ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2022, Vol. 16, No. 3, pp. 512–523. © Pleiades Publishing, Ltd., 2022.
Russian Text © The Author(s), 2022, published in Diskretnyi Analiz i Issledovanie Operatsii, 2022, Vol. 29, No. 3, pp. 64–84.

Modeling and Optimizing Large-Scale Production-Level Transportation Systems

E. A. Nurminskiy[1*] and N. B. Shamray[2]**

[1]*Far Eastern Federal University, Vladivostok, 690090 Russia*
[2]*Institute of Automation and Control Processes, Far East Branch, Russian Academy of Sciences, Vladivostok, 690041 Russia*
*e-mail: *nurminskiy.ea@dvfu.ru, **shamray@dvo.ru*

Received May 2, 2022; revised May 2, 2022; accepted May 5, 2022

Abstract—Large-scale economic modeling is becoming a reality for major businesses, and it pushes their analytic and planning departments into very complicated areas of big data analytics and control. At the same time, it demands research communities in academia and elsewhere to develop adequate tools to operate models with millions of variables and gigabytes of data, where traditional off-the-shelf solutions fail. In the present paper, we describe our experience with one rather common high-dimensional logistics problem and some of the mathematical and computational ideas we applied to deal with it.

Keywords: *large-scale economic modeling, production-level transport expedition system, linear optimization, projection algorithm*

DOI: 10.1134/S1990478922030152

INTRODUCTION

From the point of view of the theory of operations research, the paper considers the linear programming problem

$$\min_{x \in X} cx, \tag{1}$$

which essentially lies in the optimization of the management of a large fleet of railway cars of various types in the interests of a large transport operator that provides its customers with services for the delivery of heterogeneous goods.

Omitting some specific details, problem (1) can be considered as a multi-index transportation linear optimization problem. The main difficulty in working with it arises from the intricate relationships between the indices of the sets used to describe various technological constraints. In real life, these relationships change from one implementation of the problem to another, often depending on the special decisions of the operator, the state of the transport infrastructure, etc.

Undoubtedly, the higher the degree of detail and realism of the developed model, the better the results of its implementation. The price to pay for achieving high quality is an increase in the dimension of the problem in terms of both resource costs for maintaining the initial data and a considerable increase in the number of variables and constraints. Even if each of the index sets of problem (1) has a rather modest cardinality, their multiplicative effect easily increases the dimension of the final problem to hundreds of millions of variables, which is the main difficulty for its modeling and solution.

Fortunately, the class of forwarding linear optimization problems has a number of specific features that can be used to develop new approaches to their solution. One of these features is the high sparseness of the matrices that form the problem, and this is usually taken into account in modern optimization software (see, e.g., [1–4]).

Another direction of work with optimization problems of gigantic dimension is decomposition—the division of the original problem into smaller subproblems that are solved independently and the subsequent organization of the process of coordinating solutions to obtain a common optimum. This is a very popular area of research since the pioneering work by G.B. Dantzig and P. Wolfe [5], and it is simply impossible to consider even the main directions of these studies within the framework of this article. However, here, too, success largely depends on the structure of the problem being solved; therefore, the decomposition approach is very peculiar.

The paper proposes to combine both of the mentioned approaches to solve the problem under consideration, namely, to decompose the original problem into several main blocks and then rely on automatic structure detection within each of the blocks using sparsity patterns.

The paper is structured as follows. In Sec. 1, we give a brief statement of the problem and further, in Sec. 2, the corresponding mathematical formalization. Section 3 describes the experience of software implementation and notes the structural features of the problem. Section 4 discusses the theoretical aspects of solving high-dimensional linear problems and presents the results of a structural analysis of some practical forwarding problems. The results obtained are summarized in the conclusions.

1. BRIEF DESCRIPTION OF THE PROBLEM

Consider the problem of managing a fleet of freight railcars, which is faced by transport operators fulfilling orders for the delivery of goods over the railroad network. The transportation network is represented by a set of railway routes connecting departure and arrival stations. Each route is considered as a separate independent object with its own characteristics (travel time, cost, etc.). The route does not contain any other internal routes. Based on the received orders, a preliminary schedule for the delivery of goods from suppliers to consumers is formed. Partial execution of orders is allowed. The operator has its own cars of various types at its disposal, and it is also possible to take cars for a short-term lease. The car type determines the cargo allowed for transportation and the routes allowed for use. Ensuring orders with cars is carried out by moving empty cars between stations. The goal of the transport operator is to maximize the total operational profit over the final planning time horizon.

The actual size of the considered railway network is approximately 1000 stations and 200 thousand routes. In practice, a little more than 30 types of cars and 100 types of cargo are considered. The monthly delivery schedule consists of more than 10 thousand shipments.

For each day of the planning period, the stations know the number of cars of each type ready for dispatch. The orders received for execution are characterized by the departure and arrival stations, the type of cargo, the maximum number of required cars, the revenue from one sent car, and the order completion time (loading/passage/unloading). The order fulfillment schedule includes the days of delivery and the number of cars for loading. The allowable deviation from planned loading days is set. The operator's cost of moving one empty car depends on its type and the type of the last cargo carried in this car. The duration of an empty passage is known for each route.

The operating profit is determined by the total proceeds from the execution of orders minus the costs of moving empty cars and short-term rental, fees for demurrage of cars at stations, etc. The transport operator needs, first, to select the most profitable orders for execution, second, to determine the chains of no-load runs that provide orders with cars, and third, if necessary, to rent cars to fulfill orders.

At the end of the current planning period, a new version of the car location is formed at the stations, which, in turn, determines the initial state of the system for the next planning period and thus affects the operator's future income. It is assumed that the operator can forecast future orders with volume data and expected revenue. This forecast is used to efficiently return cars to the logistics of future transportation.

2. FORMALIZATION OF THE PROBLEM

Owing to its substantial size and variety of internal processes and input data, a mathematical formalization of the problem under consideration presents a considerable difficulty. Note that the model is in constant development and responds to changes in technology, the structure of supply and demand, the transformation of the transport network, etc. The mathematical model below describes the main building blocks of the transportation system, its dynamics, and flow balances and also shows the scale and complexity of the problem.

2.1. Notation

Following the tradition of most algebraic modeling languages (see, for example, AMPL [6, 7], GAMS [8], and GMPL [9]), we split the notation introduced into three categories of sets, parameters, and variables. The sets define the objects of the problem and their discrete characteristics. Parameters specify numerical values of measurable data associated with various combinations of elements from the sets. Variables are unknown quantities whose values are determined in the course of solving the problem.

Sets:

T, days of the planning period ($t \in T$).

S, departure-arrival stations ($s \in S$).

S^0, stations where storage of empty cars is prohibited ($S^0 \subset S$).

$R = S \times S$, railway routes ($(s_1, s_2) \in R$).

V, car types ($v \in V$).

K, cargo types ($k \in K$).

O, shipping orders ($o \in O$).

$\overline{O}$, orders forecast for the period following T ($\overline{o} \in \overline{O}$).

Parameters:

q^t_{svk}, number of cars of type v ready for use after unloading cargo k at station s on day t.

χ^t_o, the maximum number of cars of order o scheduled for loading on day t.

p_{ov}, revenue per one car of type v delivered on order o.

τ_o, duration of fulfillment of order o (loading/passage/unloading) (in days).

σ_o, the maximum allowable deviation (in days) from the fulfillment schedule of order o (i.e., cars can be delivered for loading on the days of the interval $[t - \sigma_o, t + \sigma_o]$).

ν_o, costs for short-term rental of cars for the execution of order o.

$\overline{\chi}_{\overline{o}}$, a forecast of the maximum number of railcars for a future order $\overline{o}$.

$\overline{p}_{\overline{o}v}$, revenue forecast per one car v of a future order $\overline{o}$.

$c_{s_1 s_2 vk}$, costs of moving one empty car of type v after cargo k from station s_1 to station s_2.

$\theta_{s_1 s_2}$, duration of one empty car run from station s_1 to station s_2 (in days).

λ_s, the daily cost of storing one car at station s.

Each order o is characterized by the departure station $s_1(o)$, the destination $s_2(o)$, and the type $k(o)$ of cargo to be transported. The schedule for the supply of cars for loading is given by the set $Q = \big\{(t, o) \in T \times O \mid \chi^t_o > 0\big\}$.

Variables:

y^t_{svk}, the number of empty cars of type v after cargo k accumulated at station s on day t.

$x^t_{s_1 s_2 vk}$, the number of empty cars of type v after cargo k sent from station s_1 to station s_2 on day t.

z^t_{ovi}, the number of loaded cars of type v dispatched in accordance with the order fulfillment schedule record $(t, o) \in Q$ with a deviation of $i \in [-\sigma_o, \sigma_o]$ days.

ζ^t_o, the number of cars in short-term lease involved for the implementation of dispatch $(t, o) \in Q$.

$\overline{y}_{sv}$, the estimated number of cars of type v at station s in the future period.

$\overline{x}_{s_1 s_2 v}$, the estimated number of empty cars of type v sent from station s_1 to station s_2 in the future period.

$\overline{z}_{\overline{o}v}$, the estimated number of loaded wagons of type v shipped on order $\overline{o}$ in the future period.

2.2. Constraints

A characteristic feature of forwarding problems is the conditions for preserving car flows—cars do not disappear anywhere and do not appear from nowhere. In the context of the problem under consideration, these conditions are formalized as balances for departures and arrivals of cars at railway stations and are mandatory for transportation.

The condition for preserving car flows must be met on each day of the planning period t for each station s for each car type v previously carrying cargo k,

$$y^t_{svk} = q^t_{svk} + y^{t-1}_{svk} + \sum_{s_1\in S} x^{t-\theta_{s_1 s}}_{s_1 svk} - \sum_{s_2\in S} x^t_{ss_2 vk} + \sum_{\substack{(t-i-\tau_o,o)\in Q:\\ s_2(o)=s,\, k(o)=k,\\ i\in[-\sigma_o,\sigma_o]}} z^{t-i-\tau_o}_{ovi} - \sum_{\substack{(t-i,o)\in Q:\\ s_1(o)=s,\\ i\in[-\sigma_o,\sigma_o]}} z^{t-i}_{ovi}. \tag{2}$$

To estimate the number of cars at stations in the future period, we also rely on the principle of conservation of flows and use

1. The location of cars by the end of the current planning period T.
2. The cars sent on the days of the planned period whose arrival is expected in the future period.
3. The forecast car traffic for the future period.

Thus, the future pool of cars is given by the equation

$$\begin{aligned} \overline{y}_{sv} = \sum_{k\in K} y^T_{svk} + \sum_{\substack{s_1\in S,\, k\in K,\\ t:\, t+\theta_{s_1 s}>T}} x^t_{s_1 svk} + \sum_{\substack{(t-\tau_o,o)\in Q:\, s=s_2(o),\\ i\in[-\sigma_o,\sigma_o],\, t+i>T}} z^{t-\tau_o}_{ovi} \\ + \sum_{s_1\in S} \overline{x}_{s_1 sv} - \sum_{s_2\in S} \overline{x}_{ss_2 v} + + \sum_{\overline{o}\in\overline{O}:\, s=s_2(\overline{o})} \overline{z}_{\overline{o}v} - \sum_{\overline{o}\in\overline{O}:\, s=s_1(\overline{o})} \overline{z}_{\overline{o}vk}. \end{aligned} \tag{3}$$

The additional constraints $y^t_{svk} = 0$ and $\overline{y}_{svk} = 0$ are imposed for all $s \in S^0$ at stations where the storage of cars is prohibited.

The total number of loaded cars sent according to the schedule entry $(t, o) \in Q$ must not exceed the specified value χ^t_o,

$$\zeta^t_o + \sum_{v\in V,\, i\in[-\sigma_o,\sigma_o]} z^t_{ovi} \le \chi^t_o, \quad (t, o) \in Q. \tag{4}$$

In a similar way, we limit the execution of orders for the future period,

$$\sum_{v\in V} \overline{z}_{\overline{o}v} \le \overline{\chi}_{\overline{o}}, \quad \overline{o} \in \overline{O}. \tag{5}$$

System (2)–(5) sets the basic model constraints. There are a number of additional conditions that take into account the specifics of transportation. These conditions will not be considered in the present paper.

2.3. Objective Function

Let us denote the income of the transport operator for the current planning period by $\Phi(x, y, z)$. The value of $\Phi(x, y, z)$ depends on

1. The total revenue from execution of orders by own fleet of railcars,

$$G_1(z) = \sum_{\substack{(t,o)\in Q,\, v\in V,\\ i\in[-\sigma_o,\sigma_o]}} p_{ov} z^t_{ovi}.$$

2. The total revenue from execution of orders by short-term lease cars,

$$G_2(\zeta) = \sum_{(t,o)\in Q} (\min_{v\in V}\{p_{ov}\} - \nu_o)\zeta_o^t.$$

3. The total costs for no-load runs,

$$F_1(x) = \sum_{\substack{t\in T,\,(s_1,s_2)\in R,\\ v\in V,\,k\in K}} c_{s_1 s_2 vk} x^t_{s_1 s_2 vk}.$$

4. The total cost of storage of idle cars,

$$F_2(y) = \sum_{\substack{t\in T,\,s\in S,\\ v\in V,\,k\in K}} \lambda_s y^t_{svk}.$$

Consequently,

$$\Phi(x, y, z, \zeta) = G_1(z) + G_2(\zeta) - F_1(x) - F_2(y).$$

With the optimal management of the car fleet in the current period, it is necessary to take into account the potential future income. Let us introduce the following indicators:

1. The projected revenue from the execution of future orders,

$$\overline{G}_1(\overline{z}) = \sum_{\overline{o}\in\overline{O},\,v\in V} \overline{p}_{\overline{o}}\overline{z}_{ov}.$$

2. The projected costs for no-load runs in the future period,

$$\overline{F}_1(\overline{x}) = \sum_{(s_1,s_2)\in R,\,v\in V} \overline{c}_{s_1,s_2 v}\overline{x}_{s_1 s_2 v}.$$

3. The projected costs of storage of idle cars,

$$\overline{F}_2(\overline{x},\overline{y},\overline{z}) = \overline{T}\sum_{s\in S,\,v\in V}\lambda_s\overline{y}_{sv} - \overline{\lambda}\Bigg(\sum_{\overline{o}\in\overline{O},\,v\in V}\tau_{\overline{o}}\overline{z}_{\overline{o}v} + \sum_{(s_1,s_2)\in R,\,v\in V}\theta_{s_1s_2}\overline{x}_{rv}\Bigg),$$

where $\overline{c}_{s_1,s_2 v}$ are average (by types of transported goods) costs of no-load runs of a car of type v from station s_1 to station s_2, $\overline{T}$ is the number of days that make up the future period, and $\overline{\lambda}$ is the average cost of storing one car per day by stations. Then

$$\overline{\Phi}(\overline{x},\overline{y},\overline{z}) = \overline{G}_1(\overline{z}) - \overline{F}_1(\overline{x}) - \overline{F}_2(\overline{x},\overline{y},\overline{z})$$

can be viewed as an upper bound of income for the future period.

For the target indicator of the performance in managing the fleet of railway cars, we take the linear function

$$L(x, y, z, \zeta, \overline{x}, \overline{y}, \overline{z}) = \Phi(x, y, z) + \overline{\Phi}(\overline{x},\overline{y},\overline{z}), \tag{6}$$

whose value will be maximized on the set of feasible car flows given by the linear system (2)–(5).

3. SOFTWARE IMPLEMENTATION AND STRUCTURAL FEATURES

For software implementation of complex multi-index optimization problems, the software market offers a variety of high-level modeling languages. In the present paper, the criterion for choosing this language was that it must have the following features:

1. A flexible and easy-to-understand set of tools for describing the main parts of the model (variables, parameters, constraints, and objective function).
2. Convenient tools for filling the model with input data.
3. Availability of services for debugging the model.
4. Tools available for decision analysis.
5. The ability to export the problem to other standards for describing the model for additional analysis and research (as a rule, this is the MPS format for linear optimization problems).

To program the problem under consideration, a license was obtained to use the AMPL algebraic modeling language [6, 7]. AMPL is not only a powerful and convenient modeling language but also a standard supported by many modern industrial solvers (for example, CPLEX [2], Gurobi [1]), and also publicly available software, which, for example, can be found on the NEOS-Server optimization service [10].

Numerical methods of linear optimization operate with real-valued matrices and vectors into which the constraints and the objective function of the model are transformed. Owing to the huge dimension of real problems, the creation of such matrices in RAM can take a very substantial time and hence requires special attention.

The first AMPL implementations of the problem under consideration were an exact copy of its algebraic notation given above: first, parameters and variables were introduced, then the constraints (2)–(5) and the objective function (6) were described in their terms. This implementation of the model corresponds to the so-called row representation of its matrices—each row of the matrix is determined by the list of elements of its columns. At the time of testing the problem, data were available for the railway network consisting of 990 stations and 135 520 routes; the operator's resource included 29 923 cars of 31 different types; 1282 orders were received for execution for a month. The first difficulty that arose was that, even for a short-term planning horizon, the formation of the problem in RAM took too much time—approximately half an hour for a weekly management of the fleet of cars—and it rapidly grew with an increase in the planning horizon; this is unacceptable for industrial use.

Note that the constraint matrix (2)–(5) has a high degree of sparseness. Thus, for example, the variable y^t_{svk} occurs only in two balance equations (2) corresponding to the indices (t, s, v, k) and $(t+1, s, v, k)$. The variable $x^t_{s_1 s_2 vk}$ occurs in Eq. (2) with index (t, s_1, v, k). Further, if $t + \theta_{s_1 s_2} \in T$, then $x^t_{s_1 s_2 vk}$ occurs in the equation with index $(t + \theta_{s_1 s_2}, s_2, v, k)$; otherwise, in the balances of the future period (3) with index (s_2, v). The variable z^t_{ovi} occurs in Eq. (2) with index $(t + i, s_1(o), v, k(o))$. Further, if $t + i + \tau_o \in T$, then z^t_{ovi} occurs in the equation with index $(t + i + \tau_o, s_2(o), v, k(o))$; otherwise, in the balance sheets of the future period (3) with index $(s_2(o), v)$. Such a peculiarity of the problem allows presenting the matrix of its constraints in the so-called column form [11]: first, the constraints and the objective function of the model are declared, and then variables are introduced for which the inclusion constraints and the corresponding coefficients are indicated. This modeling method imitates the description of sparse matrices using columns and is natural for simplex-like algorithms.

AMPL provides means to describe the model in both row and column form. For comparison, a column representation of problem (2)–(5), (6) was implemented and an analysis was made of the data formation time in RAM for both recording forms. The results are given in Table 1. We see that the problem generation time for the weekly planning horizon with the column implementation is 14 times shorter than that with its row notation. Moreover, this difference increases with an increase in the dimension of the problem. Taking into account such a significant difference in the machine time, the column implementation of the model was adopted as a working one.

Concerning the dimension of the problem and consumed resources, the following can be noted. For the current planning period of 15 days and the future forecast for orders over the next 14 days, we obtained a linear optimization problem with 68 374 100 variables and 499 996 constraints. When exporting the problem to the MPS format, a file was created that occupied approximately 5.8 GB of disk space, and the file generation took approximately 7 min. The PC configuration for testing was as follows: Intel Xeon Gold 6136 processor with a frequency of 3 GHz, 96 GB of RAM. The problem was solved with the CPLEX package (using up to 8 threads). Connecting the problem preprocessing

Table 1. AMPL model formation time in RAM

$\|T\|$	Time, s, row record	Time, s, column record	Number of variables	Number of constraints	Number of nonzero elements
1	43.82	35.88	7 911 511	63 254	15 844 030
2	633.03	52.46	11 883 012	94 455	23 827 873
3	877.92	68.5	15 860 277	125 775	31 835 993
4	1130.81	86.93	19 831 744	156 933	39 858 570
5	1407.22	102.56	23 801 582	188 144	47 958 557
6	1682.08	120.39	27 771 335	219 335	56 147 993
7	1937.29	138.67	31 741 619	250 541	64 508 807

option (presolver) made it possible to reduce the problem to a smaller dimension: 20 819 229 variables, 188 338 constraints, and 43 762 028 nonzero elements. The total time for the formation and solution of the problem was approximately 4 h.

4. MATHEMATICAL ASPECTS OF SOLVING LARGE-DIMENSION TRANSPORT AND LOGISTICS PROBLEMS

As one can see in the general description of the model in Sec. 1, its main part consists of multiproduct multi-index interperiod balances of the form

$$\begin{aligned} \sum_{J:\,(I,J)\in W} A^e_{I,J} x_{I,J} = b^e_I, \quad I \in S_e, \\ \sum_{I:\,(I,J)\in W} B^e_{I,J} x_{I,J} = b^e_J, \quad J \in D_e, \end{aligned} \tag{7}$$

where $\{I, J\}$ are composite indices selected from a subset W of the Cartesian product of product sets $\{P\}$, vehicle types $\{V\}$, time periods $\{T\}$, departure and reception stations $\{S\}$, and maybe something else. The variables $x_{I,J}$, $I, J \in W$, are the planned volumes of cars intended for the performance of prescribed transportation carried out from the sender, coded by index I, to the consumer, coded by index J.

These relations vary from one considered problem to another and often depend on decisions taken in manual mode, the state of the transport infrastructure, and other factors. Some can be described as additional inequality constraints of the form

$$\begin{aligned} \sum_{J:\,(I,J)\in W} A^l_{I,J} x_{I,J} \le b^l_I, \quad I \in S_l, \\ \sum_{I:\,(I,J)\in W} B^l_{I,J} x_{I,J} \le b^l_J, \quad I \in D_l, \end{aligned} \tag{8}$$

and practice shows that $|S^e \cup D_e| \ggg |S^l \cup D_l|$. We should also note the important role of the nonnegativity conditions for $x_{I,J} \ge 0$, which at first glance look deceptively simple.

4.1. A Projection Approach to the Linear Optimization Problem

If we keep in mind the planned development of the constructed model, then it is worth considering some new approaches to solving high-dimensional linear optimization problems that can be configured for application to problem (2)–(5), (6). Such development is motivated by a considerable growth and complication of the model with an increase in the planning horizon, a more detailed description of the transported goods, types of cars, transportation technologies and other factors;

this removes the problem from the field of problems solved by the standard optimization apparatus. In this regard, the idea arose of using projection algorithms, which have proven themselves quite well in other areas but have not yet been seriously used to solve linear optimization problems.

To simplify the notation, we define and denote an orthogonal projection operator by

$$p \downarrow X = \operatorname{argmin} \min_{z \in X} \|p - x\|, \tag{9}$$

i.e., $\min_{x \in X} \|p - x\| = \|p - p \downarrow X\|$, where $\|\cdot\|$ stands for the standard Euclidean norm. Taking this opportunity, by $|N|$ we also denote the number of elements in a finite set N of some objects. By Cone (D) we denote the cone shell of the finite set $D = \{z^1, z^2, \dots, z^m\}$ of vectors of some vector space.

It was shown in [12] that under not too restrictive assumptions, a solvable linear optimization problem (1) can be solved by a single operation of projection onto its admissible set X. To state one of the versions of such assumptions, we recall the standard definitions of admissible and dual cones for an admissible set of a linear optimization problem.

Definition 1. For $x \in X$, in (1) we define the admissible cone as

$$K(x) = \big\{z \mid x + \lambda z \in X \text{ for some } \lambda > 0\big\}. \tag{10}$$

It is essential that for linear optimization problems $K(x)$ is a convex closed set if it is nonempty.

Definition 2. For $x \in X$, in (1) we define the dual cone as

$$K^\star(x) = \big\{z \mid xz \le 0 \text{ for all } x \in K(x)\big\}. \tag{11}$$

If $x^\star$ is a solution of problem (1), then the dual cone $K^\star(x^\star)$ will be denoted by $K^\star$.

The simplest assumption guaranteeing that the solution of problem (1) can be obtained by a single projective operation is that the interior of $K^\star$ is not empty. In the first place, it follows from this assumption in an elementary way that the optimal solution is unique in this case. The converse is also true, but this requires some reasoning. The problem is that the dual factors may not be unique in this case. The following lemma solves this problem.

Lemma 1. *If a solution $x^\star_J$ of problem* (1) *is unique, then $K^\star$ has a nonempty interior.*

To prove this, consider problem (1) and its solution $x^\star_J$ in more detail. By S_e we denote a subset of the constraint set S such that

$$\sum_{J:\,(I,J)\in W} A_{I,J} x^\star_J = b_I, \quad I \in S_e \subset S. \tag{12}$$

It follows from the uniqueness of the optimal solution that $|S_e| \ge |J|$. Otherwise, we can move from the point $x^\star$ into a nontrivial orthogonal subspace of the linear span A_I, $I \in S_e$, where, for brevity, A_I denotes the Ith row of the matrix A, while keeping the equality constraints and optimality; this contradicts uniqueness. Note that also, according to the remarkable result in [13], there exists a strictly complementary solution of the linear optimization problem (1) with constraints (7), (8) with strictly positive dual factors u_I, $I \in S_e$ such that

$$c = \sum_{I \in S_e} u_I A_I. \tag{13}$$

Further, we will show that there exists an $S'_e \subset S_e$ such that the A_I, $I \in S'_e$, are linearly independent, $|S'_e| = n$, and there exist strictly positive u_I, $I \in S'_e$ (dual factors) such that

$$c = \sum_{I \in S'_e} u_I A_I, \tag{14}$$

which implies their dual optimality. Such a set S'_e can be constructed using the successive elimination process described below.

Indeed, if $|S_e| > n$, then $A_I,\ I \in S_e$, are linearly dependent, and there exist nonzero $\mu_I,\ I \in S_e$ such that

$$0_J = \sum_{I \in S_e} \mu_I A_I, \tag{15}$$

where 0_J is the null vector of appropriate dimension; consequently,

$$c = \sum_{I \in S_e} (u_I + t\mu_I) A_I \tag{16}$$

for each t.

In this case, we can choose a t such that $u_{I'} + t\mu_{I'}$, $I' \in S_e$, vanishes for at least one I' and t' and the remaining $u_I + t\mu_I$, $I \neq I'$, $I \in S_e$, remain nonnegative. As a result, we can exclude I' from the set S_e (as well as other $I'' \in S_e$ such that $u_{I''} + t\mu_{I''} = 0$ by coincidence), redefine u_I and S_e, and continue until we obtain $|S_e| = n$ with linear independence of A_I, $I \in S_e$.

In turn, this guarantees that for each c'_J, $J \in J$, close enough to c, in any metric there exist $u'_I > 0$, $I \in S_e$, such that $c' = \sum_{I \in S_e} u'_I A_I$; hence $c' \in K^\star$, and so $K^\star$ has a nonempty interior.

As shown in [12], the solution of the projective problem of finding $(x^0 - \tau c) \downarrow X$, where X denotes a feasible set (1), yields a solution of (1) for arbitrary x^0 and sufficiently large $\tau > 0$, depending, of course, on x^0. In other words, $(x^0 - \tau c) \downarrow X = x^\star$.

4.2. Implementation and Numerical Experiments

Projection algorithms, especially their alternating variants, are very popular for solving convex feasibility problems. The development of these algorithms began in 1940s [14]; detailed bibliography can be found in [15]. Their development continues till now (see, say, [16] for a survey of the state-of-the-art in this field).

The problem of finding the minimum distance to a convex set (9) has received much less attention, except perhaps for the most famous case [14], where the set X can be represented as an algebraic sum of two sets $X = X_1 + X_2$ and alternating projections onto X_1 and X_2 are easily realized. Another case where finite convergence has been proved is when X is an (almost) canonical simplex [17, 18].

Unfortunately, we are dealing with a polyhedron X of almost general form, but it is possible to apply a number of transformations that will help to considerably reduce the dimension of auxiliary problems and make it possible to solve practical problems.

Transformation of the projection problem. The first step is to reduce the problem $p \downarrow X$ to the problem of finding the minimum norm element. Since $p \downarrow A = (p - a) \downarrow (A - a) + a$ for any a, we have

$$(x^0 - \tau c) \downarrow X = 0 \downarrow (X - x^0 + \tau p) + x^0 - \tau c$$

and the main part of the problem on the right is the problem of finding the minimum norm vector

$$0 \downarrow (X - x^0 + \tau p) = 0 \downarrow Q \tag{17}$$

for $Q = X - x^0 + \tau p$, the displaced polyhedron X. This does not change the type of constraints that describe X, and so $Q = Q_e \cap Q_l$, where

$$Q_e = \left\{ x \,\middle|\, \sum_{J\colon (I,J) \in W} A^e_{IJ} x_J = b'^e_I,\ I \in S_e \right\}, \tag{18}$$

$$Q_l = \left\{ x \,\middle|\, \sum_{J\colon (I,J) \in W} A^l_{IJ} x_J = b'^l_I,\ I \in S_l \right\}, \tag{19}$$

which differ from (7) and (8) only in the right-hand sides b'^e_I, $I \in S_e$, and b'^l_I, $I \in S_l$.

The next step is to replace the minimum norm problem $0 \downarrow Q$ with the projection of the special point p^0 onto the polyhedral cone generated by the rows of the constraint matrix (18), (19),

$$\min_{z \in \operatorname{Cone}(\overline{Q}),} \|z - p^0\|^2 = \|p^0 \downarrow \overline{Q} - p^0\|^2, \tag{20}$$

where $\overline{Q} = \operatorname{Cone}(\overline{Q}_e) + \operatorname{Cone}(\overline{Q}_l)$. The matrices $\operatorname{Cone}(\overline{Q}_e)$ and $\operatorname{Cone}(\overline{Q}_l)$ are the matrices Q_e and Q_l with extra columns $-b_I'^e$, $I \in S_e$, and $-b_I'^l$, $I \in S_l$,

$$\begin{aligned} \overline{Q}_e &= \left\| Q_e \quad -b'^e \right\|, \\ \overline{Q}_l &= \left\| Q_l \quad -b'^l \right\|. \end{aligned} \tag{21}$$

The details of this transformation are described in [19]; here we consider only the decomposition possibilities provided by (20) for the projection onto a polyhedral cone.

Decomposition of high-dimensional linear optimization problems. The first idea for simplifying the design problem (20) using decomposition is based on the fact that the cone $\operatorname{Cone}(\overline{Q}_e) = L_e$ is a linear subspace and the projection onto it is a series of standard linear algebra operations for which there are efficient implementations [4] that take into account the sparseness of data traditional for linear optimization.

Then, as was shown in [20], problem (20) can be reduced to the projection operation $p_l^0 \downarrow \overline{Q}_{l,e}$, where

$$p_e^0 = p^0 \downarrow \overline{Q}_e, \quad \overline{Q}_{l,e} = \overline{Q}_l \downarrow \overline{Q}_e = Q_l \downarrow L_e,$$
$$\overline{Q}_l \downarrow \overline{Q}_e = \left\{ z = q \downarrow \overline{Q}_e \mid q \in \overline{Q}_l \right\} = \overline{Q}_l \downarrow L_e.$$

Thus, in essence, the equality constraints that form a majority in transport and logistics problems are eliminated, and the size of the problem is considerably reduced. Of course, this requires additional computational cost to project all $q \in \overline{Q}_l$ onto L_e, but since each such vector q can be represented as a nonnegative combination of the generators $\overline{Q}_l$,

$$\overline{Q}_l \downarrow L_e = \left(\sum_I \mu_I \overline{Q}_{l,I} \right) \downarrow L_e = \sum_I \mu_I \left(\overline{Q}_{l,I} \downarrow L_e \right) = \sum_I \mu_I z_I,$$

this reduces to projecting all rows of $\overline{Q}_l$ onto L_e.

Of course, this is still quite a computationally expensive operation, but its cost can be substantially reduced if $\overline{Q}_l$ has the right structure.

To identify features of such a structure appropriate for speeding up calculations, we can use some approaches based on the graph theory.

4.3. Structural Analysis of Linear Optimization Problems. Practical Applications

To consider a linear optimization problem from the point of view of the graph theory, it is useful first to consider the adjacency matrix of constraints, i.e., consider rows of constraint matrices as vertices of some graph. Two row vertices can be considered connected by an edge if these two rows simultaneously have nonzero coefficients in some column. Such a connection graph of constraint vertices can be divided into several nonintersecting connected components, it follows that the operation of projection onto such a matrix is greatly simplified. Undoubtedly, the effect of such a decomposition depends on the nature of the sparseness of matrices. Real problems are too large for theoretical analysis, but the problem of finding connected components is polynomially solvable and hence can be applied efficiently even for large matrices. Next, we present some examples of such an analysis for transport logistics problems.

Table 2. Characteristics of test problems

N	Number of rows	Number of columns	Number of nonzero coefficients	Density, %	Number of components
1	28 935	5 794 061	–	–	–
2	39 932	51 090	99 608	4.88×10^{-3}	535
3	122 347	16 675 027	31 192 305	1.53×10^{-3}	32
4	361 024	53 404 781	106 608 366	5.42×10^{-4}	233

Table 3

Size of component	Number of such components	Average share of nonzero elements
25 592	1	3.907×10^{-5}
1743	1	5.737×10^{-4}
1596	1	6.266×10^{-4}
1162	1	8.606×10^{-4}
872	1	1.147×10^{-3}
727	2	1.376×10^{-3}
726	1	1.377×10^{-3}
581	2	0.721
437	2	2.288×10^{-3}
436	4	2.294×10^{-3}
291	7	3.436×10^{-3}
1	970	1.000

For example, we considered four transport and logistics problems that differ in the planning horizon, the number of vehicles, routes, etc. The characteristics of these models are presented in Table 2. The last column gives the number of connected components. On the one hand, it is quite large, and this allows us to hope that the dimensions of the components will be considerably smaller than the dimensions of the entire problem. On the other hand, the number of connected components is not too large; this makes it possible to use standard parallel computing tools.

The details of the distribution of the number of connected components as a function of their size are given in Table 3. The data obtained show a large variability in the number of connected components, although this variability is mainly due to a large number of small components. The number of relatively large components in all experiments is approximately 10% of the total number of constraints; this allows us to hope for a positive effect from coarse-grained decomposition.

CONCLUSIONS

The use of increasingly large-scale mathematical models that are already beyond the capabilities of standard algorithms and programs is expanding in the practice of managing large companies. Accordingly, there is a need for qualified use of tools for describing such models and algorithms that can take into account the specific features of the structure of the problem and offer new approaches to its decomposition. In this paper, we propose efficient methods for employing the widely used AMPL modeling language and study an algorithmic approach to solving semistructured high-dimensional optimization problems based on the use of a projection operator. As shown in the paper, the efficient use of the AMPL language can substantially speed up the preparation of the model for calculation, and the decomposition of projective operators creates new opportunities for parallelism.

FUNDING

This research was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. 075–02–2022–880.

REFERENCES

1. *Gurobi Optimizer Reference Manual* (Gurobi Optimization, Beaverton, 2021). www.gurobi.com/documentation/9.5/refman/index.html. Accessed February 27, 2022.
2. *IBM ILOG CPLEX Optimizer* (IBM, Armonk, 2022). www.ibm.com/analytics/cplex-optimizer. Accessed February 27, 2022.
3. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide* (SIAM, Philadelphia, 1999).
4. *SuiteSparse: A Suite of Sparse Matrix Software* (College Station, Texas A&M Univ., 2022). people.engr.tamu.edu/davis/suitesparse.html. Accessed February 27, 2022.
5. G. B. Dantzig and P. Wolfe, "The decomposition algorithm for linear programming," Econometrica **9** (4), 767–778 (1961).
6. *AMPL Homepage* (AMPL Optim., Mountain View, CA, 2022). ampl.com. Accessed February 27, 2022.
7. R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming* (Cengage Learning, Boston, 2003).
8. *GAMS—A User's Guide* (GAMS Software, Frechen, 2022). www.gams.com/35/docs/UG_MAIN.html. Accessed February 27, 2022.
9. *GLPK (GNU Linear Programming Kit)* (Free Software Found., Boston, MA, 2012). www.gnu.org/software/glpk/. Accessed February 27, 2022.
10. *NEOS Server* (Univ. Wisconsin, Madison, WI, 2022). neos-server.org/neos/. Accessed February 27, 2022.
11. G. Kahan, "Walking through a columnar approach to linear programming of a business," Interfaces **2** (3), 32–39 (1982).
12. E. A. Nurminski, "Single-projection procedure for linear optimization," J. Glob. Optim. **66** (1), 95–110 (2016).
13. A. J. Goldman and A. W. Tucker, *Theory of Linear Programming, Linear Inequalities and Related Systems. Vol. 38* (Princeton Univ. Press, Princeton, NJ, 1956), pp. 53–97.
14. J. von Neumann, "On rings of operators. Reduction theory," Ann. Math. **50** (2), 401–485 (1949).
15. H. H. Bauschke and J. M. Borwein, "On the convergence of von Neumann's alternating projection algorithm for two sets," Set-Valued Anal. (1), 185–212 (1993).
16. R. Escalante and M. Raydan, *Alternating Projection Methods* (SIAM, Philadelphia, 2011).
17. C. Michelot, "A finite algorithm for finding the projection of a point onto the canonical simplex of E^n," J. Optim. Theory Appl. **50** (1), 195–200 (1986).
18. V. N. Malozemov and G. Sh. Tamasyan, "Two fast algorithms for projecting a point onto the canonical simplex," Zh. Vychisl. Mat. Mat. Fiz. **56** (5), 742–755 (2016) [Comput. Math. Math. Phys. **56** (5), 730–743 (2016)].
19. E. A. Nurminski, "Projection onto polyhedra in outer representation," Zh. Vychisl. Mat. Mat. Fiz. **48** (3), 387–396 (2008) [Comput. Math. Math. Phys. **48** (3), 367–375 (2008)].
20. E. A. Nurminski and N. B. Shamray, "Row-oriented decomposition in large-scale linear optimization," in *Optimization and Applications. Proc. Int. Conf. OPTIMA 2021* (Petrovac, Montenegro, September 27–October 1, 2021), vol. 13078 of *Lect. Notes Comput. Sci.* (Springer, Heidelberg, 2021), pp. 50–63.
21. H. J. Greenberg, J. R. Lundgren, and J. S. Maybee, "Graph theoretic methods for the qualitative analysis of rectangular matrices," SIAM J. Algebraic Discrete Methods **2** (3), 227–239 (1981).