

Тверской государственный университет

На правах
рукописи

Мазурова Ирина Сергеевна

ЧИСЛЕННЫЕ МЕТОДЫ ПОСТРОЕНИЯ
ОПТИМАЛЬНОГО УПРАВЛЕНИЯ В СИСТЕМАХ С ЗАПАЗДЫВАНИЕМ

Специальность 01.01.09 - Дискретная математика
и математическая кибернетика

Диссертация на соискание ученой степени кандидата
физико-математических наук

Научный руководитель –
доктор физико-математических наук,
профессор Андреева Е.А.

Тверь – 2014

Оглавление

Введение	3
Глава 1. Обобщенная популяционная модель хищник-жертва, описываемая системой интегро-дифференциальных уравнений типа Вольтерра	14
§1. Постановка задачи оптимального управления в обобщенной популяционной модели хищник-жертва	14
§2. Применение методологии БАД для построения оптимального управления в обобщенной популяционной модели хищник-жертва	25
2.1. Описание метода БАД для оптимизации систем, описываемых интегро-дифференциальными уравнениями	25
2.2. Практическая реализация численного алгоритма на основе метода БАД	32
2.3. Блок-схема численного алгоритма оптимизации на основе метода БАД	35
2.4. Анализ влияния параметров задачи на оптимальное решение системы	35
§3. Применение генетического алгоритма для построения оптимального решения системы, описываемой интегро-дифференциальными уравнениями	47
3.1. Описание генетического алгоритма	47
3.2. Практическая реализация генетического алгоритма	55
3.3. Блок-схема генетического алгоритма	58
3.4. Анализ влияния параметров генетического алгоритма	58
§4. Сравнение результатов работы метода БАД и генетического алгоритма	62
Глава 2. Модель нейронной сети, описываемая системой интегро-дифференциальных уравнений	67
§1. Постановка задачи оптимального управления в модели искусственной нейронной сети	67
§2. Применение методологии БАД для обучения искусственной нейронной сети	72
2.1 Алгоритм построения приближенного оптимального решения	74
2.2 Анализ влияния параметров задачи на оптимальное решение	75
§3 Применение генетического алгоритма	86
3.1. Практическая реализация генетического алгоритма	86
3.2 Анализ применения генетического алгоритма в зависимости от параметров	88
§4. Сравнение результатов работы метода БАД и генетического алгоритма	90
§5. Задача обучения искусственной нейронной сети как задача оптимального управления с нефиксированным временем	94
Глава 3. Методы сведения многокритериальной задачи к задаче с одним критерием и программная реализация разработанных алгоритмов	98
§1. Общие сведения о методах сведения исходной многокритериальной задачи к задачам с единым критерием	98
§2. Практическая реализация методов сведения исходной многокритериальной задачи к задачам с единым критерием	101
2.1 Метод линейной свертки	101
2.2 Свертка с неотрицательными весовыми коэффициентами	102
2.3. Принцип гарантированного результата	104
§3. Программная реализация разработанных алгоритмов	107
3.1 Проектирование пользовательского интерфейса	107
3.2. Разработка модульных тестов.	108
3.3. Компьютерные программы для статистической обработки данных	110
3.4. Программная реализация разработанных алгоритмов	112
Заключение	115
Список литературы	117

Введение

Изучение многих процессов, происходящих в различных биологических, экологических, экономических и технических системах, сводится к построению и анализу их математических моделей. Рассмотрение модели произвольной непрерывной динамической системы как взаимосвязанной совокупности элементов, выходы и входы которых связаны причинными отношениями, приводит к описанию их в общем случае системой интегро-дифференциальных уравнений типа Вольтерра.

Имеется обширная литература, посвященная изучению частных классов интегро-дифференциальных уравнений, либо некоторым частным вопросам теории интегро-дифференциальных уравнений. К последним, в частности, относятся монография Быкова [38], посвященная исследованию ряда вопросов качественной теории интегро-дифференциальных уравнений, монографии Вольтерра [43]. В ряде случаев интегро-дифференциальные уравнения исследуются как частный случай различных классов функциональных (В.И.Сумин, J.Warga), функционально-дифференциальных (Н.В.Азбелев, А.И.Булгаков, В.П.Максимов, J.Hale) уравнений.

Во многих прикладных задачах приходится осуществлять целенаправленное воздействие на такие системы — управлять ими. При этом часто возникает вопрос об отыскании наилучшего (оптимального) в том или ином смысле управления. Основные вопросы, которые возникают при теоретическом исследовании оптимизационных задач, — существование, единственность, и нахождение оптимального управления при условии его существования.

Истоки теории оптимального управления восходят к работам Л.С. Понтрягина [100],[27],[28], Р. Беллмана [26], Н.Н. Красовского [71], У. Флеминга, А. Фридмана.

Большой вклад в развитие теории оптимального управления внесли В.Г. Болтянский, Л. Д. Беркович, Е. А. Брайсон, Дж. Варга, Р. Ф. Габасов, Р.В. Гамкрелидзе, А. Я. Дубовицкий, Ю. Г. Евтушенко, В. И. Zubov, А. Д. Иоффе, Ф.

М. Кириллова, А.А. Колесников, В. Ф. Кротов, Г. Лейтман, А. А. Милютин, Е.Ф. Мищенко, Н. Н. Моисеев, Н. Н. Петров, Л.И. Розонэр, В. М. Тихомиров, Ф. Л. Черноусько, С. В. Чистяков, В. А. Якубович, Д. Эллиот.

Теория оптимального управления разнообразными сосредоточенными и распределенными системами получила свое развитие как в отечественных работах (С.А.Авдонин, Ю.Г.Борисович, В.А.Брусин, А.Г.Бутковский, Ф.П.Васильев, М.И.Зеликин, А.Д.Иоффе, А.З.Ишмухаметов, Ю.В.Орлов, М.М.Потапов, Л.И.Розоноэр, и др.[1]-[3],[34]-[37],[84],[88],[93]), так и за рубежом (М.М.Denn, Н.О.Fattorini, А. V.Fridman, J.Warga, L.J.Young и др.).

Оптимальное управление системами интегро-дифференциальных уравнений исследовалось в работах А.Н.Джорбенадзе, В.И.Сумина и др. [56].

Одной из важнейших проблем остается численное решение задач оптимального управления. Наиболее часто встречаются методы конечно-разностной аппроксимации уравнения (методы сеток, прямых); метод моментов, методы, использующие принцип Беллмана, градиентные методы, методы финитного управления. До сих пор продолжается разработка методов, учитывающих специфические особенности того или иного класса экстремальных задач, за счет чего облегчается их численная реализация. Большую сложность при этом представляет проблема обоснования их сходимости.

Исторически развитие численных методов решения задач оптимального управления началось с методов первого порядка, известных как градиентные методы, одновременно с созданием современной теории оптимального управления. В числе основоположников отметим Р. Куранта [110], Д.Е. Охоцимского и Т.М. Эннеева [88–89],[109], Л.В. Канторовича [67], Л.И. Шатровского [107], Дж. Келли [68]. Наряду с этим реализовались и другие методы, родственные градиентным, основанные на принципе максимума Понтрягина [75–76], [40].

Цикл работ по численным методам оптимального управления был выполнен Н.Н. Моисеевым и его учениками. В этих работах был разработан подход,

основанный на вариациях в пространстве состояний. Другое направление, развиваемое Р.П. Федоренко, базируется на использовании идей метода линеаризации. Методы условного градиента и проекции градиента были перенесены В.Ф. Демьяновым и А.М. Рубиновым на задачи оптимального управления.

В третьей четверти XX века делались многочисленные попытки реализовать численные методы решения задач оптимального управления на основе идей динамического программирования, наиболее подробно описанного в книгах Р.Беллмана. Этот подход оказался эффективен лишь для задач небольшой размерности. Решение реальных практических задач требовало чрезвычайно большой машинной памяти. "Проклятие размерности" делало этот подход мало приемлемым для решения практических задач. В настоящее время ситуация заметно изменяется. Стремительный рост производительности и объемов памяти вычислительных систем создали хорошие предпосылки для реализации методов, базирующихся на применении динамического программирования.

В одной из первых монографий по численным методам оптимального управления, написанной Н.Н. Моисеевым [86] в 1971 г., приведено подробное изложение численных схем, идущих от теории оптимального управления, вскользь указано на возможность иного подхода, основанного на методах нелинейного программирования и нашедшего впоследствии широкое развитие, в частности, в книгах Ю.Г. Евтушенко [60], Э. Полака [97], А.И. Пропоя [101] и многих других. В этих работах управление аппроксимируется кусочно-постоянной функцией. Таким образом, производится конечномерная параметризация искомого управления, функционалы исходной задачи становятся функциями конечного числа параметров. Такой подход сводит задачу оптимального управления к оптимизации функций многих переменных, причем задачи оптимального управления с краевыми условиями, с фазовыми ограничениями сравнительно просто сводятся к задачам нелинейного программирования.

В ВЦ РАН под руководством Ю. Г. Евтушенко [5], [60], [116]-[117] разработана методология быстрого автоматического дифференцирования (БАД-методология), позволяющая с единых позиций определять градиенты для явно и неявно определенных функций и для вычислительных процессов, которые являются результатом дискретизации непрерывных систем, описываемых дифференциальными и интегро-дифференциальными уравнениями. В этом подходе математическая модель управляемой системы, ее начальные и граничные условия аппроксимированы соответствующей дискретной задачей выбора оптимальных параметров, которая, в свою очередь, рассматривается как задача математического программирования и, следовательно, может быть решена с помощью существующих алгоритмов нелинейного программирования. Эффективные алгоритмы математического программирования, основанные на градиентной технике, используются для решения задачи оптимального выбора параметров. Алгоритмы градиентного типа обычно требуют существенно меньшего количества итераций и меньшего количества вычислений, чем методы без использования производных.

Этот подход был применен в диссертационной работе для построения оптимального решения.

Наряду с традиционными методами в последнее время бурно развиваются неклассические методы решения задач оптимального управления – генетические алгоритмы [52],[103],[119]. Генетические алгоритмы применяются при разработке программного обеспечения, в системах искусственного интеллекта, оптимизации, искусственных нейронных сетях и в других отраслях знаний. Генетический алгоритм может быть использован в тех случаях, когда не работают традиционные методы, например, когда функция не дифференцируема, или множество допустимых значений управления не компактное.

Основной целью диссертации является разработка численных методов и алгоритмов построения оптимального решения для систем с распределенным запаздыванием в многокритериальной задаче.

Для достижения этой цели решены следующие задачи:

1. Разработка алгоритма построения оптимального управления на основе методологии быстрого автоматического дифференцирования для модели взаимодействия произвольного конечного числа хищников и жертв и модели искусственной нейронной сети, описываемых системами интегро-дифференциальных уравнений;
2. Разработка модифицированного генетического алгоритма для построения оптимального решения в моделях с распределенным запаздыванием;
3. Анализ применения численных методов к решению задач оптимального управления при различных типах минимизируемых функционалов;
4. Разработка практической реализации численного метода построения оптимального решения для систем с распределенным запаздыванием;
5. Анализ влияние параметров методов и задачи на оптимальное решение системы.

Научная новизна проведенного исследования заключается в следующем: в диссертационной работе разработаны численные методы построения оптимального решения для задач с большим числом управляющих параметров. Разработаны алгоритмы построения оптимального решения для систем, описываемых интегро-дифференциальными уравнениями при различных типах минимизируемых функционалов. Разработан модифицированный генетический алгоритм, позволяющий учитывать большую размерность задачи. Построена обобщенная модель типа хищник-жертва с произвольным числом классов хищников и жертв.

Практическая значимость диссертационной работы состоит в том, что результаты работы могут быть использованы для решения конкретных практических задач, связанных с процессом использования ресурсов: планирование добычи, проведение информационно-образовательной работы, вычисление денежных затрат на добычу ресурсов. Программный код предложенных алгоритмов выполнен в форме независимых модулей,

реализованных на языке Java, и может быть использован как составляющий элемент программного комплекса по решению оптимизационных задач.

Результаты диссертационной работы докладывались на международных и всероссийских конференциях:

Воронежской весенней математической школы «Понтрягинские чтения — XXII» (Воронеж, 2011), Международной Научно-практической конференции «Информатизация как целевая ориентация и стратегический ресурс образования» (Архангельск, 2012), Всероссийской научной конференции с международным участием «Спектральная теория операторов и её приложения» (Архангельск, 2012), X Международной научно-технической конференции "Новые информационные технологии системы" (Пенза, 2012), Третьей Российской школы – конференции для молодых ученых с международным участием "Математика, информатика, их приложения и роль в образовании" (Тверь, 2013), I Международной научно-практической конференции «Современные проблемы компьютерных наук (спкн-2013)» (Пенза, 2013), Международной научно-практической конференции "Теоретические и практические вопросы науки XXI века" (Уфа, 2014).

Диссертация состоит из введения, трех глав, заключения и списка литературы из 127 наименования. Общий объем работы составляет 127 страниц, в тексте содержится 19 таблиц и 48 рисунков.

В первой главе рассматривается обобщенная популяционная модель взаимодействия произвольного количества хищников и жертв, которая описывается системой интегро-дифференциальных уравнений, и учитывают прошлое системы. Глава состоит из четырех параграфов. В первом параграфе ставится задача оптимального управления в обобщенной модели хищник-жертва. Пусть $x_i(t)$, $i = 1, m$ - численность популяции жертв i -ого класса, $y_j(t)$, $j = 1, n$ - численность популяции хищников j -ого класса в момент времени $t \in [0, T]$, R_i - численность популяции жертв i -ого класса, недоступная хищникам, тогда

динамика взаимодействия популяций описывается системой интегро-дифференциальных уравнений:

$$\dot{x}_i(t) = x_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) - \sum_{j=1}^n b_{ji} y_j(t) (x_i(t) - R_i) - u_i(t), i = 1, m \quad (1)$$

$$\begin{aligned} \dot{y}_j(t) = & y_j(t) \left(-\alpha_j - \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^m d_{jl} (x_l(t) - R_l) \right) + \\ & + y_j(t) \sum_{l=1}^m \gamma_{jl} \int_{t-r}^t F_{jl}(t-\tau) (x_l(\tau) - R_l) d\tau - v_j(t), j = 1, n \end{aligned} \quad (2)$$

при заданных начальных условиях

$$x_i(0) = x_i^0, x_i(\theta) = \varphi_i(\theta), \theta \in [-r, 0], y_j(0) = y_j^0, i = 1, m, j = 1, n. \quad (3)$$

Здесь m, k - количество классов хищников и жертв соответственно, $x_i(t), i = 1, m$ - численность популяции жертв i -ого класса, $y_j(t), j = 1, k$ - численность популяции хищников j -ого класса в момент времени $t \in [0, T]$, R_i - численность популяции жертв i -ого класса, недоступная хищникам, $e_i, \alpha_j, a_{il}, b_{ji}, c_{jl}, d_{jl}, \gamma_{jl}$ - действительные положительные коэффициенты, характеризующие взаимодействие популяций хищников и жертв. Функции распределения $F_{jl}(t-\tau)$ характеризуют состояния системы в прошлом, описывая влияние доступной для хищника жертвы на характерном отрезке времени, связанном с ростом популяции жертв. Функциями управления являются функции $u_i(t)$ - скорость отлова популяции жертв i -ого класса, $v_j(t)$ - скорость отлова популяции хищников j -ого класса, удовлетворяющие ограничениям

$$0 \leq u_i(t) \leq u_{i\max}, 0 \leq v_j(t) \leq v_{j\max}, i = 1, m, j = 1, n, \quad (4)$$

где $u_{i\max}, v_{j\max}$ - максимальные скорости отлова популяции жертв и хищников соответственно, далее $u = (u_1, \dots, u_m), v = (v_1, \dots, v_n)$.

Оптимальное управление отловом строится из условия оптимизации заданного функционала

$$J(u, v) = \int_0^T f_0(t, x, y, u, v) dt + \Phi(x(T), y(T)), \quad (5)$$

или решается многокритериальная задача. В частности, если целью отлова является получение максимальной прибыли от продажи популяций, то максимизируемый функционал в задаче (1)-(4) имеет вид

$$J_1(u, v) = \int_0^T \left[\sum_{i=1}^m [\rho_i(t, x_i) u_i - d_i(t, u_i)] + \sum_{j=1}^n [\tilde{\rho}_j(t, y_j) v_j - \tilde{d}_j(t, v_j)] \right] e^{-\lambda t} dt, \quad (6)$$

здесь $\rho_i(t, x_i), \tilde{\rho}_j(t, y_j)$ - стоимость единицы соответствующей популяции, $d_i(t, u_i), \tilde{d}_j(t, v_j)$ - стоимость отлова популяции единицу времени, λ - дисконтирующий множитель.

Терминальный функционал

$$J_2(u, v) = \sum_{i=1}^m M_i (x_i(T) - A_i)^2 + \sum_{j=1}^n N_j (y_j(T) - B_j)^2, \quad (7)$$

отвечает за сохранность популяции жертв и хищников в момент времени T на заданном уровне $A_i, i = 1, m$ и $B_j, j = 1, n$ соответственно.

Граничные условия в виде неравенств $x_i(T) \geq A_i, i = 1, m, y_j(T) \geq B_j, j = 1, n$ учитываются функционалом

$$J_3(u, v) = \sum_{i=1}^m M_i (A_i - x_i(T))_+^{2n} + \sum_{j=1}^n N_j (B_j - y_j(T))_+^{2n}, \quad (8)$$

здесь $M_i, i = 1, m, N_j, j = 1, n$ - положительные весовые коэффициенты.

Заметим, что если в предложенной модели используется предположение о том, что $x_i(t) \geq A_i, i = 1, m, y_j(t) \geq B_j, j = 1, n, t \in [0, T]$, это предполагает ограничения на выбор коэффициентов динамической системы или использования принципа максимума для задачи с фазовыми ограничениями. С вычислительной точки зрения фазовые ограничения учитываются с помощью штрафного функционала

$$J_4(u, v) = \sum_{i=1}^m M_i \int_0^T (A_i - x_i(t))_+^{2n} dt + \sum_{j=1}^n N_j \int_0^T (B_j - y_j(t))_+^{2n} dt, \quad (9)$$

Во втором параграфе с помощью обобщенной БАД-методологии вычисляется градиент целевой функции. Приводятся также формулы для вычисления градиента целевой функции, соответствующие прямому методу дифференцирования. Показано, что для задачи оптимального управления (1)-(5) применение метода быстрого автоматического дифференцирования для вычисления градиента целевой функции в $q(n+m)$ раз менее трудоемко, чем применение прямого метода дифференцирования в зависимости от размерности задачи $(n+m)$ и шага аппроксимации $\Delta t = T / q$. Разработан алгоритм построения оптимального решения на основе метода БАД. Проведен анализ влияния параметров на решение задачи. В третьем параграфе первой главы разработан алгоритм построения оптимального управления для задачи (1)-(5) на основе генетического алгоритма. С помощью численных экспериментов определены оптимальные параметры алгоритма. В четвертом параграфе проведено сравнение результатов работы метода БАД и генетического алгоритма.

Во второй главе рассматривается модель искусственной нейронной сети, динамика которой описывается системой интегро-дифференциальных уравнений. Целью оптимизации является определение оптимальных весовых коэффициентов, характеризующих связи между нейронами. Глава состоит из пяти параграфов. В первом параграфе ставится задача оптимального управления, которая заключается в минимизации функционала:

$$J(\omega, u) = \int_0^T E(x(t), \omega(t), u(t)) dt + \Phi(x(T)) \quad (10)$$

где $E(x(t), \omega(t), u(t))$ - энергия рассматриваемой нейронной сети, зависящая от текущей конфигурации сети $x(t)$,

$\omega(t)$ - весовые коэффициенты системы,

$u(t)$ - внешнее воздействие на систему в момент времени t ;

$\Phi(x(T))$ - терминальное слагаемое - характеризует состояние системы в конечный момент времени

Динамика нейронной сети описывается системой интегро-дифференциальных уравнений:

$$\dot{x}_i(t) = -\beta_i x_i(t) + f_i \left(\int_{t-r}^t \sum_{j=1}^n \omega_{ij}(\tau) x_j(\tau - \delta_j) d\tau \right) + \gamma_i u_i(t), \quad (11)$$

$$t \in [0, T], \quad i = \overline{1, n},$$

где функция $x_i(t)$ характеризует состояние i -го нейрона в момент времени t , первое слагаемое $(-\beta_i x_i(t))$ характеризует затухание (β_i - скорость затухания возбуждения i -го нейрона), второе слагаемое $f_i \left(\int_{t-r}^t \sum_{j=1}^n \omega_{ij}(\tau) x_j(\tau - \delta_j) d\tau \right)$ характеризует внешнее воздействие на нейрон всех других нейронов, f_i - функция активации, $\omega_{ij}(\tau)$ - весовой коэффициент, воздействие на i -ый нейрон, j -го нейрона, при этом $\omega_{ii}(\tau) \equiv 0$, третье слагаемое $\gamma_i u_i(t)$ - внешнее воздействие на нейрон в момент времени t .

Начальные условия:

$$x_i(t) = \theta_i(t), \quad t \in [-r - \delta, 0], \quad \delta = \max_{i=1, n} \delta_i \quad (12)$$

где τ, δ_i - запаздывания, $\theta_i(t)$, $i = \overline{1, n}$, - заданные непрерывные функции, $\varepsilon_i, \beta_i, \delta_j, \gamma_i$, $i, j = \overline{1, n}$, - заданные неотрицательные параметры модели.

Весовые коэффициенты $\omega_{ij}(t)$ и внешние воздействия $u_i(t)$ ограничены

$$|\omega_{ij}(t)| \leq a_{ij}, \quad |u_i(t)| \leq a_i, \quad i, j = \overline{1, n}, \quad \text{п.в. } t \in [t_0, t_1] \quad (13)$$

где a_{ij}, a_i , $i, j = \overline{1, n}$, заданные положительные параметры модели, полагаем $\omega_{ii}(t) \equiv 0$.

Задача оптимального управления заключается в том, чтобы обучить нейронную сеть таким образом, чтобы минимизировать функционал (10) и

построить оптимальные значения весовых коэффициентов, внешнего управляющего воздействия.

Во втором параграфе с помощью обобщенной БАД-методологии решается задача обучения искусственной нейронной сети. Разработан алгоритм построения оптимальных значений весовых коэффициентов, внешнего управляющего воздействия. Проведен анализ влияния параметров на решение задачи. В третьем параграфе второй главы разработан алгоритм построения оптимального управления для задачи (10)-(13) на основе генетического алгоритма. Проведен анализ влияния параметров метода на работу алгоритма. В четвертом параграфе проведено сравнение результатов работы метода БАД и генетического алгоритма. В пятом параграфе задача обучения искусственной нейронной сети рассмотрена как задача оптимального управления с нефиксированным временем процесса. Исходная задача сведена к дискретной задаче оптимального управления, которая решается на основе методологии быстрого автоматического дифференцирования.

В третьей главе рассматриваются общие сведения о методах сведения исходной многокритериальной задачи к задачам с единым критерием. Исследованы методы сведения многокритериальной задачи к задаче с одним критерием такие, как линейная свертка, способ свертки с неотрицательными весовыми коэффициентами, принцип гарантированного результата. Рассмотрены этапы построения программной реализации разработанных алгоритмов. В частности, рассматриваются вопросы проектирования пользовательского интерфейса, разработка модульных тестов, компьютерные программы для статистической обработки данных.

Глава 1. Обобщенная популяционная модель хищник-жертва, описываемая системой интегро-дифференциальных уравнений типа Вольтерра

Математические модели, описываемые интегро-дифференциальными уравнениями, возникают при исследовании экологических, физических, экономических процессов. Эти модели являются неавтономными, поэтому важным вопросом является исследование периодических решений, их устойчивости и управляемости, построение оптимального управления и разработка численных методов и алгоритмов для систем, описываемых интегро-дифференциальными уравнениями или дифференциальными уравнениями с распределенным запаздыванием. Рассмотрим задачи оптимального управления для систем, описываемых интегро-дифференциальными уравнениями типа Вольтерра.

§1. Постановка задачи оптимального управления в обобщенной популяционной модели хищник-жертва

В данном параграфе рассматривается модель взаимодействия произвольного количества хищников и жертв, которая описывается системой интегро-дифференциальных уравнений и учитывают прошлое системы.

Пусть далее $x_i(t)$, $i = 1, m$ - численность популяции жертв i -ого класса, $y_j(t)$, $j = 1, n$ - численность популяции хищников j -ого класса в момент времени $t \in [0, T]$, R_i - численность популяции жертв i -ого класса, недоступная хищникам. Динамика взаимодействия популяций описывается системой интегро-дифференциальных уравнений:

$$\dot{x}_i(t) = x_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) - \sum_{j=1}^n b_{ji} y_j(t) (x_i(t) - R_i) - u_i(t), i = 1, m \quad (1. 1)$$

$$\begin{aligned} \dot{y}_j(t) = y_j(t) & \left(-\alpha_j - \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^m d_{jl} (x_l(t) - R_l) \right) + \\ & + y_j(t) \sum_{l=1}^m \gamma_{jl} \int_{t-r}^t F_{jl}(t-\tau) (x_l(\tau) - R_l) d\tau - v_j(t), \quad j=1, n \end{aligned} \quad (1.2)$$

при заданных начальных условиях

$$x_i(0) = x_i^0, x_i(\theta) = \varphi_i(\theta), \theta \in [-r, 0], y_j(0) = y_j^0, i=1, m, j=1, n \quad (1.3)$$

Здесь $e_i, \alpha_j, a_{il}, b_{jl}, c_{jl}, d_{jl}, \gamma_{jl}$ - действительные положительные коэффициенты, характеризующие взаимодействие популяций хищников и жертв. Функции распределения $F_{jl}(t-\tau)$ характеризуют состояния системы в прошлом, описывая влияние доступной для хищника жертвы на характерном отрезке времени, связанном с ростом популяции жертв. Функциями управления являются функции $u_i(t)$ - скорость отлова популяции жертв i -ого класса, $v_j(t)$ - скорость отлова популяции хищников j -ого класса, удовлетворяющие ограничениям

$$0 \leq u_i(t) \leq u_{i\max}, \quad 0 \leq v_j(t) \leq v_{j\max}, \quad i=1, m, j=1, n, \quad (1.4)$$

где $u_{i\max}, v_{j\max}$ - максимальные скорости отлова популяции жертв и хищников соответственно, далее $u = (u_1, \dots, u_m), v = (v_1, \dots, v_n)$.

Заметим, что на управляющие функции могут быть наложены и другие ограничения, зависящие от технологии отлова. Например, на управления могут быть дополнительно наложены суммарные ограничения типа

$$\begin{aligned} \sum_{i=1}^m \gamma_i u_i(t) & \leq B, \quad 0 \leq u_i(t) \leq u_{i\max}, \\ \sum_{j=1}^n \rho_j v_j(t) & \leq A, \quad 0 \leq v_j(t) \leq v_{j\max}, \end{aligned} \quad (1.4')$$

Если ставится задача отлова части популяции, то в зависимости от способов отлова в динамических уравнениях можно использовать управление типа $u_i(t) = \alpha_i x_i u(t), v_i(t) = \beta_i y_i v(t)$ и др.

Оптимальное управление отловом строится из условия оптимизации заданного функционала

$$J(u, v) = \int_0^T f_0(t, x, y, u, v) dt + \Phi(x(T), y(T)), \quad (1.5)$$

или решается многокритериальная задача. В частности, если целью отлова является получение максимальной прибыли от продажи популяций, то максимизируемый функционал в задаче (1.1)-(1.5) имеет вид

$$J_1(u, v) = \int_0^T \left[\sum_{i=1}^m [\rho_i(t, x_i) u_i - d_i(t, u_i)] + \sum_{j=1}^n [\tilde{\rho}_j(t, y_j) v_j - \tilde{d}_j(t, v_j)] \right] e^{-\lambda t} dt, \quad (1.6)$$

здесь $\rho_i(t, x_i), \tilde{\rho}_j(t, y_j)$ - стоимость единицы соответствующей популяции, $d_i(t, u_i), \tilde{d}_j(t, v_j)$ - стоимость отлова популяции единицу времени, λ - дисконтирующий множитель.

Терминальный функционал

$$J_2(u, v) = \sum_{i=1}^m M_i (x_i(T) - A_i)^2 + \sum_{j=1}^n N_j (y_j(T) - B_j)^2, \quad (1.7)$$

отвечает за сохранность популяции жертв и хищников в момент времени T на заданном уровне $A_i, i = 1, m$ и $B_j, j = 1, n$ соответственно.

Граничные условия в виде неравенств $x_i(T) \geq A_i, i = 1, m, y_j(T) \geq B_j, j = 1, n$ учитываются функционалом

$$J_3(u, v) = \sum_{i=1}^m M_i (A_i - x_i(T))_+^{2n} + \sum_{j=1}^n N_j (B_j - y_j(T))_+^{2n}, \quad (1.8)$$

здесь $M_i, i = 1, m, N_j, j = 1, n$ - положительные весовые коэффициенты.

Заметим, что если в предложенной модели используется предположение о том, что $x_i(t) \geq A_i, i = 1, m, y_j(t) \geq B_j, j = 1, n$, это предполагает ограничения на выбор коэффициентов динамической системы или использования принципа максимума для задачи с фазовыми ограничениями. С вычислительной точки зрения фазовые ограничения учитываются с помощью штрафного функционала

$$J_4(u, v) = \sum_{i=1}^m M_i \int_0^T (A_i - x_i(t))_+^{2n} dt + \sum_{j=1}^n N_j \int_0^T (B_j - y_j(t))_+^{2n} dt, \quad (1.9)$$

Оптимальное управление $\bar{u}(t), \bar{v}(t), t \in [0, T]$, в задаче (1.1)-(1.5) удовлетворяет принципу максимума

$$\max_{0 \leq \omega_i \leq u_{\max}, 0 \leq \psi_j \leq v_{\max}} \left[-\lambda_0 f_0(t, \bar{x}(t), \bar{y}(t), \omega, \psi) - \sum_{i=1}^m p_i(t) \omega_i - \sum_{j=1}^n r_j(t) \psi_j \right] =$$

$$- \lambda_0 f_0(t, \bar{x}(t), \bar{y}(t), \bar{u}(t), \bar{v}(t)) - \sum_{i=1}^m p_i(t) \bar{u}_i(t) - \sum_{j=1}^n r_j(t) \bar{v}_j(t), \quad (1.10)$$

$$\text{где } f_0(t, x, y, u, v) = \left[\sum_{i=1}^m [\rho_i(t, x_i) u_i - d_i(t, u_i)] + \sum_{j=1}^n [\tilde{\rho}_j(t, y_j) v_j - \tilde{d}_j(t, v_j)] \right] e^{-\lambda t},$$

а сопряженные функции $p_i(t), r_j(t), i = 1, m, j = 1, n$ являются решением системы интегро-дифференциальных уравнений:

$$\dot{p}_i(t) = -p_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) + \sum_{l=1}^m p_l(t) a_{li} x_l(t) + p_i(t) \sum_{j=1}^n b_{ji} y_j(t)$$

$$- \sum_{j=1}^n r_j(t) y_j(t) d_{ji} - \sum_{j=1}^n r_j(t) y_j(t) \gamma_{ji} \int_t^{t+r} F_{ji}(\tau - t) d\tau, \quad i = 1, m$$

$$\dot{r}_j(t) = r_j(t) \alpha_j + r_j(t) \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^m p_l(t) b_{lj} (x_l(t) - R_l)$$

$$+ \sum_{l=1}^n r_l(t) y_l(t) c_{lj} - r_l(t) \sum_{l=1}^m d_{jl} (x_l(t) - R_l) -$$

$$- r_j(t) \sum_{l=1}^m \gamma_{jl} \int_{t-r}^t F_{jl}(t - \tau) (x_l(\tau) - R_l) d\tau, \quad j = 1, n$$

с условиями трансверсальности на правом конце

$$p_i(T) = -\lambda_0 \frac{\partial J_2}{\partial x_i}(x(T), y(T)), i = 1, m, \quad p_i(t) \equiv 0 \text{ при } t > T.$$

$$r_j(T) = -\lambda_0 \frac{\partial J_2}{\partial y_j}(x(T), y(T)), j = 1, n, \quad r_j(t) \equiv 0 \text{ при } t > T.$$

Если в исходной задаче (1.1)-(1.5) функция распределения, характеризующая состояния системы в прошлом, имеет вид $F_{jl}(t - \tau) = \eta_{jl} e^{\varepsilon_{jl}(t - \tau)}$, то, вводя обозначения $z_{jl}(t) = \int_{t-r}^t F_{jl}(t - \tau) (x_l(\tau) - R_l) d\tau$, исходная система уравнений (1.1)-

(1.2) перепишется в виде системы дифференциальных уравнений с постоянным запаздыванием (1.11)-(1.13):

$$\dot{x}_i(t) = x_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) - \sum_{j=1}^n b_{ji} y_j(t) (x_i(t) - R_i) - u_i(t), i = 1, m, \quad (1.11)$$

$$\begin{aligned} \dot{y}_j(t) = & y_j(t) \left(-\alpha_j - \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^m d_{jl} (x_l(t) - R_l) \right) + \\ & + y_j(t) \sum_{l=1}^m \gamma_{jl} z_{jl}(t) - v_j(t), j = 1, n \end{aligned} \quad (1.12)$$

$$\dot{z}_{jl}(t) = F_{jl}(0)(x_l(t) - R_l) - F_{jl}(r)(x_l(t-r) - R_l) + \varepsilon_{jl} z_{jl}, j = 1, n, l = 1, m, \quad (1.13)$$

Оптимальное управление в задаче (1.11)-(1.13), (1.4)-(1.5) $\bar{u}(t), \bar{v}(t), t \in [0, T]$, удовлетворяет принципу максимума (1.10), а сопряженные функции $p_i(t), r_j(t), s_{ji}(t), i = 1, m, j = 1, n$ являются решением системы дифференциальных уравнений с отклоняющимся аргументом:

$$\begin{aligned} \dot{p}_i(t) = & \lambda_0 \left(\frac{\partial f_0(t, x(t), y(t), u(t), v(t))}{\partial x_i} \right) - p_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) + \sum_{l=1}^m p_l(t) x_l(t) a_{li} + \\ & p_i(t) \sum_{j=1}^n b_{ji} y_j(t) - \sum_{j=1}^n r_j(t) y_j(t) d_{ji} - \sum_{j=1}^n \left(s_{ji}(t) \eta_{ji} - s_{ji}(t+r) e^{\varepsilon_{ji} r} \right), \\ \dot{r}_j(t) = & \lambda_0 \left(\frac{\partial f_0(t, x(t), y(t), u(t), v(t))}{\partial y_j} \right) + r_j(t) \alpha_j + r_j(t) \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^n r_l(t) y_l(t) c_{lj} - \\ & - r_j(t) \sum_{l=1}^m d_{jl} (x_l(t) - R_l) - r_j(t) \sum_{l=1}^m \gamma_{jl} z_{jl}(t) + \sum_{i=1}^m p_i(t) b_{ij} (x_i(t) - R_i), \\ \dot{s}_{ji}(t) = & -r_j(t) y_j(t) \gamma_{ji} - \varepsilon_{ji} s_{ji}(t), \end{aligned}$$

с граничными условиями на правом конце

$$p_i(T) = -\lambda_0 \frac{\partial \Phi}{\partial x_i}(x(T), y(T)), i = 1, m, \quad p_i(t) \equiv 0 \text{ при } t > T.$$

$$r_j(T) = -\lambda_0 \frac{\partial \Phi}{\partial y_j}(x(T), y(T)), j = 1, n, r_j(t) \equiv 0 \text{ при } t > T.$$

$$s_{jl}(T) = 0, j = 1, n, l = 1, m, s_{jl}(t) \equiv 0 \text{ при } t > T.$$

Если r мало, то $x_l(t-r) = x_l(t) - r\dot{x}_l(t) + O(r)$, то исходная система уравнений (1.1)-(1.2) перепишется в виде системы дифференциальных уравнений (1.14)-(1.16) :

$$\dot{x}_i(t) = x_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) - \sum_{j=1}^n b_{ji} y_j(t) (x_i(t) - R_i) - u_i(t), i = 1, m \quad (1.14)$$

$$\begin{aligned} \dot{y}_j(t) = & y_j(t) \left(-\alpha_j - \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^m d_{jl} (x_l(t) - R_l) \right) + \\ & + y_j(t) \sum_{l=1}^m \gamma_{jl} z_{jl}(t) - v_j(t), j = 1, n \end{aligned} \quad (1.15)$$

$$\begin{aligned} \dot{z}_{ji}(t) = & F_{ji}(0)(x_i(t) - R_i) - F_{ji}(r)(x_i(t) - r x_i(t)) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) + \\ & r \sum_{j=1}^n b_{ji} y_j(t) (x_i(t) - R_i) + r u_i(t) - R_i + \varepsilon_{ji} z_{ji}, j = 1, n, i = 1, m \end{aligned} \quad (1.16)$$

Оптимальное управление в задаче (1.14)-(1.16), (1.4)-(1.5), $\bar{u}(t), \bar{v}(t), t \in [0, T]$, удовлетворяет принципу максимума

$$\begin{aligned} \max_{0 \leq \omega_i \leq u_{\max}, 0 \leq \psi_j \leq v_{\max}} & \left[-\lambda_0 f_0(t, \bar{x}(t), \bar{y}(t), \omega, \psi) - \sum_{i=1}^m p_i(t) \omega_i - \sum_{j=1}^n r_j(t) \psi_j \right] = \\ & -\lambda_0 f_0(t, \bar{x}(t), \bar{y}(t), \bar{u}(t), \bar{v}(t)) - \sum_{i=1}^m \left(p_i(t) + \sum_{j=1}^n s_{ji}(t) \eta_{ji} e^{\varepsilon_{ji} r} r \right) \bar{u}_i(t) - \sum_{j=1}^n r_j(t) \bar{v}_j(t), \end{aligned}$$

а сопряженные функции $p_i(t), r_j(t), s_{ji}(t), i = 1, m, j = 1, n$ являются решением системы дифференциальных уравнений с отклоняющимся аргументом:

$$\begin{aligned}
\dot{p}_i(t) = & \lambda_0 \left(\frac{\partial f_0(t, x(t), y(t), u(t), v(t))}{\partial x_i} \right) - p_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) + \sum_{l=1}^m p_l(t) x_l(t) a_{li} + \\
& + p_i(t) \sum_{j=1}^n b_{ji} y_j(t) - \sum_{j=1}^n r_j(t) y_j(t) d_{ji} - \sum_{j=1}^n \left(s_{ji}(t) \eta_{ji} - s_{ji}(t) e^{\varepsilon_{ji} r} \right) - \\
& - \sum_{j=1}^n \left(s_{ji}(t) \eta_{ji} e^{\varepsilon_{ji} r} r \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) \right) + r \sum_{j=1}^n \sum_{l=1}^m \left(s_{jl}(t) \eta_{jl} e^{\varepsilon_{jl} r} x_l(t) a_{li} \right) \\
& + r \sum_{j=1}^n \left(s_{ji}(t) \eta_{ji} e^{\varepsilon_{ji} r} \sum_{l=1}^n b_{li} y_l(t) \right), \quad i = \overline{1, m} \\
\dot{r}_j(t) = & \lambda_0 \left(\frac{\partial f_0(t, x(t), y(t), u(t), v(t))}{\partial y_j} \right) + r_j(t) \alpha_j + r_j(t) \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^n r_l(t) y_l(t) c_{lj} - \\
& - r_j(t) \sum_{l=1}^m d_{jl} (x_l(t) - R_l) - r_j(t) \sum_{l=1}^m \gamma_{jl} z_{jl}(t) + \sum_{i=1}^m p_i(t) b_{ij} (x_i(t) - R_i) + \\
& + r \sum_{l=1}^m \left(s_{jl}(t) \eta_{jl} e^{\varepsilon_{jl} r} b_{jl} (x_l(t) - R_l) \right), \quad j = \overline{1, n}
\end{aligned}$$

$$\dot{s}_{ji}(t) = -r_j(t) y_j(t) \gamma_{ji} - \varepsilon_{ji} s_{ji}(t), \quad i = \overline{1, m}, j = \overline{1, n}$$

с граничными условиями на правом конце

$$p_i(T) = -\lambda_0 \frac{\partial \Phi}{\partial x_i}(x(T), y(T)), i = \overline{1, m},$$

$$r_j(T) = -\lambda_0 \frac{\partial \Phi}{\partial y_j}(x(T), y(T)), j = \overline{1, n},$$

$$s_{jl}(T) = 0, j = \overline{1, n}, l = \overline{1, m}.$$

Если в исходной задаче (1.1)-(1.2) функция распределения, характеризующая состояния системы в прошлом, имеет вид $F_{jl}(t - \tau) = \begin{cases} \eta_{jl}, \tau \in [t - r, t] \\ 0, \tau \notin [t - r, t] \end{cases}$, то,

обозначив $z_{jl}(t) = \int_{t-r}^t F_{jl}(t - \tau) (x_l(\tau) - R_l) d\tau$, исходная система уравнений (1.1)-(1.2)

перепишется в виде системы дифференциальных уравнений с запаздыванием (1.17)-(1.19):

$$\dot{x}_i(t) = x_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) - \sum_{j=1}^n b_{ji} y_j(t) (x_i(t) - R_i) - u_i(t), i = 1, m \quad (1.17)$$

$$\begin{aligned} \dot{y}_j(t) = & y_j(t) \left(-\alpha_j - \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^m d_{jl} (x_l(t) - R_l) \right) + \\ & + y_j(t) \sum_{l=1}^m \gamma_{jl} z_{jl}(t) - v_j(t), j = 1, n \end{aligned} \quad (1.18)$$

$$\dot{z}_{jl}(t) = \eta_{jl} (x_l(t) - x_l(t-r)), j = 1, n, l = 1, m \quad (1.19)$$

Оптимальное управление в задаче (1.17)-(1.19), (1.4)-(1.5), $\bar{u}(t), \bar{v}(t), t \in [0, T]$, удовлетворяет принципу максимума (1.10), а сопряженные функции $p_i(t), r_j(t), s_{ji}(t), i = 1, m, j = 1, n$ являются решением системы интегро-дифференциальных уравнений:

$$\begin{aligned} \dot{p}_i(t) = & \lambda_0 \left(\frac{\partial f_0(t, x(t), y(t), u(t), v(t))}{\partial x_i} \right) - p_i(t) \left(e_i - \sum_{l=1}^m a_{il} x_l(t) \right) + \sum_{l=1}^m p_l(t) x_l(t) a_{li} + \\ & p_i(t) \sum_{j=1}^n b_{ji} y_j(t) - \sum_{j=1}^n r_j(t) y_j(t) d_{ji} - \eta_{ji} \sum_{j=1}^n (s_{ji}(t) - s_{ji}(t+r)), \\ \dot{r}_j(t) = & \lambda_0 \left(\frac{\partial f_0(t, x(t), y(t), u(t), v(t))}{\partial y_j} \right) + r_j(t) \alpha_j + r_j(t) \sum_{l=1}^n c_{jl} y_l(t) + \sum_{l=1}^n r_l(t) y_l(t) c_{lj} - \\ & - r_j(t) \sum_{l=1}^m d_{jl} (x_l(t) - R_l) - r_j(t) \sum_{l=1}^m \gamma_{jl} z_{jl}(t) + \sum_{i=1}^m p_i(t) b_{ij} (x_i(t) - R_i), \\ \dot{s}_{ji}(t) = & -r_j(t) y_j(t) \gamma_{ji}, \end{aligned}$$

с граничными условиями на правом конце

$$p_i(T) = -\lambda_0 \frac{\partial \Phi}{\partial x_i}(x(T), y(T)), i = 1, m,$$

$$r_j(T) = -\lambda_0 \frac{\partial \Phi}{\partial y_j}(x(T), y(T)), j = 1, n,$$

$$s_{jl}(T) = 0, j = 1, n, l = 1, m.$$

Заметим, что, если в функционале (1.6) функции $d_i(t, u_i) = d_i(t)u_i$, $\tilde{d}_j(t, v_j) = \tilde{d}_j(t)v_j$ линейны по управлению и на управление наложены ограничения (1.4), то оптимальное управление определяется знаком функций переключения $\psi_i(t)$, $\varphi_j(t)$:

$$\bar{u}_i(t) = \begin{cases} u_{i\max}, & \text{если } \psi_i(t) > 0 \\ 0, & \text{если } \psi_i(t) < 0 \end{cases}$$

$$\bar{v}_j(t) = \begin{cases} v_{j\max}, & \text{если } \varphi_j(t) > 0 \\ 0, & \text{если } \varphi_j(t) < 0 \end{cases}$$

$$\text{где } \psi_i(t) = \lambda_0 e^{-\lambda t} (\rho_i(t, x_i) - d_i(t)) - p_i(t), \quad i = \overline{1, m},$$

$$\varphi_j(t) = \lambda_0 e^{-\lambda t} (\tilde{\rho}_j(t, y_j) - \tilde{d}_j(t)) - r_j(t), \quad j = \overline{1, n}$$

В случае если на управление наложены ограничения (1.4'), то оптимальное управление определяется из условий:

$$\begin{aligned} \sum_{i=1}^m u_i \psi_i(t) &\rightarrow \max, \quad \sum_{i=1}^m \gamma_i u_i \leq B \\ \sum_{j=1}^k v_j \varphi_j(t) &\rightarrow \max, \quad \sum_{j=1}^k \tilde{\gamma}_j v_j \leq A, \end{aligned}$$

Введем обозначения $\gamma_i u_i = \tilde{u}_i$, $\tilde{\gamma}_j v_j = \tilde{v}_j$ и $\frac{\psi_i}{\gamma_i} = \tilde{\psi}_i$, $\frac{\varphi_j}{\tilde{\gamma}_j} = \tilde{\varphi}_j$, тогда из принципа

максимума следует

$$\begin{aligned} \sum_{i=1}^m \tilde{u}_i \tilde{\psi}_i &\rightarrow \max \quad \sum \tilde{u}_i \leq B \\ \sum_{j=1}^k \tilde{v}_j \tilde{\varphi}_j &\rightarrow \max \quad \sum \tilde{v}_j \leq A \end{aligned}$$

Если $0 < \tilde{\psi}_{i_0} = \max_i \tilde{\psi}_i$, $0 < \tilde{\varphi}_{j_0} = \max_j \tilde{\varphi}_j$, тогда

$$\bar{u}_i(t) = \begin{cases} \frac{B}{\gamma_{i\max}}, & \text{если } \frac{B}{\gamma_{i\max}} \leq u_{i\max}, \tilde{\psi}_i(t) = \tilde{\psi}_{i\max} > 0, \\ u_{i\max}, & \text{если } \frac{B}{\gamma_{i\max}} > u_{i\max}, \tilde{\psi}_i(t) = \tilde{\psi}_{i\max} > 0, \\ 0, & \text{если } \tilde{\psi}_i(t) \neq \tilde{\psi}_{i\max} > 0 \\ 0, & \text{если } \tilde{\psi}_i(t) < 0 \end{cases}$$

$$\bar{v}_j(t) = \begin{cases} \frac{A}{\tilde{\gamma}_{j\max}}, & \text{если } \frac{A}{\tilde{\gamma}_{j\max}} \leq v_{j\max}, \tilde{\phi}_j(t) = \tilde{\phi}_{j\max}(t) > 0 \\ v_{j\max}, & \text{если } \frac{A}{\tilde{\gamma}_{j\max}} > v_{j\max}, \tilde{\phi}_j(t) = \tilde{\phi}_{j\max}(t) > 0 \\ 0, & \text{если } \tilde{\phi}_j(t) \neq \tilde{\phi}_{j\max}(t) > 0 \\ 0, & \text{если } \tilde{\phi}_j(t) < 0 \end{cases}$$

Заметим, что

если $\tilde{\psi}_{i_0} = \tilde{\psi}_{l_0}$, то $\tilde{u}_{i_0} + \tilde{u}_{l_0} = B$, где,

если $B < \gamma_{i_0} u_{i_0\max}$, то $0 \leq \tilde{u}_{i_0} \leq B$, если $B < \gamma_{l_0} u_{l_0\max}$, то $0 \leq \tilde{u}_{l_0} \leq B$,

если $\gamma_{i_0} u_{i_0\max} > B > \gamma_{i_0} u_{i_0\max} + \gamma_{l_0} u_{l_0\max}$, (1.20)

то $B - \gamma_{l_0} u_{l_0\max} \leq \tilde{u}_{i_0} \leq \gamma_{i_0} u_{i_0\max}$,

если $\gamma_{l_0} u_{l_0\max} > B > \gamma_{i_0} u_{i_0\max} + \gamma_{l_0} u_{l_0\max}$, (1.21)

то $B - \gamma_{i_0} u_{i_0\max} \leq \tilde{u}_{l_0} \leq \gamma_{l_0} u_{l_0\max}$, $\tilde{u}_i = 0$, $i \neq i_0, i \neq l_0$ $i = \overline{1, m}$

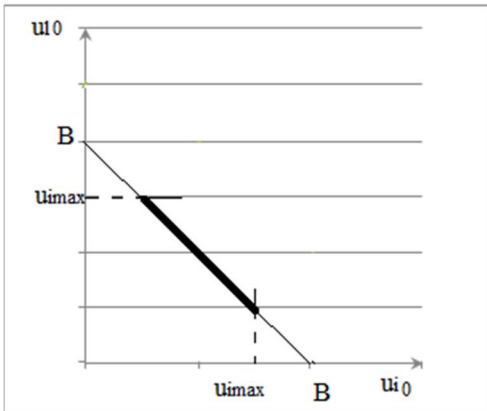


Рис. 1.1 Множество оптимальных управлений $\tilde{u}_{i_0} + \tilde{u}_{l_0} = B$ при ограничениях (1.20), (1.21)

Аналогично, для функции $\tilde{\varphi}_j$. Если $\tilde{\varphi}_{j_0} = \tilde{\varphi}_{l_0}$, то $\tilde{v}_{j_0} + \tilde{v}_{l_0} = A$,

где, если $A < \tilde{\gamma}_{j_0} v_{j_0 \max}$, то $0 \leq \tilde{v}_{j_0} \leq A$, если $A < \tilde{\gamma}_{l_0} v_{l_0 \max}$, то $0 \leq \tilde{v}_{l_0} \leq A$,

если $\tilde{\gamma}_{i_{j_0}} v_{j_0 \max} > A > \tilde{\gamma}_{j_0} v_{j_0 \max} + \tilde{\gamma}_{l_0} v_{l_0 \max}$, то $A - \tilde{\gamma}_{l_0} v_{l_0 \max} \leq \tilde{v}_{j_0} \leq \tilde{\gamma}_{j_0} v_{j_0 \max}$,

если $\tilde{\gamma}_{l_0} v_{l_0 \max} > A > \tilde{\gamma}_{j_0} v_{j_0 \max} + \tilde{\gamma}_{l_0} v_{l_0 \max}$, то $A - \tilde{\gamma}_{j_0} v_{j_0 \max} \leq \tilde{v}_{l_0} \leq \tilde{\gamma}_{l_0} v_{l_0 \max}$,

$\tilde{v}_j = 0, \quad j \neq j_0, j \neq l_0 \quad j = \overline{1, n}$

Рассмотрим дискретную аппроксимацию начальной задачи (1.1)-(1.5), в которой используется правило левых прямоугольников для аппроксимации интеграла и схема Эйлера аппроксимации производных. Выберем шаг интегрирования $\Delta t = T / q$ так, что $r = v \cdot \Delta t$, $t_k = k \Delta t$, $k = 0, \dots, q$, $t_q = T$.

Дискретная задача оптимального управления состоит в минимизации функции

$$I = \Delta t \sum_{k=0}^{q-1} F_1(x^k, y^k, u^k, v^k) + F_0(x^q, y^q) \rightarrow \inf \quad (1.22)$$

при заданных рекуррентных соотношениях:

$$x_i^{k+1} = x_i^k + \Delta t x_i^k \left(e_i - \sum_{l=1}^m a_{il} x_l^k \right) - \Delta t \sum_{j=1}^n b_{ji} y_j^k (x_i^k - R_i) - \Delta t u_i^k, \quad i = \overline{1, m} \quad (1.23)$$

$$\begin{aligned} y_j^{k+1} = & y_j^k + \Delta t y_j^k (-\alpha_j - \sum_{l=1}^n c_{jl} y_l^k + \sum_{l=1}^m d_{jl} (x_l^k - R_l)) \\ & + \Delta t^2 y_j^k \sum_{l=1}^m \gamma_{jl} \sum_{s=k-v}^{k-1} F_{jl}^{k-s} (x_l^s - R_l) - \Delta t v_j^k, \quad j = \overline{1, n} \end{aligned} \quad (1.24)$$

начальных условиях

$$x_i^l = \varphi_i(t_l), l = \overline{-v, 0}, y_j^0 = \theta_j \quad i = \overline{1, m}, j = \overline{1, n} \quad (1.25)$$

и ограничениях на управление:

$$0 \leq u_i^k \leq u_{i \max}, \quad 0 \leq v_j^k \leq v_{j \max}, \quad i = \overline{1, m}, j = \overline{1, n} \quad (1.26)$$

Здесь функции $F_1(x^k, y^k, u^k, v^k)$, $F_0(x^q, y^q)$, непрерывны по u и непрерывно дифференцируемы по x .

Для решения поставленной задачи в работе будут применены метод быстрого автоматического дифференцирования и генетический алгоритм.

§2. Применение методологии БАД для построения оптимального управления в обобщенной популяционной модели хищник-жертва

2.1. Описание метода БАД для оптимизации систем, описываемых интегро-дифференциальными уравнениями

Для рекуррентных соотношений (1.23)-(1.24) введем следующие обозначения:

$$x^l = F_1(l, X^{1l}, Y^{1l}, U^{1l}, V^{1l}) \quad (1.27)$$

$$y^l = F_2(l, X^{2l}, Y^{2l}, U^{2l}, V^{2l}) \quad (1.28)$$

где $X^{il}, Y^{il}, U^{il}, V^{il}$ - заданные наборы векторов x^l, y^l, u^l, v^l соответственно, $i = \overline{1, 2}$, l принимает значения от 1 до q , в общем виде будем писать $l \in D$.

Будем говорить, что соотношения (1.27)-(1.28) задают q -шаговый процесс. Формулы (1.27)-(1.28) определяют вектора x^l, y^l на l -ом шаге.

Для каждого $j \in D$ введем индексные наборы $Q_{1j}, R_{1j}, S_{1j}, W_{1j}, Q_{2j}, R_{2j}, S_{2j}, W_{2j}$, содержащие индексы всех векторов x^l, y^l и $u^l, v^l, l \in D$, принадлежащих, соответственно, наборам X^{1j}, Y^{1j} и U^{1j}, V^{1j} в (1.27):

$$Q_{1j} = \{i \in D : x^i \in X^{1j}\}, S_{1j} = \{i \in D : u^i \in U^{1j}\}$$

$$R_{1j} = \{i \in D : y^i \in Y^{1j}\}, W_{1j} = \{i \in D : v^i \in V^{1j}\}$$

и X^{2j}, Y^{2j} и U^{2j}, V^{2j} в (1.28):

$$Q_{2j} = \{i \in D : x^i \in X^{2j}\}, S_{2j} = \{i \in D : u^i \in U^{2j}\}$$

$$R_{2j} = \{i \in D : y^i \in Y^j\}, W_{2j} = \{i \in D : v^i \in V^j\}$$

Введем сопряженные индексные наборы:

$$\overline{Q}_{1j} = \{i \in D : x^j \in X^i\}, \overline{S}_{1j} = \{i \in D : u^j \in U^i\}$$

$$\overline{R}_{1j} = \{i \in D : y^i \in Y^j\}, \overline{W}_{1j} = \{i \in D : v^i \in V^j\}$$

$$\overline{Q}_{2j} = \{i \in D : y^j \in Y^i\}, \overline{S}_{2j} = \{i \in D : v^j \in V^i\}$$

$$\overline{R}_{2j} = \{i \in D : y^i \in Y^j\}, \overline{W}_{2j} = \{i \in D : v^i \in V^j\}.$$

Между элементами введенных множеств есть связь. Если $i \in Q_{1j}, s \in S_{1j}$, то $j \in \overline{Q}_{1i}, j \in \overline{S}_{1s}$. С помощью введенных множеств можно записать $X^{1j} = \{x^i : i \in Q_{1j}\}, U^{1j} = \{u^i : i \in S_{1j}\}$.

Из приведенных определений следует, что если $x^q \in \overline{X}^{1j}, y^q \in \overline{Y}^{1j}$, то имеет место следующие функциональные зависимости:

$x^q = F_1(q, \dots, x^j, \dots)$, $y^q = F_2(q, \dots, y^j, \dots)$. Следовательно, наборы $Q_{1j}, R_{1j}, S_{1j}, W_{1j}, Q_{2j}, R_{2j}, S_{2j}, W_{2j}$ могут быть названы наборами индексов входных переменных x, y и u, v соответственно, в то время как $\overline{Q}_{1j}, \overline{R}_{1j}, \overline{S}_{1j}, \overline{W}_{1j}, \overline{Q}_{2j}, \overline{R}_{2j}, \overline{S}_{2j}, \overline{W}_{2j}$ называются наборами индексов выходных векторов x, y и u, v соответственно.

Определим матрицы

$$\begin{aligned} N_{1ij} &= \partial x^{jT} / \partial u^i, N_{2ij} = \partial y^{jT} / \partial u^i \\ M_{1ij} &= \partial x^{jT} / \partial v^i, M_{2ij} = \partial y^{jT} / \partial v^i, \end{aligned} \quad (1.29)$$

где

$$\begin{aligned} N_{1ij} &= \delta_{1ij} F_{1u^i}^T(j, X^j, Y^j, U^j, V^j) + \sum_{k \in Q_{1j}} N_{1ik} F_{1x^k}^T(j, X^j, Y^j, U^j, V^j) \\ &+ \sum_{k \in R_{1j}} N_{2ik} F_{1y^k}^T(j, X^j, Y^j, U^j, V^j), \end{aligned}$$

$$\begin{aligned}
M_{2ij} &= \delta_{2ij} F_{2vi}^T(j, X^j, Y^j, U^j, V^j) + \sum_{k \in R_{2j}} M_{2ik} F_{2yk}^T(j, X^j, Y^j, U^j, V^j) \\
&\quad + \sum_{k \in Q_{2j}} M_{1ik} F_{2xk}^T(j, X^j, Y^j, U^j, V^j), \\
\delta_{1ij} &= \begin{cases} 1, i \in S_{1j} \\ 0, i \notin S_{1j} \end{cases}, \delta_{2ij} = \begin{cases} 1, i \in S_{2j} \\ 0, i \notin S_{2j} \end{cases}
\end{aligned}$$

$$\begin{aligned}
M_{1ij} &= \sum_{k \in Q_{1j}} M_{1ik} F_{1xk}^T(j, X^j, Y^j, U^j, V^j) + \\
&\quad + \sum_{k \in R_{1j}} M_{2ik} F_{1yk}^T(j, X^j, Y^j, U^j, V^j), \\
N_{2ij} &= \sum_{k \in R_{2j}} N_{2ik} F_{2yk}^T(j, X^j, Y^j, U^j, V^j) + \\
&\quad + \sum_{k \in Q_{2j}} N_{1ik} F_{2xk}^T(j, X^j, Y^j, U^j, V^j),
\end{aligned}$$

Для процесса (1.27)-(1.28) формула (1.29) запишется следующим образом:

$$\begin{aligned}
N_{1i+1j} &= \frac{\partial x^{i+1}}{\partial u^j} = -\Delta t \delta_{i+1j} + \frac{\partial x^i}{\partial u^j} (1 + \Delta t(e - ax^i) - \\
&\quad - \Delta tax^i - \Delta tb^T y^i) + \frac{\partial y^i}{\partial u^j} (-\Delta tb^T x^i) \\
M_{1i+1j} &= \frac{\partial x^{i+1}}{\partial v^j} = \frac{\partial x^i}{\partial v^j} (1 + \Delta t(e - ax^i) - \Delta tax^i - \\
&\quad - \Delta tb^T y^i) + \frac{\partial y^i}{\partial v^j} (-\Delta tb^T x^i) \\
M_{2i+1j} &= \frac{\partial y^{i+1}}{\partial v^j} = -\Delta t \delta_{i+1j} + \frac{\partial y^i}{\partial v^j} (1 + \Delta t(-\alpha - c^T y^i + d^T(x^i - R)) - \Delta tc^T y^i + \\
&\quad + \Delta t^2 \gamma^T \sum_{s=i-v}^{i-1} F^{i-s}(x^s - R)) + \frac{\partial x^i}{\partial v^j} (\Delta ty^i d^T) + \Delta t^2 \sum_{s=i+1}^{q-1} \frac{\partial x^s}{\partial v^j} y^s \gamma^T F^{s-i} \\
N_{2i+1j} &= \frac{\partial y^{i+1}}{\partial u^j} = \frac{\partial y^i}{\partial u^j} (1 + \Delta t(-\alpha - c^T y^i + d^T(x^i - R)) + \Delta ty^i c^T + \\
&\quad + \Delta t^2 \gamma^T \sum_{s=i-v}^{i-1} F^{i-s}(x^s - R)) + \frac{\partial x^i}{\partial u^j} (\Delta ty^i d^T) + \Delta t^2 \sum_{s=i+1}^{q-1} \frac{\partial x^s}{\partial u^j} y^s \gamma^T F^{s-i}
\end{aligned}$$

$$\delta_{i+1j} = \begin{cases} 1, i+1=j \\ 0, i+1 \neq j \end{cases}$$

Градиент целевой функции по $u^l, l = \overline{0, q-1}$ для процесса (1.27)-(1.28) запишется следующим образом:

$$\begin{aligned} \frac{d\Omega(u, v)}{du^l} = & I_{u^l}(x, y, u, v) + \sum_{k=1}^q N_{1lk} I_{x^k}(x, y, u, v) + \\ & + \sum_{k=1}^q N_{2lk} I_{y^k}(x, y, u, v) \end{aligned} \quad (1.30)$$

Градиент целевой функции по $v^l, l = \overline{0, q-1}$ для процесса (1.27)-(1.28) запишется следующим образом:

$$\begin{aligned} \frac{d\Omega(u, v)}{dv^l} = & I_{v^l}(x, y, u, v) + \sum_{k=1}^q M_{2lk} I_{y^k}(x, y, u, v) + \\ & + \sum_{k=1}^q M_{1lk} I_{x^k}(x, y, u, v) \end{aligned} \quad (1.31)$$

Основное вычислительное время при расчете градиента по формулам (1.30)-(1.31) затрачивается на нахождение матриц $N_{1ij}, N_{2ij}, M_{1ij}, M_{2ij}$, для этого необходимо решить $2q(n+m)$ раз систему уравнений с $q^2(n+m)^2$ неизвестными.

Введем множители Лагранжа p^l, r^l , тогда функция Лагранжа запишется в следующем виде

$$\begin{aligned} L(x, y, u, v, p, r) = & I(x, y, u, v) + \sum_{l=1}^q [F_1^T(l, X^l, Y^l, U^l, V^l) - x^{lT}] p^l \\ & + \sum_{l=1}^q [F_2^T(l, X^l, Y^l, U^l, V^l) - y^{lT}] r^l \end{aligned}$$

Множители Лагранжа определяются из решения системы уравнений:

$$L_{x^i}(x, y, u, v, p, r) = I_{x^i}(x, y, u, v) + \sum_{l \in Q_{1i}} F_{1x^i}^T(l, X^l, Y^l, U^l, V^l) p^l + \sum_{l \in Q_{2i}} F_{2x^i}^T(l, X^l, Y^l, U^l, V^l) r^l - p^i = 0$$

$$L_{y^i}(x, y, u, v, p, r) = I_{y^i}(x, y, u, v) + \sum_{l \in R_{1i}} F_{1y^i}^T(l, X^l, Y^l, U^l, V^l) p^l + \sum_{l \in R_{2i}} F_{2y^i}^T(l, X^l, Y^l, U^l, V^l) r^l - r^i = 0$$

Тогда множители Лагранжа $p_i^k, r_j^k, i = \overline{1, m}, j = \overline{1, n}, k = \overline{0, q}$ для процесса (1.27)-(1.28) запишутся следующим образом:

$$p_i^k = I_{x_i^k}(x, y, u, v) + p_i^{k+1} + \Delta t p_i^{k+1} \left(e_i - \sum_{l=1}^m a_{il} x_l^k \right) - \Delta t \sum_{l=1}^m p_l^{k+1} a_{li} x_l^k - \Delta t p_i^{k+1} \sum_{j=1}^n b_{ji} y_j^k + \Delta t \sum_{j=1}^n r_j^{k+1} y_j^k d_{ji} + \Delta t^2 \sum_{s=k+1}^{k+v} \sum_{j=1}^n r_j^{s+1} y_j^s \gamma_{ji} F_{ji}^{s-k}, i = \overline{1, m} \quad (1.32)$$

$$r_j^k = I_{y_j^k}(x, y, u, v) + r_j^{k+1} - \Delta t r_j^{k+1} \alpha_j - \Delta t r_j^{k+1} \sum_{l=1}^n c_{jl} y_l^k - \Delta t \sum_{l=1}^m p_l^{k+1} b_{lj} (x_l^k - R_l) - \Delta t \sum_{l=1}^n r_l^{k+1} y_l^k c_{lj} + \Delta t r_j^{k+1} \sum_{l=1}^m d_{jl} (x_l^k - R_l) + \Delta t^2 r_j^{k+1} \sum_{l=1}^m \gamma_{jl} \sum_{s=k-v}^{k-1} F_{jl}^{k-s} (x_l^s - R_l), j = \overline{1, n} \quad (1.33)$$

Градиент целевой функции по $u_i^k, i = \overline{1, m}, k = \overline{0, q-1}$ для процесса (1.27)-(1.28) запишется следующим образом:

$$\frac{d\Omega(u, v)}{du_i^k} = L_{u_i^k}(x, y, u, v, p, r) = I_{u_i^k}(x, y, u, v) + \Delta t p_i^{k+1} \quad (1.34)$$

Градиент целевой функции по $v_j^k, j = \overline{1, n}, k = \overline{0, q-1}$ для процесса (1.27)-(1.28) запишется следующим образом:

$$\frac{d\Omega(u, v)}{dv_j^k} = L_{v_j^k}(x, y, u, v, p, r) = I_{v_j^k}(x, y, u, v) + \Delta t r_j^{k+1} \quad (1.35)$$

Основное вычислительное время при расчете градиента по формулам (1.34)-(1.38) тратится на нахождение векторов p^l и r^l , $l = \overline{0, q}$, для этого необходимо решить 2 системы уравнений с $q(n + m)$ неизвестными.

Выражения для вычисления градиента «прямым» и «обратным» методами математически эквивалентны, но с вычислительной точки зрения имеется существенно отличие. Показано, что для расчета градиента по формулам «прямого» метода дифференцирования необходимо решить $2q(n + m)$ раз систему уравнений с $q^2(n + m)^2$ неизвестными. Для расчета градиента по формулам БАД необходимо решить 2 системы уравнений с $q(n + m)$ неизвестными.

Для нахождения градиента по формулам прямого метода дифференцирования и метода БАД необходимо вычислить:

для прямого метода дифференцирования:	для метода БАД:
$q^2 m^2$ значений $\partial x^{jT} / \partial u^i$, $q^2 mn$ значений $\partial y^{jT} / \partial u^i$, $q^2 mn$ значений $\partial x^{jT} / \partial v^i$, $q^2 n^2$ значений $\partial y^{jT} / \partial v^i$, qm значений $\frac{dI(u)}{du^i}$, qn значений $\frac{dI(u)}{dv^i}$,	qm значений p^i , qn значений r^i , qm значений $\frac{dI(u)}{du^i}$, qn значений $\frac{dI(u)}{dv^i}$,
всего $q^2(n + m)^2 + q(n + m)$ значений	всего $2q(n + m)$ значений

Таким образом, применение прямого метода вычисления градиента целевой функции в $q(n + m)$ раз более трудоемко, чем расчеты по формулам «прямого» метода дифференцирования.

В случае если решение, полученное на основе использования формул (1.32)-(1.35) не позволяет достичь требуемой точности нахождения решения, а также в

случае «овражного» вида минимизируемой функции, может возникнуть необходимость применения численных методов оптимизации второго порядка.

По аналогичной схеме введем новые множители Лагранжа $\tilde{p}^l, \tilde{\tilde{p}}^l, \tilde{r}^l, \tilde{\tilde{r}}^l$, тогда функция Лагранжа $\bar{L}_i^k$ запишется в следующем виде

$$\begin{aligned}
\bar{L}_i^k = & L_{u_i^k} + \sum_{k=0}^{q-1} \sum_{i=1}^m \tilde{p}_i^{k+1} (x_i^{k+1} - x_i^k - \Delta t x_i^k (l_i - \sum_{l=1}^m a_{il} x_l^k)) + \Delta t \sum_{j=1}^n b_{ji} y_j^k (x_i^k - R_i) + \Delta t u_i^k + \\
& + \sum_{k=0}^{q-1} \sum_{j=1}^n r_j^{k+1} (y_j^{k+1} - y_j^k - \Delta t y_j^k (-d_j - \sum_{l=1}^n l_{jl} y_l^k + \sum_{l=1}^m d_{jl} (x_l^k - R_l))) - \\
& - \Delta t^2 y_i^k \sum_{l=1}^m \gamma_{jl} \sum_{s=k-v}^{k-1} F_{jl}^{k-s} (x_l^s - R_l) + \Delta t v_i^k + \\
& + \sum_{k=0}^{q-1} \sum_{i=1}^m \tilde{\tilde{p}}_i^{k+1} (p_i^k - p_i^{k+1} - \Delta t p_i^{k+1} (l_i - \sum_{l=1}^m a_{il} x_l^k)) + \sum_{l=1}^m \Delta t p_l^{k+1} x_l^k a_{li} + \\
& + \Delta t \sum_{j=1}^n b_{ji} y_j^k p_i^{k+1} - \Delta t \sum_{j=1}^n y_j^k r_j^{k+1} d_{ji} - \Delta t^2 \sum_{j=1}^n r_j^{k+1} y_j^k \gamma_{ji} \sum_{s=k+1}^{k+v} F_{ji}^{s-k} + \\
& + \sum_{k=0}^{q-1} \sum_{j=1}^n \tilde{\tilde{r}}_j^{k+1} (r_j^k - r_j^{k+1} + \Delta t r_j^{k+1} \alpha_j + \Delta t r_j^{k+1} \sum_{l=1}^n l_{jl} y_l^k + \Delta t \sum_{l=1}^m p_l^{k+1} b_{jl} (x_l^k - R_l) + \\
& + \Delta t \sum_{l=1}^n r_l^{k+1} y_l^k c_{lj} - \Delta t r_j^{k+1} \sum_{l=1}^m d_{jl} (x_l^k - R_l) - \Delta t^2 r_j^{k+1} \sum_{l=1}^m \gamma_{jl} \sum_{s=k-v}^{k-1} F_{jl}^{k-s} (x_l^s - R_l))
\end{aligned}$$

где производные функции $\bar{L}_i^k$ по множителям p^l, r^l будут иметь следующий вид:

$$\begin{aligned}
\frac{\partial \bar{L}_i^k}{\partial p_i^s} = & \frac{\partial L_{u_i^k}}{\partial p_i^s} + \tilde{\tilde{p}}_i^{s+1} - \tilde{\tilde{p}}_i^s - \Delta t \tilde{\tilde{p}}_i^s (e_i - \sum_{j=1}^m a_{ij} x_j^{s-1}) + \sum_{j=1}^m \Delta t \tilde{\tilde{p}}_j^s x_i^{s-1} a_{ij} + \\
& + \Delta t \tilde{\tilde{p}}_i^s \sum_{j=1}^n b_{jl} y_j^{s-1} + \Delta t \sum_{j=1}^n \tilde{\tilde{r}}_j^s b_{jl} (x_l^{s-1} - R_l) = 0 \\
\frac{\partial \bar{L}_i^k}{\partial r_i^s} = & \frac{\partial L_{u_i^k}}{\partial r_i^s} + \tilde{\tilde{r}}_i^{s+1} - \tilde{\tilde{r}}_i^s + \Delta t \tilde{\tilde{r}}_i^s \alpha_j + \Delta t \tilde{\tilde{r}}_i^s \sum_{j=1}^n c_{lj} y_j^{s-1} + \Delta t \sum_{j=1}^n \tilde{\tilde{r}}_j^s y_j^{s-1} c_{jl} -
\end{aligned}$$

$$\begin{aligned}
& -\Delta t \tilde{r}_l^s \sum_{j=1}^m d_{lj} (x_j^{s-1} - R_j) - \Delta t^2 \tilde{r}_l^s \sum_{j=1}^m \gamma_{lj} \sum_{z=s-v-1}^{s-2} F_{lj}^{s-z-1} (x_j^z - R_j) - \\
& - \Delta t \sum_{j=1}^m \tilde{p}_j^s y_l^{s-1} d_{lj} - \Delta t^2 \sum_{j=1}^m \tilde{p}_j^s y_l^{s-1} \gamma_{lj} \sum_{z=s}^{s+v+1} F_{lj}^{z-s-1} = 0.
\end{aligned}$$

Производные функции $\bar{L}_i^k$ по x_i^k, y_j^k определяются из следующих формул:

$$\begin{aligned}
\frac{\partial \bar{L}_i^k}{\partial x_i^k} &= \tilde{p}_i^k - \tilde{p}_i^{k+1} - \Delta t \tilde{p}_i^{k+1} (e_i - \sum_{l=1}^m a_{il} x_l^k) + \sum_{l=1}^m \Delta t \tilde{p}_i^{k+1} x_l^k a_{li} + \\
&+ \Delta t \sum_{j=1}^n b_{ji} y_j^k \tilde{p}_i^{k+1} - \Delta t \sum_{j=1}^n y_j^k r_j^{k+1} d_{ji} - \Delta t^2 \sum_{j=1}^n \tilde{r}_j^{k+1} y_j^k \gamma_{ji} \sum_{s=k+1}^{k+v} F_{ji}^{s-k} + \\
&+ \sum_{l=1}^m \Delta t \tilde{p}_{il}^{k+1} p_{il}^{k+1} a_{li} + \Delta t \sum_{l=1}^m \tilde{p}_l^{k+1} p_i^{k+1} a_{il} + \sum_{j=1}^n \tilde{r}_j^{k+1} p_i^{k+1} b_{ji} - \Delta t \sum_{j=1}^n \tilde{r}_j^{k+1} r_j^{k+1} d_{ji}
\end{aligned}$$

Тогда градиент функции $\bar{L}_i^k$ по управлению u_l^k, v_l^k будет иметь следующий

вид:

$$\begin{aligned}
\frac{\partial \bar{L}_i^k}{\partial u_l^s} &= \frac{\partial^2 L}{\partial u_i^k \partial u_l^s} + \Delta t \tilde{p}_l^{s+1} = \frac{\partial^2 L}{\partial u_i^k \partial u_l^s} + \Delta t p_i^{k+1} + \Delta t \tilde{p}_l^{s+1} \\
\frac{\partial \bar{L}_i^k}{\partial v_l^s} &= \frac{\partial^2 L}{\partial u_i^k \partial v_l^s} + \Delta t \tilde{r}_l^{s+1} = \frac{\partial^2 L}{\partial u_i^k \partial v_l^s} + \Delta t p_i^{k+1} + \Delta t \tilde{r}_l^{s+1}
\end{aligned}$$

2.2. Практическая реализация численного алгоритма на основе метода

БАД

Алгоритм построения приближенного оптимального решения для задачи (1.22)-(1.26) будет следующим:

1. Зададим произвольный набор матриц $U^{(0)} = \{u_i^k\}^{(0)}, V^{(0)} = \{v_j^k\}^{(0)}$, элементы которых удовлетворяют условию (1.26).

2. По рекуррентным соотношениям (1.23)-(1.24), с использованием начальных значений (1.25) и наборов $U^{(0)}, V^{(0)}$ вычислим $x_i^{k+1}, y_j^{k+1}, i = \overline{1, m}, j = \overline{1, n}, k = \overline{0, q-1}$. В результате получим набор векторов $x^1, \dots, x^q, y^1, \dots, y^q$, соответствующий выбранным $U^{(0)}, V^{(0)}$, который обозначим $(x^k)^{(0)}, (y^k)^{(0)}$.

3. Вычислим значение функции (1.22), используя $(x^k)^{(0)}, (y^k)^{(0)}$ и $U^{(0)}, V^{(0)}$, и обозначим эту величину $I^{(0)}$. Здесь верхний индекс в скобках соответствует номеру итерации.

4. Вычислим сопряженные векторы по формулам (1.22) и (1.23), начиная с индекса q и заканчивая индексом 1.

5. Вычислим значения $u_i^k, v_j^k, i = \overline{1, m}, j = \overline{1, n}, k = \overline{0, q-1}$ соответствующие первой итерации: $U^{(1)} = \{u_i^k\}^{(1)}, V^{(1)} = \{v_j^k\}^{(1)}$, по формулам

$$(u_i^k)^{(l+1)} = (u_i^k)^{(l)} - \alpha^{(i)} \left(\frac{dL(u, v)}{du_i^k} \right)^{(l)},$$

$$(v_j^k)^{(l+1)} = (v_j^k)^{(l)} - \alpha^{(j)} \left(\frac{dL(u, v)}{dv_j^k} \right)^{(l)},$$

где $\alpha^{(i)} > 0$ - величина шага градиентного спуска.

Если $(u_i^k)^{(l+1)} < 0$, то $(u_i^k)^{(l+1)} = 0$; если $(u_i^k)^{(l+1)} > u_{i \max}$, то $(u_i^k)^{(l+1)} = u_{i \max}$.

Если $(v_j^k)^{(l+1)} < 0$, то $(v_j^k)^{(l+1)} = 0$; если $(v_j^k)^{(l+1)} > v_{j \max}$, то $(v_j^k)^{(l+1)} = v_{j \max}$.

6. С полученными значениями управления переходим ко второму шагу, используя вместо матриц $U^{(0)}, V^{(0)}$ матрицы $U^{(1)}, V^{(1)}$, полученные в результате первой итерации, строим $(x^k)^{(1)}, (y^k)^{(1)}$ по формулам (1.23), (1.24). Найдем

значение минимизируемой функции (1.22), используя $(x^k)^{(1)}$, $(y^k)^{(1)}$, $U^{(1)}$, $V^{(1)}$, и обозначим эту величину $I^{(1)}$. Вычислим приращение $\Delta I^{(1)} = I^{(0)} - I^{(1)}$.

7. Если $\Delta I^{(1)} > 0$, то заменим на втором шаге $U^{(0)}$, $V^{(0)}$ на $U^{(1)}$, $V^{(1)}$, если $\Delta I^{(1)} \leq 0$, то уменьшим шаг градиентного спуска в два раза. Вычисления продолжим до тех пор, пока $\Delta I^{(1)} < \varepsilon$, где $\varepsilon > 0$ – заданная точность.

В качестве критерия остановки в работе рассмотрены следующие методы:

1. $I^{(k+1)} < I^{(k)}$, $|I^{(k)} - I^{(k+1)}| < \varepsilon$

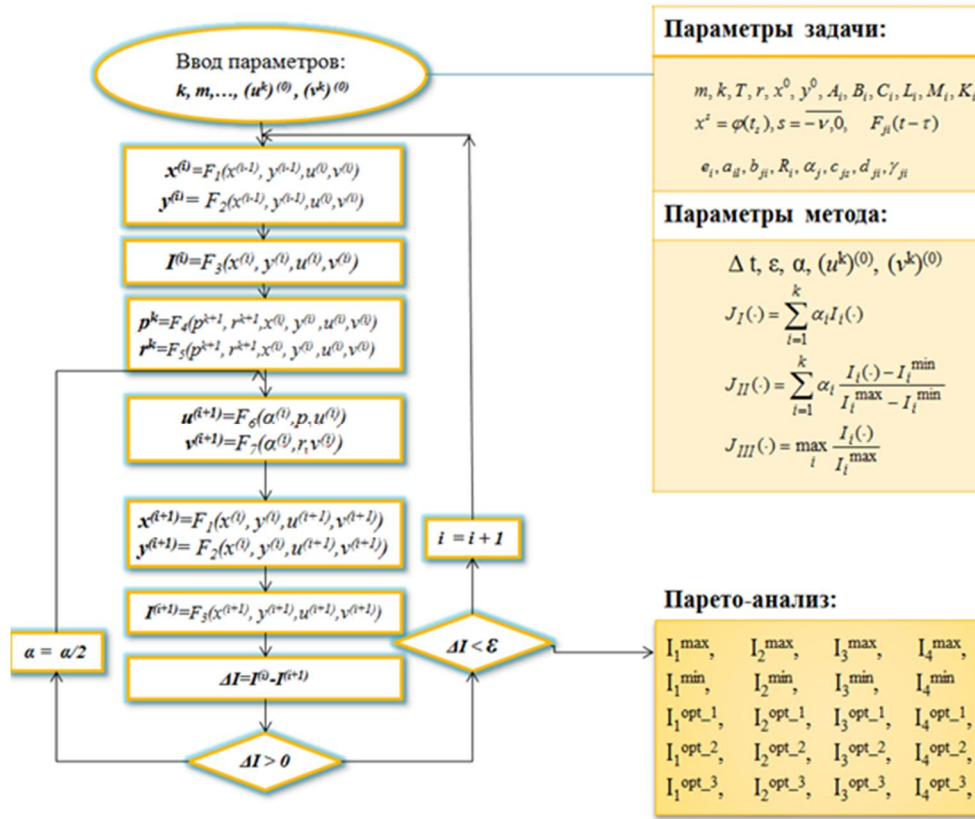
2. $\|z^{(k)} - z^{(k+1)}\| < \varepsilon$,

$$\sqrt{\sum_{l=0}^q \left(\sum_{i=1}^m (x_i^{l(k)} - x_i^{l(k+1)})^2 + \sum_{j=1}^n (y_j^{l(k)} - y_j^{l(k+1)})^2 \right)} < \varepsilon$$

3. $\|w^{(k)} - w^{(k+1)}\| < \varepsilon$,

$$\sqrt{\sum_{l=0}^{q-1} \left(\sum_{i=1}^m (u_i^{l(k)} - u_i^{l(k+1)})^2 + \sum_{j=1}^n (v_j^{l(k)} - v_j^{l(k+1)})^2 \right)} < \varepsilon$$

2.3. Блок-схема численного алгоритма оптимизации на основе метода БАД



2.4. Анализ влияния параметров задачи на оптимальное решение системы

Метод отработывался на примере взаимодействия двух популяции при следующих параметрах задачи:

$$\begin{aligned}
 J(u) = & -\int_0^T (d_1 x_1(t) - e_1 u_1(t)) + (\tilde{d}_1 y_1(t) - \tilde{e}_1 v_1(t)) dt + \\
 & + M_1 \cdot ((A_1 - x_1(T), 0)_+^2) + \tilde{M}_1 \cdot ((\tilde{A}_1 - y_1(T), 0)_+^2) \\
 & + N_1 \int_0^T ((C_1 - x_1(t), 0)_+ dt + \tilde{N}_1 \int_0^T ((\tilde{C}_1 - y_1(t), 0)_+ dt
 \end{aligned}$$

$$\begin{aligned}
 T=10, m=1, n=1, x_1(0)=6, y_1(0)=6, R_i=0, e_i=0.75, c_{11}=0, d_{11}=0, \alpha_1=0.75, \\
 \gamma_{11}=0.1, a_{11}=0.0375, b_{11}=0.0375, A_1=7.5, \tilde{A}_1=13, C_1=6.5, \tilde{C}_1=5,
 \end{aligned}$$

$d_1 = 1, \tilde{d}_1 = 0.01, \quad e_1 = \tilde{e}_1 = 0.01, \quad M_1 = \tilde{M}_1 = N_1 = \tilde{N}_1 = 100, \quad 0 \leq u_1 \leq 1, \quad 0 \leq v_1 \leq 1,$
 $\Delta t = 0.01, \alpha = 0.1,$ точность метода $\varepsilon = 0,0000001$.

На рисунке (1.2) представлены графики оптимального решения системы и соответствующего ему оптимального управления при заданных параметрах и виде минимизируемого функционала:

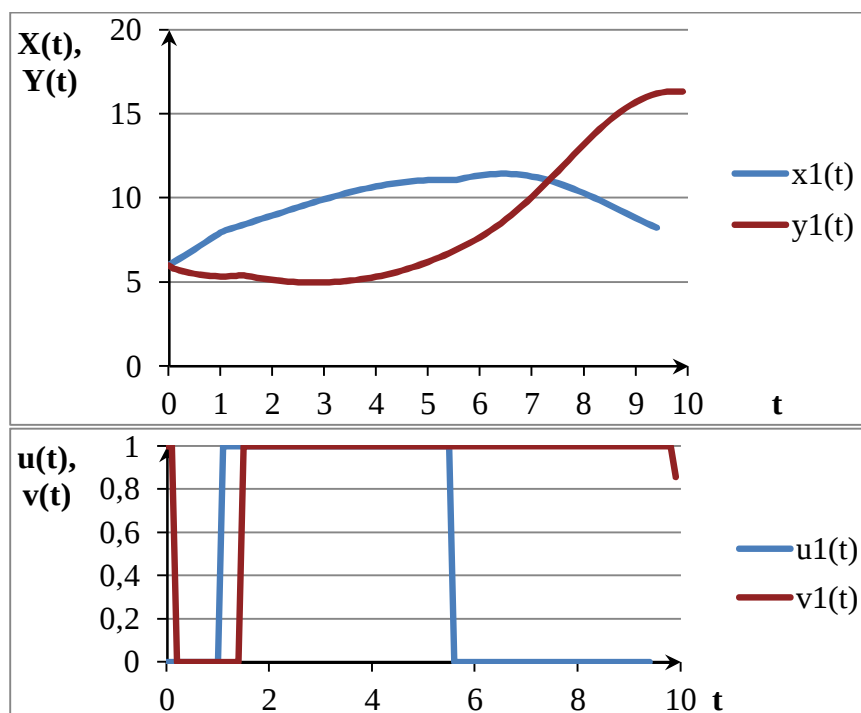


Рис. 1.2 Графики численности популяций хищников $y_1(t)$ и жертв $x_1(t)$ на заданном отрезке времени $t \in [0, 10]$ и оптимальные скорости отлова хищников $v_1(t)$ и жертв $u_1(t)$.

На рисунках (1.3)-(1.5) представлены соответствующие полученному оптимальному решению график зависимости значения минимизируемого функционала от числа итераций и графики зависимости нормы изменения вектора состояния и нормы изменения вектора управления от номера итерации:

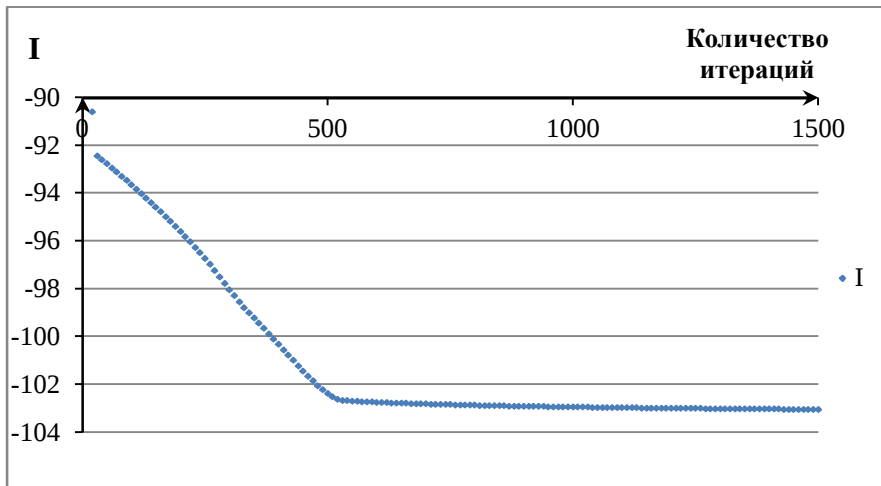


Рис. 1.3 График изменения значения минимизируемого функционала в зависимости от номера итерации

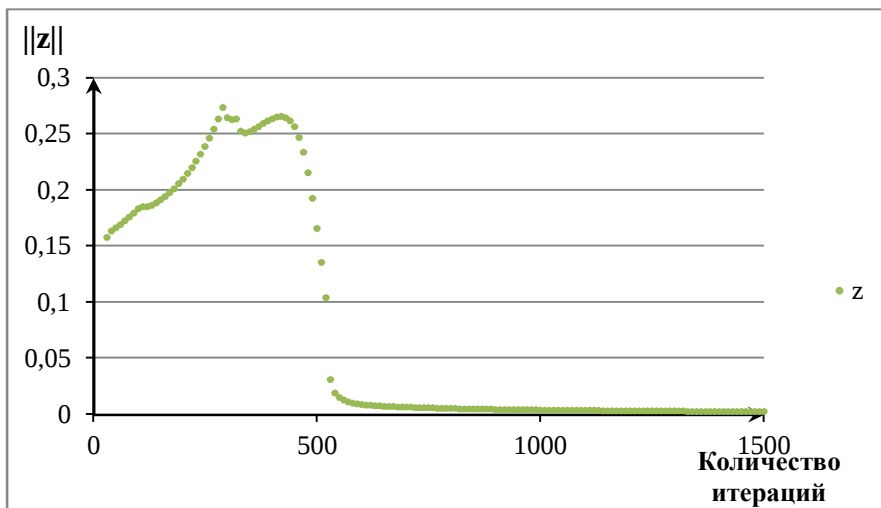


Рис. 1.4 График изменения нормы вектора состояния в зависимости от числа итераций

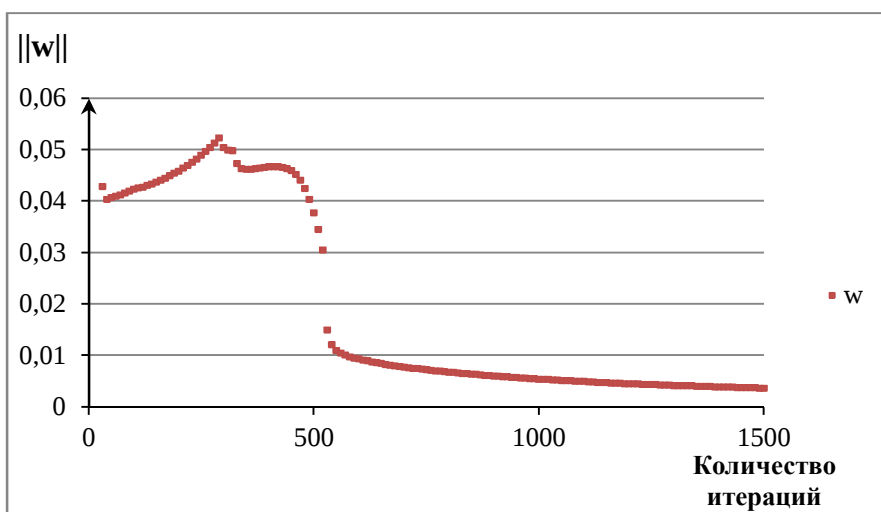


Рис. 1.5 График изменения нормы вектора управления в зависимости от числа итераций

Проведем анализ влияния параметров метода на оптимальное решение. Параметрами метода являются точность метода ε , шаг градиентного спуска α , начальное управление $u^{(0)}, v^{(0)}$.

В работе построено оптимальное управление при различном значении шага градиентного спуска α . На рисунке (1.6) представлен соответствующий график зависимости значения минимизируемого функционала от шага градиентного спуска:

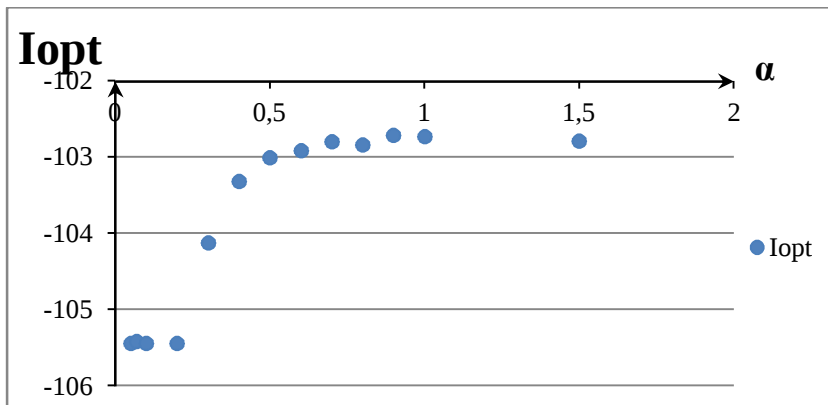


Рис. 1.6. График зависимости значения минимизируемого функционала от шага градиентного спуска

По графику видно, что при увеличении значения шага градиентного спуска значение минимизируемого функционала увеличивается.

В работе построено оптимальное решение при различном стартовом наборе управлений. На рисунке (1.7) представлен соответствующий график зависимости минимизируемого функционала от стартового управления:

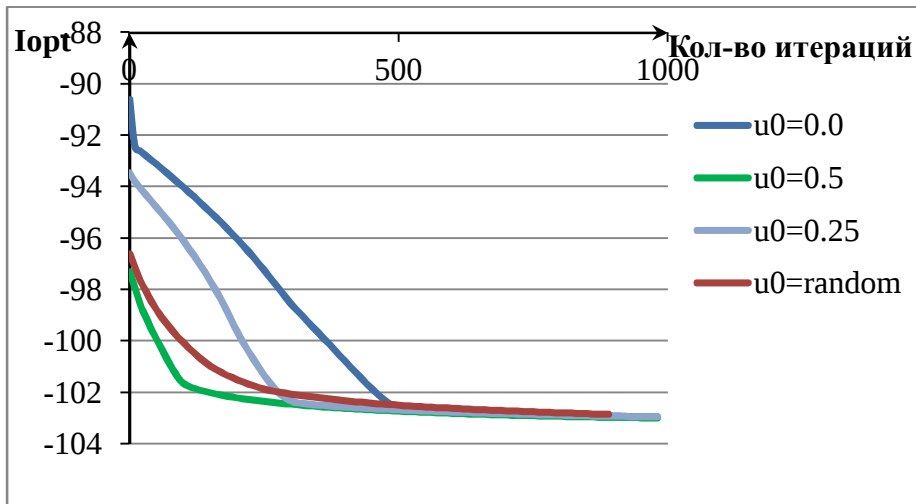


Рис. 1.7. График зависимости значения минимизируемой функции от числа итераций при различных стартовых значениях управления:

По графику 1.7 видно, что стартовое управление влияет на скорость работы алгоритма и на значение минимизируемой функции. «Удачное» стартовое управление позволяет сократить количество итераций алгоритма и получить более точное оптимальное решение. Однако стартовое управление не оказывает значительного влияния на структуру решения, так норма разности векторов функций состояния при различном стартовом управлении не превышает 0,01% от длины вектора состояния.

Существенное влияние на работу алгоритма оказывает точность метода ε .

Ниже в таблицах 1.1 – 1.3 представлены зависимости значения минимизируемого функционала и количества итераций от точности метода ε при различных критериях остановки алгоритма, таких как:

$$|I^{(k)} - I^{(k+1)}| < \varepsilon_1, \quad (1.36)$$

$$\sqrt{\sum_{l=0}^q \left(\sum_{i=1}^m (x_i^{l(k)} - x_i^{l(k+1)})^2 + \sum_{j=1}^n (y_j^{l(k)} - y_j^{l(k+1)})^2 \right)} < \varepsilon_2, \quad (1.37)$$

$$\sqrt{\sum_{l=0}^{q-1} \left(\sum_{i=1}^m (u_i^{l(k)} - u_i^{l(k+1)})^2 + \sum_{j=1}^n (v_j^{l(k)} - v_j^{l(k+1)})^2 \right)} < \varepsilon_3. \quad (1.38)$$

ε_1	1,00E-04	1,00E-05	1,00E-06	1,00E-07	1,00E-08	1,00E-09	1,00E-10
I_{opt}	-103,079	-103,196	-103,22	-103,226	-103,2276	-103,2278	-103,2278
Кол-во итераций	1720	5144	12553	28784	66608	94248	94248

Табл. 1.1 Значения минимизируемого функционала и количества итераций от точности метода ε_1 , определяемой условием (1.36).

ε_2	0,01	0,001	0,0001	0,00001	0,000001	0,0000001
I_{opt}	-102,7367	-103,1565	-103,2208	-103,2269	-103,228	-103,228
Кол-во итераций	574	3072	12553	38555	94248	94248

Табл. 1.2 Значения минимизируемого функционала и количества итераций от точности метода ε , определяемой условием (1.37).

ε_3	0,01	0,001	0,0001	0,00001	0,000001	0,0000001
I_{opt}	-102,726	-103,197	-103,227	-103,2269	-103,228	-103,228
Кол-во итераций	563	5205	31192	38555	94248	94248

Табл. 1.3 Значения минимизируемого функционала и количества итераций от точности метода ε , определяемой условием (1.38).

Из таблицы 1.1 следует, что с увеличением на порядок точности метода, экспоненциально увеличивается число итерации алгоритма, показано, что точность метода составляет 10^{-9} , дальнейшее увеличение точности не приводит к улучшению результата. В случае если необходим более точный результат, можно воспользоваться методами второго порядка, для этого вторые производные также вычислены в работе.

Рассмотрим влияние запаздывания на решение задачи (1.22)-(1.26). Сначала проведем анализ влияния запаздывания на динамику неуправляемой системы.

На рисунке 1.8 представлены графики зависимости численности популяций при различных значениях запаздывания r при следующих параметрах системы:

$$T=200, \quad m=1, \quad n=1, \quad R_i = 0, \quad e_i = 0.75, \quad c_{11} = 0, \quad d_{1i} = 0, \quad \alpha_1 = 0.75, \quad \gamma_{11} = 0.1, \\ a_{11} = 0.0375, \quad b_{11} = 0.0375, \quad \Delta t = 0.01,$$

$$F_{jl}^k(t, \tau) = \begin{cases} Dc, & \tau \in [t-r, t] \\ 0, & \tau \notin [t-r, t] \end{cases}$$

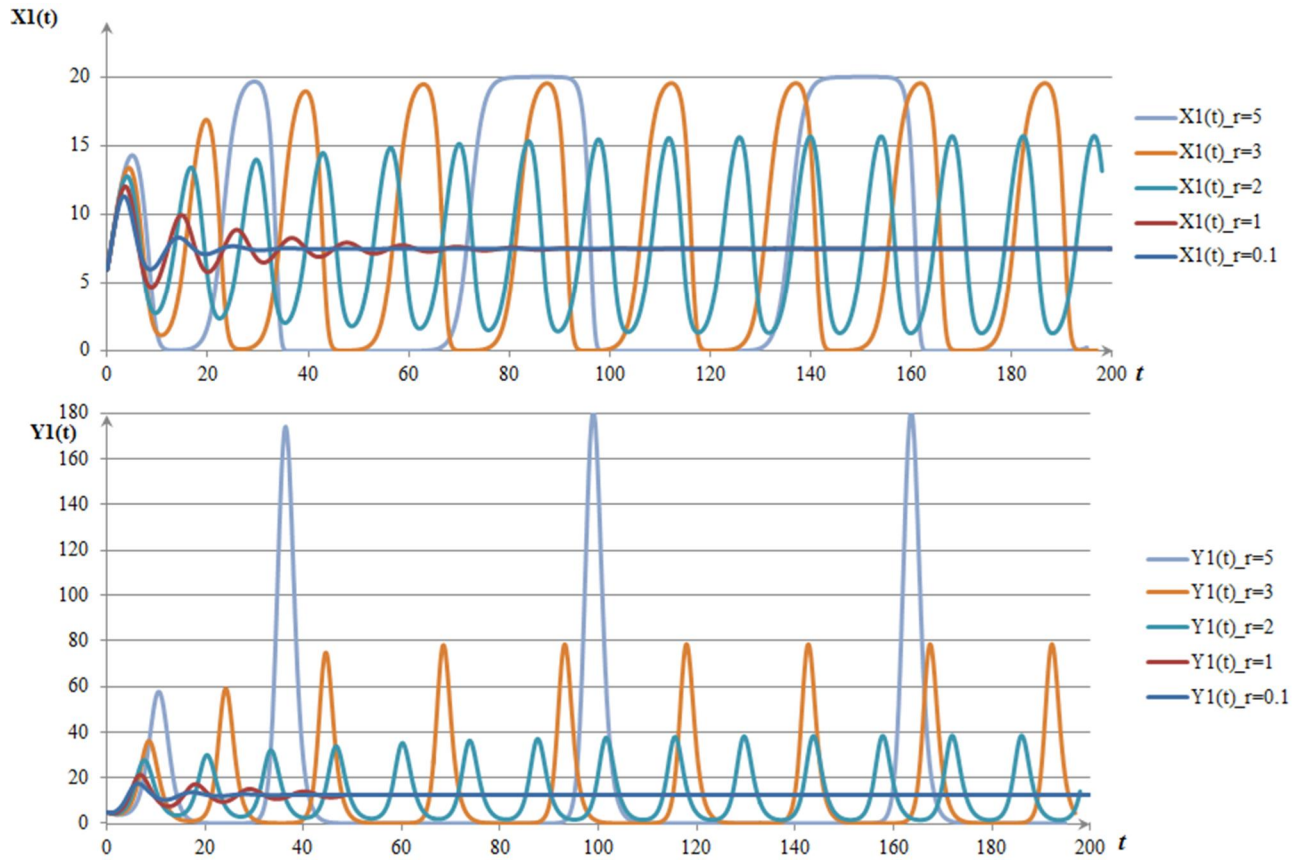


Рис. 1.8 Графики зависимости численности популяций при различных значениях запаздывания r .

По графикам видно, что при малых значениях запаздывания, системы переходят в состояние равновесия, которое для модели без запаздывания определяется из условия $\bar{x}_1 = \frac{\alpha_1}{\gamma_{11}}, \bar{y}_1 = \frac{\gamma_{11}e_1 - a_{11}\alpha_1}{\gamma_{11}b_{11}}$. В частности, при определённых выше параметрах задачи, $\bar{x}_1 = 7,5, \bar{y}_1 = 12,5$. Но при увеличении запаздывания, решение становится периодическим. Периоды колебаний хищников и жертв совпадают, при этом при увеличении запаздывания амплитуда колебаний численности жертв увеличивается до максимально возможного значения $\frac{e_1}{a_{11}} = 20$, а амплитуда колебания хищников с ростом запаздывания увеличивается экспоненциально.

На рисунках (1.9) – (1.11) представлены графики зависимости численности популяций при различных значениях запаздывания r для случая двух популяций жертв и одной популяции хищников при следующих параметрах системы:

$$T=200, \quad m=2, \quad n=1, \quad R_i=0, \quad e_i=0.75, \quad c_{11}=0, \quad d_{1i}=0.1, \quad \alpha_1=0.75, \quad \gamma_{i1}=0.75, \\ a_{11}=0.0375, \quad a_{12}=0.03, \quad a_{21}=0.035, \quad a_{22}=0.037, \quad b_{11}=0.0375, \quad b_{21}=0.037 \\ \Delta t=0.01, \quad F_{jl}^k(t, \tau) = \begin{cases} Dc, & \tau \in [t-r, t] \\ 0, & \tau \notin [t-r, t] \end{cases}$$

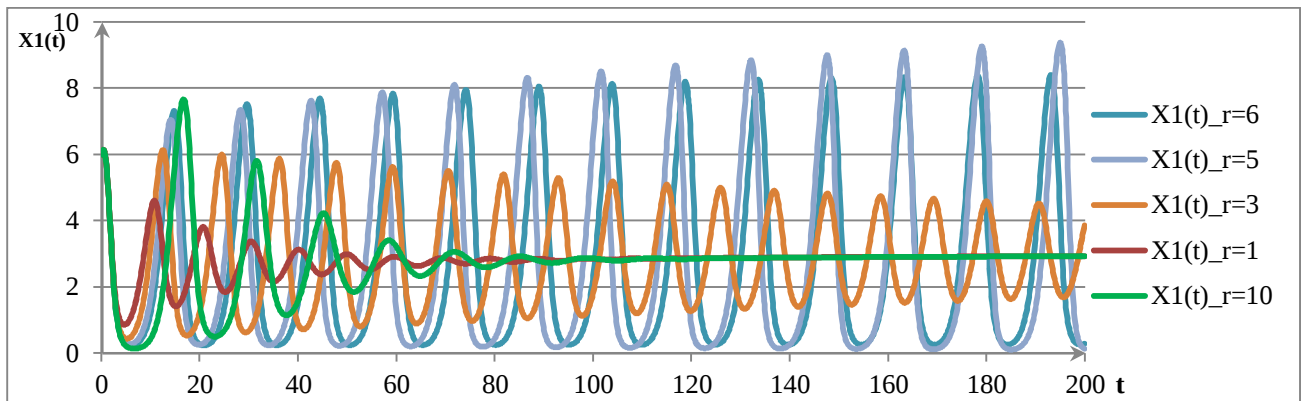


Рис. 1.9. График зависимости $\bar{x}_1(t)$ при различных значениях запаздывания r

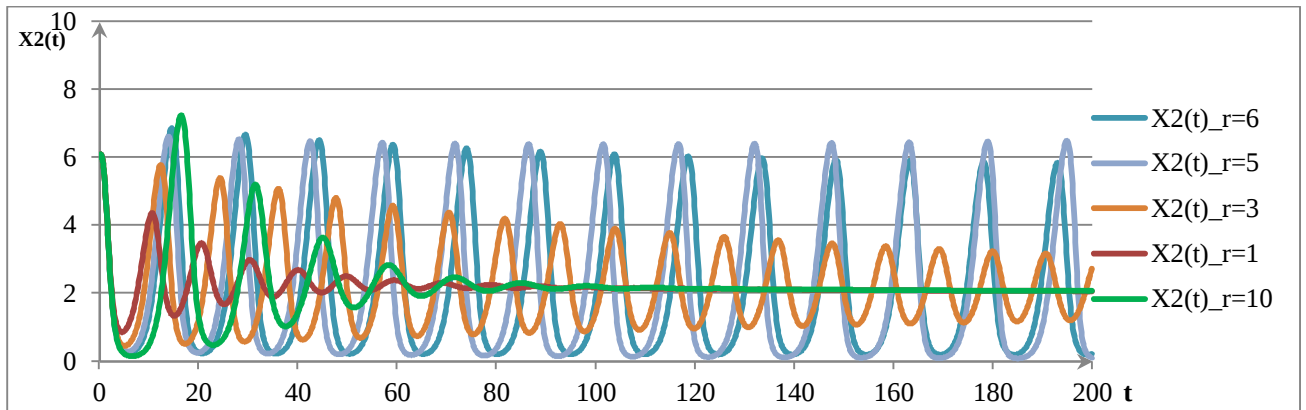


Рис. 1.10. График зависимости $\bar{x}_2(t)$ при различных значениях запаздывания r

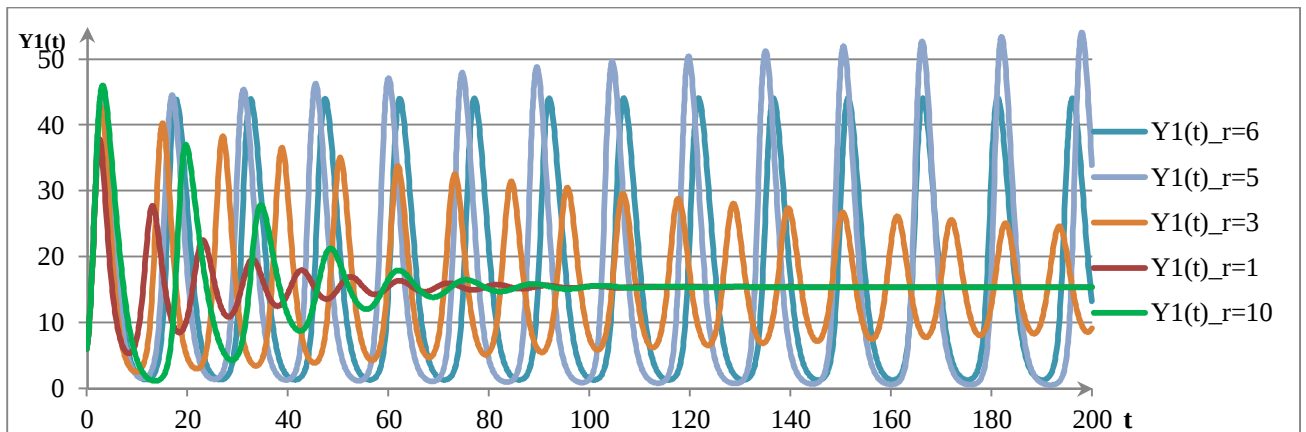


Рис. 1.11. График зависимости $\bar{y}_1(t)$ при различных значениях запаздывания r

По графикам функций $x_1(t)$, $x_2(t)$, $y_1(t)$, представленных на рисунках (1.9)–(1.11), видно, что при малых значениях запаздывания, системы переходят в состояние равновесия, при увеличении запаздывания (до $r = 5$), решение становится периодическим. Периоды колебаний хищников и жертв совпадают, при этом при увеличении запаздывания амплитуда колебаний численности жертв и хищников увеличивается, частота колебаний уменьшается. При $r > 5$ амплитуда колебаний уменьшается, и при дальнейшем увеличении запаздывания система переходит в состояние равновесия.

На основе метода быстрого автоматического дифференцирования построено оптимальное решение задачи (1.22)–(1.26). Ниже на рисунках (1.12)–(1.17) представлены графики зависимости оптимального управления $\bar{u}_1(t)$, $\bar{u}_2(t)$, $\bar{v}_1(t)$ и численности популяции $\bar{x}_1(t)$, $\bar{x}_2(t)$, $\bar{y}_1(t)$ при различных значениях запаздывания r при следующих параметрах системы:

$T=25$, $m = 2$, $n = 1$, $J(u, v) = M \cdot \max(A_1 - x_1(T), 0)^2 + N \cdot \max(B_1 - y_1(T), 0)^2$, $R_i = 0$,
 $e_i = 0.75$, $c_{11} = 0$, $d_{1i} = 0.1$, $\alpha_1 = 0.75$, $\gamma_{i1} = 0.05$, $i = 1, 2$, $a_{11} = 0.038$, $a_{12} = 0.03$,
 $a_{21} = 0.035$, $a_{22} = 0.038$, $b_{11} = 0.038$, $b_{21} = 0.037$, $A_1 = 4$, $A_2 = 3$, $0 \leq u_i \leq 0.1$, $i = 1, 2$,
 $\Delta t = 0.01$, $\alpha = 0.1$, точность метода $\varepsilon = 0,0000001$:

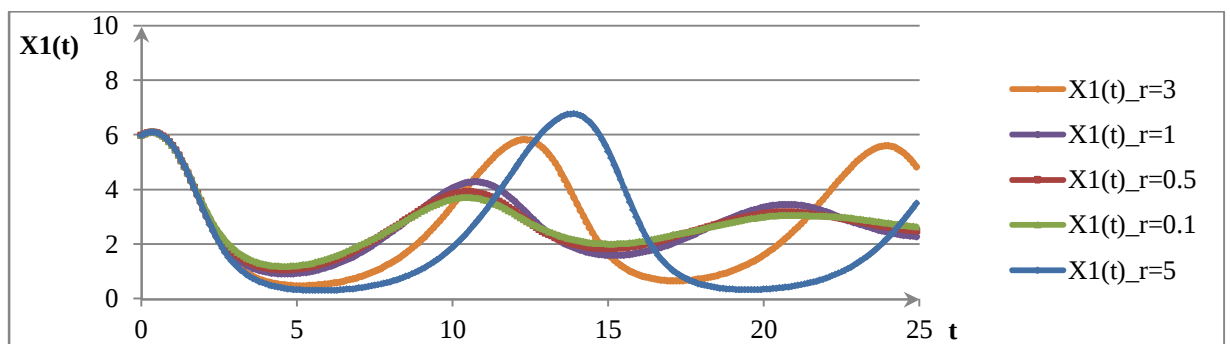


Рис. 1.12. График зависимости $\bar{x}_1(t)$ при различных значениях запаздывания r

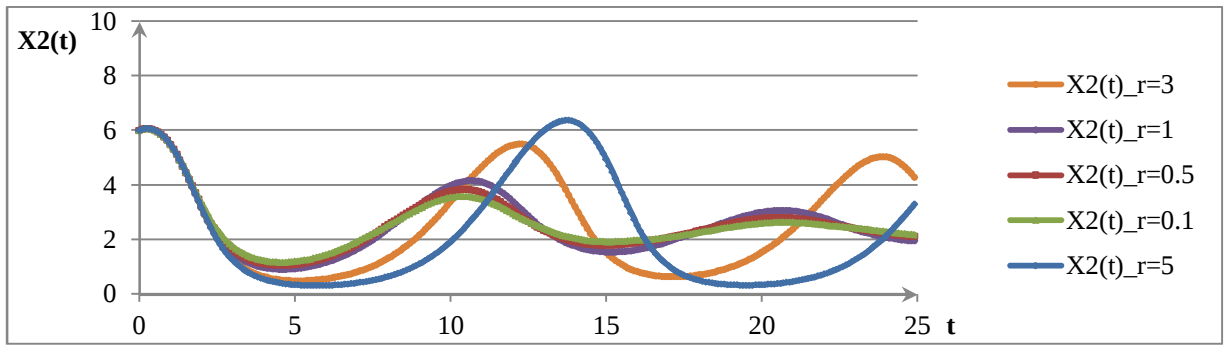


Рис. 1.13. График зависимости $\bar{x}_2(t)$ при различных значениях запаздывания r

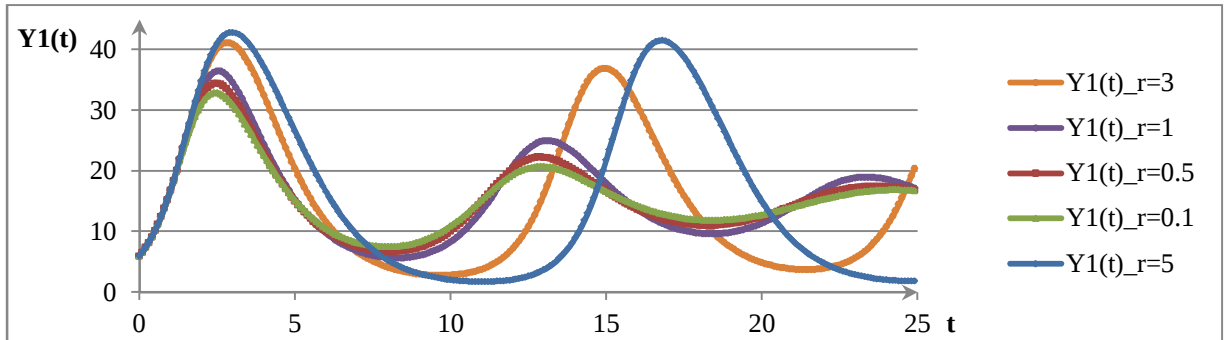


Рис. 1.14. График зависимости $\bar{y}_1(t)$ при различных значениях запаздывания r

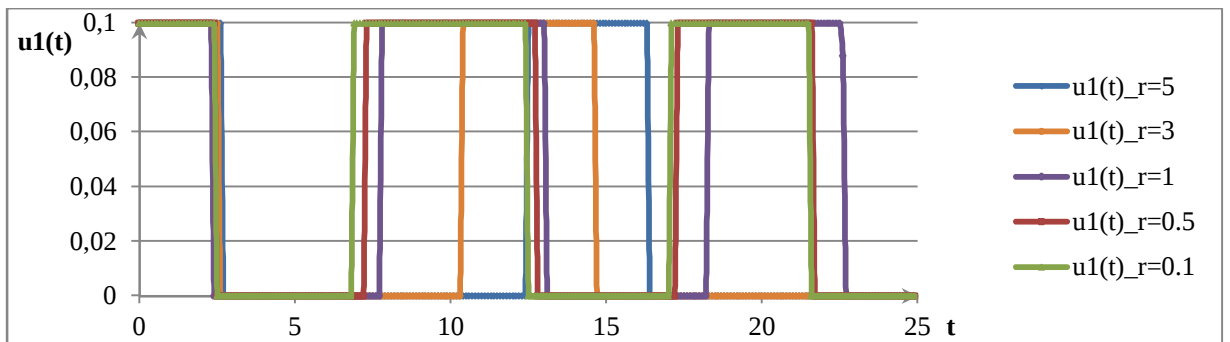


Рис. 1.15. График зависимости $\bar{u}_1(t)$ при различных значениях запаздывания r

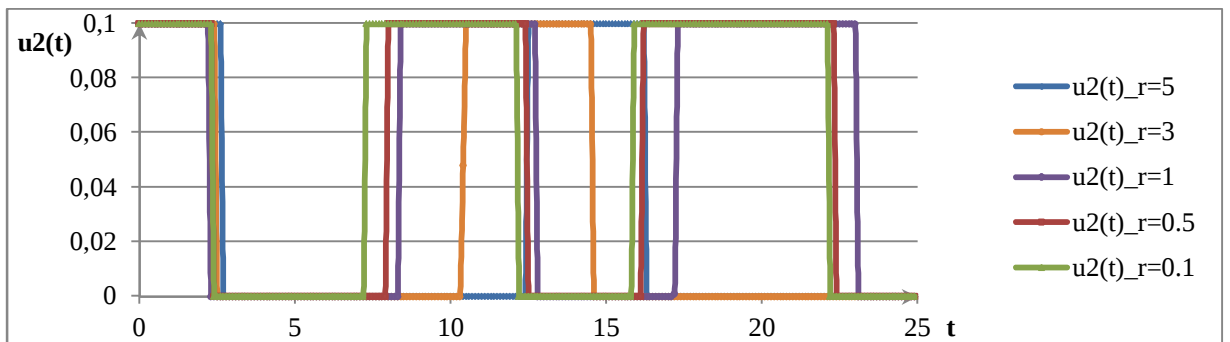


Рис. 1.16. График зависимости $\bar{u}_2(t)$ при различных значениях запаздывания r

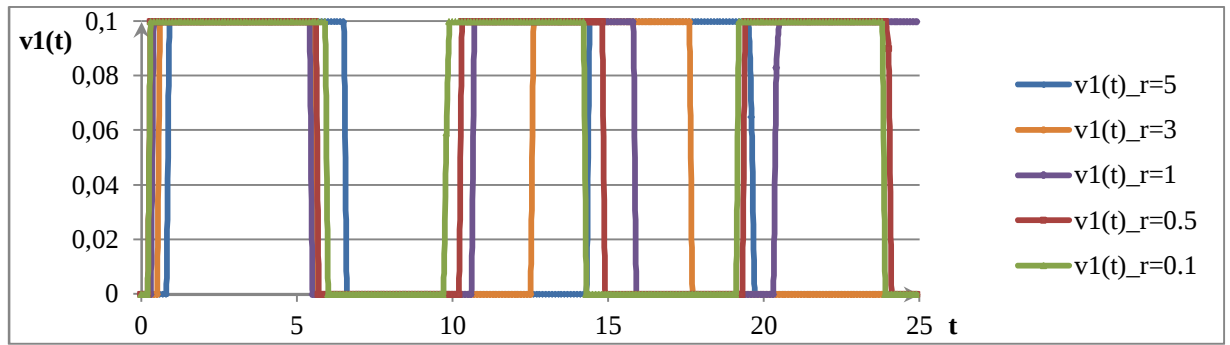


Рис. 1.17. График зависимости $\bar{v}_1(t)$ при различных значениях запаздывания r

Легко видеть, что полученное численными методами оптимальное управление удовлетворяет принципу максимума.

В работе построено оптимальное решение для различных типов подынтегральной функции. На рисунках (1.18)-(1.19) представлены графики численности популяции $\bar{x}_i(t)$ для различных типов функций $G(t, \tau)$:

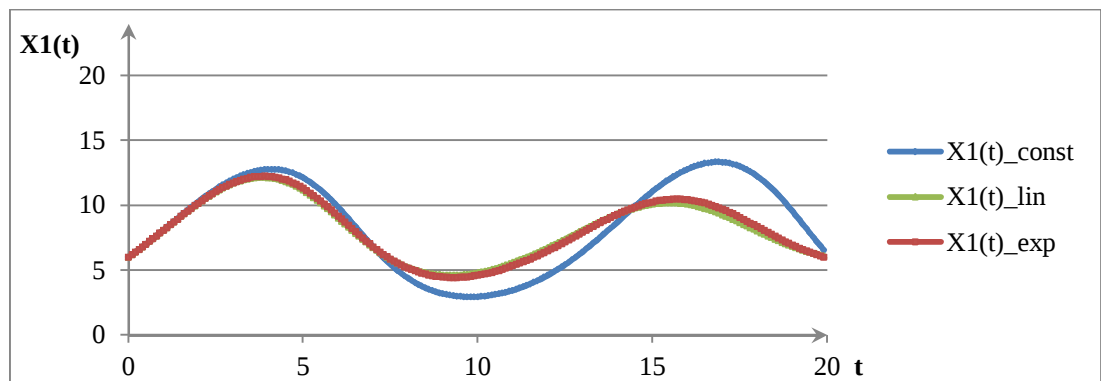


Рис. 1.18. График зависимости $\bar{x}_1(t)$ для различных подынтегральных функций

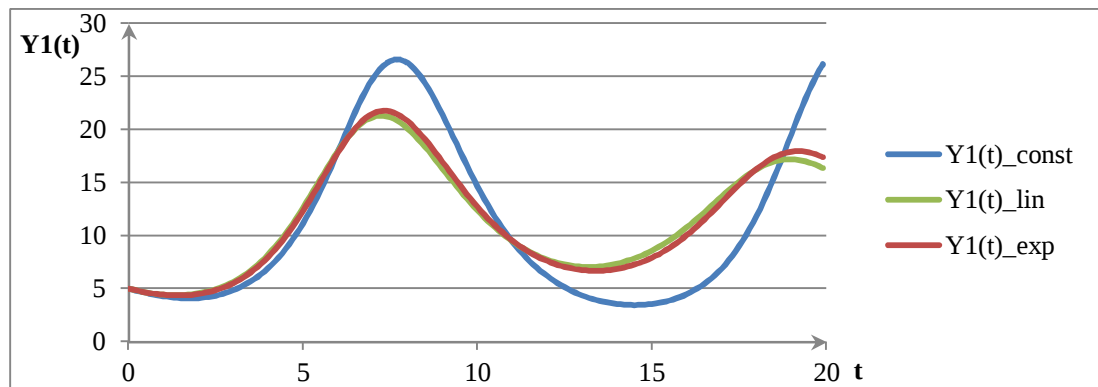


Рис. 1.19. График зависимости $\bar{x}_2(t)$ для различных подынтегральных функций

где

$X1(t)_{opt_const}$ соответствует численности популяции, для которой функция распределения определяется по формуле:

$$G(t, \tau) = \begin{cases} c, & \tau \in [t-r, t] \\ 0, & \tau \notin [t-r, t] \end{cases}$$

$X1(t)_{opt_lin}$ соответствует численности популяции, для которой функция распределения определяется по формуле:

$$G(t, \tau) = \begin{cases} c \cdot (\tau - t + r), & \tau \in [t-r, t] \\ 0, & \tau \notin [t-r, t] \end{cases}$$

$X1(t)_{opt_exp}$ соответствует численности популяции, для которой функция распределения определяется по формуле:

$$G(t, \tau) = \begin{cases} c \cdot \exp(\tau - t), & \tau \in [t-r, t] \\ 0, & \tau \notin [t-r, t] \end{cases}$$

где $c = const$.

На рисунках (1.20)-(1.21) представлены соответствующие графики оптимального управления для различных типов функций $G(t, \tau)$:

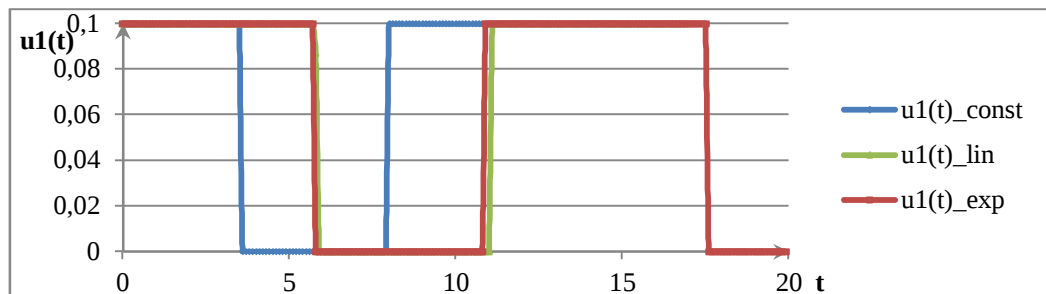


Рис. 1.20. График зависимости $\bar{u}_1(t)$ для различных подынтегральных функций.

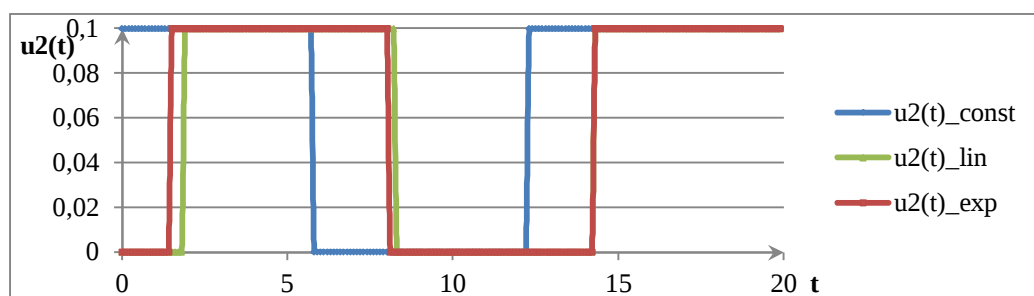


Рис. 1.21. График зависимости $\bar{u}_2(t)$ для различных подынтегральных функций

Легко видеть, что полученное численными методами оптимальное управление удовлетворяет принципу максимума, сформулированному в предыдущей главе. Оптимальное управление определяется функцией переключения и имеет конечное число переключений на фиксированном интервале времени.

§3. Применение генетического алгоритма для построения оптимального решения системы, описываемой интегро-дифференциальными уравнениями

3.1. Описание генетического алгоритма

Генетический алгоритм - это алгоритм, который позволяет найти удовлетворительное решение к аналитически неразрешимым или сложнорешаемым проблемам через последовательный подбор и комбинирование искоемых параметров с использованием механизмов, напоминающих биологическую эволюцию.

Генетические алгоритмы применяются для решения следующих задач:

- Оптимизация функций
- Разнообразные задачи на графах (задача коммивояжера, раскраска, нахождение паросочетаний)
- Настройка и обучение искусственной нейронной сети
- Задачи компоновки
- Составление расписаний

- Игровые стратегии
- Аппроксимация функций
- Искусственная жизнь
- Биоинформатика

Подоплекой возникновения этих методов стали несколько открытий в биологии. Сначала Чарльз Дарвин опубликовал в 1859 году свою знаменитую работу "Происхождение видов", где были провозглашены основные принципы эволюционной теории:

- Наследственность (потомки сохраняют свойства родителей)
- Изменчивость (потомки почти всегда не идентичны)
- Естественный отбор (выживают наиболее приспособленные).

Тем самым было показано, какие принципы приводят к эволюционному развитию флоры и фауны под влиянием окружающей среды. Однако вопрос о том, как генетическая информация передается от родителей потомкам, долгое время оставался открытым. В 1944 году Эйвери, Маклеод и Маккарти опубликовали результаты своих исследований, доказывавших, что за наследственные процессы ответственна "кислота дезоксирибозного типа". Однако о том, как работает ДНК, стало известно позднее. В 1953 году в номере журнала "Nature" вышла статья Уотсона и Крика, которые впервые предложили модель двухцепочной спирали ДНК.

Таким образом, стали известны все необходимые компоненты для реализации эволюционной модели.

Первые публикации, которые можно отнести к генетическим алгоритмам, принадлежат Баричелли Н.А. Его работы "Symbiogenetic evolution processes realised by artificial methods" (1957)[60], "Numerical testing of evolution theories" (1960)[62]-[63] были направлены, прежде всего, на понимание природного феномена наследственности. В 1965 году Л.Дж. Фогель, А.Дж. Оуэнс, М.Дж. Уолш предложили и исследовали эволюцию простых автоматов, предсказывающих символы в цифровых последовательностях [65]-[69].

Родителем современной теории генетических алгоритмов считается Д.Х. Холланд. Однако сначала его интересовала, прежде всего, способность природных систем к адаптации, а его мечтой было создание такой системы, которая могла бы приспосабливаться к любым условиям окружающей среды. В 1975 году Холланд публикует свою самую знаменитую работу «Adaptation in Natural and Artificial Systems» [51]. В ней он впервые ввёл термин «генетический алгоритм» и предложил схему классического генетического алгоритма (canonical GA). В дальнейшем понятие «генетические алгоритмы» стало очень широким, и зачастую к ним относятся алгоритмы, сильно отличающиеся от классического ГА.

Ученики Холланда – Кеннет Де Йонг и Дэвид Голдберг – внесли огромный вклад в развитие ГА. Наиболее известная работа Голдберга – «Genetic algorithms in search optimization and machine learning» (1989) [73].

Генетические алгоритмы применяются при разработке программного обеспечения, в системах искусственного интеллекта, оптимизации и в других отраслях знаний. Генетические алгоритмы часто используются совместно с нейронными сетями. Они могут поддерживать нейронные сети или наоборот, либо оба метода взаимодействуют в рамках гибридной системы, предназначенной для решения конкретной задачи.

Генетические алгоритмы основаны на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Они отличаются от традиционных методов оптимизации несколькими базовыми элементами. В частности, генетические алгоритмы:

- 1) обрабатывают не значения параметров самой задачи, а их закодированную форму.
- 2) осуществляют поиск решения исходя не из единственной точки, а из их некоторой популяции;
- 3) используют только целевую санкцию, а не ее производные либо иную дополнительную информацию,

4) применяют вероятностные, а не детерминированные правила выбора.

Перечисленные четыре свойства могут быть также сформулированы как кодирование параметров, операции на популяциях, использование минимума информации о задаче и рандомизация операций.

При описании генетических алгоритмов используются определения, заимствованные из генетики, например, популяции особей, а в качестве базовых понятий применяются ген, хромосома, генотип, фенотип, аллель. Также используются соответствующие этим терминам определения из технического лексикона, в частности, цепь, двоичная последовательность, структура.

Популяция - это конечное множество особей.

Особи, входящие в популяцию, в генетических алгоритмах представляются хромосомами с закодированными в них множествами параметров задачи, т.е. решений, которые иначе называются точками в пространстве поиска (search points). В некоторых работах особи называются организмами.

Хромосомы (другие названия - цепочки или кодовые последовательности) - это упорядоченные последовательности генов.

Ген (также называемый свойством, знаком или детектором) - это атомарный элемент генотипа, в частности, хромосомы.

Генотип или структура - это набор хромосом данной особи. Следовательно, особями популяции могут быть генотипы либо единичные хромосомы (в довольно распространенном случае, когда генотип состоит из одной хромосомы).

Фенотип - это набор значений, соответствующих данному генотипу, т.е. декодированная структура или множество параметров задачи.

Аллель - это значение конкретного гена, также определяемое как значение свойства или вариант свойства.

Локус или позиция указывает место размещения данного гена в хромосоме (цепочке).

Очень важным понятием в генетических алгоритмах считается функция приспособленности (fitness function), иначе называемая функцией оценки. Она

представляет меру приспособленности данной особи в популяции. Эта функция играет важнейшую роль, поскольку позволяет оценить степень приспособленности конкретных особей в популяции и выбрать из них наиболее приспособленные (т.е. имеющие наибольшие значения функции приспособленности) в соответствии с эволюционным принципом выживания «сильнейших» (лучше всего приспособившихся). Функция приспособленности также получила свое название непосредственно из генетики. Она оказывает сильное влияние на функционирование генетических алгоритмов и должна иметь точное и корректное определение. В задачах оптимизации функция приспособленности, как правило, оптимизируется и называется целевой функцией.

При использовании генетического алгоритма задача кодируется таким образом, чтобы её решение могло быть представлено в виде вектора («хромосома»). Случайным образом создаётся некоторое количество начальных векторов («начальная популяция»). Они оцениваются с использованием «функции приспособленности», в результате чего каждому вектору присваивается определённое значение («приспособленность»), которое определяет вероятность выживания организма, представленного данным вектором. После этого с использованием полученных значений приспособленности выбираются вектора (селекция), допущенные к «скрещиванию». К этим векторам применяются «генетические операторы» (в большинстве случаев «скрещивание» — crossover и «мутация» — mutation), создавая, таким образом, следующее «поколение». Особи следующего поколения также оцениваются, затем производится селекция, применяются генетические операторы и т. д. Так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма. Таким критерием может быть:

- нахождение глобального, либо субоптимального решения;
- исчерпание числа поколений, отпущенных на эволюцию;
- исчерпание времени, отпущенного на эволюцию.

Существуют различные методы кодирования хромосом: двоичное, логарифмическое, код Грея. Двоичное кодирование является, наверное, самым популярным способом кодирования чисел. Так, в классической реализации генетического алгоритма применяется именно этот метод. Напомним, что двоичное кодирование основано на известном способе записи десятичных чисел в двоичной системе, где каждый бит двоичного кода соответствует очередной степени цифры 2. В генетических алгоритмах можно также использовать код Грея, который характеризуется тем, что двоичные последовательности, соответствующие двум последовательным целым числам, отличаются только одним битом. Такой способ кодирования хромосом может оказаться оправданным при использовании операции мутации.

Логарифмическое кодирование – применяется в генетических алгоритмах для уменьшения длины хромосом. Оно используется, главным образом, в задачах многомерной оптимизации с большими пространствами поиска решений.

При логарифмическом кодировании первый бит кодовой последовательности – это бит знака показательной санкции, второй бит – бит знака степени этой функции, а остальные биты представляют значение самой степени:

$$[\alpha\beta \text{ BIN}] = (-1)^\beta \cdot e^{(-1)^\alpha \text{ BIN}_{10}}$$

где BIN_{10} означает десятичное значение числа, закодированного в виде двоичной последовательности BIN .

При реализации генетических алгоритмов важным понятием является стратегия селекции. Селекция – это выбор тех хромосом, которые будут участвовать в создании потомков для следующей популяции, т.е. для очередного поколения. Такой выбор производится согласно принципу естественного отбора, по которому наибольшие шансы на участие в создании новых особей имеют хромосомы с наибольшими значениями функции приспособленности. Существуют различные методы селекции. Рассмотрим их подробнее.

Основанный на принципе колеса рулетки метод селекции считается для генетических алгоритмов основным методом отбора особей для родительской популяции с целью последующего их преобразования генетическими операторами, такими как скрещивание и мутация. Несмотря на случайный характер процедуры селекции, родительские особи выбираются пропорционально значениям их функций приспособленности: каждой хромосоме сопоставлен сектор колеса рулетки, величина которого устанавливается пропорциональной значению функции приспособленности данной хромосомы, поэтому, чем больше значение функции приспособленности, тем больше сектор на колесе рулетки. Отсюда вытекает, что чем больше сектор на колесе рулетки, тем выше шанс, что будет выбрана именно эта хромосома. Слабая сторона этого метода заключается в том, что особи с очень малым значением функции приспособленности слишком быстро исключаются из популяции, что может привести к преждевременной сходимости генетического алгоритма. В связи с вышесказанным, созданы и используются альтернативные алгоритмы селекции.

При турнирной селекции все особи популяции разбиваются на подгруппы с последующим выбором в каждой из них особи с наилучшей приспособленностью. Различаются два способа такого выбора: детерминированный выбор и случайный выбор. Детерминированный выбор осуществляется с вероятностью, равной 1, а случайный выбор – с вероятностью, меньшей 1. Подгруппы могут иметь произвольный размер, но чаще всего популяция разделяется на подгруппы по 2-3 особи в каждой. Турнирный метод пригоден для решения задач как максимизации, так и минимизации функции. Помимо того, он может быть легко распространен на задачи, связанные с многокритериальной оптимизацией, т.е. на случай одновременной оптимизации нескольких функций. В турнирном методе допускается изменение размера подгрупп, на которые подразделяется популяция. Исследования подтверждают, что турнирный метод действует эффективнее, чем метод рулетки.

При ранговой селекции особи популяции ранжируются по значениям их функции приспособленности. Это можно представить себе как отсортированный список особей, упорядоченных по направлению от наиболее приспособленных к наименее приспособленным (или наоборот), в котором каждой особи приписывается число, определяющее ее место в списке и называемое рангом. Количество копий каждой особи, введенных в родительскую популяцию, рассчитывается по априорно заданной функции в зависимости от ранга особи. Достоинство рангового метода заключается в возможности его применения как для максимизации, так и для минимизации функции.

Представленные выше методы (рулетки, турнирный и ранговый) применяются чаще всего, но существуют так называемые особые процедуры селекции: элитарная стратегия и генетический алгоритм с частичной заменой популяции.

Элитарная стратегия заключается в защите наилучших хромосом на последующих итерациях. В классическом генетическом алгоритме самые приспособленные особи не всегда переходят в следующее поколение. Это означает, что новая популяция не всегда содержит хромосому с наибольшим значением функции приспособленности из предыдущей популяции. Элитарная стратегия применяется для предотвращения потери такой особи. Эта особь гарантированно включается в новую популяцию.

Генетический алгоритм с частичной заменой популяции, иначе называемый генетическим алгоритмом с зафиксированным состоянием, характеризуется тем, что часть популяции переходит в следующее поколение без каких-либо изменений. Это означает, что входящие в эту часть хромосомы не подвергаются операциям скрещивания и мутации. Часто в конкретных реализациях алгоритма данного типа на каждой итерации заменяются только одна или две особи вместо скрещивания и мутации в масштабе всей популяции.

3.2. Практическая реализация генетического алгоритма

Разработка генетического алгоритма начинается с конструирования двоичной хромосомы, представляющей закодированные возможные решения этой задачи. Решением задачи (1.22)-(1.26) являются $m + n$ векторов длиной q . В соответствии с этим необходимо закодировать $(m + n) \cdot q$ значений управления. Поскольку на управления наложены ограничения (1.26), то, при условии, что управления u_i^l , v_j^l принимают дискретные значения на заданных отрезках $[0, u_{i\max}]$ и $[0, v_{j\max}]$, можно разбить отрезки $[0, u_{i\max}]$ и $[0, v_{j\max}]$ на $2^k - 1$ отрезков, и каждое значение управления закодировать с помощью бинарного кода. Длина хромосомы в этом случае будет составлять $k \cdot (m + n) \cdot q$.

Например, если выбрать $k = 3$, то управления u_i^l будет принимать дискретные значения из множества:

$$\left\{ 0, \frac{u_{i\max}}{8}, \frac{2 \cdot u_{i\max}}{8}, \frac{3 \cdot u_{i\max}}{8}, \frac{4 \cdot u_{i\max}}{8}, \frac{5 \cdot u_{i\max}}{8}, \frac{6 \cdot u_{i\max}}{8}, \frac{7 \cdot u_{i\max}}{8}, u_{i\max} \right\}.$$

Соответственно каждое значение можно закодировать последовательностью из 3 бит, тогда закодированное множество значений управления будет:

$$\{000, 001, 010, 011, 100, 101, 110, 111\}.$$

Длина хромосомы существенно влияет на работы генетического алгоритма, поэтому при большом значении параметра k для решения задачи (1.22)-(1.26) требуются очень большие вычислительные ресурсы. Для сокращения размерности задачи в работе предложено кодировать одним геном несколько значений управлений u_i^l , v_j^l , для этого введен дополнительный параметр алгоритма z_{uv} — степень сжатия решения, который, например, позволяет закодировать значения $u_i^0, u_i^1, \dots, u_i^{z_{uv}-1}$ одним общим значением гена. В результате длина хромосомы будет составлять $\frac{k \cdot (m + n) \cdot q}{z_{uv}}$ бит.

Далее можно выделить следующие этапы генетического алгоритма:

1. Произвольным способом создается P хромосом (где P – размер популяции), т.е. формируются P векторов, состоящих из 0 или 1, длиной $\frac{(m+n) \cdot q}{Z_{uv}}$, которые представляют собой P наборов управления.

$$\begin{array}{c}
 \frac{k \cdot (m+n) \cdot q}{Z_{uv}} \\
 \begin{array}{c}
 P \text{ наборов} \\
 \text{управлений}
 \end{array}
 \left[\begin{array}{ccc}
 00110 & . & . & . & 1000 \\
 10010 & . & . & . & 0011 \\
 \vdots & & & & \vdots \\
 10001 & . & . & . & 1001
 \end{array} \right]
 \end{array}$$

2. Выполняются операции скрещивания и мутации, в результате чего к имеющимся P хромосомам добавляются еще $P_m \cdot P + P_c \cdot P$ новых хромосом, где P_m - доля мутированных особей от общего размера популяции, P_c - доля скрещенных особей от общего размера популяции.

Ниже представлена схема одноточечного скрещивания и одноточечной мутации:

Скрещивание:

0	1	1	0	1	0		0	0	1	1	1	0	1	0	1	1	1	0
1	0	1	1	0	0		1	0	1	0	1	1	0	0	0	1	0	0
0	1	1	0	1	0		1	0	1	0	1	1	0	0	0	1	0	0
1	0	1	1	0	0		0	0	1	1	1	0	1	0	1	1	1	0

Мутация:

0	1	1	1	0	0	0	1	1	1	0	1	0	1	1	0	0	0	0
0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0

Поскольку задача (1.22)-(1.26) является задачей большой размерности, то для ее решения используются операции многоточечного скрещивания и многоточечной мутации. В классическом генетическом алгоритме операция скрещивания и мутации представляет собой одноточечное скрещивание и одноточечную мутацию, однако в задачах большой размерности при использовании этих операций достаточно быстро выделяется один-единственный генотип, который представляет собой локальный минимум, а затем все элементы

популяции проигрывают ему отбор, и вся популяция «забивается» копиями этой особи. По этой причине для решения задачи (1.22)-(1.26) используются операции многоточечного скрещивания и многоточечной мутации, при этом количества точек скрещивания и мутации зависят от длины хромосомы и вычисляются по формулам $H_c \cdot \frac{(m+n) \cdot q}{z_{uv}}$ и $H_m \cdot \frac{(m+n) \cdot q}{z_{uv}}$ соответственно, где H_m - доля мутированных генов от общего размера хромосомы, H_c - степень скрещивания (доля точек скрещивания от общего размера хромосомы).

3. Оценивается степень приспособленности каждой хромосомы с помощью вычисления состояния системы и минимизируемой функции (1.21). Для этого каждая хромосома декодируется в набор управлений, а затем вычисляются значения $x_i^{k+1}, y_j^{k+1}, i = \overline{1, m}, j = \overline{1, n}, k = \overline{0, q-1}$ по рекуррентным соотношениям (1.23)-(1.24), с использованием начальных значений (1.25) и значения $I(u, v)$ по формуле (1.22).

4. Выбираем P особей с наименьшим значением минимизируемой функции $I(u, v)$ (процедура селекции).

Шаги 1 – 4 повторяются G раз, где G – число поколений.

Предлагаемый модифицированный генетический алгоритм был реализован на языке Java. Дополнительно реализован модуль, использующий встроенную многопоточность Java, позволяющий проводить серию вычислений при изменении определенного параметра алгоритма в заданном диапазоне значений. Проведенные численные эксперименты позволили определить оптимальные параметры алгоритма.

3.3. Блок-схема генетического алгоритма



3.4. Анализ влияния параметров генетического алгоритма

Параметрами самого алгоритма являются G - количество поколений, P - размерность популяции, P_m - доля мутированных особей от общего размера популяции, H_m - доля мутированных генов от общего размера хромосомы, P_c - доля скрещенных особей от общего размера популяции, H_c - степень скрещивания (доля точек скрещивания от общего размера хромосомы) и z_{uv} - степень сжатия решения.

Исследуем влияние параметра z_{uv} (степень сжатия решения) на решение задачи.

Далее в таблице 1.1 приведены значения минимизируемой функции в зависимости от степени сжатия решения и количества особей при следующих параметрах задачи:

$$m = 1, n = 1, J(u, v) = M \cdot (A_1 - x_1(T), 0)^2_+ + N \cdot (B_1 - y_1(T), 0)^2_+, T = 10,$$

$$R_1 = 0, e_1 = 0.75, c_{11} = d_{11} = 0, \alpha_1 = 0.75, \gamma_{11} = 0.75, a_{11} = 0.0375, b_{11} = 0.0375,$$

$$A_1 = 7.5, A_2 = 13, u_{\max} = 0.1, v_{\max} = 0.1, \Delta t = 0.01$$

и при следующих параметрах метода:

$$G = 200, P_m = 0.2, H_m = 0.05, P_c = 0.8, H_c = 0.01.$$

$\begin{matrix} P \\ z_{uv} \end{matrix}$	300	400	500	600	700	800	900	1000
2	16,653	16,493	16,349	16,337	16,327	16,337	16,328	16,326
4	16,588	16,367	16,348	16,336	16,329	16,326	16,326	16,326
6	16,537	16,408	16,369	16,38	16,329	16,334	16,343	16,326
8	16,431	16,364	16,326	16,327	16,327	16,326	16,326	16,326
10	16,327	16,326	16,326	16,326	16,326	16,326	16,326	16,326
12	16,329	16,33	16,327	16,331	16,326	16,329	16,329	16,329
14	16,333	16,328	16,328	16,328	16,328	16,328	16,328	16,328
16	16,33	16,33	16,33	16,33	16,33	16,33	16,33	16,33
18	16,33	16,336	16,33	16,328	16,33	16,33	16,33	16,33
20	16,333	16,339	16,328	16,328	16,327	16,328	16,328	16,328

Табл.1.1 Значения минимизируемой функции в зависимости от степени сжатия решения и количества особей

Параметр z_{uv} изменяется в диапазоне от 1 до 20, это означает, что при определенных выше параметрах задачи, длина хромосомы изменяется в диапазоне от 2000 до 100 бит. Численные эксперименты показывают, что при $z_{uv} = 1$ для успешной работы алгоритма требуется задавать численность популяции равным не менее 800 особей, что значительно замедляет работу алгоритма. Увеличение параметра z_{uv} с 1 до 20 позволяет повысить скорость работы алгоритма в 4 раза, но приводит к снижению точности решения. Поэтому для дальнейших численных

экспериментов был выбран параметр $z_{uv}=10$, поскольку это значение сокращает время работы в 2 раза, но позволяет находить оптимальное решение без потерь в точности.

Далее приведены зависимости значения минимизируемой функции от количества особей и количество поколений при следующих параметрах задачи:

$$m = 1, n = 1, J(u, v) = M \cdot (A_1 - x_1(T), 0)^2_+ + N \cdot (B_1 - y_1(T), 0)^2_+, T = 10,$$

$$R_1 = 0, e_1 = 0.75, c_{11} = d_{11} = 0, \alpha_1 = 0.75, \gamma_{11} = 0.75, a_{11} = 0.0375, b_{11} = 0.0375,$$

$$A_1 = 7.5, A_2 = 13, u_{\max} = 0.1, v_{\max} = 0.1,$$

и при следующих параметрах метода:

$$P_m = 0.2, H_m = 0.05, P_c = 0.8, H_c = 0.01, z_{uv} = 10.$$

$\begin{matrix} P \\ G \end{matrix}$	300	400	500	600	700	800	900	1000
100	16,508	16,329	16,328	16,326	16,326	16,326	16,326	16,326
200	16,359	16,332	16,326	16,326	16,326	16,326	16,326	16,326
300	16,326	16,336	16,326	16,326	16,326	16,326	16,326	16,326
400	16,389	16,338	16,326	16,326	16,326	16,326	16,326	16,326
500	16,338	16,328	16,326	16,326	16,326	16,326	16,326	16,326
600	16,382	16,353	16,326	16,326	16,326	16,326	16,326	16,326
700	16,340	16,464	16,326	16,326	16,326	16,326	16,326	16,326
800	16,366	16,327	16,326	16,326	16,326	16,326	16,326	16,326
900	16,326	16,339	16,326	16,326	16,326	16,326	16,326	16,326
1000	16,331	16,380	16,326	16,326	16,326	16,328	16,326	16,326

Табл.1.2 Зависимость значения минимизируемой функции от количества особей и количество поколений

Анализируя данные таблицы 1.2, можно сделать вывод, что при использовании данного алгоритма с определенными выше параметрами целесообразно задавать количество поколений не менее 200. Заметим, что

параметр $z_{uv} = 10$ позволил сократить число особей с 800 до 500, дальнейшее увеличения числа поколений и особей в популяции увеличивает время работы алгоритма, но не приводит к существенному улучшению решения исходной задачи.

Ниже на рисунке показана зависимость скорости сходимости генетического алгоритма от доли скрещенных особей от общего размера популяции:

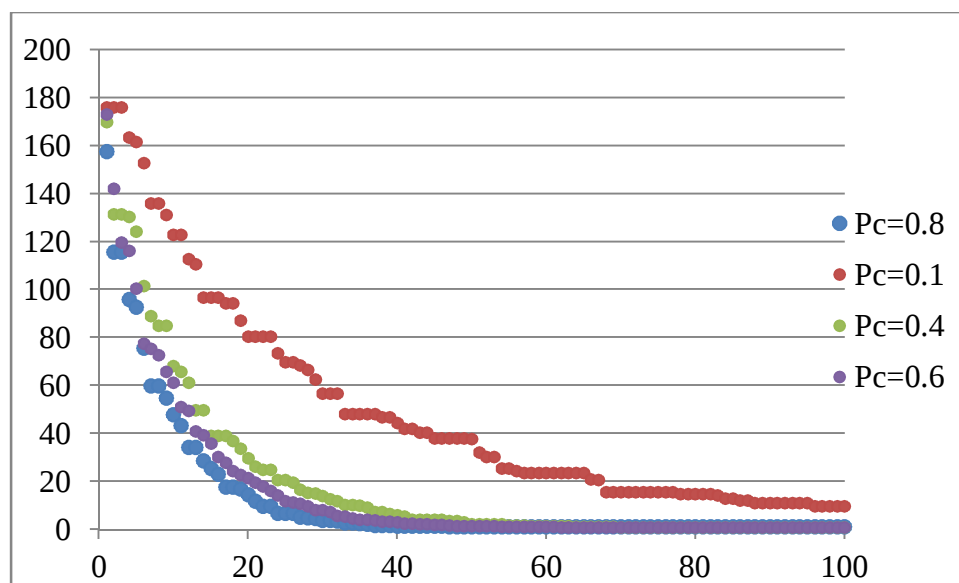


Рис. 1.18. График зависимости значения минимизируемой функции от числа итераций при различных значениях параметра P_c

Чем выше значение P_c , тем быстрее уменьшается значение минимизируемой функции, и приводит к более точному оптимальному значению.

Ниже на рисунках 1.19-1.20 представлены графики зависимости оптимального управления $\bar{u}_1(t)$, $\bar{v}_1(t)$ и функций состояния $\bar{x}_1(t)$, $\bar{y}_1(t)$, найденные с помощью модифицированного генетического алгоритма при тех же параметрах задачи:

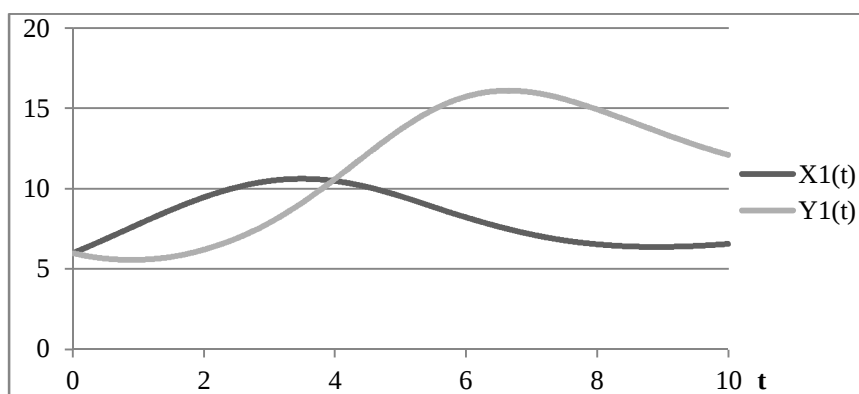


Рис. 1.19. График зависимости функций состояния $\bar{x}_1(t)$, $\bar{y}_1(t)$ от t

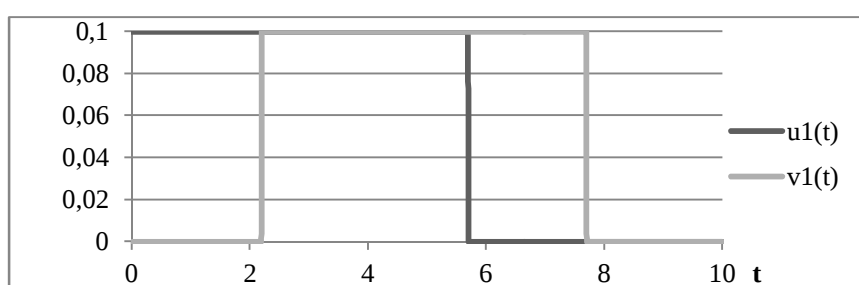


Рис. 1.20. График зависимости оптимального управления состояния $\bar{u}_1(t)$, $\bar{v}_1(t)$ от t

Значение минимизируемой функции при указанных выше параметрах $I(u, v) = 16,32636$, при этом время работы алгоритма составило 35 сек. Таким образом, разработанный алгоритм позволяет находить оптимальное решение задачи (1.22)-(1.26) за приемлемое время.

§4. Сравнение результатов работы метода БАД и генетического алгоритма

Ниже на рисунке (1.21) приведены графики численности популяции хищников и жертв, построенные для неуправляемой модели и полученные методом БАД и генетическим алгоритмом при следующих параметрах задачи:

$$J(u) = J_1(x, y, u, v) + J_2(x, y, u, v) + J_4(x, y, u, v)$$

$$J_1(x, y, u, v) = - \int_0^T \left(\sum_{i=1}^m (d_i x_i(t) - e_i u_i(t)) + \sum_{j=1}^n (\tilde{d}_j y_j(t) - \tilde{e}_j v_j(t)) \right) dt$$

$$J_2(x, y, u, v) = \sum_{i=1}^m N_i \int_0^T ((C_i - x_i(t), 0)_+) dt + \sum_{j=1}^n \tilde{N}_j \int_0^T ((\tilde{C}_j - y_j(t), 0)_+) dt$$

$$J_4(x, y, u, v) = \sum_{i=1}^m M_i \cdot ((A_i - x_i(T), 0)_+^2) + \sum_{j=1}^n \tilde{M}_j \cdot ((\tilde{A}_j - y_j(T), 0)_+^2)$$

$$T = 10, r = 0.5, m = 3, n = 2, x(0) = (6 \ 7 \ 5), y(0) = (5 \ 5), R = (1 \ 0.5 \ 0.5), e_i = 0.75,$$

$$a = \begin{pmatrix} 0,035 & 0,03 & 0,037 \\ 0,035 & 0,037 & 0,0375 \\ 0,03 & 0,0375 & 0,0375 \end{pmatrix}, b = \begin{pmatrix} 0,035 & 0,0375 \\ 0,035 & 0,035 \\ 0,0375 & 0,0375 \end{pmatrix},$$

$$d = \begin{pmatrix} 0,055 & 0,1 & 0,075 \\ 0,1 & 0,05 & 0,1 \end{pmatrix}, \gamma = \begin{pmatrix} 0,1 & 0,1 & 0,1 \\ 0,1 & 0,05 & 0,1 \end{pmatrix}$$

$$c_{ij} = 0, \quad \alpha_j = 0.75, \quad A_i = 7.5, \quad \tilde{A}_j = 13, \quad C_i = 6.5, \quad \tilde{C}_j = 5, \quad d_i = 1, \quad \tilde{d}_j = 0.01,$$

$$e_i = \tilde{e}_j = 0, \quad M_i = \tilde{M}_j = N_i = \tilde{N}_j = 100, \quad 0 \leq u_i \leq 1, \quad 0 \leq v_j \leq 1, \quad \Delta t = 0.01, \quad \alpha = 0.1,$$

$$\text{точность метода } \varepsilon = 0,0000001, z_{uv} = 50, G = 5000, P = 1000, P_m = 0.2, H_m = 0.05, P_c = 0.8, H_c = 0.01:$$

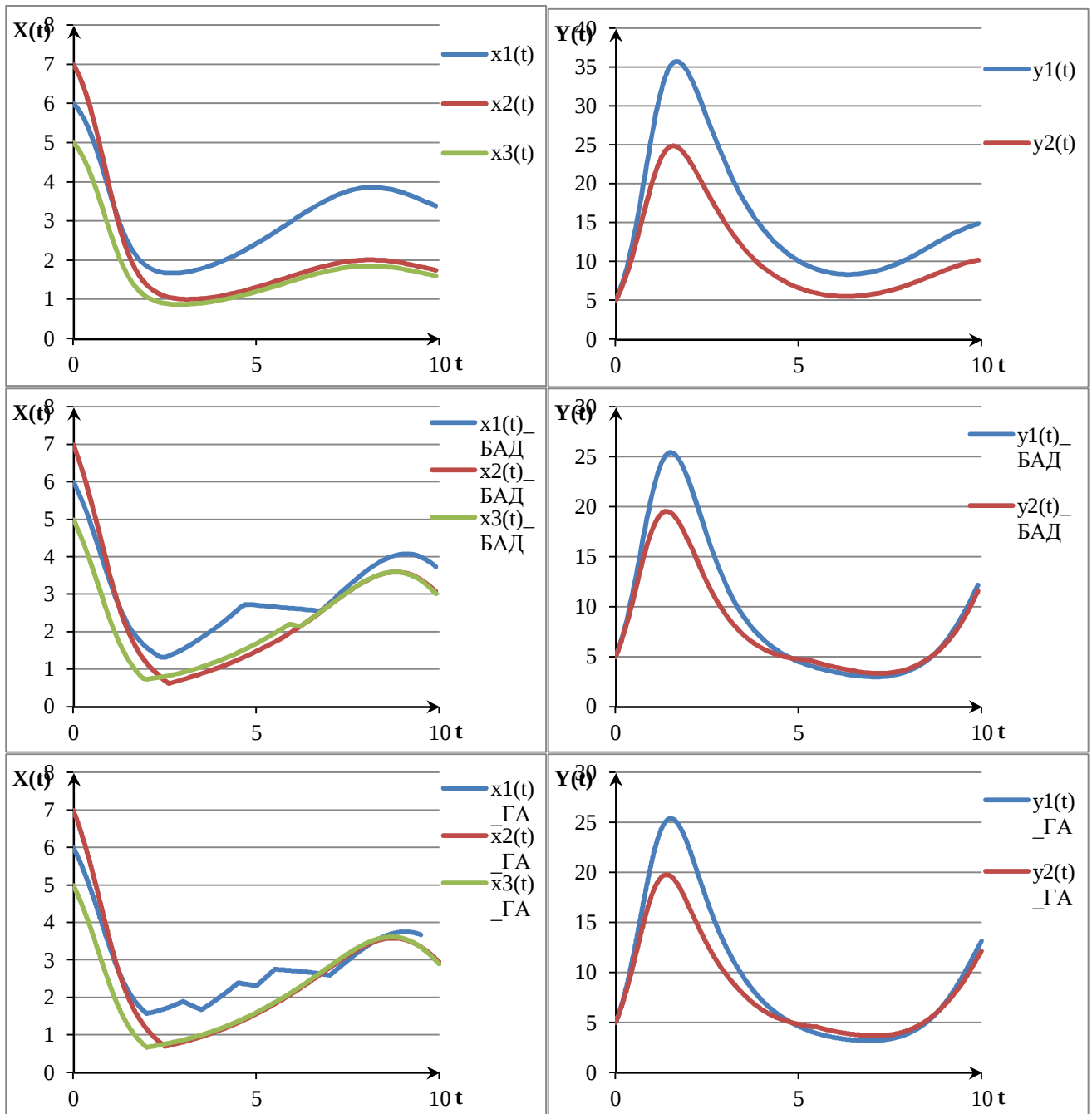


Рис.1.21 Графики численности популяции хищников $\bar{y}_j(t)$ и жертв $\bar{x}_i(t)$, построенные для модели без управления и найденные с помощью генетического алгоритма и с помощью метода БАД.

На рисунке (1.22) представлены соответствующие графики функций оптимального управления, найденные с помощью генетического алгоритма и с помощью метода БАД:

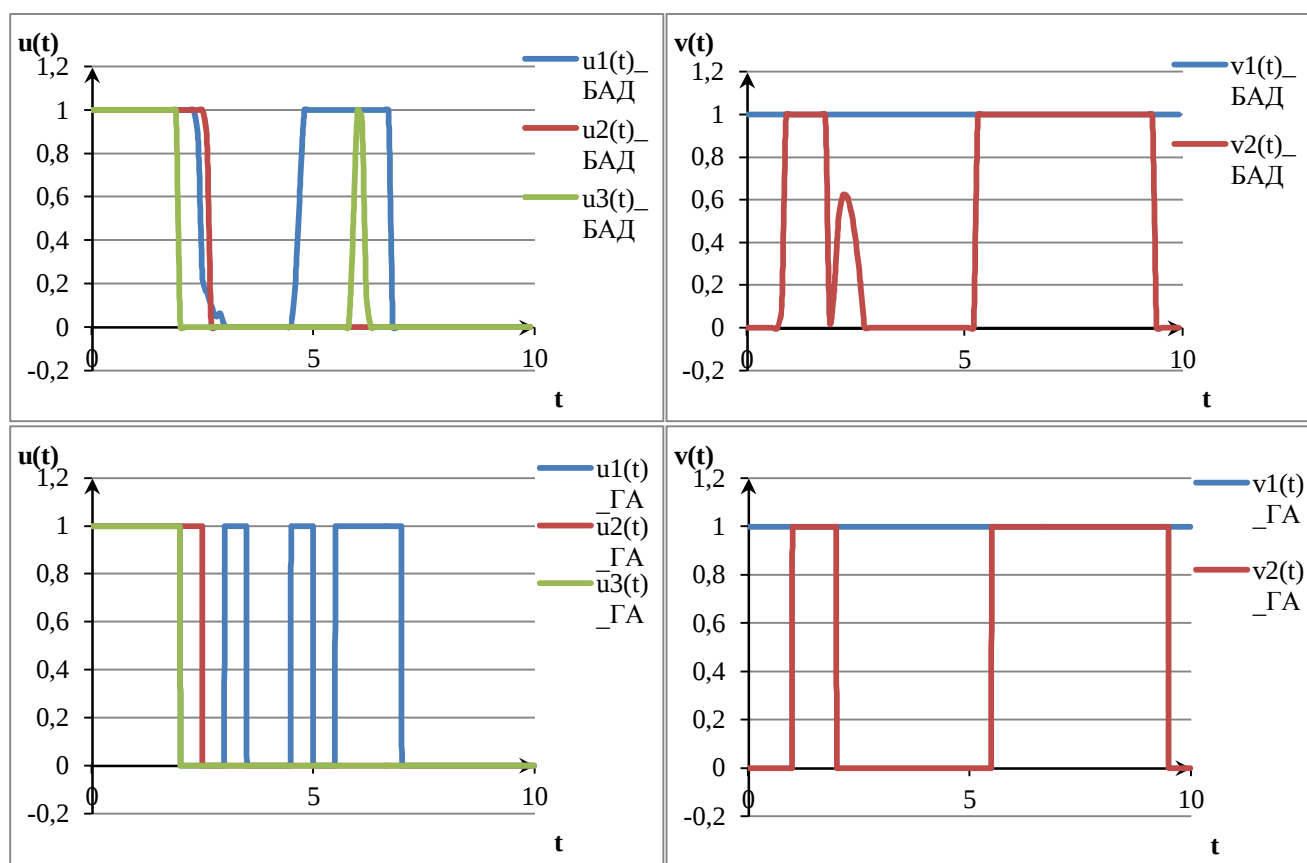


Рис.1.22 Графики оптимального управления $\bar{u}_i(t)$ и $\bar{v}_j(t)$, найденные с помощью генетического алгоритма и с помощью метода БАД.

Для генетического алгоритма за 5000 итераций при размерности популяции 1000 особей значение минимизируемых функций при указанных выше параметрах достигло следующих значений: $J_1(x, y, u, v) = -53,28$, $J_2(x, y, u, v) = 323,09187$, $J_4(x, y, u, v) = 37,64271$, для метода БАД - $J_1(x, y, u, v) = -53,717$, $J_2(x, y, u, v) = 322,58$, $J_4(x, y, u, v) = 35,155$, количество итераций алгоритма - 76181. Таким образом, решение, найденное методом БАД, соответствует более точному выполнению граничных условий и большему значению прибыли, полученной от реализации продукции.

Метод БАД работает быстрее, но при этом требует большего объема исходной информации о задаче. Особенностью генетического алгоритма является то, что он обрабатывает не значения параметров самой задачи, а их закодированную форму, и применяет вероятностные, а не детерминированные правила выбора. Преимуществом генетического алгоритма является то, что он

прост в реализации, использует только целевую функцию, а не ее производные либо иную дополнительную информацию. Генетический алгоритм может быть использован, когда не работает метод проекции градиента, например, когда функция не дифференцируема, или множество допустимых значений управления не компактное. В частности в поставленной задаче множество допустимых значений управления выбиралось дискретным: $u_i^l = \{0, 0.1\}$, $v_j^l = \{0, 0.1\}$. Для повышения скорости сходимости генетического алгоритма и улучшения устойчивости его работы применены модификации, учитывающие особенности решаемой задачи, а именно введен дополнительный параметр алгоритма z_{uv} – степень сжатия решения, позволяющий сократить размерность задачи, введены многоточечная мутация и многоточечное скрещивание, подобраны оптимальные параметры алгоритма, позволяющие построить оптимальное решение за наименьшее время.

Глава 2. Модель нейронной сети, описываемая системой интегро-дифференциальных уравнений

§1. Постановка задачи оптимального управления в модели искусственной нейронной сети

Искусственные нейронные сети (ИНС) — математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

В искусственной нейронной сети под нейроном подразумевается искусственный нейрон, то есть ячейка нейронной сети. Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Каждый синапс характеризуется величиной синаптической связи или ее весом, который по физическому смыслу эквивалентен электрической проводимости.

Текущее состояние нейрона определяется, как взвешенная сумма его входов:

$$s = \sum_{i=1}^n x_i \cdot w_i$$

Выход нейрона есть функция его состояния:

$$y = F\left(\sum_{i=1}^n x_i \cdot w_i\right) = F(s)$$

Нелинейная функция F называется активационной и может иметь различный вид.

Функция единичного скачка имеет вид

$$F(s) = \begin{cases} 0, s < T, \\ 1, s \geq T. \end{cases}$$

Такая функция активации используется в классическом искусственном нейроне. Это схема чрезмерно упрощена и не позволяет моделировать схемы с непрерывными сигналами. Отсутствие первой производной затрудняет применение градиентных методов для обучения таких нейронов.

Функция «линейный порог» (гистерезис) имеет вид

$$F(s) = \begin{cases} 0, s < 0, \\ \frac{s}{\Delta}, 0 \leq s < \Delta, \\ 1, s \geq \Delta. \end{cases}$$

Эта функция рассчитывается легко, но имеют разрывную первую производную в точках $s = 0$ и $s = \Delta$, что усложняет алгоритм обучения.

Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т.е. функция S-образного вида):

$$F(s) = \frac{1}{1 + e^{-\alpha s}}$$

Сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

Функция гиперболический тангенс имеет следующий вид

$$F(s) = th(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}.$$

Эта функция также часто применяется для сетей с непрерывными сигналами, она симметрична относительно начала координат, что является преимуществом по сравнению с сигмоидной функцией.

Возвращаясь к общим чертам, присущим всем НС, следует отметить принцип параллельной обработки сигналов, который достигается путем объединения большого числа нейронов в так называемые слои и соединения определенным образом нейронов различных слоев, а также, в некоторых конфигурациях, и нейронов одного слоя между собой, причем обработка взаимодействия всех нейронов ведется послойно.

Очевидно, что процесс функционирования НС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, необходимо найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

Этот этап называется обучением НС. Обучение НС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

В последние десятилетия большое внимание уделяется исследованию искусственных нейронных сетей, которые эффективно используются для решения задач классификации, оптимизации, прогнозирования, распознавания образов и т.д.

При моделировании нейронных сетей существенным является сосредоточенное и распределенное запаздывание, которое возникает при передаче сигнала от одного нейрона к другому.

Рассмотрим искусственную нейронную сеть, динамика которой описывается системой интегро-дифференциальных уравнений. Целью оптимизации является определение оптимальных весовых коэффициентов, характеризующих связи между нейронами.

Задача оптимального управления заключается в минимизации функционала:

$$J(\omega, u) = \int_0^T E(x(t), \omega(t), u(t)) dt + \Phi(x(T)) \quad (2.1)$$

где $E(x(t), \omega(t), u(t))$ - энергия рассматриваемой нейронной сети, зависящая от текущей конфигурации сети $x(t)$,

$\omega(t)$ - весовые коэффициенты системы,

$u(t)$ - внешнее воздействие на систему в момент времени t ;

$\Phi(x(T))$ - терминальное слагаемое - характеризует состояние системы в конечный момент времени

Динамика нейронной сети описывается системой интегро-дифференциальных уравнений:

$$\dot{x}_i(t) = -\beta_i x_i(t) + f_i \left(\int_{t-r}^t \sum_{j=1}^n \omega_{ij}(\tau) x_j(\tau - \delta_j) d\tau \right) + \gamma_i u_i(t), \quad (2.2)$$

$$t \in [0, T], \quad i = \overline{1, n},$$

где

функция $x_i(t)$ характеризует состояние i -го нейрона в момент времени t ,

первое слагаемое $-\beta_i x_i(t)$ характеризует затухание (β_i - скорость затухания возбуждения i -го нейрона),

второе слагаемое $f_i \left(\int_{t-r}^t \sum_{j=1}^n \omega_{ij}(\tau) x_j(\tau - \delta_j) d\tau \right)$ характеризует внешнее воздействие на нейрон всех других нейронов,

f_i - функция активации,

$\omega_{ij}(\tau)$ - весовой коэффициент, воздействие на i -ый нейрон, j -го нейрона, при этом $\omega_{ii}(\tau) \equiv 0$.

третье слагаемое $\gamma_i u_i(t)$ - внешнее воздействие на нейрон в момент времени t .

Начальные условия:

$$x_i(t) = \theta_i(t), t \in [-r - \delta, 0], \delta = \max_{i=1, n} \delta_i \quad (2.3)$$

τ, δ_i - запаздывания,

где $\theta_i(t)$, $i = \overline{1, n}$, - заданные непрерывные функции, $\varepsilon_i, \beta_i, \delta_j, \gamma_i$, $i, j = \overline{1, n}$, - заданные неотрицательные параметры модели.

Весовые коэффициенты $\omega_{ij}(t)$ и внешние воздействия $u_i(t)$ ограничены

$$|\omega_{ij}(t)| \leq a_{ij}, |u_i(t)| \leq a_i, i, j = \overline{1, n}, \text{ п.в. } t \in [t_0, t_1] \quad (2.4)$$

где a_{ij}, a_i , $i, j = \overline{1, n}$, заданные положительные параметры модели, полагаем $\omega_{ii}(t) \equiv 0$.

Задача оптимального управления заключается в том, чтобы обучить нейронную сеть таким образом, чтобы минимизировать функционал (2.1) и построить оптимальные значения весовых коэффициентов, внешнего управляющего воздействия.

Пусть $\bar{\omega}_{ij}(t)$, $i, j = \overline{1, n}$ - фиксированные весовые коэффициенты, $\bar{u}_i(t)$, $i = \overline{1, n}$ - оптимальные управляющие внешние воздействия, $[\bar{x}, \bar{\omega}, \bar{u}]$ - оптимальный процесс в задаче (2.1) - (2.4), функция $E(x, \omega, u)$ выпукла по u_i , тогда оптимальные значения $\bar{u}_i(t)$, $i = \overline{1, n}$ находятся из соотношения (2.5):

$$\bar{u}_i(t) = \begin{cases} a_i, & \Psi_i(t) > a_i \\ \Psi_i(t), & |\Psi_i(t)| \leq a_i \\ -a_i, & \Psi_i(t) < -a_i \end{cases} \quad \text{где } \Psi_i(t) = \frac{p_i(t)\gamma_i}{\lambda_0 \frac{\partial E(x, \bar{\omega}, \bar{u})}{\partial \bar{u}_i}}, \quad i = \overline{1, n}, \quad (2.5)$$

а сопряженные функции $p_i(t)$ удовлетворяют системе интегро-дифференциальных уравнений (2.6):

$$\begin{aligned} \dot{p}_l(t) = \lambda_0 \left\{ \frac{\partial E(\bar{x}(t), \bar{\omega}(t), \bar{u}(t))}{\partial x_l} + \right. \\ \left. + \beta_l p_l(t) - \sum_{i=1}^n \bar{\omega}_{il}(t + \delta_l) \int_{t+\delta_l}^{t+\delta_l+r} p_i(\tau) \frac{\partial f_i}{\partial z_i}(\bar{z}_i(\tau)) d\tau, \right. \\ \left. l = \overline{1, n} \right. \end{aligned} \quad (2.6)$$

и условиям трансверсальности на правом конце (2.7):

$$p_l(T) = -\lambda_0 \frac{\partial \Phi(x(T))}{\partial x} (x(T)), \quad p_l(t) = 0, \quad t > T, l = \overline{1, n}, \quad (2.7)$$

$$z_i(t) = \int_{t-r}^t \sum_{j=1}^n \omega_{ij}(s) x_j(s - \delta_j) ds$$

где

§2. Применение методологии БАД для обучения искусственной нейронной сети

Рассмотрим дискретную аппроксимацию начальной задачи, в которой используется правило левых прямоугольников для аппроксимации интеграла и схема Эйлера аппроксимации производных. Выберем шаг интегрирования

$$\Delta t = T / q \quad \text{так, что} \quad r = \nu_1 \cdot \Delta t, \quad \delta = \nu_2 \cdot \Delta t, \quad \nu > \nu_j > 0, \quad j = \overline{1, n},$$

$$t_k = k\Delta t, \quad k = 0, \dots, q, \quad t_q = T.$$

Дискретная задача оптимального управления состоит в минимизации функции

$$I(\omega, u, x) = \Delta t \sum_{k=0}^{q-1} \{E_k(x^k, \omega^k, u^k)\} + \Phi(x^q) \rightarrow \inf \quad (2.8)$$

при условиях

$$x_i^{k+1} = x_i^k (1 - \Delta t \beta_i) + \Delta t \left(f_i \left(\Delta t \sum_{s=k-\nu}^{k-1} \sum_{j=1}^n \omega_{ij}^s x_j^{s-\nu_j} \right) + \gamma_i u_i^k \right), \quad (2.9)$$

$$k = \overline{0, q-1}, \quad i = \overline{1, n},$$

где $E_k(x^k, \omega^k, u^k) = M \sum_{i,j=1}^n (\omega_{ij}^k)^2$, $\Phi(x^q) = \sum_{i=1}^n M_i (x_i^q - A_i)^2$,

$$x_i^k = x_i(t_k), \quad i = \overline{1, n}, \quad k = \overline{0, q},$$

$$\omega_{ij}^s = \omega_{ij}(t_s), \quad u_i^s = u_i(t_s) \quad i, j = \overline{1, n}, \quad s = \overline{0, q-1};$$

$$x_i^k = \theta_i(t_k), \quad k = -\nu - \nu_i, \dots, 0; \quad r = \nu \Delta t, \quad \delta_i = \nu_i \Delta t, \quad i = \overline{1, n}. \quad (2.10)$$

На функции управления наложены ограничения

$$|\omega_{ij}^k| \leq a_{ij}, \quad \omega_{ii}^k \equiv 0, \quad |u_i^k| \leq a_i, \quad k = \overline{0, q-1}, \quad i, j = \overline{1, n}, \quad (2.11)$$

Функция Лагранжа для этой задачи может быть представлена следующим выражением:

$$\begin{aligned} L(x, \omega, u, p, \lambda_0) = & \lambda_0 \Delta t \sum_{k=1}^{q-1} \left(M \sum_{i,j=1}^n (\omega_{ij}^k)^2 \right) + \lambda_0 \sum_{i=1}^n M_i (x_i^q - A_i)^2 + \\ & + \sum_{k=0}^{q-1} \sum_{i=1}^n p_i^{k+1} (x_i^{k+1} - x_i^k + \Delta t \beta_i x_i^k - \Delta t f_i (\Delta t \sum_{s=k-\nu}^{k-1} \sum_{j=1}^n \omega_{ij}^s x_j^{s-\nu_j}) - \Delta t \gamma_i \tilde{u}_i^k) \end{aligned} \quad (2.12)$$

Введем обозначение

$$z_i^k = \Delta t \sum_{s=k-\nu}^{k-1} \sum_{j=1}^n \omega_{ij}^s x_j^{s-\nu_j}.$$

Градиент минимизируемой функции

$$\frac{\partial L}{\partial \omega_{lp}^m} = -(\Delta t)^2 \sum_{k=1+m}^{\nu+m} p_l^{k+1} \frac{\partial f_l}{\partial z_l^k} (z_l^k) x_p^{m-\nu_p} + \lambda_0 2M \omega_{lp}^m \Delta t, \quad (2.13)$$

$$l, p = \overline{1, n}, \quad m = \overline{0, q-1},$$

где

$$p_l^m = (1 - \Delta t \beta_l) p_l^{m+1} + (\Delta t)^2 \sum_{i=1}^n \sum_{k=1+\nu_l}^{\nu+\nu_l+m} p_i^{k+1} \frac{\partial f_i}{\partial z_i^k} (\bar{z}_i^k) \bar{\omega}_{il}^{\nu_l+m}, \quad (2.14)$$

$$l = \overline{1, n}, \quad m = \overline{1, q-1}$$

$$p_l^q = -\lambda_0 2M_l (x_l^q - A_l), \quad p_l^m = 0, \quad m > q, \quad l = \overline{1, n}, \quad (2.15)$$

2.1 Алгоритм построения приближенного оптимального решения

Используя результаты предыдущего параграфа, опишем метод проекции градиента и метода штрафных функций для построения приближенного оптимального решения.

1. Зададим произвольный набор матриц $(W^k)^{(0)} = \{\omega_{ij}^k\}^{(0)}$, на диагоналях которых стоят нулевые элементы, и $|\omega_{ij}^k| \leq a_{ij}$, $i, j = \overline{1, n}$, $k = \overline{0, q-1}$.

2. По рекуррентным соотношениям (2.9), с использованием начальных значений $x_i^k = \theta_i(t_k)$, $k = -v - v_i, \dots, 0$, $i = \overline{1, n}$, где $\theta_i(t_k) \in R^n$ - заданные векторы в R^n , и наборов $(W^k)^{(0)}$ вычислим x_i^k , $i = \overline{1, n}$, $k = \overline{1, q}$. В результате получим набор векторов $x^1, \dots, x^q$, соответствующий выбранным $(W^k)^{(0)}$, который обозначим $(x^k)^{(0)}$.

3. Вычислим значение функции (2.8), используя $(x^k)^{(0)}$ и $(W^k)^{(0)}$, и обозначим эту величину $I^{(0)}$. Здесь верхний индекс в скобках соответствует номеру итерации.

4. Вычислим сопряженные векторы по формулам (2.15) и (2.14), начиная с индекса q и кончая индексом 1.

5. Вычислим коэффициенты ω_{ij}^k , $i, j = \overline{1, n}$, $k = \overline{0, q-1}$, соответствующие первой итерации: $(W^k)^{(1)} = \{\omega_{ij}^k\}^{(1)}$, по формулам

$$(\omega_{ij}^k)^{(l+1)} = (\omega_{ij}^k)^{(l)} - \alpha^{(l)} \left(\frac{\partial L}{\partial \omega_{ij}^k} \right)^{(l)},$$

где $\alpha^{(l)} > 0$ - величина шага градиентного спуска.

Если $(\omega_{ij}^k)^{(l+1)} < -a_{ij}$, то $(\omega_{ij}^k)^{(l+1)} = -a_{ij}$; если $(\omega_{ij}^k)^{(l+1)} > a_{ij}$, то $(\omega_{ij}^k)^{(l+1)} = a_{ij}$.

6. С полученными значениями весовых коэффициентов переходим ко второму шагу, используя вместо матриц $(W^k)^{(0)}$ матрицы $(W^k)^{(1)}$, полученные в результате первой итерации, строим $(x^k)^{(1)}$ по формулам (2.9), (2.10). Найдем значение минимизируемой функции, используя $(x^k)^{(1)}$, $(W^k)^{(1)}$, и обозначим эту величину $I^{(1)}$. Вычислим приращение $\Delta I^{(1)} = I^{(0)} - I^{(1)}$.

Если $\Delta I^{(1)} > 0$, то заменим на втором шаге $(W^k)^{(0)}$ на $(W^k)^{(1)}$, если $\Delta I^{(1)} \leq 0$, то уменьшим шаг градиентного спуска в два раза. Вычисления продолжим до тех пор, пока $\Delta I^{(l)} < \varepsilon$, где $\varepsilon > 0$ – заданная точность.

В качестве критерия остановки в работе рассмотрены следующие методы:

1. $I^{(k+1)} < I^{(k)}, |I^{(k)} - I^{(k+1)}| < \varepsilon$
2. $\|x^{(k)} - x^{(k+1)}\| < \varepsilon, \sqrt{\sum_{l=0}^q \left(\sum_{i=1}^n (x_i^{l(k)} - x_i^{l(k+1)})^2 \right)} < \varepsilon$
3. $\|w^{(k)} - w^{(k+1)}\| < \varepsilon, \sqrt{\sum_{l=0}^{q-1} \left(\sum_{i=1}^m \sum_{j=1}^n (w_{ij}^{l(k)} - w_{ij}^{l(k+1)})^2 \right)} < \varepsilon$

2.2 Анализ влияния параметров задачи на оптимальное решение

Метод отрабатывался при следующих параметрах задачи:

$$I(\omega, x) = I_1(\omega) + I_2(x^q), \quad I_1(\omega) = M \Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2, \quad I_2(x^q) = \sum_{k=1}^n M_k (x_k^q - A_k)^2$$

$$f_i(x) = K_i x, \quad x_0=1, \beta_i=0.01, A_1=11, A_2=12, A_3=7, \quad K_i = 1; \quad r = 1, \delta = 0.5 \quad \Delta t = 0.1, T = 10, N = 3, |\omega_{ij}^k| \leq 1, \text{ штрафы } M_k = 10000, M=1; \alpha = 0,1; \varepsilon = 10^{-8}.$$

На рисунках (2.1) - (2.2) представлены графики оптимального решения системы и соответствующего ему оптимального управления при заданных параметрах и виде минимизируемого функционала:

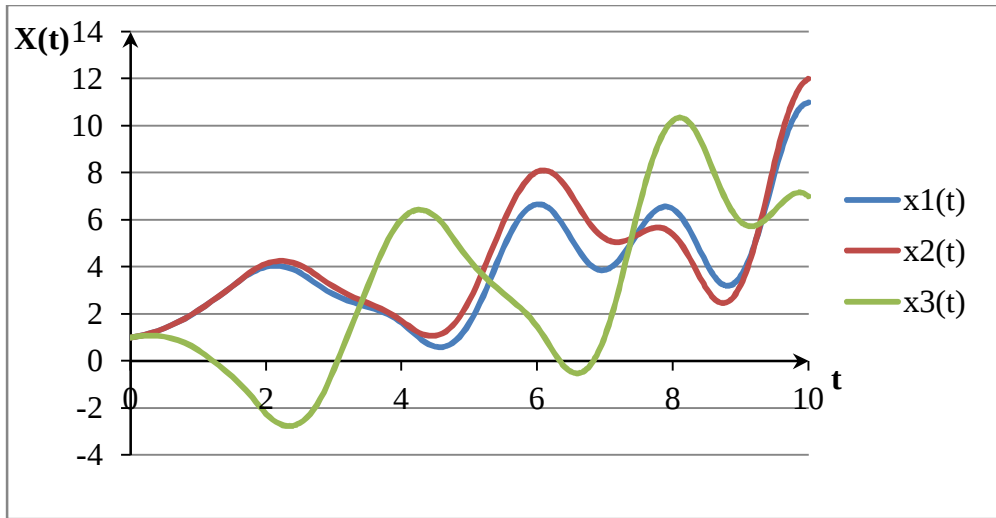


Рис. 2.1. График функций состояния нейронов $\bar{x}_i(t)$, $i = \overline{1,3}$

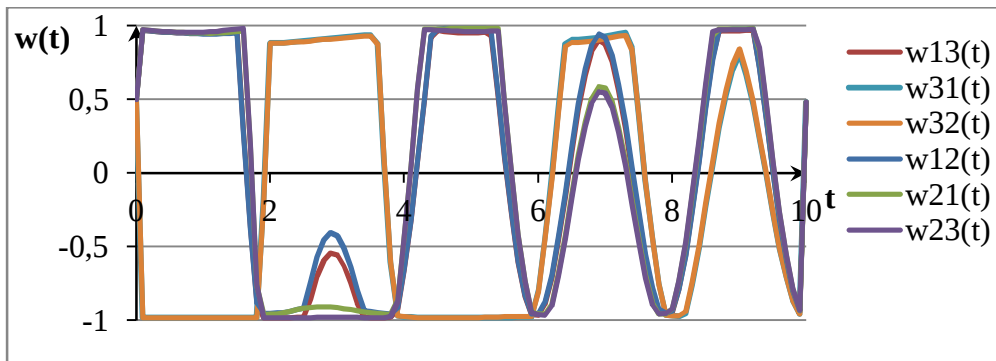


Рис. 2.2. График оптимальных весовых коэффициентов $\bar{w}_{ij}(t)$, $i, j = \overline{1,3}, i \neq j$

При этом значение минимизируемых функций для найденных оптимальных весовых коэффициентов соответственно равны $I_1(\omega) = 40,67$ и $I_2(x^q) = 4,14 \cdot 10^{-7}$.

На рисунках (2.3)-(2.5) представлены соответствующие полученному оптимальному решению график зависимости значения минимизируемой функции от числа итераций и графики зависимости нормы изменения вектора состояния и нормы изменения вектора управления от номера итерации:

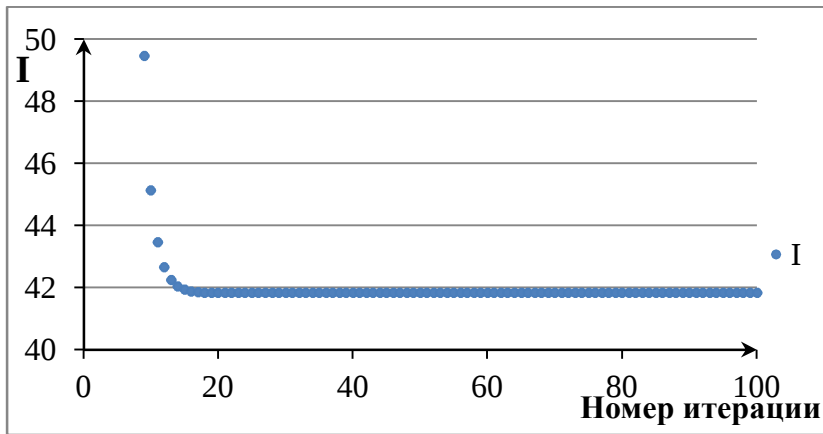


Рис. 2.3 График изменения значения минимизируемой функции в зависимости от числа итераций.

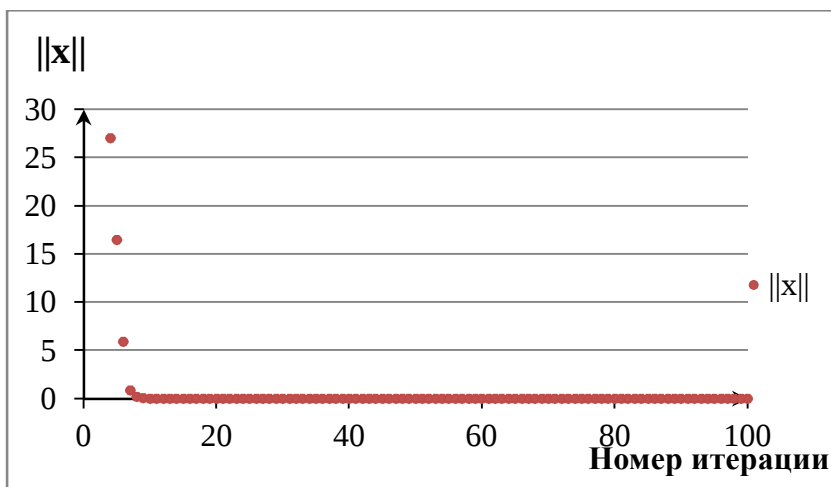


Рис. 2.4 График изменения нормы вектора состояния в зависимости от числа итераций

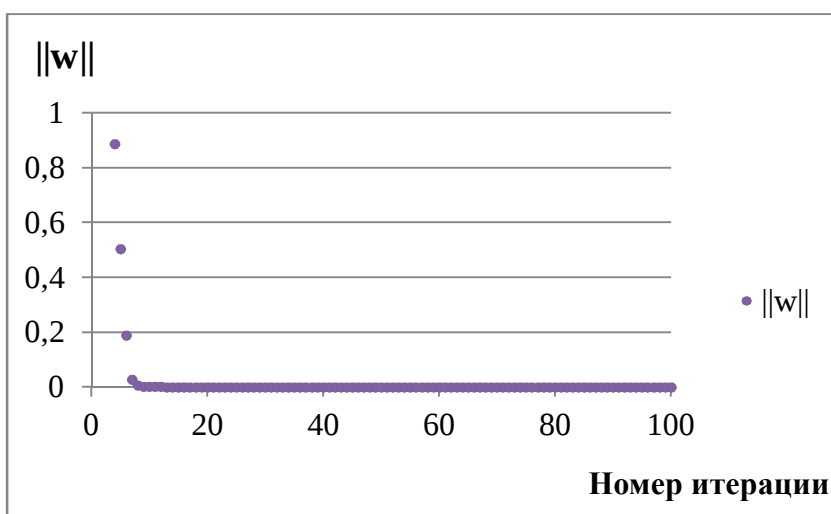


Рис. 2.5 График изменения нормы вектора управления в зависимости от числа итераций

За 100000 итераций значение минимизируемых функций достигло величин $I_1(\omega) = 40,67$ и $I_2(x^q) = 4,14 \cdot 10^{-7}$.

На рисунках (2.3)-(2.5) значения минимизируемой функции и нормы изменения вектора состояния и вектора управления приведены для первых 100 итераций, поскольку на следующих итерациях скорость изменения указанных значений достаточно мала.

Ниже в таблицах 2.1–2.3 представлены зависимости значения минимизируемого функционала и количества итераций от точности метода ε при различных критериях остановки алгоритма при следующих параметрах задачи:

$$f_i(x) = K_i \arctg(x), x_0=1, \beta_i=0.01, A_1=11, A_2=12, A_3=7, K_i = 1; r = 0.5, \delta = 0.1 \Delta t = 0.1, T = 10, N = 3, |\omega_{ij}^k| \leq 1, \text{штрафы } M_k = 10000, M=1; \alpha = 0,1,$$

$$|I^{(k)} - I^{(k+1)}| < \varepsilon_1 \quad (2.16)$$

$$\sqrt{\sum_{l=0}^q \left(\sum_{i=1}^n (x_i^{l(k)} - x_i^{l(k+1)})^2 \right)} < \varepsilon_2 \quad (2.17)$$

$$\sqrt{\sum_{l=0}^{q-1} \left(\sum_{i=1}^m \sum_{j=1}^n (w_{ij}^{l(k)} - w_{ij}^{l(k+1)})^2 \right)} < \varepsilon_3 \quad (2.18)$$

ε_1	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
I_{opt}	18,45631	18,44484	18,44356	18,44352	15,62689	15,39214
Кол-во итераций	153	185	217	223	260810	499989

Табл. 2.1 Значения минимизируемого функционала и количества итераций от точности метода ε , определяемой условием (2.16)

ε_2	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}	10^{-16}	10^{-17}
I_{opt}	18,4398	18,43728	18,43434	18,43085	15,63156	15,63146	15,39214
Кол-во итераций	229	328	445	584	259301	259331	499989

Табл. 2.2 Значения минимизируемого функционала и количества итераций от точности метода ε , определяемой условием (2.17)

ε_3	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}	10^{-16}	10^{-17}
I_{opt}	18,43985	18,43982	18,43848	18,43685	16,50003	15,62837	15,62832
Кол-во итераций	227	228	280	345	116369	260312	260321

Табл. 2.3 Значения минимизируемого функционала и количества итераций от точности метода ε , определяемой условием (2.18)

Из таблиц (2.1)-(2.3) следует, что вне зависимости от выбранного критерия остановки увеличение точности метода приводит к уменьшению значения минимизируемого функционала и увеличению количества итераций алгоритма. При этом, как следует из таблиц (2.1)-(2.3) и рисунков (2.3)-(2.5), на каждой итерации алгоритма происходит уменьшение значения минимизируемого функционала и уменьшения изменений нормы разности векторов состояния и управления.

Далее приведем анализ нейронной сети, при условии, что на весовые коэффициенты наложены ограничения $|\omega_{ij}| \leq A$, $\omega_{ii} = 0, i, j = 1, n$.

Проведем анализ решения в зависимости от величины запаздывания при различных функциях активации.

В дальнейшем рассмотрим случаи, когда функция активация имеет вид:

1. $f_i(x) = K_i x$
2. $f_i(x) = K_i \arctg(x)$

1. $f_i(x) = K_i x$

Исследуем влияние запаздывания δ при следующих параметрах задачи:

$x=1, \beta_1=0, A_1=11, A_2=12, A_3=7, K_i = 1; r = 0.5, \Delta t = 0.1, T = 10, N = 3, |\omega_{ij}^k| \leq 1$,
штрафы $M_k = 10000, M=0; \alpha = 0,1; \varepsilon = 10^{-8}$;

$$I(\omega, x) = M \Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2 + \sum_{k=1}^n M_k (x_k^q - A_k)^2$$

Ниже представлены графики зависимости функций состояния от величины запаздывания δ ($\delta = \nu 2 \cdot \Delta t$) и соответствующие этой величине запаздывания

значения функций состояния в момент времени T , значения ошибки обучения искусственной нейронной сети и достижимые значения функций состояния при определенных выше параметрах задачи:

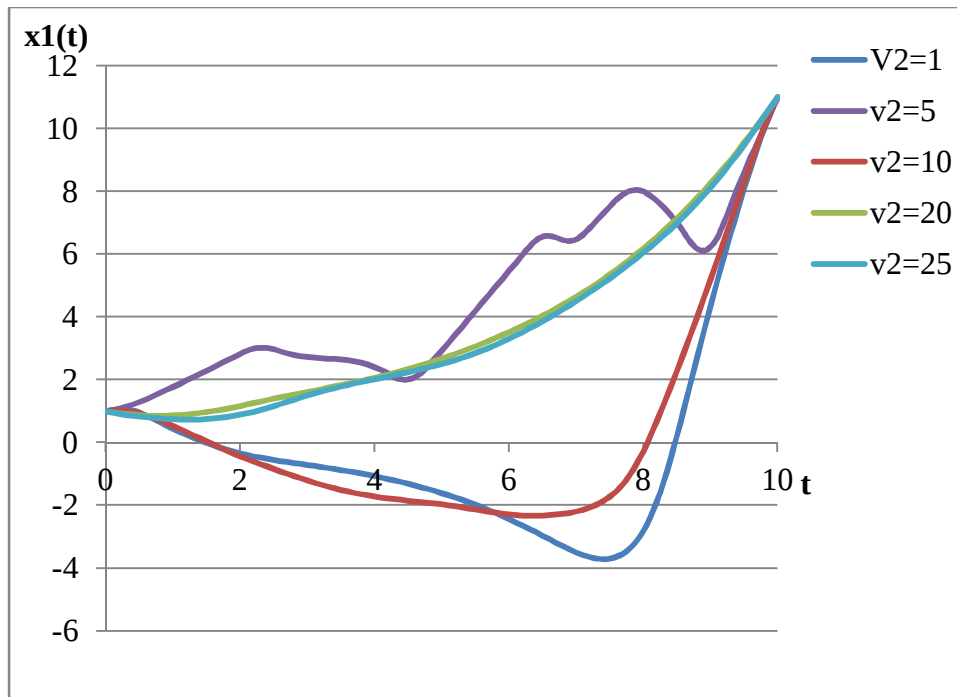


Рис. 2.6. График зависимости $\bar{x}_1(t)$ при различных значениях запаздывания δ

V2	$X_1(T)$	$ A_1 - X_1(T) $	I_{opt}	X_1 достиж.
1	10,99999	2,2455E-06	2,10244E-11	1326,631
5	11,000001	1,1122E-06	2,81522E-12	400,9746
10	10,99999	9,40602E-07	3,33994E-12	151,2506
20	10,99996	2,52603E-07	1,56804E-13	40,88523
25	11,000001	1,4365E-07	2,29828E-13	30,04821

Табл. 2.4. Значения $\bar{x}_1(T)$, I_{opt} и достижимое значение $\bar{x}_1(t)$ при различных значениях запаздывания δ

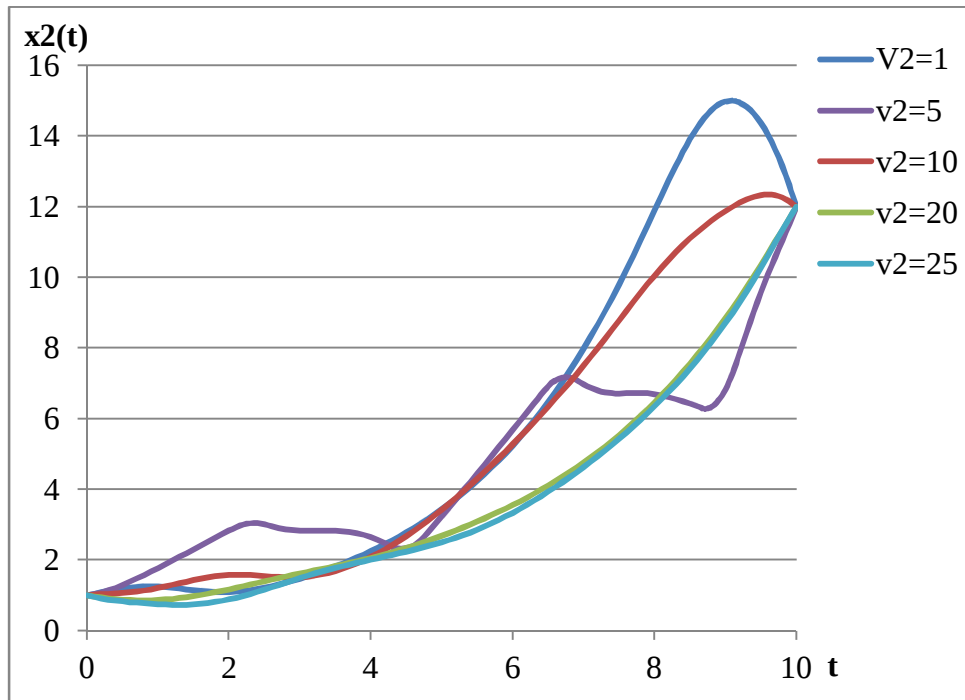


Рис. 2.7. График зависимости $\bar{x}_2(t)$ при различных значениях запаздывания δ

V2	$X_2(T)$	$ A_2 - X_2(T) $	I_{opt}	$X_{2 \text{ достиж.}}$
1	12,000003	3,11012E-06	2,10244E-11	1326,631
5	11,999998	1,25324E-06	2,81522E-12	400,9746
10	12,000001	1,19317E-06	3,33994E-12	151,2506
20	11,9999997	2,52336E-07	1,56804E-13	40,88523
25	11,999999	3,30698E-07	2,29828E-13	30,04821

Табл. 2.5. Значения $\bar{x}_2(T)$, I_{opt} и достижимое значение $\bar{x}_2(t)$ при различных значениях запаздывания δ

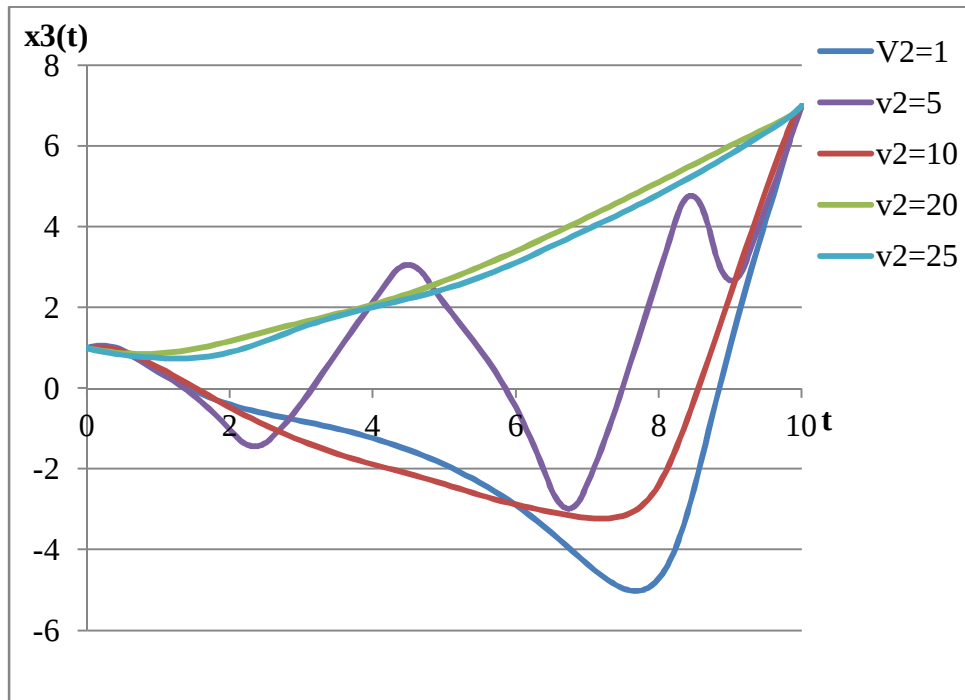


Рис. 2.8. График зависимости $\bar{x}_3(t)$ при различных значениях запаздывания δ

V2	$X_3(T)$	$ A_3 - X_3(T) $	I_{opt}	$X_{3 \text{ достиж.}}$
1	6,99999	2,51182E-06	2,10244E-11	1326,631
5	7,00000008	8,73167E-08	2,81522E-12	400,9746
10	6,99999	1,01566E-06	3,33994E-12	151,2506
20	6,99999	1,71238E-07	1,56804E-13	40,88523
25	7,0000003	3,15961E-07	2,29828E-13	30,04821

Табл. 2.6. Значения $\bar{x}_3(T)$, I_{opt} и достижимое значение $\bar{x}_3(t)$ при различных значениях запаздывания δ

С увеличением запаздывания δ точность сходимости оптимального решения увеличивается, значение минимизируемой функции уменьшается. При увеличении запаздывания δ множество достижимых значений уменьшается.

$$2. f_i(x) = K_i \arctg(x)$$

Исследуем влияние запаздывания δ при следующих параметрах задачи:

$x=1$, $\beta_i=0$, $A_1=11$, $A_2=12$, $A_3=7$, $K_i = 5$; $r = 0.5$, $\Delta t = 0.1$, $T = 10$, $N = 3$, $|\omega_{ij}^{\kappa}| \leq 1$,
штрафы $M_k = 10000$, $M=1$; $\alpha = 0,1$; $\varepsilon = 10^{-8}$;

$$I_{\text{opt}} = M \Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2 + \sum_{k=1}^n M_k (x_k^q - A_k)^2$$

Ниже представлены графики зависимости функций состояния от величины запаздывания δ ($\delta = v_2 \cdot \Delta t$) и соответствующие этой величине запаздывания значения функций состояния в момент времени T и значения ошибки обучения искусственной нейронной сети:

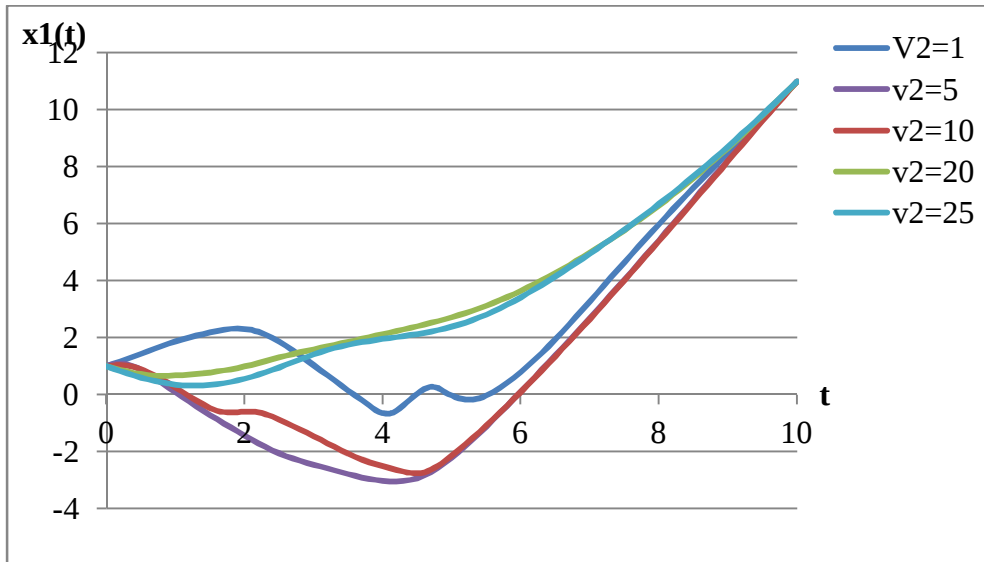


Рис. 2.9. График зависимости $\bar{x}_1(t)$ при различных значениях запаздывания δ

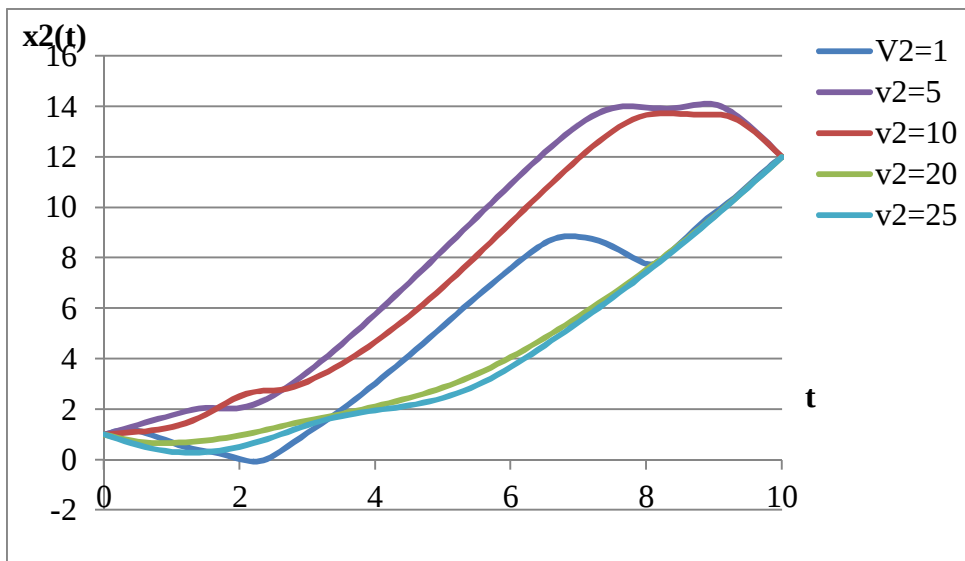


Рис. 2.10. График зависимости $\bar{x}_2(t)$ при различных значениях запаздывания δ

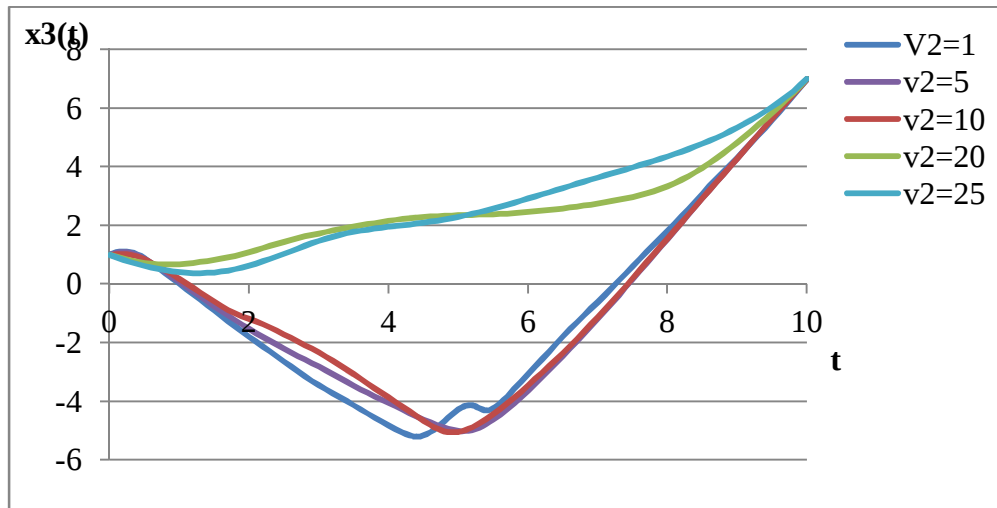


Рис. 2.11. График зависимости $\bar{x}_3(t)$ при различных значениях запаздывания δ

V1	A1- X1(T)	A2- X2(T)	A3- X3(T)	I1opt	I2opt	Iopt
1	2,01E-05	1,26E-05	4,20E-05	24,5436	2,3E-09	24,543555
5	4,02E-05	2,5756E-05	7,44E-05	45,1937	7,8E-09	45,19367
10	5,88E-05	3,0282E-05	8,2836E-05	27,2803	1,1E-08	27,280318
20	1,64E-05	2,6648E-05	2,429E-06	6,49857	9,9E-10	6,4985722
25	2,14E-05	3,3161E-05	3,6199E-06	5,75454	1,6E-09	5,7545357

Табл. 2.7. Значения ошибки обучения и I_{opt} при различных значениях запаздывания δ

С увеличением запаздывания δ точность сходимости оптимального решения увеличивается, значение минимизируемой функции увеличивается.

Исследуем влияние запаздывания r при следующих параметрах задачи:

$x=1$, $\beta_1=0$, $A_1=5$, $A_2=3$, $A_3=4$, $K_i = 1$; $\delta = 0.2$, $\Delta t = 0.1$, $T = 10$, $N = 3$, $|\omega_{ij}^{\kappa}| \leq 1$,
штрафы $M_k = 10000$, $M=1$; $\alpha = 0,1$; $\varepsilon = 10^{-8}$;

$$I_{\text{opt}} = M\Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2 + \sum_{k=1}^n M_k (x_k^q - A_k)^2$$

Ниже представлены графики зависимости функций состояния от величины запаздывания r ($r = v_1 \cdot \Delta t$) и соответствующие этой величине запаздывания значения функций состояния в момент времени T и значение ошибки обучения искусственной нейронной сети:

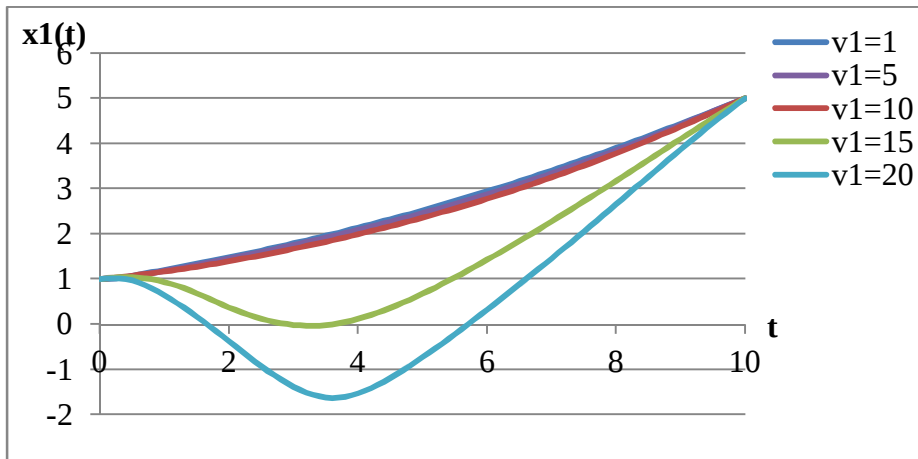


Рис. 2.12. График зависимости $\bar{x}_1(t)$ при различных значениях запаздывания r

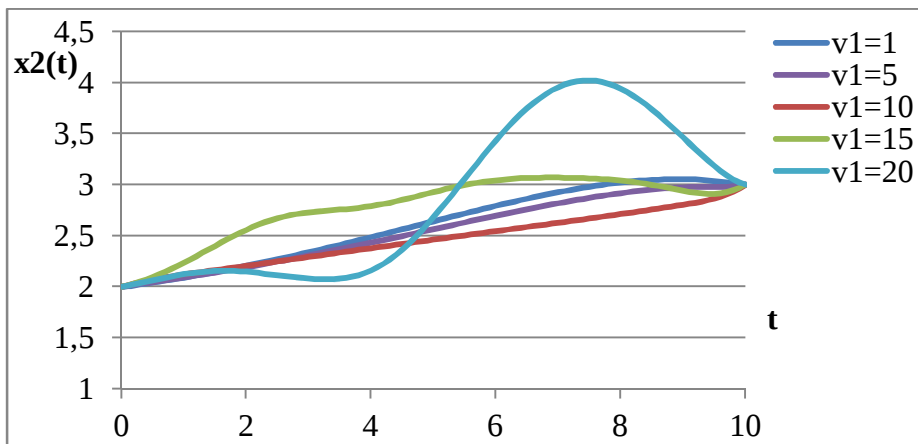


Рис. 2.13. График зависимости $\bar{x}_2(t)$ при различных значениях запаздывания r

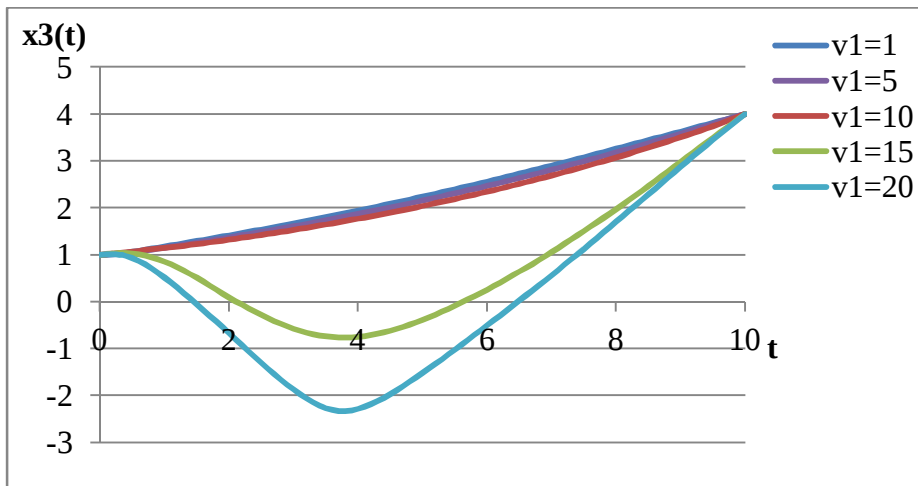


Рис. 2.14. График зависимости $\bar{x}_3(t)$ при различных значениях запаздывания r

V1	$ A_1 - X_1(T) $	$ A_2 - X_2(T) $	$ A_3 - X_3(T) $	$I1_{opt}$	$I2_{opt}$	I_{opt}
1	0,000389	2,128E-05	0,000144624	25,5237	0,001729	25,52543
5	1,68E-05	1,175E-06	6,48506E-06	1,086759	3,26E-06	1,086763
10	4,559E-06	3,918E-07	1,79002E-06	0,310785	2,41E-07	0,310785

15	6,267E-06	4,518E-07	8,39939E-06	1,39347	1,10E-06	1,393472
20	1,019E-05	2,58E-06	1,17153E-05	4,265248	2,48E-06	4,26525

Табл. 2.8. Значения ошибки обучения и I_{opt} при различных значениях запаздывания r

Проведено исследование влияния штрафного коэффициента M_k на значение минимизируемого функционала.

Ниже в таблице 2.9 представлены значения минимизируемой функции при различных значениях штрафного коэффициента:

M_k	I_{opt}	I_1	I_2
100	13,61725	13,57733	0,039922
500	13,65019	13,64205	0,008138
1000	13,67428	13,67019	0,004096
5000	15,32398	15,32309	0,000883
10000	18,79978	18,79933	0,000444

Табл. 2.9.

Из таблицы 2.6 следует, что если целью управления является минимизация энергии сети и минимизация ошибки обучения, то с увеличением штрафного коэффициента M_k точность сходимости оптимального решения увеличивается, но при этом увеличивается и энергия искусственной нейронной сети.

§3 Применение генетического алгоритма

3.1. Практическая реализация генетического алгоритма

Рассмотрим решение задачи (2.5)-(2.7) с помощью генетического алгоритма при условии, что весовые коэффициенты ω_{ij}^k принимают дискретные значения на заданном отрезке $[0; 1]$. В работе в качестве дополнительных значений весовых коэффициентов рассматривается $\{0; 0,142857; 0,285714; 0,428571; 0,571428; 0,714285; 0,857142; 1\}$. Закодируем каждое значение весовых коэффициентов с помощью бинарного кода. Тогда значения весовых коэффициентов закодируются следующим образом $\{000, 001, 010, 011, 100, 101, 110, 111\}$. Каждое значение

весовых коэффициентов кодируется тремя битами, набор ω_{ij}^k размером $N*N*(q+1)$, тогда длина хромосомы будет определяться следующей величиной $3*N*N*(q+1)$.

1. Произвольным способом создается L хромосом (где L – размер популяции), т.е. L наборов весовых коэффициентов.

		$3 \cdot N \cdot N \cdot (q+1)$													
L наборов весовых коэффициентов	[	0	1	1	0	1	.	.	.	.	1	0	1	1	1
		1	0	1	1	1	.	.	.	.	0	1	0	1	0
		1	1	0	0	1	.	.	.	.	0	0	0	1	0
		.	.	.	.	.	.	.	.	.	.	.	.	.	.
		0	1	1	0	1	.	.	.	.	0	1	0	1	1

2. Выполняются операции скрещивания и мутации, в результате чего к имеющимся L хромосомам добавляются еще d новых хромосом.

Скрещивание:

0	1	1	0	1	0		0	0	1	1	1	0	1	0	1	1	1	0
1	0	1	1	0	0		1	0	1	0	1	1	0	0	0	1	0	0
0	1	1	0	1	0		1	0	1	0	1	1	0	0	0	1	0	0
1	0	1	1	0	0		0	0	1	1	1	0	1	0	1	1	1	0

Мутация:

0	1	1	1	0	0	1	1	1	0	1	0	1	1	0	0	0	0	0
0	1	1	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0

3. Оценивается степень приспособленности каждой хромосомы с помощью вычисления состояния системы и минимизируемой функции $I(x, \omega)$. Для этого каждая хромосома декодируется в набор весовых коэффициентов, а затем вычисляются значения x_k по формуле (2.6) и значения $I(x, \omega)$ по формуле (2.5).

		$3 * N * N * (q+1)$									
L наборов весовых коэффициентов	[	0 1 1 0 1	.	.	.	1 0 1 1 1	$\Rightarrow I([x]^1, [\omega]^1)$				
		1 0 1 1 1	.	.	.	0 1 0 1 0	$\Rightarrow I([x]^2, [\omega]^2)$				
		1 1 0 0 1	.	.	.	0 0 0 1 0	$\Rightarrow I([x]^3, [\omega]^3)$				
		.	.	.	.	.	.				
		0 1 1 0 1	.	.	.	0 1 0 1 1	$\Rightarrow I([x]^L, [\omega]^L)$				
+d наборов весовых коэффициентов	[	1 1 0 0 1	.	.	.	0 0 0 1 0	$\Rightarrow I([x]^{L+1}, [\omega]^{L+1})$				
		0 1 1 0 1	.	.	.	0 1 0 1 1	$\Rightarrow I([x]^{L+d}, [\omega]^{L+d})$				

4. Выбираем L особей с наименьшим значением минимизируемой функции $I(x, \omega)$ (процедура селекции).

Шаги 1 – 4 повторяются S раз, где S – число поколений.

3.2 Анализ применения генетического алгоритма в зависимости от параметров

Исследуем влияние штрафных коэффициентов M_k на оптимальное решение при следующих параметрах задачи:

$$x=1, \beta_1=0, A_1=11, A_2=12,5, A_3=7, v_1=2, v_2=1, t=2, K_i=10, q=5, N=3,$$

$$\omega_{ij}^k \in \{0; 0,142857; 0,285714; 0,428571; 0,571428; 0,714285; 0,857142; 1\},$$

$$\text{кол-во поколений } S=100, \text{ кол-во особей } L=300, M=0, M_k=100,$$

$$I_{opt} = M \Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2 + \sum_{k=1}^n M_k (x_k^q - A_k)^2$$

В таблице 2.10 представлена зависимость значения минимизируемой функции от штрафа M_k :

M_k	$ A_1 - X_1(T) $	$ A_2 - X_2(T) $	$ A_3 - X_3(T) $	I_{opt}
10	0,004378	0,000825	0,000416	$2 \cdot 10^{-5}$
100	0,003224	0,003219	0,001583	$2,326 \cdot 10^{-7}$
1000	0,00257	0,000513	8,06E-05	$6,804 \cdot 10^{-9}$

10000	0,002254	0,001415	0,001309	$8,797 \cdot 10^{-10}$
-------	----------	----------	----------	------------------------

Табл. 2.10

С увеличением штрафного коэффициента M_k точность сходимости оптимального решения увеличивается, значение минимизируемой функции уменьшается.

Исследуем влияние параметров алгоритма на его работу. Параметрами самого алгоритма являются количество особей в популяции L и количество поколений S .

Далее приведены зависимости значения минимизируемой функции от количества особей и количество поколений при следующих параметрах задачи:

$x=1$, $\beta_i=0$, $A_1=4$, $A_2=2,5$, $A_3=3$, $K_i = 1$; $\tau = 0.2$, $\delta_i=0.1$, $\Delta t = 0.1$, $T = 5$, $N = 3$,
 $\omega_{ij}^k \in \{0; 0,0142857; 0,0285714; 0,0428571; 0,0571428; 0,0714285; 0,0857142; 0,1\}$,
 штрафы $M_k = 10000$, $M=0$; $\alpha = 0,1$; $\varepsilon = 10^{-8}$;

$$I_{\text{opt}} = M\Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2 + \sum_{k=1}^n M_k (x_k^q - A_k)^2$$

В таблице 2.11. представлена зависимость значения минимизируемой функции от количества особей и количество поколений:

Особи Поколения	500	600	700	800	900	1000
200	0,028747	0,010964	0,00929	0,013543	0,007181	0,00758
300	0,009305	0,006177	0,005885	0,005854	0,004956	0,004619
400	0,005528	0,004685	0,003405	0,002725	0,003908	0,003505
500	0,003949	0,003748	0,002026	0,001248	0,003022	0,001579
600	0,003589	0,003197	0,001844	0,001029	0,00142	0,001263
700	0,003564	0,003037	0,001049	0,001363	0,00122	0,001015
800	0,003159	0,003074	0,000342	0,001742	0,001041	0,000343
900	0,002151	0,001822	0,00024	0,000934	0,000478	0,000575
1000	0,001529	0,001102	0,000254	0,000152	0,000148	0,000225

Табл. 2.11

На рисунке 2.15 представлен соответствующий график зависимости значения минимизируемой функции от количества поколений:

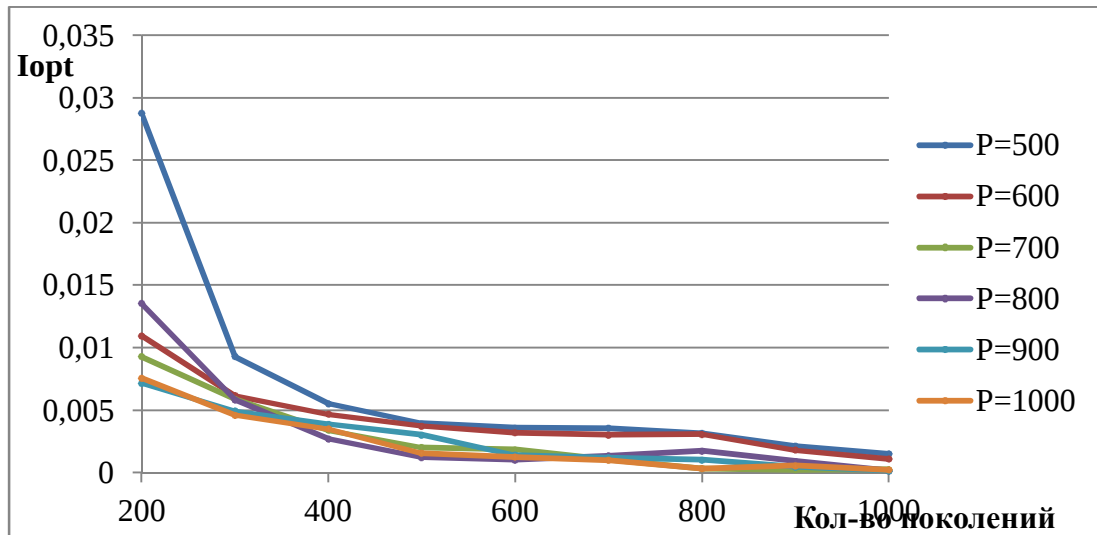


Рис. 2.15. График зависимости значения минимизируемой функции от количества поколений

Анализируя полученные данные, можно сделать вывод, что при использовании данного алгоритма целесообразно задавать количество особей не менее 700, а количество поколений не менее 500.

§4. Сравнение результатов работы метода БАД и генетического алгоритма

Для сравнения двух алгоритмов, рассмотрим случай, когда весовые коэффициенты могут принимать значения на отрезке $[0,0.1]$. Разобьем отрезок $[0,0.1]$ на $2^3 - 1$ отрезков, тогда значения весовых коэффициентов могут быть закодированы следующим образом:

0	0,014285	0,028571	0,042857	0,057142	0,071428	0,085714	0,1
000	001	010	011	100	101	110	111

Так как значение минимизируемой функции, получаемое при работе генетического алгоритма является случайной величиной, то для оценки результатов работы генетического алгоритма необходимо получить среднее значение искомой величины. Для этого каждое измерение проводится 100 раз, и вычисляется среднее значение искомой величины.

Ниже представлен график зависимости значения минимизируемой функции от номера итерации при работе генетического алгоритма при следующих параметрах задачи и метода:

$x_i^0=1$, $\beta_i=0$, $A_1=11$, $A_2=12$, $A_3=7$, $v_1 = 2$, $v_2= 1$, $t = 2$, $K_i = 10$, $q = 5$, $N = 3$, кол-во поколений = 100, кол-во особей = 300, $M=1$, $M_k=100$,

$$I_1 = M\Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2, I_2 = \sum_{k=1}^n M_k (x_k^q - A_k)^2, I_{\text{opt}} = I_1 + I_2.$$

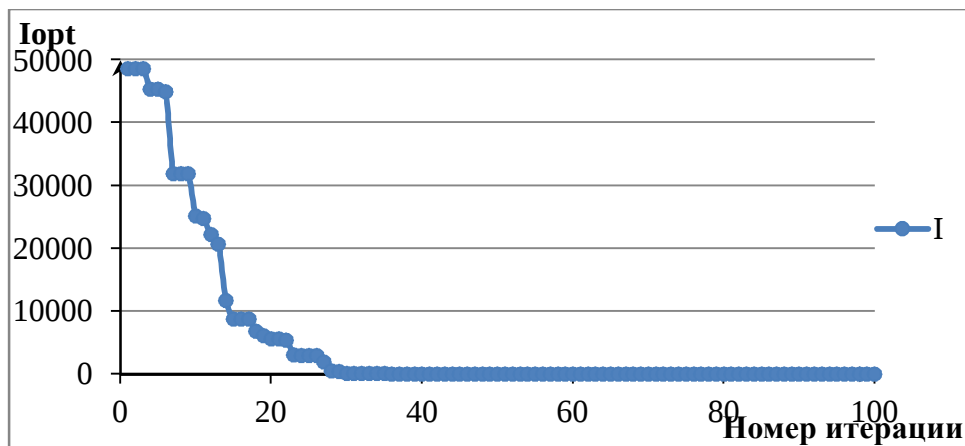


Рис. 2.16. График зависимости значения минимизируемой функции от номера итерации

Далее представлен график зависимости значения минимизируемой функции от номера итерации при работе метода БАД при следующих параметрах задачи и метода:

$x_i^0=1$, $\beta_i=0$, $A_1=11$, $A_2=12$, $A_3=7$, $v_1 = 2$, $v_2= 1$, $t = 2$, $K_i = 10$, $\alpha = 0.1$, $\varepsilon = 0.0001$,

$q = 5$, $N = 3$, $0 \leq \omega_{ij} \leq 0,1$; $M=1$, $M_k=100$, $I_1 = M\Delta t \sum_{k=0}^{q-1} \sum_{i,j=1}^n (\omega_{ij}^k)^2$,

$$I_2 = \sum_{k=1}^n M_k (x_k^q - A_k)^2, I_{\text{opt}} = I_1 + I_2$$

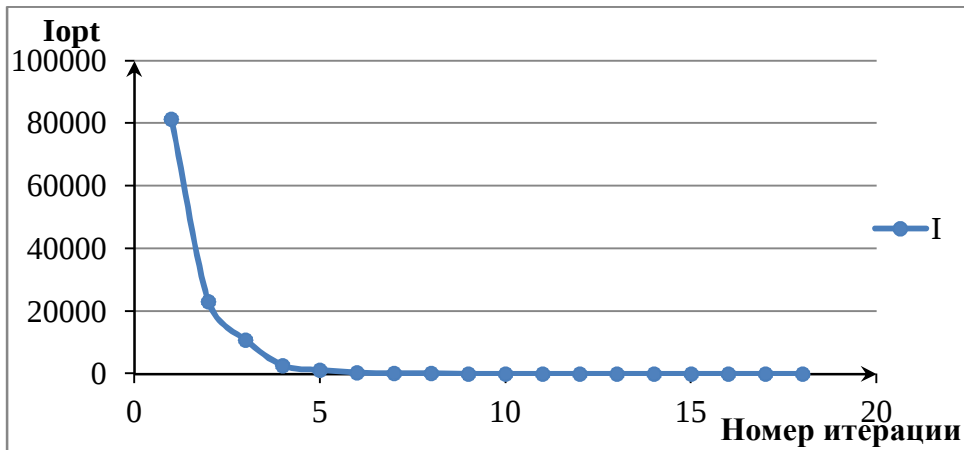


Рис. 2.17. График зависимости значения минимизируемой функции от номера итерации

По графикам 2.16, 2.17 видно, что при работе генетического алгоритма не на всех итерациях значение минимизируемой функции уменьшается, при этом метод БАД требует меньшего количества итераций.

Графики функций состояния для метода проекции градиента и генетического алгоритма:

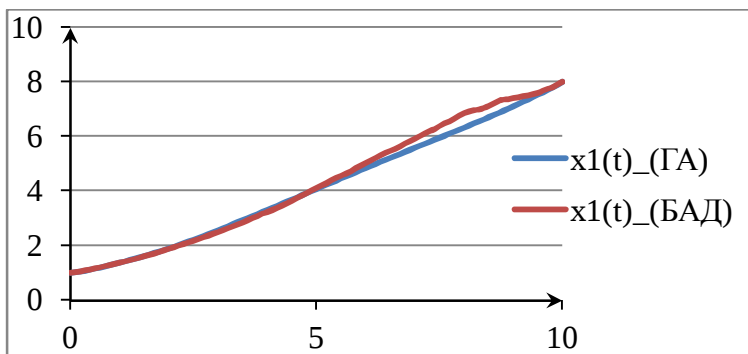


Рис. 2.18. График функции $\bar{x}_1(t)$ для метода БАД и генетического алгоритма

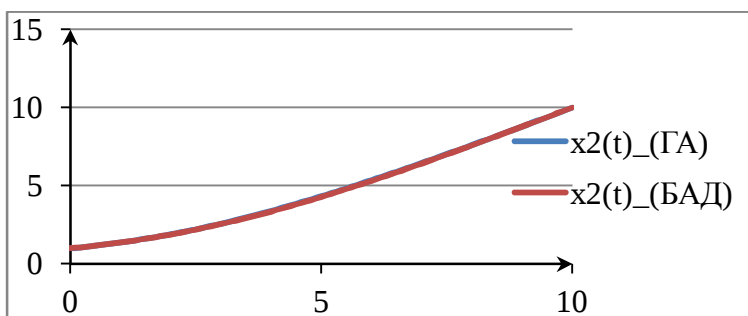


Рис. 2.19. График функции $\bar{x}_2(t)$ для метода БАД и генетического алгоритма

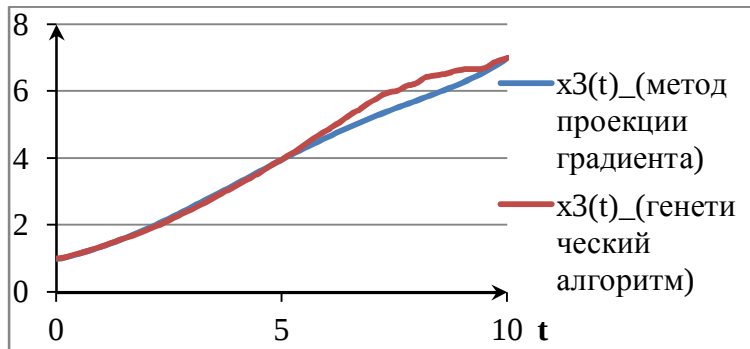


Рис. 2.20. График функции $\bar{x}_3(t)$ для метода БАД и генетического алгоритма

i	$ A_i - X_i(T) $ (МПГ)	$ A_i - X_i(T) $ (ГА)
1	5,758E-08	0,00045433
2	6,43269E-07	0,001067946
3	4,11908E-08	0,002101344
I_{opt}	4,18807E-13	5,76257E-06

МПГ – метод проекции градиента;

ГА – генетический алгоритм.

Особенностью генетического алгоритма является то, что он обрабатывает не значения параметров самой задачи, а их закодированную форму, и применяет вероятностные, а не детерминированные правила выбора. Преимуществом генетического алгоритма является то, что он использует только целевую функцию, а не ее производные либо иную дополнительную информацию. Генетический алгоритм может быть использован, когда не работает метод проекции градиента, например, когда функция не дифференцируема, или множество допустимых значений управления не компактное. В частности в поставленной задаче множество допустимых значений управления выбиралось дискретным: $\omega = \{0,1\}$. Для сравнения генетического алгоритма с алгоритмом метода штрафных функций было выбрано множество $\{0, 0,0142857, 0,0285714, 0,0428571, 0,0571428, 0,0714285, 0,0857142, 0,1\}$. Метод проекции градиента работает быстрее и с большей точностью, но при этом требует большего объема исходной информации о задаче.

§5. Задача обучения искусственной нейронной сети как задача оптимального управления с нефиксированным временем

Рассмотрим задачу обучения искусственной нейронной сети (2.2)-(2.4) как задачу оптимального управления с нефиксированным временем процесса.

Задача оптимального управления заключается в минимизации функционала:

$$J(\omega, u) = \int_0^T E(x(t), \omega(t), u(t)) dt + \Phi(x(T)) + T \quad (2.19)$$

где первое слагаемое характеризует энергию рассматриваемой нейронной сети, зависящую от текущей конфигурации сети, второе слагаемое характеризует ошибку обучения нейронной сети.

Задача оптимального управления заключается в том, чтобы обучить нейронную сеть таким образом, чтобы минимизировать функционал (2.19) и построить оптимальные значения весовых коэффициентов и внешнего управляющего воздействия.

Для решения задачи (2.2)-(2.4), (2.19) перейдем от задачи оптимального управления с нефиксированным временем процесса к задаче с фиксированным временем процесса. Для этого осуществим следующую параметризацию: $t(\tau) = \xi\tau$, $\tau \in [0, T_0]$, $t(T_0) = \xi T_0 = T$, введем обозначения $x(t(\tau)) = \tilde{x}(\tau)$, $\omega(t(\tau)) = \tilde{\omega}(\tau)$, $u(t(\tau)) = \tilde{u}(\tau)$ и введем новые переменные

$$\tilde{y}_i(\tau) = \xi \int_{\tau-r}^{\tau} \sum_{j=1}^n \tilde{\omega}_{ij}(\tilde{z}) \tilde{x}_j(\tilde{z} - \delta_j) d\tilde{z}, \quad i = \overline{1, n},$$

В этих обозначениях задача оптимального управления примет вид (2.20)-(2.23):

$$\begin{aligned} J(\tilde{x}(\tau), \tilde{\omega}(\tau), \tilde{u}(\tau), t(\tau)) = \\ = \int_0^{T_0} \xi E(\tilde{x}(\tau), \tilde{\omega}(\tau), \tilde{u}(\tau)) d\tau + \Phi(\tilde{x}(T_0)) + t(T_0) \end{aligned} \quad (2.20)$$

$$\dot{\tilde{x}}_i(\tau) = -\xi \beta_i \tilde{x}_i(\tau) + \xi f_i(\tilde{y}_i(\tau)) + \xi \gamma_i \tilde{u}_i(\tau), \quad i = \overline{1, n} \quad (2.21)$$

$$\dot{\tilde{y}}_i(\tau) = \xi \sum_{j=1}^n (\tilde{\omega}_{ij}(\tau) \tilde{x}_j(\tau - \delta_j) - \tilde{\omega}_{ij}(\tau - r) \tilde{x}_j(\tau - r - \delta_j)), \quad i = \overline{1, n}, \quad (2.22)$$

$$\dot{t}(\tau) = \xi \quad (2.23)$$

Пусть $\tilde{\omega}_{ij}(\tau), i, j = \overline{1, n}$ - фиксированные весовые коэффициенты, $\tilde{u}_i(\tau), i = \overline{1, n}$ - оптимальные управляющие внешние воздействия, $\bar{\xi}$ - оптимальное значение параметра, функция $E(\tilde{x}, \tilde{\omega}, \tilde{u})$ выпукла по $\tilde{u}_i$, тогда оптимальные значения $\tilde{u}_i(\tau), i = \overline{1, n}$ находятся из соотношения (2.24):

$$\tilde{u}_i(\tau) = \begin{cases} a_i, & \Psi_i(\tau) > a_i \\ \Psi_i(\tau), & |\Psi_i(\tau)| \leq a_i \\ -a_i, & \Psi_i(\tau) < -a_i \end{cases} \quad \text{где } \Psi_i(\tau) = \frac{p_i(\tau)\gamma_i}{\lambda_0 \frac{\partial E(\tilde{x}, \tilde{\omega}, \tilde{u})}{\partial \tilde{u}_i}}, \quad i = \overline{1, n}, \quad (2.24)$$

а сопряженные функции $p_i(\tau)$ удовлетворяют системе дифференциальных уравнений с запаздывающим аргументом (2.25):

$$\begin{aligned} \dot{p}_i(\tau) = & \xi p_i(\tau) \beta_i - \xi \sum_{j=1}^n (r_j(\tau + \delta_j) \tilde{\omega}_{ji}(\tau + \delta_j) - \\ & - r_j(\tau + r + \delta_j) \tilde{\omega}_{ji}(\tau + \delta_j)), \end{aligned} \quad (2.25)$$

$$\dot{r}_i(\tau) = -\xi p_i(\tau) \frac{\partial f(\tilde{y}_i)}{\partial \tilde{y}_i}, \quad \dot{s}(\tau) = 0, \quad i = \overline{1, n}$$

и условиями трансверсальности на правом конце (2.26):

$$\begin{aligned} p_i(T_0) = & -\lambda_0 \frac{\partial \Phi(\tilde{x}_i(T_0))}{\partial \tilde{x}_i(T_0)}, \quad p_i(\tau) = 0, \tau > T_0, \\ r_i(T_0) = & 0, \quad s(T_0) = -\lambda_0, \quad i = \overline{1, n} \end{aligned} \quad (2.26)$$

Для определения оптимального решения рассмотрим дискретную аппроксимацию начальной задачи. Вводя следующие обозначения $\Delta\tau = T_0 / q$, $\tau_k = k\Delta\tau$, $t^q = T_0$, $\tilde{x}_i^k = \tilde{x}_i(\tau_k)$, $\tilde{y}_i^k = \tilde{y}_i(\tau_k)$, $\tilde{\omega}_{ij}^k = \tilde{\omega}_{ij}(\tau_k)$, $\tilde{u}_i^k = \tilde{u}_i(\tau_k)$, $i, j = \overline{1, n}, k = \overline{0, q-1}$, построим дискретную задачу оптимального управления, которая состоит в минимизации функции (2.27):

$$I(t, \tilde{x}, \tilde{y}, \tilde{\omega}, \tilde{u}, \xi) = \Delta\tau \xi \sum_{k=1}^{q-1} E(\tilde{x}^k, \tilde{\omega}^k, \tilde{u}^k) + \Phi(\tilde{x}^q) + t^q, \quad (2.27)$$

при ограничениях

$$\tilde{x}_i^{k+1} = \tilde{x}_i^k - \Delta\tau\xi\beta_i\tilde{x}_i^k + \Delta\tau\xi f_i(\tilde{y}_i^k) + \Delta\tau\xi\gamma_i\tilde{u}_i^k, \quad (2.28)$$

$$\tilde{y}_i^{k+1} = \tilde{y}_i^k + \Delta\tau\xi \sum_{j=1}^n (\tilde{\omega}_{ij}^k \tilde{x}_j^{k-\nu j} - \tilde{\omega}_{ij}^{k-\nu} \tilde{x}_j^{k-\nu-\nu j}),$$

$$t^{k+1} = t^k + \Delta\tau\xi, \quad i = \overline{1, n}, k = \overline{0, q-1}$$

На функции управления наложены ограничения

$$|\omega_{ij}^k| \leq a_{ij}, \omega_{ii}^k \equiv 0, |u_i^k| \leq a_i, \quad i, j = \overline{1, n}, k = \overline{0, q-1}, \quad (2.29)$$

Для решения задачи (2.27)-(2.29) применим метод быстрого автоматического дифференцирования, в соответствии с которым введем множители Лагранжа p , r и s , тогда функция Лагранжа для данной задачи может быть представлена следующим выражением:

$$\begin{aligned} L(t, \tilde{x}, \tilde{y}, \tilde{\omega}, \tilde{u}, \xi, r, p, s, \lambda_0) = & \lambda_0 (\Delta\tau\xi \sum_{k=1}^{q-1} E(\tilde{x}^k, \tilde{\omega}^k, \tilde{u}^k) + \Phi(\tilde{x}^q) + t^q) \\ & + \sum_{k=0}^{q-1} \sum_{i=1}^n p_i^{k+1} (\tilde{x}_i^{k+1} - \tilde{x}_i^k + \Delta\tau\xi\beta_i\tilde{x}_i^k - \Delta\tau\xi f_i(\tilde{y}_i^k) - \Delta\tau\xi\gamma_i\tilde{u}_i^k) + \\ & \sum_{k=0}^{q-1} \sum_{i=1}^n r_i^{k+1} (\tilde{y}_i^{k+1} - \tilde{y}_i^k - \Delta\tau\xi \sum_{j=1}^n (\tilde{\omega}_{ij}^k \tilde{x}_j^{k-\nu j} - \tilde{\omega}_{ij}^{k-\nu} \tilde{x}_j^{k-\nu-\nu j})) + \\ & + \sum_{k=0}^{q-1} s^{k+1} (t^{k+1} - t^k - \Delta\tau\xi) \end{aligned}$$

Тогда градиент минимизируемой функции запишется следующим образом:

$$\frac{\partial L}{\partial \tilde{u}_l^m} = \lambda_0 \Delta\tau\xi \frac{\partial E(\tilde{x}^m, \tilde{\omega}^m, \tilde{u}^m)}{\partial \tilde{u}_l^m} - \Delta\tau\xi \gamma_l p_l^{m+1}, \quad (2.30)$$

$$\begin{aligned} \frac{\partial L}{\partial \tilde{\omega}_{lj}^m} = & \lambda_0 \Delta\tau\xi \frac{\partial E(\tilde{x}^m, \tilde{\omega}^m, \tilde{u}^m)}{\partial \tilde{\omega}_{lj}^m} - \Delta\tau\xi (r_l^{m+1} \tilde{x}_j^{m-\nu j} - r_l^{m+1+\nu} \tilde{x}_j^{m-\nu j}), \\ (2.31) \end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial \xi} = & \lambda_0 \Delta \tau \sum_{k=1}^{q-1} E(\tilde{x}^k, \tilde{\omega}^k, \tilde{u}^k) + \sum_{k=0}^{q-1} \sum_{i=1}^n p_i^{k+1} (\Delta \tau \beta_i \tilde{x}_i^k - \Delta \tau f_i(\tilde{y}_i^k)) - \\
& \Delta \tau \gamma_i \tilde{u}_i^k + \sum_{k=0}^{q-1} \sum_{i=1}^n r_i^{k+1} (\Delta \tau \sum_{j=1}^n (\tilde{\omega}_{ij}^k \tilde{x}_j^{k-\nu j} - \tilde{\omega}_{ij}^{k-\nu} \tilde{x}_j^{k-\nu-\nu j})) - \\
& - \sum_{k=0}^{q-1} s^{k+1} \Delta \tau
\end{aligned}
\tag{2.32}$$

где

$$\begin{aligned}
p_l^m = & p_l^{m+1} - \Delta \tau \xi p_l^{m+1} \beta_l + \\
& + \Delta \tau \xi \sum_{j=1}^n r_j^{m+1+\nu j} \tilde{\omega}_{jl}^{m+\nu j} - r_j^{m+1+\nu j+\nu} \tilde{\omega}_{jl}^{m+\nu j},
\end{aligned}
\tag{2.33}$$

$$p_l^q = -\lambda_0 \frac{\partial \Phi(\tilde{x}_l^q)}{\partial \tilde{x}_l^q}, \quad p_l^m = 0, \quad m > q
\tag{2.34}$$

$$r_l^m = \Delta \tau \xi p_l^{m+1} \frac{\partial f_l(\tilde{y}_l^m)}{\partial \tilde{y}_l^m} + r_l^{m+1}, \quad r_l^q = 0,
\tag{2.35}$$

$$s^m = s^{m+1}, \quad s^q = -\lambda_0
\tag{2.36}$$

$$l, j = \overline{1, n}, \quad k, m = \overline{0, q-1}$$

Предлагаемый подход позволяет вычислить оптимальные значения весовых коэффициентов и внешнего управляющего воздействия. При этом основное вычислительное время при расчете градиента по формулам (5.30)-(5.32) требуется на нахождение векторов p и r , для этого необходимо решить 2 системы уравнений с $q \cdot (n^2 + n)$ неизвестными.

Глава 3. Методы сведения многокритериальной задачи к задаче с одним критерием и программная реализация разработанных алгоритмов

§1. Общие сведения о методах сведения исходной многокритериальной задачи к задачам с единым критерием

Почти любая сложная практическая задача принятия решений является многокритериальной. В связи с этим в настоящее время особое значение приобретает теория принятия решений при наличии многих критериев.

Одним из фундаментальных понятий в этой теории является понятие оптимальности по Парето, или эффективности решения, которое представляет собой обобщение понятия точки максимума числовой функции на случай нескольких функций. Основная идея теории заключается в том, что решение является Парето - оптимальным, если значение любого из критериев можно улучшить, лишь за счет ухудшения значений остальных критериев. Термин "Парето-оптимальность" связан с именем итальянского экономиста и социолога В. Парето (1848 - 1923 гг.), который одним из первых начал использовать его при математических исследованиях процесса рыночного обмена товаров.

Вопросы о свойствах множества Парето и методов отыскания Парето-оптимальных решений возникают в теории игр, математической экономике, теории статистических решений, исследовании операций, теории оптимального управления и в других научных дисциплинах, в которых изучаются многокритериальные модели принятия рациональных решений.

Концепция принятия решения в качестве элемента деятельности рассматривает решение как сознательный выбор одной из ряда альтернатив. Этот выбор производит некоторое лицо, принимающее решение и стремящееся к достижению определенных целей. В роли такого лица выступает человек или группа людей. Если при принятии решений используется применение математических методов, то это предполагает построение подходящей математической модели, формально представляющей проблемную ситуацию, т.е.

ситуацию выбора решений. Для задач принятия решений (задач оптимизации) в условиях определённости, когда отсутствуют случайные или неопределённые факты, компонентами такой модели являются элементы множества всех альтернативных решений. Из данного множества необходимо выбрать один элемент, который будет являться наилучшим. Такой элемент называют оптимальным решением.

В многокритериальной задаче оптимизации сравнение решений по предпочтительности осуществляется при помощи числовых функций $f_1, f_2, \dots, f_m$. Данные функции называются критериями или показателями качества эффективности.

Общая многокритериальная задача ставится следующим образом: требуется минимизировать функции $f_i(x)$ на заданном множестве D , $D = \{x \in R^n : g_i(x) \leq 0, i = 1, k, g_i(x) = 0, i = m + 1, s\}$. Множество D называется множеством допустимых значений.

Существенное отличие этой задачи от традиционной однокритериальной состоит в понятии оптимальности. В однокритериальной задаче под оптимальным понимается решение, обеспечивающее максимальное значение критерия. При многих критериях увеличение одних критериев приводит к уменьшению других (редкие исключения не представляют практического интереса) и поэтому понятие оптимальности требует принципиальных уточнений. Очевидно, что без дополнительной информации о предпочтениях ЛПР (Лицо, принимающее решение) бессмысленно говорить об оптимальном решении и тем более формализовано искать его.

Допустимое множество D строится в n -мерном пространстве переменных. Каждое решение $X \in D$ полностью характеризуется соответствующими значениями всех частных критериев, т.е. вектором Y . Числовое m -мерное пространство E^m , координатами которого являются $y_i = f_i(X)$, называется критериальным пространством. Очевидно, что каждому X можно поставить в соответствие точку в критериальном пространстве. Если же решение X

допустимо, то соответствующая точка в E^m , определяемая вектором Y , является достижимой. Множество таких точек в критериальном пространстве называется множеством достижимости (достижимых векторов). Таким образом, векторная функция $f(X)$ отображает допустимое множество D на множестве достижимости G :

$$G = \{Y \in E^m \mid Y = f(X), X \in D\}$$

и задача состоит в выборе вектора из этого множества, наилучшего с точки зрения ЛПР.

В общем случае построение множества G для реальных задач весьма проблематично, но для задач с «хорошими» свойствами, например, линейных, множество достижимости может быть построено.

В зависимости от того, на какой стадии процесса выявляются и используются предпочтения ЛПР, можно выделить три группы многокритериальных методов принятия решений:

- 1) основаны на том, что ЛПР может выразить свои предпочтения до начала процесса многокритериальной оптимизации;
- 2) интерактивные (диалоговые) методы;
- 3) методы построения множества эффективных решений с последующим представлением его ЛПР.

В методах первой группы используются различные способы свёртки критериев, лексикографическое упорядочение критериев, установление желаемых уровней критериев и др. Вторая группа методов основана на непосредственном участии ЛПР в процессе оптимизации, когда на каждой итерации компьютер предлагает решения, а ЛПР их оценивает, и с учётом этих оценок компьютер ищет новые решения. Методы третьей группы отличаются друг от друга различными способами построения и представления множества эффективных решений.

§2. Практическая реализация методов сведения исходной многокритериальной задачи к задачам с единым критерием

2.1 Метод линейной свертки

Наиболее простой и часто применяемый метод формирования единой целевой функции основан на линейной свертке всех частных критериев в один критерий:

$$J(x(\cdot), u(\cdot)) = L_1 J_1(x(\cdot), u(\cdot)) + L_2 J_2(x(\cdot), u(\cdot)) + \\ + L_3 J_3(x(\cdot), u(\cdot)) + L_4 J_4(x(\cdot), u(\cdot)), \quad (3.1)$$

в котором весовые коэффициенты $L_i, i=1,4$, рассматриваются как показатели относительной значимости отдельных критериев.

Построено оптимальное управление в задаче (2.2)-(2.5) при различных значениях весовых коэффициентах L_4 .

На рисунке 3.1 представлены значения минимизируемых функционалов, $I_{1-opt} = - \int_0^T \sum_{i=1}^2 [l_i(x_i)x_i - C_i(x_i)u_i] dt$, $I_{4-opt} = \sum_{i=1}^2 (\max(A_i - x_i(T), 0))^2$, в зависимости от значения штрафного коэффициента L_4 , который изменяется в диапазоне от 10 до 5000, и запаздывания r при следующих параметрах системы:

$R_i = 0$, $e_1 = 0.75$, $c_{11} = 0$, $d_{11} = 0.1$, $\alpha_1 = 0.75$, $\gamma_{11} = 0.05$, $a_{11} = 0.0375$, $b_{11} = 0.0375$,
 $A_1 = 7,4$; $A_2 = 13$; $B_1 = 11$; $B_2 = 17$; $C_1 = 6,5$; $C_2 = 5$; $0 \leq u_i \leq 0.1, i=1,2$; $L_2 = L_3 = 0$;
 $T=20$; $q=2000$; точность метода $\varepsilon = 0,00000001$:

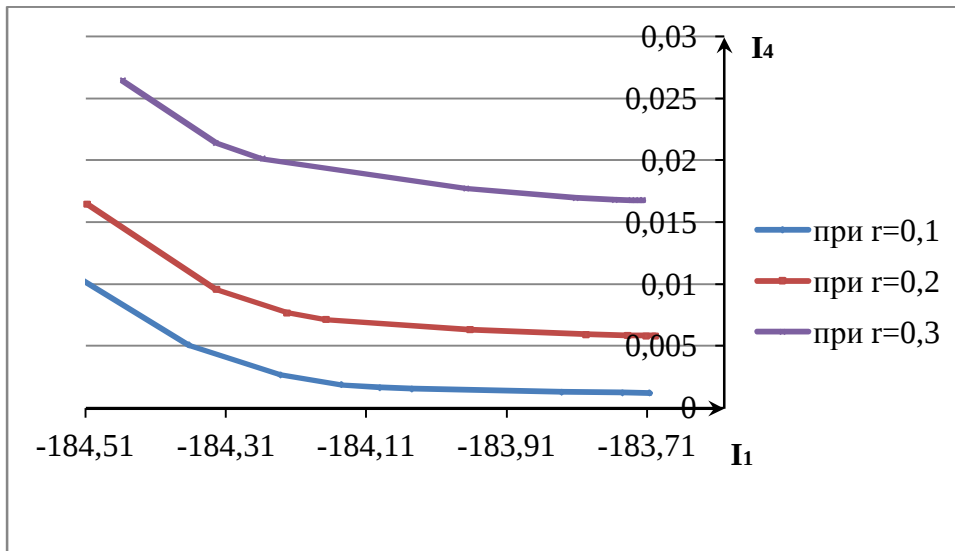


Рис.3.1 Значения минимизируемых функционалов I_{4_opt} , I_{1_opt}

При увеличении значения штрафного коэффициента L_4 значение функционала I_{1_opt} увеличивается, а значение функционала I_{4_opt} уменьшается, что соответствует более точному выполнению граничного условия $x_i(T) \geq A_i$, $i = 1, 2$ за счет уменьшения прибыли от реализации продукции.

2.2 Свертка с неотрицательными весовыми коэффициентами

Используя данный метод необходимо решить последовательность однокритериальных задач, т.е. найти минимальные и максимальные значения, которые может принимать на допустимом множестве каждая целевая функция.

Тогда исходная задача сведется к решению задачи (2.2)-(2.5) с целевым функционалом вида:

$$J(x(\cdot), u(\cdot)) = \sum_{i=1}^k \alpha_i \frac{J_i(x(\cdot), u(\cdot)) - J_i^{\min}}{J_i^{\max} - J_i^{\min}} \quad (3. 2)$$

где α_i – положительные числа.

В таблице 3.1 представлены значения минимизируемых функционалов,

$$I_{1_opt} = - \int_0^T \sum_{i=1}^2 [l_i(x_i(t))x_i(t) - C_i(x_i(t))u_i(t)] dt, \quad I_{2_opt} = \int_0^T \sum_{i=1}^2 (\max(C_i - x_i(t), 0))^2 dt,$$

$$I_{3\text{-opt}} = \int_0^T \sum_{i=1}^2 (\max(x_i(t) - B_i, 0))^2 dt, \quad I_{4\text{-opt}} = \sum_{i=1}^2 (\max(A_i - x_i(T), 0))^2$$

в зависимости от значения штрафных коэффициентов L_4 при следующих параметрах системы:

$R_i = 0$, $e_1 = 0.75$, $c_{11} = 0$, $d_{11} = 0.1$, $\alpha_1 = 0.75$, $\gamma_{11} = 0.05$, $a_{11} = 0.0375$,
 $b_{11} = 0.0375$, $A_1 = 7$; $A_2 = 12,9$; $B_1 = 11$; $B_2 = 17$; $C_1 = 6,2$; $C_2 = 1$; $0 \leq u_i \leq 0.1, i = 1,2$;
 $r = 0.1$; $T=20$; $q=2000$; точность метода $\varepsilon = 0,00000001$:

L_4	$I_{1\text{-opt}}$	$I_{2\text{-opt}}$	$I_{3\text{-opt}}$	$I_{4\text{-opt}}$
20	1,237784	0,556341	0,21715	0,013641
40	1,333807	0,558513	0,217426	0,009429
80	1,393755	0,56351	0,218095	0,007435
160	1,428922	0,573354	0,219912	0,006157
320	1,447056	0,58998	0,224552	0,005183
640	1,454009	0,615677	0,235829	0,004382
1280	1,451603	0,651913	0,259158	0,00374
2560	1,431619	0,688178	0,289544	0,003374
5120	1,395539	0,708748	0,321103	0,003244

Табл.3.1 Значение минимизируемых функционалов при различных значениях L_4

Из таблицы 3.1 следует, что при увеличении штрафного коэффициента L_4 значения функционалов $I_{1\text{-opt}}$, $I_{2\text{-opt}}$, $I_{3\text{-opt}}$ увеличиваются на 13%, 42% и 52% соответственно, а значение функционала $I_{4\text{-opt}}$ уменьшается на 77%, что соответствует более точному выполнению граничного условия $x_i(T) \geq A_i$, $i = 1,2$ за счет уменьшения значений остальных критериев.

2.3. Принцип гарантированного результата

Основываясь на принципе гарантированного результата, построено оптимальное решение задачи (3.2)-(3.4) с целевым функционалом:

$$J(x, u) = \max_i \frac{J_i(x(\cdot), u(\cdot))}{J_i^{\max}}. \quad (3.3)$$

В таблице 3.2 представлены значения минимизируемых функционалов,

$$I_{1_opt} = - \int_0^T \sum_{i=1}^2 [l_i(x_i(t))x_i(t) - C_i(x_i(t))u_i(t)] dt, \quad I_{2_opt} = \int_0^T \sum_{i=1}^2 (\max(C_i - x_i(t), 0))^2 dt,$$

$$I_{3_opt} = \int_0^T \sum_{i=1}^2 (\max(x_i(t) - B_i, 0))^2 dt, \quad I_{4_opt} = \sum_{i=1}^2 (\max(A_i - x_i(T), 0))^2$$

в зависимости от значения штрафных коэффициентов L_4 при следующих параметрах системы:

$T=10$; $m = 1$, $n = 1$, $R_i = 0$, $e_1 = 0.75$, $c_{11} = 0$, $d_{11} = 0.1$, $\alpha_1 = 0.75$, $\gamma_{11} = 0.05$,
 $a_{11} = 0.0375$, $b_{11} = 0.0375$, $A_1 = 7.4$; $A_2 = 13$; $B_1 = 10$; $B_2 = 16$; $C_1 = 6.5$; $C_2 = 5$;
 $r = 0.1$; $0 \leq u_i \leq 0.1$, $i = 1, 2$, $\Delta t = 0.01$, точность метода $\varepsilon = 0.00000001$:

L_4	I_{1_opt}	I_{2_opt}	I_{3_opt}	I_{4_opt}
100	-93,699	0,029	0,398	1,5205
150	-93,72	0,03	0,405	1,502
200	-93,736	0,031	0,41	1,492
250	-93,749	0,031	0,414	1,485
300	-93,76	0,032	0,418	1,48
350	-93,769	0,032	0,421	1,477
400	-93,776	0,033	0,424	1,473
450	-93,784	0,033	0,428	1,47
500	-93,791	0,034	0,431	1,468
550	-93,796	0,034	0,433	1,467

600	-93,8	0,034	0,434	1,466
650	-93,806	0,034	0,436	1,465
700	-93,81	0,034	0,438	1,464
750	-93,814	0,034	0,439	1,464
800	-93,818	0,034	0,441	1,463
850	-93,822	0,034	0,442	1,463
900	-93,825	0,035	0,443	1,462
950	-93,827	0,035	0,444	1,462
1000	-93,83	0,035	0,445	1,462

Табл.3.2 Значение минимизируемых функционалов при различных значениях L_4

Из таблицы 3.2 следует, что при увеличении штрафного коэффициента L_4 значения I_{4_opt} уменьшается за счет увеличения значений остальных функционалов, что соответствует более точному выполнению граничного условия $x_i(T) \geq A_i$, $i = 1, 2$ за счет уменьшения значений остальных критериев.

Таким образом, разработанная программа позволяет находить оптимальное решения многокритериальной задачи при различных методах сведения многокритериальной задачи к задаче с единым критерием, а именно используются такие методы, как линейная свертка (метод 1), способ свертки с неотрицательными весовыми коэффициентами (метод 2), принцип гарантированного результата (метод 3), а также находит минимальные и максимальные значения минимизируемых функционалов и значения целевых функционалов в задаче без управления. Ниже в таблице 3.3 представлены минимальные, максимальные и оптимальные значения минимизируемых функционалов

$$I_{1_opt} = - \int_0^T \sum_{i=1}^2 [l_i(x_i(t))x_i(t) - C_i(x_i(t))u_i(t)] dt, \quad I_{2_opt} = \int_0^T \sum_{i=1}^2 (\max(C_i - x_i(t), 0))^2 dt,$$

$$I_{3_opt} = \int_0^T \sum_{i=1}^2 (\max(x_i(t) - B_i, 0))^2 dt, \quad I_{4_opt} = \sum_{i=1}^2 (\max(A_i - x_i(T), 0))^2$$

в зависимости от выбранного критерия оптимальности при следующих параметрах системы:

$T=10$; $m = 1$, $n = 1$, $R_i = 0$, $e_1 = 0.75$, $c_{11} = 0$, $d_{11} = 0.1$, $\alpha_1 = 0.75$, $\gamma_{11} = 0.05$,
 $a_{11} = 0.0375$, $b_{11} = 0.0375$, $A_1 = 7.4$; $A_2 = 13$; $B_1 = 10$; $B_2 = 16$; $C_1 = 6.5$; $C_2 = 5$;
 $r = 0.1$; $0 \leq u_i \leq 0.1$, $i = 1, 2$, $\Delta t = 0.01$, точность метода $\varepsilon = 0,00000001$:

	I_1	I_2	I_3	I_4
Значение фнкционалов в задаче без управления	-93,3139	0,142375	1,08935	2,472991
Минимальные значения функционалов	-94,3151	0,027103	0,38519	1,458864
Максимальные значения функционалов	-92,7587	0,390153	1,914227	4,678318
Оптимальные значения функционалов для метода 1	-93,7565	0,033461	0,420542	1,475194
Оптимальные значения функционалов для метода 2	-93,7079	0,029925	0,401174	1,510724
Оптимальные значения функционалов для метода 3	-93,6988	0,029469	0,39848	1,520453

Табл.3.3 минимальные, максимальные и оптимальные значения минимизируемых функционалов I_1 , I_2 , I_3 , I_4 .

Из таблицы 3.3 следует, что оптимальное решение, полученное при использовании метода линейной свертки (метод 1) соответствует более точному граничному условию, тогда как способ свертки с неотрицательными весовыми коэффициентами (метод 2) и принцип гарантированного результата (метод 3) находят оптимальные решения, соответствующие более точному выполнению условий $C_1 \leq x_1 \leq B_1$, $C_2 \leq y_1 \leq B_2$, которые отвечают за сохранность популяции жертв и хищников на заданных уровнях.

В предлагаемой работе разработаны численные методы и алгоритмы, позволяющие свести исходную многокритериальную задачу к задачам с единым критерием, зависящим от параметров L_i , в котором конкретное значение параметра выбирается экспертом. В многокритериальной задаче при увеличении значимости одного из критериев за счет увеличения соответствующего штрафного коэффициента значение соответствующего функционала уменьшается, значение остальных критериев увеличивается.

§3. Программная реализация разработанных алгоритмов

В процессе практической реализации численных методов были поставлены и решены следующие задачи:

1. Проектирование пользовательского интерфейса.
2. Разработка модульных тестов.
3. Программная реализация.

3.1 Проектирование пользовательского интерфейса

Основными требованиями к пользовательскому интерфейсу являются гибкость, простота и возможность введения всех исходных данных и отображения результатов работы алгоритма в форме, удобной для непосредственного восприятия пользователем программы: в виде числовых значений, таблиц и графиков. В настоящее время существует большое количество средств для проектирования пользовательского интерфейса: OmniGraffle, Gui design studio, FlairBuilder и др. В качестве основного инструмента была выбрана программа Balsamiq Mockups. Balsamiq Mockups - это программа для изготовления макетов пользовательских интерфейсов. Макеты, получаемые с помощью Balsamiq Mockups, относятся к так называемым макетам с низкой степенью детализации (low-fidelity mockups). Предполагается, что именно скорость создания макетов является ключевым преимуществом Balsamiq Mockups. Это обусловлено тем, что зачастую от менее удачного варианта психологически сложнее отказаться, если на

его создание было затронуто много усилий. Balsamiq Mockups предоставляет возможность экспорта макета в форматы png и pdf, а также в xml (можно использовать в качестве резервной копии или пересылки макета по почте в его минимальном размере). Является условно бесплатным программным обеспечением. Результат проектирования пользовательского интерфейса приведен на рисунке 4.1.



Рис. 4.1. Результат проектирования пользовательского интерфейса.

3.2. Разработка модульных тестов.

Сегодня все большую популярность приобретает test-driven development (TDD), техника разработки программного обеспечения, при которой сначала пишется тест на определенный функционал, а затем пишется реализация этого функционала. Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны.

Преимущества модульного тестирования:

- Поощрение изменений. Модульное тестирование позже позволяет проводить рефакторинг, будучи уверенными, что модуль по-прежнему работает корректно (регрессионное тестирование). Это поощряет программистов к изменениям кода, поскольку достаточно легко проверить, что код работает и после изменений.
- Упрощение интеграции. Модульное тестирование помогает устранить сомнения по поводу отдельных модулей и может быть использовано для подхода к тестированию «снизу вверх»: сначала тестируются отдельные части программы, затем программа в целом.
- Документирование кода. Модульные тесты можно рассматривать как «живой документ» для тестируемого класса. Клиенты, которые не знают, как использовать данный класс, могут использовать юнит-тест в качестве примера.
- Отделение интерфейса от реализации. Поскольку некоторые классы могут использовать другие классы, тестирование отдельного класса часто распространяется на связанные с ним классы.

Для получения выгоды от модульного тестирования требуется строго следовать технологии тестирования на всём протяжении процесса разработки программного обеспечения. Нужно хранить не только записи обо всех проведённых тестах, но и обо всех изменениях исходного кода во всех модулях. С этой целью следует использовать систему контроля версий программного обеспечения. Таким образом, если более поздняя версия программного обеспечения не проходит тест, который был успешно пройден ранее, будет несложным сверить варианты исходного кода и устранить ошибку. Также необходимо убедиться в неизменном отслеживании и анализе неудачных тестов. Игнорирование этого требования приведёт к лавинообразному увеличению неудачных тестовых результатов.

3.3. Компьютерные программы для статистической обработки данных

Разработанная программа позволяет вводить все параметры вручную, однако параметры модели могут быть получены и из внешнего источника: файла, базы данных.

При этом коэффициенты могут быть вычислены по имеющимся статистическим данным. Для расчета коэффициентов можно воспользоваться методами эконометрики, которые позволяют получить коэффициенты, или существующими пакетами обработки статистических данных.

На сегодняшний день на рынке представлено около тысячи компьютерных программ для статистической обработки данных (далее – статистические пакеты). Разнообразие статистических пакетов обусловлено многоплановостью задач обработки данных с применением различных типов статистических процедур анализа для поиска ответов на вопросы из различных областей человеческой деятельности.

Перед пользователями различных категорий встает вопрос выбора оптимального статистического пакета для поиска верных ответов на существующие вопросы. При выборе пакета учитываются следующие параметры:

соответствие характеру решаемых задач;

объем обрабатываемых данных;

требования, предъявляемые к квалификации пользователя (уровень знаний в области статистики).

Статистические пакеты по признаку функциональности могут быть разделены на 3 основные группы:

Универсальные пакеты, или пакеты общего назначения (например, SPSS, STATA, STATISTICA, S-PLUS, Stadia, STATGRAPHICS, SYSTAT, Minitab).

Эти пакеты не ориентированы на специфическую предметную область и могут применяться для анализа данных из различных областей деятельности. Как правило, они предлагают широкий диапазон статистических методов и имеют относительно простой интерфейс. Многопрофильность универсального пакета

позволяет провести пробный анализ различных типов данных с использованием широкого диапазона статистических методов. Большинство существующих универсальных пакетов имеют много пересечений по составу встроенных статистических процедур.

Профессиональные пакеты (например, SAS, BMDP).

Профессиональные пакеты отличаются от универсальных тем, что позволяют работать со сверхбольшими объемами данных, применять узкоспециализированные методы анализа, создавать собственную систему обработки данных. Подготовленным пользователям работа с профессиональным пакетом предоставит больше возможностей для глубокого и детального анализа данных, построения сложных моделей и адаптации системы к собственным потребностям.

Специализированные пакеты (например, BioStat, MESOSAUR, DATASCOPE).

В некоторых областях деятельности анализируемые данные настолько специфичны, что к ним следует применять особые методы статистического анализа, как правило, не представленные в универсальных пакетах.

Специализированные пакеты позволяют проводить анализ с использованием ограниченного числа специализированных статистических методов или применимы к использованию для решения вопросов, относящихся к отдельно взятой предметной области. Как правило, с подобными статистическими пакетами работают специалисты, хорошо знакомые с методами анализа данных в той области, на которую ориентирован пакет. Так, статистический пакет BioStat создан для анализа данных в области биологии и медицины и будет подробнее рассмотрен ниже. Российский статистический пакет MESOSAUR специализируется на анализе одномерных и многомерных временных рядов и построении регрессионных моделей. Еще один российский статистический пакет DATASCOPE специализируется на проведении анализа многомерных данных.

Необходимо отметить, что существует минимальный набор статистических методов анализа, который включен во все рассмотренные пакеты:

описательная статистика (базовые статистические методы, проверка нормальности распределения данных);

дисперсионный анализ;

непараметрическая статистика (анализ таблиц сопряженности, непараметрические сравнения, дисперсионный анализ);

контроль качества;

анализ выживаемости;

кластерный анализ;

факторный анализ;

дискриминантный анализ;

регрессионный анализ;

обработка данных (сортировка, отбор, трансформация данных).

3.4. Программная реализация разработанных алгоритмов

Для программной реализации алгоритма был выбран язык Java. Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Основной особенностью языка является то, что программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Достоинство подобного способа выполнения программ — в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы

(например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят то, что исполнение байт-кода виртуальной машиной может снижать производительность программ и алгоритмов, реализованных на языке Java. В последнее время был внесен ряд усовершенствований, которые несколько увеличили скорость выполнения программ на Java:

- применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде,

- широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,

- аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами фирмы ARM).

Для реализации модульного тестирования была использована бесплатная библиотека jUnit версии 4. JUnit — библиотека для модульного тестирования программного обеспечения на языке Java. Были разработаны модульные тесты, прохождение которых является необходимым условием корректной работы разрабатываемого алгоритма.

Разработанный программный продукт имеет следующие особенности:

- Кроссплатформенность. Поскольку языком разработки программного обеспечения является Java, разработанный программный продукт может работать на любой операционной системе, для которой существует виртуальная java-машина. Пользовательский интерфейс разработан с использованием библиотеки Swing, которая является также кроссплатформенной и позволяет создавать привычный для пользователя интерфейс.

- Многопоточность. Разработанный алгоритм позволяет проводить вычисления параллельно и воспользоваться встроенной поддержкой

многопоточности в Java. Таким образом, программный продукт максимально использует возможности современных многоядерных мультипроцессорных систем.

- Формирование отчетов по проделанной работе. Разработанный программный продукт позволяет экспортировать параметры задачи и результаты вычислений в виде числовых значений таблиц и графиков по заданному шаблону в виде документа общепринятого формата.

- Бесплатное программное обеспечение. Программный продукт был разработан с использованием бесплатного и свободно распространяемого программного обеспечения.

На основе построенного алгоритма был разработан программный продукт, который позволяет строить оптимальное управление при различных параметрах системы. Программа позволяет вводить все необходимые параметры задачи и метода. По описанным выше алгоритмам программа находит оптимальное решение системы, строит графики численности популяций и соответствующие им графики управления, выводит минимальные, максимальные и оптимальные значения минимизируемых функционалов в зависимости от выбранного критерия оптимальности, а также позволяет сформировать отчет, экспортируя параметры задачи и результаты вычислений в виде числовых значений, таблиц и графиков по заданному шаблону в отдельный документ для проведения дополнительного анализа.

Заключение

В предлагаемой работе в первой главе рассмотрена задача оптимального управления для модели хищник-жертва, описываемой системой интегродифференциальных уравнений. Сформулирован принцип максимума для исходной непрерывной задачи.

Для нахождения численного решения задачи применен метод быстрого автоматического дифференцирования и генетический алгоритм.

Проведен анализ влияния параметров методов и задачи на оптимальное управление в модели хищник-жертва (1.1)-(1.5).

Показано, что полученные численными методами решения согласуются с принципом максимума для исходной непрерывной задачи.

Проведено сравнение метода быстрого автоматического дифференцирования и генетического алгоритма, в частности, показано, что метод БАД работает быстрее, но при этом требует большего объема исходной информации о задаче. Особенностью генетического алгоритма является то, что он обрабатывает не значения параметров самой задачи, а их закодированную форму, и применяет вероятностные, а не детерминированные правила выбора. Преимуществом генетического алгоритма является то, что он прост в реализации, использует только целевую функцию, а не ее производные либо иную дополнительную информацию. Генетический алгоритм может быть использован, когда не работает метод проекции градиента, например, когда функция не дифференцируема, или множество допустимых значений управления не компактное. В частности в поставленной задаче множество допустимых значений управления выбиралось дискретным: $u_i^l = \{0, 0.1\}$, $v_j^l = \{0, 0.1\}$.

Во второй главе рассмотрена задача оптимального управления для модели искусственной нейронной сети, описываемой системой интегродифференциальных уравнений.

Сформулирована и решена задача обучения искусственной нейронной сети как задача оптимального управления с нефиксированным временем процесса.

Определены необходимые условия оптимальности для построения весовых коэффициентов нейронной сети с учетом заданных ограничений и вида функционала.

Разработаны численные методы обучения нейронной сети на основе метода быстрого автоматического дифференцирования и генетического алгоритма.

Проведен анализ влияния параметров методов и задачи обучения искусственной нейронной сети на оптимальное решение. Проведено сравнение результатов, полученных методом быстрого автоматического дифференцирования и генетическим алгоритмом.

В третьей главе разработаны численные методы, позволяющие свести исходную многокритериальную задачу к задаче с единым критерием. Оптимальное управление построено для различных типов функционалов в зависимости от цели управления, которая может заключаться в переводе системы из одного заданного состояния системы в другое, уменьшении затрат, связанных с функционированием системы, или получение максимальной прибыли.

Список литературы

1. Авдонин С.А., Горшкова О.Я. Об управляемости и квазиуправляемости систем параболического типа с запаздыванием // Дифференциальные уравнения в частных производных. Л., 1986. С.53-55.
2. Авдонин С.А., Горшкова О.Я. Управляемость многомерных параболических систем с запаздыванием. Ленингр. гос. пед. инт-т. Л., 1987. 25с. Деп. в ВИНТИ 14.08.87. N 5989.
3. Авдонин С.А., Иванов С.А. Управляемость систем с распределенными параметрами и семейства экспонент. Киев: УМК ВО, 1989. 244с.
4. Азбелев Н.В., Максимов В.П., Рахматуллина Л.Ф. Введение в теорию функционально-дифференциальных уравнений. М.:Наука, 1991. 280с.
5. Айда-Заде К. Р., Евтушенко Ю. Г. Быстрое автоматическое дифференцирование на ЭВМ // Матем. моделирование. 1989. Т. 1. С. 121-139.
6. Андреева Е. А. Мазурова И.С. Оптимальное управление в модели хищник-жертва с распределенным запаздыванием // Труды X Международной научно-технической конференции "Новые информационные технологии системы", Пенза, Издательство ПГУ, 2012, - С.256-259
7. Андреева Е. А. Оптимальное управление динамическими системами. Тверь: ТвГУ, 1999.
8. Андреева Е. А. Оптимизация искусственной нейронной сети// Применение функционального анализа в теории приближений: Сб. научн. тр. Тверь: ТвГУ, 1996. С. 7-12.
9. Андреева Е. А., Джеед М. Оптимальное управление системами, описываемыми интегральными и интегродифференциальными уравнениями, - Тверь, 2003.
10. Андреева Е. А., Оптимизация нейронных сетей, - Тверь: 2008.
11. Андреева Е. А., Пустарнакова Ю. А. Оптимизация нейронной сети с запаздыванием. Ч. 1. Применение функционального анализа в теории приближений: Сб. научн. тр. Тверь: ТвГУ, 2000. С. 14-30.

12. Андреева Е. А., Пустарнакова Ю. А. Численный метод обучения искусственных нейронных сетей с учётом запаздывания // Журнал вычислительной математики и математической физики. 2002. Т. 42 №9. С.1436–1444.

13. Андреева Е.А. Выбор оптимальных весовых функций в модели нейронной сети с запаздыванием // Применение функционального анализа в теории приближений. Тверь, 1996.

14. Андреева Е.А., Большакова И.С. Оптимальное управление в модели хищник-жертва с учетом сосредоточенного и распределенного запаздывания // Математические методы управления: сб. науч. тр. – Тверь: Твер. гос. ун-т, 2011,- С.5-15.

15. Андреева Е.А., Большакова И.С. Оптимальное управление динамическими системами с запаздыванием // Современные методы теории краевых задач. Материалы Воронежской весенней математической школы “Понтрягинские чтения – XXI”. – Воронеж: Издательско-полиграф. центр Воронежского гос. ун-та, 2011. – С. 9–10.

16. Андреева Е.А., Большакова И.С. Численные методы построения оптимального управления системами, описываемыми интегродифференциальными уравнениями // Материалы Международной Научно-практической конференции «Информатизация как целевая ориентация и стратегический ресурс образования». Архангельск, 2012, - С.461-468.

17. Андреева Е.А., Евтушенко Ю.Г. Численные методы решения задач оптимального управления для систем, описываемых интегродифференциальными уравнениями типа Фредгольма// Модели и методы оптимизации. 1989. №1. С. 4-13.

18. Андреева Е.А., Колмановский В.Б., Шайхет Л.Е. Управление системами с последствием. М.: Наука, 1992.

19. Андреева Е.А., Мазурова И.С. Оптимальное управление динамическими системами в многокритериальных задачах // Материалы всероссийской научной

конференции с международным участием «Спектральная теория операторов и её приложения». Архангельск, 2012.

20. Андреева Е.А., Мазурова И.С. Оптимальное управление динамическими системами в многокритериальных задачах // Труды Третьей Российской школы – конференции для молодых ученых с международным участием "Математика, информатика, их приложения и роль в образовании", Тверь, 2013, - С. 20-25.

21. Андреева Е.А., Семькина Н.А. Оптимальное управление процессом распространения инфекционного заболевания с учетом латентного периода // Журнал вычислительной математики и математической физики. 2005. Т. 45. № 7. С. 1133-1140.

22. Андреева Е.А., Цирулева В.М. Вариационное исчисление и методы оптимизации. М.: Высшая школа, 2006.

23. Атанс М., Фалб П. Оптимальное управление. М.: Машиностроение, 1968.- 764 с.

24. Афанасьев А.П., Дикусар В.В., Милютин А.А., Чуканов С.А. Необходимое условие в оптимальном управлении. М.:Наука, 1990. 320с.

25. Барский А.Б. Нейронные сети: распознавание, управление, принятие решений - М.: Финансы и статистика, 2004. – 176 с.

26. Беллман Р. Теория устойчивости решений дифференциальных уравнений. — ИЛ, 1954.

27. Болтянский В.Г., Гамкрелидзе Р.В., Понтрягин Л.С. К теории оптимальных процессов. // ДАН СССР, 1956, т.110, № 1. С. 7-10.

28. Болтянский В.Г., Гамкрелидзе Р.В., Понтрягин Л.С. Теория оптимальных процессов. Принцип максимума. // Изв. АН СССР, серия матем., 1960, т. 24, № 1. -С. 3-42.

29. Болтянский В.Г. Математические методы оптимального управления. М.:Наука, 1969. 408с.л

30. Большакова И.С. Оптимизация нейронной сети, описываемой системой интегро-дифференциальных уравнений // Математические методы управления: сб. науч. тр. – Тверь: Твер. гос. ун-т, 2009, - С.20-37.
31. Большакова И.С. Оптимизация нейронной сети, описываемой системой интегро-дифференциальных уравнений // Математические методы управления: сб. науч. тр. – Тверь: Твер. гос. ун-т, 2010, - С.46-54.
32. Большакова И.С., Шаронов Д.А. Обучение нейронной сети с запаздыванием // Международный журнал «Программные продукты и системы» №2, 2011, - С.35-37.
33. Будаков Б.М., Васильев Ф.П. Некоторые вычислительные аспекты задач оптимального управления. М.: Изд-во МГУ, 1975.
34. Бутковский А.Г. Теория оптимального управления системами с распределенными параметрами. М.: Наука, 1965. 474с.
35. Бутковский А.Г. Методы управления системами с распределенными параметрами. М.: Наука, 1975. 568с.
36. Бутковский А.Г. Теория оптимального управления системами с распределенными параметрами. М.: Наука, 1965. 474с.
37. Бутковский А.Г. Управление системами с распределенными параметрами (обзор) // Автоматика и телемеханика. 1979. N 11. С. 16-65.
38. Быков Я.В. О некоторых задачах теории интегро-дифференциальных уравнений, Фрунзе, 1957, 1-320 (РЖМат, 1959, 482К).
39. Васильев А.Б. Тихонов А.Н. Интегральные уравнения. М.: Наука, 1989.
40. Васильев О. В., Тятюшкин А. И. Об одном методе решения задач оптимального управления, основанном на принципе максимума // Журнал вычислительной математики и математической физики, 1981. 21, № 6.
41. Васильев Ф.П. Лекции по методам решения экстремальных задач. М.:Изд-во МГУ, 1974. 374с.
42. Верлань А.Ф.,Сизиков В.С. Интегральные уравнения: методы, алгоритмы программы, - Киев, 1986.

43. Вольтерра В. Теория функционалов, интегральных и интегродифференциальных уравнений, Издательство М.: Наука, Главная редакция физико-математической литературы, 1982. - 304 с.

44. Габасов Р., Кириллова Ф.М. Мансимов К.Б. Необходимые условия оптимальности второго порядка (обзор). Минск. 1982. 47с. (Препринт / АН БССР. Ин-т математики, N 30).

45. Габасов Р., Кириллова Ф.М. Мансимов К.Б. Необходимые условия оптимальности второго порядка для систем с распределенными параметрами. Минск. 1982. 32с. (Препринт/АН БССР. Ин-т математики, N 31)

46. Габасов Р., Кириллова Ф.М. Особые оптимальные управления. М.:Наука, 1973. 256с.

47. Гамкрелидзе Р.В, Оптимальные процессы управления при ограниченных фазовых координатах. // Изв. АН СССР, 1960, т. 24, № 3. С. 315-356.

48. Гамкрелидзе Р.В. К общей теории оптимальных процессов. // ДАН СССР, 1958, т. 123, № 2. С. 223-226.

49. Гамкрелидзе Р.В. К теории оптимальных процессов в линейных системах. // ДАН СССР, 1957, т. 116, № 1. С. 9-11.133

50. Гамкрелидзе Р.В. Оптимальные по быстродействию процессы при ограниченных фазовых координатах. // ДАН СССР, 1959, т. 125, № 3. С. 475-478.

51. Гамкрелидзе Р.В. Теория оптимальных по быстродействию процессов в линейных системах. // Изв. АН СССР, серия матем., 1958, т. 22, № 4. -С. 449-474.

52. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учебное пособие. – М: Физматлит, 2006.

53. Гольдштейн А. Л. Теория принятия решений. Задачи и методы исследования операций и принятия решений

54. Горбань А. Н. Обучение нейронных сетей. М.: Изд. СССР-США СП "ПараГраф", 1990.

55. Дикусар В.В., Милютин А.А. Качественные и численные методы в принципе максимума. 1989г. М.: Наука, 144стр.

56. Джорбенадзе А.Н., Цуцунава Т.С. Об оптимальном управлении для одной системы, описываемой интегро-дифференциальным уравнением в частных производных // Тр. Тбилисского ун-та. 1991. N 299. С. 175-181.

57. Дубовицкий А.Я., Милютин А.А. Теория принципа максимума. В сб. "Методы теории экстремальных задач в экономике". М.: Наука, 1981. С.6-47.

58. Дубовицкий А.Я., Милютин А.А. Необходимые условия слабого экстремума в общей задаче оптимального управления. 1971 г. М.: Наука, 112 стр.

59. Евтушенко Ю. Г. Методы решения экстремальных задач и их применение в системах оптимизации. М.: Наука, 1982, 432 с.

60. Евтушенко Ю.Г. Оптимизация и быстрое автоматическое дифференцирование. Научное издание ВЦ РАН, 2013.- 144 с.

61. Егоров А.И. Об условиях оптимальности в одной задаче управления процессами теплопередачи // Журн. Вычислит, матем. и матем. физ. 1972. N 3. С.791-799.

62. Егоров А.И. Об устойчивости и оптимизации систем с распределенными параметрами // Прикл. матем. 1984. Т.20. N 4. С.95-100.

63. Егоров А.И. Оптимальное управление тепловыми и диффузионными процессами. М.:Наука, 1979. 464с.

64. Егоров Ю.В. Некоторые задачи теории оптимального управления // Ж. вычисл. матем. и матем. физ. 1963. Т.3. N 5. С.887-904.

65. Егоров Ю.В. Необходимые условия оптимальности в банаховых пространствах T // Матем. сб. 1964. Т.64(106). N 1. С.79-101.

66. Иоффе А.Д., Тихомиров В.М. Теория экстремальных задач. М.: Наука, 1974. 480с.

67. Канторович Л. В. Функциональный анализ и прикладная математика// УМН, 1948. 3, № 6, с. 89–185. "1

68. Келли Г. Д. Метод градиентов // Методы оптимизации с приложениями к механике космического полета / ред. Лейтман Д.. - М. :Наука, 1965

69. Колесник В.П. Метод синтеза оптимальных по быстродействию нелинейных систем высокого порядка с ограничениями и в условиях неопределенностей. Дис. д-ра техн. Наук 05.13.02. М., 1982. 383 с.

70. Колесников А.А. О синтезе оптимального по быстродействию управления нелинейными объектами одного класса, // Изв. Вузов. Электромеханика. 1978. -№3.- С. 310-320.

71. Красовский Н. Н. Теория управления движением. Москва: Наука, 1968.

72. Красовский Н.Н. К теории оптимального регулирования. // Автоматика и телемеханика, 1957, № 11. С. 960-970.

73. Красовский Н. Н. Некоторые задачи теории устойчивости движения. — М.: Физматгиз, 1959.

74. Круглов В. В., Борисов В. В., Искусственные нейронные сети. Теория и практика, — М.: Горячая линия - Телеком, 2001. — С. 382.

75. Крылов И. А., Черноусько Ф. Л. О методе последовательных приближений для задач оптимального управления // Журнал вычислительной математики и математической физики, 1962. 2, № 6.

76. Крылов И. А., Черноусько Ф. Л. Решение задач оптимального управления методом локальных вариаций // Журнал вычислительной математики и математической физики, 1966. 6, № 2.

77. Ларичев О.И. Теория и методы принятия решений. - М.: Логос, 2000.

78. Ларичев О.И., Поляков О.А. Человеко-машинные процедуры принятия решений многокритериальных задач МП: (Обзор). Экономика и математические методы. – 1980. – Т.16, вып. 1. – С. 127-145.

79. Мазурова И.С. Построения оптимального решения в модели хищник-жертва с помощью генетического алгоритма // Труды I Международной научно-практической конференции «Современные проблемы компьютерных наук (спкн-2013)», Пенза, 2013

80. Мазурова И.С. Решение задачи оптимального управления методом БАД // Труды Третьей Российской школы – конференции для молодых ученых с

международным участием "Математика, информатика, их приложения и роль в образовании", Тверь, 2013

81. Майоров В. В., Мышкин И. Ю. Математическое моделирование нейронов сети на основе уравнений с запаздыванием, Матем. моделирование, 2:11 (1990), 64–76.

82. Малкин И. Г. Теория устойчивости движения. — М.: Наука, 1966.

83. Маркуш И.И., Бобочко В.Н. Линейные интегро-дифференциальные уравнения -Ужгород, 1981.

84. Матвеев А.С., Якубович В.А. Оптимальное управление некоторыми системами с распределенными параметрами // Сиб. матем. журн. 1978. Т.19. N 5. С.1109-1140.

85. Милютин А.А. Принцип максимума в общей задаче оптимального управления. – М.: ФИЗМАЛИТ, 2001. – 304 с.

86. Моисеев Н.Н. Численные методы в теории оптимальных систем. Наука, Москва, 1971.

87. Моисеев Н.Н. Методы динамического программирования в теории оптимальных управлений. 1. Системы, допускающие использование шкалы управлений // Журнал вычислительной математики и математической физики. 1964. Т. 4. №3. С. 485-494.

88. Орлов Ю.В. Теория оптимальных систем с обобщенным управлением. М.: Наука. 1988. 192с.

89. Охоцимский Д. Е. К теории движения ракет // Прикладная математика и механика, 1946. 10, № 2. "1

90. Охоцимский Д. Е., Энеев Т. М. Некоторые вариационные задачи, связанные с запуском искусственного спутника Земли // Успехи физических наук, 1957. 15, № 1а. "1

91. Плотников В.И. Необходимые и достаточные условия оптимальности и условия единственности оптимизирующих функций для управляемых систем общего вида // Изв. АН СССР. Сер. матем. 1972. Т.36. N 3. С.652-679.

92. Плотников В.И. Необходимые условия оптимальности для управляемых систем общего вида // ДАН СССР. 1971. Т.199. N 2. С.275-278 .
93. Плотников В.И. Об оптимальном управлении системами с распределенными параметрами // ДАН СССР. 1967. Т.175. N 6. С.1238-1241.
94. Плотников В.И., Старобинец И.М. Фазовые включения в задачах оптимального управления // Дифференциальные уравнения. 1986. Т.22. N 2. С.236-247.
95. Плотников В.И., Сумин В.И. О первой вариации и сопряженной задаче в теории оптимального управления // Функциональный анализ и его приложения. 1976. Т. 10. Выл.4. С.95-96.
96. Подиновский В.В., Ногин В.Д.. Парето-оптимальные решения многокритериальных задач. - М.: Наука, 1982.
97. Полак Э. Численные методы оптимизации. Единый подход, - М.: Мир, 1974 -376с.
98. Поляк Б.Т. Методы линеаризации при наличии ограничений // Итоги науки и техники. Матем. анализ Е. 2 / ВИНТИ. М., 1974. С. 147 – 148.
99. Поляк Б.Т. Методы решения задач на условный экстремум при наличие случайных помех // ВМ и МФ. М., 1979. Т. 19, № 1. С. 147 – 148.
100. Понтрягин Л.С., Болтянский В.Г., Гамкредидзе Р.В., Мищенко Е.Ф. Математическая теория оптимальных процессов. 4-е изд., стереотипное. М.: Наука, 1983. - 393 с
101. Пропой А.И. Элементы теории оптимальных дискретных процессов, Наука, Москва, 1973.
102. Розоноэр Л.И. Принцип максимума Л.С. Понтрягина в теории оптимальных систем. Автоматика и телемеханика, 1959, №№ 10-12. С. 1320-1334, 1441- 1458, 1561-1578.
103. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. - М.: Горячая линия - Телеком, 2006. - С. 383.

104. Трикоми Ф. Интегральные уравнения. М.: Иностранной литературы, 1960.
105. Федоренко Р.П. Приближенное решение задач оптимального управления. М.: Наука, 1978. 488 с.
106. Четаев Н. Г. Устойчивость движения. — М.: Гостехиздат, 1955.
107. Шатровский Л. И. Об одном численном методе решения задач оптимального управления // Журнал вычислительной математики и математической физики, 1962, № 2. "1
108. Штойер Р. Многокритериальная оптимизация: теория, вычисления и приложения. - М.: Радио и связь, 1992.
109. Энеев Т. М. О применении градиентного метода в задачах теории оптимального управления // Космические исследования, 1968. 4, № 4
110. Courant R. Variational Methods for Solutions of Problems of Equilibrium and Vibrations // Bull. Amer. Math. Soc., 1943. 49, no. 1.
111. D.B. Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 3rd ed., IEEE Press, NY, 2006.
112. De Jong K.A. An analysis of the behavior of a class of genetic adaptive systems. Unpublished PhD thesis. University of Michigan, Ann Arbor, 1975. (Also University Microfilms No. 76-9381).
113. De Jong K.A., Spears W.M. A formal analysis of the role of multi-point crossover in genetic algorithms. // Annals of Mathematics and Artificial Intelligence, no. 5(1), 1992.
114. De Jong K.A., Spears W.M. An Analysis of the Interacting Roles of Population Size and Crossover // Proceedings of the International Workshop «Parallel Problems Solving from Nature» (PPSN'90), 1990.
115. E. Beretta, V. Capassa, F. Rinaldi Global stability results for a generalized Lotka_Volterra system with distributed delays. Journal of Math. Biology (1988) №26

116. Evtushenko Yu.G. Automatic differentiation viewed from optimal control theory // Automatic Different, of Algorithms. Theory, Implementation and Applic. Philadelphia: SIAM, 1991. P. 25-30.
117. Evtushenko Yu.G. Computation of exact gradients in distributed dynamic systems // Optimizat. Meth. and Software. 1998. V. 9. P. 45-75.
118. Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning. - Addison-Wesley, 1989
119. Holland J. H. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975.
120. L.J. Fogel, "Autonomous automata," Industrial Research, Vol. 4, pp. 14-19, 1962.
121. L.J. Fogel, "On the organization of intellect," Ph.D. dissertation, UCLA, 1964.
122. L.J. Fogel, A.J. Owens, and M.J. Walsh, "Artificial intelligence through a simulation of evolution," Biophysics and Cybernetic Systems: Proc. 2nd Cybern. Sciences Symp., M. Maxfield, A. Callahan, and L.J. Fogel (eds.), Spartan Books, Washington D.C., pp. 131-155, 1965.
123. L.J. Fogel, A.J. Owens, and M.J. Walsh, "Intelligent decision making through a simulation of evolution," Behavioral Science, Vol. 11:4, pp. 253-272, 1965.
124. L.J. Fogel, A.J. Owens, and M.J. Walsh, "Intelligent decision making through a simulation of evolution," IEEE Trans. Human Factors in Electronics, Vol. HFE-6:1, pp. 13-23, 1965.
125. L.J. Fogel, A.J. Owens, and M.J. Walsh, "On the evolution of artificial intelligence," Proc. 5th National Symp. Human Factors in Engineering, IEEE, San Diego, CA, pp. 63-76, 1964.
126. Mitchell M. An Introduction to Genetic Algorithms. MA: MIT Press., 1999
127. N.A. Barricelli, "Numerical testing of evolution theories: I. Theoretical introduction and basic tests," Acta Biotheoretica, Vol. 16:1-2, pp. 69-98, 1962.