. Heger, Dominique A. (2004), "A Disquisition on The Performance Behavior of Binary Search Tree Data Structures" European Journal for the Informatics Professional, 5 (5): 67–75
4. Gonnet, Gaston. "Optimal Binary Search Trees". Scientific Computation. ETH Zürich. December 2013.

Авхадиев Рустем Ахмедович. Аспирант Казанского Национального Исследовательского Технического Университета им. А.Н. Туполева. Окончил КНИТУ-КАИ в 2004 году. Количество печатных работ: 6. Область научных интересов: информационные технологии. E-mail: psilon@inbox.ru

Scheduling the resource delivery to production work centers of the plant

R.A. Avkhadiev

Abstract. During the automation of production scheduling, it is essential to optimize resources reallocation between different production sites (sources and consumers). Available data: set of resource consumers, consumption schedule (defining when and where each resource should be delivered), set of resource sources, delivery schedule, set of loaders (objects ensuring transportation of resources) with their respective parameters (speed, carrying capacity, bodyspace). The objective is to generate the optimal motion schedule for loaders. When generating the schedule, two conditions should be met: all resources should be delivered not later than when they are required, and total transportation effort of all loaders (tonne-kilometres traveled) should be minimized.

Keywords: production scheduling, logistics, delivery schedule.

References

1. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2009) [1990]. Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4.
2. Mehlhorn, Kurt; Sanders, Peter (2008). Algorithms and Data Structures: The Basic Toolbox
3. Heger, Dominique A. (2004), "A Disquisition on The Performance Behavior of Binary Search Tree Data Structures" European Journal for the Informatics Professional, 5 (5): 67–75
4. Gonnet, Gaston. "Optimal Binary Search Trees". Scientific Computation. ETH Zürich. December 2013.

R.A. Avkhadiev postgraduate student of Kazan National Research Technological University (KNRTU-KAI), 68 Karl Marx street, Kazan, Republic of Tatarstan, Russian Federation. Specialist, in 2004 completed KNRTU-KAI. Count of published works: 6, Science interests: information technologies. Email: psilon@inbox.ru