

Список цитированных источников

1. Мосты автодорожные. Правила выполнения диагностики: ТКП 227-2018 (33200). – Минск, 2018 – 118 с.
2. Мосты и трубы. Правила обследований и испытаний: ТКП 45-3.03-60-2009 (02250). – Минск, 2009 г. – 29 с.
3. Правила определения грузоподъемности железобетонных и сталежелезобетонных балочных пролетных строений автодорожных сооружений. ТКП 479-2019. – Минск, 2019. – 248 с.
4. Методика оценки технического состояния мостовых сооружений на автомобильных дорогах. Федеральное дорожное агентство (Росавтодор). ОДМ 218.3.014-2011. – Москва, 2013. – 80 с.

References

1. Road bridges. Rules for performing diagnostics: ТКП 227-2018 (33200). – Minsk, 2018 – 118 p.
2. Bridges and pipes. Rules of surveys and tests: ТКП 45-3.03-60-2009 (02250). – Minsk, 2009 – 29 p.
3. Rules for determining the load capacity of reinforced concrete and steel-reinforced concrete girder superstructures of road structures. ТКП 479-2019. – Minsk, 2019. – 248 p.
4. Methodology for assessing the technical condition of bridge structures on highways. Federal Road Agency (Rosavtodor). ODM 218.3.014-2011. – Moscow, 2013. – 80 p.

УДК 624.012

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА И «МЯГКИХ ВЫЧИСЛЕНИЙ» ПРИ ПРОЕКТИРОВАНИИ МОНОЛИТНЫХ ПЛИТ НА ОСНОВАНИИ

А. Е. Желткович¹, В. В. Молош², К. Г. Пархоц³, Н. Г. Савейко⁴

¹Доцент, к.т.н., доцент, Брестский Государственный Технический Университет, Беларусь, e-mails: gelpek@mail.ru

²Доцент, к.т.н., доцент, Брестский Государственный Технический Университет, Беларусь, e-mails: m.vic@rambler.ru

³Инженер-программист, Беларусь, e-mails: konstantinparhoc@gmail.com

⁴Инженер-программист, Беларусь, e-mails: nick-2009@live.ru

Реферат

В статье проиллюстрирована возможность конвергенции механики, нейротехнологии и биоподобных технологий. Показана возможность применения, в задачах связанных с проектированием, мягких вычислений (soft-computing). В работе представлены результаты самонапряжений в плите на основании, полученные с помощью нейросети объединённой в систему с генетическим алгоритмом. Рассмотрена возможность оптимизации геометрических параметров плиты при заданных или изменяемых входных данных (прочность, самонапряжение, и др.) путем включения/отключения искусственных генетических признаков. Показано, что для описания состояния конструкции, где кинетика формирования бетонной структуры подчиняется нелинейным зависимостям, применение нейротехнологий и генетических алгоритмов наиболее оправдано. В статье описан процесс разработки нейронной сети и генетического алгоритма, обсуждается вопрос качества полученных решений.

Ключевые слова: генетический алгоритм, мягкие вычисления, хромосомы, гены, функция приспособленности, самоупругающийся бетон, нейронные сети.

APPLICATION OF GENETIC ALGORITHM AND "SOFT COMPUTING" IN THE DESIGN OF MONOLITHIC SLABS ON BASE

Zheltkovich A., Molosh V., Parchotz K., Saveiko N.

Abstract

The article illustrates the possibility of convergence of mechanics, neurotechnology, and biosimilar technologies. Shown the possibility of using so-called soft-computing in design-related tasks. The work presents the results of self-stresses in the slab on ground obtained using a neural network combined into a system with a genetic algorithm. The possibility of optimizing the geometric parameters of the slab at the given or variable input parameters (strength, self-stress, etc.) by turning on/off artificial genetic features is considered. It has been shown that to describe the state of the structure, where the kinetics of the formation of the concrete structure obeys nonlinear behaviors, the use of neurotechnologies and genetic algorithms is most justified. The article describes the process of developing a neural network and a genetic algorithm, discusses the quality of the solutions obtained.

Keywords: genetic algorithm, soft computing, chromosomes, genes, fitness function, self-stressed concrete, neural networks.

Мягкие или жесткие решения?

На протяжении многих лет в инженерной науке жёсткие вычисления (hard-computing) находили самое широкое применение. Среди многих инженеров по сей день существует устойчивый стереотип, что метод классических вычислений столь надёжен, и эффективен, что преждевременно даже обсуждать альтернативы. Но так ли он безальтернативен с позиции эффективности? В таблице 1 [1] представлен краткий обзор существующих жестких и получающих всё более широкое применение мягких методов вычислений.

Таблица 1 – Обзор существующих методов вычислений.

Мягкие вычисления*	Жесткие вычисления
свободны от неточностей, неопределенностей, частичной истины и приближения	нуждаются в модели анализа состояний
опираются на формальную логику и вероятностные рассуждения	полагаются на двоичную логику «false-true» и четкую систему
имеют стохастический характер	детерминированы
работают с неоднозначными и зашумленными данными	работают с точными данными
могут выполнять параллельные вычисления	выполняют последовательные вычисления
дают приблизительные результаты	дают точные результаты
вырабатывают свои собственные программы	требуют написания программ

*Мягкие вычисления (Soft Computing) – термин, введенный Лотфи Заде в 1994 году [2]

«Жесткие вычисления – это тот древний подход, используемый в вычислениях, который требует от младшего [...] персонала точно заявленной аналитической модели. Результатом жесткого вычислительного подхода является гарантированный, установленный, правильный результат, (*получаемый* – ред.) [...] с использованием математической модели или алгоритмического правила [...]. Жесткие вычисления не способны найти решение реальной проблемы» [1]. От себя добавим, что применяя соответствующие критерии соответствия решение, конечно же, может быть найдено, но для абстрактной, геометрически или математически идеализированной модели реальности, не для самой реальности.

В публикации [3] указывается, что традиционный метод вычислений, к которым несомненно относятся жесткие вычисления, подходит для решения реальных проблем, но основной связанный с этим недостаток заключается в том, что он требует большого количества времени и затрат на вычисления. По этой причине мягкие вычисления являются лучшей альтернативой.

Согласно сложившемуся подходу, мягкие вычисления включают в себя:

1. Задачи с нечёткой логикой.
2. Задачи, решаемые при помощи нейронных сетей.
3. Эволюционное моделирование – использование положения теории Дарвина для построения интеллектуальных систем.

4. Задачи, решаемые с использованием теории хаоса или аппарата, описывающего поведение некоторых нелинейных динамических систем, подверженных при определённых условиях явлению, известному как хаос. Поведение такой системы кажется случайным, даже если модель, описывающая систему, является детерминированной [4]. Примерами подобных систем являются турбулентность в атмосфере, потоки плазмы, биологические процессы, и другие. Сложные системы чрезвычайно зависимы от первоначальных условий, и небольшие изменения приводят к бифуркациям с последующим каскадом фрактальных расщеплений, что приводит к непредсказуемым последствиям (или потере контроля над процессом при применении классических методов вычислений).

В исследовании [5] указывается, что сущность мягких вычислений состоит в том, что в отличие от традиционных, они нацелены на приспособление к всеобъемлющей неточности реального мира. Руководящим принципом мягких вычислений является: «терпимость к неточности, неопределённости и частичной истинности для достижения [...] лучшего согласия с реальностью».

Исследователям, работающим с напрягающими бетонами, вообще с конструкциями из искусственных материалов, к коим без сомнения относится бетон, должна быть хорошо понятна фраза: «Поведение [...] системы кажется случайным, даже если модель, описывающая систему, является детерминированной». По этой причине при исследовании напрягающего бетона был применён метод мягких вычислений (нейронная сеть, объединённая в систему с генетическим алгоритмом).

Если говорить о выборе оптимизационных алгоритмов при решении нелинейных задач, к коим относятся задачи связанные с определением напряжённо-деформированного состояния (НДС) в конструкциях на напрягающем бетоне, то по эффективности работы, генетический алгоритм является достаточно универсальным (рисунок 1).

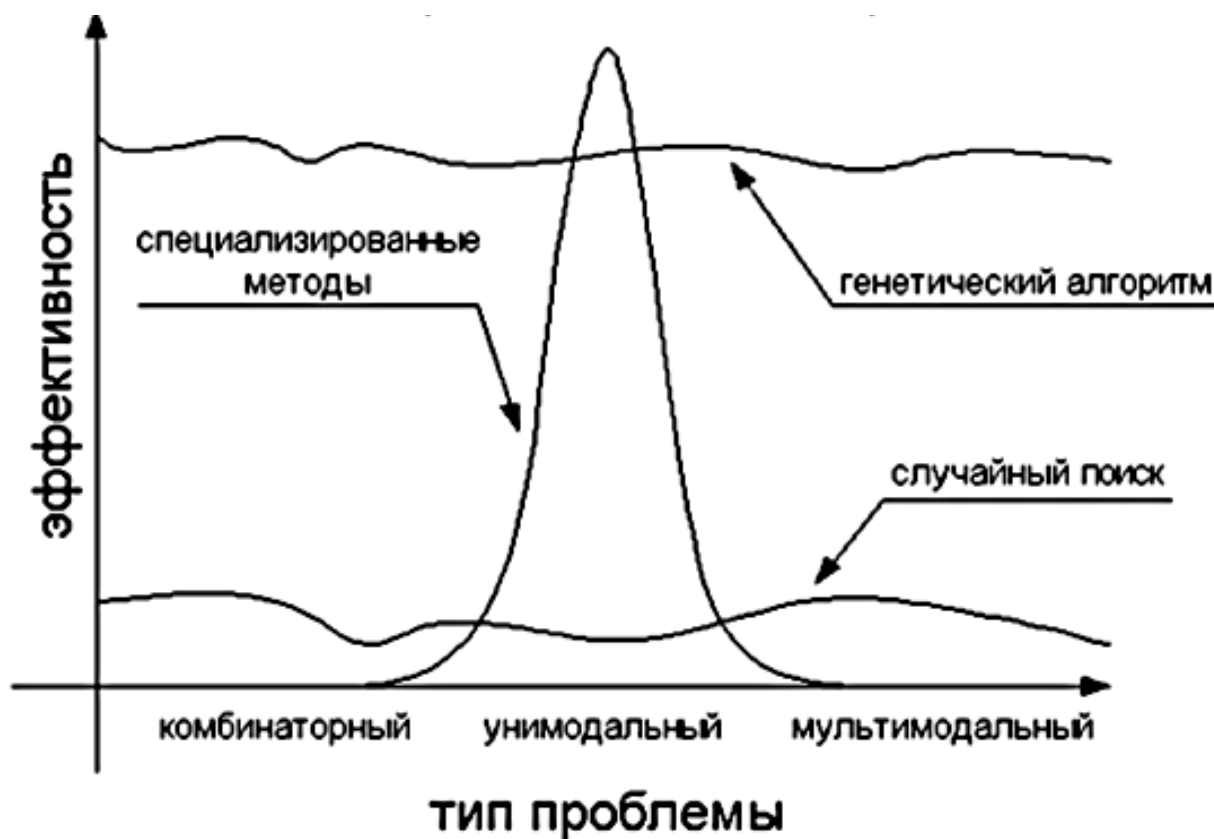


Рисунок 1 - Сравнительная эффективность алгоритмов оптимизации

1 Мягкие вычисления. Искусственная нейросеть

Подход, который использует вычислительные модели, имитирующие архитектурные, структурные или функциональные аспекты биологических нейронных сетей, таких как человеческий мозг, известны с середины XX века.

Искусственные нейронные сети (ИНС или НС), также известные как Neural Networks (NN) или Connection Models - алгоритмические математические модели, имитирующие поведенческие характеристики нервных сетей животных и выполняющие распределённую, параллельную обработку информации. Эти сети полагаются на особую систему обработки информации путем корректировки взаимосвязей между большим количеством внутренних узлов. НС, состоящая из нескольких слоёв связанных нейронов, называется глубокой. Глубокие нейронные сети включают в себя входной слой, скрытые слои и выходной слой; ИНС может соединять узлы нейронов левого уровня с узлами правого уровня (см. рис. 2б).

Структурно в НС можно выделить два основных способа улучшения работоспособности нейронной сети, а именно горизонтальное расширение за счет увеличения числа нейронных узлов в каждом слое и вертикальное расширение за счет увеличения количества слоев нейронной сети [6].

Полносвязные НС имеют несколько основных особенностей.

1. Нейроны в одном слое не связаны друг с другом.
2. Каждый нейрон в слое N соединен со всеми нейронами в слое N-1 (при полном соединении), а выход нейронов в слое N-1 является входом в нейроны в слое N.
3. Каждое соединение (синапс) имеет вес.

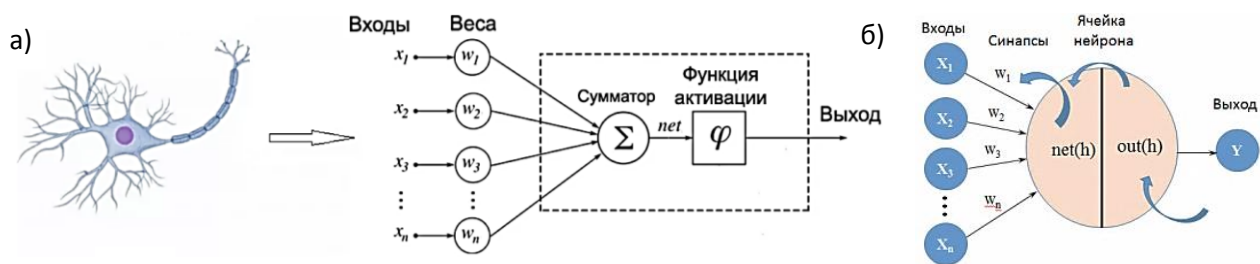


Рисунок 2 – Принципиальная схема нейрона (а), метод обратного распространения ошибки при обновлении весов синапсов (б)

Структура НС для прогнозирования НДС

НС играет роль экстрактора признаков в алгоритме обнаружения цели. Эта схема позволяет НС автономно улавливать зависимости из огромных объемов данных. НС может преобразовывать сигналы от входа к выходу с помощью простой операции сложения и умножения. Таким образом, хорошо обученная сеть может быстро реагировать без громоздких операций вычислений, в отличие от программ использующих методы жестких вычислений (например, метод конечных элементов). Чтобы уточнить характеристики НДС, была разработана и обучена искусственная нейронная сеть полносвязной архитектуры с семью входящими нейронами, двумя скрытыми слоями и десятью нейронами выходного слоя. Каждый из десяти нейронов выходного слоя учился «видеть» самонапряжение в соответствующей координате плиты. На входные нейроны подавались данные полученные в ходе лабораторных экспериментов: L (B) – длина и ширина плиты; H – толщина плиты; $f_{c,cube}^m$ – средняя прочность бетона на сжатие; $\tau_{1,R(t)}$ – пиковое напряжение на контактной поверхности между плитой и основанием; $u_{1,R(t)}$ – смещение, соответствующее максимальному напряжению на контакте [7]. ε_0 – деформация свободного расширения бетона; σ_{CE} – самонапряжение бетона при однопроцентном ограничении.

Формирование матриц входных и выходных данных

Поскольку количество параметров НС велико, необходим эффективный метод записи данных (обычно поступающие на вход и извлекаемые на выходе данные представлены в виде матриц или векторов). Для этого создавалась матрица входных и выходных данных для 149 различных временных интервалов. Для данных был создан специальный файл с расширением .csv. Во второй части матрицы (данные для выходного слоя ИНС) помещены напряжения, предварительно сгенерированные при помощи базовой физ.-мат. модели (для десяти точек плиты) на базе экспериментальных наблюдений за перемещениями в соответствующих координатах.

Механизм работы ИНС

Каждый нейрон может рассматриваться как оператор, который изменяет входные данные [8]. На вход он получает некую взвешенную сумму от всех нейронов предыдущего слоя, сигнал же на выходе из нейрона может быть определен следующим образом:

$$y_i = \sum x_i \cdot w_i + b \quad (1)$$

где x и y – входной и выходной сигналы НС,
 w – весовой параметр синапса,
 b – смещение (рисунок 2а).

Критерии качества работы ИНС

При тестировании нейросети на тестовой выборке использовалась средне-квадратическая (MSE метод) и относительная ошибки, поскольку эти метрики достаточно достоверно отражают точность результата прогнозирования, как для пакета из нескольких образцов, так и для единичного образца. Функция потерь (losses) предлагается для оценки разрыва между истинным результатом (целью) и предсказанием, формируемым нейронной сетью. Таким образом, процесс обучения нейронной сети становится задачей оптимизации, целью которой является минимизация потерь. Функция потерь определяется, как:

$$E = \frac{1}{n} \sum (Y_{target} - Y_{predicted})^2, \quad (2)$$

где: n – количество выборок,
 Y_{target} – фактические данные,
 $Y_{predicted}$ – предсказанные.

Параметрическая оптимизация, алгоритм градиентного спуска

Целью оптимизации параметров являлось нахождение минимального значения функции E на гиперповерхности потерь. На каждой итерации алгоритм обновляет весовые параметры w в нейронах. Чтобы найти минимум функции потерь, необходимо взять отрицательный градиент. В модели использовался оптимизатор Adam [9]. Минимум функции потерь, определялся, как частная производная потерь, по весу:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial out(h)} \cdot \frac{\partial out(h)}{\partial net(h)} \cdot \frac{\partial net(h)}{\partial w_i} \quad (3)$$

где $net(h) = \sum_{i=1}^n x_i \cdot w_i$ – значение взвешенной суммы от нейронов слоя N-1 входящей на вход некоторого нейрона в слое N (рисунок 2б),

$out(h) = f(net(h))$ – значение преобразованной (при помощи оператора «ReLU») функции (1) на выходе из нейрона,

$f(net(h)) = "ReLU"$ – функция активации нейрона (это наиболее распространённая функция активации, в особенности, если решается не классификационная задача, а задача регрессии [10]).

После вычисления потерь, информация антиградиента передается от выходного слоя к предыдущим, слой за слоем - обратно к входному слою, обновляя все веса синапсов. Этот метод поиска глобального минимума получил название – метод обратного распространения ошибки (back propagation algorithm - BP) [8, 11, 12]. Уравнение для итерации обновления веса формулируется следующим образом:

$$w_{i+1} = w_i - \alpha \cdot \frac{\partial E}{\partial w_i}, \quad (4)$$

где: α – параметр скорости обучения.

Таким образом, с помощью градиентной информации НС может искать относительно оптимальные спуски на гиперповерхности потерь.

Обучение ИНС

Следующим шагом являлось обучение нейросети. Все этапы создания НС, обучения и проверки реализованы на программном языке Python в рамках среды Tensorflow [13]. Количество эпох устанавливалось равным 250. В процессе регуляризации случайным образом отобранные 90% входных данных назначались в качестве тренировочной выборки (training dataset) и 10% данных было оставлено для проверки (validation) качества работы модели. Поскольку разработанная нейросеть, легкая, ей требовалось примерно 10-15 секунд для обучения.

2 Мягкие вычисления. Генетический алгоритм

Методы оптимизации в общих чертах можно разделить на две категории: метод на основе градиентного спуска и метод без градиента. Метод на основе градиента, такой как метод Ньютона [14] (метод обратного распространения ошибки) или метод сопряженного градиента [15], трудно кодировать в сложной проблеме, в частности связанной с поиском оптимальных топологий, геометрий, и т.д. Генетический алгоритм (ГА), использующий метод оптимизации без градиента, может быть наиболее рационален.

Проектирование плиты на основании является, по сути, задачей выбора оптимальных геометрических параметров плиты при необходимости обеспечения определённого уровня самонапряжения (например, в центре плиты/заданных координатах). С другой стороны это и задача подбора оптимального состава бетона при известных (или «плавающих») параметрах описывающих контактные характеристики плиты и основания. На первом этапе работы нейросети была получена реакция плиты (НДС) в зависимости от воздействия входных параметров, которые указаны выше. На этапе 2, данные весов для наиболее оптимальной архитектуры (в частности это была сеть с 7x10x10x10 нейронами) помещались в код ГА для решения уже обратной оптимизационной задачи. При оптимальных весовых значениях (синапсов) подобранных НС, для определения напряжений в плите размером 4x4 м, ставилась задача получить оптимальную толщину плиты H , прочность бетона, значение самонапряжения в кондукторе (при однопроцентном ограничении расширению), контактные характеристики плиты с основанием, свободное расширение. Уровень самонапряжения назначался – 0,55 МПа, для удобства сравнения с результатами расчётов полученных по базовой физ.-мат. модели.

Механизм генетического алгоритма

Генетический алгоритм – это своего рода эвристический подход к поиску наиболее оптимального решения, имитирующий процесс эволюции в природе [16]. Поскольку ГА был построен по аналогии эволюции в живой природе, он

использует биологические термины [17]. ГА использует принцип «генетического» хранения информации, принцип «размножения» (crossing over), принцип «мутации» (mutation) и «естественного/искусственного отбора» (natural selection).

В самых общих чертах эволюция основана на следующих принципах:

1. Наследственная изменчивость (обеспечивается путем скрещивания и мутации).

2. Борьба за существование. В ходе внутривидовой борьбы организмы конкурируют за ограниченные ресурсы – пищевые, территориальные (в данном исследовании «пищевым» ресурсом являлась награда – оценка за приспособленность (fitness score), которую получала особь при верно выбранном пути к цели).

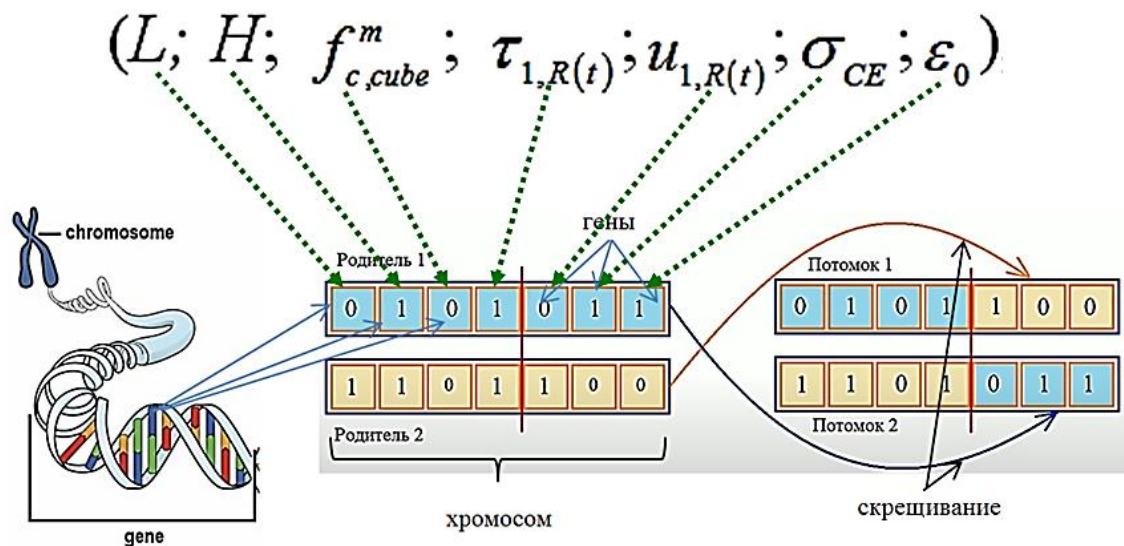
3. Естественный/искусственный отбор. Отбираются самые приспособленные, вместе с тем в новой популяции оставляется какое-то количество особей проигравших конкуренцию (из биологии известно, что особи проигравшие конкуренцию могут нести важную «генетическую» информацию, что при некоторых обстоятельствах, может дать выгодный эффект в будущем).

Основная идея ГА состоит в том, чтобы генерировать несколько решений случайным образом, а затем оценивать оптимальность решений, используя функцию приспособленности (fitness function).

Назначение генетических признаков для хромосом

В ГА особь с определёнными признаками записывается в виде вектора, называемого хромосомой. Как и в биологии в искусственной хромосоме содержится генетическая информация индивидуума. В данной работе хромосома представлена вектором признаков, который сохраняет информацию о новом решении в процессе эволюции. Каждая хромосома содержит семь ген - признаков ($L; H; f_{c.cube}^m; \tau_{1,R(t)}; u_{1,R(t)}; \sigma_{CE}; \varepsilon_0$), что означает, что вектор параметров имеет размерность 1×7 . Генетический алгоритм генерирует множество особей - хромосом за один раз, используя алгоритм случайных чисел и назначенный диапазон вакантных значений для признаков. Интервал возможной изменчивости гена, например, отвечавшего за среднюю прочность бетона, назначался в пределах от нуля до 40 МПа, толщина плиты от нуля до 0,3 м, и т.д. Таким образом, каждая хромосома с генами - признаками отличалась одна от другой. Все особи в одном поколении называются популяцией. В данном исследовании в силу небольшого числа ген в хромосоме эволюция составляла – 60 поколений популяций.

Чтобы генерировать следующую популяцию, особи «спариваются» друг с другом. При моделировании создавалось 40 особей - хромосом, из которых 2 пары наделялись возможностью скрещиваться. Не взаимодействовавшие особи переходили в следующее поколение без изменений. Ген потомства зависел от гена родительского поколения. При не бинарном кодировании новый ген (его значение) случайным образом отбирался из определённого интервала значений. После перемешивания (скрещивания) получался соответствующий ген родительской генерации. Упрощённая схема получения хромосомы нового поколения проиллюстрирована на рисунок 3.



а) – хромосом и ген в ГА, б) – скрещивание между хромосомами
 Рисунок 3 – Упрощённая схема получения хромосомы нового поколения

Естественный отбор или селекция?

При эволюции популяции была использована процедура селекции. От нее зависит, с какой скоростью и будет ли вообще популяция двигаться к глобальному оптимуму или застрянет в локальном экстремуме. Известно несколько подходов селективного отбора: колесо рулетки, турнирный метод, ранговый метод, комбинированный метод, др.

Индивидуум с высоким уровнем приспособленности имеет большую вероятность быть выбранным. Особь с наименьшей приспособленностью отсеивается с наибольшей вероятностью. В данной работе использовался алгоритм наиболее устойчивого состояния (steady-state selection) для выбора особи переходящей в новое поколение [18]. Блок-схема генетического алгоритма показана на рисунок 4а. При вычислении оптимальных признаков хромосомы следующей популяции, чтобы избежать преждевременной сходимости алгоритма, вводилась мутация, вероятность которой принималась равной – 0,05.

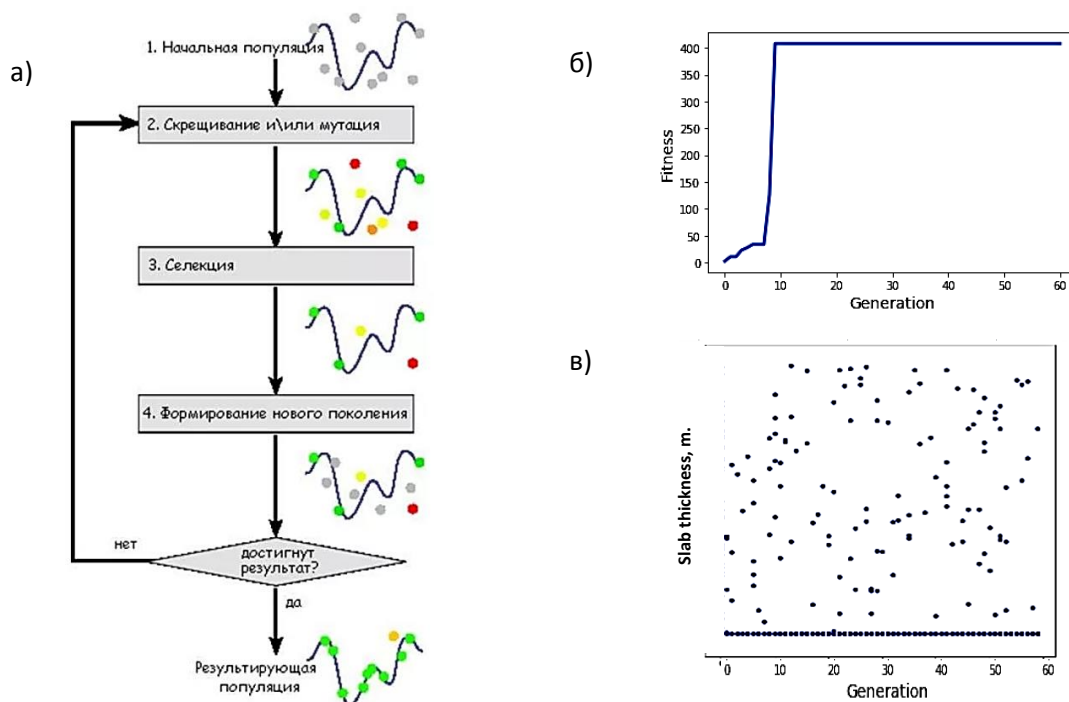
Включение/выключение генетических признаков

Важным преимуществом ГА является возможность управлять вариативностью требуемых ген. При необходимости определённые гены могут быть заблокированы для эволюционной изменчивости. Например, в данной работе блокировались гены, отвечавшие за длину плиты и за свободное расширение бетона. И наоборот, другим генам обеспечивалась возможность эволюционировать, например, генам ответственным за прочность бетона, самонапряжение, толщину плиты, и др.

Функция приспособленности

Функция приспособленности важнейшая при естественном или селективном отборе (рисунок 4б). Функция приспособленности представляет собой разрыв (степень приближения) между реакцией на входное воздействие (заданный уровень напряжения) хромосомы предыдущего поколения, к реакции хромосомы текущего поколения на тоже воздействие. На основе оценки приспособленности, алгоритм выбирает некоторое выдающееся решение. Когда (после

определённого количества итераций) показатель фитнес функции стабилизируется (выходит на минимальную изменчивость), считается, что популяция достигла оптимальности.



а) блок-схема генетического алгоритма, б) уровень приспособленности, достигнутый в процессе эволюции, в) эволюция гена ответственного за толщину плиты
Рисунок 4 – Блок-схема ГА и результаты эволюционной оптимизации

Нормализация

Для того, чтоб скрещивать особи между собой или манипулировать генами (нейронами) любым другим образом, гены (нейроны) должны быть выражены через безразмерный статистический показатель (используемый, как правило, для сравнения значений разной размерности). В математической статистике известен ряд методов: десятичное масштабирование, минимальная нормализация, нормализация средним (Z-нормализация). В работе была применена нормализация средним (в том числе и на первом этапе, для работы нейросети). Z-нормализация устанавливает среднее значение (математическое ожидание) и дисперсию для каждого гена (нейрона) и представляется формулой:

$$z = (x - \mu) / \sigma, \quad (7)$$

где: μ и σ – математическое ожидание (mean) и стандартное отклонение (standard deviation) соответственно.

Величины, полученные по формуле (7), называют Z-оценками (Z-scores). Z-score (переменная) соответствует количеству стандартных отклонений конкретного параметра до среднего значения. [19]. Нормализация для любого параметра z может быть представлена графически, как показано на рисунке 5.

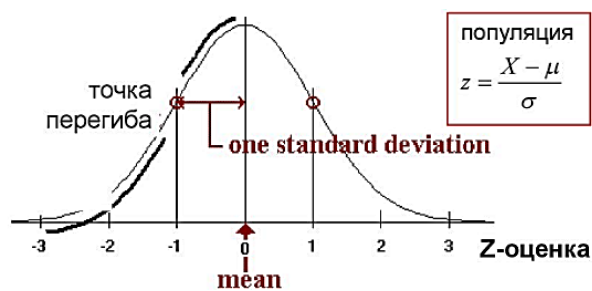
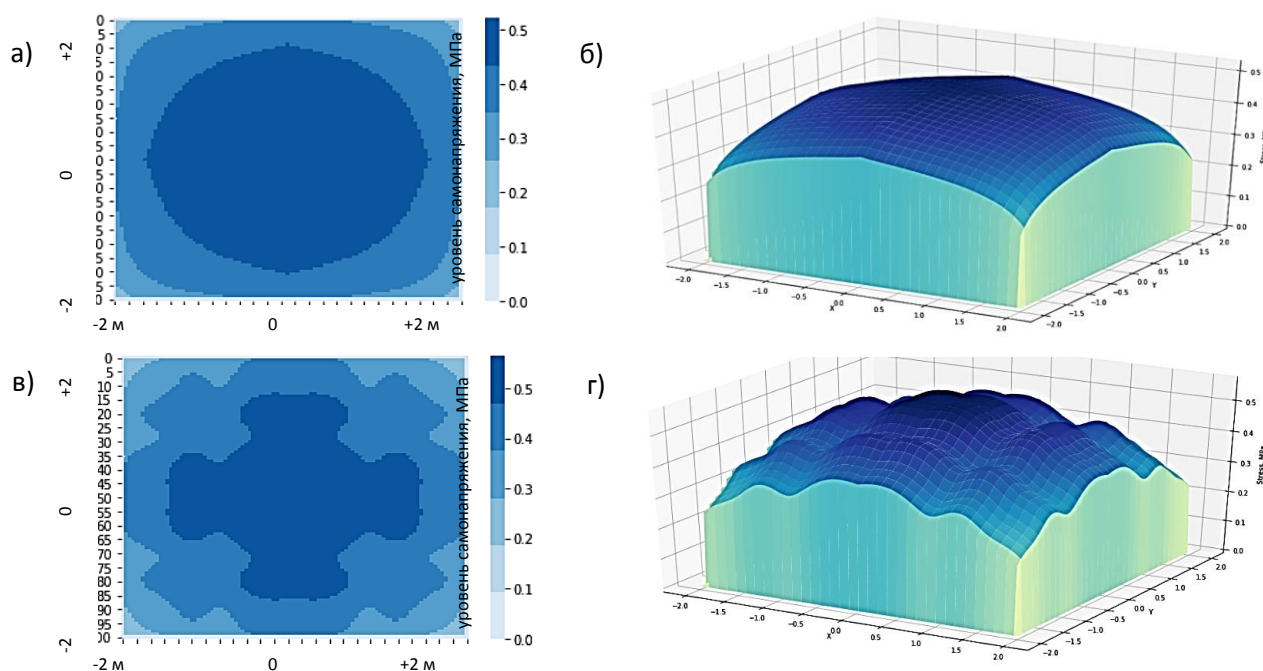


Рисунок 5 – Нормализация данных

Результаты

На рисунке 6, представлены результаты самонапряжений полученные при помощи НС (а, б) и ГА (в, г), в плите 4x4x0,1 м, для времени – 80 часов от затворения бетона.

ГА сгенерировал плиту 4x4 м, толщиной 0,100286 м (при этом геометрические размеры реальной физической полосы-плиты составляли – 4x0,1 м.). Самонапряжение при одноосном ограничении составило – 1,17 МПа (при этом в образцах-кондукторах самонапряжение на 80 час от затворения составляло – 0,81 МПа), прочность бетона на сжатие – 12,9 МПа (в экспериментах на сжатие на 80 час была достигнута прочность – 10,6 МПа), пиковое напряжение на контакте – 0,7 МПа (экспериментальное значение – 0,545 МПа), перемещение соответствующее этому напряжению – 19 мкм, (реальное значение – 18 мкм).



а, б) – нейросети; в, г) – генетического алгоритма.

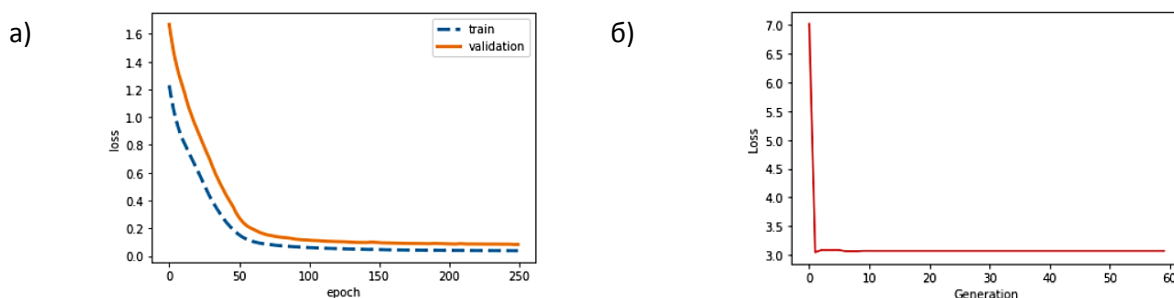
Рисунок 6 – Самонапряжение в плите 4x4x0,1 м. полученное при помощи:

Оценка точности

В ходе работы с нейросетью формировался отдельный файл .csv для тестовой выборкой данных. Было установлено, что наибольшая сходимость достигается при обучении нейросети с архитектурой 7x10x10x10 нейронов (функции потерь на тренировочной и валидационной выборках представлены на рисунке 7а).

По мере увеличения нейронов в промежуточных слоях потери в тестовой выборке непрерывно уменьшались и, для сети $7 \times 10 \times 10$ достигли значения $E = 0,011$. Более того, для оценки точности прогнозирования напряжений с использованием НС дополнительно рассчитывалась относительная ошибка тестовой выборки. Относительная ошибка составила – 1,9%.

В процессе эволюции, в ходе работы с ГА, потери от поколения к поколению уменьшались, приближаясь к нулю, (рисунок 7б). Из графика потерь можно сделать вывод, что уже к третьему поколению модель достигла оптимального результата.



а) при обучении НС, б) при оптимизации с помощью ГА
Рисунок 7 – Функции минимизации ошибок

На рисунке 8 представлены результаты моделирования напряжений в плите $4 \times 4 \times 0,1$ м, при помощи базовой физ.-мат. модели (на базе экспериментальных данных), нейросети с архитектурой $7 \times 10 \times 10$ нейронов и с использованием генетического алгоритма.

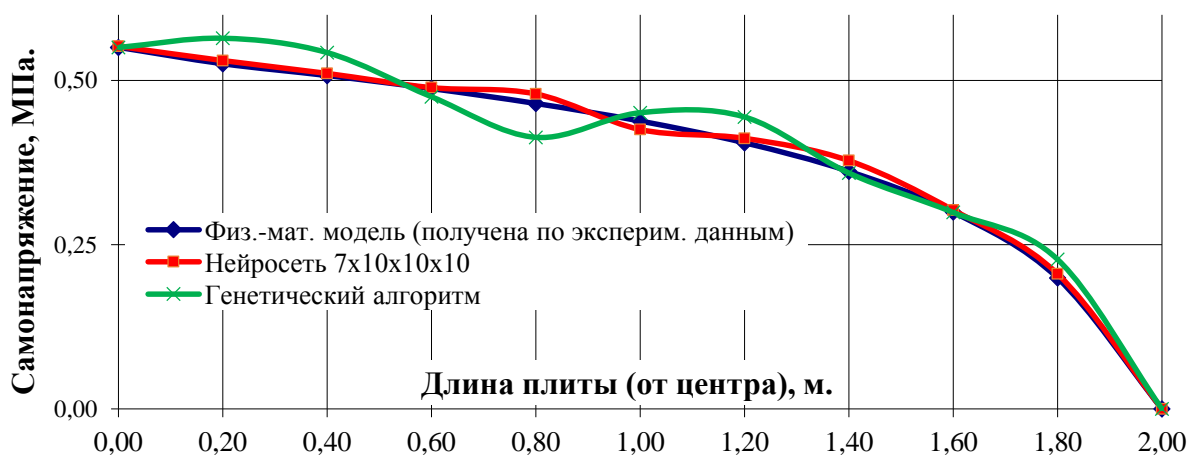


Рисунок 8. Самонапряжение на 80 час от момента затворения

Заключение

1. В настоящий момент жесткие вычисления являются практически безальтернативным средством описания процессов и явлений, они проверены практикой и чрезвычайно эффективны и, тем не менее, они не описывают непосредственно саму реальность. В случае несложных и линейных процессов, жесткие вычисления достигают порой почти 100% сходимостью с реальностью, однако сама реальность продолжает оставаться «вещью в себе». Мягкие вычисления (ос-

нованные на методах нечёткой логики, нейросетевых моделях, генетических алгоритмах), хоть и не раскрывают сути физических явлений, но воспринимая поток данных поступающих непосредственно из окружающей среды (при помощи датчиков, сенсоров, и т.д.), как бы нащупывают связь между ними. При этом если увеличивать плотность потока данных или иметь достаточно времени для наблюдений, то мягкие вычисления (НС, ГА) могут быть подстроены практически под любую задачу (имитируя работу нейронов и/или эволюцию в природе), при этом точность прогнозирования может превысить точность традиционных алгоритмов (особенно в области моделирования нелинейных процессов). Вместе с тем, на определённом этапе (при недостатке экспериментальных данных, этапе предварительной настройки параметров), нейросеть и ГА могут тестироваться и на данных сгенерированных при помощи традиционных физико-механических моделей.

2. Для определения самонапряжения в плитах на основании была разработана и обучена полносвязная нейросеть с архитектурой $7 \times 10 \times 10 \times 10$ нейронов в слоях. На тестирующей выборке, для плиты $4 \times 4 \times 0,1$ м. была достигнута высокая сходимость с физ.-мат. моделью. Среднеквадратичная ошибка составила – 0,011. Относительная ошибка составила – 1,9%.

3) При проектировании плиты, при заявленном требовании обеспечения самонапряжения в центре плиты на уровне 0,55 МПа (как в плите, из тестовой выборки НС) при помощи ГА была сгенерирована плита размерами $4 \times 4 \times 0,100286$ м. Самонапряжение в центре плиты составило – 0,549 МПа, при этом максимальная приспособленность генетических признаков была достигнута уже к десятому поколению. Это свидетельствует, что для данного класса задач, изначально назначенные 60 поколений эволюционного процесса несколько избыточны и в дальнейшем могут быть пересмотрены в сторону уменьшения без существенной потери точности для решения.

Список цитированных источников

1. Разница между мягкими вычислениями и жесткими вычислениями. Progler. [Электронный ресурс]. – Режим доступа: <https://progler.ru/blog>. – Дата доступа: 10.11.2022.
2. Zadeh, Lotfi A., «Fuzzy Logic, Neural Networks, and Soft Computing», Communications of the ACM, March 1994, Vol. 37 No. 3, pages 77-84.
3. Разница между мягкими вычислениями и жесткими вычислениями. Surveillancepackages. [Электронный ресурс]. – Режим доступа: <https://ru1.surveillancepackages.com/difference-between-soft-computing-and-hard-computing-4b2a>. – Дата доступа: 11.11.2022.
4. Игорь Н. фон Бекман / Нелинейная динамика сложных систем: теория и практика. Метанаука. Эволюция систем. Материалы к курсу лекций и учебнику / МГУ, Москва – 2018, С. 613
5. Применение интеллектуальной обработки информации Soft Computing в физике. [Электронный ресурс]. – Режим доступа: <http://spacelab.mininuniver.ru>. – Дата доступа: 12.11.2022.
6. Shin H.C, Roth H.R, Gao M, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning [J]. IEEE transactions on medical imaging, 2016, 35(5): pp. 1.285–1.298.
7. Желткович, А.Е. Расчёт вынужденных перемещений и напряжений от усадки в монолитных бетонных плитах, взаимодействующих с основанием / А. Е. Желткович, В. В. Тур // Строительная наука и техника. – 2011. – № 2 (35) : – С. 120–125.
8. Chen H, Lu F, He B. Topographic property of backpropagation artificial neural network: From human functional connectivity network to artificial neural network [J]. Neurocomputing, 2020, 418: 200-210.

9. Adam: A Method for Stochastic Optimization. [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/1412.6980>. – Дата доступа: 14.11.2022.
10. Функции активации нейросети: сигмоида, линейная, ступенчатая, ReLu. [Электронный ресурс]. – Режим доступа: <https://neurohive.io/ru/osnovy-data-science/activation-functions>. – Дата доступа: 14.11.2022.
11. Галушкин А. И. Синтез многослойных систем распознавания образов. — М.: «Энергия», 1974.
12. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
13. Citing TensorFlow. [Электронный ресурс]. – Режим доступа: <https://www.tensorflow.org/about/bib>. – Дата доступа: 16.11.2022.
14. Polyak B T. Newton's method and its use in optimization [J]. European Journal of Operational Research, 2007, 181(3): pp. 1.086–1.096.
15. Nazareth J. L. Conjugate gradient method [J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2009, 1(3): pp. 348–353.
16. Katoch S, Chauhan S.S, Kumar V.A. Review on genetic algorithm: past, present, and future [J]. Multimedia Tools and Applications, 2021, 80(5): pp. 8.091–8.126.
17. Safitri A.R, Muslim M.A. Improved accuracy of naive bayes classifier for determination of customer churn uses smote and genetic algorithms [J]. Journal of Soft Computing Exploration, 2020, 1(1): 70-75.
18. Qian W, Chai J, Xu Z, et al. Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection[J]. Applied Intelligence, 2018, 48(10): pp. 3.612–3.629.
20. Нормализация данных (Data normalization). [Электронный ресурс]. – Режим доступа: <https://wiki.loginom.ru/articles/data-normalization>. – Дата доступа: 15.11.2022.

References

1. Raznica mezhdu myagkimi vychisleniyami i zhestkimi vychisleniyami. Progler. [Elektronnyj resurs]. – Rezhim dostupa: <https://progler.ru/blog>. – Data dostupa: 10.11.2022.
2. Zadeh, Lotfi A., «Fuzzy Logic, Neural Networks, and Soft Computing», Communications of the ACM, March 1994, Vol. 37 No. 3, pages 77-84.
3. Raznica mezhdu myagkimi vychisleniyami i zhestkimi vychisleniyami. Surveillancpackages. [Elektronnyj resurs]. – Rezhim dostupa: <https://ru1.surveillancpackages.com/difference-between-soft-computing-and-hard-computing-4b2a>. – Data dostupa: 11.11.2022.
4. Igor' N. fon Bekman / Nelinejnaya dinamika slozhnyh sistem: teoriya i praktika. Metanauka. Evolyuciya sistem. Materialy k kursu lekcij i uchebniku / MGU, Moskva – 2018, С. 613
5. Primenenie intellektual'noj obrabotki informacii Soft Computing v fizike. [Elektronnyj resurs]. – Rezhim dostupa: <http://spacelab.mininuniver.ru>. – Data dostupa: 12.11.2022.
6. Shin H.C, Roth H.R, Gao M, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning [J]. IEEE transactions on medical imaging, 2016, 35(5): pp. 1.285–1.298.
7. Zheltkovich, A.E. Raschyot vyzhdenykh peremeshchenij i napryazhenij ot usadki v monolitnyh betonnyh plitah, vzaimodejstvuyushchih s osnovaniem / A. E. Zheltkovich, V. V. Tur // Stroitel'naya nauka i tekhnika. – 2011. – № 2 (35) : – S. 120–125.
8. Chen H, Lu F, He B. Topographic property of backpropagation artificial neural network: From human functional connectivity network to artificial neural network [J]. Neurocomputing, 2020, 418: 200-210.
9. Adam: A Method for Stochastic Optimization. [Elektronnyj resurs]. – Rezhim dostupa: <https://arxiv.org/abs/1412.6980>. – Data dostupa: 14.11.2022.
10. Funkcii aktivacii nejroseti: sigmoida, linejnaya, stupenchataya, ReLu. [Elektronnyj resurs]. – Rezhim dostupa: <https://neurohive.io/ru/osnovy-data-science/activation-functions>. – Data dostupa: 14.11.2022.
11. Galushkin A. I. Sintez mnogoslojnyh sistem raspoznavaniya obrazov. — М.: «Energiya», 1974.

12. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
13. Citing TensorFlow. [Elektronnyj resurs]. – Rezhim dostupa: <https://www.tensorflow.org/about/bib>. – Data dostupa: 16.11.2022.
14. Polyak B T. Newton's method and its use in optimization [J]. *European Journal of Operational Research*, 2007, 181(3): pp. 1.086–1.096.
15. Nazareth J. L. Conjugate gradient method [J]. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2009, 1(3): pp. 348–353.
16. Katoch S, Chauhan S.S, Kumar V.A. Review on genetic algorithm: past, present, and future [J]. *Multimedia Tools and Applications*, 2021, 80(5): pp. 8.091–8.126.
17. Safitri A.R, Muslim M.A. Improved accuracy of naive bayes classifier for determination of customer churn uses smote and genetic algorithms [J]. *Journal of Soft Computing Exploration*, 2020, 1(1): 70-75.
18. Qian W, Chai J, Xu Z, et al. Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection[J]. *Applied Intelligence*, 2018, 48(10): pp. 3.612–3.629.
19. Normalizaciya dannyh (Data normalization). [Elektronnyj resurs]. – Rezhim dostupa: <https://wiki.loginom.ru/articles/data-normalization>. – Data dostupa: 15.11.2022.