

ЗАДАЧА ОПТИМАЛЬНОГО ПЛАНИРОВАНИЯ РАБОТ ПО ОБНОВЛЕНИЮ ЖЕЛЕЗНОДОРОЖНОЙ ЛИНИИ: ПОСТАНОВКА, АЛГОРИТМЫ РЕШЕНИЯ

Ю.О. Петровец, Д.Л. Андрианов

Рассмотрена задача долгосрочного оптимального планирования работ по обновлению железнодорожной инфраструктуры. Предложен алгоритм решения задачи, основанный на идее метода ветвей и границ и применении динамического программирования для получения нижних оценок ветвей.

Ключевые слова: железнодорожная инфраструктура, задачи оптимального планирования, метод ветвей и границ, динамическое программирование.

ВВЕДЕНИЕ

Путевое хозяйство железных дорог является наиболее капиталоемкой, трудоемкой и материалоемкой отраслью железнодорожного транспорта. Например, в России в путевом хозяйстве сосредоточено 54 % всех основных фондов железных дорог, а затраты материалов на капитальные работы в путевом хозяйстве составляют более половины от соответствующих затрат всех хозяйств железной дороги [1]. В этой связи крайне важно искать пути рационального использования имеющихся средств. В данной статье будет рассмотрена задача оптимизации расходов железнодорожной компании путем правильного выбора сроков проведения работ.

Благодаря чему можно добиться экономии? При правильном определении момента обновления актива (примеры активов — рельсы, шпалы, балластный слой, стрелочные переводы и пр.) можно избежать дополнительных расходов на техническое обслуживание (ТО) и/или необязательных экономических потерь, связанных с вводом временных ограничений на скорость передвижения. При организации более крупных проектов можно снизить себестоимость работ благодаря эффекту масштаба, а также сократить время, в течение которого линия, на которой проводятся работы, будет закрыта. Кроме того, при удачном выборе сроков проекта можно добиться экономии средств путем совместного обновления активов.

Возникающая задача оказывается нетривиальной ввиду, как правило, очень большого числа неизвестных, наличия целочисленных ограничений

и специфического характера функций, описывающих динамику основных показателей. Есть способы представить данную задачу как задачу линейного целочисленного программирования [2, 3], однако, эффективность данного подхода требует проверки.

1. ПОСТАНОВКА ЗАДАЧИ

Железнодорожная линия — это крупный инфраструктурный объект, состоящий из большого числа относительно непротяженных участков пути (сегментов). В исследуемой задаче деление на сегменты основано на инженерных соображениях: все характеристики, определяющие износ железнодорожных активов (радиус кривизны, дата последнего обновления, трафик и пр.) в пределах всей длины сегмента должны быть полностью идентичны [4]. Сегмент состоит из одного или нескольких, обычно не более трех, активов, которые обновляются или ремонтируются целиком на всей длине сегмента.

Пронумеруем все сегменты и активы. Введем обозначения: S — множество номеров всех сегментов, s — число сегментов, A — множество номеров всех активов, A_j — множество номеров активов на сегменте j , a — число всех активов, a_j — число активов на сегменте j , \underline{a}_j и \bar{a}_j — наименьший и наибольший номера актива на сегменте j . Схематически структура линии представлена на рис. 1.

Рассмотрим износ линии в моменты времени $1, \dots, T$. Предполагается, что один момент времени — это календарный или финансовый год, а

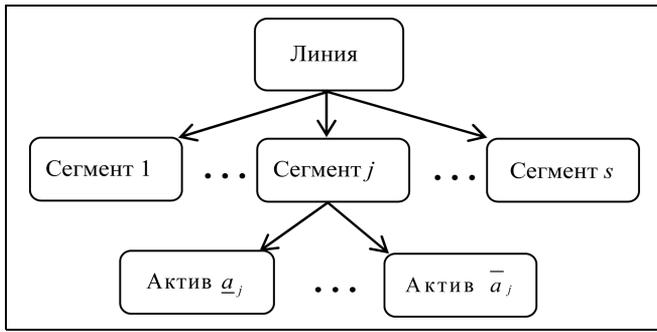


Рис. 1. Структура железнодорожной линии

длина прогнозного периода равна нескольким десяткам лет. Железнодорожная компания оказывает на износ управляющее воздействие в виде периодического обновления активов [5]. Введем оператор управления $u: \{1, \dots, T\} \rightarrow \{0, 1\}^n$, где $\{0, 1\}^n$ — вектор из n элементов, каждая компонента которого равна 0 или 1. Будем понимать $u_k(t)$ как решение обновить, если $u_k(t) = 1$ или не обновить, если $u_k(t) = 0$, актив k в момент времени t .

Все характеристики износа конкретного актива могут быть однозначно определены по его сроку службы. Далее, говоря об износе, будем подразумевать возраст актива (в годах). Обозначим $x: \{0, \dots, T\} \rightarrow \mathbb{N}^a$ — оператор, определяющий износ линии. А именно, будем понимать $x_k(t)$ — возраст актива k в момент времени t . Динамика износа определяется как

$$x_k(t) = \varphi(u_k(t), x_k(t-1)) = \begin{cases} x_k(t-1) + 1, & u_k(t) = 0, \\ 1, & u_k(t) = 1, \end{cases} \quad k \in A, \quad t \in \{1, \dots, T\}, \quad (1)$$

$$x_k(0) = x_k^0, \quad k \in A. \quad (2)$$

Введем ограничения на максимальный и минимальный срок службы:

$$x_k(t) \in X_k^t \subset \mathbb{N}, \quad k \in A, \quad t \in \{1, \dots, T\}. \quad (3)$$

В зависимости от износа и управления могут быть однозначно определены следующие показатели деятельности железнодорожной компании: расходы на обновление, расходы на ТО, экономические потери. Все показатели вычисляются для каждого сегмента, в зависимости от состояния его активов. Для удобства введем обозначения:

$$u^j(t) = (u_{a_j}(t), \dots, u_{\bar{a}_j}(t)), \quad x^j(t) = (x_{a_j}(t), \dots, x_{\bar{a}_j}(t)), \quad j \in S, \quad t \in \{1, \dots, T\}.$$

Расходы на обновление на сегменте j в момент времени t определяются функцией $f_j^{sr}(u^j(t), t)$, $j \in S, t \in \{1, \dots, T\}$.

Эти расходы обладают важной особенностью — они зависят от сочетания обновляемых активов. Эта особенность является одной из ключевых предпосылок для оптимизации, так как позволяет путем объединения работ по обновлению разных видов активов на одном сегменте уменьшить суммарную стоимость обновления. Справедливо неравенство

$$\sum_{k \in A_j} u_k(t) f_j^{sr}(e^j(k - a_j + 1), t) \geq f_j^{sr}(u^j(t), t), \quad j \in S, \quad t \in \{1, \dots, T\},$$

где $e^j: \{1, \dots, a_j\} \rightarrow \{0, 1\}_j^a$ определяется равенствами

$$e^j(i) = \begin{cases} e_k^j(i) = 1, & k = i, \\ e_k^j(i) = 0, & k \neq i, \end{cases} \quad j \in S.$$

Расходы на ТО на сегменте j в момент времени t определяются функцией

$$f_j^{sm}(x^j(t), t) = \sum_{k \in A_j} f_k^{am}(x_k(t), t), \quad j \in S, \quad t \in \{1, \dots, T\},$$

где $f_k^{am}(x_k(t), t)$ — расходы на ТО отдельного актива k в момент времени t .

Расходы на ТО начинают стремительно возрастать после превышения активом срока службы как из-за удорожания, так и из-за учащения работ отдельных видов [6, 7].

Возникновение экономических потерь связано с уменьшением пропускной способности, вызванной вводом ограничений на скорость передвижения пассажирских и грузовых поездов. Чем больше срок службы актива, тем больше вероятность, что ограничение будет введено.

Вероятность ввода ограничения в результате чрезмерного износа актива k в момент времени t

$$P(ATSR_{k,t}) = f_k^{ah}(x_k(t), t), \quad k \in A, \quad t \in \{1, \dots, T\}$$

считается равной нулю, если срок службы актива меньше инженерного срока.

Сегмент может состоять из нескольких активов, и ограничение вводится в том случае, если хотя бы один актив не удовлетворяет требованиям безопасности. Ввод ограничений в результате чрезмерного износа одного актива считается статистически независимым от износа другого актива. Математическое ожидание экономических потерь на сегменте j в момент времени t запишем в виде

$$\begin{aligned} & f_j^{sh}(x^j(t), t) = \\ & = P(ATSR_{a_j,t} \text{ или } \dots \text{ или } ATSR_{\bar{a}_j,t}) f_j^{tsr}(t), \quad j \in S, \quad t \in \{1, \dots, T\}, \end{aligned}$$

где $f_j^{sr}(t)$ — размер экономических потерь (в случае ввода ограничения) на сегменте j в момент времени t .

Известно, что работы по обновлению активов лучше объединять в *проекты*. Это позволяет учесть: эффект экономии от масштаба, сокращение времени и соответственно расходов из-за неиспользования линии, недовольство пользователей железнодорожных услуг. Поэтому добавляются ограничения на паузы между обновлениями:

$$\begin{aligned} \|u(t)\| \cdot \|u(t - \psi)\| &= 0, \quad t \in \{1, \dots, T\}, \\ \psi &\in \{1, \dots, \rho\}, \quad t - \psi > 0, \end{aligned} \quad (4)$$

где $\rho > 1$ — пауза между обновлениями.

Для решения проблем, связанных с ограниченностью прогнозного периода, вводятся штрафы за обновление активов раньше рекомендованного срока:

$$\begin{aligned} f_j^{sp}(u^j(t), x^j(t-1), t) &= \sum_{k \in A_j} f_k^{ap}(u_k(t), x_k(t-1), t) = \\ &= \sum_{k \in A_j} u_k(t) f_j^{sr}(e^j(k - a_j + 1), t) \times \\ &\times \frac{x_k(t-1) + 1}{f_k^{life}(t - x_k(t-1) - 1)}, \quad j \in S, \quad t \in \{1, \dots, T\}, \end{aligned}$$

где $f_j^{life}(\tau)$ — функция, определяющая срок службы актива k , при условии, что в последний раз он был обновлен в момент времени τ , а

$$\begin{aligned} z_k(t) &= \omega(x_k(t-1)) = x_k(t-1) + 1, \\ k &\in A, \quad t \in \{1, \dots, T\}. \end{aligned} \quad (5)$$

Целевая функция

$$\begin{aligned} &\sum_{t=1}^T \sum_{j \in S} f_j^s(u^j(t), x^j(t), z^j(t)) = \\ &= \sum_{t=1}^T \sum_{j \in S} (f_j^{sm}(x^j(t), t) + f_j^{sr}(u^j(t), t) + \\ &+ f_j^{sh}(x^j(t), t) + f_j^{sp}(u^j(t), z^j(t), t)) \rightarrow \min. \end{aligned} \quad (6)$$

Чтобы не усложнять запись, будем полагать, что все слагаемые содержат множитель дисконтирования.

2. АНАЛИЗ ЗАДАЧИ КАК ЗАДАЧИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Сформулированная задача представляет собой задачу динамического программирования, в которой T — число шагов (этапов), u — управляющее

воздействие, x и z — фазовые переменные, а множество допустимых процессов управления

$$\begin{aligned} G = \{(u, x, z) : u : \{1, \dots, T\} \rightarrow \{0, 1\}^a, \\ x, z : \{0, \dots, T\} \rightarrow N^a, (1)–(5)\} \end{aligned}$$

(здесь и далее для упрощения выражений вместо громоздких формул иногда будем писать их порядковые номера).

Несмотря на то, что формально задача относится к хорошо известному классу задач, воспользоваться напрямую для ее решения общепринятым методом — принципом оптимальности Беллмана — не представляется возможным ввиду так называемого «проклятия размерности» [8].

Отметим несколько особенностей числа состояний системы, касающихся его определения:

— оно зависит от числа активов a ; с помощью принципа оптимальности за разумное время можно найти решение только при a не больше трех—четырёх;

— оно ограничено из-за наличия ограничений (3);

— его можно резко сократить, если для некоторого подмножества моментов времени $B \subset \{1, \dots, T\}$ дополнительно ввести ограничения вида $u(t) = \text{col}(0, \dots, 0)$, $t \in B$.

Для иллюстрации этих особенностей рассмотрим два примера. В первом из них (рис. 2) представлена динамика числа состояний для одного сегмента из 1—3 активов при следующих условиях: все активы были обновлены в $t = 0$, максимальный срок службы всех активов 50 лет, пауза между обновлениями 5 лет.

Видно, что число состояний системы для трех активов существенно больше, чем для двух или

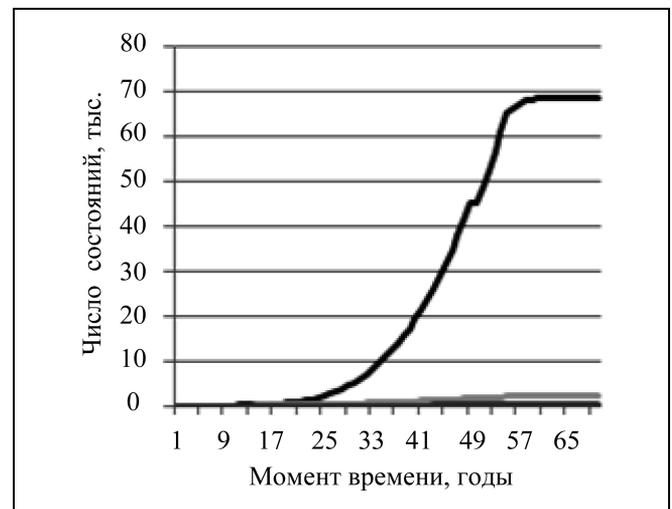


Рис. 2. Пример 1: динамика числа состояний:

— 3 актива, — 2 актива, — 1 актив

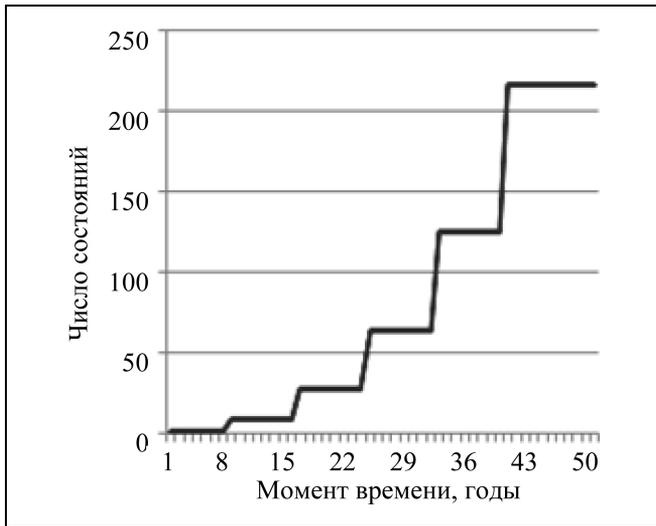


Рис. 3. Пример 2: динамика числа состояний

одного. Начиная с некоторого момента, число состояний стабилизируется. В случае трех активов максимальное число состояний равно 68 660. Если запретить обновления в некоторые годы $B = \{1, \dots, T\} \setminus \{8, 16, 24, 32, 40, 48\}$, то максимальное число состояний системы из трех активов будет значительно меньше — 216 (рис. 3). Данное свойство будет использовано при решении задачи методом ветвей и границ.

3. ПРИМЕНЕНИЕ МЕТОДА ВЕТВЕЙ И ГРАНИЦ

3.1. Вектор проектов

Под проектами допустимого управления u будем понимать булев вектор $p = \text{col}(\max u(1), \dots, \max u(T))$; т. е. проект применительно к отдельному активу означает лишь возможность или разрешение провести обновление, а не сам факт обновления.

Процессом управления, удовлетворяющим вектору проектов $p = \text{col}(p_1, p_2, \dots, p_t)$, $t \leq T$, будем называть процесс (u, x, z) , принадлежащий множеству

$$G_p = \{(u, x, z) \in G : u(t) = \text{col}(0, \dots, 0), \forall t : p_t = 0\}.$$

Очевидно,

$$\begin{aligned} \min_u \{(6) | (u, x, z) \in G\} &\equiv \\ &\equiv \min_p \{ \min_u \{(6) | (u, x, z) \in G_p\} \}. \end{aligned}$$

Заметим, что для фиксированного вектора проектов p , элементы которого являются проектами некоторого допустимого управления u , задача «рас-

падается» на множество небольших подзадач оптимизации работ для каждого сегмента:

$$\min_u \{(6) | (u, x, z) \in G_p\} = \sum_{j \in S} \left(\min_{u^{s_j}} \left\{ \sum_{t=1}^T f_j^s(u^{s_j}(t), x^{s_j}(t), z^{s_j}(t)) | (u^{s_j}, x^{s_j}, z^{s_j}) \in G_p^{s_j} \right\} \right),$$

где $G_p^{s_j} = \{(u, x, z) : u : \{1, \dots, T\} \rightarrow \{0, 1\}^{a_j}, x, z : \{0, \dots, T\} \rightarrow \mathbb{N}^{a_j}, (7)–(10)\}$,

$$u(t) = \text{col}(0, \dots, 0), \forall t : p_t = 0, \quad (7)$$

$$\begin{aligned} x_k(t) &= \varphi(u_k(t), x_k(t-1)), \quad z_k(t) = \omega(x_k(t-1)), \\ k &\in \{1, \dots, a_j\}, \quad t \in \{1, \dots, T\}, \end{aligned} \quad (8)$$

$$x_k(0) = x_{k+a_j-1}^0, \quad k \in \{1, \dots, a_j\}, \quad (9)$$

$$x_k(t) \in X_{k+a_j-1}^t, \quad k \in \{1, \dots, a_j\}, \quad t \in \{1, \dots, T\}. \quad (10)$$

Каждая такая подзадача достаточно тривиально решается методом динамического программирования, о чем говорилось выше.

Идея ввода дополнительных ограничений с помощью вектора (векторов) проектов может оказаться конструктивной, если существует априорная информация о том, какие моменты времени больше всего подходят для реализации проектов. Если же такой информации нет, то попытки экстенсивным способом перебрать все возможные векторы проектов, скорее всего, окажутся неудачны, так как их число может оказаться очень велико. Например, при $T = 50$, $\rho = 5$ число векторов может достигать 153 315.

3.2. Процедуры ветвления

Осуществить поиск наилучшего вектора проектов можно с помощью метода ветвей и границ. Рассмотрим два варианта процедуры ветвления. Условно говоря, первый вариант отвечает на вопрос «Что должно произойти в следующий момент времени?». Процедура ветвления множества допустимых процессов управления схематически представлена на рис. 4.

Второй вариант отвечает на вопрос «Когда необходимо организовать следующий проект?». Для того чтобы сузить перечень моментов времени, в которые может быть организован проект, но при этом не исключить оптимального решения, можно предложить процедуру ветвления, показанную на рис. 5. Она должна при необходимости обеспечивать перебор всех векторов проектов. Однако данной процедурой исключаются из рассмотрения все векторы, в которых число подряд идущих

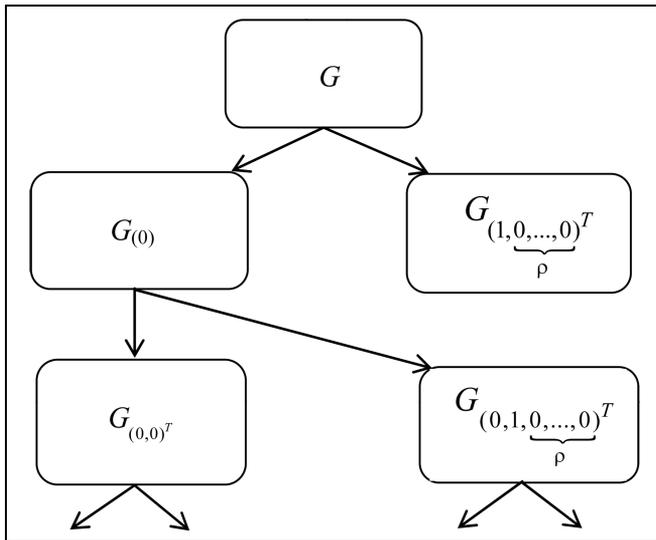


Рис. 4. Процедура ветвления (первый вариант)

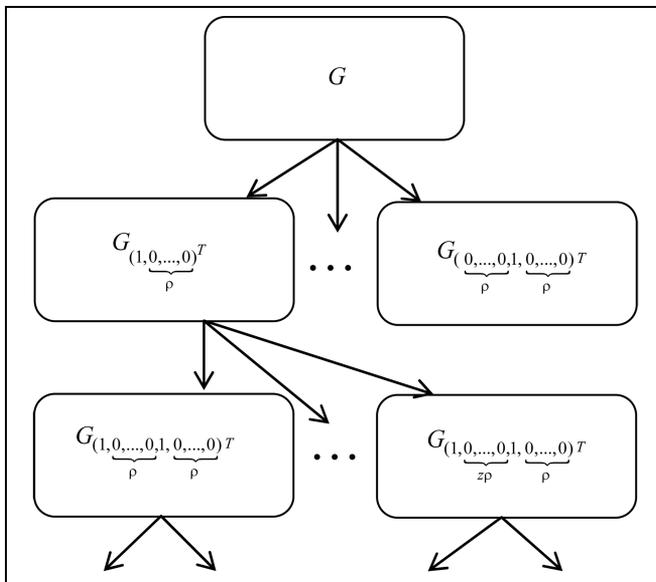


Рис. 5. Процедура ветвления (второй вариант)

нулевых элементов превосходит 2ρ . Это допустимо ввиду следующего свойства задачи (1)–(6): *если* $p^0, p^1 \in \{0, 1\}^T : p_t^0 \leq p_t^1, \forall t \in \{1, \dots, T\}$, *то* $\min_u \{(6)|(u, x, z) \in G_p^0\} \geq \min_u \{(6)|(u, x, z) \in G_p^1\}$. Это справедливо, поскольку $G_p^0 \subseteq G_p^1$.

3.3. Оценивание ветви

Для получения нижней оценки ветви в методе ветвей и границ обычно искусственно расширяют область допустимых процессов управления таким образом, чтобы можно было легко получить реше-

ние на этой области [9]. В этом случае полученное решение будет гарантированно «не хуже», чем решение исходной задачи.

Рассмотрим произвольную ветвь b . Будем искать оценку в виде

$$\begin{aligned} & \min_u \{(6)|(u, x, z) \in G_b\} \geq \\ & \geq \sum_{j \in S} \min_{u^{s_j}} \left\{ \sum_{t=1}^{d_b} f_j^s(u^{s_j}(t), x^{s_j}(t), z^{s_j}(t), t) + \right. \\ & \quad \left. + M_j(x^{s_j}(d_b), d_b)|(u^{s_j}, x^{s_j}, z^{s_j}) \in F_b^{s_j} \right\}, \end{aligned}$$

где d_b — число элементов в векторе b и

$$\begin{aligned} F_b^{s_j} &= \{(u, x, z) : u : \{1, \dots, d_b\} \rightarrow \{0, 1\}^{a_j}, \\ & \quad x, z : \{0, \dots, d_b\} \rightarrow \mathbb{N}^{a_j}\}, \quad (11)–(14), \end{aligned}$$

$$u(t) = \text{col}(0, \dots, 0), \quad t \in \{1, \dots, d_b\} : b_t = 0, \quad (11)$$

$$\begin{aligned} x_k(t) &= \varphi(u_k(t), x_k(t-1)), \quad z_k(t) = \omega(x_k(t-1)), \\ & \quad k \in \{1, \dots, a_j\}, \quad t \in \{1, \dots, d_b\}, \quad (12) \end{aligned}$$

$$x_k(0) = x_{k+a_j-1}^0, \quad k \in \{1, \dots, a_j\}, \quad (13)$$

$$x_k(t) \in X_{k+a_j-1}^t, \quad k \in \{1, \dots, a_j\}, \quad t \in \{1, \dots, d_b\}, \quad (14)$$

а $M_j(x^\tau, \tau)$ — любая функция, удовлетворяющая условию:

$$\begin{aligned} M_j(x^\tau, \tau) &\leq \min_{u^{s_j}} \left\{ \sum_{t=\tau+1}^T f_j^s(u^{s_j}(t), x^{s_j}(t), \right. \\ & \quad \left. z^{s_j}(t), t) | (u^{s_j}, x^{s_j}, z^{s_j}) \in G_{x^\tau, \tau}^{s_j} \right\}, \quad (15) \end{aligned}$$

где $G_{x^\tau, \tau}^{s_j} = \{(u, x, z) : u : \{\tau+1, \dots, T\} \rightarrow \{0, 1\}^{a_j}, \\ x, z : \{\tau, \dots, T\} \rightarrow \mathbb{N}^{a_j}\}, \quad (16)–(19),$

$$x(\tau) = x^\tau, \quad (16)$$

$$\begin{aligned} x_k(t) &= \varphi(u_k(t), x_k(t-1)), \quad z_k(t) = \omega(x_k(t-1)), \\ & \quad k \in \{1, \dots, a_j\}, \quad t \in \{\tau+1, \dots, T\}, \quad (17) \end{aligned}$$

$$\begin{aligned} \|u(t)\| \cdot \|u(t-\psi)\| &= 0, \quad t \in \{\tau+1, \dots, T\}, \\ \psi &\in \{1, \dots, \rho\}, \quad t-\psi > \tau, \quad (18) \end{aligned}$$

$$\begin{aligned} x_k(t) &\in X_{k+a_j-1}^t, \quad k \in \{1, \dots, a_j\}, \\ & \quad t \in \{\tau+1, \dots, T\}. \quad (19) \end{aligned}$$

Таким образом, нахождение оценки сводится к решению s задач оптимизации работ на сегменте на периоде $t \in \{1, \dots, d_b\}$ с терминальным членом. Решение данных задач упрощается благодаря на-



личию ограничений (11). Эффективность оценки будет определяться сложностью вычисления функций $M_j(x^\tau, \tau)$, а также грубостью приближения (15). Точный минимум в формуле (15) также является оценкой, однако для его определения могут потребоваться ресурсоемкие вычисления (например, как случай трех активов, см. рис. 2).

Для сокращения объема вычислений целесообразно найти функции $f_j^{ss}(u, x, z, t) : \{0, 1\}_j^a \times \mathbb{N}_j^a \times \mathbb{N}_j^a \times \{1, \dots, T\} \rightarrow \mathbb{N}$, сепарабельные по элементам векторов u, x и z , т. е. такие, чтобы для любых u, x, z, t выполнялось

$$f_j^s(u, x, z, t) \geq f_j^{ss}(u, x, z, t) = \sum_{k=1}^{a_j} f_j^{ak}(u_k(t), x_k(t), z_k(t), t), \quad j \in S,$$

где $f_j^{ak}(u, x, z, t) : \{0, 1\} \times \mathbb{N} \times \mathbb{N} \times \{1, \dots, T\} \rightarrow \mathbb{N}$, $j \in S, k \in \{1, \dots, a_j\}$.

В этом случае для получения оценки нужно решить локальные задачи оптимизации для каждого отдельного актива, размерность которых принципиально меньше (случай одного актива (см. рис. 2). При использовании сепарабельной функции опять расширяется множество процессов управления из-за ограничения (18), т. е. оценка становится более грубой.

Функции $f_j^{sm}(x, t)$ и $f_j^{sp}(u, z, t)$ сепарабельны. В качестве функции всюду не превосходящей $f_j^{sr}(u, t)$ и сепарабельной по элементам вектора управления u можно использовать следующую функцию

$$f_j^{ssr}(u, t) = \sum_{k=1}^{a_j} u_k(f_j^{sr}(e^j(k), t) - \max_{v \in \{0, 1\}^{a_j} \setminus \text{col}(0, \dots, 0)} \left\{ \left(\sum_{m=1}^{a_j} v_m f_j^{sr}(e^j(m), t) - f_j^{sr}(v, t) \right) \middle/ \sum_{m=1}^{a_j} v_m \right\}), \quad j \in S.$$

Смысл этой функции заключается в том, что при обновлении конкретного актива всегда используются расходы на обновление за вычетом максимально возможной экономии от совместного обновления в пересчете на один актив.

Утверждение 1. Для любого $n \in \mathbb{N}$ справедливо неравенство

$$P(A_1 \text{ или } \dots \text{ или } A_n) \geq \sum_{i=1}^n \frac{1 - (1 - P(A_i))^n}{n}, \quad (20)$$

где A_1, \dots, A_n — любые независимые события.

Доказательство. Применим метод математической индукции. Для $n = 1$: $P(A_1) \equiv P(A_1)$. Предполагаем, что для некоторого $n = k$ выполняется

$$P\left(\underset{\text{Событие } B}{(A_1 \text{ или } \dots \text{ или } A_k)}\right) \geq \sum_{i=1}^k \frac{1 - P^k(\bar{A}_i)}{k} \Rightarrow \Rightarrow \sum_{i=1}^k \frac{P^k(\bar{A}_i)}{k} \geq P(\bar{B}).$$

Тогда для $n = k + 1$, с одной стороны, можно записать

$$P(A_1 \text{ или } \dots \text{ или } A_k \text{ или } A_{k+1}) = = 1 - P(\bar{B})P(\bar{A}_{k+1}) \geq 1 - P(\bar{B}).$$

С другой стороны, выполняется

$$\begin{aligned} \sum_{i=1}^{k+1} \frac{1 - P^{k+1}(\bar{A}_i)}{k+1} &= 1 - \sum_{i=1}^k \frac{P^{k+1}(\bar{A}_i)}{k+1} - \frac{P^{k+1}(\bar{A}_{k+1})}{k+1} \leq \\ &\leq 1 - \sum_{i=1}^k \frac{P^k(\bar{A}_i)}{k} - \frac{P^{k+1}(\bar{A}_{k+1})}{k+1} \leq \\ &\leq 1 - P(\bar{B}) - \frac{P^{k+1}(\bar{A}_{k+1})}{k+1} \leq 1 - P(\bar{B}). \end{aligned}$$

Откуда следует справедливость неравенства (20). Утверждение 1 доказано. ♦

Заметим, если вероятности событий A_1, \dots, A_n равны, то неравенство (20) обращается в тождество. Таким образом, в качестве функции, сепарабельной по x и всюду не превосходящей $f_j^{sh}(x, t)$, можно использовать функцию

$$f_j^{ssh}(x, t) = \frac{f_j^{tsr}(t)}{a_j} \sum_{k=1}^{a_j} \left(1 - (1 - f_{k+a_j-1}^{ah}(x_k, t))^{a_j} \right), \quad j \in S.$$

3.4. Программные способы ускорения вычислений

Рассмотрим некоторые особенности алгоритма, которые могут позволить проводить компьютерные вычисления быстрее:

- организация параллельных вычислений;
- кэширование значений функций $M_j(x^\tau, \tau)$;
- повторное использование решений.

Очевидно, что при вычислении ветви можно искать решения подзадач оптимизации для каждого сегмента в отдельном потоке, что позволит создать масштабируемое приложение.

Также нет необходимости каждый раз рассчитывать значения функции $M_j(x^\tau, \tau)$, если для ее вычисления используются сепарабельные функции, описанные в п. 3.3. Вместо этого стоит один раз вычислить значения этой функции для всех возможных параметров x^τ и τ .

Параметры и время расчета

Линия	Время, с	Число неизвестных	Пауза
71	258	331 600	5
16	134	249 700	5
98	98	150 250	5
21	66	263 200	8
15a	43	102 550	5
23b	36	337 800	10
32	35	191 050	10
83	32	146 350	10
55	32	132 400	10
85	30	148 650	10

При ветвлении узла дерева решений можно повторно использовать результаты, полученные на предыдущем шаге. Необходимо пересчитывать не все этапы динамического программирования, а только те, которые уточняются в новой ветви. Данным приемом целесообразно воспользоваться при поиске решения с помощью «жадного» алгоритма. Иначе для железнодорожных линий с большим числом сегментов хранение всех матриц стоимостей перехода может требовать большого объема оперативной памяти.

4. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Данная модель и метод оптимизации были применены в компании «Vanedanmark» (Дания). На реальных данных, описывающих железнодорожную сеть этой страны, были определены оптимальные проекты на прогнозном периоде, равном 50-ти годам.

Общее число неизвестных равно 5 249 950. Чистое время расчета задач оптимизации для 49 линий составило 20 мин 23 с. Расчет производился в четыре потока на компьютере с параметрами: Intel® Core i5 CPU K 655 @ 3,20 ГГц 3,19 ГГц; 16 Гб RAM; x64. Приложение написано на языке C# под .NET Framework 4 и интегрировано в АК «Прогноз-5» [10]. В таблице показаны основные параметры и время десяти наиболее продолжительных расчетов (в таблице приведены официальные номера линий Датской железной дороги). Видно, что пауза между обновлениями оказывает большее влияние, чем число неизвестных.

ЗАКЛЮЧЕНИЕ

Сформулирована задача оптимального планирования работ по обновлению инфраструктуры железнодорожной линии. Данная задача может содержать сотни тысяч неизвестных, что осложняет

применение классических методов решения, таких как принцип оптимальности Беллмана. Однако используя специфические особенности постановки, удается предложить алгоритм поиска решения, основанный на методе ветвей и границ, показавший высокую эффективность на реальных данных.

Дальнейшие исследования могут быть посвящены обобщению задачи для всей железнодорожной сети с учетом ресурсных ограничений и анализу специфики обновления нетипичных объектов инфраструктуры (мостов, тоннелей и др.), а также рассмотрению различных методик определения экономических потерь.

ЛИТЕРАТУРА

1. Петров Ю.Д., Купоров А.И., Шкурина Л.В. Планирование в структурных подразделениях железнодорожного транспорта. — М.: Высшее образование, 2008. — 340 с.
2. Поносков Д.А., Чадов А.Л. К вопросу о вычислительной сложности нахождения решения одного класса задач целочисленного программирования // Тр. 2-й междунар. конф. «Трехмерная визуализация научной, технической и социальной реальности. Технологии высокополигонального моделирования» / УдГУ. — Ижевск, 2010. — Т. 1. — С. 125—127.
3. Муртаф Б. Современное линейное программирование. — М.: Мир, 1984. — 224 с.
4. Carretero J., Gonzalez J., Perez J., Romera R. Optimal railway infrastructure maintenance and repair policies to manage risk under uncertainty with adaptive control / Universidad Carlos III de Madrid, 2006. — 15 p.
5. Цыгичко В.Н. Руководителю — о принятии решений. — 2-е изд. — М.: Инфра-М, 1996. — 272 с.
6. Riessberger K. Key elements in the maintenance of high speed track // Railway Gazette International. — 1989. — Vol. 145, N 3. — P. 147—149.
7. Mlinaric T.J., Pirnar M. Optimizing track infrastructure availability // Promet — Traffic & Transportation. — 2009. — Vol. 21, N 2. — P. 113—121.
8. Беллман Р. Динамическое программирование. — М.: Иностранная литература, 1960. — 400 с.
9. Зайченко Ю.П. Исследование операций. — Киев: Вища школа, 1975.
10. Свидетельство Российского агентства по патентам и товарным знакам № 2005610980 от 22.04.2005 об официальной регистрации программы для ЭВМ / Д.Л. Андрианов, Г.К. Полушкина и др.

Статья представлена к публикации членом редколлегии В.Н. Бурковым.

Юрий Олегович Петровец — аспирант, Пермский государственный национальный исследовательский университет; вед. специалист, ЗАО «Прогноз», г. Пермь, ☎ (342) 240-36-63, доб. 11-58, ✉ petrovets@prognoz.ru,

Дмитрий Леонидович Андрианов — д-р физ.-мат. наук, зав. кафедрой, профессор, Пермский государственный национальный исследовательский университет; ген. директор, ЗАО «Прогноз», г. Пермь, ☎ (342) 237-17-53, ✉ andrianov@econ.psu.ru.