

АДАПТИВНАЯ СИСТЕМА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ОПТИМИЗАЦИИ ИНФОРМАЦИОННЫХ СИСТЕМ

ADAPTIVE ARTIFICIAL INTELLIGENCE SYSTEM FOR OPTIMIZING INFORMATION SYSTEMS

V. Gerasimov

Summary. The article proposes a new architecture of the artificial intelligence system aimed at solving the problems of optimization and modernization of information systems. The architecture allows optimization of various information systems using multi-criteria optimization methods, while remaining as flexible as possible. The approach allows artificial intelligence to automatically analyze and classify systems and propose optimization strategies. The focus is on the application of multi-criteria optimization methods, self-learning, and interaction with the user of the artificial intelligence system. Classical optimization methods and their applicability to the studied field of application and to the system are investigated. The problems of applying classical optimization methods in multi-criteria optimization are identified and a methodology for optimizing an artificial intelligence system based on two optimization methods is proposed: SLSQP and NSGA-II. The principle and necessity of the self-learning ability of such artificial intelligence systems are considered, and optimal methods of artificial intelligence training are proposed based on the study. Further studies of this system are considered to further improve it and increase efficiency to solve the problems of optimizing information systems.

Keywords: artificial intelligence, neural network architecture, optimization, multi-criteria optimization methods.

Герасимов Василий Александрович

аспирант, Технологический университет
имени дважды Героя Советского Союза,
летчика-космонавта А.А. Леонова, г. Королев
vasiliy.gerasimov.97@mail.ru

Аннотация. В статье предлагается новая архитектура системы искусственного интеллекта, направленная на решение задач оптимизации и модернизации информационных систем. Архитектура позволяет проводить оптимизацию различных информационных систем с использованием методов многокритериальной оптимизации, при этом оставаясь максимально гибкой. Подход позволяет искусственному интеллекту автоматически анализировать и классифицировать системы, и предлагать стратегии оптимизации. Основное внимание уделяется применению многокритериальных методов оптимизации, самообучению и взаимодействию с пользователем системы искусственного интеллекта. Исследованы классические методы оптимизации и их применимость к исследуемой области применения и к системе в целом. Выявлены проблемы применения классических методов оптимизации при многокритериальной оптимизации и предложена методика оптимизации системы искусственного интеллекта на основе двух методов оптимизации: SLSQP и NSGA-II. Рассмотрен принцип и необходимость способности к самообучению подобных систем искусственного интеллекта, а также предложены на основе исследования оптимальные методы обучения искусственного интеллекта. Рассмотрены дальнейшие исследования данной системы в целях ее дальнейшего улучшения и повышения эффективности для решения задач оптимизации информационных систем.

Ключевые слова: искусственный интеллект, архитектура нейронных сетей, оптимизация, многокритериальные методы оптимизации.

Введение

Современные информационные системы представляют собой довольно сложные системы, которые представляют собой мощные инструменты в решении тех или иных задач. Однако, с повышением сложности итоговой системы ее дальнейшая модернизация или оптимизация прямо пропорционально усложняется. Решить и подобрать оптимальную стратегию оптимизации информационной системы (больше уклон будет отдаваться именно оптимизации), несмотря на сложность задача выполнима, но затраты человеческого ресурса и времени колоссально большие, особенно если речь идет о средних или крупных предприятиях, где сосредоточено несколько различных информационных систем, которые не только сами по себе являются сложными структурами в цифровой экосистеме предприятия, но и могут быть взаимосвязаны друг с другом, а также не стоит забывать о разнообразии систем (веб-сайты, базы данных, ERP-системы и т.д.). Помимо этого, необходимо

учитывать множество критериев (некоторые из которых могут быть скрытые или неявные) одновременно, чтобы добиться максимально эффективной стратегии. Для этого необходима команда специалистов, которая сможет анализировать все состояния систем и в динамике реагировать на все возможные аномалии или учитывать важные критерии при ходе задачи. Все это требует больших сил и средств.

Поэтому актуальна тема создания мощного и полезного инструмента, способного значительно облегчить такую трудоемкую задачу.

Необходимо создание гибкой, что важно, системы искусственного интеллекта, способной адаптироваться к различным типам информационных систем и выявлять ключевые метрики для их дальнейшей оптимизации. В данной работе предложена подобная система искусственного интеллекта, направленная на решение данной проблемы. Архитектура позволяет искусствен-

ному интеллекту различать информационные системы, классифицировать, выявлять ключевые показатели для оптимизации и предлагать динамические стратегии по оптимизации. Помимо этого, в архитектуре присутствуют модули самообучения, аргументирование своих решений пользователю.

Основное внимание уделяется применению многокритериальных методов, самообучению и достижения максимальной гибкости и адаптируемости искусственного интеллекта, что позволяет сделать исследуемую систему максимально пригодной для широкого спектра применений в различных средах.

Существующие методы оптимизации

Необходимо понимать, что задача оптимизации не рассматривается для одной информационной системы (далее — ИС). Сегодня в любой цифровой системе любой компании сосредоточены несколько ИС, которые разнообразны, и каждая из них имеет свои собственные параметры. И именно их оптимизация и, в большинстве случаев, в совокупности, может повысить эффективность компании и ее внутреннюю или внешнюю работу.

Существующие классические методы оптимизации нельзя назвать неэффективными, так как данные методы тоже имеют свои преимущества и активно используются в решении задач оптимизации.

Например, линейное программирование или метод жадных алгоритмов. Каждый из методов способен решить задачу оптимизации, однако в них есть и недостатки, которые влияют, как бы это не звучало парадоксально, на решение оптимизационных задач. В любом из методов требуется большое человеческое участие, данным методам трудно адаптироваться к меняющимся условиям, и основной недостаток — возникают трудности в решении многокритериальных задач. А задачу по оптимизации ИС рассматривается именно как многокритериальная задача.

Линейное программирование ограничивает работу с функциями и задачами в целом, так как трактует, чтобы целевые функции и ограничения были линейными. Здесь возникает проблема: многие показатели параметров системы могут быть нелинейными и имеют нелинейные взаимосвязи. Это приводит к нахождению не оптимального решения, а субоптимального.

Рассматривая решения многокритериальных задач в линейном программировании, можно увидеть, что часто применяется метод свертки. Все цели сворачивают в одну с помощью весовых коэффициентов. С одной стороны, это хорошее решение и все цели в любом случае будут учитываться в решении задач, так как используют-

ся коэффициенты, но с другой стороны, результат будет зависеть от выбора весов, который может быть субъективным и не отражать реальные планируемые результаты или приоритеты. Иными словами, проблему можно обозначить следующим образом: выбор весов сильно влияет и способен изменить оптимальное решение, которое в конечном итоге может не соответствовать реалиям. Также в таком подходе не учитываются конфликтующие цели.

Наконец в линейном программировании возможно найти только одно решение, однако, снова происходит столкновение с реальностью, так как важно иметь несколько альтернативных решений, каждое из которых оптимизирует либо различные параметры, либо предлагает различные стратегии решений, которое в дальнейшем уже может выбрать человек с учетом своих основных целей или приоритетов.

В случае жадных алгоритмов все несколько проще. Жадные алгоритмы принимают локально оптимальные решения на каждом этапе решения задачи оптимизации в условии того, что конечный результат также будет оптимальным.

Таким образом, можно выявить следующие минусы в использовании данного метода в многокритериальной оптимизации:

1. Оптимальность сохраняется только локально.
2. При учете нескольких критериев практически отсутствует гибкость.
3. Нет механизма поиска компромиссных решений.
4. Отсутствие механизма возврата к предыдущим оптимальным решениям.

Рассмотрим конкретный пример, показывающий малоэффективность классических методов оптимизации при рассматриваемой методике работы ИИ и оптимизации при многокритериальном подходе:

Оптимизация серверной инфраструктуры, где ключевые параметры:

- Производительность системы ($A(x)$) — зависимость от конфигурации оборудования, процессоров и объема памяти.
- Энергопотребление ($E(x)$) — объем затраченного энергопотребления зависит напрямую от конфигурации и производительности системы.
- Затраты на обслуживание ($S(x)$) — сумма затрат на конфигурацию оборудования и энергопотребление.

Введем три переменные:

- x_1 — количество процессоров.
- x_2 — объем ОЗУ.
- x_3 — объем сетевых ресурсов.

Исследование и мониторинг показало, что необходимо максимизировать производительность, но при этом минимизировать энергопотребление и затраты.

Таким образом, описать каждый параметр через формулы можно следующим образом:

$$P(x) = \alpha_1 \cdot \log(x_1) + \alpha_2 \cdot \sqrt{x_2} + \alpha_3 \cdot e^{x_3}$$

$$E(x) = \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot \log(x_3)$$

$$S(x) = \gamma_1 \cdot x_1 + \gamma_2 \cdot \sqrt{x_2} + \gamma_3 \cdot x_3$$

Теперь рассмотрим применение линейного программирования для этой задачи.

Прежде всего мы сталкиваемся сразу с проблемой нелинейных зависимостей. Если рассмотреть формулу производительности, то в ней сосредоточены квадратичная и экспоненциальная функции. Привести такую задачу к линейному виду крайне затруднительно. Но линейное программирование требует соблюдения линейности в целевых функциях и ограничениях.

Вторая проблема, которую можно увидеть в исходных данных задачи — конфликтность некоторых целей.

Повышение производительности напрямую зависит от энергопотребления и затрат. Если добавлять количество процессоров или изменять конфигурацию серверов в целом, это неизбежно приведет к изменениям в энергопотреблении и затратам. И наоборот, если минимизировать энергопотребление и затраты, то необходимого прироста производительности достичь невозможно.

То есть, линейное программирование отлично подходит для решения одной из целей, но становится неэффективным при условии, когда необходимо учитывать несколько целей одновременно, которые, в свою очередь, конфликтуют между собой. Применение классического метода оптимизации в данном случае неизбежно ведет к полному игнорированию иных важных параметров оптимизации, что приводит к снижению результатов проведенной оптимизации в целом.

Поэтому необходимо использовать другую концепцию, методику и методы оптимизации, которые будут способны решить проблемы классических методов оптимизации и быть не менее результативными.

Таким образом, была выдвинута идея использовать искусственный интеллект и все преимущества данной технологии к обработке больших объемов данных, методика и архитектура системы ИИ, которая будет способна к решению задач оптимизации с учетом множества сложных критериев.

Описание адаптивной системы искусственного интеллекта

Стоит заранее отметить, что в данном пункте будет в основном отражена архитектура предлагаемой системы, методы оптимизации, будут вынесены отдельным пунктом, так как исследование методов оптимизации крайне важны для выбора архитектуры модуля оптимизации для системы в целом.

Итак, рассматривается система искусственного интеллекта, способная решать задачи по оптимизации ИС. Система ИИ выбрана по ряду причин:

1. Современные технологии и научные исследования позволяют применять данные системы для решения подобных трудоемких задач.
2. ИИ способен автоматизировать анализ систем и выявлять различные проблемы и аномалии для дальнейшей оптимизации. [2]

Архитектура системы представляет собой совокупность модулей, которые выполняет каждый свои задачи:

- Модуль анализа системы. Данный модуль предназначен для определения типа системы, которую рассматривает ИИ. Для каждой системы используются свои метрики или общие метрики, которые заложены для анализа систем.
- Модуль многокритериальной оптимизации. Данный модуль предназначен непосредственно для решения задачи оптимизации, где учитываются несколько метрик и используются дополнительно отдельные методы оптимизации.
- Модуль самообучения. Данный модуль предназначен для обучения ИИ в целом, будь это машинное обучение на начальном этапе развертывания системы, или же обучение на накопленных данных с предыдущих взаимодействий и решений ИИ с ИС, а также на основе обратной связи с пользователем.
- Модуль общения с пользователем. Модуль предоставляет возможность ИИ общаться с пользователем и наоборот. Также данный модуль определяет возможность системе выдвигать свои стратегии оптимизации пользователю, аргументировать их и вступать в диалог с человеком для достижения наиболее оптимального результата работы.

Алгоритм работы ИИ можно описать следующей схемой, представленной на рисунке 1.

ИИ изучает ИС, к которой ему дали доступ и определяет по различным метрикам что это за система, например по количеству операций в единицу времени, можно определить, что система является базой данных.

Далее ИИ начинает мониторинг исследуемой ИС или пользователь начинает диалог с системой ИИ и дает вво-

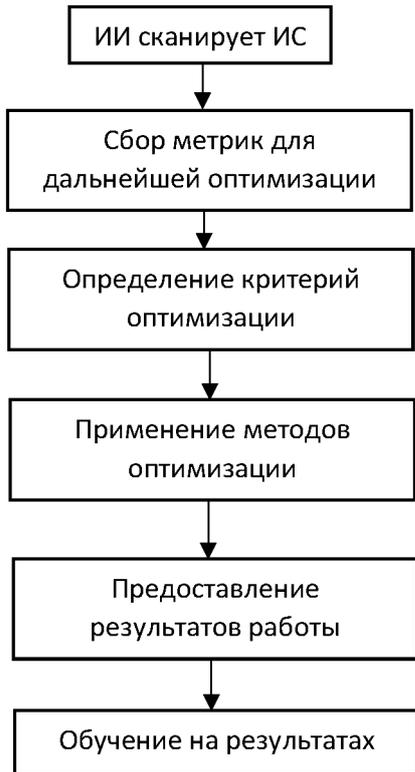


Рис. 1. Алгоритм работы ИИ

данные или задачу по оптимизации. Система начинает поиск решений, с учетом выполнения необходимых целей и всех критериев и переменных, которые возникают в решении проблемы. После обработки всех данных ИИ предоставляет несколько стратегий решений задачи оптимизации, где аргументирует свои выборы и дает пользователю скорректировать результаты для достижения максимально оптимального плана действий. Далее система ИИ сопровождает ИС на всем протяжении этапов оптимизации и оказывает поддержку в принятии решений на различных этапах. По завершению всех работ, система ИИ переходит в режим мониторинга ИС и самообучения на основе выполненных задач и процессов во время оптимизации.

Методы многокритериальной оптимизации

Самые главные опорные модули в системе ИИ для максимально эффективного решения задач по оптимизации являются модули многокритериальной оптимизации и модуля самообучения системы ИИ.

Почему все же многокритериальная оптимизация? Задачи многокритериальной оптимизации способны решать задачи с несколькими целевыми функциями одновременно, с учетом того, что эти функции могут конфликтовать друг с другом. Этот факт является одним из основных, так как в реальности, практически любая задача по оптимизации сталкивается с множеством различных критериев, некоторые из которых могут кон-

фликтовать друг с другом. И все это необходимо учитывать на всем протяжении решения задачи.

Многокритериальная оптимизация во многом выигрывает в оптимизации, в сравнении с другими методами, например с линейным программированием и схемой свертки, которая упоминалась ранее. Однако, многокритериальная оптимизация намного сложнее в анализе и решении, по сравнению с ее аналогами, а также дополнительную сложность создают нахождение компромиссов между критериями. [1]

Для решения данных проблем, и используются, прежде всего технологии ИИ. Помимо этого, были анализированы и найдены наиболее практичные методы оптимизации при условии, что система ИИ адаптивна и в ней заложена многокритериальная оптимизация.

Рассматриваются два метода оптимизации:

1. SLSQP (последовательное программирование методом наименьших квадратов). Метод позволяет работать с задачами с ограничениями, отличается большой скоростью, где необходима оптимизация нескольких переменных, на которых наложены как линейные, так и нелинейные ограничители. Метод требует, чтобы целевые функции были дифференцируемыми, что подходит для многих задач оптимизации. Метод подходит для задач, где необходимо оптимизировать несколько связанных метрик, при этом соблюдая определенные ограничения.

Основные преимущества:

- Эффективность для задач с несколькими ограничениями.
- Быстрое схождение в условии требования баланса между целевыми функциями.
- Учет дифференцируемости функций.

Недостатки:

- Существует шанс плохой работы с задачами с дискретными переменными.
- Требование дифференцируемости функций не могут выполняться во всех ИС.

2. Второй метод, который стоит рассмотреть — NSGA-II. Алгоритм многоцелевой оптимизации, основанный на процедуре быстрой сортировки решений на слои, которые не доминируют друг над другом.

Метод позволяет использовать многокритериальный подход, работать со сложными и дискретными функциями и эффективен в поиске Парето-оптимальных решений, что необходимо важно, при условии предложения нескольких альтернативных вариантов решения задачи.

Однако его основные недостатки: более медленный по сравнению с SLSQP и существует шанс, что метод не приведет к точному и единственному решению.

Таким образом, имеется два метода оптимизации, которые удовлетворяют идеи системы ИИ, но оба варианта имеют сильные положительные и отрицательные стороны. Поэтому было предложена идея комбинации обоих методов в одном модуле.

Прежде всего идея основана и аргументирована преимуществами каждого из методов, так как комбинация позволит обеспечить наиболее эффективный и гибкий инструмент для решения многокритериальных задач.

Суть комбинации заключается в поэтапном использовании каждого из методов. SLSQP более простой и быстрый метод и его целесообразнее использовать для начальной оптимизации. Затем для нахождения набора решений и учитывая более сложные критерии при оптимизации использовать NSGA-II метод.

Следовательно структура системы ИИ, будет следующей:

1. Анализ и изучение ИС.
2. Начальная оптимизация (использование метода SLSQP).
3. Генерирование нескольких стратегий оптимизации (на основе метода NSGA-II).
4. Коммуникация с пользователем и решение поставленной задачи с последующим самообучением ИИ.

Таким образом, система способна быстро найти решения, которые способны удовлетворить основным ограничениям и минимизировать метрики, а также найти альтернативные пути решения задачи, что также немало важно для нахождения компромиссного решения.

Самообучение исследуемой системы ИИ

Архитектура модуля многокритериальной оптимизации в системе ИИ является основным рабочим инструментом системы в процессе принятия решения. Самообучение является ключевым в вопросе гибкости и адаптивности системы ИИ. Основная задача модуля — накапливать данные о предыдущих взаимодействиях и оптимизациях и анализировать их для улучшения показателей результативности системы. [4] Поэтому модуль должен:

1. Накапливать знания — записывать результаты каждой оптимизации и действий с ИС в целом для того, чтобы изучать системы в динамике, анализировать и запоминать какие действия были совершены над той или иной ИС.

2. Принимать обратную связь — способность ИИ получать и обрабатывать обратную связь от пользователя и от системы для оценки успешности проделанной работы.
3. Анализ и обновление данных — на основе полученной информации улучшать свою модель оптимизации.

Для достижения и выполнения всех необходимых задач модулю самообучения необходима довольно сложная методология. Наиболее подходящим является Усиленное обучение с элементами онлайн-обучения.

Усиленное обучение — метод, при котором ИИ учится на основе обратной связи с окружающей средой. Поскольку система ИИ должна быть гибкой и работать в реальном времени, то необходимо дополнить методологию методом онлайн-обучения. Это позволит обновлять модель ИИ по мере накопления новых данных. Это позволит повысить адаптацию системы к изменениям, которые могут происходить в режиме реального времени. Помимо этого, онлайн-обучение способно дать ИИ возможность быстро реагировать и изучить новые типы данных, конфигурацию и вариативность ИС, которые могут быть развернуты в той или иной компании.

При этих данных наиболее подходящими алгоритмами обучения могут быть:

Deep Q-learning — алгоритм, использующий нейронные сети для аппроксимации функций для ситуаций с несколькими количествами состояний.

Proximal Policy Optimization — продвинутый алгоритм, который применяется для сложных и высоко размерных состояний и действий. ТО есть подходит для масштабных задач оптимизации.

Заключение

Таким образом, была предложена и описана новая архитектура системы искусственного интеллекта, которая направлена на автоматизацию и решения задач оптимизации информационных систем. Использование многокритериального метода оптимизации позволяет решить и исключить недостатки классических методов оптимизации. Предложенная архитектура и методика оптимизации позволяет адаптироваться и оставаться гибким искусственному интеллекту вне зависимости от типов и конфигурации ИС. ИИ способен быстро и аргументировано предлагать стратегии оптимизации и достигать самых оптимальных результатов. Методика самообучения способна сохранять историю всех взаимодействий с системами и исследовать их в динамике, адаптируясь к различным изменениям во внешней среде. Это позволяет не прерывать процесс мониторинга и оптимизации,

а также снижает порог возникновения неверных решений при резком или более плавном изменении ИС или иных сущностей, которые влияют на критерии оптимизации информационных систем.

Данная архитектура может быть улучшена, так как это одна из возможных итераций подобных адаптивных ИИ-систем.

Необходимо дополнительно исследовать архитектуру системы, рассмотреть ее развитие в динамике, чтобы

определить возможности добавления элементов предсказательного моделирования. Это необходимо, чтобы система прогнозировала будущие потребности в оптимизации системы.

Также в перспективе развития стоит возможность интегрирования с большими объемами данных и более глубокое использование методов машинного обучения для достижения еще большей гибкости и универсальности системы ИИ в целом.

ЛИТЕРАТУРА

1. Бахтин В.В., Подлесных И.А., Тюрин С.Ф. Решение задачи многокритериальной оптимизации вариантов декомпозиции нейронной сети и компоновки каскада вычислительных устройств методом парето // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. 2022. №43. С. 136–156.
2. Бородин Г.Д. Краткий обзор и классификация искусственных нейронных сетей // Известия ТулГУ. Технические науки. 2021. №11. С. 45–53.
3. Зобина О.В., Дю А.И., Бабаева Ю.А. Многокритериальная оптимизация // StudNet. 2021. №1. С. 769–775.
4. Косенкова Е.Ю., Гаев Л.В. Самообучение искусственного интеллекта: Новые возможности и риски // Международный журнал гуманитарных и естественных наук. 2024. №5–4 (92). С. 22–24.
5. Мамедова Л.Э., Иванова Л.Н., Алтаев Е.С. Основные аспекты технологии искусственного интеллекта // Известия ВУЗов ЭФиУП. 2023. №3 (57). С. 78–88.
6. Миронов А.Н., Лисицкий В.В. Метод многокритериальной нелинейной оптимизации сложных организационно-технических систем на основе минимизации невязок в условиях временных ограничений // Известия ТулГУ. Технические науки. 2019. №9. С. 330–343.
7. Панарин В.М., Гришаков К.В., Маслова А.А., Гришакова О.В., Архипов А.В. Применение искусственных нейронных сетей в инженерно-технических и экологических разработках // Известия ТулГУ. Технические науки. 2022. №4. С. 278–284.
8. Пиявский С.А. Метод универсальных коэффициентов при принятии многокритериальных решений // Онтология проектирования. 2018. №3 (29). С. 449–468.
9. Черепанов Н.В., Буслаев С.П. Проблемы и задачи развития искусственного интеллекта на машиностроительном предприятии // Инновации и инвестиции. 2021. №7. С. 175–179.
10. Чупакова А.О., Гудин С.В., Хабибулин Р.Ш. Разработка и обучение модели искусственной нейронной сети для создания систем поддержки принятия решений // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2020. №3. С. 61–73.

© Герасимов Василий Александрович (vasiliy.gerasimov.97@mail.ru)
Журнал «Современная наука: актуальные проблемы теории и практики»