УДК 519.688

Реализация привязки поездов на нитки вариантного графика с помощью решения задачи о назначениях

Н.Г. Рябых[1], Е.М. Захарова[1], И.К. Минашина[1], Ф.Ф.Пащенко[2]

[1]Московский физико-технический институт (государственный университет)

[2] Институт Проблем Управления им. В.А. Трапезникова Ран, ГУ

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №13-01-13105 офи_м_РЖД «Качественный анализ и моделирование алгоритмов оптимизации мультиагентной системы управления перевозочным процессом на железнодорожном транспорте».

Вступление

Задача планирования расписания грузоперевозок является одной из наиболее сложных и актуальных задач, возникающих в рамках управления движением на железнодорожном транспорте. В задачу планирования расписания грузоперевозок можно включить задачу составления расписания движения поездов (а именно, выбор оптимального времени отправления поездов со станций), а также задачу привязки к поездам локомотивов и локомотивных бригад. При этом, хотя и эти задачи требуется решать в комплексе, задача составления расписания движения поездов является базовой, на которую впоследствии накладываются ограничения задач о прикреплении тяговых ресурсов.

В статье будет рассказано об одной из задач проекта, запланированного в рамках гранта РФФИ по конкурсу ориентированных фундаментальных исследований по актуальным междисциплинарным темам в интересах ОАО «РЖД». В данной работе будет описан способ составления расписания движения грузовых поездов, а именно привязка поездов на нитки вариантного графика движения с помощью сведения данной задачи к задаче о назначениях и решения ее методом аукционов.

Постановка задачи

Главное ограничение в задаче составления расписания движения поездов заключается в том, что для поездов нельзя выбирать произвольные времена отправления и прибытия на станции. Поезда, в соответствии с технологией, должны двигаться только по определенным ниткам вариантного графика, который составляется на основе нормативного графика движения поездов на каждые следующие железнодорожные сутки. Поэтому задачу составления расписания можно поставить как задачу поиска оптимальных ниток вариантного графика для поездов.

Формально задачу можно сформулировать следующим образом. На вход модуля планирования подаются следующие данные:

- Исходное расположение поездов на полигоне на начало планирования;
- Сформированные и готовые к отправлению поезда, находящиеся на станциях;
- План составообразования поездов на ближайшие сутки с указанием станции формирования и приблизительного времени завершения формирования каждого поезда;
- Вариантный график движения: набор ниток, на которые можно прикреплять поезда.

Для каждого поезда известно его текущее местоположение и маршрут до станции назначения. Задача заключается в том, чтобы оптимальным образом выбрать нитки вариантного графика, на которые требуется прикрепить поезд. При этом поезд необязательно должен занимать всю нитку целиком. Возможны случаи, когда по одной нитке движутся два или более поездов (на разных участках), а также случаи, когда один поезд следует по нескольким ниткам последовательно на разных участках.

Сведение к задаче о назначениях

Задача о назначениях – это стандартная математическая задача, она является типичным примером задачи комбинаторной оптимизации. В общей форме эту задачу можно сформулировать следующим образом:

- Имеется некоторое число работ (N) и некоторое число исполнителей (M) этих работ. Каждый исполнитель умеет выполнять некоторое количество P из этих работ ($0 \leq P \leq N$), причем эффективность выполнения «своих» работ у исполнителя разная: некоторые работы он умеет выполнять лучше, некоторые – хуже. Требуется распределить работы между исполнителями так, чтобы суммарная эффективность назначения была максимальной.

Для каждой пары <работа, исполнитель> вводится «функция полезности» U_{ij} - числовое выражение эффективности назначения данного исполнителя на данную работу. Таким образом, оптимизация в рамках данной задачи сводится к максимизации функции $\sum U_{ij}$ по всем назначениям исполнителей на работы при условии, что каждый исполнитель назначен не более чем на одну работу.

Рассмотрим, как свести задачу привязки поездов на нитки графика к задаче о назначениях:

- В качестве «исполнителей» будем считать поезда, которые требуется прикрепить к ниткам. Тогда главным требованием задачи о назначениях будет обеспечение работами всех исполнителей (что будет эквивалентно назначению всех поездов на нитки графика).
- В качестве «работ» возьмем нитки вариантного графика. Тогда условие, что «каждый исполнитель может выполнять некоторое количество работ» соответствует тому, что для каждого поезда имеется выбор, к какой нитке оказаться прикрепленным. При этом для некоторых пар <поезд, нитка> прикрепление вообще невозможно: это, например, происходит для случаев, когда маршрут поезда не имеет совпадений с маршрутом нитки.

Поскольку каждый поезд может быть назначен на разные нитки на разных фрагментах маршрута, то для решения общей задачи о прикреплении поездов на нитки будем решать последовательность задач о назначениях:

- Каждая задача о назначениях будет ставиться для отправления поездов с одной станции.
- Для каждой станции определим набор поездов, которые на данной итерации требуется запланировать к отправлению с данной станции. На первом шаге это поезда, которые находятся на данной станции на начало планирования, а также те, которые уже были запланированы к отправлению и текущая станция является следующей необработанной станцией их маршрута.
- После того как мы выполнили привязку поездов на одной станции (решили задачу о назначениях), мы знаем время, в которое каждый поезд прибудет на следующую станцию своего маршрута. На следующей итерации каждый привязанный поезд попадет в множество поездов «которые уже были запланированы к отправлению» (см. предыдущий пункт) в задаче о назначениях на следующей станции их маршрута.
- Таким образом, перед каждой итерацией мы можем вычислить количество поездов, которые требуется привязать на нитки вариантного графика на каждой станции планирования. Тогда в качестве обрабатываемой на данном шаге станции будем выбирать станцию, на которой это количество поездов максимально. Последовательно проходя все станции, мы выполним привязку всех поездов на всем протяжении их маршрутов.

Вычисление функции полезности

Ключевым вопросом для формулировки задачи привязки поездов на нитки как задачи о назначениях является правило вычисления функции полезности для каждой пары <поезд, нитка>. Вычисление функции полезности будем вести следующим образом. Сначала определим критерии, которые определяют эффективность назначения поезда на нитку. Таким критериями являются:

1. Время ожидания поезда. Поскольку долгий простой поездов на станциях нецелесообразен, время стоянки поезда на станции в ожидании следующей нитки должно быть, по возможности, маленьким. Для этого критерия есть несколько нюансов использования:
    a. На станции формирования поезда этот критерий требуется учитывать с меньшим весом, исходя из того, что на станции формирования поезд может простаивать в ожидании отправления дольше, чем на промежуточных станциях по ходу движения.
    b. Для станций смены бригад и локомотивов должно учитываться время на стоянку поезда для соответствующей смены.
2. Равномерность почасового отправления поездов:
    a. Количество поездов, запланированных к отправлению в данный час, должно быть близко к среднему количеству поездов, которое надо отправить в час с этой станции. Среднее количество можно найти, взяв объемный суточный план отправления поездов и разделив количество поездов из этого плана на 24.
    b. Требуется оставлять «свободные пропускные способности» на случай внепланового отправления поездов. Например, если в какой-то час со станции проложено 5 свободных ниток вариантного графика, а в другой час – только 2 нитки, то скорее надо назначать поезда на отправление в первый час, а не во второй.
3. Процент совпадения маршрута поезда и нитки. Поезда надо по возможности привязывать к тем ниткам, маршрут которых полностью совпадает с маршрутом поезда. Тогда при назначении поезда не останется лишних «хвостов» ни от маршрута поезда, ни от нитки.
4. Сохранение предыдущей подвязки поезда. Если поезд уже прикреплен на нитку на предыдущем участке, то надо стремиться сохранить текущую нитку.

Каждому критерию ставится в соответствие нормированное численное значение u_k, а затем вычисляется значение функции полезности для пары <поезд, нитка> как $U_{ij} = \sum_k c_k u_k$, где c_k – вес соответствующего k-го критерия. Веса подбираются экспериментальным образом в ходе настройки системы и могут быть указаны отдельно для разных участков полигона.

Заметим, что функции полезности имеет смысл вычислять только для тех ниток, для которых назначение поезда хотя бы теоретически возможно. Для остальных ниток значение функции полезности задаваться не должно. В множество невозможных ниток могут, например, попадать нитки, не имеющие пересечений с маршрутом поезда, или нитки с категорией, не подходящей под категорию поезда. Набор условий для такого отсева ниток может задаваться отдельно для каждого фрагмента полигона.

Алгоритм решения задачи о назначениях с помощью аукционов

Задача о назначениях является частным случаем транспортной задачи, а она, в свою очередь, является частным случаем задачи линейного программирования. Эти задачи можно решать стандартным симплекс-методом, однако лучше использовать специализированные алгоритмы.

Самым распространенным алгоритмом решения задачи о назначениях является «венгерский алгоритм». Этот алгоритм был предложен в 1955 году и имеет вычислительную сложность $O(n^4)$, где n – количество работ и исполнителей (при условии, что эти количества одинаковые). Этот алгоритм можно модифицировать так, чтобы достичь времени выполнения $O(n^3)$. Данный алгоритм был реализован для решения исследуемой задачи о назначении поездов на нитки графика, но его производительность оказалась неприемлемой. Поэтому был найден другой алгоритм – решение задачи о назначениях с помощью аукционов.

Впервые метод решения задачи о назначениях с помощью аукционов был предложен Д. Бертсекасом в статье 1989 года [2] и был доработан им же и Д. Кастаноном в 1992 году [3]. Алгоритм изначально предназначен для ассиметричной задачи о назначениях, когда количество работ и количество исполнителей не совпадают. Основные идеи алгоритма заключаются в следующем:

1. Для каждой работы (в данном случае – для нитки) вводится понятие «цены» p_j. Это значение показывает дополнительную «стоимость», которую исполнитель (поезд) должен «заплатить», чтобы оказаться назначенным на данную работу. В начале алгоритма все нитки имеют нулевую цену.
2. На первом шаге берется произвольный поезд и назначается на нитку с максимальной функцией полезности. На этом же шаге происходит изменение цены данной «наилучшей» нитки следующим образом. Выбирается «вторая по полезности» нитка – это нитка, значение функции полезности (по данному поезду) для которой максимальное среди всех остальных, неприкрепленных ниток. В качестве цены «наилучшей» нитки устанавливается разность между «максимальным» и «вторым» значением функции полезности.
3. Смысл такой установки цены следующий. Допустим, для какого-то другого поезда максимальной функцией полезности также оказалась функция, вычисленная для этой, уже занятой нитки. Тогда, чтобы назначить новый поезд на эту нитку, предыдущий поезд требуется

переназначить на другую нитку – вторую по полезности. Но тогда суммарная полезность всего назначения понизится как раз на разность между максимальным и вторым значением функции полезности для первого поезда. Таким образом, чтобы суммарная функция полезности не стала меньше, от второго поезда требуется, чтобы это максимальное значение функции полезности для второго поезда превышало «второе» значение функции полезности второго поезда не менее чем на цену уже занятой нитки.
4. Таким образом, можно сказать, что поезда начинают «торговаться» за нитки, предлагая фиксированные суммы, но цены на нитки постоянно изменяются.

Итого, шаги алгоритма можно описать следующим образом:

1. Для неприкрепленного поезда i выбирается наилучшая нитка j: $j_i = \arg\max_{j \in A(i)}\{U_{ij} - p_j\}$, где p_j – цена нитки j, $A(i)$ – множество ниток, допустимых для поезда i.
2. Вычисляется второе по величине значение «выгоды» от назначения поезда на нитку j_i : $w_i = \max_{j \in A(i), j \neq j_i}\{U_{ij} - p_j\}$. Если других возможных назначений поезда, кроме нитки j_i, нет, то w_i принимается равным $-\infty$.
3. Далее вычисляется новая цена для нитки j: $p_j = \max\ \{\lambda; U_{ij_i} - w_i + \varepsilon\}$, где λ – пороговое значение (константа), ниже которого цену устанавливать запрещается, ε – малая величина порядка *1/N*.
4. Если $\lambda \leq u_{ij_i} - w_i + \varepsilon$, то поезд i назначается на нитку j, а предыдущее назначение на нитку j сбрасывается (поезд *i'*, который ранее был назначен на нитку j, удаляется из множества прикрепленных поездов).
5. Шаги 1-4 повторяются до тех пор, пока все поезда не окажутся назначенными на нитки. В том случае, если количество ниток меньше количества поездов, алгоритм на каком-то шаге достигнет оптимального распределения поездов и набор назначенных поездов не будет далее изменяться. В этом случае также требуется остановить работу алгоритма.

Строгое доказательство сходимости данного алгоритма, а также обоснование выбора значений величин λ и ε приведено в статье [3].

Результаты и пути улучшения работы алгоритма

Для сравнения производительности решения задачи о назначениях с помощью венгерского алгоритма и алгоритма аукционов эти алгоритмы были реализованы на языке программирования AgentSpeak. Тестирование на модельных примерах показало, что уже на задачах размерности около 200x200 алгоритм аукционов показывает существенный прирост в скорости сходимости по сравнению с венгерским алгоритмом: 20 секунд вместо 3-4 минут (данные временные показатели не стоит оценить в абсолютном значении, поскольку AgentSpeak сам по себе довольно «медленный» язык). Оценка показывает, что алгоритм аукционов сходится за время около $Nlog(N)$ и существенно выигрывает в производительности у венгерского алгоритма.

Далее алгоритм аукционов был реализован на языке Java в модуле планирования системы ИСУЖТ на Восточном полигоне, который осуществляет привязку поездов на нитки графика на Восточном полигоне. Планирование производилось на 350 выделенных станциях, маршруты поездов включали в себя от 5 (поезда, заканчивающие свое следование) до 100 станций (поезда, следующие почти через весь полигон), общее количество поездов для привязки на нитки – около 2000. Общее время работы модуля привязки поездов на нитки составляет около 3 минут, что является приемлемым временем работы.

Возможные пути улучшения алгоритма заключаются в следующем:

- Помимо процедуры, описанной в предыдущем разделе, Бертсекас и Кастанон вводят процедуру «обратного» аукциона, когда не поезда подбирают нитки под себя, а наоборот – свободные нитки графика подбирают поезда, которые следует назначить «на себя». В этой процедуре также производится переопределение значения «пороговой» цены λ: пороговая цена теперь может меняться на каждом шаге. Общий алгоритм включает в себя как «прямую», так и «обратную» процедуру аукциона. Согласно исследованиям Бертсекаса и Кастанона, использование такого комбинированного аукциона в некоторых случаях может на порядок сократить время решения задачи о назначениях.
- В работе М. Завланоса, Л. Спесивцева и Г. Паппаса [4] приводится способ распределенного мультиагентного решения задачи о назначениях с помощью аукционов. В этом варианте аукциона каждый поезд представляется в виде агента. Процедура аукциона, описанная в предыдущем разделе, сохраняется (с некоторыми изменениями) для одного агента, но с помощью правильно организованного обмена знаниями между агентами удается также добиться прироста производительности. Текущая реализация алгоритма в системе ИСУЖТ на Восточном полигоне также является мультиагентной, но в качестве агентов фигурируют только агенты станций, которые хранят данные об уже запланированных поездах и управляют последовательностью, в которой решаются задачи о назначениях.

Литература

1. Таха, Хемди А., Введение в исследование операций, 7-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005: гл 5.4 Задача о назначениях.

2. Bertsekas, D. P. (1988). The auction algorithm: A distributed relaxation method for the assignment problem. Annals of Operations Research, 14(1), 105–123.
3. Bertsekas, D., & Castanon, D. (1992). A forward/reverse auction algorithm for asymmetric assignment problems. Computational Optimization and Applications, 1(3), 277–297.
4. Zavlanos, M. M., Spesivtsev, L., & Pappas, G. J. (2008). A Distributed Auction Algorithm for the Assignment Problem. In Decision and Control, 2008. CDC 2008. 47th IEEE Conference on (pp. 1212–1217).