

Моделирование железнодорожных перевозок на территории России*

А. О. Рубцов¹, А. С. Тарасов²

¹ *Московский физико-технический институт*

² *Институт системного анализа РАН*

В статье описывается вариант транспортной задачи. С целью выбора подходящего решения и МIP-солвера была создана распределенная система тестирования данной задачи реализованная с целью нагрузочного тестирования и выбора подходящего МIP-солвера для решения приведенной задачи. Были подключены и проанализированы ЛП-солверы lp_solve, glpk.

Введение

По инициативе компании Lester занимающейся разработкой программного обеспечения класса ERP для железнодорожных экспедиторов, мы занялись задачей расчета оптимальной диспетчеризации полученных заказов между свободными вагонами.

Задача состоит в следующем. Экспедитор имеет собственный или арендованный подвижной состав (вагоны и контейнеры). При получении заказа экспедитор должен решить, принимает он или отказывается от заказа, и если принимает, то какими вагонами он планирует перевозить заказ. Оптимальное решение должно максимизировать прибыль экспедитора с учетом всех факторов, влияющих на прибыль и расходы.

1. Общее описание задачи

Здесь рассматривается задача создания системы оптимального назначения свободных вагонов заявкам на перевозки.

* Работа выполнена при поддержке гранта РФФИ № 07–07–00071-а и Совета по грантам Президента Российской Федерации (грант НШ–5511.2008.9).

Некоторый железнодорожный оператор имеет под управлением парк вагонов и обслуживает заявки на перевозки различных грузов между несколькими пунктами. Каждый день он получает различные заявки, и ему необходимо решать, на какие заявки соглашаться, и назначать на них вагоны, а от каких заявок отказываться. Количество вагонов порядка 10 000, а количество заявок порядка нескольких сотен. В каждой заявке имеется список параметров (в скобках указано обозначение используемое в математической модели): ID — уникальный номер заявки (r), *откуда* — код станции ($WF(r)$), *куда* — код станции ($WT(r)$), *валовая прибыль* — прибыль на один вагон, при полном выполнении заявки ($P_{c,r}$), *объем заявки* — объем заявки в единицах указанных в размерности (q). Все заявки считаются одновременными и взаимоисключающими.

Каждому из вагонов приспаны следующие параметры: ID — уникальный номер вагона (c), *станция на которой доступен вагон* — ($AS(c)$), *объем вагона* в тех же единицах, что и объем заявки (q_c).

Также, задача имеет набор общих параметров, не относящихся к вагонам и станциям: средняя скорость перемещения вагона (тип вагона, код станции A , код станции B , признак порожняка, кратность группы), расстояние (пункт 1, пункт 2).

2. Математическая постановка задачи

1. **Переменные.** Переменные в задаче представляются таблицей $x_{c,r}$ размерности $C \times R$, $1 \leq c \leq C$; и $1 \leq r \leq R$, где C — число доступных вагонов, R — количество заявок, поданных на данный момент.

Элементами таблицы являются числа от нуля до единицы в модели линейного программирования и целые числа нуль и один в случае смешанно-целочисленного программирования. Нуль соответствует решению не использовать данный вагон для перевозки данной заявки, единица соответствует использованию вагона в данной заявке, промежуточное значение — частичному использованию вагона (например, в случае неполной загрузки).

Результатом задачи является заполненная таблица переменных и соответствующее этой таблице значение целевой функции. Конкретный набор значений таблицы называется назначением.

2. **Целевая функция.** В линейном случае целевой функцией задачи, подлежащей максимизации, является скалярное произведение матрицы переменных с матрицей стоимостей. Матрица стоимостей — это матрица $P_{c,r}$ той же размерности, что и матрица переменных. В каждой ячейке матрицы стоимостей находится число, т.е. целевая функция является линейной зависящей от переменных задачи. В общем случае целевая функция может иметь нелинейные компоненты, т.е. иметь вид многочлена степени больше 1.

3. **Условия-ограничения.** Условия на переменные в таблице имеют различную природу и перечислены в порядке добавления к задаче (индексы c и r пробегает значения от 1 до C и от 1 до R и соответствуют номеру вагона и номеру заявки):

$$x_{c,r} \geq 0;$$

$x_{c,r} \leq 1$, первые два условия порождены природой элементов таблицы, как решений «да-нет». Нецелые значения возможны в задаче ЛП и в некоторых других случаях;

$\sum_{r \in R} x_{c,r} \leq 1$, где R — множество всех заявок. Таким образом вагон не может перевезти более одной заявки;

$\sum_c x_{c,r} \leq q(r)$, нельзя перевезти больше вагонов, чем указано в заявке; $x_{c,r} = 0$, при невозможности выполнения заявки r вагоном c . Например, если не совпадает их тип, или вагон невозможно пригнать на станцию отправления заявки к моменту отправления;

$x_{c,r} = 1$, при получении соответствующей команды от пользователя, либо для специальных «внутренних» заданий, таких как ремонт.

3. Численное моделирование

Есть вычислительно сложная транспортная задача нахождения соответствия имеющегося парка вагонов поданным заявкам с целью максимизации прибыли. Задача является NP-полной. Однако структура данной задачи позволяет легко сформулировать упрощенную задачу в терминах линейного и смешанно-целочисленного программирования (MIP). О такой реализации данной задачи в терминах линейного и смешанно-целочисленного программирования и пойдет речь ниже. После составления линейной модели задачи она подается на вход двум независимым солверам задач линейного программирования симплекс-методом, а также методом ветвей и границ для MIP-модели. Это решатели `lp_solve` и `glpk`. В дальнейшем планируются эксперименты с другими решателями.

Численный эксперимент построен с использованием распределенной математической среды MathCloud [1, 2]. Были созданы необходимые сервисы, сценарий выполнения эксперимента, подключены необходимые библиотеки. После чего были проведены численные эксперименты.

Список созданных сервисов

Генератор случайных тестов. REST-сервис `Generator2` генерирует случайные расстояния между станциями, случайные начальные расположения вагонов и случайные заявки. Принимает на вход количество заявок, среднее количество вагонов в заявке, общее количество вагонов и количество станций. Выдает четыре файла формата CSV, содержащих списки станций, расстояний между станциями, вагонов и заявок.

Сборщик модели. REST-сервис LinearModel2 составляет из исходных данных математическую модель задачи в формате lp. Принимает на вход четыре файла CSV формата, содержащие списки станций, расстояний между станциями, вагонов и заявок. Выдает MIP-модель задачи в виде файла формата lp.

Солвер lp_solve_lp_only. REST-сервис lp_solve_lp_only реализует решение задачи линейного программирования симплекс-методом. Принимает на вход файл модели линейного программирования в формате lp. Выдает результат в формате lp.

Солвер lp_solve. REST-сервис lp_solve реализует решение MIP-задачи методом ветвей и границ. Принимает на вход файл MIP-модели в формате lp. Если в исходном файле содержится модель линейного программирования, то она будет решена симплекс-методом, без привлечения метода ветвей и границ.

Преобразователь lp2mps. REST-сервис lp2mps преобразует файлы моделей линейного программирования формата lp в файлы формата mps. Принимает на вход файл формата lp, содержащий модель линейного про-

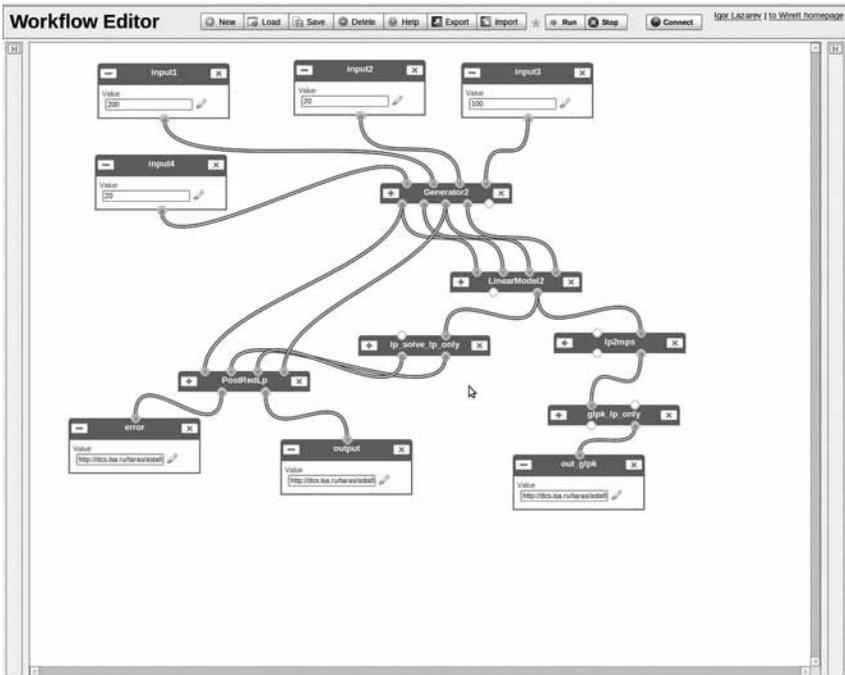


Рис. 1

граммирования или MIP-модель. Выдает файл формата mps, содержащий ту же модель.

Солвер glpk.lp_only. REST-сервис реализует решение задачи линейного программирования симплекс-методом. Принимает на вход файл модели линейного программирования в формате mps, в MIP-модели целочисленные переменные будут заменены действительными. Выдает текстовый файл, содержащий значение целевой функции.

Солвер glpk. REST-сервис glpk реализует решение MIP-задачи методом ветвей и границ. Принимает на вход файл модели MIP в формате mps. Если в исходном файле содержится модель линейного программирования, то она будет решена симплекс-методом, без привлечения метода ветвей и границ. Выдает текстовый файл, содержащий значение целевой функции.

Сервис постобработки PostRed Lp. REST-сервис PostRedLp преобразует выходные файлы сервисов lp_solve и lp_solve_lp_only в более удобный для восприятия человеком вид и добавляет к ним техническую информацию о работе солвера в целях отладки. Сервис так же требует для работы исходные файлы задачи, содержащие данные о вагонах и заявках.

В системе управления сценариями [2, 3] был создан соответствующий сценарий (см. рис. 1).

Литература

1. *Тарасов А. С.* Распределенная математическая среда на основе IARnet. Труды ИСА РАН 2008. Т. 32. С. 200–215.
2. *Афанасьев А. П., Сухорослов О. В., Тарасов А. С., Лазарев И. В., Астафьев А. С.* Научная сервис-ориентированная среда на основе технологий Web и распределенных вычислений // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской научной конференции (21–26 сентября 2009 г., г. Новороссийск). Принято к публикации.
3. *Лазарев И. В., Сухорослов О. В.* Реализация распределенных вычислительных сценариев в среде MathCloud // В настоящем сборнике.