



FIELD OF SCIENCE ENGINEERING AND TECHNICAL SCIENCES
SCIENTIFIC DISCIPLINE TECHNICAL INFORMATION TECHNOLOGY
AND TELECOMMUNICATIONS

DOCTORAL THESIS

Algorithmic information complex based on
variational inequalities for optimization problems in
networks.

Author: Serhii Denysov

First supervisor: Prof. Janusz Kacprzyk, PhD, DSc
Assisting supervisor: Prof. Artur Zaporozhets, PhD, DSc

Completed in: Systems Research Institute Polish Academy of Sciences

Warsaw, 2025

Abstract

With the rapid growth of problems tackled by information technology, the need for efficient algorithms keeps increasing. Choosing an effective algorithm for a specific problem (or a class of problems) usually requires both theoretical and numerical study, which, in turn, requires a corresponding toolset to assist the research process.

One important class of algorithms with wide application area are algorithms for solving variational inequalities, as it models many tasks in operations research, economics, traffic flow, game theory, and some problems of modern machine learning. Because these problems are large and complex, algorithmic efficiency is critical. To solve a problem with algorithms from the class we require both theory – applicability and convergence analysis under realistic assumptions – and computation-systematic experiments that reveal practical speed and robustness.

This dissertation provides numerical and theoretical research on iterative algorithms for variational inequalities together with a software framework, which allows solving corresponding class of problems and investigating behavior of algorithms.

Theoretical part devoted to obtaining new or improved algorithms for specific classes of problems and proving their convergence under widest possible assumptions.

The software framework simplifies experiments with new and modified algorithms and problems, because to add a new method or a problem a user can extend provided base classes and make only minimal changes. Also, for the implemented problems and algorithms it is possible to make different runs, set parameters from command line or code and obtain detailed execution log together with convergence graphs.

Most examples of practical applications are based on the problems of the so-called network economy, when some kind of optimization or equilibrium problem is considered on the network or graph. For the blood supply chain model a graphical UI is provided (web), which allows setting parameters for nodes and edges, selecting algorithms subset and obtaining the solution together with detailed execution logs and convergence graphs without modifying the python code.

The dissertation has classic structure. The first chapter is devoted to the literature review about modeling practical problems with variational inequalities and developing corresponding algorithms.

The second chapter describes a new variant of extragradient algorithm with monotone step adjustment together with proof of convergence under maximally wide assumptions.

The third chapter is devoted to a new version of extrapolation from the past algorithm, leveraging the idea of special distance (Bregman divergence) to simplify computations for a special class of problems. Corresponding proof of convergence also provided.

The fourth chapter is about constructing the analogue of the extrapolation from the past algorithm without requiring the problem space being Hilbert space, with corresponding convergence proof.

The fifth (the last) chapter is devoted to describing the software framework architecture and usage scenarios, together with demonstration of applicability of the described algorithms to the problem of blood supply chain network optimization, problem of traffic equilibrium search and PageRank problem. This chapter also demonstrates the full cycle from formulating a model for a problem and rewriting it as a variational inequality, to obtaining a solution and analyzing the behavior of algorithms on the selected problem using capabilities of the developed software system.

Streszczenie

Wraz z szybkim wzrostem liczby problemów rozwiązywanych przez technologie informatyczne, stale rośnie zapotrzebowanie na wydajne algorytmy. Wybór skutecznego algorytmu dla konkretnego problemu (lub klasy problemów) wymaga zazwyczaj zarówno badań teoretycznych, jak i numerycznych, co z kolei wymaga odpowiedniego zaplecza narzędziowego wspomagającego proces badawczy.

Ważną klasą algorytmów o szerokim zastosowaniu są metody rozwiązywania nierówności wariacyjnych, modelują one bowiem wiele zadań z zakresu badań operacyjnych, ekonomii, przepływów w sieciach transportowych, teorii gier oraz niektórych zagadnień współczesnego uczenia maszynowego. Ze względu na dużą skalę i złożoność tych problemów, kluczowa jest efektywność obliczeniowa. Rozwiązanie problemu za pomocą algorytmów z tej klasy wymaga zarówno podbudowy teoretycznej – analizy stosowności i zbieżności przy realistycznych założeniach – jak i systematycznych eksperymentów obliczeniowych, ujawniających praktyczną szybkość i odporność metod.

Niniejsza rozprawa przedstawia badania numeryczne i teoretyczne nad iteracyjnymi algorytmami dla nierówności wariacyjnych wraz z autorskim szkieletem programowym (frameworkiem), który umożliwia rozwiązywanie omawianej klasy problemów oraz badanie zachowania algorytmów.

Część teoretyczna poświęcona jest opracowaniu nowych lub ulepszonych algorytmów dla określonych klas problemów oraz udowodnieniu ich zbieżności przy możliwie najszerszych założeniach.

Opracowany szkielek programowy upraszcza prowadzenie eksperymentów z nowymi i zmodyfikowanymi algorytmami oraz problemami, ponieważ dodanie nowej metody lub zadania wymaga od użytkownika jedynie rozszerzenia dostarczonych klas bazowych i wprowadzenia minimalnych zmian. Ponadto, dla zaimplementowanych problemów i algorytmów możliwe jest wielokrotne uruchamianie symulacji, ustawianie parametrów z poziomu wiersza poleceń lub kodu oraz uzyskanie szczegółowego logu wykonania wraz z wykresami zbieżności.

Większość przykładów praktycznych zastosowań opiera się na problemach tzw. ekonomii sieciowej, gdzie rozważany jest pewien rodzaj problemu optymalizacyjnego lub problemu równowagi w sieci (grafie). Dla modelu łańcucha dostaw krwi udostępniono graficzny interfejs

użytkownika (webowy), który pozwala na ustawianie parametrów dla węzłów i krawędzi, wybór podzbioru algorytmów oraz uzyskanie rozwiązania wraz ze szczegółowymi logami i wykresami zbieżności bez konieczności modyfikacji kodu w języku Python.

Rozprawa ma strukturę klasyczną. Rozdział pierwszy poświęcony jest przeglądowi literatury dotyczącej modelowania problemów praktycznych za pomocą nierówności wariacyjnych oraz opracowywania odpowiednich algorytmów.

Rozdział drugi opisuje nowy wariant metody ekstragradientowej z monotoniczną regulacją kroku wraz z dowodem zbieżności przy maksymalnie szerokich założeniach.

Rozdział trzeci poświęcony jest nowej wersji algorytmu ekstrapolacji z przeszłości (extrapolation from the past), wykorzystującej ideę specjalnej odległości (dywergencja Bregmana) w celu uproszczenia obliczeń dla specjalnej klasy problemów. Przedstawiono również odpowiedni dowód zbieżności.

Rozdział czwarty dotyczy konstrukcji analogu algorytmu ekstrapolacji z przeszłości bez wymogu, aby przestrzeń problemu była przestrzenią Hilberta (w ogólnych przestrzeniach Banacha), wraz z odpowiednim dowodem zbieżności.

Rozdział piąty (ostatni) poświęcony jest opisowi architektury szkieletu programowego i scenariuszy jego użycia, wraz z demonstracją stosowalności opisanych algorytmów do problemu optymalizacji sieci łańcucha dostaw krwi, problemu równowagi ruchu w sieciach transportowych oraz problemu PageRank. Rozdział ten demonstrowe również pełny cykl badawczy: od sformułowania modelu problemu i przepisania go w formie nierówności wariacyjnej, aż do uzyskania rozwiązania i analizy zachowania algorytmów na wybranym problemie z wykorzystaniem możliwości opracowanego systemu.

Contents

1	State of problem and literature review	1
1.1	Introduction	1
1.2	Models Based on Variational Inequalities	3
1.3	Network Economics Problems	4
1.3.1	Market Equilibrium	5
1.3.2	Optimal Traffic Flow Distribution	5
1.3.3	Supply Chains	6
1.3.4	Determining the Importance of Network Nodes	7
1.4	Constructing Efficient Algorithms for Solving Problems Modeled by Variational Inequalities	7
1.4.1	Economic Algorithms	8
1.4.2	Adaptive Algorithms	10
1.4.3	Algorithms with Non-Euclidean Distance	11
1.4.4	Algorithms in Hadamard Spaces	13
1.5	Current State of the Field	13
	Contribution of this Thesis	16
2	Algorithm with Monotone Step Size Adjustment for Variational Inequalities with Additional Conditions	19
2.1	Necessary Theory and Problem Statement	20
2.2	Extragradient Algorithm with Monotone Step-Size Adjustment	22
2.3	Convergence proof	23
2.4	Algorithm Variant for an Operator Equation with A Priori Information	27
2.5	Summary of Chapter 2	27
3	Algorithms with Accelerated Projection Using Bregman Divergence	29
3.1	Problem Statement and Preliminaries	29
3.2	Algorithm	32
3.3	Proof of Convergence	33
3.4	Adaptive Variant of the Extrapolation from the Past Algorithm	36

3.5	Summary of Chapter 3	41
4	Extrapolation from the Past Method for Variational Inequalities in Uniformly Convex Banach Spaces	42
4.1	Preliminaries and Problem Statement	42
4.2	Extrapolation from the Past Algorithm	46
4.3	Convergence Proof	46
4.4	Adaptive Extrapolation from the Past Algorithm	50
4.5	Summary of Chapter 4	53
5	Software framework and applications of the algorithms	54
5.1	Software framework for applying and investigating extragradient-type algorithms .	54
5.2	Problems of Network Economics and Variational Inequalities	56
5.3	Optimization of the Blood Collection and Delivery Chain	57
5.3.1	Problem Statement	57
5.3.2	Formulation of the Mathematical Model of the Blood Supply Chain	59
5.3.3	Optimization problem and variational inequality	61
5.3.4	Application of Algorithms to the Blood Supply Chain Problem	62
5.3.5	Computational results	62
5.4	The Web Page Ranking Problem — PageRank	64
5.4.1	Mathematical Model	64
5.4.2	Formulation as a Variational Inequality	65
5.4.3	Model Construction Based on the Graph	67
5.4.4	Numerical Experiments	68
5.5	Traffic Equilibrium Problem	72
5.5.1	Mathematical Model	73
5.5.2	Numerical Experiments	74
5.6	Conclusions to Chapter 5	76
	Concluding Remarks	78
	References	82

Chapter 1

State of problem and literature review

1.1 Introduction

Computational power available to humanity has grown dramatically in the past 30 years, and some real-world problems, which looked unthinkable to solve using computers and algorithms some time ago, now solved easily even with smartphones. But despite the fact that our computing capacity has grown and this growth continues with a very fast pace, we still struggle with it being insufficient or too expensive for our tasks. Also, as people realised not too long ago, solving huge problems with current approaches to computation can be very environmentally unfriendly.

One of the main reasons for this situation is the fact that people start to use information technologies in many new areas, and now we have much more information to process and much more problems to solve using computers and algorithms. The example is on everyone's lips - nobody thought about training and using large language models of such scale ten years ago for the mass market - and now companies are competing for electricity to power enormous datacenters for that.

But the extensive approach, like building more datacenters and power plants to supply required electrical power, looks really dangerous, and the case of fall of stocks of some AI companies after DeepSeek release shows that it is clear for many market participants in high-tech industry. Obviously, a successful company can invest billions of euros to build a datacenter which is ten times more powerful than the one of their competitors. But it takes years to accomplish - and if their competitors improve their algorithms by the same tenfold in that time, they will be able to gain an advantage much faster. Even if the algorithm is only half as computationally intensive, the resource savings for large tasks can be enormous. Imagine that instead of two power plants it is enough to build one, because with a better algorithm it is possible to solve the same problem twice faster. And if the algorithm is asymptotically faster, the outcome can be even more impressive.

Consequently, the tasks of designing efficient algorithms and investigating their properties are highly relevant, and their importance continues to grow – together with another important task of choosing suitable algorithms for solving specific kind of real-world problems.

When developing and analyzing algorithms, it is essential to combine theoretical studies (such as proving and analyzing convergence conditions) with practical investigations in the form of computational experiments. Such experiments make it possible to validate theoretical results and to generate new ideas for improving the behavior of the methods. They can also reveal difficulties for certain classes of problems or parameter ranges that then require theoretical investigation and, where possible, improvement.

To support this research process efficiently, this work develops a software framework that provides convenient tools both for studying algorithm behavior on test problems and for computing solutions to real problems, provided their mathematical models fit within the variational inequality framework.

A broad class of problems can be described within the framework of, so called, network economics. This class includes diverse economic problems, for example supply network optimization in various settings and market/price equilibrium models. Similar formulations also arise for certain social processes, such as migration and information diffusion in social networks. Other important problems in network economics include finding equilibrium in transportation networks and modeling the delivery of critical supplies by humanitarian organizations during natural or man-made disasters.

These models have been actively studied by the research group at the University of Massachusetts Amherst (USA)—notably the work of A. Nagurney, A. Masoumi, Min Yu, J. Dong, and others. Most network economics problems naturally reduce to a variational inequality and can therefore be addressed by corresponding algorithms.

In this dissertation, we provide detailed study for a model for the blood supply chain optimization problem. For this problem our software includes web UI, which allows to set network parameters, get a solution and analyze the behavior of selected algorithms. Also we study models and demonstrate application of algorithms for transport equilibrium search problem and so called PageRank problem (problem of detecting the most important nodes in a network based on the network structure).

Variational inequality is a powerful mathematical model, which can be used to formulate problems of operations research, economics, network traffic optimization and game theory. It is also used in machine learning area, as it is a straightforward model for GAN (generative adversarial networks). Owing to this modeling power, efficient algorithms for solving variational inequalities and their practical applications are of interest for both theoretical and computational analysis.

The development and study of such algorithms has proceeded, with varying intensity, since the 1970s – beginning with the works of Korpelevich and Antipin – and has been especially active over the past thirty years. Significant contributions have been made by I. Konnov, I. Ryazantseva, Y. Censor, A. Gibali, A. Iusem, S. Reich, J.-J. Strodiot, B. Svaiter, P. Tseng, W. Takahashi, and H. K. Xu. This line of research intersects with the broader study of optimization algorithms, where

intensive theoretical work has also been carried out; key contributors include J.-L. Lions, P.-L. Lions, Y. Alber, L. Bregman, A. Nemirovskii, P. L. Combettes, H. Bauschke, D. Bertsekas, P.-E. Maingé, M. Krasnosel'skii, Y. Nesterov, E. Nurminskii, B. Polyak, B. Pshenichnyi, M. Solodov, A. Antipin, A. Bakushinsky, F. Vasiliev, Yu. Danilin, Y. Ermoliev, I. Yeremin, P. Stetsyuk, N. Shor, and many others.

In this dissertation, we continue research on abovementioned family of algorithms, proposing several new ones, proving corresponding convergency theorems, conducting numerical experiments and demonstrating application of the algorithms to several problems.

1.2 Models Based on Variational Inequalities

Like many fundamental models widely applied in contemporary mathematical research and engineering, the concept of a variational inequality emerged quite early – back in the 1960s. The theory originated with works by Fichera, Stampacchia, and Lions [90, 40, 93, 61], where this approach was used to investigate and solve unilateral boundary-value problems for partial differential equations. Variational and optimization formulations for such problems were also examined, notably in monographs [54, 7]. The modeling approach based on variational inequalities evolved with varying intensity over subsequent decades, and currently, the range of problems it addresses is extensive.

Naturally, this approach continues to be used for problems in mathematical physics, such as elasticity theory, fluid behavior models, impurity diffusion, and others [38, 16, 62, 92, 60], as well as optimal control problems for systems of differential equations arising in modeling such processes [63]. Recently, models traditionally used for physical processes, with minor modifications, have increasingly been applied to social or economic processes, generating renewed research interest. For example, the diffusion model effectively describes information spread in social networks [106] or disease transmission during epidemics [107], allowing to tackle various identification and optimal control problems for such processes.

In classical finite-dimensional space $\mathbb{R}^n$, the problem of finding a solution to a variational inequality is formulated as follows [54]: for a closed convex set $K \subset \mathbb{R}^n$, find

$$x \in K : (F(x), y - x) \geq 0 \text{ for all } y \in K.$$

An intuitively clear example of this formulation is the concise condition for the necessary and sufficient minimum of a continuously differentiable convex function f on the interval $[a, b] \subset \mathbb{R}$. Indeed, finding such a minimum is equivalent to solving the following "variational inequality": find $x_0 \in [a, b] : f(x_0)(x - x_0) \geq 0 \forall x \in [a, b]$.

In a Hilbert space H , the traditional formulation of the problem is: for a closed convex set $C \subset H$ and an operator $A : H \rightarrow H$, find

$$x \in C : (Ax, y - x) \geq 0 \quad \forall y \in C. \quad (1.1)$$

Further specifics for particular models depend on the choice and properties of the operator A and the set C . Similar problem formulation will be used for Banach spaces later on.

A significant motivation for creating a whole research direction within variational inequalities theory was the realization that they suits naturally for the equilibrium programming [13, 50], encompassing a wide class of important and relevant problems. For example, starting with works by Stella Dafermos [23, 22], variational inequalities have established themselves among the primary approaches for mathematical models of transportation equilibrium. Other equilibrium (Nash equilibrium) problems have also frequently been reformulated as variational inequalities to theoretically investigate problem properties and get a solution using appropriate algorithms – this approach is extensively used by the research group at the University of Massachusetts Amherst for tackling various network economics problems [70], examples of which are given later in this chapter.

After Ian Goodfellow and colleagues proposed the GAN architecture (Generative Adversarial Networks) as a promising approach for creating neural networks for content generation [48], this field began rapidly developing. The training problem for such network pairs and other adversarial architectures is essentially a saddle-point problem, naturally modeled as a variational inequality. This approach enables theoretical convergence analysis and algorithm construction for machine learning. For example, work [42] uses variational inequalities to investigate classical algorithms for training Wasserstein Linear-Quadratic Generative Adversarial Networks (LQ-GAN) and proposes an alternative algorithm with improved theoretical properties. Another model, a GAN with multiple discriminators, which also can be reformulated as a minimax problem and studied through variational inequalities – see [41].

Work [44] also demonstrates the effective use of variational inequalities in modeling human cognition, inherently not an optimization problem (due to multiple incompatible criteria, making it difficult to construct a single-objective function). However, it can be described as an equilibrium problem and solved via variational inequalities. Similarly, [43] demonstrates the applicability of variational inequality-based models to simultaneously modeling interactions between content providers (e.g., Netflix) and data transmission providers (e.g., Verizon) – see also [84].

1.3 Network Economics Problems

A broad and significant class of real problems can be modeled as processes occurring on networks of various types – transportation, economic and social connections, supply networks, etc. When these problems involve optimization or saddle-point formulations related to resource optimization or profit maximization, they are termed network economics problems [70].

1.3.1 Market Equilibrium

Tracing the beginnings of this research direction is challenging due to its long history. Classical economic problems date back to Cournot's 19th-century work [21], examining competition between two producers and utilizing the equilibrium principle later generalized as Nash equilibrium. Despite extensive market equilibrium research, it continues actively as computational power and data availability grow, enabling more sophisticated and accurate models, often analyzable through variational inequalities. Work [71], for instance, examines spatial oligopoly models for m companies supplying n markets. Equilibrium search reduces to solving the variational inequality

$$-\sum_{i=1}^m \sum_{j=1}^n \frac{\partial u_i(T^*)}{\partial T_{ij}} \times (T_{ij} - T_{ij}^*) \geq 0 \quad \forall T \in \mathbb{R}_+^{mn}.$$

Here, matrix $T \in \mathbb{R}_+^{mn}$ represents supply volumes of every company to the corresponding markets, and $u_i(T)$ denotes the profit of company i .

1.3.2 Optimal Traffic Flow Distribution

Another well-known example of a network economics problem — is the optimal distribution of traffic flow mentioned above [73, 22]. The most common approach to building models of such problems is based on Wardrop's first principle [108], where an equilibrium is a state in which no participant of traffic benefits from changing their route. Although this principle was formulated already in 1952, the relevance of the problem keeps increasing together with the number of related studies. The problem of finding the system optimum in a transport network (Wardrop's second principle), also described already in the 1960s [12], is becoming increasingly important — but with the widespread use of global navigation systems (such as Google Maps) and the appearance of monitoring and real-time traffic information systems, opportunities have finally appeared for solving it in practice. Therefore, such research is becoming increasingly active [64]. One common approach — again, is to use the framework of variational inequalities. For example, in [74] the transportation equilibrium problem in the presence of different classes of network users and different criteria that they try to optimize is reduced to the following variational inequality:

$$(F(X^*), X - X^*) \geq 0 \quad \forall X \in K,$$

where F denotes the vector $(u, -\lambda, d - d(\lambda))$ and $X = (\tilde{f}, d, \lambda)$.

In expanded form, the expression has the form:

$$\begin{aligned} & \sum_{i=1}^k \sum_{a \in L} u_a^i(\tilde{f}^*) \times (f_a^i - f_a^{i*}) - \sum_{i=1}^k \sum_{\omega \in W} \lambda_\omega^{i*} \times (d_\omega^i - d_\omega^{i*}) \\ & + \sum_{i=1}^k \sum_{\omega \in W} (d_\omega^{i*} - d_\omega^i(\lambda^*)) \times (\lambda_\omega^i - \lambda_\omega^{i*}) \geq 0 \quad \forall (\tilde{f}, d, \lambda) \in K, \end{aligned}$$

where f_a^i corresponds to the flow of users of class i along link (network edge) a , u_a^i is the corresponding cost function, which combines travel time losses and the monetary cost of passing the link, d_ω^i defines the travel demand for the origin–destination pair ω for class i , and λ_ω^i is the loss (disutility) function of users of the respective class for the corresponding pair of vertices.

Recently, researchers have also paid attention to the problem of finding transportation equilibrium for multimodal transport flow – that is, a situation where several alternative transport modes are used. One of the first works in this direction can be considered [39], where different models are studied for the situation when users sequentially employ two transport modes. For example, in the morning a person drives a private car to a certain transport hub—this may be the nearest public transport station to their home. Then, to reach the destination (workplace), public transport is used. In the evening, the reverse sequence is taken to return home. In [39], a model is studied in detail where such a regime is considered as an alternative route to using only a private car.

Further developing research in this direction, models and algorithms are actively constructed for a more general case, where many transport modes and routes are taken into account. In particular, to improve environmental conditions, many developed cities build the necessary infrastructure and motivate road users to use bicycles or light electric transport (e-scooters, scooters, etc.). A model for combinations of motorized and non-motorized transport is described, for example, in [109], where the authors consider combinations of using a car and walking. The constructed model is formulated as a variational inequality and can describe more than two travel modes, including possible transfers between them.

Another important extension of such models is taking into account different criteria by the user for choosing a travel mode (for example, traveling by bus or metro may be cheaper and even faster, but during peak hours it can be very uncomfortable because of overcrowding). Such aspects, together with the stochastic nature of travel demand, are described, for instance, in [112], where the model is also reduced to a variational inequality.

1.3.3 Supply Chains

A somewhat less known, but also important, problem of network economics is the optimization of the supply chain. This problem is especially relevant in the modern global world, where supply chains can be complex and have a crucial impact on the efficiency of a company's performance on the market. Within this problem, many different models are proposed and studied [70, 77]. For example, for a supply chain optimization model that takes into account environmental pollution, the formulation in the form of a variational inequality has the following form

$$\begin{aligned} & \sum_{a \in L} \left[\frac{\partial \hat{c}_a(f_a^*)}{\partial f_a} + \omega \frac{\partial e_a(f_a^*)}{\partial f_a} + \beta_a^* \right] \times [f_a - f_a^*] + \\ & + \sum_{a \in L} \left[\frac{\partial \hat{\pi}_a(u_a^*)}{\partial u_a} + \omega \frac{\partial \hat{e}_a(u_a^*)}{\partial u_a} - \beta_a^* \right] \times [u_a - u_a^*] + \end{aligned}$$

$$+ \sum_{a \in L} [u_a^* - f_a^*] \times [\beta_a - \beta_a^*] \geq 0, \quad \forall (f, u, \beta) \in K$$

Here f denotes the flows on network edges, u the capacities of edges, c the costs of using edges, π the capital expenses for increasing the edge capacity, e and $\hat{e}$ the functions that define the amount of harmful emissions (for example, carbon) during operation and initial construction of a network edge, which are included in the objective function with coefficient ω . Also, β are the Lagrange multipliers that arise in the construction of the objective function. The flows on edges are assumed to be nonnegative and also satisfy the classical additional conditions for such problems (the detailed formulation is presented in [77]).

Among supply chain models, one can highlight the problem of delivering a perishable product – since in this case, at each stage one has to take into account potential product losses and the related costs of disposal. For this type of goods, not only the case of shortage at the final consumption point, but also in the case of excessive supply incurs significant losses for the company – since then additional costs for disposal arise. A similar problem occurs in the design of the blood supply chain [76]. It also contains differences related to the specifics of the corresponding processes – for example, risks at the stage of blood collection from donors are additionally included in the objective function, since this procedure is voluntary and a donor may refuse the planned procedure for various reasons. The mathematical model of this problem of network economics and its reduction to a variational inequality are examined in detail in the final chapter of the dissertation.

1.3.4 Determining the Importance of Network Nodes

Another interesting problem is the problem of determining the importance of nodes in a network. A classical approach for this task is the PageRank algorithm and its numerous modifications [81]. Although in the narrow original search engine results ranking sense inclusion of this task in the scope of network economics is questionable, nowadays this approach has found its application in almost all types of networks [46]. Among clearly economic problems where PageRank is an effective approach, one can highlight the problem of evaluating the importance of nodes in large production networks [85]. Solving such a problem, for instance, allows making decisions on production optimization by shutting down the nodes with the lowest ranks. As will be shown below, this problem also can be modelled as variational inequality and solved with corresponding set of algorithms.

1.4 Constructing Efficient Algorithms for Solving Problems Modeled by Variational Inequalities

As seen from the overview above, variational inequalities serve as a model for a wide range of practical problems. Therefore, the development of efficient algorithms and the establishment of

results concerning their applicability and convergence, depending on the properties of the problem itself, is a relevant issue.

The development of such algorithms also has a long history and begins with the observation that a variational inequality can be viewed as a fixed-point problem for a corresponding operator [54]. Specifically, a point x is a solution to the variational inequality (1.1) if and only if it is a fixed point of the operator $P_C(I - \lambda A)$, where I is the identity operator, P_C is the projection operator onto the set C , and λ is a non-negative real number. For theoretical guarantees of convergence, it is required that the main operator of the problem, A , is Lipschitz continuous, i.e., $\exists L > 0 : \|Ax - Ay\| \leq L\|x - y\| \quad \forall x, y \in H$ (in this section, in most cases, the problem settings are considered in a Hilbert space H , unless stated otherwise).

Based on this equivalence, the problem can be viewed as a fixed-point problem. For such a problem, it is natural at first glance to apply the simple iteration method (which, in the case of minimization, coincides with the projected gradient method), that is, to search for a solution using the following iterative procedure:

$$x_{n+1} = P_C(x_n - \lambda Ax_n), \quad \lambda > 0,$$

where $\lambda > 0$ is chosen based on the knowledge of the Lipschitz constant of the operator A .

Such an approach was proposed in works [47, 59], and it is classical for constrained optimization problems. However, the monotonicity of the main operator alone is not sufficient to guarantee the convergence of this method in the case of a variational inequality. Stricter conditions are required – either strong monotonicity or inverse strong monotonicity (cocoercivity). Under these conditions, one can prove the convergence of the sequence of approximations generated by the algorithm to the solution of the variational inequality (either weak or strong convergence, depending on the properties of the operator).

1.4.1 Economic Algorithms

These conditions impose rather strict constraints on the problem operator, therefore, a number of alternative algorithms have been developed that allow eliminating this requirement. The beginning of the development of such algorithms can be considered the extragradient method [57], proposed back in 1976 (around the same time, with slightly different formulation, it was also proposed in [3]). For this algorithm, monotonicity of the operator is sufficient.

To solve the variational inequality (1.1), the main step of this algorithm is as follows:

$$\begin{aligned} y_n &= P_C(x_n - \lambda Ax_n), \\ x_{n+1} &= P_C(x_n - \lambda Ay_n). \end{aligned} \tag{1.2}$$

A notable drawback of this algorithm is that on each iteration it is necessary to compute the main operator of the problem (operator A) at two different points, and to perform two projections

onto the admissible set, also at two different points. Furthermore, in the original work [57], the algorithm was considered only in the finite-dimensional Euclidean space $\mathbb{R}^n$, where convergence of both the sequence x_n and the sequence y_n to the solution set of problem (1.1) was proven.

Because of this, many researchers have worked and continue to work on developing more efficient algorithms of this class, as well as on theoretical and practical investigation of their properties. For example, in [18], a modification of the method is proposed that allows simplifying one projection onto the admissible set at each iteration. The algorithm described in the work for a variational inequality in a Hilbert space H is as follows (we preserve the original notation from [18]):

Algorithm 1.1

Step 0. Choose an initial approximation $x^0 \in H$, choose a parameter $\tau > 0$, and assign $k = 0$.

Step 1. Given the current approximation x^k , compute

$$y^k = P_C (x^k - \tau f(x^k)),$$

construct an auxiliary half-space T_k such that the corresponding hyperplane is a supporting hyperplane for the set C at the point y^k :

$$T_k := \{w \in H \mid ((x^k - \tau f(x^k)) - y^k, w - y^k) \leq 0\}$$

and compute the next approximation

$$x^{k+1} = P_{T_k} (x^k - \tau f(y^k)).$$

Step 2. If $x^k = y^k$, the algorithm terminates and x^k is an approximation of the solution. Otherwise, set $k \leftarrow (k + 1)$ and return to Step 1.

In [18] the weak convergence of the sequence generated by Algorithm 1.1 to the solution set of the variational inequality (1.1) is proved, provided that the operator f is monotone and Lipschitz continuous with constant L , the solution set of (1.1) is nonempty, and $0 < \tau < 1/L$. Despite the more complicated formulation compared to the calculation procedure (1.2), this algorithm is more computationally efficient when the feasible set has a complex structure. This is achieved because instead of the second projection onto the feasible set at each step, a projection onto a special auxiliary hyperplane is used, which can be written explicitly.

Among the algorithms that allow performing fewer computations per iteration, one can also highlight the forward-backward splitting algorithm of Tseng, proposed in 2000 in [95]. In that work, the problem of finding $x \in H : 0 \in F(x) + N_C(x)$ is considered, where $N_C(x)$ is the normal cone at point x to a nonempty convex closed set $C \subset H$, and F is a maximally monotone,

continuous and single-valued operator on C , H is a Hilbert space. This problem is equivalent to the variational inequality $(F(x), x' - x) \geq 0 \forall x' \in C$, and the author proposes the following algorithm for solving it:

$$\begin{aligned}\bar{x}^k &= P_C (x^k - \alpha_k F(x^k)), \\ x^{k+1} &= \bar{x}^k - \alpha_k (F(\bar{x}^k) - F(x^k)),\end{aligned}\tag{1.3}$$

for $k = 0, 1, \dots$. As in the previous algorithm, at each iteration here one projection onto the feasible set is used, instead of two in Algorithm (1.2).

Algorithm 1.1 and Tseng's algorithm with formula (1.3) are economical in terms of the number of projections at each iteration – which provides an advantage in situations when projection onto the feasible set is computationally difficult. But if the more costly part is the computation of the problem operator, they do not provide a significant advantage compared to the classical extragradient method with formula (1.2), since they also use two operator evaluations at every iteration. One of the first methods that is economical in terms of the number of operator evaluations was proposed by L.D. Popov in [83] as early as 1980. Initially, the method was formulated for the problem of finding a saddle point of a convex–concave function $f(x, y)$ on the set $G = Q \times S \subset \mathbb{R}^n \times \mathbb{R}^m$ in the following form:

$$\begin{aligned}z_0 \in G, z_{t+1} &= P_G [z_t - \tau g(\bar{z}_t)], t = 0, 1, \dots \\ \bar{z}_0 \in G, \bar{z}_{t+1} &= P_G [z_{t+1} - \tau g(\bar{z}_t)], t = 0, 1, \dots\end{aligned}\tag{1.4}$$

where $z = (x, y)$, $g(z) = (f_x(x, y), -f_y(x, y))$

In this algorithm only one evaluation of the operator (the derivative of a convex–concave function in the original version) is used at each iteration (but two projections). This method was rediscovered in [45] under the name “extrapolation from the past”, and became a popular research subject in connection with the development of such an area of machine learning as adversarial learning, and especially generative adversarial networks.

1.4.2 Adaptive Algorithms

All the methods described above have theoretical convergence guarantees under certain restrictions on the operator of the variational inequality and the step size of the algorithm. And, to determine the step size, a priori knowledge of the Lipschitz constant of the operator is used. But for complex problem statements, determining this constant is a serious problem by itself. Therefore, another important research direction of algorithms for variational inequalities is the construction of so-called adaptive versions of the corresponding algorithms – where the step size is not set in advance based on the Lipschitz constant, but is recalculated during the algorithm based on the information available at the corresponding moment.

One of the first adaptive modifications of the extragradient method was proposed in [53]. The author considered the problem of finding a solution of the variational inequality (we keep original

formulation here):

$$(T(x), y - x) \geq 0$$

on the subset Q of the Euclidean space E^n with a continuous, monotone, single-valued operator $T : E^n \rightarrow E^n$. The modified algorithm from [53] has the following form:

$$\begin{aligned} \bar{u}^k &= P_Q (u^k - \alpha_k T(u^k)), u^0 \in Q, \\ u^{k+1} &= P_Q (u^k - \alpha_k T(\bar{u}^k)). \end{aligned} \quad (1.5)$$

where the step α_k at each iteration is chosen from the condition

$$0 < \alpha_k \leq \min \left\{ \bar{\alpha}, \varepsilon \frac{\|u^k - \bar{u}^k\|}{\|T(u^k) - T(\bar{u}^k)\|} \right\},$$

where $\bar{\alpha}$ (the maximal step) and $0 < \varepsilon < 1$ are additional parameters of the step-size selection rule. In [53] the convergence of the above algorithm to the solution set of the problem is proved without requiring Lipschitz continuity of the operator, and modifications of the algorithm for finding saddle points of convex–concave functions are also presented.

Starting from [53], many researchers have been working on constructing efficient adaptive versions of algorithms for variational inequalities. For example, one can highlight the work [69], where a version of algorithm with the main step (1.5) for a network equilibrium problem was proposed and studied. For the spatial price equilibrium in the standard formulation, the main step of the algorithm from [69] has the following form:

$$\begin{aligned} \bar{x}_{ij}^{p,k} &= \max \left\{ 0, x_{ij}^{p,k} - \alpha_k \left[F_{ij}(X^k) + \lambda_i^{p,k} - \lambda_j^{p,k} \right] \right\} \\ \bar{\lambda}_i^{p,k+1} &= \lambda_i^{p,k} - \alpha_k \left[\sum_{j \in \mathcal{A}(i)} x_{ij}^{p,k} - \sum_{j \in \mathcal{B}(i)} x_{ji}^{p,k} - b_i^p \right] \\ x_{ij}^{p,k+1} &= \max \left\{ 0, \bar{x}_{ij}^{p,k} - \alpha_k \left[F_{ij}(\bar{X}_k) + \bar{\lambda}_i^{p,k} - \bar{\lambda}_j^{p,k} \right] \right\} \\ \lambda_i^{p,k+1} &= \lambda_i^{p,k} - \alpha_k \left[\sum_{j \in \mathcal{A}(i)} \bar{x}_{ij}^{p,k} - \sum_{j \in \mathcal{B}(i)} \bar{x}_{ji}^{p,k} - b_i \right], \end{aligned}$$

where p is the index of the commodity delivered through the network, X_k and $\bar{X}_k$ are the flows (shipment volumes) along the network edges.

1.4.3 Algorithms with Non-Euclidean Distance

Another interesting and promising research direction for improving the efficiency of algorithms for variational inequalities is the replacement of metric projection with an alternative one, in order to simplify this operation for a specific problem (for the given feasible set). Most methods within this approach are based on the so-called Bregman distance, which, although not a distance in the formal sense, possesses the necessary properties for use in algorithms for variational inequalities.

The concept that later became known as Bregman divergence (also called discrepancy or Bregman distance) was introduced in [15], where the problem of finding a common point of a family of convex sets with nonempty intersection was studied. L. Bregman introduced a function $D(x, y)$ that satisfies a set of axioms, partially similar to the axioms of a distance, and proposed an approach for constructing algorithms for the studied problem, where a particular algorithm is determined by the choice of this function. He also provided examples of constructing such functions. In particular, in a real Hilbert space X the usual squared norm of the difference $D(x, y) = (x - y, x - y)$ satisfies all the required properties, which allows one to regard the function $D(x, y)$ as a generalization of a distance.

As a rule, when constructing algorithms for variational inequalities, the following approach to building Bregman divergence [15] is utilised: Select some f – strictly convex differentiable function defined on a convex subset of the Euclidean space $S \subset E^p$. Then construct the divergence in a next way:

$$D(x, y) = f(x) - f(y) - (\nabla f(y), x - y). \quad (1.6)$$

As shown in the mentioned work, such a function satisfies the given axioms, and it can be used for projection onto the feasible set instead of the Euclidean one, under some additional conditions on the function $f(x)$ and the set S . For example, such conditions can be the closedness of S and the continuous differentiability of f . In the same work, among specific examples, the function

$$f(x) = \sum_{j=1}^p x_j \ln x_j,$$

defined on the set $S = \{x \in E^p : x_i > 0, i = 1, \dots, p\}$, is given, for which (1.6) takes the form

$$D(x, y) = \sum_{j=1}^p (y_j - x_j + x_j(\ln x_j - \ln y_j)). \quad (1.7)$$

If the probability simplex is chosen as the set, then (1.7) turns into the Kullback-Leibler divergence – and based on this idea, algorithms are constructed and studied in the third chapter of this dissertation.

An elegant idea related to the use of alternative spaces was proposed by Nemirovski in [79]. Essentially, the work justifies the use of a conjugate space, which is not identified with the main space, for performing the gradient step. On this basis, an algorithm is constructed, whose speed depends less on the dimension of the space (logarithmic dependence instead of linear). The authors show that the usual step of the gradient method is, in fact, a mirror of a step in another space. At the same time, in [11] it is shown that, as a result, in the case of a feasible set of the form of a probability simplex, the obtained algorithm coincides with a modification of the extragradient method where the Bregman projection is used instead of the Euclidean one.

1.4.4 Algorithms in Hadamard Spaces

Another important direction, also related to improving the properties of the problem when moving to alternative spaces, is the construction and study of optimization algorithms in Hadamard spaces [6]. In [20] it is demonstrated how the transition to such spaces allows one to obtain an unconstrained optimization problem instead of a constrained one, or a convex minimization problem instead of a nonconvex one. In that work, formulations of equilibrium search problems (in the sense of Nash) in Hadamard spaces are also studied — and such problems, as described above, are an important class of problems that are reduced to variational inequalities.

1.5 Current State of the Field

Research within the directions outlined above continues to be actively conducted. In recent years, there were advances in directions, described above, together with investigations related to application of algorithms for variational inequalities to various problems.

For example, in [66] a version of the algorithm with one projection (and one auxiliary projection) per step for equilibrium programming problems was proposed (in the general case, this method coincides with Algorithm 1.1). In that work, the weak convergence of the sequence of approximations generated by the algorithm to the solution set of the problem was also proved.

In [67], Yu. Malitsky proposed an algorithm that can be called a projected reflected method. Its computational procedure is noticeably simpler than most previous algorithms and has the following form:

$$\begin{aligned} y_n &= 2x_n - x_{n-1}, \\ x_{n+1} &= P_C(x_n - \lambda A y_n). \end{aligned}$$

In the mentioned work, the weak convergence of the sequences generated by the algorithm to the solution set of (1.1) is proved if $\lambda \in (0, (\sqrt{2} - 1)/L)$. Obviously, the presented computational procedure is also economical and uses one projection and one operator evaluation at each iteration. Further research and improvements of this algorithm are presented in many works, in particular [118, 116].

At the same time, the extrapolation from the past algorithm from (1.4) (see also [119]) turned out to be a powerful basis for constructing other efficient algorithms. For example, in [68], based on method (1.4), an algorithm was built for solving variational inequalities with monotone operators, which uses one operator evaluation and one projection onto the feasible set at each step. It was achieved by replacing the second projection with projection onto a special hyperplane, similar to Algorithm 1.1.

The calculation procedure of the algorithm proposed in [68] has the following form:

$$\begin{aligned} x_{n+1} &= P_{T_n} (x_n - \lambda A y_n), \\ y_{n+1} &= P_C (x_{n+1} - \lambda A y_n), \end{aligned} \quad (1.8)$$

where

$$T_n = \{z \in H : (x_n - \lambda A y_{n-1} - y_n, z - y_n) \leq 0\}.$$

The convergence of the algorithm was proved under the condition $0 < \lambda < \frac{1}{3L}$. Later, in [65], a modification of the algorithm from [68] was proposed for Ky Fan inequalities, which made it possible to construct a two-step proximal method for equilibrium problems. Problems of this type have been one of the important research directions in recent years – see, among others, works [114, 30, 25].

Finally, in [19, 80], based on the above-described approaches with the use of an alternative distance, an efficient two-step proximal mirror method is proposed in the following form:

$$\begin{aligned} x_{n+1} &= \operatorname{argmin}_{y \in C} \left(F(y_n, y) + \frac{1}{\lambda} D_\varphi(y, x_n) \right), \\ y_{n+1} &= \operatorname{argmin}_{y \in C} \left(F(y_n, y) + \frac{1}{\lambda} D_\varphi(y, x_{n+1}) \right), \end{aligned} \quad (1.9)$$

for the problem

$$\text{find } x \in C : F(x, y) \geq 0 \quad \forall y \in C,$$

where $F(y, y) = 0 \quad \forall y \in C$ is a pseudomonotone, lower semicontinuous, convex bifunction on the set C , and the function φ is a continuously differentiable strongly convex function on C with strong convexity parameter δ . Accordingly,

$$D_\varphi(a, b) = \varphi(a) - \varphi(b) - (\nabla \varphi(b), a - b), \quad \forall a \in C, b \in C$$

is the Bregman divergence constructed on the basis of the function φ . Here, at each step, one needs to compute the bifunction F , and solve two optimization problems with strictly convex functions, which in terms of the number of operations is similar to (1.4). The development and study of modern algorithms with Bregman divergence (distance) is also the subject of [80], and the continuation of research and the construction of algorithms based on (1.4) with non-Euclidean distance is the topic of the third chapter of this work.

Active work is also being carried out on constructing and studying analogues of extragradient algorithms for equilibrium problems in the Hadamard spaces mentioned above. A method based on the Korpelevich algorithm for these spaces was studied in [52]. Various versions of adaptive and regularized methods of this type have been investigated in [102, 104, 100, 103, 115, 86, 99]. Finally, in [113], an adaptive version of the extraproximal algorithm was proposed, where the rule for choosing the step size includes not only its decrease, but also its increase, which has a positive

effect on the convergence speed of the algorithm, especially at the initial stages. The algorithm itself has the following main stages:

Initialization. Select $x_1 \in C, \tau \in (0, 1), \lambda_1 \in (0, +\infty)$, and define a nonnegative sequence (μ_n) with finite sum.

Main iterative step of the algorithm.

$$\begin{aligned} y_n &= {}_{\lambda F(x_n, \cdot)} x_n, \\ x_{n+1} &= {}_{\lambda F(y_n, \cdot)} x_n, \end{aligned} \tag{1.10}$$

Adaptive rule for updating the step size.

$$\lambda_{n+1} = \begin{cases} \lambda_n + \mu_n, & \text{if } F(x_n, x_{n+1}) - F(x_n, y_n) - F(y_n, x_{n+1}) \leq 0 \\ \min \left\{ \lambda_n + \mu_n, \frac{\tau}{2} \frac{d^2(x_n, y_n) + d^2(x_{n+1}, y_n)}{(F(x_n, x_{n+1}) - F(x_n, y_n) - F(y_n, x_{n+1}))} \right\}, & \text{otherwise} \end{cases}$$

where

$${}_{\lambda F(z, \cdot)} x = {}_{y \in C} \left(F(z, y) + \frac{1}{2\lambda} d^2(y, x) \right).$$

In [113] proof of convergence of the procedure above is also obtained. In this work, we also will use the adaptive step size rule from this algorithm.

Another important research direction is the construction of algorithms for variational inequalities in the case when there is a priori information about the solution in the form of the requirement that it belongs to the set of fixed points of an additional operator. Among the first works where such problems are described in detail and algorithms are proposed, one can highlight [98, 97, 96]. In 2015, in [28], the following adaptive algorithm for such a problem was proposed (which is an adaptive modification of the algorithm from [18]). Namely, the problem of finding a solution of (1.1), which belongs to the set $F(S) = \{x \in H : Sx = x\}$ of fixed points of a quasi-nonexpansive operator $S : H \rightarrow H$, is considered. Under the condition that the set $VI(A, C) \cap F(S)$ is nonempty, [28] proposes and proves the convergence of the following adaptive algorithm.

Initialization. Set $\sigma > 0, \tau \in (0, 1), \Theta \in (0, 1)$, choose a sequence $(\delta_n) \subseteq (0, 1)$ and also an initial approximation $x_0 \in H$. Let $n = 1$.

Computation of the current step size.

$$\begin{aligned} j(n) &= \min \{j \geq 0 : \\ &\sigma \tau^j \|AP_C(x_n - \sigma \tau^j Ax_n) - Ax_n\| \leq \Theta \|P_C(x_n - \sigma \tau^j Ax_n) - x_n\|\}, \\ \lambda_n &= \sigma \tau^{j(n)}. \end{aligned}$$

Main iterative step of the algorithm.

$$\begin{aligned} y_n &= P_C(x_n - \lambda_n Ax_n), \\ x_{n+1} &= \delta_n x_n + (1 - \delta_n) SP_{T_n}(x_n - \lambda_n Ay_n), \end{aligned}$$

where $T_n = \{z \in H : (x_n - \lambda_n Ax_n - y_n, z - y_n) \leq 0\}$.

Further development of these approaches is the subject of the next chapter of this work.

Contribution of this Thesis

The thesis is devoted to development, investigation and application of new and modified algorithms which allows to solve problems modeled as variational inequality - especially, equilibrium or optimum search on a network. One of the main goals was to develop and approbate software framework suitable for solving such problems using corresponding algorithms together with assisting in process of algorithm development itself.

In this work, the following results there obtained:

1. An open source software framework has been developed that enables solving a problem, modelled as variational inequality, with the extragradient-type algorithms.

In current implementation, it provides four base variants of algorithms for variational inequalities. Both adaptive and static versions are implemented, as well as modifications using Bregman divergence instead of the Euclidean distance for projection, totaling of more than ten algorithms.

Also more than ten types of problems are pre-implemented, ranging from simple test cases to more realistic problems such as PageRank computation or equilibrium search in transportation networks.

All functionality can be accessed either from a driver script (selecting the problem, the set of algorithms, and run parameters), which is convenient for researchers with Python programming skills, or without modifying the code by launching it from the command line with appropriate arguments (instructions and a short description of the software are provided together with the code on GitHub at the link above). For the blood supply chain optimization problem, web-based UI is provided to set network parameters, select algorithms and obtaining a solution together with convergence graph and detailed execution log. The software architecture is extensible, which allows researchers easily plug in own problems and algorithms, leveraging the same capabilities for computational experiments as demonstrated for built-in ones.

2. An adaptive analogue of the extragradient method has been constructed for solving variational inequalities on the set of fixed points of an auxiliary operator, and its convergence has been proven;

Step-size modification rule does not require linear search procedure and extra operator or projection calculation.

3. Adaptive variants of the extrapolation-from-the-past algorithm and the extragradient method have been proposed with the use of Kullback–Leibler divergence instead of Euclidean distance, and the corresponding convergence theorems have been proven;

This algorithm allows use closed-form formula for projection if the feasible set has specific structure. In the thesis it applied to the PageRank problem, as the feasible set there is a probability simplex.

4. An adaptive variant of the extrapolation-from-the-past algorithm has been developed in Banach spaces with uniform smoothness and 2-uniform convexity, and a convergence theorem for this algorithm has been proven;
5. Behavior of adaptive algorithms and provided software framework have been demonstrated on the blood supply chain optimization problem, transport equilibrium problem and PageRank problem.

Between others experiments, practical usefulness of step size adaptation rule with step increase possibility was numerically demonstrated.

The main results have been presented in the next international conferences with corresponding theses publications:

1. Traffic Equilibrium Modeling with Modern Algorithms for Variational Inequalities [36], presented by the author at MODS 2024 international conference, November 11, 2024, Chernihiv, Ukraine (online)
2. Optimizing network economics problem with adaptive algorithms for variational inequalities [35], presented by the author at DSMSI-2023 international conference, December 05, 2023, Kyiv, Ukraine (online)
3. Adaptive Variants of the Extrapolation from the Past Method and the Operator Extrapolation Method [33], presented by the author at MODS 2022 international conference, November 14, 2022, Chernihiv, Ukraine (online)
4. Convergence of adaptive operator extrapolation method for operator inclusions in Banach Spaces [34], presented by the author at CMIS-2022 international conference, May 12, 2022, Zaporizhzhia, Ukraine (online)

and published in the next scientific papers

1. Convergence of the Method of Extrapolation from the Past for Variational Inequalities in Uniformly Convex Banach Spaces [90], Cybernetics and Systems Analysis, 2022
2. Adaptive Two-Stage Bregman Method for Variational Inequalities [91], Cybernetics and Systems Analysis, 2021
3. Bregman Extragradient Method with Monotone Rule of Step Adjustment [29], Cybernetics and Systems Analysis, 2019
4. Convergence of extragradient algorithm with monotone step size strategy for variational inequalities and operator equations [31], Journal of Automation and Information Sciences, 2019

Chapter 2

Algorithm with Monotone Step Size Adjustment for Variational Inequalities with Additional Conditions

An important and difficult class of problems studied in the theory of ill-posed problems are problems with additionally given a priori information. In such problems, together with the main problem (for example, in the form of an inequality or a system of equations), an additional set is also specified, which restricts the search for the solution. Moreover, this set is often not defined explicitly, as in classical formulations, but is described by an auxiliary problem, also in the form of a system of equations and constraints. The auxiliary problem usually implicitly describes key properties of the studied object or process. Problems of this type and approaches to their solution are described in the already mentioned works [98, 96].

Incorporating additional information about the essence of the problem directly into the algorithm quite often makes it possible to improve the accuracy of the obtained solution of the original ill-posed problem. Moreover, the problem itself, obtained through mathematical modeling of an applied problem, often does not have a unique solution. In such a situation, successfully chosen additional information makes it possible to single out the solution that corresponds to the real behavior of the process.

In the case when the main problem is modeled by an operator equation of the first kind, good results are obtained by modeling additional constraints of the solution set through inclusion into the set of fixed points of certain operators. Such an approach is studied, for example, in [98], where for this scenario one-step iterative algorithms are proposed, which are a combination of the main scheme for solving the corresponding problem and the Krasnoselskii–Mann algorithm. In [28, 105], similar methods are studied based on combining a modified extragradient method and the Halpern algorithm [49], or again the Krasnoselskii–Mann algorithm.

This chapter provides further development of the ideas and approaches of [28, 105].

An algorithm for approximate solution of variational inequalities in a Hilbert space H is proposed and studied.

The main difference is the presence of additional conditions of the type of inclusion into the set of fixed points of a given operator. Thus, the problem consists in finding an element $x \in C$, such that

$$(Ax, y - x) \geq 0 \quad \forall y \in C \text{ and, simultaneously, } x = Sx.$$

Here the main feasible set is $C \subseteq H$, the main operator of the problem is $A : H \rightarrow H$, and the auxiliary operator is $S : H \rightarrow H$. To solve this class of problems, a modified extragradient method is constructed. Thanks to the procedure of monotone adjustment of the step size, the algorithm can be applied without knowing the Lipschitz constant of the main operator. An advantage of the method is also the absence of a requirement to use additional evaluations of the variational inequality operator and the projection operator at each iteration. For the sequence of approximations produced by the algorithm, weak convergence is proved. It is established under the assumption that the main operator is pseudomonotone, Lipschitz continuous, and sequentially weakly continuous, and that the operator describing the additional constraints is quasi-nonexpansive.

2.1 Necessary Theory and Problem Statement

In this chapter, the problem and algorithms are considered in a real Hilbert space H . The scalar product is denoted by $(\cdot, \cdot)$, and the corresponding norm is denoted by $\|\cdot\|$.

For the projection operator onto a convex closed set C , we will use the notation P_C . Recall that $P_C(x)$ is the unique element of the set C such that

$$\|P_C(x) - x\| = \min_{z \in C} \|z - x\|.$$

The following properties of the element $P_C(x)$ are important [8]:

$$y = P_C(x) \Leftrightarrow y \in C \text{ and } (y - x, z - y) \geq 0 \quad \forall z \in C, \quad (2.1)$$

$$y = P_C(x) \Leftrightarrow y \in C \text{ and } \|y - z\|^2 \leq \|x - z\|^2 - \|y - x\|^2 \quad \forall z \in C. \quad (2.2)$$

An operator $T : H \rightarrow H$ is called quasi-nonexpansive (Fejér-type) if [8]

$$F(T) = \{x \in H : Tx = x\} \neq \emptyset \text{ and } \|Tx - y\| \leq \|x - y\| \quad \forall x \in H, y \in F(T).$$

The set of fixed points $F(T)$ of a quasi-nonexpansive operator is closed and convex.

An operator $S : C \rightarrow H$ is called demiclosed at $y \in H$ if for a sequence of points $x_n \in C$ with $x_n \rightarrow x$ weakly and $Sx_n \rightarrow y$ strongly, it follows that $Sx = y$. For a nonexpansive operator $T : C \rightarrow H$, the operator $I - T$ is demiclosed at zero [8].

Let C be a nonempty subset of the space H , and A an operator acting in H . We will consider the variational inequality of the form (1.1), whose solution set will be denoted by $VI(A, C)$. From (2.1) it follows that $x \in VI(A, C)$ if and only if

$$x = P_C(x - \lambda Ax),$$

where $\lambda > 0$ [54].

An operator A is called pseudomonotone if for all $x, y \in H$ with $(Ax, y - x) \geq 0$ it follows that $(Ay, y - x) \geq 0$ [55]. If the operator $A : H \rightarrow H$ is pseudomonotone and continuous, and the set $C \subseteq H$ is convex and closed, then $x \in VI(A, C)$ if and only if $x \in C$ and $(Ay, y - x) \geq 0$ for all $y \in C$ [55]. Moreover, the set $VI(A, C)$ is convex and closed.

To prove weak convergence of sequences of elements of a Hilbert space, we use the well-known Opial's lemma [8].

Lemma 2.1. Let the sequence (x_n) of elements of a Hilbert space H converge weakly to an element $x \in H$. Then for all $y \in H \setminus \{x\}$ we have $\lim_{n \rightarrow \infty} \|x_n - x\| < \lim_{n \rightarrow \infty} \|x_n - y\|$.

So, we consider the problem of finding a solution of the variational inequality (1.1) that is a fixed point of a given operator S :

$$\text{find } x \in VI(A, C) \cap F(S). \quad (2.3)$$

Assume the following conditions hold:

- the set $C \subseteq H$ is convex and closed;
- the operator $A : H \rightarrow H$ is pseudomonotone, Lipschitz continuous, and sequentially weakly continuous;
- the operator $S : H \rightarrow H$ is quasi-nonexpansive and such that the operator $I - S$ is demi-closed at zero;
- the set $VI(A, C) \cap F(S)$ is nonempty.

Remark 2.1. Let $g : H \rightarrow \mathbb{R}$ be a convex differentiable function. If the set $D = \{x \in H : g(x) \leq 0\}$ is nonempty, then it can be treated as the set of fixed points of a quasi-nonexpansive operator

$$Sx = \begin{cases} x - \frac{g(x)}{\|g'(x)\|^2} g'(x), & \text{if } g(x) > 0, \\ x, & \text{if } g(x) \leq 0, \end{cases}$$

where $g'(x) \in H$ is the derivative of g at the point $x \in H$. For demi-closedness at zero of the operator $I - S$, it is sufficient that g is bounded on any bounded set [98].

Remark 2.2. If the operator A is monotone, then the results presented in this section hold without the assumption of sequential weak continuity of A .

2.2 Extragradient Algorithm with Monotone Step-Size Adjustment

We construct an algorithm that generates a sequence converging to a solution of the problem (2.3), i.e. to some element of $VI(A, C) \cap F(S)$. This algorithm can be viewed as a combination of modified extragradient algorithm with an adaptive step-size selection procedure, and Krasnoselskii–Mann algorithm.

Algorithm 2.1

Initialization. Choose an element $x_1 \in H$, $\tau \in (0, 1)$, a positive number λ_1 , and a sequence $(\delta_n) \subseteq [a, b] \subseteq (0, 1)$. Assume $n = 1$.

Step 1. Compute

$$y_n = P_C (x_n - \lambda_n A x_n).$$

Step 2. Compute

$$z_n = P_{T_n} (x_n - \lambda_n A y_n)$$

where

$$T_n = \{z \in H : (x_n - \lambda_n A x_n - y_n, z - y_n) \leq 0\}.$$

Step 3. Compute

$$x_{n+1} = \delta_n x_n + (1 - \delta_n) S z_n$$

Step 4. Compute

$$\lambda_{n+1} = \begin{cases} \min \left\{ \lambda_n, \tau \frac{\|y_n - x_n\|}{\|A y_n - A x_n\|} \right\}, & \text{if } A x_n \neq A y_n, \\ \lambda_n, & \text{otherwise.} \end{cases}$$

Set $n := n + 1$ and go to Step 1.

Remark 2.3. In contrast to the step-size choice rules λ_n from [53, 28, 105], the algorithm does not require additional calculations of operator A values and projections P_C .

Remark 2.4. The auxiliary subspace T_n , onto which the projection is performed at the second step of the algorithm, contains the feasible set C . This can be easily proved by contradiction — indeed, if $\exists w \in C \setminus T_n$, then the inequality

$$(x_n - \lambda_n A x_n - y_n, w - y_n) > 0$$

and the equality $y_n = P_C (x_n - \lambda_n A x_n)$ cannot hold simultaneously.

2.3 Convergence proof

For the elements of the sequences generated by the algorithm, we need tools to estimate the distance between these elements and the set $VI(A, C) \cap F(S)$. Let us prove the corresponding inequalities.

Lemma 2.2. For the sequences (x_n) , (y_n) and (z_n) generated by Algorithm 2.1, the following inequality holds:

$$\|z_n - z\|^2 \leq \|x_n - z\|^2 - (1 - \mu_n) \|y_n - x_n\|^2 - (1 - \mu_n) \|z_n - y_n\|^2,$$

where $z \in VI(A, C)$, $\mu_n = \tau(\lambda_n/\lambda_{n+1})$.

Proof. Let $z \in VI(A, C)$. Write the identity

$$\|z_n - z\|^2 = \|x_n - z\|^2 - \|z_n - x_n\|^2 + 2(z_n - x_n, z_n - z). \quad (2.4)$$

From the construction of the points z_n it follows that

$$\lambda_n (Ay_n, z - z_n) + (z_n - x_n, z - z_n) \geq 0. \quad (2.5)$$

Using inequality (2.5) to estimate the scalar product in (2.4), we obtain

$$\|z_n - z\|^2 \leq \|x_n - z\|^2 - \|z_n - x_n\|^2 + 2\lambda_n (Ay_n, z - z_n).$$

The norm $\|z_n - x_n\|^2$ can be rewritten as

$$\|z_n - x_n\|^2 = \|z_n - y_n\|^2 + \|y_n - x_n\|^2 + 2(y_n - x_n, z_n - y_n).$$

Thus, we have

$$\begin{aligned} \|z_n - z\|^2 &\leq \|x_n - z\|^2 - \|z_n - y_n\|^2 - \|y_n - x_n\|^2 + \\ &+ 2(x_n - \lambda_n Ay_n - y_n, z_n - y_n) + 2\lambda_n (Ay_n, z - y_n). \end{aligned}$$

From the pseudomonotonicity of the operator A it follows that

$$(Ay_n, z - y_n) \leq 0.$$

Then we obtain the following estimate of the difference:

$$\begin{aligned} \|z_n - z\|^2 &\leq \|x_n - z\|^2 - \|z_n - y_n\|^2 - \|y_n - x_n\|^2 + \\ &+ 2(x_n - \lambda_n Ay_n - y_n, z_n - y_n). \end{aligned}$$

Since $z_n \in T_n$, we have

$$(x_n - \lambda_n Ay_n - y_n, z_n - y_n) = \underbrace{(x_n - \lambda_n Ax_n - y_n, z_n - y_n)}_{\leq 0} +$$

$$+\lambda_n (Ax_n - Ay_n, z_n - y_n) \leq \lambda_n (Ax_n - Ay_n, z_n - y_n).$$

Taking this into account in the previous inequality, we obtain

$$\|z_n - z\|^2 \leq \|x_n - z\|^2 - \|z_n - y_n\|^2 - \|y_n - x_n\|^2 + 2\lambda_n (Ax_n - Ay_n, z_n - y_n).$$

Now let us estimate the term $2\lambda_n(Ax_n - Ay_n, z_n - y_n)$ as follows:

$$\begin{aligned} 2\lambda_n(Ax_n - Ay_n, z_n - y_n) &\leq 2\lambda_n \|Ax_n - Ay_n\| \|z_n - y_n\| \leq \\ &\leq \frac{2\lambda_n}{\lambda_{n+1}} \tau \|x_n - y_n\| \|z_n - y_n\| \leq \mu_n (\|x_n - y_n\|^2 + \|z_n - y_n\|^2). \end{aligned}$$

Finally, applying this estimate to the previous inequality, we obtain

$$\|z_n - z\|^2 \leq \|x_n - z\|^2 - \|x_n - y_n\|^2 - \|y_n - z_n\|^2 + \mu_n (\|x_n - y_n\|^2 + \|z_n - y_n\|^2),$$

which completes the proof. $\square$

Lemma 2.3. For the sequences (x_n) , (y_n) , and (z_n) generated by Algorithm 2.1, the following inequality holds:

$$\|x_{n+1} - z\|^2 \leq \|x_n - z\|^2 - (1 - \delta_n)(1 - \mu_n)\|y_n - x_n\|^2 - (1 - \delta_n)(1 - \mu_n)\|z_n - y_n\|^2 - \delta_n(1 - \delta_n)\|x_n - Sz_n\|^2,$$

where $z \in VI(A, C) \cap F(S)$.

Proof. Since the operator S is quasi-nonexpansive, for all $z \in VI(A, C) \cap F(S)$ we have

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \delta_n \|x_n - z\|^2 + (1 - \delta_n) \|Sz_n - z\|^2 = \\ &= \delta_n \|x_n - z\|^2 + (1 - \delta_n) \|Sz_n - z\|^2 - \delta_n(1 - \delta_n) \|x_n - Sz_n\|^2 \leq \\ &\leq \delta_n \|x_n - z\|^2 + (1 - \delta_n) \|z_n - z\|^2 - \delta_n(1 - \delta_n) \|x_n - Sz_n\|^2. \end{aligned}$$

Using in this expression inequality of Lemma 2.2 to estimate the term $(1 - \delta_n) \|z_n - z\|^2$, we obtain the desired result. $\square$

Using the lemmas above, we now formulate and prove the convergence theorem for the algorithm.

Theorem 2.1. The sequences (x_n) , (y_n) and (z_n) generated by Algorithm 2.1 converge weakly to some point $z \in VI(A, C) \cap F(S)$.

Proof. The sequence (λ_n) is non-increasing and bounded below by

$$\min \left\{ \lambda_1, \frac{\tau}{L} \right\},$$

where L is the Lipschitz constant of operator A . Hence, $\lim_{n \rightarrow \infty} \lambda_n > 0$. And since

$$1 - \mu_n = 1 - \tau (\lambda_n / \lambda_{n+1}) \rightarrow 1 - \tau \in (0, 1) \quad \text{as } n \rightarrow \infty,$$

from Lemma 2.3 we can conclude that for some $\theta \in (0, 1)$ and $m \in \mathbb{N}$ the following inequality holds:

$$\begin{aligned} \|x_{n+1} - z\|^2 &\leq \|x_n - z\|^2 - (1 - \delta_n) \theta \|y_n - x_n\|^2 - \\ &\quad - (1 - \delta_n) \theta \|z_n - y_n\|^2 - \delta_n (1 - \delta_n) \|x_n - Sz_n\|^2 \quad \forall n \geq m \end{aligned}$$

where $z \in VI(A, C) \cap F(S)$.

From the obtained inequality it follows that the sequence (x_n) is Fejér monotone with respect to the set $VI(A, C) \cap F(S)$, that is,

$$\|x_{n+1} - z\|^2 \leq \|x_n - z\|^2 \quad \forall n \geq m \quad \forall z \in VI(A, C) \cap F(S).$$

This implies the boundedness of the sequence (x_n) . Moreover, the following inequalities hold:

$$\begin{aligned} \|x_n - y_n\|^2 + \|z_n - y_n\|^2 &\leq \frac{\|x_n - z\|^2 - \|x_{n+1} - z\|^2}{(1 - \delta_n) \theta}, \\ \|x_n - Sz_n\|^2 &\leq \frac{\|x_n - z\|^2 - \|x_{n+1} - z\|^2}{\delta_n (1 - \delta_n)}. \end{aligned}$$

Hence, it follows that

$$\lim_{n \rightarrow \infty} \|x_n - y_n\| = \lim_{n \rightarrow \infty} \|z_n - y_n\| = \lim_{n \rightarrow \infty} \|x_n - Sz_n\| = 0. \quad (2.6)$$

Consider a subsequence (x_{n_k}) that converges weakly to some point $z \in H$. Then $(y_{n_k}), (z_{n_k})$ converge weakly to z , and $z \in C$. Let us show that z is a solution of the variational inequality (1.1). Assume that $Az \neq 0$ (otherwise the proof is complete, since $z \in VI(A, C)$). We have

$$(y_{n_k} - x_{n_k} + \lambda_{n_k} Ax_{n_k}, x - y_{n_k}) \geq 0 \quad \forall x \in C \quad (2.7)$$

Then

$$\begin{aligned} 0 &\leq \frac{(y_{n_k} - x_{n_k} + \lambda_{n_k} Ax_{n_k}, x - y_{n_k})}{\lambda_{n_k}} = \frac{(y_{n_k} - x_{n_k}, x - y_{n_k})}{\lambda_{n_k}} + \\ &\quad + (Ax_{n_k} - Ay_{n_k}, x - y_{n_k}) + (Ay_{n_k}, x - y_{n_k}). \end{aligned}$$

Passing to the limit as $k \rightarrow \infty$, we obtain

$$\varliminf_{k \rightarrow \infty} (Ay_{n_k}, x - y_{n_k}) \geq 0 \quad (2.8)$$

From (2.8) it follows that $\forall k \in \mathbb{N}$ there exists the smallest number $m_k \in \mathbb{N}$ such that

$$(Ay_{n_i}, x - y_{n_i}) \geq -2^{-k} \quad \text{for all } i \geq m_k.$$

The sequence (m_k) is increasing. From the sequential weak continuity of the operator A and the weak lower semicontinuity of the norm we obtain

$$0 < \|Az\| \leq \varliminf_{k \rightarrow \infty} \|Ay_{n_{m_k}}\| \quad (2.9)$$

We may assume that $Ay_{n_{m_k}} \neq 0$. Set $v_k = \|Ay_{n_{m_k}}\|^{-2} Ay_{n_{m_k}}$. Clearly,

$$(Ay_{n_{m_k}}, v_k) = 1 \quad \text{for all } k \in \mathbb{N}.$$

We have

$$(Ay_{n_{m_k}}, x + 2^{-k}v_k - y_{n_{m_k}}) \geq 0$$

From the pseudomonotonicity of the operator A it follows that

$$(A(x + 2^{-k}v_k), x + 2^{-k}v_k - y_{n_{m_k}}) \geq 0$$

Hence we obtain

$$(Ax, x - y_{n_{m_k}}) \geq (Ax - A(x + 2^{-k}v_k), x + 2^{-k}v_k - y_{n_{m_k}}) - 2^{-k}(Ax, v_k).$$

From (2.9) it follows that the sequence $\|v_k\| = \|Ay_{n_{m_k}}\|^{-1}$ is bounded above. Therefore, $\lim_{k \rightarrow 0} 2^{-k} \|v_k\| = 0$. Passing to the limit in the last inequality, we have

$$(Ax, x - z) = \lim_{k \rightarrow \infty} (Ax, x - y_{n_{m_k}}) = \varliminf_{k \rightarrow \infty} (Ax, x - y_{n_{m_k}}) \geq 0 \quad \forall x \in C.$$

Thus, $z \in VI(A, C)$.

Now let us prove that z is a fixed point of the operator S . Taking into account that

$$\|z_n - Sz_n\| \leq \|z_n - y_n\| + \|y_n - x_n\| + \|x_n - Sz_n\|,$$

from (2.6) it follows that

$$\lim_{n \rightarrow \infty} \|z_n - Sz_n\| = 0.$$

The operator $I - S$ is demiclosed at zero. Hence, from the weak convergence $z_{n_k} \rightarrow z$ and the fact that $\lim_{k \rightarrow \infty} \|z_{n_k} - Sz_{n_k}\| = 0$, we obtain that $z \in F(S)$.

Let us now show that $x_n \rightarrow z$ weakly. Then from (2.6) it will also follow that the sequences $(y_n), (z_n)$ converge weakly to z . We prove this by contradiction. Suppose that there exists a subsequence (x_{m_k}) such that $x_{m_k} \rightarrow z'$ weakly and $z \neq z'$. Clearly, $z' \in VI(A, C) \cap F(S)$. Applying Opial's lemma twice, we obtain

$$\begin{aligned} \lim_{n \rightarrow \infty} \|x_n - z\| &= \varliminf_{k \rightarrow \infty} \|x_{n_k} - z\| < \varliminf_{k \rightarrow \infty} \|x_{n_k} - z'\| = \lim_{n \rightarrow \infty} \|x_n - z'\| = \\ &= \varliminf_{k \rightarrow \infty} \|x_{m_k} - z'\| < \varliminf_{k \rightarrow \infty} \|x_{m_k} - z\| = \lim_{n \rightarrow \infty} \|x_n - z\|. \end{aligned}$$

This is a contradiction — hence, $z = z'$.

Remark 2.5. In the case of monotonicity of the operator A , the limit passage in inequality (2.7) to prove $z \in VI(A, C)$ is carried out without the assumption of sequential weak continuity. Indeed, in this situation we have the inequality

$$\begin{aligned} 0 &\leq \frac{(y_{n_k} - x_{n_k}, x - y_{n_k})}{\lambda_{n_k}} + (Ax_{n_k}, x_{n_k} - y_{n_k}) + (Ax_{n_k}, x - x_{n_k}) \leq \\ &\leq \frac{(y_{n_k} - x_{n_k}, x - y_{n_k})}{\lambda_{n_k}} + (Ax_{n_k}, x_{n_k} - y_{n_k}) + (Ax, x - x_{n_k}). \end{aligned}$$

And by taking the limit, we immediately obtain

$$(Ax, x - z) \geq 0 \quad \forall x \in C,$$

that is, $z \in VI(A, C)$. □

Remark 2.5. Let the sequence (x_n) , obtained by Algorithm 2.1, converge weakly to some $z \in VI(A, C) \cap F(S)$. Then it can be shown that strong convergence holds: $P_{VI(A, C) \cap F(S)}(x_n) \rightarrow z$ [8]. And when the set $VI(A, C) \cap F(S)$ is an affine manifold, then x_n converges strongly to the projection of the initial approximation onto the solution set of the problem, that is, $x_n \rightarrow P_{VI(A, C) \cap F(S)}(x_0)$ [8].

2.4 Algorithm Variant for an Operator Equation with A Priori Information

Let us now consider the operator equation with a priori information given in the form of the set of fixed points of a quasi-nonexpansive operator $T : H \rightarrow H$:

$$Ax = f, \quad x \in F(T), \tag{2.10}$$

where $f \in H$. Similar problems were considered earlier in the monograph [98].

Algorithm 2.1 for the problem (2.10) becomes Algorithm 2.2 below, and the analogue of 2.1 also holds in the form of theorem 2.2

For this problem formulation and the given algorithm, the analogue of Theorem 2.1 is the following theorem 2.2.

Theorem 2.2. For the sequences (x_n) , (y_n) and (z_n) generated by Algorithm 2.2, weak convergence to some point z from the solution set of the problem holds under the following conditions:

1. The operator $A : H \rightarrow H$ is pseudomonotone, Lipschitz continuous, and sequentially weakly continuous.
2. The operator $T : H \rightarrow H$ is quasi-nonexpansive, and the operator $I - T$ is demi-closed at zero.
3. The solution set $A^{-1}f \cap F(T)$ is nonempty.

2.5 Summary of Chapter 2

This chapter considered variational inequalities (and operator equations) in a Hilbert space with additional conditions of the type of inclusion into the set of fixed points of a given operator. For approximate solution of such problems, an algorithm was proposed, which is a composition of a

Algorithm 2.2

Initialization. Choose an element $x_1 \in H$, $\tau \in (0, 1)$, a positive number λ_1 , and a sequence $(\delta_n) \subseteq [a, b] \subseteq (0, 1)$. Assume $n = 1$.

Step 1. Compute

$$y_n = x_n - \lambda_n (Ax_n - f).$$

Step 2. Compute

$$z_n = x_n - \lambda_n (Ay_n - f)$$

Step 3. Compute

$$x_{n+1} = \delta_n x_n + (1 - \delta_n) T z_n$$

Step 4. Compute

$$\lambda_{n+1} = \begin{cases} \min \left\{ \lambda_n, \tau \frac{\|y_n - x_n\|}{\|Ay_n - Ax_n\|} \right\}, & \text{if } Ax_n \neq Ay_n, \\ \lambda_n, & \text{otherwise.} \end{cases}$$

Set $n := n + 1$ and go to Step 1.

modified extragradient algorithm with monotone step-size adjustment (that does not require knowledge of the Lipschitz constant of the operator) and the Krasnoselskii–Mann scheme for fixed-point approximation. Unlike the step-size selection rules used earlier, in this algorithm no additional evaluations of the operator and the projection mapping are performed.

The main result is the theorem on weak convergence of the algorithm for problems with pseudomonotone, Lipschitz continuous, sequentially weakly continuous operators, and quasi-nonexpansive operators defining additional conditions.

The results of this chapter were published in [31].

Chapter 3

Algorithms with Accelerated Projection Using Bregman Divergence

A widely used variant of the gradient projection method for solving the variational inequality (1.1) is the extragradient method of Korpelevich, described in more detail in the first chapter.

As already mentioned, one of the effective modifications of this method is the mirror method, proposed by A. Nemirovski in [78]. In this method, the projection is performed using Bregman divergence instead of the Euclidean distance, which significantly increases its efficiency for some practically important problem formulations. For example, if the feasible set C in (1.1) is a probability simplex, then instead of the Euclidean distance one can use the Kullback–Leibler divergence, which allows replacing the iterative projection algorithm onto the simplex by computing the projection in closed form. In turn, the Kullback–Leibler divergence is a Bregman divergence constructed on the basis of negative entropy (see more details in the first chapter, as well as in the theoretical preliminaries below).

In this chapter, this approach is further developed, and a new variant of the extragradient method is proposed for solving variational inequalities with pseudomonotone and Lipschitz continuous operators acting in a finite-dimensional linear normed space. The methods employ Bregman divergence instead of the Euclidean distance and a new step-size selection rule that does not require knowledge of the Lipschitz constant of the operator. Unlike the step-size rules applied earlier, in the proposed methods no additional computations of operator values and proximal mappings are performed. Convergence theorems for the methods are proved.

3.1 Problem Statement and Preliminaries

Let E be a finite-dimensional real linear space. We introduce a norm $\|\cdot\|$ (which may be non-Euclidean) on E . The dual space will be denoted by E^* . For $a \in E^*$ and $b \in E$, we denote by (a, b) the value of the linear functional a at the point b .

The dual norm $\|\cdot\|_*$ on E^* is defined in the standard way:

$$\|a\|_* = \max\{(a, b) : \|b\| = 1\}.$$

Note that the Cauchy–Schwarz inequality holds:

$$(a, b) \leq \|a\|_* \|b\| \quad \text{for all } a \in E^*, b \in E.$$

We will consider the problem of finding a solution of the variational inequality (1.1). Assume that C is a nonempty subset of the space E , A is an operator acting from E to E^* , and the following conditions hold:

- the set $C \subseteq E$ is convex and closed;
- the operator $A : E \rightarrow E^*$ is pseudomonotone and Lipschitz continuous with constant $L > 0$ on C ;
- the solution set S of (1.1) is nonempty.

Remark 3.1. Pseudomonotonicity of the operator A on the set C means that for all $x, y \in C$, from $(Ax, y - x) \geq 0$ it follows that $(Ay, y - x) \geq 0$. In turn, monotonicity of the operator A on the set C means that for all $x, y \in C$ the inequality $(Ax - Ay, x - y) \geq 0$ holds.

As an example, consider the operator $T : \mathbb{R}^m \rightarrow \mathbb{R}^m$ of the form

$$Tx = (e^{-(Ax, x)} + \beta)(Cx + d), \quad x \in \mathbb{R}^m,$$

where A is some positive definite matrix, $\beta > 0$, C is a positive semidefinite matrix, and $d \in \mathbb{R}^m$. Such an operator will not be monotone in the general case, but it will be pseudomonotone and Lipschitz continuous on $\mathbb{R}^m$.

Consider the dual variational inequality [56]:

$$\text{find } x \in C : \quad (Ay, x - y) \leq 0 \quad \forall y \in C. \quad (3.1)$$

We denote the solution set of (3.1) by S^d . Note that the set S^d is convex and closed. Inequality (3.1) is sometimes called the weak or dual formulation of (1.1), and the solutions of (3.1) are weak solutions of (1.1) [56]. For pseudomonotone operators A we always have $S \subseteq S^d$. Under our assumptions, $S^d = S$ [56].

Now let us introduce constructions necessary for formulating the algorithms. Let a function $\varphi : E \rightarrow \overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ satisfy the following conditions:

1. $\text{int dom } \varphi \subseteq E$ is a nonempty convex set;
2. φ is continuously differentiable on $\text{int dom } \varphi$;

3. if $\text{int dom } \varphi \ni x_n \rightarrow x \in \text{fr dom } \varphi$, then $\|\nabla \varphi(x_n)\|_* \rightarrow +\infty$;
4. φ is strongly convex with respect to the norm $\|\cdot\|$ with constant $\sigma > 0$:

$$\varphi(a) \geq \varphi(b) + (\nabla \varphi(b), a - b) + \frac{\sigma}{2} \|a - b\|^2 \quad \forall a \in \text{dom } \varphi, b \in \text{int dom } \varphi.$$

From the function φ with the above properties one can construct the Bregman divergence as follows [10]:

$$V(a, b) = \varphi(a) - \varphi(b) - (\nabla \varphi(b), a - b) \quad \forall a \in \text{dom } \varphi, b \in \text{int dom } \varphi. \quad (3.2)$$

Remark 3.2. Sometimes the Bregman divergence is called a distance [78, 88, 10], but this is not formally correct: from the metric axioms, for V in the general case only

$$V(x, y) = 0 \Leftrightarrow x = y$$

holds.

Many classical variants of Bregman divergences are presented in [10], where their possible applications are also outlined. The simplest example is the squared distance in a Hilbert space. Indeed, if we take $\varphi(\cdot) = \frac{1}{2} \|\cdot\|_2^2$, where $\|\cdot\|_2$ is the Euclidean norm, then the Bregman divergence has the form

$$V(x, y) = \frac{1}{2} \|x - y\|_2^2.$$

The Bregman divergence used in this chapter is constructed on the basis of the negative entropy function $\varphi(x) = \sum_{i=1}^m x_i \ln x_i$, considered on the set $\mathbb{R}_+^m = \{x \in \mathbb{R}^m : x_i \geq 0\}$. If we consider the probability simplex $S_m = \{x \in \mathbb{R}^m : x_i \geq 0, \sum_{i=1}^m x_i = 1\}$ with the corresponding ℓ_1 -norm, then on it this function will be strongly convex, with strong convexity constant equal to 1. The induced Bregman divergence has the form

$$V(x, y) = \sum_{i=1}^m x_i \ln(x_i/y_i) - \sum_{i=1}^m (x_i - y_i), \quad x \in \mathbb{R}_+^m, y \in \mathbb{R}_{++}^m = \text{int}(\mathbb{R}_+^m).$$

Note that this construction is known as the *Kullback–Leibler divergence* and has an independent value in information theory.

To obtain the theoretical results of this chapter, the following equality is important [10]:

$$V(a, c) = V(a, b) + V(b, c) + (\nabla \varphi(b) - \nabla \varphi(c), a - b). \quad (3.3)$$

Since the function φ is strongly convex, the following inequality holds:

$$V(a, b) \geq \frac{\sigma}{2} \|a - b\|^2 \quad \forall a \in \text{dom } \varphi, b \in \text{int dom } \varphi. \quad (3.4)$$

Let $K \subseteq \text{dom } \varphi$ be a nonempty closed convex set, with $K \cap \text{int dom } \varphi \neq \emptyset$. Consider the strongly convex minimization problems of the form

$$P_x^K(a) = \arg \min_{y \in K} \{-(a, y - x) + V(y, x)\} \quad \forall a \in E^*, x \in \text{int dom } \varphi. \quad (3.5)$$

It is known [10] that problem (3.5) has a unique solution $z \in K \cap \text{int dom } \varphi$, and moreover

$$-(a, y - z) + (\nabla \varphi(z) - \nabla \varphi(x), y - z) \geq 0 \quad \forall y \in K. \quad (3.6)$$

The mapping $P_x^K : E^* \rightarrow K \cap \text{int dom } \varphi$ is called the prox-mapping.

Remark 3.3. The point $P_x^K(a)$ in the Euclidean case coincides with the Euclidean metric projection $P_K(x + a) = \arg \min_{y \in K} \|y - (x + a)\|_2$.

Remark 3.4. For the simplex $S_m = \{x \in \mathbb{R}^m : x_i \geq 0, \sum_{i=1}^m x_i = 1\}$ and the Kullback–Leibler divergence we have [10]

$$P_x^{S_m}(a) = \left(\frac{x_1 e^{a_1}}{\sum_{j=1}^m x_j e^{a_j}}, \frac{x_2 e^{a_2}}{\sum_{j=1}^m x_j e^{a_j}}, \dots, \frac{x_m e^{a_m}}{\sum_{j=1}^m x_j e^{a_j}} \right), \quad a \in \mathbb{R}^m, x \in \text{ri}(S_m).$$

3.2 Algorithm

To find an element of the solution set (1.1), the following algorithm is proposed:

Algorithm 3.1.

Initialization. Set $x_1 \in \text{int dom } \varphi$, $\tau \in (0, \sigma)$, and $\lambda_1 > 0$. Start with $n = 1$.

Step 1.

$$y_n = P_{x_n}^C(-\lambda_n A x_n)$$

Step 2. If $y_n = x_n$, then the algorithm is terminated. Otherwise, continue

$$x_{n+1} = P_{x_n}^C(-\lambda_n A y_n),$$

Step 3. Adaptive step-size adjustment

$$\lambda_{n+1} = \begin{cases} \min \left\{ \lambda_n, \tau \sqrt{\frac{2}{\sigma} \frac{V(y_n, x_n)}{\|A y_n - A x_n\|_*^2}} \right\}, & \text{if } A x_n \neq A y_n, \\ \lambda_n, & \text{otherwise.} \end{cases}$$

End of iteration. Set $n := n + 1$ and return to Step 1.

Remark 3.5. Unlike the step-size choice rules λ_n in [53, 28, 105, 78], Algorithm 3.1 does not perform additional evaluations of the operator A or the prox-mapping $P_{x_n}^C$.

Remark 3.6. The sequence (λ_n) is non-increasing and bounded below by $\min \left\{ \lambda_1, \frac{\tau}{L} \right\}$. Hence, $\lim_{n \rightarrow \infty} \lambda_n > 0$.

Remark 3.7. If $\varphi(\cdot) = \frac{1}{2} \|\cdot\|_2^2$, then Algorithm 3.1 takes the form of the extragradient method with simple monotone step-size adjustment:

$$\begin{aligned} y_n &= P_C(x_n - \lambda_n A x_n), \\ x_{n+1} &= P_C(x_n - \lambda_n A y_n), \\ \lambda_{n+1} &= \begin{cases} \min \left\{ \lambda_n, \tau \frac{\|x_n - y_n\|_2}{\|A x_n - A y_n\|_2} \right\}, & \text{if } A x_n \neq A y_n, \\ \lambda_n, & \text{otherwise,} \end{cases} \end{aligned}$$

where $\tau \in (0, 1)$, $\lambda_1 > 0$. This adjustment was proposed in [111] for Tseng's algorithm [95].

If for some $n \in \mathbb{N}$ in Algorithm 3.1 we have $y_n = x_n$, then $x_n \in S$. Indeed, in this case $y_n = P_{x_n}^C(-\lambda_n A x_n)$. From inequality (3.6) it follows that

$$(A x_n, y - x_n) + \frac{(\nabla \varphi(x_n) - \nabla \varphi(x_n), y - x_n)}{\lambda_n} = (A x_n, y - x_n) \geq 0 \quad \forall y \in C,$$

that is, $x_n \in S$.

Assume that for all $n \in \mathbb{N}$ the stopping condition in Step 2 is not satisfied, and proceed to justify the convergence of the proposed algorithm.

3.3 Proof of Convergence

A key role in the proof is played by the inequality with respect to the Bregman divergence between the approximations computed by Algorithm 3.1 and some element of the solution set S of problem (1.1).

Lemma 3.1. Let $(x_n), (y_n)$ be the sequences obtained by Algorithm 3.1. Then the following inequality holds:

$$V(z, x_{n+1}) \leq V(z, x_n) - \left(1 - \frac{\mu_n}{\sigma}\right) \cdot V(y_n, x_n) - \left(1 - \frac{\mu_n}{\sigma}\right) \cdot V(x_{n+1}, y_n),$$

where $z \in S$, $\mu_n = \tau(\lambda_n/\lambda_{n+1})$.

Proof. Let $z \in S$. Recall the important equality (3.3):

$$\begin{aligned} V(z, x_{n+1}) &= V(z, x_n) - V(x_{n+1}, x_n) + \\ &\quad + (\nabla \varphi(x_{n+1}) - \nabla \varphi(x_n), x_{n+1} - z). \end{aligned} \quad (3.7)$$

From the definition of the points x_{n+1} and (3.6) it follows that

$$\lambda_n (A y_n, z - x_{n+1}) + (\nabla \varphi(x_{n+1}) - \nabla \varphi(x_n), z - x_{n+1}) \geq 0.$$

Using this inequality to estimate the inner product in (3.7), we obtain

$$V(z, x_{n+1}) \leq V(z, x_n) - V(x_{n+1}, x_n) + \lambda_n (A y_n, z - x_{n+1}).$$

Taking into account that

$$\begin{aligned} V(x_{n+1}, x_n) &= V(x_{n+1}, y_n) + V(y_n, x_n) + \\ &\quad + (\nabla \varphi(y_n) - \nabla \varphi(x_n), x_{n+1} - y_n), \end{aligned}$$

we obtain

$$\begin{aligned} V(z, x_{n+1}) &\leq V(z, x_n) - V(x_{n+1}, y_n) - V(y_n, x_n) + \\ &\quad + (\nabla \varphi(x_n) - \lambda_n A y_n - \nabla \varphi(y_n), x_{n+1} - y_n) + \lambda_n (A y_n, z - y_n). \end{aligned}$$

From the pseudomonotonicity of the operator A it follows that

$$(Ay_n, z - y_n) \leq 0.$$

Thus,

$$\begin{aligned} V(z, x_{n+1}) &\leq V(z, x_n) - V(x_{n+1}, y_n) - V(y_n, x_n) + \\ &\quad + (\nabla\varphi(x_n) - \lambda_n Ay_n - \nabla\varphi(y_n), x_{n+1} - y_n). \end{aligned}$$

Since $x_{n+1} \in C$, we have

$$\begin{aligned} &(\nabla\varphi(x_n) - \lambda_n Ay_n - \nabla\varphi(y_n), x_{n+1} - y_n) \\ &= \underbrace{(\nabla\varphi(x_n) - \lambda_n Ax_n - \nabla\varphi(y_n), x_{n+1} - y_n)}_{\leq 0} + \\ &\quad + \lambda_n (Ax_n - Ay_n, x_{n+1} - y_n) \leq \lambda_n (Ax_n - Ay_n, x_{n+1} - y_n). \end{aligned}$$

Taking this into account in the previous inequality, we arrive at

$$V(z, x_{n+1}) \leq V(z, x_n) - V(x_{n+1}, y_n) - V(y_n, x_n) + \lambda_n (Ax_n - Ay_n, x_{n+1} - y_n).$$

Now let us estimate the term $\lambda_n (Ax_n - Ay_n, x_{n+1} - y_n)$ using inequality (3.4). We have

$$\begin{aligned} \lambda_n (Ax_n - Ay_n, x_{n+1} - y_n) &\leq \lambda_n \|Ax_n - Ay_n\|_* \|x_{n+1} - y_n\| \leq \\ &\leq \frac{2}{\sigma} \frac{\lambda_n}{\lambda_{n+1}} \tau \sqrt{V(y_n, x_n)} \sqrt{V(x_{n+1}, y_n)} \leq \\ &\leq \frac{\mu_n}{\sigma} V(y_n, x_n) + \frac{\mu_n}{\sigma} V(x_{n+1}, y_n). \end{aligned}$$

Applying this to the previous inequality, we obtain

$$\begin{aligned} V(z, x_{n+1}) &\leq V(z, x_n) - V(x_{n+1}, y_n) - V(y_n, x_n) + \\ &\quad + \frac{\mu_n}{\sigma} V(y_n, x_n) + \frac{\mu_n}{\sigma} V(x_{n+1}, y_n) = \\ &= V(z, x_n) - \left(1 - \frac{\mu_n}{\sigma}\right) V(x_{n+1}, y_n) - \left(1 - \frac{\mu_n}{\sigma}\right) V(y_n, x_n), \end{aligned}$$

which completes the proof of Lemma 3.1.

To prove the convergence of the method, we will need an elementary lemma related to numerical sequences.

Lemma 3.2. Let $(a_n), (b_n)$ be two sequences of nonnegative numbers that satisfy the inequality

$$a_{n+1} \leq a_n - b_n \quad \text{for all } n \in \mathbb{N}.$$

Then the limit $\lim_{n \rightarrow \infty} a_n$ exists and $\sum_n b_n < +\infty$.

We now formulate the main theorem of this chapter, which characterizes the convergence of Algorithm 3.1.

Theorem 3.1. Let the set $C \subseteq E$ be convex and closed, and let the operator $A : E \rightarrow E^*$ be pseudomonotone and Lipschitz continuous with constant $L > 0$. Assume that the solution set of the problem is nonempty, $S \neq \emptyset$. Then the sequences (x_n) and (y_n) generated by Algorithm 3.1 converge to a solution $\bar{z} \in S$.

Proof. Let $z \in S$. Define

$$\begin{aligned} a_n &= V(z, x_n), \\ b_n &= \left(1 - \frac{\mu_n}{\sigma}\right) V(x_{n+1}, y_n) + \left(1 - \frac{\mu_n}{\sigma}\right) V(y_n, x_n). \end{aligned}$$

Then the inequality from Lemma 3.1 becomes

$$a_{n+1} \leq a_n - b_n.$$

The asymptotic relation

$$1 - \mu_n/\sigma = 1 - (\tau/\sigma)(\lambda_n/\lambda_{n+1}) \rightarrow 1 - \tau/\sigma \in (0, 1) \quad \text{as } n \rightarrow \infty,$$

together with Lemma 3.2 gives the existence of the limit

$$\lim_{n \rightarrow \infty} V(z, x_n)$$

and the boundedness

$$\sum_{n=1}^{\infty} (V(x_{n+1}, y_n) + V(y_n, x_n)) < +\infty.$$

This implies that

$$\lim_{n \rightarrow \infty} V(x_{n+1}, y_n) = \lim_{n \rightarrow \infty} V(y_n, x_n) = 0.$$

From the fact that these limits are zero we obtain

$$\lim_{n \rightarrow \infty} \|x_{n+1} - y_n\| = \lim_{n \rightarrow \infty} \|y_n - x_n\| = 0. \quad (3.8)$$

Finally, from this and the inequality

$$V(z, x_n) \geq \frac{\sigma}{2} \|z - x_n\|^2$$

we can conclude that the sequences $(x_n), (y_n)$ are bounded.

Now consider a subsequence (y_{n_k}) that converges to some point $\bar{z} \in C$. Then from (3.8) it follows that $x_{n_k} \rightarrow \bar{z}$. Let us show that $\bar{z} \in S$. We have

$$(Ax_{n_k}, y - y_{n_k}) + \frac{(\nabla\varphi(y_{n_k}) - \nabla\varphi(x_{n_k}), y - y_{n_k})}{\lambda_{n_k}} \geq 0 \quad \forall y \in C. \quad (3.9)$$

Passing to the limit in (3.9) taking into account (3.8), we obtain

$$(A\bar{z}, y - \bar{z}) \geq 0 \quad \forall y \in C,$$

that is, $\bar{z} \in S$.

Let us show that $x_n \rightarrow \bar{z}$ and $y_n \rightarrow \bar{z}$. It is known that the finite limit

$$\lim_{n \rightarrow \infty} V(\bar{z}, x_n) = \lim_{n \rightarrow \infty} (\varphi(\bar{z}) - \varphi(x_n) - (\nabla \varphi(x_n), \bar{z} - x_n))$$

exists. Since $\lim_{n \rightarrow \infty} V(\bar{z}, x_{n_k}) = 0$, then also $\lim_{n \rightarrow \infty} V(\bar{z}, x_n) = 0$. Hence,

$$\|x_n - \bar{z}\| \rightarrow 0.$$

From (3.8) it follows that $\|y_n - \bar{z}\| \rightarrow 0$ as well.

Remark 3.8. As can be seen from the proof of Theorem 3.1, for the sequence (x_n) , starting from some index N , the Fejér property in Bregman divergence holds with respect to the set S .

3.4 Adaptive Variant of the Extrapolation from the Past Algorithm

The following algorithm is also based on the use of Bregman divergence. Let the function φ again satisfy conditions 1–3 given above, and, moreover, be strongly convex with strong convexity constant 1. That is,

$$\varphi(a) \geq \varphi(b) + (\nabla \varphi(b), a - b) + \frac{1}{2} \|a - b\|^2 \quad \forall a \in \text{dom } \varphi, \quad b \in \text{int dom } \varphi.$$

The Bregman divergence has the form (3.2). Taking into account the three-point identity (3.3), the inequality with respect to the solution (3.6) can be written in terms of the Kullback–Leibler divergence as follows:

$$-(a, y - z) + (\nabla \varphi(z) - \nabla \varphi(x), y - z) \geq 0 \quad \forall y \in K.$$

Further we will assume that $C \subseteq \text{int dom } \varphi$.

In [88], for solving problem (1.1), the following algorithm was proposed:

$$\begin{cases} y_n = P_{x_n}^C(-\lambda A y_{n-1}), \\ x_{n+1} = P_{x_n}^C(-\lambda A y_n), \end{cases}$$

where the step λ had to satisfy the inequality $\lambda \in (0, (\sqrt{2} - 1)\frac{1}{L})$. As a result, to use this algorithm it was necessary to know the Lipschitz constant of the operator A . Also, in [88] the convergence of the sequences (x_n) and (y_n) generated by the algorithm to an element of the solution set of the variational inequality (1.1) was shown. Moreover, if the operator A is monotone and the feasible set $C \subseteq E$ is compact, the authors obtained an estimate for the gap function:

$$G(z_N) = \max_{y \in C} (Ay, z_N - y) \leq \frac{\lambda^{-1} R_C(x_1) + LV(x_1, y_0)}{N}, \quad (3.10)$$

where $z_N = \frac{1}{N} \sum_{n=1}^N y_n$ is the arithmetic mean of the elements of the sequence generated by the algorithm, and $R_C(x_1) = \max_{y \in C} V(y, x_1)$ [80].

Building upon the iterative scheme of this algorithm and the works [31, 29], we construct a two-step Bregman algorithm with adaptive step-size selection, which does not require knowledge of the Lipschitz constants and does not involve evaluating the operator A at additional points (that is, without line search procedures).

Thus, for solving the variational inequality (1.1), the following algorithm is proposed.

Algorithm 3.2. “Extrapolation from the Past” with Adaptive Step-Size Adjustment.

Initialization. Let $y_0 \in E$, $x_1 \in \text{int dom } \varphi$, $\tau \in (0, \frac{1}{3})$, and $\lambda_1 > 0$. Initially, set $n = 1$.

Step 1.

$$y_n = P_{x_n}^C(-\lambda_n A y_{n-1})$$

Step 2.

$$x_{n+1} = P_{x_n}^C(-\lambda_n A y_n).$$

If $y_n = x_n = x_{n+1}$, then the algorithm is terminated. Otherwise, proceed to the next step.

Step 3. Adaptive step-size adjustment.

$$\lambda_{n+1} = \begin{cases} \min \left\{ \lambda_n, \tau \frac{\|y_n - y_{n-1}\|}{\|A y_n - A y_{n-1}\|_*} \right\}, & \text{if } A y_{n-1} \neq A y_n, \\ \lambda_n, & \text{otherwise.} \end{cases} \quad (3.11)$$

The iteration of the algorithm is completed. Set $n := n + 1$ and return to Step 1.

Remark 3.9. If we choose $\varphi(\cdot) = \frac{1}{2} \|\cdot\|_2^2$, then Algorithm 3.2 takes the form:

$$\begin{aligned} y_n &= P_C(x_n - \lambda_n A y_{n-1}), \\ x_{n+1} &= P_C(x_n - \lambda_n A y_n), \\ \lambda_{n+1} &= \begin{cases} \min \left\{ \lambda_n, \tau \frac{\|y_n - y_{n-1}\|_2}{\|A y_n - A y_{n-1}\|_2} \right\}, & \text{if } A y_{n-1} \neq A y_n, \\ \lambda_n, & \text{otherwise,} \end{cases} \end{aligned}$$

where $\tau \in (0, \frac{1}{3})$, $\lambda_1 > 0$.

Remark 3.10. Let the feasible set be the standard simplex, that is,

$$S_m = \left\{ x \in \mathbb{R}^m : x_i \geq 0, \sum_{i=1}^m x_i = 1 \right\}.$$

Consider the problem of finding a solution of the variational inequality:

$$\text{find } x \in S_m : (Ax, y - x) \geq 0 \quad \forall y \in S_m.$$

Exactly such a model arises in some important problems of network economics, in particular in the problem of finding the importance of nodes of a graph (network) using the PageRank algorithm,

which is described in more detail in the last chapter. If instead of the Euclidean distance we take the Kullback–Leibler divergence, then the projection operation of the updated approximation point in Algorithm 3.2 can be written in closed form [10]. Accordingly, the algorithm takes the following form:

$$\begin{aligned} y_i^n &= \frac{x_i^n \exp(-\lambda_n (Ay_{n-1})_i)}{\sum_{j=1}^m x_j^n \exp(-\lambda_n (Ay_{n-1})_j)}, \quad i = 1, \dots, m, \\ x_i^{n+1} &= \frac{x_i^n \exp(-\lambda_n (Ay_n)_i)}{\sum_{j=1}^m x_j^n \exp(-\lambda_n (Ay_n)_j)}, \quad i = 1, \dots, m, \\ \lambda_{n+1} &= \begin{cases} \min \left\{ \lambda_n, \tau \frac{\sum_{j=1}^m |y_j^n - y_j^{n-1}|}{\max_{j=1, \dots, m} |(Ay_n)_j - (Ay_{n-1})_j|} \right\}, & \text{if } Ay_{n-1} \neq Ay_n, \\ \lambda_n, & \text{otherwise,} \end{cases} \end{aligned} \quad (3.12)$$

where $(Ay_n)_i \in \mathbb{R}$ is the i -th coordinate of the vector $Ay_n \in \mathbb{R}^m$.

The sequence (λ_n) obtained by procedure (3.11) is non-increasing. Moreover, $\lambda_n \geq \min \left\{ \lambda_1, \frac{\tau}{L} \right\}$.

Thus, there exists $\lim_{n \rightarrow \infty} \lambda_n > 0$.

The following inequalities with respect to the sequences of approximations $(x_n), (y_n)$ generated by Algorithm 3.2 are important for proving the main convergence theorem:

$$-\lambda_n (Ay_{n-1}, y - y_n) \leq V(y, x_n) - V(y_n, x_n) - V(y, y_n) \quad \forall y \in C. \quad (3.13)$$

$$-\lambda_n (Ay_n, y - x_{n+1}) \leq V(y, x_n) - V(x_{n+1}, x_n) - V(y, x_{n+1}) \quad \forall y \in C. \quad (3.14)$$

The latter inequality allows us to obtain a stopping rule for Algorithm 3.2. Indeed, if $x_{n+1} = x_n = y_n$, then from (3.14) we get

$$(Ax_n, y - x_n) \geq 0 \quad \forall y \in C,$$

or equivalently, $x_n \in S$.

Let us prove the convergence of Algorithm 3.2.

First, we prove the main estimate that relates the distances between the points generated by Algorithm 3.2 and an arbitrary element of the solution set S .

Lemma 3.3. For the sequences $(x_n), (y_n)$ generated by Algorithm 3.2, the following inequality holds:

$$\begin{aligned} V(z, x_{n+1}) &\leq V(z, x_n) - \left(1 - 2\tau \frac{\lambda_n}{\lambda_{n+1}}\right) V(x_{n+1}, y_n) - \\ &\quad - \left(1 - \tau \frac{\lambda_n}{\lambda_{n+1}}\right) V(y_n, x_n) + \tau \frac{\lambda_n}{\lambda_{n+1}} V(x_n, y_{n-1}), \end{aligned} \quad (3.15)$$

where $z \in S$.

Proof. Let $z \in S$. Since the operator A is pseudomonotone, we have

$$(Ay_n, z - y_n) \leq 0. \quad (3.16)$$

From (3.14) it follows that

$$V(z, x_{n+1}) \leq V(z, x_n) - V(x_{n+1}, x_n) + \lambda_n (Ay_n, z - x_{n+1}).$$

To the right-hand side of this inequality we add the term $\lambda_n (Ay_n, y_n - z) > 0$. We obtain

$$V(z, x_{n+1}) \leq V(z, x_n) - V(x_{n+1}, x_n) + \lambda_n (Ay_n, y_n - x_{n+1}). \quad (3.17)$$

From (3.13) we get

$$-\lambda_n (Ay_{n-1}, x_{n+1} - y_n) \leq V(x_{n+1}, x_n) - V(y_n, x_n) - V(x_{n+1}, y_n). \quad (3.18)$$

Using the last inequality, we can obtain an estimate for $-V(x_{n+1}, x_n)$. Substituting it into (3.17), we get

$$\begin{aligned} V(z, x_{n+1}) &\leq V(z, x_n) - V(x_{n+1}, y_n) - V(y_n, x_n) + \\ &\quad + \lambda_n (Ay_n - Ay_{n-1}, y_n - x_{n+1}). \end{aligned} \quad (3.19)$$

Now, using the rule for computing λ_{n+1} , let us estimate from above the term $\lambda_n (Ay_n - Ay_{n-1}, y_n - x_{n+1})$. We have

$$\begin{aligned} \lambda_n (Ay_n - Ay_{n-1}, y_n - x_{n+1}) &\leq \lambda_n \|Ay_n - Ay_{n-1}\|_* \|y_n - x_{n+1}\| \leq \\ &\leq \tau \frac{\lambda_n}{\lambda_{n+1}} \|y_n - y_{n-1}\| \|x_{n+1} - y_n\| \leq \\ &\leq \tau \frac{\lambda_n}{\lambda_{n+1}} (\|y_n - x_n\| + \|x_n - y_{n-1}\|) \|x_{n+1} - y_n\| \leq \\ &\leq \tau \frac{\lambda_n}{2\lambda_{n+1}} \|y_n - x_n\|^2 + \tau \frac{\lambda_n}{2\lambda_{n+1}} \|x_n - y_{n-1}\|^2 + \tau \frac{\lambda_n}{\lambda_{n+1}} \|x_{n+1} - y_n\|^2 \leq \\ &\leq \tau \frac{\lambda_n}{\lambda_{n+1}} V(y_n, x_n) + \tau \frac{\lambda_n}{\lambda_{n+1}} V(x_n, y_{n-1}) + 2\tau \frac{\lambda_n}{\lambda_{n+1}} V(x_{n+1}, y_n). \end{aligned}$$

Using this inequality in (3.19), we obtain

$$\begin{aligned} V(z, x_{n+1}) &\leq V(z, x_n) - V(x_{n+1}, y_n) - V(y_n, x_n) + \\ &\quad + \tau \frac{\lambda_n}{\lambda_{n+1}} V(y_n, x_n) + \tau \frac{\lambda_n}{\lambda_{n+1}} V(x_n, y_{n-1}) + 2\tau \frac{\lambda_n}{\lambda_{n+1}} V(x_{n+1}, y_n), \end{aligned}$$

which is what we needed to prove.

For the further proof we again use Lemma 3.2 of this chapter on the properties of nonnegative numerical sequences.

The main convergence result for Algorithm 3.2 is given by the following theorem.

Theorem 3.2. Let the set $C \subseteq E$ be convex and closed, the operator $A : E \rightarrow E^*$ be pseudomonotone and Lipschitz continuous with constant $L > 0$, and let $S \neq \emptyset$. Then the sequences (x_n) and (y_n) generated by Algorithm 3.2 converge to some point $\bar{z} \in S$.

Proof. Let $z' \in S$. Define

$$\begin{aligned} a_n &= V(z', x_n) + \tau \frac{\lambda_n}{\lambda_{n+1}} V(x_n, y_{n-1}), \\ b_n &= \left(1 - \tau \frac{\lambda_n}{\lambda_{n+1}}\right) V(y_n, x_n) + \left(1 - \tau \frac{\lambda_{n+1}}{\lambda_{n+2}} - 2\tau \frac{\lambda_n}{\lambda_{n+1}}\right) V(x_{n+1}, y_n). \end{aligned}$$

Inequality (3.15) takes the form

$$a_{n+1} \leq a_n - b_n.$$

Since $\lim_{n \rightarrow \infty} \lambda_n > 0$ exists, we have

$$1 - \tau \frac{\lambda_n}{\lambda_{n+1}} \rightarrow 1 - \tau \in (0, 1), \quad \text{and} \quad 1 - \tau \frac{\lambda_{n+1}}{\lambda_{n+2}} - 2\tau \frac{\lambda_n}{\lambda_{n+1}} \rightarrow 1 - 3\tau \in (0, 1) \quad \text{as } n \rightarrow \infty.$$

From Lemma 3.2 on numerical sequences we can conclude that the limit

$$\lim_{n \rightarrow \infty} \left(V(z', x_n) + \tau \frac{\lambda_n}{\lambda_{n+1}} V(x_n, y_{n-1}) \right)$$

exists, and also

$$\sum_{n=1}^{\infty} \left(\left(1 - \tau \frac{\lambda_n}{\lambda_{n+1}}\right) V(y_n, x_n) + \left(1 - \tau \frac{\lambda_{n+1}}{\lambda_{n+2}} - 2\tau \frac{\lambda_n}{\lambda_{n+1}}\right) V(x_{n+1}, y_n) \right) < +\infty.$$

From this we obtain

$$\lim_{n \rightarrow \infty} V(y_n, x_n) = \lim_{n \rightarrow \infty} V(x_{n+1}, y_n) = 0 \quad (3.20)$$

and the convergence of the numerical sequences $(V(z', x_n))$ for all $z' \in S$. From the latter double equality it follows that

$$\lim_{n \rightarrow \infty} \|y_n - x_n\| = \lim_{n \rightarrow \infty} \|x_{n+1} - y_n\| = 0. \quad (3.21)$$

From the same equality together with the inequality

$$V(z', x_n) \geq \frac{1}{2} \|z' - x_n\|^2$$

we obtain that the sequences $(x_n), (y_n)$ are bounded. Moreover, from inequalities (3.17), (3.18) and the limit relations (3.20) it follows that

$$\lim_{n \rightarrow \infty} V(x_{n+1}, x_n) = 0$$

and, obviously,

$$\lim_{n \rightarrow \infty} \|x_{n+1} - x_n\| = 0. \quad (3.22)$$

Consider a subsequence (x_{n_k}) that converges to some point $\bar{z} \in E$. Clearly, $\bar{z} \in C$. From the limit relations (3.21) it follows that $y_{n_k} \rightarrow \bar{z}$ and $x_{n_k+1} \rightarrow \bar{z}$. Let us show that $\bar{z} \in S$. We have

$$(Ay_{n_k}, y - x_{n_k+1}) + \frac{1}{\lambda_{n_k+1}} (\nabla \varphi(x_{n_k+1}) - \nabla \varphi(x_{n_k}), y - x_{n_k+1}) \geq 0 \quad \forall y \in C.$$

Passing to the limit in this inequality taking into account (3.22), we obtain

$$(A\bar{z}, y - \bar{z}) \geq 0 \quad \forall y \in C,$$

that is, $\bar{z} \in C$.

Let us show that $x_n \rightarrow \bar{z}$ (then from $\|x_n - y_n\| \rightarrow 0$ it will follow that $y_n \rightarrow \bar{z}$ as well). It is known that the limit

$$\lim_{n \rightarrow \infty} V(\bar{z}, x_n) = \lim_{n \rightarrow \infty} (\varphi(\bar{z}) - \varphi(x_n) - (\nabla \varphi(x_n), \bar{z} - x_n))$$

exists. Since $\lim_{n \rightarrow \infty} V(\bar{z}, x_{n_k}) = 0$, then also $\lim_{n \rightarrow \infty} V(\bar{z}, x_n) = 0$. Hence,

$$\|x_n - \bar{z}\| \rightarrow 0.$$

3.5 Summary of Chapter 3

In this chapter, two algorithms that use Bregman divergence for projection instead of the Euclidean distance were proposed and their convergence was studied. The first algorithm can be regarded as a Bregman modification of the extragradient method with the addition of an economical step-size adjustment. Conceptually, it is a continuation of the works [88, 89, 19, 80], while the proposed step-size computation rule does not require evaluating the operator and the projection at additional points — that is, it is more efficient than the corresponding procedures in [53, 28, 105, 78]. The second algorithm is a variant of the two-step Bregman method [88], known as “Extrapolation from the Past” (see also [31, 29]), with a step-size rule similar to Algorithm 3.1 but without the line search procedure. For both algorithms, convergence theorems were formulated and proved under the assumption that the operator in (1.1) is pseudomonotone and Lipschitz continuous.

The results of this chapter were published in [91, 29, 33].

Chapter 4

Extrapolation from the Past Method for Variational Inequalities in Uniformly Convex Banach Spaces

This chapter considers algorithms for finding a solution of a variational inequality in a uniformly convex Banach space. The algorithms themselves are based on the method originally proposed in [82], which has recently become widely used and received the name “extrapolation from the past algorithm.” For example, an application of this algorithm to machine learning problems can be found in [45]. In the proposed algorithms, an alternative projection (different from the metric projection) is used.

The first method is stationary and is constructed by replacing the standard Euclidean projection in the extrapolation from the past algorithm with the generalized Alber projection. The second method, based on the first one, uses an adaptive procedure for recalculating the step-size at each iteration, which allows avoiding the requirement of knowing the Lipschitz constant in advance. An advantage of the proposed rule is that it does not involve computing the operator at additional points compared to the main iteration step of the algorithm. The chapter also studies the convergence conditions of the proposed algorithms and proves the corresponding theorems.

4.1 Preliminaries and Problem Statement

To obtain the main results of the chapter, we will use notation and some results related to the geometry of Banach spaces, which are given below. For a detailed description of the corresponding theory, one can refer to [2, 1, 37, 9, 4, 110].

Let E be a real Banach space with norm $\|\cdot\|$, and let E^* be the dual space to E . For $x^* \in E^*$ and $x \in E$, denote by $\langle x^*, x \rangle$ the value of the functional x^* at the element x . The norm in E^* will be denoted by $\|\cdot\|_*$.

Let $S_E = \{x \in E : \|x\| = 1\}$. A Banach space E is called strictly convex if for all $x, y \in S_E$ with $x \neq y$ we have $\left\|\frac{x+y}{2}\right\| < 1$. The modulus of convexity of the space E is defined as

$$\delta_E(\varepsilon) = \inf \left\{ 1 - \left\| \frac{x+y}{2} \right\| : x, y \in S_E, \|x-y\| = \varepsilon \right\} \quad \forall \varepsilon \in (0, 2].$$

A Banach space E is called uniformly convex if $\delta_E(\varepsilon) > 0$ for all $\varepsilon \in (0, 2]$. A Banach space E is called 2-uniformly convex if there exists $c > 0$ such that $\delta_E(\varepsilon) \geq c\varepsilon^2$ for all $\varepsilon \in (0, 2]$. Clearly, every 2-uniformly convex space is uniformly convex. It is known that a uniformly convex Banach space is reflexive [37].

A Banach space E is called smooth if the limit

$$\lim_{t \rightarrow 0} \frac{\|x + ty\| - \|x\|}{t}$$

exists for all $x, y \in S_E$. A Banach space E is called uniformly smooth if this limit exists uniformly for $x, y \in S_E$. There is a duality between the convexity and smoothness of a Banach space E and its dual E^* [9, 4]:

- if E^* is strictly convex $\Rightarrow E$ is smooth;
- if E^* is smooth $\Rightarrow E$ is strictly convex;
- E is uniformly convex $\Leftrightarrow E^*$ is uniformly smooth;
- E is uniformly smooth $\Leftrightarrow E^*$ is uniformly convex.

Note that under the assumption of reflexivity of E , the first two implications can be reversed.

It is known that Hilbert spaces and L_p spaces ($1 < p \leq 2$) are 2-uniformly convex and uniformly smooth (the L_p spaces are uniformly smooth for $p \in (1, \infty)$) [37, 9].

A multivalued operator $J : E \rightarrow 2^{E^*}$, defined by

$$Jx = \{x^* \in E^* : \langle x^*, x \rangle = \|x\|^2 = \|x^*\|_*^2\},$$

is called the normalized duality mapping. It is known [9] that:

- if the space E is smooth, then the mapping J is single-valued;
- if the space E is strictly convex, then the mapping J is injective and strictly monotone;
- if the space E is reflexive, then the mapping J is surjective;
- if the space E is uniformly smooth, then the mapping J is uniformly continuous on bounded subsets of E .

Let E be a smooth Banach space. Consider the functional introduced by Ya. Alber [1]:

$$\phi(x, y) = \|x\|^2 - 2\langle Jy, x \rangle + \|y\|^2 \quad \forall x, y \in E.$$

From the definition of ϕ follows a useful three-point identity:

$$\phi(x, y) - \phi(x, z) - \phi(z, y) = 2\langle Jz - Jy, x - z \rangle \quad \forall x, y, z \in E.$$

If the space E is strictly convex, then for $x, y \in E$ we have $\phi(x, y) = 0 \Leftrightarrow x = y$.

Lemma 4.1 ([1]). Let E be a uniformly convex and uniformly smooth Banach space, and let $(x_n), (y_n)$ be bounded sequences of elements of E . Then

$$\|x_n - y_n\| \rightarrow 0 \Leftrightarrow \|Jx_n - Jy_n\|_* \rightarrow 0 \Leftrightarrow \phi(x_n, y_n) \rightarrow 0.$$

Lemma 4.2 ([110]). Let E be a 2-uniformly convex and smooth Banach space. Then for some $\mu \geq 1$ the inequality holds

$$\phi(x, y) \geq \frac{1}{\mu} \|x - y\|^2 \quad \forall x, y \in E.$$

For Banach spaces ℓ_p, L_p and W_p^m ($1 < p \leq 2$) we have $\mu = \frac{1}{p-1}$ [110]. For Hilbert space, the inequality of Lemma 4.2 becomes an identity with $\mu = 1$.

Let K be a nonempty closed and convex subset of a reflexive, strictly convex, and smooth space E . It is known [1] that for each $x \in E$ there exists a unique element $z \in K$ such that $\phi(z, x) = \inf_{y \in K} \phi(y, x)$. This element z is denoted $\Pi_K x$, and the corresponding operator $\Pi_K : E \rightarrow K$ is called the generalized projection of E onto K (Alber's generalized projection) [1]. Note that if E is a Hilbert space, then Π_K coincides with the metric projection onto the set K .

Lemma 4.3 ([1]). Let K be a closed and convex subset of a reflexive, strictly convex, and smooth space E , with $x \in E, z \in K$. Then

$$z = \Pi_K x \quad \Leftrightarrow \quad \langle Jz - Jx, y - z \rangle \geq 0 \quad \forall y \in K.$$

The inequality of Lemma 4.3 is equivalent to the following inequality [1]:

$$\phi(y, \Pi_K x) + \phi(\Pi_K x, x) \leq \phi(y, x) \quad \forall y \in K.$$

Remark 4.1. The main element of the algorithms considered below is the computation, for given $x \in E$ and $x^* \in E^*$, of a new point

$$x^+ = \Pi_K J^{-1}(Jx - x^*).$$

From Lemma 4.3 and the three-point identity follows the fundamental inequality for the analysis of algorithms:

$$\phi(y, x^+) \leq \phi(y, x) - \phi(x^+, x) + 2\langle x^*, y - x^+ \rangle \quad \forall y \in K.$$

Consider the variational inequality (1.1), where C is a nonempty subset of a 2-uniformly convex and uniformly smooth Banach space E , and A is an operator acting from E into E^* . Denote the solution set of (1.1) by S .

Assume the following conditions hold:

- the set $C \subseteq E$ is convex and closed;
- the operator $A : E \rightarrow E^*$ is monotone and Lipschitz continuous on C (with constant $L > 0$);
- the set S is nonempty.

Remark 4.2. The variational inequality (1.1) can be reformulated as a fixed-point problem [1]:

$$x = \Pi_C J^{-1}(Jx - \lambda Ax), \quad (4.1)$$

where $\lambda > 0$. From this formulation, the idea of constructing an algorithm arises naturally. Indeed, the solution can be obtained by the iteration

$$x_{n+1} = \Pi_C J^{-1}(Jx_n - \lambda Ax_n),$$

which, in particular, was studied in [51] for inverse strongly monotone operators $A : E \rightarrow E^*$. However, without such rather restrictive assumptions on the operator, this scheme may fail to converge. In particular, monotonicity and Lipschitz continuity of the operator are not sufficient for the convergence of the method.

Consider the dual variational inequality:

$$\text{find } x \in C : \langle Ay, x - y \rangle \leq 0 \quad \forall y \in C. \quad (4.2)$$

Denote the solution set of (4.2) by S^d . Note that the set S^d is convex and closed [54]. Inequality (4.2) is sometimes called the weak or dual formulation of (1.1) (or Minty's inequality), and the solutions of (4.2) are called weak solutions of (1.1). For monotone operators A we always have $S \subseteq S^d$. Under our assumptions, $S^d = S$ [54].

One of the main tasks is to estimate the number of algorithm iterations required to obtain an approximate solution of a given accuracy. The quality of an approximate solution $x \in C$ of the variational inequality (1.1) will be evaluated using the so-called gap function [78]:

$$\text{gap}(x) = \sup_{y \in C} \langle Ay, x - y \rangle \geq 0. \quad (4.3)$$

Note that for the correctness of the definition of the gap function (4.3), boundedness of the feasible set C is required.

Lemma 4.4. Let the operator A be monotone. If $x \in C$ is a solution of (1.1), then $\text{gap}(x) = 0$. Conversely, if for $x \in C$ we have $\text{gap}(x) = 0$, then x is a solution of (1.1).

4.2 Extrapolation from the Past Algorithm

To solve the variational inequality (1.1), we propose the following algorithm.

Algorithm 4.1. Extrapolation from the Past.

Initialization. Choose $x_1 = y_0 \in E$, $\lambda_n > 0$. Set $n = 1$.

Step 1.

$$y_n = \Pi_C J^{-1} (Jx_n - \lambda_n A y_{n-1}).$$

Step 2.

$$x_{n+1} = \Pi_C J^{-1} (Jx_n - \lambda_n A y_n),$$

if $x_{n+1} = y_n = x_n$, then the algorithm terminates. Otherwise, set $n := n + 1$ and return to Step 1.

Remark 4.3. The given algorithm is a modification of the algorithm from [82, 65] for problems in Banach spaces, which uses Alber's generalized projection [1] instead of the metric projection. The convergence of the algorithm in Hilbert space and in Euclidean space with Bregman divergence instead of the Euclidean distance was studied in the previous chapters.

The stopping rule is justified by the identity (4.1), which is equivalent to the variational inequality (1.1). Indeed, if $x_{n+1} = y_n = x_n$, we have $y_n = \Pi_C J^{-1} (Jy_n - \lambda_n A y_n)$, from which $y_n \in S$.

4.3 Convergence Proof

For proving the main theorem on convergence we need

Lemma 4.5. For the sequences (x_n) , (y_n) generated by Algorithm 4.1, the following inequality holds:

$$\begin{aligned} \phi(y, x_{n+1}) &\leq \phi(y, x_n) - (1 - 2\mu\lambda_n L) \phi(x_{n+1}, y_n) - (1 - \mu\lambda_n L) \phi(y_n, x_n) + \\ &+ \mu\lambda_n L \phi(x_n, y_{n-1}) + 2\lambda_n \langle Ay, y - y_n \rangle, \end{aligned} \quad (4.4)$$

where $y \in C$.

Proof. If some point $y \in C$, then for it we have

$$\phi(y, x_{n+1}) \leq \phi(y, x_n) - \phi(x_{n+1}, x_n) + 2\lambda_n \langle Ay_n, y - x_{n+1} \rangle.$$

Using the monotonicity property of A , we can show that

$$\begin{aligned} \langle Ay_n, y - x_{n+1} \rangle &= \langle Ay_n, y - y_n \rangle + \langle Ay_n, y_n - x_{n+1} \rangle \leq \\ &\leq \langle Ay, y - y_n \rangle + \langle Ay_n, y_n - x_{n+1} \rangle. \end{aligned}$$

And then

$$\begin{aligned}
\phi(y, x_{n+1}) &\leq \\
&\leq \phi(y, x_n) - \phi(x_{n+1}, x_n) + 2\lambda_n \langle Ay_n, y_n - x_{n+1} \rangle + 2\lambda_n \langle Ay, y - y_n \rangle = \\
&= \phi(y, x_n) - \phi(x_{n+1}, x_n) + 2\lambda_n \langle Ay_{n-1}, y_n - x_{n+1} \rangle + \\
&\quad + 2\lambda_n \langle Ay_n - Ay_{n-1}, y_n - x_{n+1} \rangle + 2\lambda_n \langle Ay, y - y_n \rangle. \quad (4.5)
\end{aligned}$$

The expression $2\lambda_n \langle Ay_{n-1}, y_n - x_{n+1} \rangle$ can be transformed as follows

$$\begin{aligned}
2\lambda_n \langle Ay_{n-1}, y_n - x_{n+1} \rangle &= \\
&= 2 \langle Jx_n - \lambda_n Ay_{n-1} - Jy_n, x_{n+1} - y_n \rangle + 2 \langle Jy_n - Jx_n, x_{n+1} - y_n \rangle.
\end{aligned}$$

Using Lemma 4.3 and the fact that $x_{n+1} \in C$, we obtain

$$\langle Jx_n - \lambda_n Ay_{n-1} - Jy_n, x_{n+1} - y_n \rangle \leq 0.$$

Then

$$\begin{aligned}
2\lambda_n \langle Ay_{n-1}, y_n - x_{n+1} \rangle &\leq 2 \langle Jy_n - Jx_n, x_{n+1} - y_n \rangle = \\
&= \phi(x_{n+1}, x_n) - \phi(x_{n+1}, y_n) - \phi(y_n, x_n). \quad (4.6)
\end{aligned}$$

Using (4.6), we can estimate the expression on the right-hand side of inequality (4.5). Then we have

$$\begin{aligned}
\phi(y, x_{n+1}) &\leq \phi(y, x_n) - \phi(x_{n+1}, y_n) - \phi(y_n, x_n) + \\
&\quad + 2\lambda_n \langle Ay_n - Ay_{n-1}, y_n - x_{n+1} \rangle + 2\lambda_n \langle Ay, y - y_n \rangle. \quad (4.7)
\end{aligned}$$

We need to estimate the term $2\lambda_n \langle Ay_n - Ay_{n-1}, y_n - x_{n+1} \rangle$ from this inequality. We obtain:

$$\begin{aligned}
2\lambda_n \langle Ay_n - Ay_{n-1}, y_n - x_{n+1} \rangle &\leq 2\lambda_n \|Ay_{n-1} - Ay_n\|_* \|x_{n+1} - y_n\| \leq \\
&\leq 2\lambda_n L \|y_{n-1} - y_n\| \|x_{n+1} - y_n\| \leq \\
&\leq 2\lambda_n L (\|y_{n-1} - x_n\| + \|x_n - y_n\|) \|x_{n+1} - y_n\| \leq \\
&\leq \lambda_n L (\|y_{n-1} - x_n\|^2 + \|x_n - y_n\|^2 + 2\|x_{n+1} - y_n\|^2). \quad (4.8)
\end{aligned}$$

Applying Lemma 4.2 to estimate the norms in the last inequality, we get

$$\begin{aligned}
2\lambda_n \langle Ay_n - Ay_{n-1}, y_n - x_{n+1} \rangle &\leq \mu\lambda_n L \phi(x_n, y_{n-1}) + \mu\lambda_n L \phi(y_n, x_n) + \\
&\quad + 2\mu\lambda_n L \phi(x_{n+1}, y_n).
\end{aligned}$$

In turn, substituting this into (4.7), we obtain

$$\phi(y, x_{n+1}) \leq \phi(y, x_n) - (1 - 2\mu\lambda_n L) \phi(x_{n+1}, y_n) -$$

$$- (1 - \mu\lambda_n L) \phi(y_n, x_n) + \mu\lambda_n L \phi(x_n, y_{n-1}) + 2\lambda_n \langle Ay, y - y_n \rangle,$$

— that is, inequality (4.4), which is the statement of the lemma.

If the feasible set of problem (1.1) is bounded and, accordingly, the function $\phi(x, y)$ is bounded on it, then one can prove a sublinear convergence rate of Algorithm 4.1 with respect to the gap function. Namely, we have the following.

Theorem 4.1. For the sequence of averaged (in the Cesàro sense) approximations obtained by Algorithm 4.1, of the form $z_N = \frac{\sum_{n=1}^N \lambda_n y_n}{\sum_{n=1}^N \lambda_n}$, the estimate holds

$$\text{gap}(z_N) \leq \frac{\sup_{y \in C} \phi(y, x_1) + \mu\lambda_1 L \phi(x_1, y_0)}{2 \sum_{n=1}^N \lambda_n},$$

under the condition that $\lambda_n \in \left(0, \frac{1}{3\mu L}\right]$.

Proof. Inequality (4.4) (the main statement of Lemma 4.5) can be written in the form:

$$\begin{aligned} 2\lambda_n \langle Ay, y - y_n \rangle &\leq (\phi(y, x_n) + \mu\lambda_n L \phi(x_n, y_{n-1})) - \\ &\quad - (\phi(y, x_{n+1}) + \mu\lambda_{n+1} L \phi(x_{n+1}, y_n)) - \\ &\quad - (1 - \mu L (\lambda_{n+1} + 2\lambda_n)) \phi(x_{n+1}, y_n) - \\ &\quad - (1 - \mu\lambda_n L) \phi(y_n, x_n). \end{aligned} \quad (4.9)$$

Summing such inequalities for $n = 1, \dots, N$, we obtain

$$2 \sum_{n=1}^N \lambda_n \langle Ay, y_n - y \rangle \leq \phi(y, x_1) + \mu\lambda_1 L \phi(x_1, y_0),$$

and further

$$\langle Ay, z_N - y \rangle \leq \frac{\phi(y, x_1) + \mu\lambda_1 L \phi(x_1, y_0)}{2 \sum_{n=1}^N \lambda_n}, \quad (4.10)$$

where $z_N = \frac{\sum_{n=1}^N \lambda_n y_n}{\sum_{n=1}^N \lambda_n}$. This gives an estimate for the gap function, which proves the theorem:

$$\text{gap}(z_N) \leq \frac{\sup_{y \in C} \phi(y, x_1) + \mu\lambda_1 L \phi(x_1, y_0)}{2 \sum_{n=1}^N \lambda_n}.$$

Corollary 4.1. Let $(x_n), (y_n)$ be the sequences generated by Algorithm 4.1 with $\lambda_n = \frac{1}{3\mu L}$. Then for the sequence of averages $z_N = \frac{1}{N} \sum_{n=1}^N y_n$ the inequality holds

$$\text{gap}(z_N) \leq \frac{\mu L^{\frac{3}{2}} \sup_{y \in C} \phi(y, x_1) + \mu L^{\frac{1}{2}} \phi(x_1, y_0)}{N}.$$

The question of the convergence conditions of Algorithm 4.1 is addressed by the following theorem.

Theorem 4.2. Let C be a nonempty closed convex subset of a 2-uniformly convex and uniformly smooth Banach space E , $A : E \rightarrow E^*$ a monotone and Lipschitz operator, and $S \neq \emptyset$. Assume

that the normalized duality mapping J is sequentially weakly continuous and the sequence (λ_n) satisfies $0 < \inf_n \lambda_n \leq \sup_n \lambda_n < \frac{1}{3\mu L}$. Then the sequences (x_n) and (y_n) generated by Algorithm 4.1 converge weakly to some point $z \in S$.

Proof. Substitute the element $y = z' \in S$ into inequality (1.1). Since $\langle Az', z' - y_n \rangle \leq 0$, we obtain

$$\begin{aligned} \phi(z', x_{n+1}) &\leq \phi(z', x_n) - (1 - 2\mu\lambda_n L) \phi(x_{n+1}, y_n) - \\ &\quad - (1 - \mu\lambda_n L) \phi(y_n, x_n) + \mu\lambda_n L \phi(x_n, y_{n-1}). \end{aligned} \quad (4.11)$$

Set

$$\begin{aligned} a_n &= \phi(z', x_n) + \mu\lambda_n L \phi(x_n, y_{n-1}), \\ b_n &= (1 - \mu\lambda_n L) \phi(y_n, x_n) + (1 - \mu\lambda_{n+1} L - 2\mu\lambda_n L) \phi(x_{n+1}, y_n). \end{aligned}$$

Inequality (4.11) takes the form $a_{n+1} \leq a_n - b_n$. From Lemma 3.2 it follows that the limit exists

$$\lim_{n \rightarrow \infty} (\phi(z', x_n) + \mu\lambda_n L \phi(x_n, y_{n-1}))$$

and that

$$\sum_{n=1}^{\infty} ((1 - \mu\lambda_n L) \phi(y_n, x_n) + (1 - \mu\lambda_{n+1} L - 2\mu\lambda_n L) \phi(x_{n+1}, y_n)) < +\infty.$$

This implies

$$\lim_{n \rightarrow \infty} \phi(y_n, x_n) = \lim_{n \rightarrow \infty} \phi(x_{n+1}, y_n) = 0$$

and the convergence of the numerical sequences $(\phi(z', x_n))$ for all $z' \in S$. From the equality to zero of these limits we have

$$\lim_{n \rightarrow \infty} \|y_n - x_n\| = \lim_{n \rightarrow \infty} \|x_{n+1} - y_n\| = 0. \quad (4.12)$$

Also, from Lemma 4.2 and (4.11) it follows that the sequences $(x_n), (y_n)$ are bounded. From the inequalities

$$\begin{aligned} \phi(z', x_{n+1}) &\leq \phi(z', x_n) - \phi(x_{n+1}, x_n) + 2\lambda_n \langle Ay_n, y_n - x_{n+1} \rangle, \\ -2\lambda_n \langle Ay_{n-1}, x_{n+1} - y_n \rangle &\leq \phi(x_{n+1}, x_n) - \phi(y_n, x_n) - \phi(x_{n+1}, y_n) \end{aligned}$$

and (4.12) it follows that $\lim_{n \rightarrow \infty} \phi(x_{n+1}, x_n) = 0$ and $\lim_{n \rightarrow \infty} \|x_{n+1} - x_n\| = 0$. Since on bounded sets the operator J is uniformly continuous, we have $\lim_{n \rightarrow \infty} \|Jx_n - Jx_{n+1}\|_* = 0$.

We now show that all weak cluster points of the sequence (x_n) belong to the set S . Consider a subsequence (x_{n_k}) that converges weakly to some point $z \in E$. Clearly, $z \in C$, and $(y_{n_k}), (x_{n_k+1})$ converge weakly to z . We show that $z \in S$. We have

$$\langle Ay_{n_k}, y - x_{n_k+1} \rangle + \frac{1}{\lambda_{n_k+1}} \langle Jx_{n_k+1} - Jx_{n_k}, y - x_{n_k+1} \rangle \geq 0 \quad \forall y \in C.$$

Hence,

$$\begin{aligned} \langle Ay, y - y_{n_k} \rangle + \langle Ay_{n_k}, y_{n_k} - x_{n_k+1} \rangle + \\ + \frac{1}{\lambda_{n_k+1}} \langle Jx_{n_k+1} - Jx_{n_k}, y - x_{n_k+1} \rangle \geq 0 \quad \forall y \in C. \end{aligned}$$

Passing to the limit in this inequality, we obtain

$$\langle Ay, y - z \rangle \geq 0 \quad \forall y \in C,$$

which means that $z \in S$.

We now prove the weak convergence of the sequence (x_n) to z . From this convergence and the fact that $\|y_n - x_n\| \rightarrow 0$, we immediately obtain the weak convergence of the sequence (y_n) to z as well. To prove by contradiction, assume that there exists a subsequence (x_{m_k}) such that $x_{m_k} \rightharpoonup z'$ weakly and $z \neq z'$. Then $z' \in S$. We have

$$2 \langle Jx_n, z - z' \rangle = \phi(z', x_n) - \phi(z, x_n) + \|z\|^2 - \|z'\|^2.$$

Hence, the limit $\lim_{n \rightarrow \infty} \langle Jx_n, z - z' \rangle$ exists. Since the duality mapping J is sequentially weakly continuous, we get

$$\langle Jz, z - z' \rangle = \lim_{k \rightarrow \infty} \langle Jx_{n_k}, z - z' \rangle = \lim_{k \rightarrow \infty} \langle Jx_{m_k}, z - z' \rangle = \langle Jz', z - z' \rangle,$$

that is, $\langle Jz - Jz', z - z' \rangle = 0$. This implies $z = z'$.

It is worth noting that similar results can also be obtained for other algorithms of this type for variational inequalities — for example, for the operator extrapolation method, see [87].

4.4 Adaptive Extrapolation from the Past Algorithm

Among the convergence conditions for the algorithm considered above were restrictions on the step λ_n , namely

$$0 < \inf_n \lambda_n \leq \sup_n \lambda_n < \frac{1}{3\mu L}.$$

Thus, to choose the step it was necessary to use the value of the Lipschitz constant of the operator A , which is usually difficult to compute. To remove these requirements for the proposed algorithm, we construct its adaptive version using the approaches of [101, 102] (see also [91, 27]). Our goal is also to obtain an adaptive step-size rule without a significant increase in the number of computations per iteration — that is, without additional projections or operator evaluations.

We will assume that the value $\mu \geq 1$ from Lemma 4.2 is known, and justify the following algorithm:

Algorithm 4.2.

Initialization. Choose $x_1 = y_0 \in E$, $\tau \in \left(0, \frac{1}{3\mu}\right)$, and a number $\lambda_1 > 0$. Set $n = 1$.

Step 1.

$$y_n = \Pi_C J^{-1} (Jx_n - \lambda_n A y_{n-1}).$$

Step 2.

$$x_{n+1} = \Pi_C J^{-1} (Jx_n - \lambda_n A y_n).$$

If $y_n = x_n = x_{n+1}$, then the algorithm terminates. Otherwise, proceed to Step 3.

Step 3.

$$\lambda_{n+1} = \begin{cases} \min \left\{ \lambda_n, \tau \frac{\|y_n - y_{n-1}\|}{\|A y_n - A y_{n-1}\|_*} \right\}, & \text{if } A y_{n-1} \neq A y_n, \\ \lambda_n, & \text{otherwise.} \end{cases}$$

Set $n := n + 1$ and return to Step 1.

Note that the sequence (λ_n) , defined at the third step of the iterative procedure in the above algorithm, is non-increasing and bounded below by $\min \left\{ \lambda_1, \frac{\tau}{L} \right\}$. Hence, there exists $\lim_{n \rightarrow \infty} \lambda_n > 0$.

Let us now prove the convergence of Algorithm 4.2. We first establish the key inequality that connects the generalized distances between the approximations computed by the algorithm and an arbitrary point of the solution set S .

Lemma 4.6. For the sequences (x_n) , (y_n) generated by Algorithm 4.2, the following inequality holds:

$$\begin{aligned} \phi(z, x_{n+1}) \leq \phi(z, x_n) - \left(1 - 2\tau\mu \frac{\lambda_n}{\lambda_{n+1}}\right) \phi(x_{n+1}, y_n) - \\ - \left(1 - \tau\mu \frac{\lambda_n}{\lambda_{n+1}}\right) \phi(y_n, x_n) + \tau\mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_n, y_{n-1}), \end{aligned}$$

where $z \in S$.

Proof. Let $z \in S$. As in the proof of Lemma 4.5, we arrive at the inequality

$$\begin{aligned} \phi(z, x_{n+1}) \leq \phi(z, x_n) - \phi(x_{n+1}, y_n) - \phi(y_n, x_n) + \\ + 2\lambda_n \langle A y_n - A y_{n-1}, y_n - x_{n+1} \rangle + 2\lambda_n \langle A z, z - y_n \rangle. \end{aligned}$$

Since $\langle A z, z - y_n \rangle \leq 0$, from this inequality it follows that

$$\begin{aligned} \phi(z, x_{n+1}) \leq \phi(z, x_n) - \phi(x_{n+1}, y_n) - \phi(y_n, x_n) + \\ + 2\lambda_n \langle A y_n - A y_{n-1}, y_n - x_{n+1} \rangle. \quad (4.13) \end{aligned}$$

For the term $2\lambda_n \langle A y_n - A y_{n-1}, y_n - x_{n+1} \rangle$, from the step-size updating rule λ_{n+1} in Algorithm 4.2, we obtain the estimate

$$2\lambda_n \langle A y_n - A y_{n-1}, y_n - x_{n+1} \rangle \leq 2\lambda_n \|A y_n - A y_{n-1}\|_* \|y_n - x_{n+1}\| \leq$$

$$\begin{aligned}
&\leq 2\tau \frac{\lambda_n}{\lambda_{n+1}} \|y_n - y_{n-1}\| \|x_{n+1} - y_n\| \leq \\
&\leq 2\tau \frac{\lambda_n}{\lambda_{n+1}} (\|y_n - x_n\| + \|x_n - y_{n-1}\|) \|x_{n+1} - y_n\| \leq \\
&\leq \tau \frac{\lambda_n}{\lambda_{n+1}} \|y_n - x_n\|^2 + \tau \frac{\lambda_n}{\lambda_{n+1}} \|x_n - y_{n-1}\|^2 + 2\tau \frac{\lambda_n}{\lambda_{n+1}} \|x_{n+1} - y_n\|^2 \leq \\
&\leq \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(y_n, x_n) + \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_n, y_{n-1}) + 2\tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_{n+1}, y_n).
\end{aligned}$$

After substituting this estimate into (4.13), we obtain

$$\begin{aligned}
\phi(z, x_{n+1}) &\leq \phi(z, x_n) - \phi(x_{n+1}, y_n) - \phi(y_n, x_n) + \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(y_n, x_n) + \\
&\quad + \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_n, y_{n-1}) + 2\tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_{n+1}, y_n),
\end{aligned}$$

which completes the proof of the lemma.

Using Lemma 4.6, we now prove the convergence of Algorithm 4.2.

Theorem 4.3. Let C be a nonempty closed convex subset of a 2-uniformly convex and uniformly smooth Banach space E , $A : E \rightarrow E^*$ a monotone and Lipschitz operator, and $S \neq \emptyset$. Assume that the normalized duality mapping J is sequentially weakly continuous. Then the sequences (x_n) and (y_n) generated by Algorithm 4.2 converge weakly to some point $z \in S$.

Proof. Let $z' \in S$. Define

$$\begin{aligned}
a_n &= \phi(z', x_n) + \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_n, y_{n-1}), \\
b_n &= \left(1 - \tau \mu \frac{\lambda_n}{\lambda_{n+1}}\right) \phi(y_n, x_n) + \left(1 - \tau \mu \frac{\lambda_{n+1}}{\lambda_{n+2}} - 2\tau \mu \frac{\lambda_n}{\lambda_{n+1}}\right) \phi(x_{n+1}, y_n).
\end{aligned}$$

Then, from the last lemma we obtain $a_{n+1} \leq a_n - b_n$. Since $\exists \lim_{n \rightarrow \infty} \lambda_n > 0$, we have

$$1 - \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \rightarrow 1 - \tau \mu \in (0, 1),$$

and also

$$1 - \tau \mu \frac{\lambda_{n+1}}{\lambda_{n+2}} - 2\tau \mu \frac{\lambda_n}{\lambda_{n+1}} \rightarrow 1 - 3\tau \mu \in (0, 1) \quad \text{as } n \rightarrow \infty.$$

From Lemma 3.2 it follows that

$$\lim_{n \rightarrow \infty} \left(\phi(z', x_n) + \tau \mu \frac{\lambda_n}{\lambda_{n+1}} \phi(x_n, y_{n-1}) \right)$$

exists. From the same lemma we have the boundedness

$$\begin{aligned}
&\sum_{n=1}^{\infty} \left(\left(1 - \tau \mu \frac{\lambda_n}{\lambda_{n+1}}\right) \phi(y_n, x_n) + \right. \\
&\quad \left. + \left(1 - \tau \mu \frac{\lambda_{n+1}}{\lambda_{n+2}} - 2\tau \mu \frac{\lambda_n}{\lambda_{n+1}}\right) \phi(x_{n+1}, y_n) \right) < +\infty.
\end{aligned}$$

Taking into account the properties of (λ_n) , τ , and μ , this implies

$$\lim_{n \rightarrow \infty} \phi(y_n, x_n) = \lim_{n \rightarrow \infty} \phi(x_{n+1}, y_n) = 0.$$

This also proves that the numerical sequences of the form $(\phi(z', x_n))$ are convergent for any $z' \in S$. With these results, the proof of the theorem is completed by fully repeating the reasoning from Theorem 4.1.

4.5 Summary of Chapter 4

In this chapter, developing the ideas from [101, 91, 102, 27] and based on the extrapolation from the past algorithm [82], two algorithms were constructed for solving problem (1.1) in Banach spaces. Theorems on the weak convergence of the algorithms were proved in the case where the Banach space is 2-uniformly convex and uniformly smooth, and the operator is monotone and Lipschitz continuous.

Similarly to [31], one can also prove weak convergence of the proposed algorithms in the case when the operator is only pseudomonotone and sequentially weakly continuous (in the same type of spaces).

It should be noted that for the efficiency of the algorithms in the above setting, the possibility of fast and stable computation of the Alber generalized projection for the corresponding feasible set is critically important — otherwise the advantage of the algorithm is lost. The results in this chapter were established under the assumption that the Banach space is 2-uniformly convex and uniformly smooth. That is, they do not include the widely used spaces L_p and W_p^m ($2 < p < +\infty$). Research in this direction is ongoing, since this restriction seems to be more technical than essential.

The results of this chapter were published in [90, 34, 117, 32, 27, 24, 26].

Chapter 5

Software framework and applications of the algorithms

This chapter describes an extendable software package which allows applying extragradient-type algorithms to problems that can be modeled as variational inequalities. Additionally, the program provides detailed statistics on the behavior of the algorithms during the solution process. Also the chapter presents results of computational experiments, where several problems from network economics field are solved using the algorithms described in the previous parts.

5.1 Software framework for applying and investigating extragradient-type algorithms

As can be seen from algorithms formulations above, they have many common elements - basically, every algorithm has one or more substeps, where it does operator calculation at some points, projections on feasible and, possibly, auxiliary sets and applies step size adaptation rule (for adaptive algorithms). The same is true about problems which are modeled as variational inequalities – the core of the problem description is the operator calculation procedure and feasible set projection operation.

This allowed us to implement an open source Python-based software framework with considerable degree of universality for both solving problems modeled as variational inequalities and conducting numerical experiments with any set of algorithms from the family. Code and description can be found on public repository ¹

As the main target audience for the software are researchers, who need not only to get a solution of some problem, but usually require possibility to tweak parameters or even logic of both algorithm and problem, we abstracted common properties and functions of algorithm to base classes. This

¹<https://github.com/compmath-sdeni/vi-alg-suite>

way to plug in new or modified algorithm to the system the researcher only needs to implement (override) logic which is different for their's specific algorithm. The same approach is used for describing a problem – again, researcher could derive the problem class from the base class in our system, and reuse existing boilerplate code, focusing on specific new logic only. Simplified class diagram for problem and algorithm classes is shown below.

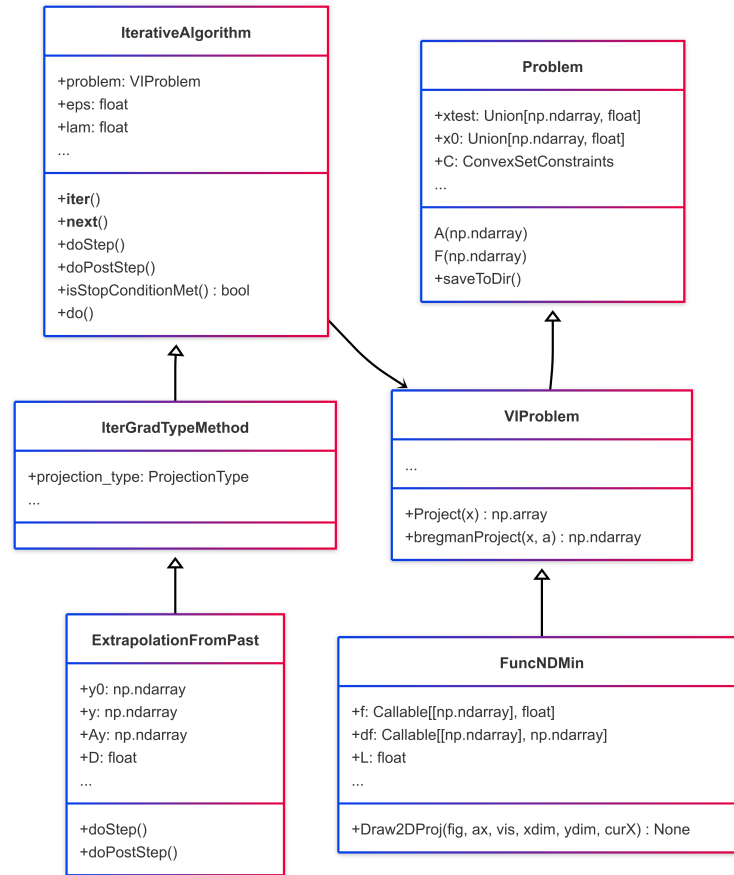


Figure 5.1: Part of algorithms and problems related classes

Additionally, the system contains helper classes and functions, which simplify common routines researchers often do during numerical experiments with algorithms and mathematical models – for example, there are classes for plotting convergence graphs. For the x axis one could use the iteration number or computational time. And for the y axis gap function (residue), the distance between two sequential approximations or distance to the exact solution (if known) can be used.

As a result, with a small amount of code, a researcher can select subset of algorithms of interest, select and set parameters for one of already implemented problems (or add own problem modelled as variational inequality), and run all algorithms for the same problem, easily comparing their behavior. Schematic usage with python code is shown below (class and variable names shortened, and some arguments skipped to save space, but the code itself is copied from real usage scenario).

Another useful feature of the proposed software is the history collection mode – detailed execution log of the algorithm can be collected and saved as csv and excel format. This way the

```

par = AlgPar(eps=1e-8, lam0=0.5, tau=0.3, hist=True)
# .....
problem = prnk.prepare(algo_params=par, ...)
efp_alg = EfPAdapt(problem, stop=par.stop_by, ...)
# .....
algs = [alg1a, alg2a, ...]
# .....
for alg in algs_to_test:
    alg.do()
    AlgTest.PrintStats(alg, max_print=20)
    alg_hist_list.append(alg.history)
    # Save history to excel
grapher = AlgStatGrapher()
grapher.plot_by_history(alg_hist_list)
# Save plots as png and eps

```

Figure 5.2: Schematic framework usage flow

researcher can analyse both generated sequence and internal algorithm state changes history to get insights and detect problematic moments. Small example of history with subset of columns are shown below:

A	B	C	D	E	F	G
iter	x	y	lam	step delta norm	gap	iter time ns
0	[0.20000,0.2	[0.20000,0.2		2.53.735131	0.34	0
1	[0.83007, 0	[0.20000,0.2		0.6393.735131	0.73077	281577
2	[0.91398,0.0	[0.55398,0.1		0.4991.568371	0.85617	409876
3	[0.89230, 0	[0.68596,0.0		0.4991.329289	0.82151	566030
4	[0.85411, 0	[0.73009,0.0		0.4990.722516	0.75792	721489
5	[0.79161, 0	[0.74382,0.0		0.499 0.7292	0.64846	875826

Figure 5.3: Part of collected algorithm execution history

Also the system allows to set a subset of algorithm and problem parameters from command line, avoiding code editing. And it is extendable in terms of adding UI editing capabilities – at the moment, there is a web UI for blood collection and delivery network optimization problem described below.

5.2 Problems of Network Economics and Variational Inequalities

Variational inequalities are a powerful approach for modeling a wide range of problems, including optimization and equilibrium problems, parts of machine learning tasks, and others. In particular, most problems of network economics are naturally reduced to solving variational inequalities. Corresponding formulations for various types of problems in network economics were obtained in a series of works by a group of authors from UMass Amherst led by Anna Nagurney [70, 72].

Among these problems, one of the most interesting is the modeling of the supply chain of perishable products [75], where the critical factor is the storage time of the product, and as a consequence,

an excess of product at the demand point entails significant disposal costs. Moreover, each link in the network where transportation and temporary storage of such a product occur usually also has associated product losses and corresponding disposal costs. A very similar model arises for the important problem of optimizing the blood supply chain, which can be considered as a sequence of stages: collection, testing, processing, and use of blood — from collection points to hospitals. Modeling this problem for the structure of the American Red Cross (ARC) was studied in [76]. In particular, the mathematical model was formulated as a variational inequality (1.1). Another variant of the mathematical model for the blood supply chain optimization problem can be seen in [5], where a variational inequality is also used as part of the optimization setting of a bilevel problem.

Another important problem that also arises in the analysis of network models is the task of ranking network nodes by importance, or computing centrality measures. One of the well-known approaches to solving this problem for a network of web pages is called PageRank. It was proposed in [17] — and it was this algorithm that enabled Google to enter the search engine market and capture the leadership it still holds today. At present, there are many variations of this algorithm, and the areas of application are almost innumerable — chemistry, biology (general and neuroscience), social networks, text analysis, recommendation systems, knowledge structuring systems, and others [46]. As shown below, the PageRank construction problem can also be formulated as a variational inequality. Moreover, the feasible set has a structure for which algorithms based on Bregman divergence instead of Euclidean distance — in particular, those described in Chapter 3 — provide an advantage.

5.3 Optimization of the Blood Collection and Delivery Chain

5.3.1 Problem Statement

We present a mathematical model of the optimization of the blood supply chain, as a problem of network economics, based on [76]. The structure of the corresponding network and the diagram of the supply stages are schematically shown in Figure 5.4.

In the diagram, the edges correspond to activities that occur within the blood supply chain, and are associated with the corresponding model parameters, such as collection risks or operational costs and disposal costs. The nodes represent the entities involved in the supply chain. The first node does not have a clear physical analogue and is added to ensure consistency of the model formulation. At the same time, it can be interpreted as a top-level organizational unit, for example, a regional branch of the Red Cross, as in [76]. The process starts with blood collection. The specificity of this stage (edges from the initial node to the blood collection facilities) is that it is associated with risks and related costs. Indeed, the process itself is risky — donors generally have no strict obligations and may cancel their appointment for many reasons. For example, there are

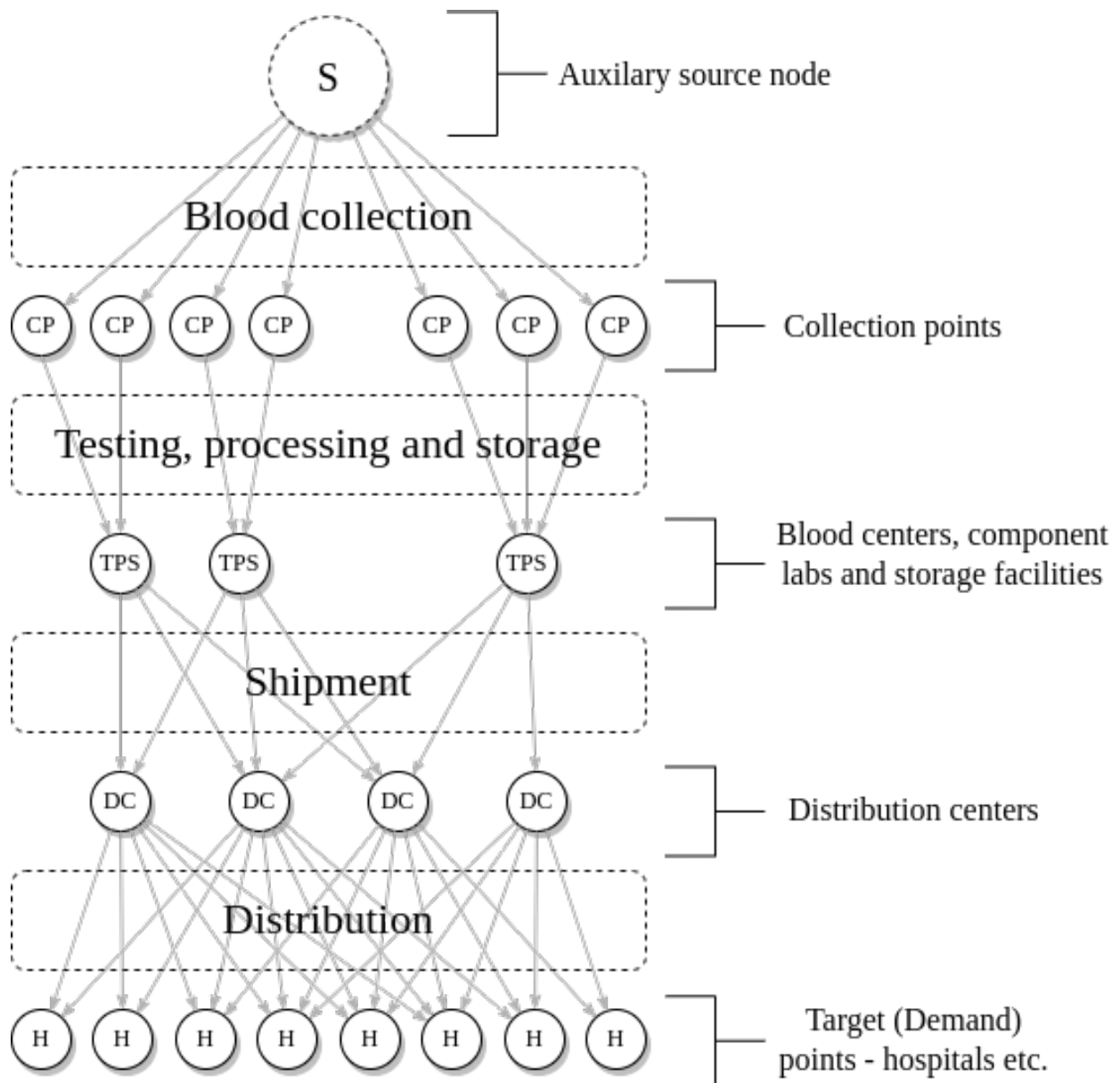


Figure 5.4: Scheme of the blood supply network.

statistics showing that unfavorable weather conditions significantly reduce the amount of collected blood. Conversely, the appearance of information about a disaster or urgent need to help victims leads to a peak in donations.

The next stages are the testing of the collected blood for infections, the separation of components (plasma and red blood cells), and sending the prepared blood to special storage centers. In the scheme, the testing, processing, and storage facilities are combined into one layer to simplify the description of the model and the diagram, without losing the key properties. In reality, these centers are also usually located at the same site — and even if they are separate, this does not lead to significant changes in the model, since all edges have the same set of associated parameters. In our model, the set of parameters of the delivery stage edges coincides with the previous one, but

physically and logically the distribution centers are almost always separated, so they are represented as a separate layer in the diagram.

Finally, the last layer corresponds to the delivery of blood directly to the destinations, which are usually hospitals. The edges of this layer have their own specifics within the model — they are associated with probabilistic parameters of the blood demand in each particular hospital, and the corresponding losses in case of blood shortage or surplus.

5.3.2 Formulation of the Mathematical Model of the Blood Supply Chain

Let n_c, n_p, n_s, n_h be the number of collection nodes, combined nodes for testing, processing and storage, distribution nodes, and final nodes (hospitals), respectively. Let the set of all edges be denoted as $\{e_j\}, j = 1, \dots, n_e$, where n_e is the total number of edges in our blood supply chain network. The input flow for edge e_j is denoted by y_j (since losses may occur on the edge, the output flow may differ). The flows along the edges together form the vector $y = (y_1, \dots, y_{n_e})$.

Each operation at every stage has a cost, also attached to the edges. Let $c_j(y_j)$ denote the unit operational cost on edge e_j , which depends on the flow along this edge. Then the total operational cost on the edge is computed as $y_j \cdot c_j(y_j)$.

Let $P = \{p_i\}, i = 1, \dots, m$ be the set of all simple paths from the dummy node 0 to each hospital, and let m be the total number of such paths. The initial flow on path $p_i = \{e_{i1}, e_{i2}, \dots, e_{ik}\}$ is denoted as $x_i \geq 0$, and it coincides with the input flow of the first edge of the path, which is always the edge between the source node and the blood collection node. Our goal is to find the optimal vector of flows for all paths $x = (x_1, \dots, x_m)$. In these notations, the first requirement for the feasible set C can be written as

$$x_i \geq 0, \quad i = 1, \dots, m. \quad (5.1)$$

At each stage of blood processing, losses may occur — for example, a small portion of the blood is taken for laboratory analysis and then discarded. Let $a_j \in (0, 1]$ be the loss coefficient for each edge e_j , where $j = 1, \dots, n_e$. We interpret it as follows: if the input flow on edge e_j is y_j , then the resulting output flow will be $a_j y_j$. For simplicity, we assume that these coefficients are independent of the flow on the edge, leaving the case of dependence for future research.

Losses during blood processing include the cost of waste disposal. These expenses can be significant for products such as blood, so it is important to incorporate them into the model. The amount of waste for edge e_j depends only on the flow along that edge and equals $(1 - a_j)y_j$. Hence, the corresponding part of the loss function also depends on the flow — let the unit disposal cost be denoted as $w_j(y_j)$, so the total disposal cost is $y_j \cdot w_j(y_j)$.

As for the path flows, the total retention on path p_i can be computed as $\beta_i = \prod_{e_j \in p_i} a_j$, where $i = 1, \dots, m$. As a result, if the initial flow on path p_i is x_i , then the corresponding final flow along

the path will be $\beta_i x_i$, for each $i = 1..m$. For each destination node H_k , where $k = 1..n_h$, let us denote the total supply as s_k . Clearly, we have

$$s_k = \sum_{p_i \in P_k} \beta_i x_i, \quad (5.2)$$

where P_k is the set of all paths from the initial node to the destination node H_k . At the same time, the actual demand d_k is stochastic — we cannot exactly predict the need for donor blood at a given hospital at a specific time in the future. However, we assume that the corresponding probability density function $F_k(t)$ and probability distribution $P_k(z) = p(d_k < z) = \int_0^z F_k(t) dt$ are known.

We are interested in minimizing the expected difference between s_k and d_k . Hence, one might be tempted to use something like $E(\sum_{k=1}^{n_h} (s_k - d_k)^2)$ as the loss function. However, the consequences of shortage and surplus of blood differ significantly, so in the model it makes sense to treat them separately.

Let the shortage of blood at destination node H_k be defined as

$$U_k = \begin{cases} d_k - s_k, & \text{if } d_k > s_k, \\ 0, & \text{if } d_k \leq s_k, \end{cases}$$

where U_k can also be written as $\max\{0, d_k - s_k\}$. The surplus of blood at the same node is defined as

$$V_k = \begin{cases} s_k - d_k, & \text{if } d_k < s_k, \\ 0, & \text{if } d_k \geq s_k, \end{cases} = \max\{0, s_k - d_k\}.$$

Then the corresponding part of the cost function for the model can be written as

$$\sum_{k=1}^{n_h} (\gamma_{uk} E(U_k) + \gamma_{vk} E(V_k)),$$

where γ_{uk}, γ_{vk} are penalty parameters (cost coefficients), and $E(U_k), E(V_k)$ are the expected shortage and surplus of blood, respectively.

Finally, the model must account for risks at the blood collection stage [76]. These risks are associated only with the edges of the first layer, and we assume that they are also functions of the flow. Let us denote the risks as $r_i(y_i)$, where $i = 1, \dots, n_c$. Minimization of risks and minimization of costs are different criteria, so in the cost function they can be weighted differently. Let θ denote the weight for the risk component, while the cost component is assigned weight one.

Remark 5.1. Since the blood collection process consists of sequential stages, it is always possible to number the edges in such a way that any edge of a later stage has an index larger than any edge of an earlier stage. For example, the index of a distribution edge is always larger than the index of a collection edge. We assume such a numbering scheme throughout the following reasoning to simplify the model formulation. As a result, the indices of the edges in any path $p_i \in P$ form an increasing sequence.

Taking into account the above remark, the flow of each edge can be calculated from the path flows:

$$y_j = \sum_{i: e_j \in p_i} x_i \alpha_{ij}, \quad (5.3)$$

where $\alpha_{ij} = \prod_{k: e_k \in p_i, k < j} \alpha_k$ is the product of all loss coefficients of the edges that precede edge e_j in path p_i .

Remark 5.2. In the given formulation, we have exactly n_c edges in the first layer, i.e., the edges $e_{ij}, j = 1, \dots, n_c$ correspond to the operation of blood collection from donors. This provides us with a simple way to write the risk component of the cost function below.

5.3.3 Optimization problem and variational inequality

Now we can formulate the blood supply chain network optimization problem as the minimization of the combined cost function

$$\begin{aligned} \Phi(\mathbf{y}) = & \sum_{j=1}^{n_e} (y_j (c_j(y_j) + w_j(y_j))) + \\ & + \sum_{k=1}^{n_h} (\gamma_{uk} E(U_k) + \gamma_{vk} E(V_k)) + \theta \sum_{j=1}^{n_c} r_j(y_j) \end{aligned} \quad (5.4)$$

subject to (5.1)–(5.3). Using (5.3), we can rewrite (5.4) in terms of path flows:

$$\Phi(\mathbf{x}) = \sum_{i=1}^m x_i (C_i(x) + W_i(x) + \theta R_i(x)) + \sum_{k=1}^{n_h} (\gamma_{uk} E(U_k) + \gamma_{vk} E(V_k)), \quad (5.5)$$

where C_i and W_i are the unit operation cost and disposal cost functions for path p_i , and R_i is the risk cost function for the same path. These functions take the form:

$$\begin{aligned} C_i(x) &= \sum_{j: e_j \in p_i} c_j(y_j), \\ W_i(x) &= \sum_{j: e_j \in p_i} w_j(y_j), \\ R_i(x) &= \sum_{j: e_j \in p_i, j=1..n_c} r_j(y_j), \end{aligned}$$

where y_j is expressed in terms of the path flows according to (5.3). The notation is slightly different from [76] to simplify the algorithm implementation, since the computations use the sequence of edges in the path. The problem can be formulated as a classical variational inequality. We need to find $x^* \in C$:

$$\left(\mathbf{x} - \mathbf{x}^*, \frac{d\Phi}{d\mathbf{x}}(\mathbf{x}^*) \right) \geq 0 \quad \forall \mathbf{x} \in C, \quad (5.6)$$

where $C = \mathbb{R}_+^m$ and

$$\begin{aligned} \frac{d\Phi}{dx_i} = \sum_{j: e_j \in p_i} \alpha_{ij} [c_j(y_j) + w_j(y_j) + y_j (c'_j(y_j) + w'_j(y_j))] + \\ + \beta_i (\lambda_{uti}(P_{ti}(v_{ti}) - 1) + \lambda_{vti}(P_{ti}(v_{ti}))) + \theta (r_{si}(y_{si}) + r'_{si}(y_{si}) \cdot y_{si}), \end{aligned}$$

where s_i and t_i are the indices of the first and the last edge in path p_i . Again, here y_j is expressed in terms of the path flows using equality (5.3).

5.3.4 Application of Algorithms to the Blood Supply Chain Problem

The blood supply chain optimization problem (5.5), formulated as the variational inequality (5.6), was integrated into our software framework, described above. For this problem there is a graphical user interface, which allows to edit network parameters, select algorithms, obtain solutions and analyze algorithm behavior. The problem selector and editor UI (Figure 5.5) are implemented as a web application using the Plotly Dash library – see figure below.

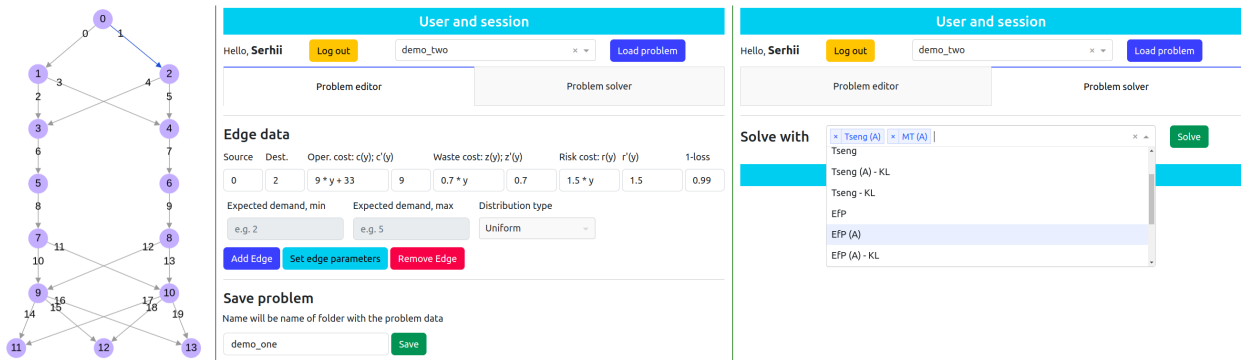


Figure 5.5: Editing problem parameters.

5.3.5 Computational results

Figure 5.6 shows how the computation results are presented in the web UI. Currently, the output includes a convergence rate plot (the x -axis corresponds to the iteration number, the y -axis corresponds to the distance between approximate solutions) and textual information with final results, runtime, and run parameters.

For problems with a known solution, a display mode can be selected where the distance to the true solution is shown instead of the distance between successive approximations, which is very useful for testing and debugging algorithms. Another mode is available in which the x -axis represents runtime instead of iteration number. This mode should not be taken as absolutely precise, since execution time in multitasking operating systems depends on the overall CPU load. However, if the problem is large enough and computations take tens of seconds or minutes rather than fractions

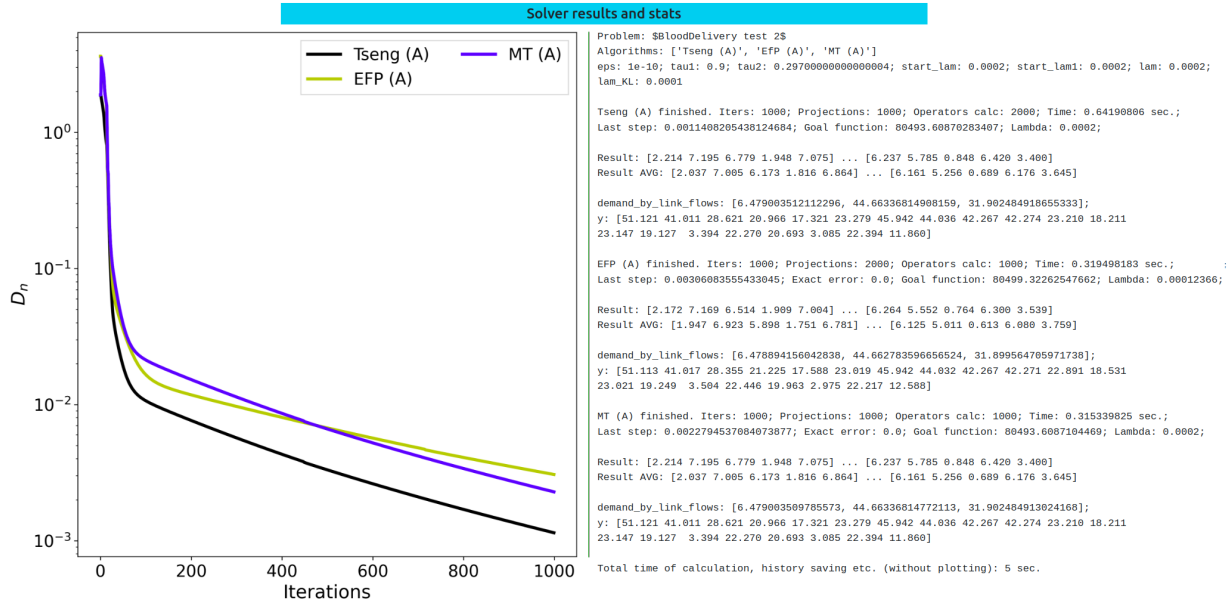


Figure 5.6: Experiment results view.

of a second, this characteristic becomes sufficiently accurate to draw conclusions about algorithm performance. In addition, the textual output includes various indicators specific to the given class of problems — for example, supply volumes for each hospital. During computations, the system also collects and stores the detailed step-by-step history for each algorithm (in tabular format), allowing a full analysis of the algorithm's state and behavior at every iteration (though such analysis becomes very time-consuming for problems of non-trivial dimensions).

Based on the numerical experiment presented, it can be observed that after 1000 iterations, the adaptive Tseng algorithm slightly outperforms both EFP and the Malitsky–Tam (MT) algorithm in terms of accuracy on this test problem. At the same time, it should be noted that the runtime for 1000 iterations of Tseng is twice as long (0.64 seconds versus 0.32 seconds for both EFP and MT), which allows us to qualitatively conclude about relative efficiency. In fact, this behavior is consistent with theory — in this problem, projection is very cheap ($C = \mathbb{R}_+^m$), while computing the operator is rather time-consuming (due to the complex differential in (5.6)). It is also worth noting that despite the much smaller step distances (0.001 for Tseng compared to 0.002 and 0.003 for MT and EFP), the objective function values differ relatively little (80493 versus 80499).

It should also be emphasized again that the runtime of a single run is not a stable performance measure due to various influencing factors, such as current CPU load by background processes. However, behavior similar to that described above was obtained in multiple repeated numerical experiments and by averaging the results.

5.4 The Web Page Ranking Problem — PageRank

The problem of ranking (selecting the “best” or “most important”) nodes of a network is an important subtask for various scenarios. One of the most practically valuable formulations is ranking web pages for search engine results — in fact, this was the task solved by the founders of Google in [81]. Today, users rarely look beyond the 10th position in search results — but this was not always the case. Google entered the market at the time with algorithms that solved this problem much better than competitors — and PageRank was the key component of their system. Similar approaches remain an important part of search result ranking even now.

5.4.1 Mathematical Model

There are currently many variants of the PageRank mathematical model, but the original idea was that a page is more important if more other pages link to it. To formalize this idea, represent the part of the web that interests us as a directed graph. Vertices correspond to pages, edges to hyperlinks, and the weight of each vertex is its importance, or rank, which must be computed. The obvious metric based simply on the number of links is not suitable for an unmoderated environment such as the web — because it is easy to create hundreds of meaningless pages linking to one advertised page, thereby artificially inflating its importance. Conversely, one could sell hundreds of links from a single page to different advertisers.

To prevent such scenarios, the metric accounts not only for the number but also the quality of links. Specifically, a page transfers through each of its links a portion of its weight, inversely proportional to the number of outgoing links from that page. Let the number of pages in the subset of the web under consideration be N . Then, for the weight of the i -th page, we have

$$x_i = \sum_{j \in L_i} \frac{x_j}{m_j}, \quad i = 1, \dots, N,$$

where L_i is the set of pages linking to page i . This system of relations can be written in matrix form as

$$x = Ax, \quad x = (x_1, x_2, \dots, x_N), \quad A = (a_{ij}), \quad (5.7)$$

where

$$a_{ij} = \begin{cases} 1/m_j, & j \in L_i, \\ 0, & j \notin L_i. \end{cases}$$

Thus, A is an $N \times N$ matrix, and this problem can be seen as finding an eigenvector corresponding to eigenvalue 1.

If there are no “dangling nodes” (vertices with no outgoing links) and no “forgotten nodes” (vertices with no incoming links), then A is a stochastic matrix (a Markov matrix). Indeed, by con-

struction, the sum of elements in each column is 1. For such a matrix, the corresponding eigenvector exists and is unique. However, in real web graphs, there usually exist vertices of both types.

Moreover, starting from the idea of counting links, we obtained a Markov matrix and a problem equivalent to computing the stationary distribution of a random walk in the graph — i.e., the limiting probability of a user being on each page after infinitely many steps. However, real user behavior differs somewhat — sometimes the user “teleports” in the network, i.e., instead of following a link, directly jumps to another page of their choice. If we assume that the probability of jumping to any page is uniform, and the probability of “teleportation” instead of following a link is $\alpha \in (0, 1)$, then the mathematical model can be modified as follows:

$$x = Mx, \quad M = \alpha A + (1 - \alpha)S, \quad (5.8)$$

where $S = (1/N)$. It is clear that the column-stochastic property of A also holds for M . Furthermore, even if matrix A contains zero rows (nodes with no outgoing edges) or independent blocks (corresponding to disconnected subgraphs of the web graph), adding matrix S removes the violation of uniqueness of the solution. The properties of this model still degrade in the presence of dangling nodes (vertices with no outgoing edges), but the formulation already allows solving real ranking problems.

5.4.2 Formulation as a Variational Inequality

The problem (5.8) can be reformulated as a variational inequality. First, note that it is equivalent to the following optimization problem:

$$\text{find } x^* \in \Delta^N : \quad \|Mx^* - x^*\|_\infty = \min_{x \in \Delta^N} \|Mx - x\|_\infty,$$

where $\Delta^N = \{x \in \mathbb{R}^n : \sum_{i=1}^N x_i = 1, x_i \geq 0\}$ is the standard simplex, and $\|z\|_\infty = \max_{i=1..N} |x_i|$. This problem can be reduced to finding a saddle point:

$$\min_{x \in \Delta^N} \max_{\|y\|_1 \leq 1} (y, Mx - x), \quad (5.9)$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$. The corresponding variational inequality takes the form

$$\begin{aligned} &\text{find } x^* \in \Delta^n, y^* \in B_1^n : \\ &((M^T - I)y^*, x - x^*) + ((I - M)x^*, y - y^*) \geq 0, \\ &\forall x \in \Delta^n, y \in B_1^n, \text{ where } B_1^n = \{x : \|x\|_1 \leq 1\}. \end{aligned} \quad (5.10)$$

In this case, projection simplifies partially, since for the standard simplex under the Kullback–Leibler distance it can be written explicitly (see (3.12)). Projection onto the unit ball is also straightforward. However, to fully leverage the properties of Bregman projection, the saddle-point problem

(5.9) can be further reformulated and reduced to a bilinear minimax problem on the Cartesian product of standard simplices [32]:

$$\min_{x \in \Delta^n} \max_{y \in \Delta^{2n}} (Jy, Mx - x), \quad (5.11)$$

where J is an $n \times 2n$ matrix of the form $(I, -I)$ (with I the identity matrix). This problem is equivalent to the following:

$$\begin{aligned} &\text{find } x^* \in \Delta^n, y^* \in \Delta^{2n} : \\ &\quad ((M^T - I) J y^*, x - x^*) + (J^T (I - M) x^*, y - y^*) \geq 0, \\ &\quad \forall x \in \Delta^n, y \in \Delta^{2n}. \end{aligned}$$

If we denote $z = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_{2n}) \in \mathbb{R}^{3n}$, and

$$\tilde{M} = \begin{pmatrix} J^T (I - M) & 0 \\ 0 & (M^T - I) J \end{pmatrix},$$

then we obtain a variational inequality of the form (1.1):

$$\text{find } z^* \in \Delta^n \times \Delta^{2n} : \left(\tilde{M} z^*, z - z^* \right) \geq 0 \quad \forall z \in \Delta^n \times \Delta^{2n}. \quad (5.12)$$

For such a variational inequality, the corresponding algorithms can be applied by writing out the Bregman projection explicitly. For example, the main step of Algorithm 3.2 takes the following form:

Initialization. Choose $p_0, z_1 \in \Delta^n \times \Delta^{2n} \subset \mathbb{R}^{3n}$, $\tau \in (0, \frac{1}{3})$ and $\lambda_1 > 0$. Start with step number $m = 1$.

Step 1.

$$p_m = P_{z_m}^{\Delta^n \times \Delta^{2n}} \left(-\lambda_m \tilde{M} p_{m-1} \right)$$

Step 2.

$$z_{m+1} = P_{z_m}^{\Delta^n \times \Delta^{2n}} \left(-\lambda_m \tilde{M} p_m \right).$$

If $p_m = z_m = z_{m+1}$, then the algorithm terminates. Otherwise, proceed to the next step.

Step 3. Adaptive step size adjustment:

$$\lambda_{m+1} = \begin{cases} \min \left\{ \lambda_m, \tau \frac{\|z_m - z_{m-1}\|_1}{\|\tilde{M} z_m - \tilde{M} z_{m-1}\|_\infty} \right\}, & \text{if } \tilde{M} z_{m-1} \neq \tilde{M} z_m, \\ \lambda_m, & \text{otherwise.} \end{cases}$$

Here again, $p = (p_1, p_3, \dots, p_{3n}) \in \mathbb{R}^{3n}$, $z = (p_1, p_3, \dots, p_{3n}) \in \mathbb{R}^{3n}$. If the Bregman divergence is chosen as the Kullback–Leibler distance, then the Bregman projection $P_{z_m}^{\Delta^n \times \Delta^{2n}}$ can be written explicitly:

$$p_i^m = \frac{x_i^m \exp \left(-\lambda_m \left(\tilde{M} p^{m-1} \right)_i \right)}{\sum_{j=1}^n x_j^m \exp \left(-\lambda_m \left(\tilde{M} p^{m-1} \right)_j \right)}, \quad i = 1, \dots, n,$$

$$p_i^m = \frac{y_i^m \exp \left(-\lambda_m \left(\tilde{M} p^{m-1} \right)_i \right)}{\sum_{j=1}^n y_j^m \exp \left(-\lambda_m \left(\tilde{M} p^{m-1} \right)_j \right)}, \quad i = 1, \dots, 2n,$$

$$x_i^{m+1} = \frac{x_i^m \exp \left(-\lambda_m \left(\tilde{M} p^m \right)_i \right)}{\sum_{j=1}^n x_j^m \exp \left(-\lambda_m \left(\tilde{M} p^m \right)_j \right)}, \quad i = 1, \dots, n,$$

$$y_i^{m+1} = \frac{y_i^m \exp \left(-\lambda_m \left(\tilde{M} p^m \right)_i \right)}{\sum_{j=1}^n y_j^m \exp \left(-\lambda_m \left(\tilde{M} p^m \right)_j \right)}, \quad i = 1, \dots, 2n,$$

where $z = (x_1, \dots, x_n, y_1, \dots, y_{2n})$. Thus, when using this version of the algorithm, there is no need to solve auxiliary subproblems at each step, as is often required in the case of direct projection onto the feasible set. On the other hand, it should be noted that efficient computation of the exponential function on standard computer architectures is also a nontrivial task.

5.4.3 Model Construction Based on the Graph

Let us consider an example of building the model and carry out the corresponding numerical experiment.

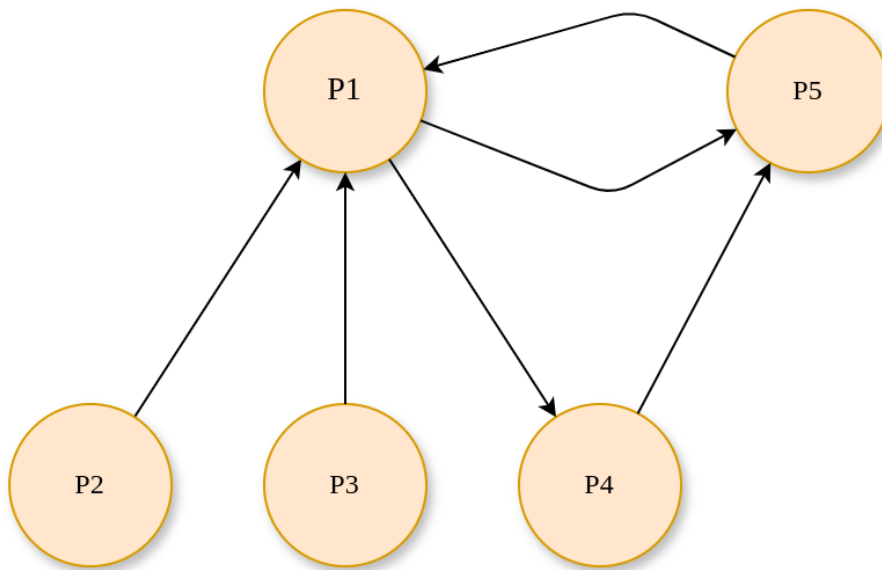


Figure 5.7: Example of a web subgraph with five pages.

If we take the graph from Figure 5.7, the weight transfer can be written mathematically as follows:

$$\begin{cases} x_1 = x_2 + x_3 + x_5, \\ x_2 = 0, \\ x_3 = 0, \\ x_4 = \frac{1}{2}x_1, \\ x_5 = \frac{1}{2}x_1 + x_4, \end{cases}$$

Accordingly, the matrix (5.7) takes the form

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 1 & 0 \end{pmatrix}.$$

We now build the matrix M from (5.8) with parameter $\alpha = 0.85$ and perform numerical experiments.

5.4.4 Numerical Experiments

For the numerical solution we apply algorithms 3.1 and 3.2 in different modifications, both for the model in the form of variational inequality (5.10).

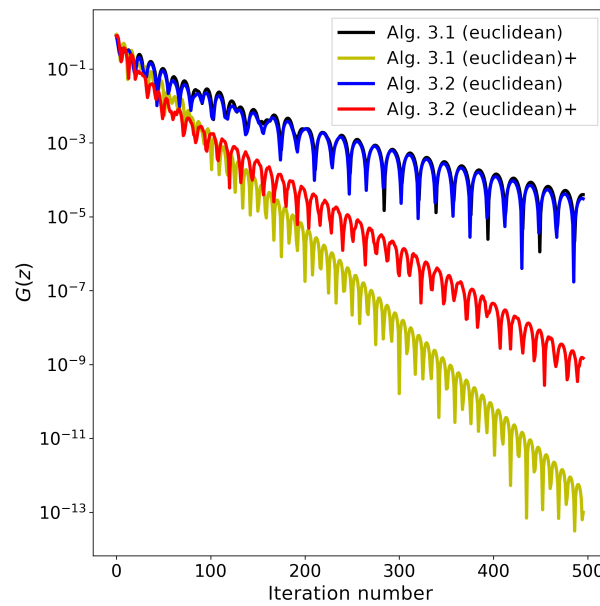


Figure 5.8: Test example of PageRank – model (5.10).

Here the projection onto the unit ball must be performed, and for (5.12), where one can either perform projection onto two simplices, or use the explicit formula due to the Kullback–Leibler divergence.

In addition to the adaptive step-size rule from Section 3, we also use an analogue of the rule (1.10) from [113], which allows temporary increase of the step-size. First, let us compare the adaptive Euclidean versions of algorithms 3.1 and 3.2 on the above test problem.

In this experiment, at each step projection is performed onto the feasible set, which is a combination of a simplex and a unit ball (see (5.10)). As we can see, in terms of convergence all algorithms behave well. At the same time, the algorithms with the modified step-size rule that allows its increase converge significantly faster. It is also clearly visible on the figure 5.8 above, that monotonicity is absent. This is consistent with theoretical results guaranteeing monotonicity only for the sequence of averaged approximations.

The variant with averaging is shown in Figure 5.9 below. As we can see, for the Cesàro averages monotonicity is violated only at the initial iterations, when the step is still too large. At the same time, the gap value after the same number of iterations is much larger. In such situations one can use averaging with a sliding window — for example, after n iterations take the average of approximations $[n/2], \dots, n$.

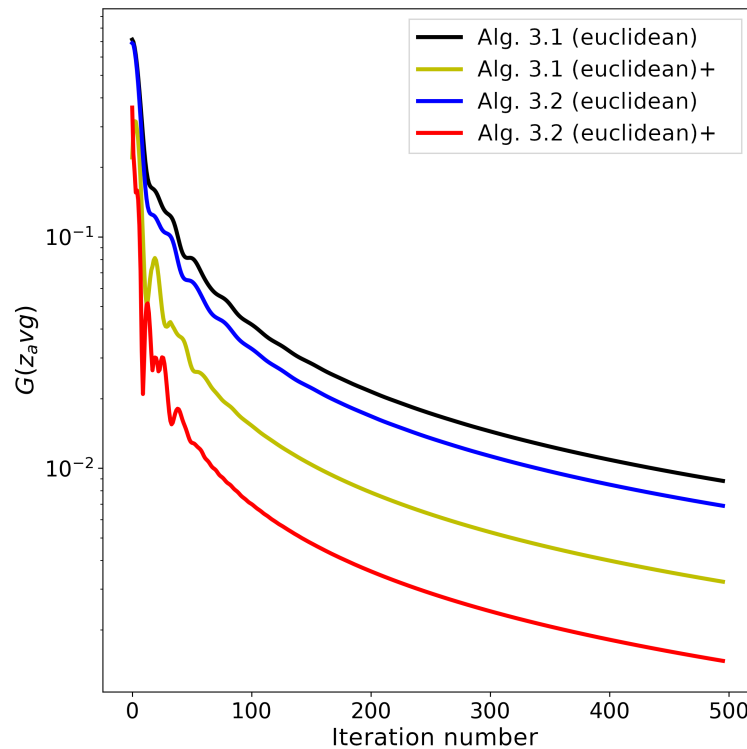


Figure 5.9: Test example of PageRank — model (5.10), with averaging.

The behavior of the same combination of algorithms after reformulating the problem into the variational inequality (5.12), where the problem dimension increases but the structure of the feasible

set is unified into the product of two probability simplices, is shown in Figure 5.10 below. The left plot shows the behavior of algorithms with Euclidean projection, and the right plot — with the explicit formula for variants with Kullback–Leibler divergence.

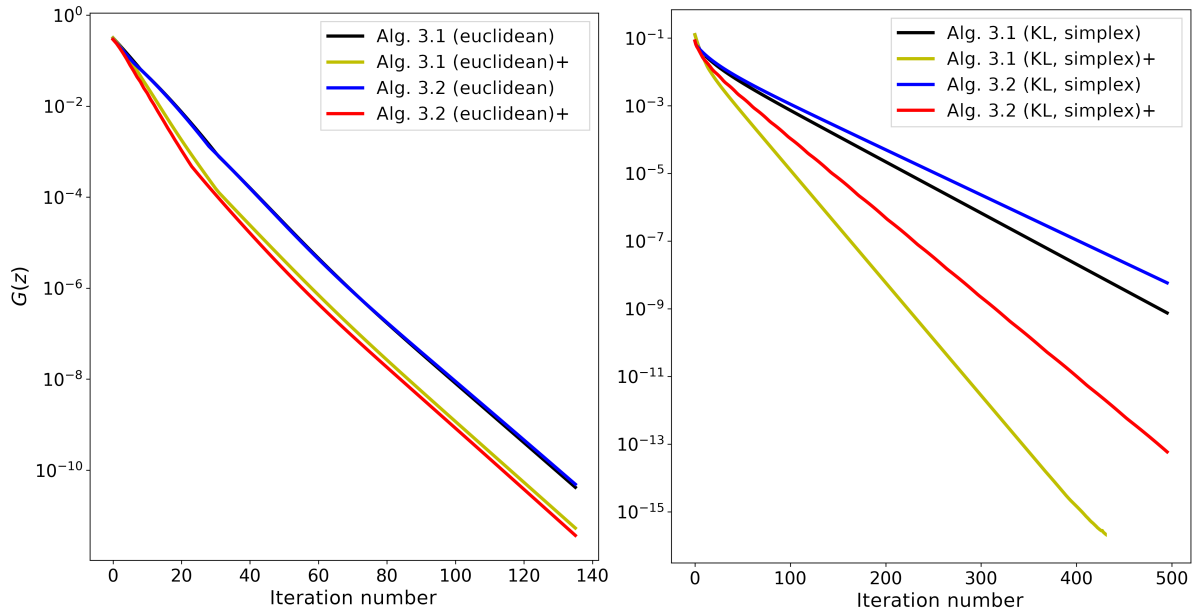


Figure 5.10: Test example of PageRank — model (5.12).

In the Euclidean versions we can see that the modified extragradient method (algorithm 3.1) and the modified extrapolation from the past method (algorithm 3.2) behave approximately the same, both for the rule that allows step-size increase, and for the adaptive procedure that only allows decrease. Moreover, the theoretically predicted behavior is observed: the step-increase heuristic accelerates convergence only at the initial stages. For the Bregman variants, with the given parameters, algorithm 3.1 (based on the extragradient method) converges faster with respect to the iteration number. However, it should be noted that in each iteration of the extrapolation from the past method only one operator evaluation is performed — therefore, the same convergence speed in terms of iteration number may mean noticeably shorter runtime to reach the same accuracy for large-scale problems (for this test problem of dimension 5 this difference is negligible).

Everywhere in the captions of convergence plots, the word “Euclidean” means that an auxiliary algorithm for projection onto the feasible set is used, while KL corresponds to explicit computation of the projection analogue. The symbol “+” denotes the step-size rule that allows step increase at some iterations. By G we denote the gap function for the corresponding approximation generated by the algorithm.

The resulting ranks for the described test example are: 0.386, 0.030, 0.030, 0.194, 0.359 (the ranks of pages from the first to the fifth). The ranking, or importance, of the pages is shown schematically in Figure 5.11 — here the circle size is proportional to the computed page rank.

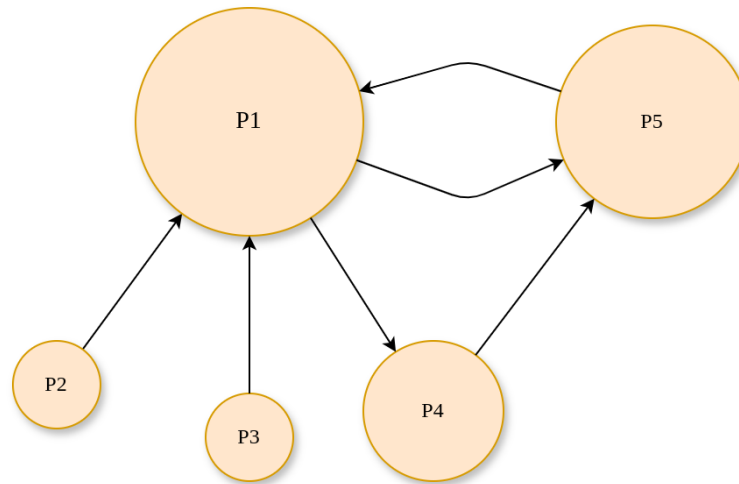


Figure 5.11: Result of PageRank computation.

Finally, let us present the results of the algorithms on a classical real dataset — the search output for the word “California” from the 2000s. This dataset contains more than 9000 nodes (web pages) and about 16,000 edges (links). The behavior of the algorithms is shown in Figure 5.12.

As we can see, convergence of the algorithm slows down as the problem dimension increases. In fact, the properties of the problem itself deteriorate — if we look at the values of the resulting ranks, they are very close to each other, which reflects the structure of the graph. At the same time, the plot shows that convergence still takes place, it just requires a larger number of iterations.

In this problem we again see that in terms of the number of iterations required to reach a given accuracy, the algorithms behave approximately the same (although the algorithms with periodic step-size increase perform noticeably better at the initial stage, which is important in practice). After 2000 iterations on the mentioned graph for the search results of the word California, an accuracy of order 10^{-5} is achieved in terms of the gap function. And while for the small test example this was not significant, in this large-scale problem algorithm 3.2 (in both modifications) turns out to be clearly more efficient in runtime — about 700 seconds versus 1300 for 2000 iterations. This is natural, since for such a graph operator computation already takes a considerable amount of time, and using a more economical algorithm gives a real advantage.

The ranks of the top 10 pages obtained after 2000 iterations of the algorithms are also presented in the table below.

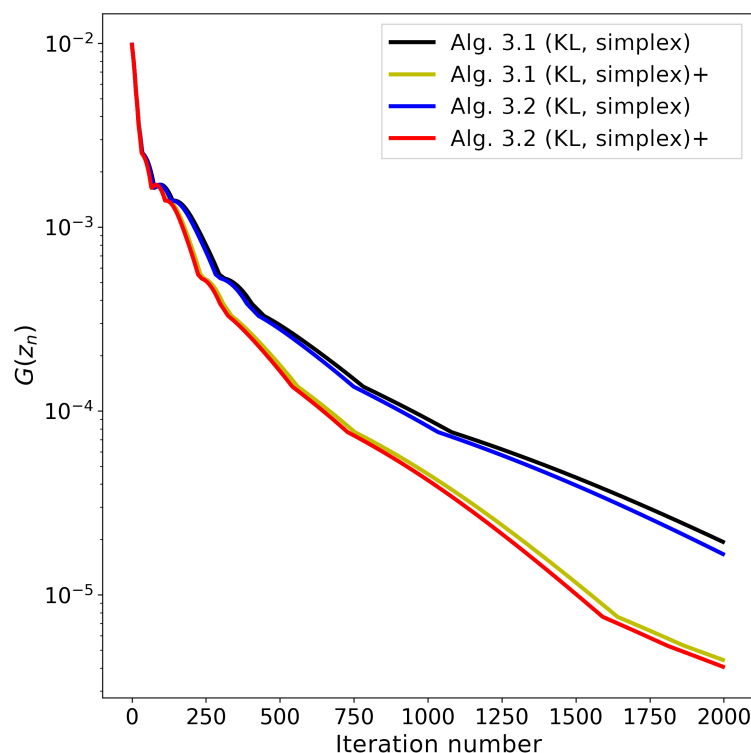


Figure 5.12: Search for the word California — ranking of results.

1	http://www.calgold.com
2	http://www.sancarlos-homes.cominfo.asp
3	http://spectacle.berkeley.edu
4	http://www.graddiv.ucr.edu
5	http://www.chico-homes.com/info.asp
6	http://www.webb.pvt.k12.ca.us
7	http://www.berkeley.edu
8	http://www.calfund.org
9	http://www.ca-probate.com
10	http://www.ppconline.com

5.5 Traffic Equilibrium Problem

As already mentioned in Chapter 1, one of the most important tasks in network economics is the optimization of traffic flows. Indeed, the problem of congestion in road networks (and transport networks of various kinds) remains relevant — and in situations where there is no systemic approach to its solution, the economic and other losses due to infrastructure inefficiency can be very large (time wasted by network participants, negative environmental impact, etc.). Moreover, similar models arise in other contexts — for example, the same model can be applied to analyze the

passage of packets in computer networks such as the Internet. Therefore, the problem of finding traffic equilibrium (the optimal distribution of traffic flows according to Wardrop's first principle) is one of the tasks that the developed software system can describe and solve.

5.5.1 Mathematical Model

Wardrop's first principle, which serves as the basis for most mathematical models of traffic equilibrium discussed in Chapter 1, can be formulated as follows:

“The travel costs on any of the routes actually used by the participants of the transport network are equal. At the same time, if any individual participant decides to switch to a route that is not currently in use, their travel costs will be higher than the current ones” [73].

(Here, in the classical version, travel costs mean travel time, but the same model can also describe fuel costs or other measures.)

Let us describe the mathematical model used in the software system.

We represent the transport network as a graph $G(V, E)$, where the set of nodes $V = \{v_1, \dots, v_n\}$ corresponds to intersections, and the set of edges $E = \{e_1, \dots, e_m\}$ corresponds to road segments between them. Among the nodes, we distinguish those that are sources of traffic (for example, nodes corresponding to residential areas of a city) and those that are destinations (such as office centers). The corresponding sets of nodes are denoted by S and T .

We assume that the demand is fixed and known — that is, for each pair of nodes ($v_i \in S, v_j \in T$) the demand d_{ij} is given, representing the number of network users who need to travel from node i to node j . For simplicity of notation and for algorithm implementation, we work with sets of node and edge indices. That is, S and T denote sets of integers containing the indices of the corresponding network nodes, and the demand can be represented in the form of a matrix $D = \{d_{ij}\}$.

The traffic flow through edge e_i is denoted by y_i . Thus, the overall flow through the network in terms of edge flows is represented by the vector $y = (y_1, \dots, y_m)$.

We also assume that for each segment of the network a travel cost function is specified, which in the general case may depend not only on the load of that segment but also on the loads of other edges in the network (for example, this is relevant when congestion occurs at intersections). The cost function for travel through edge e_i is denoted by $f_i(y)$. Thus, we obtain the vector-function $f(y) : \mathbb{R}^m \rightarrow \mathbb{R}^m$, which determines the travel costs for each edge of the network depending on the distribution of traffic flows.

Let P be the set of all paths in the network. Assume that M is the total number of such paths. Each path $p_k \in P$ is a set of edges, i.e., $p_k = \{j_{k_1}, j_{k_2}, \dots, j_{k_{L_k}}\}$. For each pair $w = (i, j), i \in S, j \in T$, one can consider the set of possible paths $P_w = \{p_{w_1}, \dots, p_{w_{k_w}}\}$, where each path corresponds

to some sequence of edges connecting the respective network nodes. Then, flows and travel costs can be expressed in terms of paths.

Denote x_i as the flow along path p_i . Given the vector $x = (x_1, \dots, x_M)$ of flows along all paths, edge flows can be computed straightforwardly. Indeed, if we define the membership of an edge in a path as

$$q_{ij} = \begin{cases} 1, & \text{if } j \in p_i, \\ 0, & \text{otherwise,} \end{cases}$$

then

$$y_j = \sum_{i=1}^M x_i q_{ij}. \quad (5.13)$$

Furthermore, if path p_i consists of edges $i_1, \dots, i_{k_i}$, then the cost function of traveling along this path is the sum of the costs of the corresponding edges. It is natural to require that the total flow along all paths connecting a given pair of nodes (i, j) equals the corresponding demand. Again, if we define the correspondence of a path to a pair of nodes as

$$w_{kij} = \begin{cases} 1, & \text{if path } k \text{ connects the pair of nodes } (i, j), \ i \in S, j \in T, \\ 0, & \text{otherwise,} \end{cases}$$

then the set of these requirements can be written as

$$\sum_{k=1}^m x_k w_{kij} = d_{ij}, \quad i \in S, \quad j \in T. \quad (5.14)$$

In addition, the flow along each path must be nonnegative:

$$x_k \geq 0, \quad k = 1, \dots, M. \quad (5.15)$$

Relations (5.14) and (5.15) define the feasible set for the problem of finding the optimal traffic flow in terms of paths.

Finally, using that $y = (y_j)$, $j = 1, \dots, m$, with $y_j = \sum_{i=1}^M x_i q_{ij}$, Wardrop's first principle can be formalized in the following form of a variational inequality [23, 70]:

$$\text{find } x^* \in \mathbb{R}^M : (f(y^*), y - y^*) \geq 0 \ \forall x \in C,$$

where y and y^* are expressed in terms of x via relation (5.13), and the set C corresponds to conditions (5.15) and (5.14).

5.5.2 Numerical Experiments

Once the problem is formulated as a variational inequality, it can be solved using the algorithms described in this work. Figure (5.13) shows the behavior of the algorithms in finding the traffic

equilibrium on a well-known test network that demonstrates the Braess paradox [14]. The network consists of five edges and has only three possible paths. For a problem of such small dimension, all algorithms perform very quickly. In the test, a linear cost function was used for the network edges. The x -axis in this case shows the distance to the exact solution, which can be found explicitly for this problem.

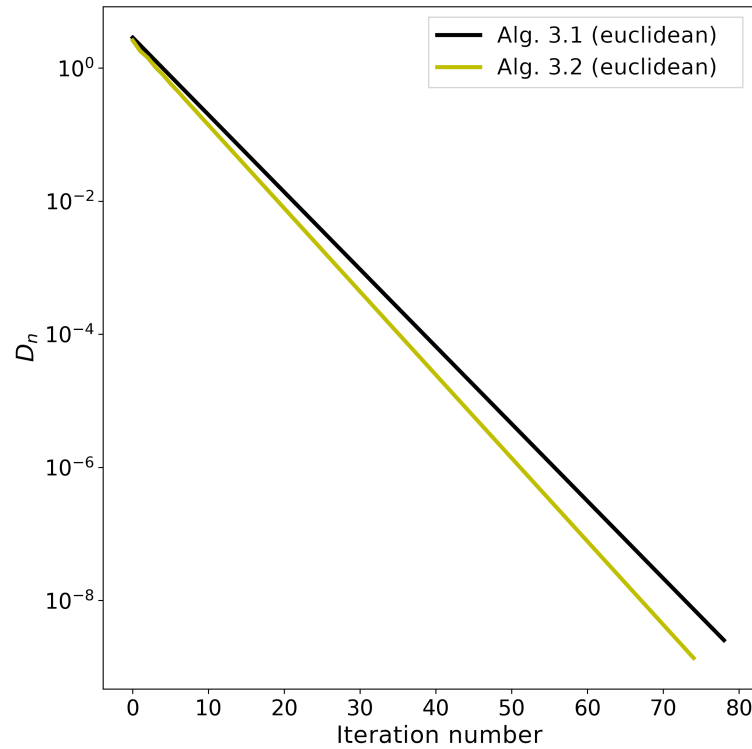


Figure 5.13: Transport network — Braess's paradox.

A more interesting experiment is solving another well-known problem, namely finding the traffic equilibrium for the Sioux Falls city street model [58]. The input network contains 24 nodes and 76 edges. The edge cost functions are of the so-called BPR type (from the Bureau of Public Roads). All parameters were taken from a public dataset of transportation networks [94]. For this problem, the set of possible paths has a dimensionality in the tens of thousands, so the system implements a heuristic that discards potentially infeasible paths, thereby reducing the problem dimension.

The behavior of the algorithms can be seen in Figure (5.14). Only the algorithms with the adaptive step-size rule that includes the possibility of step-size increase are shown, since they demonstrated significantly higher convergence speed on this problem. The feasible set of the problem is not a probability simplex, so only Euclidean versions of the algorithms are used. As can be seen from the graph, the adaptive modification of the extrapolation-from-the-past algorithm performed noticeably more effectively in terms of accuracy after 5000 iterations. It is also worth noting that for this problem the runtime for both algorithms was approximately the same (about 25 minutes),

since the operator evaluation is relatively inexpensive compared to the projection operation and the auxiliary computations related to selecting potentially promising paths.

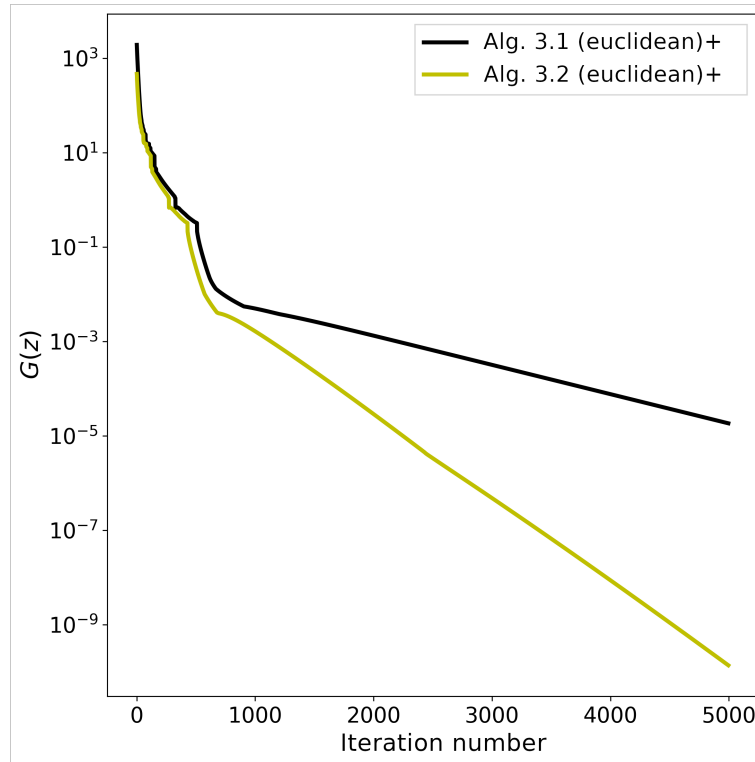


Figure 5.14: Transportation network — Sioux Falls.

All computations in this section were performed on a machine running Ubuntu Linux 20 with an Intel Core i7-1065G7 1.3–3.9GHz processor and 16GB RAM, using Python 3.10 and the numpy 1.24.3 library.

5.6 Conclusions to Chapter 5

The algorithms considered, based on the extrapolation-from-the-past and extragradient methods, allow efficient solving of practical problems — in particular, the optimization of a blood supply chain, the ranking of search engine results, and the computation of transportation equilibrium. At the same time, the proposed adaptive step-size rule avoids the complicated process of manually selecting the step size and initial parameters. Moreover, the described step recalculation procedure uses the operator and projection values at the same points as in the main step of the algorithm — thus the number of computational operations per iteration hardly increases.

From the numerical experiments, it can be concluded that although the variant of the step-size rule that allows temporary increases does not affect the theoretical estimate of the asymptotic convergence rate, experimentally it shows a clear performance gain on specific examples.

The convenient formulation of the mathematical model for the blood supply chain enables effective algorithmic implementation for this problem and can also be applied to other network economics tasks, particularly in supply network optimization.

It is also worth noting that the developed software system for applying algorithms to solve variational inequalities significantly reduces the time required both to obtain solutions to problems formulated as variational inequalities and to conduct numerical experiments and compare the behavior of extragradient-type algorithms. In particular, by having the complete execution history and final results (in terms of runtime, number of iterations, error measures, and the step size at the last iteration), the researcher can draw conclusions and make assumptions about the properties of the algorithm and possible ways to improve it.

As a future direction of development, it would be useful to add other types of algorithms to the software suite, including specialized methods for particular classes of problems. Researchers should also be provided with an easier way to connect their own algorithms and problems to the system — including specifying the programmatic interface that each problem and algorithm should implement. Furthermore, it would be beneficial to enable adding implementations of algorithms and problem descriptions not only in Python but also in other programming languages.

The results of this chapter were published in [33, 35].

Concluding Remarks

This thesis presented a theoretical and practical investigation of **algorithms** for variational inequalities and **their applications** to optimization problems over networks.

New and **modified** algorithms have been proposed, together with a **software framework** for their application and corresponding computational experiments. The software includes both test problems and network economy problems, offering the flexibility to modify parameters and easily **add new problems and algorithms**. For all algorithms described in the dissertation, the corresponding **convergence theorems** have been proven.

The following **results** have been obtained:

1. A new adaptive algorithm for solving variational inequalities on the set of fixed points of an auxiliary operator in a Hilbert space has been proposed and its convergence proven.
2. An adaptive algorithm based on Bregman divergence (instead of Euclidean distance) has been proposed and theoretically justified; this method does not require additional projections or operator evaluations for step-size selection.
3. A software framework has been created that allows combining a problem modeled as a variational inequality with a set of algorithms of the corresponding class. It enables numerical experiments with detailed execution logs for each algorithm, as well as a graphical comparison of their behavior.
4. Using the developed software framework, numerical experiments were conducted to analyze the behavior of the proposed adaptive algorithms for the Blood Supply Chain optimization problem, the Traffic Equilibrium problem, and the PageRank problem. The experiments demonstrated the applicability of the algorithms to the mentioned problems and the efficiency of the adaptive step-size rule, particularly the advantage of the variant with step-size increase.

Implementation and applications

1. Within the software framework, a graphical interface (Web UI) has been implemented for editing and solving the Blood Supply Chain optimization problem, with possibility of implementation for other problems within the same architecture.

2. Selected results of this dissertation have been incorporated into the educational process within the Applied Mathematics program at the Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv.
3. The theoretical results obtained can be used for the development of more efficient approaches to solving problems modeled by variational inequalities. The software framework can be applied both to solve real problems within the implemented models and to conduct scientific research in the field. It is published as an open-source software, allowing researchers to add their own problems and algorithms while reusing existing functionality for computational experiments.

Author's Scientific Achievements

Papers in scientific journals

1. V. Semenov and S. Denisov. “Convergence of the Method of Extrapolation from the Past for Variational Inequalities in Uniformly Convex Banach Spaces.” In: Cybernetics and Systems Analysis 58.4 (2022), pp. 564–575.

Contribution Algorithms design and the main convergence theorem.

2. V. Semenov, S. Denisov and A. Kravets. Adaptive Two-Stage Bregman Method for Variational Inequalities, Cybernetics and Systems Analysis, 2021, 57(6), pp. 959–967.

Contribution Adaptive version of the extrapolation from the past algorithm with Bregman divergence and it's convergence proof.

3. S. Denisov, V. Semenov and P. Stetsyuk. “Bregman Extragradient Method with Monotone Rule of Step Adjustment.” In: Cybernetics and Systems Analysis 55.3 (2019), pp. 377–383.

Contribution Adaptive step-size rule and main convergence theorem.

4. S. Denisov, D. Nomirovskii, B. Rublyov and V. Semenov. Convergence of extragradient algorithm with monotone step size strategy for variational inequalities and operator equations, Journal of Automation and Information Sciences, 2019, 51(6), pp. 16–29.

Contribution Algorithms for variational inequality with extra restrictions and main convergence theorem.

International conferences with published materials

1. S. Denysov and V. Semenov. Traffic Equilibrium Modeling with Modern Algorithms for Variational Inequalities. In: Kazymyr, V., et al. Mathematical Modeling and Simulation of Systems. MODS 2024. Lecture Notes in Networks and Systems, vol 1391. Springer, Cham. DOI: 10.1007/978-3-031-90735-7_5

Contribution Model adaptation, software implementation and computational experiments.

2. S. Denysov and V. Semenov. Optimizing network economics problem with adaptive algorithms for variational inequalities. Conference Proceedings of Dynamical System Modeling and Stability Investigation (DSMSI-2023), December 05-07, 2023, Kyiv, Ukraine, pp. 157–167.

Contribution Model adaptation, software implementation and computational experiments.

3. S. Denysov and V. Semenov. Convergence of adaptive operator extrapolation method for operator inclusions in Banach Spaces. Workshop Proceedings of The Fifth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2022), Zaporizhzhia, Ukraine, May 12, 2022, CEUR Workshop Proceedings, 2022, Vol-3137, pp. 186–199.

Contribution Model adaptation, software implementation and computational experiments.

4. S. Denysov and V. Semenov. Theoretical bound of the complexity of some extragradient-type algorithms for variational inequalities in Banach spaces. VIII International Scientific Conference «Information Technology and Implementation» (IT&I-2021), Kyiv, Ukraine, December 1-3, 2021, CEUR Workshop Proceedings, 2022, Vol-3179, pp. 144–155.

Contribution Results for the extrapolation from the past algorithm.

5. S. Denysov and V. Semenov. Adaptive Variants of the Extrapolation from the Past Method and the Operator Extrapolation Method. Mathematical Modeling and Simulation of Systems. MODS 2022. Lecture Notes in Networks and Systems, vol 667. Springer, Cham, pp. 49–60.

Contribution Adaptive extrapolation from the past algorithm with corresponding convergence proof. Computational experiments.

6. S. Denisov, V. Semenov and Y. Vedel. A Novel Algorithm with Self-adaptive Technique for Solving Variational Inequalities in Banach Spaces. OPTIMA-2021, Petrovac, Montenegro, 27.09–1.10, 2021. Advances in Optimization and Applications. Communications in Computer and Information Science, Springer, Cham, vol 1514., pp. 50–64.

Contribution Modification of the algorithm for variational inequality, computational experiments.

7. S. Denisov and V. Semenov. Convergence of Adaptive Forward-Reflected-Backward Algorithm for Solving Variational Inequalities, CEUR Workshop Proceedings, 2021, 3106, pp. 116–127.

Contribution Adaptive step size rule for the algorithm.

References

- [1] Y. Alber. "Metric and generalized projection operators in Banach spaces: properties and applications." In: *Theory and Applications of Nonlinear Operators of Accretive and Monotone Type*. Vol. 178. New York: Dekker, 1996, pp. 15–50.
- [2] Y. Alber and I. Ryazantseva. *Nonlinear Ill Posed Problems of Monotone Type*. Dordrecht: Springer, 2006. 410 pp.
- [3] A.S. Antipin. "On a Method for Convex Programs Using a Symmetrical Modification of the Lagrange Function." In: *Ekonomika i Matematicheskie Metody* 6 (1976), pp. 1164–1173.
- [4] K. Aoyama and F. Kohsaka. "Strongly relatively nonexpansive sequences generated by firmly nonexpansive-like mappings." In: *Fixed Point Theory and Applications* (2014). DOI: 10.1186/1687-1812-2014-95.
- [5] M. Attari, S. Pasandideh, and S. Niaki. "A hybrid robust stochastic programming for a bi-objective blood collection facilities problem (Case study: Iranian blood transfusion network)." In: *Journal of Industrial and Production Engineering* 36 (2019), pp. 154–167. DOI: 10.1080/21681015.2019.1645747.
- [6] M. Bačák. *Convex Analysis and Optimization in Hadamard Spaces*. Berlin; Boston: De Gruyter, 2014, p. 185.
- [7] C. Baiocchi and A. Capello. *Variational and Quasivariational Inequalities. Applications to Free Boundary Problems*. New York: Wiley, 1984, p. 448.
- [8] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Berlin; Heidelberg; New York: Springer, 2011. 408 pp. DOI: 10.1007/978-3-319-48311-5.
- [9] B. Beauzamy. *Introduction to Banach Spaces and Their Geometry*. Amsterdam: North-Holland, 1985. 307 pp.
- [10] A. Beck. *First-Order Methods in Optimization*. Philadelphia: Society for Industrial and Applied Mathematics, 2017. 479 pp.
- [11] A. Beck and M. Teboulle. "Mirror descent and nonlinear projected subgradient methods for convex optimization." In: *Operations Research Letters* 31.3 (2003), pp. 167–175.
- [12] M. Beckmann, C. McGuire, and C. Winsten. *Studies in the Economics of Transportation*. Cowles Commission for Research in Economics, 1956, p. 232.

- [13] A. Bensoussan. ``Points de Nash dans le cas de fonctionnelles quadratiques et jeux différentiels linéaires à N personnes." French. In: *SIAM Journal on Control* 12 (1974), pp. 460–499.
- [14] D. Braess, A. Nagurney, and T. Wakolbinger. ``On a Paradox of Traffic Planning." In: *Transportation Science* 39.4 (2005), pp. 446–450.
- [15] L. Bregman. ``The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming." In: *USSR Computational Mathematics and Mathematical Physics* 7.3 (1967), pp. 200–217.
- [16] H. Brezis and L. Evans. ``A variational inequality approach to the Bellman--Dirichlet equation for two elliptic operators." In: *Archive for Rational Mechanics and Analysis* 71 (1979), pp. 1–13.
- [17] S. Brin and L. Page. ``The anatomy of a large-scale hypertextual web search engine." In: *Computer Networks and ISDN Systems* 30.1--7 (1998), pp. 107–117. DOI: 10 . 1016 / S0169 - 7552 (98) 00110 -X.
- [18] Y. Censor, A. Gibali, and S. Reich. ``The subgradient extragradient method for solving variational inequalities in Hilbert space." In: *Journal of Optimization Theory and Applications* 148 (2011), pp. 318–335.
- [19] L. Chabak, V. Semenov, and Y. Vedel. ``A New Non-Euclidean Proximal Method for Equilibrium Problems." In: *Recent Developments in Data Science and Intelligent Analysis of Information*. Ed. by O. Chertov et al. Vol. 836. Advances in Intelligent Systems and Computing. Cham: Springer, 2019, pp. 50–58.
- [20] V. Colao et al. ``Equilibrium problems in Hadamard manifolds." In: *Journal of Mathematical Analysis and Applications* 388 (2012), pp. 61–77.
- [21] A. A. Cournot. *Researches into the Mathematical Principles of the Theory of Wealth*. English translation of the 1838 edition. London: Macmillan, 1897.
- [22] S. Dafermos. ``Relaxation algorithms for the general asymmetric traffic equilibrium problem." In: *Transportation Science* 16 (1982), pp. 231–240.
- [23] S. Dafermos. ``Traffic equilibria and variational inequalities." In: *Transportation Science* 14.1 (1982), pp. 42–54.
- [24] S. Denisov and V. Semenov. ``Convergence of Adaptive Forward-Reflected-Backward Algorithm for Solving Variational Inequalities." In: *II International Scientific Symposium ``Intelligent Solutions" (IntSol-2021)*. Vol. 3106. CEUR Workshop Proceedings. 2021, pp. 116–127.
- [25] S. Denisov, V. Semenov, and Y. Vedel. ``Adaptive Algorithm for Variational Inequality Problem Over the Set of Solutions the Equilibrium Problem." In: *2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT)*. Kyiv, Ukraine, 2020, pp. 325–329.
- [26] S. Denisov, V. Semenov, and Y. Vedel. ``A novel adaptive method for operator inclusions." In: *Modeling, Control and Information Technologies: Proceedings of the V International Scientific-Practical Conference on Modeling, Control and Information Technology*. Rivne, Nov. 4–6, 2021, pp. 33–35.

- [27] S. Denisov, V. Semenov, and Y. Vedel. "A Novel Algorithm with Self-adaptive Technique for Solving Variational Inequalities in Banach Spaces." In: *Advances in Optimization and Applications (OPTIMA-2021)*. Vol. 1514. Communications in Computer and Information Science. Cham: Springer, 2021, pp. 50–64.
- [28] S. V. Denisov, V. V. Semenov, and L. M. Chabak. "Convergence of the Modified Extragradient Method for Variational Inequalities with Non-Lipschitz Operators." In: *Cybernetics and Systems Analysis* 51.5 (2015), pp. 757–765.
- [29] S. V. Denisov, V. V. Semenov, and P. I. Stetsyuk. "Bregman Extragradient Method with Monotone Rule of Step Adjustment." In: *Cybernetics and Systems Analysis* 55.3 (2019), pp. 377–383.
- [30] S. V. Denisov et al. "A New Mirror-prox Algorithm For Variational Inequalities." Ukrainian. In: *Журнал обчисл. та прикл. матем.* 1 (124) (2017), pp. 15–29.
- [31] S. V. Denisov et al. "Convergence of Extragradient Algorithm with Monotone Step Size Strategy for Variational Inequalities and Operator Equations." In: *Journal of Automation and Information Sciences* 51.6 (2019), pp. 12–24.
- [32] S. Denysov and V. Semenov. "Theoretical bound of the complexity of some extragradient-type algorithms for variational inequalities in Banach spaces." In: *VIII International Scientific Conference "Information Technology and Implementation" (IT&I-2021)*. Vol. 3179. CEUR Workshop Proceedings. Kyiv, Ukraine, Dec. 1–3, 2021, pp. 144–155.
- [33] S. Denysov and V. Semenov. "Adaptive Variants of the Extrapolation from the Past Method and the Operator Extrapolation Method." In: *Mathematical Modeling and Simulation of Systems. MODS 2022*. Vol. 667. Lecture Notes in Networks and Systems. Cham: Springer, 2022, pp. 49–60.
- [34] S. Denysov and V. Semenov. "Convergence of adaptive operator extrapolation method for operator inclusions in Banach Spaces." In: *The Fifth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2022)*. Vol. 3137. CEUR Workshop Proceedings. Zaporizhzhia, Ukraine, May 12, 2022, pp. 186–199.
- [35] S. Denysov and V. Semenov. "Optimizing network economics problem with adaptive algorithms for variational inequalities." In: *Conference Proceedings of Dynamical System Modeling and Stability Investigation (DSMSI-2023)*. Kyiv, Ukraine, Dec. 5–7, 2023, pp. 157–167.
- [36] Serhii Denysov and Vladimir Semenov. "Traffic Equilibrium Modeling with Modern Algorithms for Variational Inequalities." In: *Mathematical Modeling and Simulation of Systems*. Ed. by Volodymyr Kazymyr et al. Cham: Springer Nature Switzerland, 2025, pp. 56–67. DOI: 10.1007/978-3-031-90735-7_5.
- [37] J. Diestel. *Geometry of Banach Spaces - Selected Topics*. Heidelberg: Springer Berlin, 1975. 289 pp. DOI: 10.1007/BFb0082079.
- [38] G. Duvaut and J.-L. Lions. *Inequalities in Mechanics and Physics*. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer, 2011, p. 400. DOI: 10.1007/978-3-642-66165-5.

- [39] E. Fernández et al. "Network equilibrium models with combined modes." In: *Transportation Science* 28 (1994), pp. 182–192.
- [40] G. Fichera. "Problemi elastostatici con vincoli unilaterali: il problema di Signorini con ambigue condizioni al contorno." Italian. In: *Memorie della Accademia Nazionale dei Lincei, Classe di Scienze Fisiche, Matematiche e Naturali* 8.7 (1964), pp. 91–140.
- [41] I. Gemp. "From Optimization to Equilibration: Understanding an Emerging Paradigm in Artificial Intelligence and Machine Learning." PhD thesis. University of Massachusetts Amherst, 2019. doi: 10.7275/13780178.
- [42] I. Gemp and S. Mahadevan. *Global Convergence to the Equilibrium of GANs using Variational Inequalities*. 2018. doi: 10.48550/arXiv.1808.01531. arXiv: 1808.01531 [cs.LG].
- [43] I. Gemp and S. Mahadevan. *Finding Equilibria in Large Games using Variational Inequalities*. https://imgemp.github.io/pubs/aaai_ss_2015.pdf. Accessed 2025-08-25.
- [44] I. Gemp and S. Mahadevan. *Modeling Context in Cognition Using Variational Inequalities*. <https://cdn.aaai.org/ocs/9202/9202-40001-1-PB.pdf>. Accessed 2025-08-25.
- [45] G. Gidel et al. *A Variational Inequality Perspective on Generative Adversarial Networks*. 2018. arXiv: 1802.10551.
- [46] D. F. Gleich. "PageRank Beyond the Web." In: *SIAM Review* 57.3 (2015), pp. 321–363.
- [47] A. A. Goldstein. "Convex programming in Hilbert space." In: *Bulletin of the American Mathematical Society* 70 (1964), pp. 709–710.
- [48] I. Goodfellow et al. "Generative Adversarial Networks." In: *Advances in Neural Information Processing Systems 27 (NeurIPS 2014)*. 2014, pp. 2672–2680.
- [49] B. Halpern. "Fixed points of nonexpanding maps." In: *Bulletin of the American Mathematical Society* 73 (1967), pp. 957–961.
- [50] P. T. Harker. "A variational inequality approach for the determination of oligopolistic market equilibrium." In: *Mathematical Programming* 30 (1984), pp. 105–111. doi: 10.1007/BF02591802.
- [51] H. Iiduka and W. Takahashi. "Weak convergence of a projection algorithm for variational inequalities in a Banach space." In: *Journal of Mathematical Analysis and Applications* 339.1 (2008), pp. 668–679. doi: 10.1016/j.jmaa.2007.07.019.
- [52] H. Khatibzadeh and V. Mohebbi. "Approximating solutions of equilibrium problems in Hadamard spaces." In: *Miskolc Mathematical Notes* 20.1 (2019), pp. 281–297. doi: 10.18514/MMN.2019.2361.
- [53] E. N. Khobotov. "Modification of the extra-gradient method for solving variational inequalities and certain optimization problems." In: *USSR Computational Mathematics and Mathematical Physics* 27.5 (1987), pp. 120–127. doi: 10.1016/0041-5553(87)90058-9.

- [54] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. Originally published: Academic Press, 1980. Philadelphia: Society for Industrial and Applied Mathematics, 2000, p. 313.
- [55] I. V. Konnov. *Combined relaxation methods for variational inequalities*. Berlin; Heidelberg; New York: Springer--Verlag, 2001. 181 pp.
- [56] I. V. Konnov. *Combined relaxation methods for variational inequalities*. Berlin; Heidelberg; New York: Springer--Verlag, 2001. 181 pp. doi: 10.1007/978-3-642-56886-2.
- [57] G. Korpelevich. "An extragradient method for finding saddle points and other problems." In: *Ekonomika i Matematicheskie Metody* 4 (1976), pp. 747–756.
- [58] L. LeBlanc, E. Morlok, and W. Pierskalla. "Transportation Research." In: *Transportation Research* 9 (1975), pp. 309–318. doi: 10.1016/0041-1647(75)90030-1.
- [59] E. Levitin and B. Polyak. "Constrained Minimization Methods." In: *USSR Computational Mathematics and Mathematical Physics* 6 (Dec. 1966), pp. 1–50. doi: 10.1016/0041-5553(66)90114-5.
- [60] H. Lewy and G. Stampacchia. "On the regularity of the solution of variational inequality." In: *Communications on Pure and Applied Mathematics* 22 (1969), pp. 153–188.
- [61] J. Lions and G. Stampacchia. "Variational inequalities." In: *Communications on Pure and Applied Mathematics* 20 (1967), pp. 493–519.
- [62] J.-L. Lions. *Some Methods of Solving Non-Linear Boundary Value Problems*. Paris: Dunod-Gauthier-Villars, 1969.
- [63] J.-L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Heidelberg: Springer Berlin, 1971, p. 400.
- [64] L. Liu and E. Boyce. "Variational inequality formulation of the system-optimal travel choice problem and efficient congestion tolls for a general transportation network with multiple time periods." In: *Regional Science and Urban Economics* 32.5 (2002), pp. 627–650. doi: 10.1016/S0166-0462(01)00095-3.
- [65] S. I. Lyashko and V. V. Semenov. "A New Two-Step Proximal Algorithm of Solving the Problem of Equilibrium Programming." In: *Optimization and Its Applications in Control and Data Sciences*. Ed. by Boris Goldengorin. Vol. 115. Springer Optimization and Its Applications. Cham: Springer, 2016, pp. 315–325.
- [66] S. I. Lyashko, V. V. Semenov, and T. A. Voitova. "Low-cost modification of Korpelevich's methods for monotone equilibrium problems." In: *Cybernetics and Systems Analysis* 4 (2011), pp. 631–639.
- [67] Y. Malitsky. "Projected Reflected Gradient Methods for Monotone Variational Inequalities." In: *SIAM Journal on Optimization* 25 (2015), pp. 502–520.
- [68] Y. V. Malitsky and V. V. Semenov. "An extragradient algorithm for monotone variational inequalities." In: *Cybernetics and Systems Analysis* 50 (2014), pp. 271–277.

- [69] P. Marcotte. "Application of Khobotov's algorithm to variational inequalities and network equilibrium problems." In: *INFOR: Information Systems and Operational Research* 29 (1991), pp. 258–270.
- [70] A. Nagurney. *Network Economics: A Variational Inequality Approach*. Dordrecht: Kluwer Academic Publishers, 1999, p. 325.
- [71] A. Nagurney. "Oligopolistic market equilibrium." In: *Encyclopedia of Optimization*. Vol. 5. Dordrecht: Kluwer Academic Publishers, 2001, pp. 119–122.
- [72] A. Nagurney. *Supply Chain Network Economics: Dynamics of Prices, Flows, and Profits*. Edward Elgar Publishing, 2008. DOI: 10.1111/j.1467-9787.2008.00567_4.x.
- [73] A. Nagurney. "Network Economics." In: *Handbook of Computational Econometrics*. Chichester, UK: John Wiley & Sons, 2009, pp. 429–486.
- [74] A. Nagurney and J. Dong. "A multiclass, multicriteria traffic network equilibrium model with elastic demand." In: *Transportation Research Part B: Methodological* 36.5 (2002), pp. 445–469.
- [75] A. Nagurney, A. Masoumi, and L. Nagurney. *Networks Against Time: Supply Chain Analytics for Perishable Products*. SpringerBriefs in Optimization. New York, NY: Springer, 2013. DOI: 10.1007/978-1-4614-6277-4.
- [76] A. Nagurney, A. Masoumi, and M. Yu. "Supply Chain Network Operations Management of a Blood Banking System with Cost and Risk Minimization." In: *Computational Management Science* 9 (2012), pp. 205–231.
- [77] A. Nagurney and L. Nagurney. "Sustainable Supply Chain Network Design: A Multicriteria Perspective." In: *International Journal of Sustainable Engineering* 3 (2010), pp. 189–197.
- [78] A. Nemirovski. "Prox-method with rate of convergence $O(1/T)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex--concave saddle point problems." In: *SIAM Journal on Optimization* 15.1 (2004), pp. 229–251.
- [79] A. Nemirovski and D. Yudin. "Efficient methods for solving large-scale convex programming problems." In: *Ekonomika i Matematicheskie Metody* 15.1 (1979), pp. 133–152.
- [80] D. A. Nemirovskii, V. V. Rublyov, and V. V. Semenov. "Convergence of Two-Stage Method with Bregman Divergence for Solving Variational Inequalities." In: *Cybernetics and Systems Analysis* 55 (2019), pp. 359–368.
- [81] L. Page et al. *The PageRank Citation Ranking: Bringing Order to the Web*. Tech. rep. Stanford InfoLab, 1999.
- [82] L. Popov. "A modification of the Arrow--Hurwicz method for search of saddle points." In: *Mathematical Notes of the Academy of Sciences of the USSR* 28.5 (1980), pp. 845–848. DOI: 10.1007/BF01141092.
- [83] L.D. Popov. "A modification of the Arrow-Hurwicz method for search of saddle points." In: *Mathematical Notes of the Academy of Sciences of the USSR* 28.5 (1980), pp. 845–848. DOI: 10.1007/BF01141092.

- [84] S. Saberi, A. Nagurney, and T. Wolf. "A Network Economic Game Theory Model of a Service-Oriented Internet with Price and Quality Competition in Both Content and Network Provision." In: *Service Science* 6.4 (2014), pp. 229–250.
- [85] B. Scholz-Reiter, F. Wirth, S. Dashkovskiy, et al. *Application of the PageRank algorithm for ranking locations of a production network*. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5a7a138ae52e2238eb8b1f67529b251ce491ff34>. Accessed 2025-08-25. 2009.
- [86] V. Semenov and Y. Vedel. "Convergence of adaptive methods for equilibrium problems in Hadamard spaces." In: *Proceedings of the 7th International Conference "Information Technology and Interactions" (IT&I-2020). Workshops Proceedings*. Vol. 2845. CEUR Workshop Proceedings. Kyiv, Ukraine, Dec. 2–3, 2020, pp. 321–335.
- [87] V. Semenov et al. "Convergence of the Operator Extrapolation Method for Variational Inequalities in Banach Spaces." In: *Cybernetics and Systems Analysis* 58.5 (2022), pp. 740–753.
- [88] V. V. Semenov. "A Version of the Mirror Descent Method to Solve Variational Inequalities." In: *Cybernetics and Systems Analysis* 53 (2017), pp. 234–243.
- [89] V. V. Semenov. "Modified Extragradient Method with Bregman Divergence for Variational Inequalities." In: *Journal of Automation and Information Sciences* 50.8 (2018), pp. 26–37.
- [90] V. V. Semenov and S. V. Denisov. "Convergence of the Method of Extrapolation from the Past for Variational Inequalities in Uniformly Convex Banach Spaces." In: *Cybernetics and Systems Analysis* 58.4 (2022), pp. 564–575.
- [91] V. V. Semenov, S. V. Denisov, and A. V. Kravets. "Adaptive Two-Stage Bregman Method for Variational Inequalities." In: *Cybernetics and Systems Analysis* 57.6 (2021), pp. 959–967.
- [92] A. Signorini. "Sopra alcune questioni di Elastostatica." In: *Atti della Società Italiana per il Progresso delle Scienze* (1933).
- [93] G. Stampacchia. "Le probleme de Dirichlet pour les équations elliptiques du second ordre à coefficients discontinus." French. In: *Annales de l'Institut Fourier* 15.1 (1965), pp. 189–257.
- [94] *TransportationNetworks*. <https://github.com/bstabler/TransportationNetworks>. Accessed 2025-08-25.
- [95] P. Tseng. "A modified forward-backward splitting method for maximal monotone mappings." In: *SIAM Journal on Control and Optimization* 38 (2000), pp. 431–446.
- [96] V. V. Vasin. "Iterative methods for solving ill-posed problems with a priori information in Hilbert spaces." In: *USSR Computational Mathematics and Mathematical Physics* 28.4 (1988), pp. 6–13.
- [97] V. V. Vasin. "Iterative processes of Fejér type in ill-posed problems with a priori information." In: *Russian Mathematics* 53 (2009), pp. 1–20. doi: 10.3103/S1066369X09020017.
- [98] V. V. Vasin and I. I. Eremin. *Operators and Iterative Processes of Fejér Type. Theory and Applications*. Berlin; New York: Walter de Gruyter, 2009. 155 pp.

- [99] Y. Vedel and V. Semenov. "Adaptive Extraproximal Algorithm for the Equilibrium Problem in Hadamard Spaces." In: *Optimization and Applications. OPTIMA 2020*. Ed. by N. Olenov et al. Vol. 12422. Lecture Notes in Computer Science. Cham: Springer, 2020, pp. 287–300.
- [100] Y. I. Vedel, S. V. Denisov, and V. V. Semenov. "Regularized Adaptive Extra-Proximal Algorithm for Equilibrium Problem in Hadamard Spaces." In: *Journal of Automation and Information Sciences* 52.9 (2020), pp. 12–26.
- [101] Y. I. Vedel, S. V. Denisov, and V. V. Semenov. "An Adaptive Algorithm for the Variational Inequality Over the Set of Solutions of the Equilibrium Problem." In: *Cybernetics and Systems Analysis* 57.1 (2021), pp. 91–100.
- [102] Y. I. Vedel, G. V. Sandrakov, and V. V. Semenov. "An Adaptive Two-Stage Proximal Algorithm for Equilibrium Problems in Hadamard Spaces." In: *Cybernetics and Systems Analysis* 56.6 (2020), pp. 978–989.
- [103] Y. I. Vedel et al. "Adaptive Extraproximal Algorithm for the Equilibrium Problem in the Hadamard Spaces." In: *Journal of Automation and Information Sciences* 52.8 (2020), pp. 46–58.
- [104] Y. I. Vedel et al. "Convergence of a Two-Stage Proximal Algorithm for the Equilibrium Problem in Hadamard Spaces." In: *Cybernetics and Systems Analysis* 56.5 (2020), pp. 784–792.
- [105] D. A. Verlan, V. V. Semenov, and L. M. Chabak. "A Strongly Convergent Modified Extragradient Method for Variational Inequalities with Non-Lipschitz Operators." In: *Journal of Automation and Information Sciences* 47.7 (2015), pp. 31–46.
- [106] Haiyan Wang, W. Feng, and K. Xu. *Modeling Information Diffusion in Online Social Networks with Partial Differential Equations*. Surveys and Tutorials in the Applied Mathematical Sciences. Cham: Springer, 2020, p. 144. doi: 10.1007/978-3-030-38852-2.
- [107] T. Wang. "Dynamics of an epidemic model with spatial diffusion." In: *Physica A* 409 (2014), pp. 119–129. doi: 10.1016/j.physa.2014.04.028.
- [108] J. G. Wardrop. "Some theoretical aspects of road traffic research." In: *Proceedings of the Institution of Civil Engineers, Part II* 1 (1952), pp. 325–378. doi: 10.1680/ipeds.1952.11362.
- [109] Z. Wu and W. Lam. "A combined modal split and stochastic assignment model for congested networks with motorized and non-motorized modes." In: *Transportation Research Record* 1831 (2003), pp. 57–64.
- [110] H. K. Xu. "Inequalities in Banach spaces with applications." In: *Nonlinear Analysis* 16.12 (1991), pp. 1127–1138.
- [111] J. Yang and H. W. Liu. "Strong convergence result for solving monotone variational inequalities in Hilbert space." In: *Numerical Algorithms* (2018). doi: 10.1007/s11075-018-0504-4.
- [112] X. Zhang et al. "A Traffic Equilibrium Model for Multi-Modal Networks with Uncertain Demands." In: *Applied Sciences* 13.23 (2023). doi: 10.3390/app132312841.

- [113] Я. Ведель, С. Денисов, and В. Семенов. ``Збіжність адаптивних екстрапроксимальних алгоритмів для задач про рівновагу в просторах Адамара." Ukrainian. In: *Журнал обчислювальної та прикладної математики* 1 (2022), pp. 62–82.
- [114] Я. І. Ведель, С. В. Денисов, and В. В. Семенов. ``Збіжність брегманівського екстраградієнтного методу." Ukrainian. In: *Доповіді НАН України* 5 (2019), pp. 18–23.
- [115] Я. І. Ведель and В. В. Семенов. ``Адаптивні алгоритми для задач про рівновагу в просторах Адамара." Ukrainian. In: *Доповіді НАН України* 8 (2020), pp. 26–34.
- [116] С. В. Денисов, В. В. Семенов, and О. С. Харьков. ``Слабка збіжність методу операторної екстраполяції для варіаційних нерівностей в рівномірно опуклих банахових просторах." Ukrainian. In: *Журнал обчислювальної та прикладної математики* 2 (2022), pp. 42–49.
- [117] С. В. Денисов, В. В. Семенов, and О. С. Харьков. ``Слабка збіжність методу операторної екстраполяції для варіаційних нерівностей в банахових просторах." Ukrainian. In: *Матеріали 7-ї міжнародної наукової конференції ``Моделювання і оптимізація у транспорті та логістиці", присвяченої 85-річчю з дня народження академіка НАН України Наума Зуселевича Шора. Київ–Кишинів–Баку, 2022*, pp. 262–271.
- [118] В. В. Семенов, Д. С. Сірик, and О. С. Харьков. ``Збіжність методу операторної екстраполяції." Ukrainian. In: *Доповіді НАН України* 4 (2021), pp. 28–35.
- [119] В. В. Семенов and О. С. Харьков. ``Алгоритм екстраполяції з минулого для варіаційних нерівностей в гільбертовому просторі." Ukrainian. In: *Журнал обчислювальної та прикладної математики* 2 (2023), pp. 52–82.