

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

На правах рукописи

Хоанг Ван Хынг

**АВТОМАТИЗАЦИЯ ВЫБОРА СХЕМЫ И ПАРАМЕТРОВ
БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ САМОЛЁТНОГО ТИПА
С ИСПОЛЬЗОВАНИЕМ МНОГОДИСЦИПЛИНАРНОЙ ОПТИМИЗАЦИИ**

2.5.13. Проектирование, конструкция, производство, испытания
и эксплуатация летательных аппаратов

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
Комаров Валерий Андреевич
доктор технических наук, профессор

Самара – 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1 СОСТОЯНИЕ ТЕОРИИ И ПРАКТИКИ ПРОЕКТИРОВАНИЯ АТМОСФЕРНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ.....	12
1.1 Состояние и тенденции мирового рынка БПЛА.....	12
1.2 Развитие методов проектирования атмосферных летательных аппаратов	17
1.3 Проектная парадигма «точного попадания»	24
1.4 Аэродинамические модели	27
1.5 Весовые модели.....	30
1.6 Методы оптимизации.....	37
1.7 Цель и задачи исследования	39
2 РАЗРАБОТКА МЕТОДА ОПТИМИЗАЦИИ СХЕМЫ И ПАРАМЕТРОВ БПЛА САМОЛЁТНОГО ТИПА	40
2.1 Математическая формулировка задачи в терминах нелинейного программирования	40
2.2 Выбор и определение целевой функции.....	41
2.3 Выбор проектных переменных	42
2.4 Ограничения	45
2.5 Многодисциплинарное математическое моделирование на этапе концептуального проектирования	46
2.6 Методика выбора оптимальных значений проектных переменных БПЛА самолётного типа.....	60
2.7 Визуализация облика в виде плановой проекции и трёхмерной модели	70
2.8 Выбор настроечных параметров оптимизационного алгоритма	70
2.9 Программная реализация алгоритма оптимизации проектных переменных БПЛА	72
2.10 Выводы по главе 2.....	75

3 ДОСТОВЕРНОСТЬ И БЫСТРОДЕЙСТВИЕ РАЗРАБОТАННОЙ АСП_КП	77
3.1 Валидация математических моделей аэродинамики	77
3.2 Оценка точности алгоритма расчёта взлётной массы	78
3.3 Оценка точности метода обеспечения балансировки самолёта	81
3.4 Оценка точности расчёта массы топлива	82
3.5 Выводы по главе 3	85
4 ДОСТОВЕРНОСТЬ И ЭФФЕКТИВНОСТЬ МЕТОДА ОПТИМИЗАЦИИ	87
4.1 Постановка и решение задач	87
4.2 Анализ результатов оптимизации	88
4.3 Анализ значимости проектных переменных для целевой функции	93
4.4 Достоверность метода оптимизации	96
4.5 Оценка значимости и эффективности включения условия балансировки в расчёт и оптимизацию взлётной массы	101
4.6 Оценка влияния на быстродействие алгоритма включения оценочной величины взлётной массы в состав вектора проектных переменных	103
4.7 Выводы по главе 4	105
ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ	107
СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ	109
СПИСОК ЛИТЕРАТУРЫ	114
ПРИЛОЖЕНИЕ А	133

ВВЕДЕНИЕ

Актуальность темы

В последние годы значительно возрос интерес к беспилотным летательным аппаратам (БПЛА), используемым в различных областях, таких как военная и гражданская авиация, сельское хозяйство, логистика и экологический мониторинг. На сегодня востребованы два типа БПЛА: самолётного и роторного типа, а также их комбинации. В частности, аппараты самолётного типа могут успешно решать задачи по дистанционному зондированию Земли, картографированию, ретрансляции связи, наблюдению и мониторингу протяжённых объектов, обнаружению лесных пожаров и т. д. Такие аппараты с большим временем полёта требуют тщательного проектирования.

Современные достижения в области компьютерного моделирования, численных методов и оптимизационных алгоритмов позволяют эффективно решать сложные междисциплинарные задачи на стыке различных дисциплин: аэродинамика; механика конструкций; системы управления; энергетика и материалы. Это способствует повышению точности и надёжности процесса разработки, а также позволяет выявлять и решать скрытые проблемы на этапе концептуального проектирования, снижая риски дорогостоящих изменений на последующих этапах и сокращая общее время и затраты на разработку. Междисциплинарная оптимизация (МДО) позволяет разрабатывать БПЛА, которые могут адаптироваться к различным задачам и условиям эксплуатации. Это особенно важно в условиях быстрого изменения специфических потребностей различных областей применения. МДО БПЛА на этапе концептуального проектирования является новой и необходимой задачей.

В настоящее время актуальной является разработка автоматизированных систем для компьютерной поддержки выбора схем и параметров БПЛА самолётного типа с целью повышения лётной эффективности и сокращения времени первоначального этапа проектирования за счёт использования методов МДО.

Степень разработанности темы

С момента зарождения авиации проектирование и производство самолётов считались формой искусства, демонстрирующей творческие способности человека. Однако по мере того, как теоретические основы проектирования авиационной техники обогащались трудами Д. П. Григоровича, В. Ф. Болховитинова, П. Д. Самсонова, К. Вуда, А. Г. Бедунковича, Л. И. Сутугина, а позднее - Н. А. Фомина, А. А. Бадягина, С. М. Эгера, Э. Торенбека, Д. Реймера, Й. Роскема и многих других, оно превратилось в систематизированную научную дисциплину.

В развитии методов проектирования всё более важную роль начинает играть использование МДО. Интерес учёных и практиков к МДО неуклонно растёт. В частности, значительный вклад в развитие данного направления внесли сотрудники Московского авиационного института - Н. К. Лисейцев, О. С. Самойлович, В. В. Мальчевский, М. А. Погосян; Центрального аэрогидродинамического института (ЦАГИ) - В. И. Бирюк, В. М. Фролов, В. В. Чедрик, С. Л. Чернышев; а также представители научной школы А. А. Комарова - Д. М. Козлова, Г. А. Резниченко, О. Е. Лукьянова. Значительные исследования в этой области ведутся зарубежными авторами из Брауншвейгского, Торонтского, Мичиганского, Мюнхенского технических университетов, Германского аэрокосмического центра и многих других научных организаций.

Теоретические основы весового проектирования авиационных конструкций рассмотрены в работах Ф. Р. Шэнли, В. М. Шейнина, В. Р. Вепдзое, В. Р. Каушег и других учёных. Определённое влияние на данную работу оказали публикации по новой проектной парадигме «*concurrent design*» (точного попадания), предложенной профессорами Т. А. Weisshaar и В. А. Комаровым.

Значительное место в исследованиях по проектированию БПЛА занимают работы К. Д. и И. К. Туркиных, П. П. Афанасьева, Н. Н. Долженкова, В. Г. Гайнутдинова и др.

Цель работы

Разработка автоматизированной системы, методик, алгоритмов и программного обеспечения (ПО) для поддержки быстрого выбора рациональных параметров облика новых БПЛА самолётного типа на этапах концептуального проектирования и модификации существующих.

Задачи исследования

1. Выбор эффективного варианта оптимизационного эволюционного алгоритма и адаптация его к проектированию БПЛА самолётного типа.
2. Разработка комплекса проектных переменных для генерации аэродинамических схем в широком диапазоне от «нормальной» до схемы «утка», включая «тандем».
3. Разработка быстрого алгоритма вычисления взлётной массы повышенной точности с учётом требований продольной устойчивости.
4. Разработка математических моделей для быстрого расчёта аэродинамических характеристик (АДХ) различных аэродинамических схем.
5. Проведение анализа и выбора весовых формул основных частей конструкции самолёта с учётом масштабного фактора.
6. Разработка «Автоматизированной системы поддержки концептуального проектирования (АСП_КП)» с использованием опубликованных и вновь разработанных программных блоков.
7. Оценка достоверности результатов проектирования в разработанной АСП_КП.

Объект исследования - БПЛА самолётного типа.

Предмет исследования

Методика выбора схемы и параметров БПЛА самолётного типа.

Научная новизна работы

1. Разработан алгоритм и ПО автоматизированной системы поддержки концептуального проектирования «АСП_КП» летательного аппарата (ЛА) самолётного типа на основе варианта оптимизационного алгоритма «дифференциальной эволюции» (ДЭ), отличающийся от известных решений

интеграцией опубликованных ПО (AVL, FreeCAD) с вновь разработанными программами: параллельных вычислений; сокращения размерности популяции; использования метода штрафной функции; введением пространства проектирования для учёта дискретных переменных.

2. Предложен компактный набор из 12 проектных переменных, способный породить в процессе оптимизации различные аэродинамические схемы от «нормальной» до схемы «утка», включая «тандем» путём введения в рассмотрение относительной площади двух несущих поверхностей.

3. Разработана новая методика и ПО уточнённого расчёта взлётной массы с внутренним циклом учёта обеспечения условия продольной устойчивости.

4. Разработан комплекс методик «*test-opt*» для исследования значимости проектных переменных в процессе оптимизации, оценки близости получаемого решения к оптимальному и исследования рельефа целевой функции в многомерном пространстве проектных переменных.

Теоретическая значимость работы

1. Разработанная методика моделирования АДХ ЛА со сложными аэродинамическими компоновками несущих поверхностей позволяет анализировать все основные известные схемы самолёта при оптимизации проектных параметров и обеспечивает возможность расширения рассмотрения различных новых конфигураций в будущем.

2. Разработанная методика использования уравнения существования ЛА с учётом дополнительного требования обеспечения продольной устойчивости сокращает число проектных переменных на две единицы и время оптимизационных расчётов.

3. Предложенный способ включения оценочной величины взлётной массы в состав вектора проектных переменных существенно ускоряет процесс оптимизации.

Практическая значимость диссертации

1. Данная работа ориентирована на применение в проектных

подразделениях предприятий авиационной промышленности.

2. Особенность методики состоит в том, что она позволяет получать, наряду с оптимальными значениями вектора проектных переменных, обширную дополнительную информацию в графическом и цифровом виде - облик самолёта, АДХ, энерговооружённость, относительные и абсолютные массы топлива, конструкции и двигателя - для использования на последующих этапах проектирования ЛА.

3. Предложен эффективный вариант алгоритма ДЭ для проектирования БПЛА самолётного типа, который позволяет значительно сократить время проектирования и проводить оценку массы компонентов для весового контроля процесса проектирования.

4. Разработанные методология и ПО поддержки концептуального проектирования использованы при выполнении НИР по теме «Концептуальное проектирование многоцелевого транспортного БПЛА средней весовой категории вертикального взлёта и посадки» в целях реализации программы развития Самарского университета до 2030 года в рамках программы стратегического академического лидерства «Приоритет-2030», шифр темы: 21в-Р042-202, и в настоящее время используется в учебном курсе «Многодисциплинарная оптимизация» в передовой инженерной аэрокосмической школе Самарского университета.

Методы исследования

Математические модели аэродинамики. Аэродинамический эксперимент. Метод дискретных вихрей (МДВ). Инженерные методы расчёта АДХ. Методы весового анализа. Методы линейной алгебры и аналитической геометрии. Методы нелинейного математического программирования (НМП).

Положения, выносимые на защиту

1. ПО автоматизированной системы поддержки концептуального проектирования «АСП_КП» на основе варианта оптимизационного алгоритма ДЭ.

2. Компактный набор из 12 проектных переменных, способный

порождать в процессе оптимизации различные аэродинамические схемы от «нормальной» до схемы «утка», включая «тандем».

3. Методика уточнённого расчёта взлётной массы с учётом обеспечения продольной устойчивости.

4. Метод ускорения оптимизационного процесса путём включения оценочной величины взлётной массы в качестве дополнительной проектной переменной.

5. Комплекс методик «*test-opt*» для исследования значимости проектных переменных, оценки близости получаемого решения к оптимальному и исследования рельефа целевой функции в многомерном пространстве проектных переменных.

Достоверность научных результатов, получаемых по предлагаемой методике, доказана специально разработанным комплексом методов и методик:

1. Путём сопоставления результатов математических моделей аэродинамики с экспериментальными данными, полученными в аэродинамической трубе;

2. Через оценку точности алгоритма расчёта взлётной массы путём сопоставления расчётного значения взлётной массы по предлагаемой методике с характеристиками двух реальных БПЛА самолётного типа;

3. Оценка точности расчёта массы топлива выполнена по двум известным методикам и предложена собственная более точная методика для ЛА с большой дальностью полёта;

4. Достоверность и эффективность разработанной методики поиска оптимальных значений проектных переменных доказаны путём выполнения реинжиниринга двух известных ЛА самолётного типа и с применением комплекса методик «*test-opt*».

Апробация работы

Основные результаты диссертационной работы были доложены на пяти конференциях различного уровня: XXVI Всероссийский семинар по

управлению движением и навигации летательных аппаратов (г. Самара, Самарский университет, 14-16 июня 2023 г.); Международная молодёжная научная конференция «XIV Королевские чтения», посвящённая 35-летию со дня первого полёта МТКС «Энергия - Буран» (г. Самара, Самарский университет, 03-05 октября 2023 г.); Всероссийская научно-практическая конференция «Авиация будущего: тренды, вызовы и возможности», посвящённая 55-летию первого полёта сверхзвукового пассажирского самолёта Ту-144 (г. Казань, КНИТУ-КАИ, 06-08 декабря 2023 г.); XXVII Всероссийский семинар по управлению движением и навигации летательных аппаратов (г. Самара, Самарский университет, 11-14 июня 2024 г.); XXII Международная конференция по методам аэрофизических исследований (ICMAR 2024) (г. Новосибирск, ИТПМ СО РАН, 01-05 июля 2024 г.).

Публикации

По теме диссертации опубликовано 7 работ. В том числе 3 статьи в научных изданиях, рекомендованных ВАК [1-3]; 1 статья в научном издании, индексируемом базой Scopus [4]; 3 статьи в прочих изданиях [5-7].

Личный вклад автора

Все результаты, представленные в диссертации, получены автором самостоятельно или при его непосредственном участии. Разработка и реализация методики численного моделирования самолёта на основе проектных переменных с использованием языка программирования Python и ПО AVL выполнена автором лично. Разработка аналитического метода решения условия продольной балансировки самолёта в различных режимах полёта выполнена при непосредственном участии автора. Модификация алгоритма дифференциальной эволюционной оптимизации для задачи оптимального выбора проектных параметров самолёта и реализация с использованием языка программирования Python выполнена автором лично. Сопоставление результатов экспериментальных исследований АДХ в аэродинамической трубе с результатами расчётов выполнены автором лично. Применение предложенного метода оптимального проектирования для

решения прикладных задач выполнено автором лично. Анализ и оценка полученных результатов выполнены при непосредственном участии автора. Совместно полученные результаты представлены с согласия соавторов.

Структура и объём работы

Работа состоит из введения, 4 глав, заключения, списка литературы из 176 наименований, списка сокращений и условных обозначений и 1 приложения (12 модулей программ). Работа содержит 154 страницы, 39 рисунков и 15 таблиц.

Соответствие работы паспорту научной специальности

Область исследования соответствует паспорту научной специальности 2.5.13. «Проектирование, конструкция, производство, испытания и эксплуатация летательных аппаратов» по пунктам: «1. Разработка методов проектирования и конструирования, математического и программно-алгоритмического обеспечения для выбора оптимальных облика и параметров, компоновки и конструктивно-силовой схемы, агрегатов и систем ЛА...»; «5. Разработка методов, моделей и программного обеспечения для принятия оптимальных решений проектно-конструкторских задач при заданных ограничениях с учётом их компромиссного характера, риска и различимости сравниваемых вариантов изделий (процессов)...»; «11. Автоматизированные системы, функциональные и инвариантные подсистемы и средства обеспечения (математического, лингвистического, информационного, программного, технического, методического, организационного) научных исследований, проектирования, конструирования...».

1 СОСТОЯНИЕ ТЕОРИИ И ПРАКТИКИ ПРОЕКТИРОВАНИЯ АТМОСФЕРНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

1.1 Состояние и тенденции мирового рынка БПЛА

В настоящее время наблюдается интенсивное развитие беспилотной авиации по многим научно-техническим направлениям. Современные беспилотное воздушное судно (БВС)ⁱ выполняются по различным аэродинамическим схемам и могут представлять собой ЛА различного типа и размерности в зависимости от их назначения. Обсуждение основных тенденций в данной области целесообразно начать с определений и краткого очерка истории.

Официальные определения

а) Согласно Воздушному кодексу Российской Федерации, БВС определяется как «воздушное судно, управляемое, контролируемое в полёте пилотом, находящимся вне борта такого воздушного судна (внешний пилот) или системой автоматического управления в полностью автономном режиме» [8].

б) Согласно определению Министерства обороны США, «БПЛА – это ЛА с двигателем, который не имеет на борту человека-оператора, использует аэродинамические силы для подъёма транспортного средства, может летать автономно или управляться дистанционно, может быть одноразовым или восстанавливаемым и может нести летальную или несмертельную полезную нагрузку» (A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload)ⁱⁱ [9].

История разработок

Идея БВС возникла ещё в глубокой древности. Свидетельством этому являются древние изображения воздушного змея в небе, управляемого людьми с Земли посредством леера [10].

ⁱ наряду с аббревиатурой БВС в русскоязычной технической литературе используется часто БПЛА.

ⁱⁱ переведено на русский автором.

Одними из первых назначений беспилотных авиационных систем (БАС) в истории оказались военные задачи. Так, в 1849 году австрийская армия применила воздушные шары для бомбардировки Венеции, а в начале XX века с появлением первых аппаратов тяжелее воздуха начали создаваться свободнопланирующие авиабомбы и торпеды [11]. Этот период также ознаменовал собой и появление первой идеи разработки беспилотных самолётов-разведчиков [12, 13].

В годы Первой и Второй мировых войн велись работы по созданию БПЛА с автономным управлением преимущественно инерциального типа или по радиоканалу. Однако уровень развития науки, техники и технологий того времени не позволял получать БПЛА, способные приблизиться по эффективности применения к пилотируемым аппаратами [14, 15].

В послевоенный период наряду с разработкой сверхзвуковых самолётов появилась необходимость использования скоростных БПЛА в качестве мишеней.

На рубеже XX - XXI веков наблюдалась интенсификация разработок БАС самых разнообразных типов и назначений во многих странах мира, связанная в основном с появлением электронных и микроэлектромеханических систем нового поколения [16] (рисунок 1).

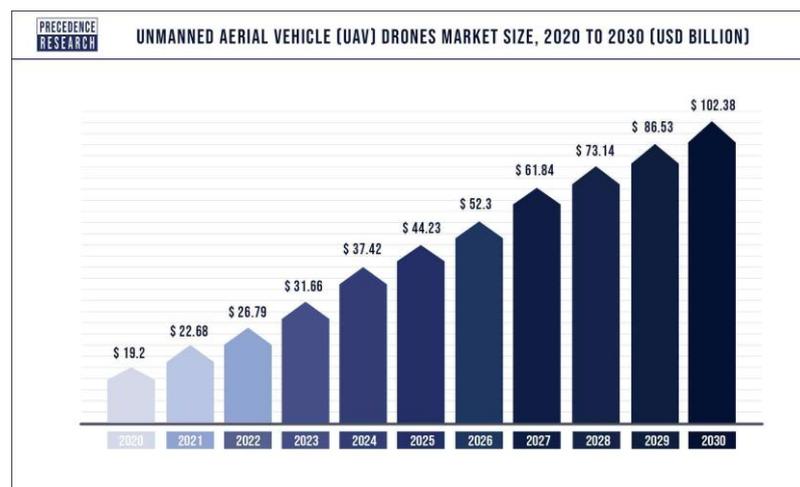


Рисунок 1 – Рост мирового рынка БПЛА [17]

Классификация

Современный рынок БПЛА чрезвычайно разнообразен, разделить БПЛА по определённому критерию затруднительно. Существуют попытки

классификации БАС по следующим критериям:

- по размеру от менее 0,1 м до более 15 м [18];
- по массе от менее 5 кг до более 2000 кг [19];
- по времени полёта от менее 5 часов до более 24 часов и дальности полёта от менее 100 км до более 1500 км [19];
- по потолку полёта от менее 1 км до более 10 км [19];
- по удельной нагрузке на крыло от менее 50 до более 100 кг/м² [19].

Очевидно, что к перечисленному списку целесообразно было бы добавить *массу полезной нагрузки*.

Более существенным признаком различия, используемым для классификации БПЛА, является их тип, определяемый способом создания подъёмной силы. На сегодня основную массу БАС составляют аппараты, выполняемые по роторному типу и по самолётному типу. В последнее время появляется значительное количество БВС комбинированного типа, сочетающего в себе признаки обоих перечисленных типов в различных вариациях.

БВС первого типа используют ротор (несущий винт) для создания подъёмной и пропульсивной силы. Данный тип нашёл своё отражение в БВС - вертолётах: БВС ВТ 45 [20]; Ка-37 [21]. Существенное уменьшение размерности БВС по сравнению с пилотируемыми аппаратами подобного типа и использование электрической силовой установки сделало возможным применять на БВС роторного типа несколько воздушных винтов с целью реализации системы управления БВС разностью роторов без использования сложных автоматов перекося. Такие БВС стали называть мультироторами или по-простому дронами (от англ. drone - «трутень»). Они могут оснащаться тремя, четырьмя, шестью или восемью роторами (редко двумя). В таком случае БВС мультироторного типа будет называться трикоптером, квадрокоптером, гексакоптером, октокоптером (бикоптером) соответственно.

Преимущество этих аппаратов - вертикальный взлёт и посадка, зависание над точкой, хорошая манёвренность. Основной недостаток - большой часовой расход энергии на полёт [1].

Мультироторный тип имеет широчайшее распространение среди БВС с электрической силовой установкой, начиная от самой малой размерности «микро» (0-0,15 кг), заканчивая аппаратами со взлётной массой до 80 кг [22] и экспериментальными аппаратами до 700 кг [23]. Крупноразмерные мультироторы нередко оснащаются двигателем внутреннего сгорания (ДВС) или гибридными установками для увеличения продолжительности полёта и в ряде случаев могут конкурировать с БВС самолётного типа [1]. Важно отметить, что энергетическая эффективность мультироторов в основном определяется аэродинамикой несущих винтов и грамотным подбором режимов работы двигателей. Аэродинамика корпуса, как и у пилотируемых вертолётов, оказывает на общую энергетическую эффективность всего аппарата значительно меньшее влияние.

БПЛА «самолётного типа» используют для создания подъёмной силы неподвижное крыло. Это способ создания подъёмной силы требует примерно в 5...10 раз меньше затрат энергии силовой установки, чем мультироторный. Поэтому именно БПЛА самолётного типа могут летать на расстояние значительно больше 1000 км и находиться в воздухе до 24 часов и более.

Оба типа БВС сегодня активно используются и органично дополняют друг друга в различных БАС.

На рисунке 2 представлен прогноз развития рынка БВС в ближайшее десятилетие.

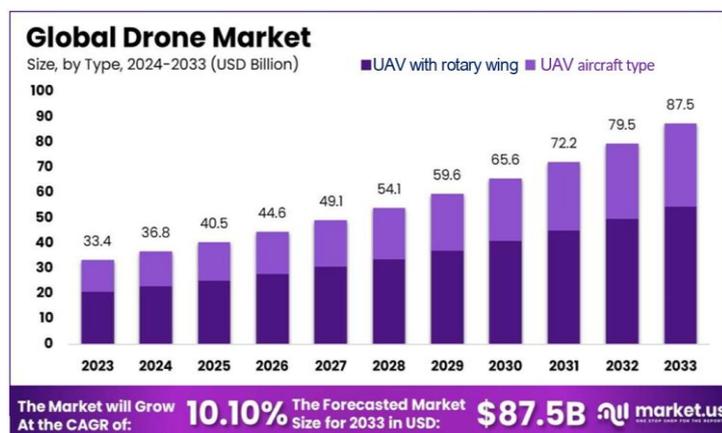


Рисунок 2 – Прогноз рынка БПЛА по типам [24]

На обоих рассмотренных типах БВС могут использоваться как тепловые, так и электрические или гибридные типы силовой установки.

Большинство существующих БПЛА самолётного типа находится в области, ограниченной по дальности - 100 км и взлётной массе - 100 кг. В то же время современный опыт использования крупноразмерных БПЛА, таких как Орион [25], Альтиус-РУ [26], U-40 [27], MQ-1 Predator [28], Байрактар ТБ-2 [29] и их модификаций с длительностью полёта более 20 часов и взлётной массой более 500 кг показывает эффективность тяжёлых БВС и их востребованность. Отсутствие экипажа на борту таких аппаратов и систем жизнеобеспечения, а также сниженные по отношению к пилотируемым аппаратам требования по безопасности полётов позволяют существенно снизить массу БВС, повысить его функциональные качества и снизить стоимость эксплуатации. Тем не менее, крупноразмерные БВС являются сложными, дорогостоящими, высокотехнологичными системами, а их конструкция по многим параметрам мало чем отличается от привычных конструкций пилотируемых ЛА: в такой же степени проявляется действие закона «квадратов-кубов» с ростом размеров конструкций, а её масса уже определяется силовой работой планера в эксплуатации, а не особенностями технологии или требованиями эксплуатации, как у малоразмерных БВС [30]. По аналогии с пилотируемыми ЛА при проектировании крупноразмерных БВС, особенно большой продолжительности полёта, возникают такие же противоречия между различными предметными областями науки, затрагивающими аэродинамику, динамику полёта, прочность, весовой анализ, термодинамику и т. д. В качестве примера можно привести дилемму выбора оптимального значения удлинения крыла: увеличение удлинения позволяет ощутимо снизить относительную массу энергоносителя, но снижает общую жёсткость конструкции, что выливается в повышение массы крыла и снижение сопротивления усталости. Аэродинамическое совершенство каждой из частей БВС самолётного типа, а также их взаимное влияние друг на друга, существенным образом влияет на эффективность аппарата в целом.

В этой связи проведение начальных стадий разработки крупноразмерных БВС необходимо выполнять с использованием

современных подходов проектирования, базирующихся на численном математическом моделировании и МДО. Применение данных подходов в проектировании авиационной техники является для БВС не менее актуальным, чем для пилотируемых самолётов, так как отсутствие множества требований по безопасности, эргономике и обзора экипажа открывают широкие возможности применения множества различных аэродинамических схем и компоновок, недоступных ранее на пилотируемых аппаратах, но открывающие новые возможности на беспилотных.

Исходя из истории, современного состояния и перспектив развития БПЛА с большими временем полёта и грузоподъёмностью для решения разнообразных задач, можно сделать вывод об актуальности разработки системы автоматизации концептуального проектирования БПЛА самолётного типа.

1.2 Развитие методов проектирования атмосферных летательных аппаратов

1.2.1 Метод копирования и имитации

Самый примитивный и самый первый в истории авиации метод проектирования - имитация (копирования) форм животных и птиц из дикой природы, которые умеют летать. В IX веке андалузский учёный и философ Аббас ибн Фирнас смоделировал крылья птиц, чтобы провести, пожалуй, первый в мире лётный эксперимент [31]. В XV веке Леонардо да Винчи, итальянский учёный и художник, изучал способность птиц летать. Он предложил несколько схем и чертежей ЛА, основанных на принципе птичьих крыльев [32].

1.2.2 Метод проб и ошибок

После ряда интенсивных поисков конструкции ЛА тяжелее воздуха в конце XIX века история авиационного проектирования прошла долгий путь испытаний реальными полётами. Недостатки предыдущих испытаний были преодолены и улучшены в последующих испытаниях. Такой метод называется методом «проб и ошибок», он же эмпирический метод. Знаменитые имена

этого периода и метода включают:

- Можайского А. Ф., русского изобретателя, который в 1882 году построил и получил патент на ЛА самолётного типа со всеми основными частями современного самолёта [33, 34];

- Отто Лилиенталя, немецкого инженера, считающегося одним из пионеров авиации, который совершил тысячи планирующих полётов в период 1891–1896 годов, используя ЛА, напоминающие птичьи крылья. Опыты Лилиенталя внесли значительный вклад в развитие основ воздухоплавания в конце XIX века [34, 35];

- братьев Райт, американских изобретателей, которые после серии неудач создали ЛА и успешно осуществили 17 декабря 1903 года управляемый пилотируемый полет с двигателем [34];

- ряда других изобретателей [36].

Метод «проб и ошибок» основан на накопленном опыте тестирования моделей и даже на реальном тестировании самим разработчиком. Они развивали свои идеи, основываясь на интуиции и простых аэродинамических принципах, таких как баланс между подъёмной силой и гравитацией. Этот метод не требует глубокого теоретического понимания, а требует лишь проведения практических испытаний. Такой подход являлся, пожалуй, единственным доступным методом на ранних этапах развития авиационной науки в отсутствие теоретических знаний в этой области и смежных областях наук. На заре авиации было важно, что этот метод не связан теоретическими предположениями и может позволить привести исследователя к открытию новых творческих решений.

Основное ограничение метода заключается в отсутствии научной теоретической основы и требует многократных испытаний, что приводит к колоссальной трате времени, средств и, как это было часто в начале пути, человеческих жизней. Результаты испытаний без теоретического аппарата трудно анализировать и систематизировать, а значит сложно объяснить причины сбоев и неудач для внесения изменений и регулировок в дальнейшем.

По мере того, как системы становятся более сложными, метод «проб и ошибок» становится труднее применять: количество отслеживаемых переменных быстро увеличивается, что делает метод непрактичным.

1.2.3 Статистический метод

В 20-30-е годы прошлого века исследователи уже смогли установить основные связи между формой, размерами ЛА и его лётными характеристиками, накопив при этом достаточно опыта получения неудачных решений. В этот период методы проектирования постепенно становятся статистическими (метод по прототипам).

Особенностью этого метода является то, что параметры ЛА новой конструкции будут определяться путём интерполяции и экстраполяции на основе статистических данных предыдущих самолётов.

Однако для получения положительных результатов прогноза необходимо накапливать прототипы в течение длительного периода времени, в то время как наука стремительно развивается с каждым днем, что приводит к тому, что этот метод не отвечает требованиям. Кроме того, неспособность учесть все многообразие взаимосвязей между параметрами и характеристиками ЛА приводит к ошибочным прогнозам того, как изменятся прототипы.

1.2.4 Становление и развитие науки о проектировании

На следующем этапе развития самолётостроения проектирование авиационной техники стало рассматриваться в качестве научного направления. Процесс проектирования был систематизирован и разбит на ряд этапов. Одними из первых обобщающих работ можно считать книги авторов: Д. П. Григоровича; В. Ф. Болховитинова [37]; П. Д. Самсонова [38]; К. Вуда [39]; Л. И. Сутугина [40]; А. Г. Бедунковича [41] и многих других, посвящённые проектированию гидросамолётов, металлических самолётов и т. п. [42].

Развитие методов расчёта массы, аэродинамики, устойчивости, управляемости и лётно-технических характеристик самолёта заложило научно-практическую основу для определения его конструктивных параметров. Эти методы основаны на решении системы уравнений, отражающих взаимосвязь

параметров и характеристик ЛА с учётом различных ограничений. Использование аналитических методов позволило исследовать влияние конструктивных параметров и ограничений на технические и экономические показатели ЛА. Следующим этапом стало определение оптимальных параметров самолёта, отвечающих заданным критериям. Это привело к развитию методов оптимального проектирования. Одной из первых работ в этой области является исследование советского инженера Н. Н. Фадеева «Изыскание рациональных размеров самолёта» (Труды ЦАГИ, вып. 461, 1939).

Проектирование ЛА с использованием такого подхода связано с необходимостью оптимизации сложных систем, но до развития информатики и современных алгоритмов выбор проектировщиков заключался в последовательной оптимизации каждой характеристики. Этот метод предполагает, что большая проблема разбивается на более мелкие подзадачи, что позволяет проектантам обрабатывать каждую задачу одну за другой, не сталкиваясь с общей сложностью.

Однако результаты метода проектирования на основе анализа частных критериев имеют низкую информативность, поскольку подсистемы могут быть оптимальными изолировано друг от друга, но сама система целиком - не оптимальна, а критерии эффективности каждой из оптимизаций могут конфликтовать друг с другом. Выбор оптимального порядка параметров также меняет конечный результат и увеличивает затраты времени проектирования из-за необходимости выполнять множество циклов с большим количеством переменных.

Начиная с 1960-х годов начали появляться фундаментальные труды по проектированию самолётов, в которых были опубликованы обширные статистические данные по существующим и успешно эксплуатируемым самолётам того периода. В числе таких трудов следует отметить работы Н. А. Фомина 1961 г. [43], А. А. Бадягина 1972 г. [44], С. М. Егера 1983 г. [45]. В этих трудах приводится большое количество *весовых формул* для предсказания относительной массы конструкций частей планера самолётов различной

размерности. Позднее эти книги переиздавались [46]. К ним также следует добавить ряд зарубежных изданий под авторством Дж. Д. Андерсона 1999 г. [47], Д. Хау 2000 г. [48], Э. Торенбика 2013 г. [49], М. Х. Садрея 2013 г. [50], П. М. Сфорци 2014 г. [51], Дж. Роскэма 2017 г. [52] и Д. П. Реймера 2018 г. [53].

В современных условиях разработки авиационной техники существенное внимание уделяется начальным стадиям проектирования, так как от качества и сроков их выполнения зависит конкурентоспособность создаваемого изделия в целом: до 80% ключевых технических решений принимается на начальных этапах проектирования при условии малых выделяемых временных и человеческих ресурсов [54, 55]. Решающим фактором успеха проекта в условиях неопределённости и недостаточности информации начальных стадий проектирования является точность прогнозирования характеристик будущего самолёта при приемлемых затратах времени. Аналитические математические модели позволяют проводить исследования в самом общем виде, однако их реализация сталкивается с трудностями, когда число рассматриваемых переменных велико, правила изменения этих переменных приводят к необходимости принятия сильно упрощающих допущений. Данное обстоятельство обуславливает необходимость развития численного моделирования физических процессов и его использования уже на самых ранних стадиях разработки проекта. Развитие теоретических основ проектирования и значительные достижения в области компьютерных технологий создают основу для возможности автоматизации большинства операций проектирования [56]. Развитие современных алгоритмов оптимизации, таких как эволюционные, генетические, топологические алгоритмы и т. п. открывают новые возможности поиска глобального минимума за более короткое время, в том числе во многокритериальной среде.

Сочетание высокоточных численных математических моделей из разных предметных областей и современных алгоритмов оптимизации открывают новое научное направление в области проектирования

авиационной техники, именуемое сегодня МДО. МДО позволяют одновременно учитывать и балансировать требования из многих различных инженерных дисциплин на основе одного или нескольких критериев.

Истоки идей МДО можно проследить от Шмидта [57, 58] и Хафтки [59-61], которые расширили свой опыт структурной оптимизации, включив в него другие дисциплины. Одним из первых применений МДО было проектирование крыльев самолётов с одновременным учётом трёх разнородных требований минимальной массы конструкции, аэродинамики и управляемости [62-65]. С тех пор количество научных публикаций, применяющих МДО в области проектирования самолётов, постоянно увеличивается [66, 67].

С позиций системного анализа задача оптимизации структуры и параметров ЛА может быть выражена в математических терминах НМП согласно [56, с. 55] в следующем виде:

$$\begin{aligned} \min F(X); \\ X \in X_{\text{доп}}; \\ X_{\text{доп}} = \{X | g_j(X) \leq 0, j = \overline{1, m}\}. \end{aligned} \quad (1)$$

Здесь $F(X)$ - целевая функция; X - проектные параметры; $g_j(X)$ - ограничения; $X_{\text{доп}}$ - допустимая область.

Сегодня, интегрируя новые технологии, такие как машинное обучение и искусственный интеллект [68, 69], модель Кригинга [70], модели обратной связи [71], МДО имеет значительные перспективы для развития новых возможностей решения более сложных и комплексных задач.

1.2.5 Особенности проектирования беспилотных летательных аппаратов самолётного типа

Особенности проектирования БПЛА рассматриваются в книгах: под редакцией П. П. Афанасьева [54], под редакцией М. А. Погосяна [46], под редакцией И. С. Голубева и И. К. Туркина [72]; в сборнике статей [73] и в ряде других изданий, например, в статье [74] подробно рассматривается проектирование БПЛА специального назначения.

БПЛА унаследовали все теоретические основы проектирования, разработанные для пилотируемых ЛА. Однако, они имеют свои особенности, влияющие определённым образом на процесс их проектирования.

Отсутствие экипажа на борту БВС, систем его жизнеобеспечения и спасения, герметичной кабины, а также сниженные по отношению к пилотируемым аппаратам требования по безопасности полётов позволяют существенно снизить массу БВС и силовой установки, улучшить АДХ планера и уменьшить внутренние объёмы фюзеляжа, повысить его функциональные качества и снизить стоимость эксплуатации. К преимуществам можно отнести и удешевление технологического процесса серийного изготовления БВС. Часто БВС имеют и более гибкие возможности для модификаций и модернизации по сравнению со своими пилотируемыми «предками». При проектировании БВС можно закладывать большие запасы по максимальным эксплуатационным перегрузкам, не ориентируясь на физические возможности человека.

БВС не касаются требования эргономики рабочего места экипажа, включая проведение проводки управления к постам управления лётчика и хороший обзор из кабины, оказывающий влияние не только на массу конструкции кабины и аэродинамику планера в целом, но и выбор аэродинамической схемы всего аппарата. Последнее обстоятельство позволяет применять к БВС уникальные аэродинамические схемы с особыми свойствами, включая такие схемы как «утка», «летающее крыло», «тандем», несимметричные схемы, комбинированные решения.

Отсутствие человека на борту позволяет БПЛА работать в воздухе сверхдлительное время без посадок, от нескольких часов до нескольких суток, повышая качество выполнения особого рода задач, как мониторинг, наблюдение, поиск, наведение, ретрансляция, интернет, гиперспектральный анализ и т. п. Для выполнения этих задач БПЛА подходит лучше, т. к. масса полезной нагрузки сопоставима или часто существенно ниже массы экипажа (0,5...40 кг для подобных задач). Таким образом, у пилотируемых конкурентов

основная нагрузка приходится на человека, который в условиях современных технологий и автоматизации является по сути лишним (но и значительным по относительной доле в общей массе) грузом.

Благодаря своей менее сложной конструкции и отсутствию строгих требований к безопасности полётов, БПЛА часто имеют более простой, быстрый и непрерывный процесс производства. Это позволяет производителям БПЛА легче вносить изменения в конструкцию в соответствии с требованиями каждой конкретной задачи без больших издержек времени и затратах на производство.

Таким образом, БПЛА самолётного типа, опираясь на весь опыт разработки самолётов с традиционными схемами, имеют ряд своих существенных особенностей, для учёта которых целесообразна разработка специализированной АСП_КП.

1.3 Проектная парадигма «точного попадания»

В авиастроении, как и в ряде других областей сложной техники, на смену традиционному - итерационному - порядку проектирования приходит новый порядок - проектная технология «точного попадания» [75]. В традиционной технологии при выборе принципиальных технических решений создаваемого самолёта в большинстве случаев используются: эвристический подход, предшествующий опыт разработчиков и данные по удачным прототипам. Далее для ускорения проекта выбираются основные геометрические, массовые и энергетические параметры самолёта с использованием упрощённых математических моделей. На основе этих решений последовательно выполняются эскизное и детальное проектирования, разрабатывается производственная документация, изготавливается опытный экземпляр изделия и проводятся испытания. На стадии испытаний, как правило, выявляются ошибки проекта, после чего начинается длительный, затратный и трудно прогнозируемый в плане результата процесс доводок и доработок (рисунок 3).

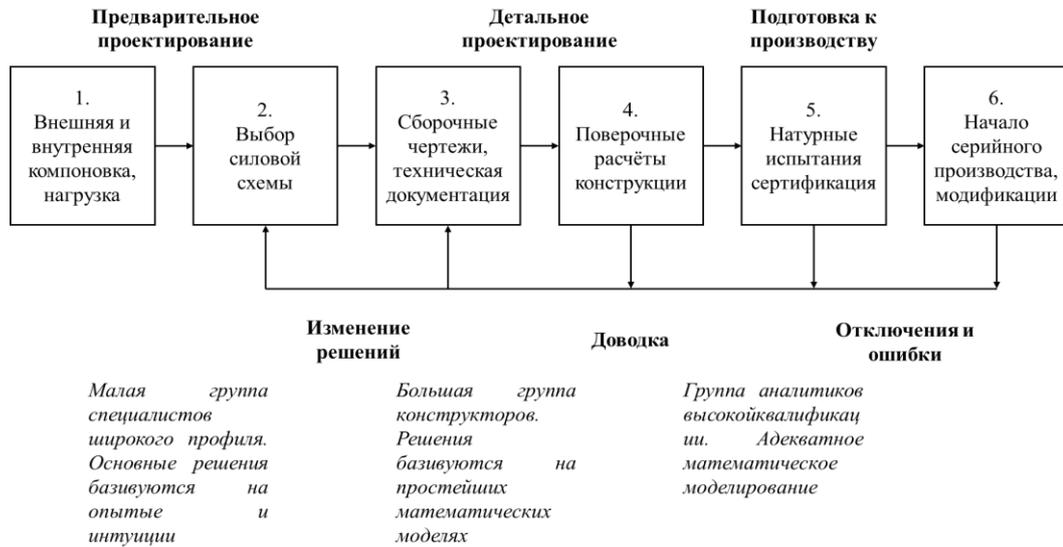


Рисунок 3 – Схема традиционного порядка проектирования авиационных конструкций [75]

Увеличение количества доработок выливается в итоге в увеличение взлётной массы, сокращение дальности полёта и ухудшению прочих характеристик будущего самолёта.

Новая проектная парадигма «точного попадания» предполагает использование высокоточных математических моделей из различных дисциплин на ранних стадиях проектирования совместно с высокопроизводительными методами оптимизации, которые позволяют решать задачи с непрерывными и дискретными проектными переменными (рисунок 4).



Рисунок 4 – Схема проектирования ЛА по технологии «точное попадание» [75]

Причины и условия появления новой парадигмы отражены в ряде публикаций:

1982 г. - группой учёных из Куйбышевского авиационного института имени академика Королёва С. П. и специалистов из ОКБ имени Ильюшина С.

В. предложен новый порядок проектирования планера в связи с разработкой самолёта Ил-96 [76]. Существо нового порядка опиралось на внедрение нового метода расчёта конструкций - метод конечных элементов (МКЭ) и метода структурной оптимизации, использующего математическую модель тела переменной плотности [77].

1995 - 2002 г. в публикациях [75, 78, 79] выполнено осмысление нового порядка проектирования как новой парадигмы и появления нового термина «*concurrent design paradigm*», который может быть переведён на русский язык как «проектная технология точного попадания». Позднее в русскоязычной литературе в этом смысле стал использоваться термин «точное проектирование» [80].

2004 - 2008 г. в [81, 82] проведён анализ причин завышения веса крыла малого удлинения и выдвижение тезиса о целесообразности использования математических моделей высокого уровня на ранних стадиях проектирования: «*Статические испытания крыла самолёта Як-130 с умеренной стреловидностью показали, что в одном из расчётных случаев нагружения угол закручивания концевое сечения составляет порядка 10° . Такое большое значение показывает, что учёт деформации крыла мог бы дать другое, более выгодное с точки зрения прочности, распределение нагрузки и соответствующее снижение массы конструкции. Однако для этого потребовалось бы переразмеривание конструкции и перерасчёты распределения давления до сходимости процесса. На завершающем этапе проектирования и в стадии доводок ЛА делать это сложно или практически невозможно [82]*».

2018 г. в [83] рассмотрены возможности разработки глубокой модификации самолёта Ан-124 с использованием конечно-элементного моделирования, метода МДВ, безразмерного коэффициента силового совершенства конструкций C_K [84] как аналога аэродинамического качества, метода групповой оптимизации Гаусса - Зейделя [85].

2022 г. в [86] показаны возможности использования технологии точного

проектирования в разработке сложных пространственных деталей из слоистого композиционного материала, экспериментальная проверка результатов проектирования.

Из приведённого обзора следует, что новая проектная парадигма начинает использоваться на практике и даёт существенные результаты как по сокращению сроков разработки новой техники, так и по улучшению основных характеристик: уменьшению взлётной массы, снижению коэффициента топливной эффективности и т. п. Ключевыми элементами новой парадигмы являются:

- использование методов оптимизации, позволяющих рассматривать большое количество как непрерывных, так и дискретно изменяющихся проектных переменных;*
- применение высокоточных численных математических моделей аэродинамики и механики твёрдого тела (весового анализа);*
- программная реализация алгоритмов с возможностью параллельных вычислений.*

Данные направления являются приоритетными для разработки системы автоматизированного концептуального проектирования БПЛА самолётного типа.

1.4 Аэродинамические модели

Одной из наиболее основных и важных задач в процессе разработки ЛА является выбор аэродинамических моделей и методов для проведения аэродинамических расчётов. Традиционным способом определения АДХ проектируемого самолёта является проведение испытаний его моделей в аэродинамических трубах. Этот метод позволяет комплексно оценить характер физических явлений в соответствии с реальными условиями эксплуатации самолёта. Развитие области авиастроения привело к появлению образцов авиационной техники с нетрадиционными аэродинамическими схемами и компоновками, что ещё больше подчёркивает роль этого метода в определении АДХ. Однако, сложность обеспечения критериев подобия,

высокая стоимость и значительные временные затраты на эксперимент делают данный метод пригодным для целей проектирования ЛА только для проверки ключевых технических решений.

Появление методов математического моделирования обтекания тел и их внедрение в процесс проектирования ЛА позволило во много заменить эксперимент и многократно расширить количество рассматриваемых вариантов технических решений с сохранением требуемой точности результатов. Наиболее объективными, а значит и точными, с точки зрения описания протекания физических процессов в аэродинамике на сегодня являются методы, использующие в своей основе численное решение уравнений Навье-Стокса. Чаще всего для замыкания систем уравнения используются модели турбулентности, обеспечивающие требуемую точность расчётов и не требующих необходимости выполнения чрезвычайно трудозатратного прямого моделирования турбулентности. Реализация решения систем уравнений Навье-Стокса с моделями турбулентности обычно выполняется методом контрольных (конечных) объёмов с применением высокопроизводительной вычислительной техники на офисных машинах или суперкомпьютерах [83, 87]. Наиболее популярное коммерческое ПО, реализующее этот метод, включает в себя: Logos CFD [88], Ansys CFX, Ansys Fluent [89], FlowVision [90], Star CD, Star CCM+ [91], Numeca [92], OpenFOAM [93] и многие другие. Применение уравнений Навье-Стокса для моделирования течения рабочей жидкости и определения АДХ ЛА даёт результаты с высокой точностью, однако основными недостатками являются высокая сложность и большое время расчёта, что не позволяет применять его на ранних стадиях проектирования. Кроме того, метод контрольных объёмов использует объёмные расчётные сетки. Генерация таких сеток является отдельным, достаточно ресурсоёмким процессом, требующим наличия достаточно подробной трёхмерной геометрической модели ЛА в требуемых конфигурациях, детальная проработка которой на начальных стадиях проектирования не всегда возможна. Отдельной, не всегда простой, задачей

является и геометрическая параметризация трёхмерных сеток, особенно структурированных, без чего невозможны параметрические исследования - неотъемлемая часть начальных стадий проектирования [94].

На начальном этапе проектирования самолёта, когда информация о форме и конструкции самолёта ещё не полная, но требуются предварительные расчёты подъёмной силы, сопротивления и воздушных нагрузок для оптимизации аэродинамики и силовой схемы конструкции, возможно использовать и другие методы численного моделирования обтекания тел, обладающие существенными преимуществами в скорости вычислений и менее придирчивые к детализации геометрии. Такие преимущества возможны за счёт грамотного сочетания ряда принимаемых допущений и комбинаций расчётных моделей. К примеру, можно упомянуть МДВ или панельные методы [95, 96], способные с достаточно высокой точностью определять индуктивные характеристики несущей системы ЛА и аэродинамической компоновки в целом, включая характеристики двигателей. Данный метод не учитывает вязкое сопротивление тел и работает с несжимаемой средой. Однако, комбинация с инженерными методами расчёта АДХ позволяет с достаточной на начальных стадиях проектирования точностью определять поправки на сжимаемость среды и рассчитывать сопротивление трения. Известны работы, где вихревые и панельные методы используются в связке с решением уравнением пограничного слоя для определения вязкой составляющей сопротивления, а также известны и другие вариации использования этих методов с другими, экономными по вычислительным ресурсам, моделями для возможности учёта прочих физических процессов, включая теплообмен и сверхзвуковые течения [97, 98].

МДВ предполагает моделирование крыла как вихревую систему, распределённую по тонкой поверхности. Расчёт АДХ заключается в определении распределения циркуляций вихревой системы в каждой точке несущей системы и поля индуктивных скоростей за ней. Эти данные позволяют определять как распределённые характеристики (давление), так и

интегральные (индуктивное сопротивление, подъёмная сила). Преимуществом метода является использование так называемых поверхностных сеток, т. к. он способен предсказывать характеристики без моделирования течения около тела. Простота, а значит, быстрота генерации сеток – немаловажное преимущество метода. Высокая скорость вычислений с использованием данного метода определяется, с одной стороны, количеством уравнений системы, равное количеству элементов поверхностной сетки (на порядки меньше элементов, чем в объёмной сетке), и возможностью линеаризации расчётов: получение решения в одну итерацию. Получение одной точки на графике с помощью МДВ занимает менее половины секунды на офисном компьютере против нескольких часов на суперкомпьютере, если применяется решение уравнений Навье-Стокса.

Этот метод реализован в виде ПО «Торнадо» [99], «XFLR5» [100], «DMVWing» [101], AVL [102], OpenVSP [103].

С учётом высказанных соображений для целей данной работы представляется целесообразным использование МДВ с необходимыми доработками для расчёта нескольких несущих поверхностей.

1.5 Весовые модели

Масса конструкции составляет большую часть общей массы ЛА [104]. Кроме того, масса конструкции существенно чувствительна к изменению относительных геометрических параметров ЛА и его размерности. Последнее проявляется в законе «квадратов-кубов» и носит нелинейный характер. В связи с этим точное прогнозирование массы конструкции на начальных этапах проектирования требует отдельного внимания. Примером из практики может служить опыт ОКБ им. О. К. Антонова во время разработки самолётов Ан-22 и Ан-124. Применение автоматизации проектирования и математического моделирования позволило достичь точности расчёта массы самолёта Ан-124 в 3-4%, тогда как для Ан-22 она составляет 8-12% [105-107].

В современной практике проектирования ЛА прогнозирование массы конструкции выполняется по направлениям:

- использование так называемых весовых формул, базирующихся на детальном анализе статистики существующих конструкций в комбинации с аналитическими зависимостями из математического аппарата общей механики материалов;
- анализ силовой работы конструкций под нагрузками, соответствующими огибающей расчётных случаев, встречающихся в эксплуатации.

Первый подход был распространён до последнего времени, но и сегодня, ввиду сложности вопросов весового анализа, имеет широкое употребление в практике проектирования и сегодня [108, 109]. Преимуществом этого подхода является его надёжность, т. к. данные формулы используют опыт обобщение опыта проектирования авиационных конструкций. Одновременно данное обстоятельство является и недостатком, т. к. обработка статистического материала возможна только на базе существующих образцов конструкций и её результаты не способны дать ответы на вопросы при создании принципиально новых, потенциально прорывных, решений в авиастроении. Более того, весовые формулы, предназначенные для проектирования аппаратов одного типа, класса, размерности, но разработанные разными авторами, могут давать существенно различающиеся результаты, а совместное использование этих формул или сравнение их результатов не всегда возможно или корректно, т. к. входящие в них параметры могут быть разнородны по типу и количеству [110-112]. Данное обстоятельство придаёт процессу проектирования достаточно субъективный характер, зависящий от опыта, интуиции, убеждений разработчика, его приверженностей к определённым стандартным техническим решениям, устоявшимся традиций в проектирующем коллективе и т. д.

Достаточно большие отклонения между результатами определения веса конструкции крыла тяжелого самолета по разным формулам показаны в [110-112] (рисунок 5).

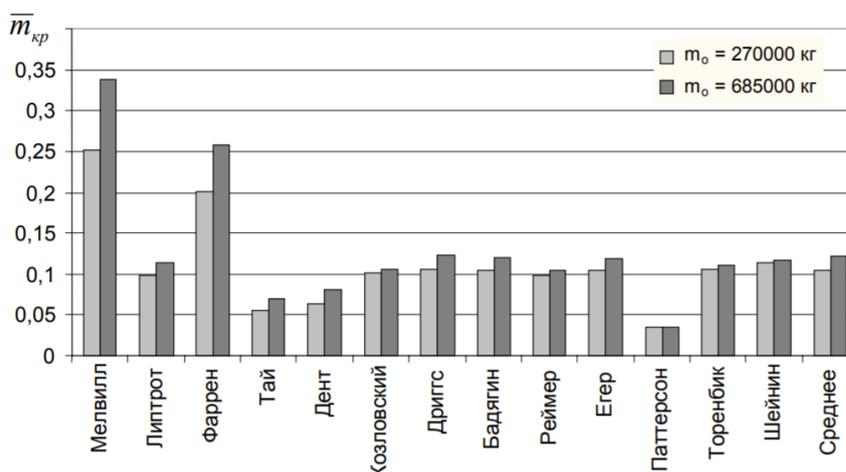


Рисунок 5 – Результаты расчётов массы конструкции крыльев по весовым формулам [112]

Для преодоления этой проблемы В. М. Шейнин предложил использовать методы множественных вычислений и двойного взвешивания [113]. Однако для правильного применения этого подхода формулы должны быть идентичны по физическому происхождению, по типу и количеству содержащихся в них параметров, а также должны включать одно и то же приближение [83].

Второй подход в определении массы авиационных конструкций базируется на силовой работе конструкции. Известны методики, предполагающие аналитические расчёты на прочность параметризованной геометрии конструкции методами строительной механики. Параметрический подбор геометрических характеристик силовых элементов конструкции обеспечивает условие прочности конструкции, а расчёт массы конструкции выполняется геометрическим способом, зная плотность материала [114, 115]. Недостаток такого подхода заключается в сложности моделирования конструкций сложных форм, включая расчёт напряженно-деформированного состояния (НДС) конструкции от распределённых нагрузок по сложным законам. Перечисленные недостатки можно преодолеть за счёт использования численных методов расчёта НДС, например, МКЭ [116, 117]. Однако, расчёт массы конструкции в этом случае выполняется только по итогу обеспечения условия прочности (устойчивости) конструкции, что предполагает итерационный процесс расчёта НДС до сходимости, варьируя

геометрическими параметрами элементов конструкции. Данный подход использован в работах [118, 119].

Отдельное внимание заслуживает подход, теоретические основы которого были разработаны А. А. и В. А. Комаровыми. Он базируется на использовании специфического интегрального критерия - *силовой фактор* (или *G-фактор*) и численном моделировании силовой работы конструкции [111, 112].

«Данный критерий выражает одновременно величину и протяжённость действия внутренних усилий в конструкции. В общем виде для нагруженной конструкции он вычисляется следующим образом:

- в аналитическом виде:

$$G = \int_V \sigma^{\text{ЭКВ}} dV, \quad (2)$$

где $\sigma^{\text{ЭКВ}}$ - эквивалентные напряжения, действующие в конструкции; V - объём материала конструкции;

- численно для конечно-элементных моделей (КЭМ):

$$G = \sum_{i=1}^n \sigma_i^{\text{ЭКВ}} V_i, \quad (3)$$

где $\sigma_i^{\text{ЭКВ}}$ - эквивалентные напряжения, действующие в i -том элементе КЭМ конструкции; V_i - объём i -того элемента МКЭ-модели конструкции[83].»

Таким образом, данный критерий позволяет определять силовые схемы конструкций, реализующие наилучшие пути передачи внутренних усилий. Использование *G-фактора* для задач оптимизации силовых схем не требует выполнения множества вычислительных итераций для подбора конкретных геометрических параметров силовых элементов с целью обеспечения условия прочности: достаточно только однократного вычисления напряжённого состояния конструкции. Данное преимущество открывает существенные возможности использования данного критерия и для оценки самой массы конструкции [83]: *«теоретически необходимый объём материала V_T полнонапряженной конструкции с определённой силовой схемой может быть оценён через G-фактор и допускаемые напряжения $[\sigma]$ путём расчёта внутренних усилий для данного распределения жёсткостей элементов через*

соотношение $V_T = \frac{G}{[\sigma]}$. Зная теоретически необходимый объём конструкции, плотность материала ρ и коэффициент полной массы φ , можно вычислить реальную (практическую или полную) массу конструкции:

$$m_k = \varphi \rho V_T = \varphi \rho \frac{G}{[\sigma]}, \quad (4)$$

или

$$m_k = \varphi \frac{G}{\bar{\sigma}}, \quad (5)$$

где $\bar{\sigma} = \frac{[\sigma]}{\rho}$ - удельная прочность материала.»

Таким образом расчёт массы конструкции определяется именно силовой работой конструкции, а не геометрическим расчётом объёма конструкции, прочность которой достигнута в процессе итерационного подбора большого количества геометрических параметров конкретных силовых элементов. В этом его ключевое преимущество перед основными методами.

«Коэффициент полной массы φ в (5) является довольно устойчивой величиной. Он учитывает прирост массы за счёт стыковочных и несиловых элементов в конструкции, отклонений от оптимального распределения материала в пользу простоты и технологичности конструкций и т. п. Его можно получать из ретроспективного анализа существующих конструкций и использовать как критерий качества детального проектирования [83].»

Основополагающими работами в данном направлении являются [120-122]. В [123, 124] раскрыты возможности этого критерия для оценки массы силовых конструкций на основе МКЭ, а также проведены детальные расчёты коэффициента φ для крыльев транспортных самолётов. Работы [112, 125] посвящены прогнозированию массы авиационных конструкций, в частности силовых панелей крыла, с использованием G -фактора [83].

Дальнейшим развитием использования критерия G является попытка сделать его безразмерным, т. к. сам G -фактор является размерной величиной [Н·м], и сравнение силовых схем различной размерности и нагруженности с помощью размерного критерия затруднено. В работе [84] предложена

модификация критерия G -фактор путём его преобразования в безразмерный коэффициент C_K , определяемый в общем виде как:

$$C_K = \frac{G}{PL}, \quad (6)$$

где P - характерная нагрузка, действующая на конструкцию; L - характерный размер конструкции.

Так, например, для крыла соотношение (6) приобретает следующий вид [84]:

$$C_K = \frac{G}{m_0 g n \sqrt{S}}, \quad (7)$$

где в качестве характерного размера принят корень квадратный из площади крыла \sqrt{S} - своего рода хорда эквивалентного по площади квадратного крыла удлинением $\lambda = 1$; характерная нагрузка - это вес ЛА, умноженный на нормальную перегрузку, то есть подъёмная сила, необходимая для уравнивания ЛА в расчётном случае А'.

С учётом (7) можно получить формулу расчёта массы крыла:

$$\bar{m}_{кр} = \frac{\varphi}{\bar{\sigma}} C_K n g \sqrt{S} = \frac{\varphi}{\bar{\sigma}} C_K n g \sqrt{\frac{m_0}{p_0}}. \quad (8)$$

Примечательно, что в формулу (8) входит коэффициент удельной прочности материала $\bar{\sigma} = \frac{\sigma}{\rho}$ - важнейшая удельная характеристика материала.

Аналогичным образом можно получить формулы для расчёта масс других частей ЛА, зная их напряжённое состояние под действием заданного спектра нагрузок.

Особенности прогнозирования массы конструкции малоразмерных и среднеразмерных БВС

Все упомянутые выше направления в области весовых расчётов авиационных конструкций хорошо применимы и для БВС, но среднекрупной и крупной размерности. Задача определения массы конструкции малоразмерных, среднеразмерных и низкоскоростных БВС имеет свои особенности. Они заключаются в том, что геометрические размеры элементов конструкции, определённые из условий прочности и устойчивости, имеют

достаточно малые толщины и площади поперечных сечений. Данное обстоятельство обусловлено малыми удельными нагрузками на крыло, что и характерно для малоразмерных и медленнолетающих аппаратов. Масса конструкции в таких случаях часто определяется условиями технологии и требований эксплуатации. Например, невозможно качественно изготовить конструкцию фюзеляжа, обшивка которого из условий прочности представляет собой один слой стеклопластика толщиной 0,16 мм. Но даже в случае изготовления подобной «ажурной» конструкции его эксплуатация затруднена, так как тонкая обшивка такого фюзеляжа будет повреждена при первых попытках его транспортировки и сборки при подготовке к полёту. При контакте фюзеляжа из тонкой обшивки с жёсткой травой или грунтом во время посадки образуются местные повреждения критического характера. Таким образом, при проектировании подобных конструкций толщины обшивок, стенок, поясов балок определяются условиями технологии производства и/или выбираются из рациональных соображений, основанных на опыте практической эксплуатации конструкций в полевых условиях. Формулы расчёта массы таких конструкций могут иметь достаточно простой вид и вычисляться геометрическим образом. Например, масса конструкции крыла малоразмерного БВС может иметь вид:

$$\bar{m}_{\text{кр}} = 2\delta n\rho\varphi \frac{1}{p_0}, \quad (9)$$

где δ - это толщина монослоя слоистого композиционного материала, из которого сделана панель крыла; n - количество слоёв укладки композита, ρ - плотность конструкционного материала; φ – коэффициент увеличения массы конструкции крыла с учётом внутреннего набора, крепежа, лючков, элементов системы управления и т. д.; p_0 - удельная нагрузка на крыло; коэффициент «2» учитывает массу верхней и нижней панели крыла.

Таким образом, определение массы конструкции БВС имеет свои особенности, которые необходимо учитывать при проектировании. Крупноразмерные БВС самолётного или вертолётного типа могут быть

рассчитаны по весовым формулам, предназначенным для пилотируемых аппаратов аналогичной размерности и взлётной массы [1, 4] или на основе критерия *силовой фактор*. Представляется также целесообразным разработку весовых формул отдельно для малоразмерных БВС, тем более что на сегодня создано значительное количество БВС подобного класса и имеется возможность собрать и проанализировать обширные статистические данные. В то же время опыт показывает, что масса конструкции малоразмерных БВС, определяемая в основном технологией производства и условиями эксплуатации (удароустойчивости), обычно зависит от абсолютных размеров (площади крыла или удельной нагрузки на крыло) (9) и в меньшей мере зависит от прочих относительных геометрических характеристик, которые учитываются в весовых формулах для крупноразмерных аппаратов из пилотируемой авиации.

В этой связи в весовом анализе необходимо использование методов определения относительной массы конструкции планера с учётом абсолютных размеров БПЛА.

1.6 Методы оптимизации

В ряде работ по параметрической оптимизации облика ЛА успешно используются поисковые методы, в частности, градиентные [126-128]. Однако они требуют вычисления частных производных целевой функции, включая уточнение расчёта взлётной массы по уравнению существования на каждой итерации. Дополнительная трудность в решении задач выбора облика состоит в разнородности и дискретности проектных переменных. Например, гибридная силовая установка может состоять из тепловых и электрических двигателей в различных сочетаниях, или ставиться задача рассмотреть разные аэродинамические схемы самолёта. Поэтому для решения таких задач развиваются методы метаэвристического поиска. Методы метаэвристики в основном имитируют процессы развития в живой природе [129-133].

В настоящее время метаэвристические алгоритмы - это мощные инструменты оптимизации, которые можно использовать для решения

сложных оптимизационных задач. Выбор алгоритма зависит от характеристик задачи и доступных вычислительных ресурсов [134].

В области самолётостроения вначале использовались относительно простые генетические алгоритмы и программы, решающие комбинаторные задачи оптимизации из однородных элементов, например, выбор ориентации и числа слоёв композиционного материала в различных участках обшивки крыла [135] или оптимизация структурных и геометрических параметров сложных композитных деталей [136, 137]. В настоящее время алгоритмы и программы этого направления интенсивно развиваются и внедряются в область предварительного проектирования авиационной техники [138-140].

В частности - алгоритмы ДЭ [141, 142], отличающиеся от известных генетических алгоритмов особенностью операции скрещивания, которая предполагает получение векторов нового поколения путём смешения генотипов каждого вектора предыдущего поколения с мутированным вектором того же поколения. Данная особенность позволяет «динамически» моделировать особенности рельефа целевой функции и эффективно преодолевать локальные овраги, обеспечивая более быструю сходимость алгоритма. Алгоритм ДЭ поддерживает разнообразие популяции посредством дифференциальной мутации, эффективно балансируя исследование новых решений и эксплуатацию известных успешных решений, тем самым снижая возможность застревания в локальных оптимумах. Кроме того, алгоритм ДЭ предлагает большую гибкость для включения адаптивных методов, использования различных форм векторного кодирования и работы с различными суррогатными методами [143]. Другим заметным преимуществом ДЭ перед другими методами является его превосходная масштабируемость для высокоразмерных задач [144-146].

Алгоритм ДЭ представляется перспективным для постановки и решения задач, связанных с оптимизацией облика БПЛА на этапе концептуального проектирования с адаптацией, направленной на учёт ряда требований к АДХ устойчивости и управляемости.

1.7 Цель и задачи исследования

На основе выполненного информационного анализа цель и задача исследования формулируются следующим образом.

Цель исследования

Разработка методики, алгоритмов и ПО быстрого выбора рациональных параметров облика новых БПЛА самолётного типа на этапах концептуального проектирования и модификации существующих аппаратов.

Задачи исследования

1. Выбор эффективного варианта эволюционного алгоритма и адаптация его к проектированию БПЛА самолётного типа.
2. Разработка комплекса проектных переменных для генерации аэродинамических схем в широком диапазоне от «нормальной» до схемы «утка», включая «тандем».
3. Разработка быстрого алгоритма вычисления взлётной массы повышенной точности с учётом требований продольной устойчивости.
4. Разработка математических моделей для быстрого расчёта АДХ различных аэродинамических схем.
5. Проведение анализа и выбора весовых формул основных частей конструкции самолёта с учётом масштабного фактора.
6. Разработка АСП_КП с использованием опубликованных и вновь разработанных программных блоков.
7. Оценка достоверности результатов проектирования в разработанной АСП_КП.

2 РАЗРАБОТКА МЕТОДА ОПТИМИЗАЦИИ СХЕМЫ И ПАРАМЕТРОВ БПЛА САМОЛЁТНОГО ТИПА

2.1 Математическая формулировка задачи в терминах нелинейного программирования

Задача выбора оптимальных параметров облика БПЛА самолётного типа рассматривается в терминах НМП, следуя [46, 56, 147, 148] и большинству работ последнего времени в данной области. В канонической форме задача НМП формулируется следующим образом [149]:

$$\begin{aligned} f(\mathbf{x}^{opt}) &\leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega, \\ \Omega &= \{\mathbf{x}: q_j(\mathbf{x}) \leq 0, h_k(\mathbf{x}) = 0\}, \end{aligned} \quad (10)$$

где $f(\mathbf{x})$ - целевая функция; $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ - вектор проектных переменных; \mathbf{x}^{opt} - оптимальное решение задачи; Ω - область допустимых проектов; $q_j(\mathbf{x}) \leq 0, j = \overline{1, p}$ - ограничения в виде неравенств; $h_k(\mathbf{x}) = 0, k = \overline{1, l}$ - ограничения в виде равенств; n, p, l - количество переменных и ограничений.

В связи с тем, что для решения задачи (10) предполагается использовать и развивать один из эволюционных алгоритмов, введём теоретико-множественные понятия «*особи*» и «*популяции*» и соответствующие обозначения.

В рассматриваемых задачах *особь* - это один из мыслимых вариантов проекта - то есть вектор \mathbf{x}_s , компонентами которого являются определённые значения проектных переменных x_i облика проектируемого ЛА [4]:

$$\mathbf{x}_s = \{x_1, x_2, \dots, x_i, \dots, x_n\}_s, \quad s = \overline{1, w},$$

где \mathbf{x}_s - вектор проектных переменных особи s в популяции \mathbf{P}_g ; s - номер особи; w - количество особей в популяции; i и n - индекс и количество проектных переменных в каждой особи s .

Популяция \mathbf{P}_g - это множество особей s на определённом этапе оптимизации (генерации) g , состоящая из особей \mathbf{x}_s :

$$\mathbf{P}_g = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_s, \dots, \mathbf{x}_w\}_g,$$

где g - номер генерации.

2.2 Выбор и определение целевой функции

Наиболее важным аспектом оптимального проектирования является выбор целевой функции. Целевая функция в целом и ЛА в частности должны отвечать следующим требованиям [45]:

- критерий должен быть измеримой (считаемой) величиной, метод расчёта которой известен;
- критерий должен учитывать основную цель, для которой создаётся объект (самолёт), а также условия и ограничения эксплуатации;
- критерий должен включать параметры и характеристики объекта, влияние которых необходимо оценить или оптимизировать;
- на каждом уровне принятия решений (на каждом этапе проектирования) критерии должны быть согласованными;
- желательно, чтобы на всех этапах проектирования этот критерий оставался единственным.

В предлагаемой методике в качестве целевой функции могут быть объявлены различные технические характеристики БПЛА, представляющие интерес для разработчиков. Однако в связи со сложностью постановки и решения задачи в достаточно общем виде далее рассматриваются особенности методики с использованием в качестве сквозных примеров задачи улучшения технических характеристик - минимизации взлётной массы m_0 - двух существующих БПЛА тяжёлого класса с различными аэродинамическими схемами: типа MQ-1 [150], с взлётной массой ~ 1000 кг и типа U-40 [151], с взлётной массой ~ 2000 кг (рисунок 6). Далее эти ЛА обозначаются как БП-1, БП-2.

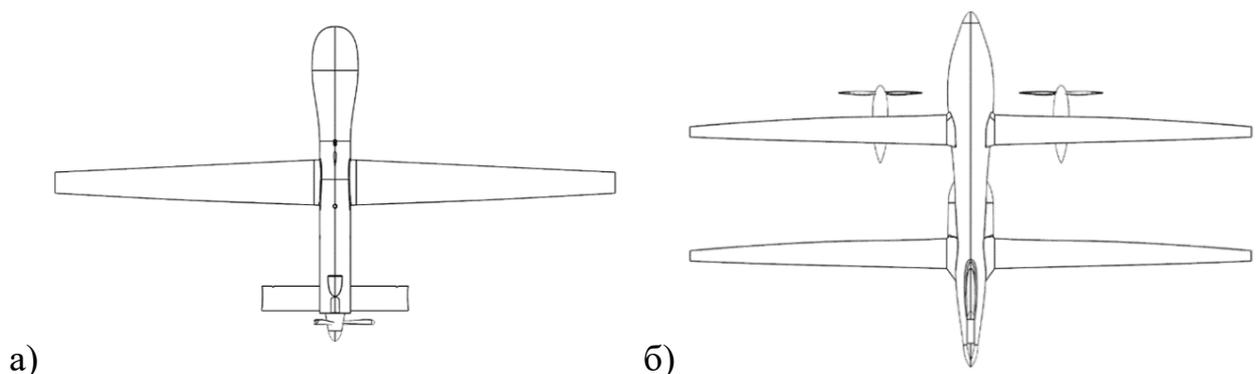


Рисунок 6 – Вид в плане рассматриваемых БПЛА тяжёлого класса:
а) БП-1; б) БП-2

В рассматриваемых задачах в качестве целевой функции в большинстве случаев выбирается взлётная масса $f(x) = m_0$. При выборе в качестве целевой функции другого показателя (энергетической или транспортной эффективности и т. д.) достаточно точный расчёт взлётной массы требуется в любом случае, так как на основе этой величины выполняется балансировка ЛА, расчёт весового и энергетического баланса и определяются абсолютные геометрические параметры [4]. С одной стороны, масса - это размерный параметр самолёта, с определения которого зачастую начинается процесс выбора и согласования остальных параметров самолёта. С другой стороны, она является важнейшей обобщённой характеристикой самолёта, выступая в качестве результата проектирования [45]. Взлётная масса ЛА зависит практически от всех параметров проекта [152]. Для её определения используется итерационное решение уравнения существования - весового баланса [37, 110, 152]:

$$m_0(x) = \frac{m_{п.н}}{1 - \bar{m}_{cy} - \bar{m}_{топ} - \bar{m}_{конс} - \bar{m}_{об}}, \quad (11)$$

где $m_{п.н}$ - масса полезной нагрузки, кг; и относительные массы: \bar{m}_{cy} - силовой установки, $\bar{m}_{топ}$ - топлива (или $\bar{m}_{акк}$ - аккумуляторных батарей в случае электрической силовой установки), $\bar{m}_{конс}$ - конструкции, $\bar{m}_{об}$ - оборудования и управления.

2.3 Выбор проектных переменных

Результаты оптимизации во многом зависят от количества, значения и физической природы проектных переменных. Различие между независимыми и зависимыми переменными в процессе проектирования также играет важную роль.

Независимые переменные - это параметры с разными диапазонами значений в процессе оптимизации, направленной на поиск наиболее подходящего самолёта для своей задачи. Характер независимых переменных будет определять, какой тип алгоритма оптимизации выбрать. Это может быть непрерывная величина в диапазоне значений, таких как геометрические параметры, скорость полёта или удельная нагрузка на крыло, это также могут

быть целочисленные переменные, такие как количество двигателей, это также могут быть дискретные переменные, такие как аэродинамическая схема самолёта. Число независимых переменных отражает свободу проектирования и называется размерностью пространства проектирования, их верхние и нижние предельные значения называются пределами допустимой области. В ряде источников такие переменные принято называть «геометрическими».

Зависимые переменные - это параметры, генерируемые в процессе проектирования (оптимизации), их значения определяются независимыми переменными. Типичные зависимые переменные, такие как АДХ, угол атаки и угол балансировочной поверхности, обеспечивающие условия балансировки самолёта, относительные массы частей конструкции и т. п.

Вектор проектных переменных x может включать в себя геометрические и удельные энергетические параметры облика ЛА, а также кинематические параметры рассматриваемых режимов полёта [4].

Особенностью данной работы является добавление в вектор x оценочной величины взлётной массы m_0^{es} . Для переменной m_0^{es} используется первый номер в векторе x . В рассматриваемых примерах использовался вектор x из 12 следующих проектных переменных:

$$x = \{m_0^{es}, \lambda_1, \chi_1, \eta_1, \delta_1, \lambda_2, \chi_2, \eta_2, \bar{L}_2, \bar{S}_2, V, p_0\},$$

где m_0^{es} - оценочная (*estimation*) величина взлётной массы, кг - *введена дополнительно* для ускорения вычислений в связи с тем, что в (11) относительные массы $\bar{m}_{су}$, $\bar{m}_{топ}$, $\bar{m}_{конс}$ зависят от величины масштабных факторов: $m_0, p_0, S_{кр}$; λ_1 - удлинение передней и λ_2 - задней несущей поверхности; χ_1 - стреловидность по передней кромке передней и χ_2 - задней несущей поверхности, °; η_1 - сужение передней и η_2 - задней несущей поверхности; δ_1 - угол заклинения передней несущей поверхности, °; $\bar{L}_2 = L/b_{САХ}$ - относительное расстояние между несущими поверхностями ($b_{САХ}$ - средняя аэродинамическая хорда (САХ), м); $\bar{S}_2 = S_2/S_1$ - относительная площадь задней несущей поверхности (S_1 - площадь передней несущей поверхности, м²); V - скорость полёта, м/с и p_0 - удельная нагрузка на крыло, кг/м².

Данный набор проектных переменных предлагается из соображений обеспечить возможности получать при их определённых цифровых значениях достаточно большое разнообразие аэродинамических схем. Особенностью предлагаемой методики является использование геометрического параметра $\bar{S}_2 = S_2/S_1$, который позволяет рассматривать дискретные аэродинамические схемы из двух несущих поверхностей непрерывно без разделения их на нормальную схему, схему «утка» или схему «тандем». Величина \bar{S}_2 является варьируемым параметром при оптимизации. В случае $\bar{S}_2 < 1$ - БПЛА получает нормальную схему, в случае $\bar{S}_2 > 1$ - схему «утка», если $\bar{S}_2 \sim 1$ - схему «тандем» (рисунок 7).

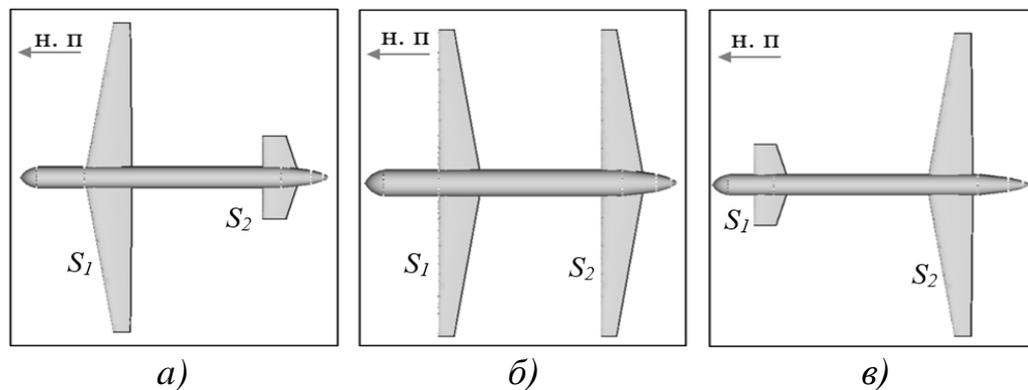


Рисунок 7 – Аэродинамические схемы ЛА в зависимости от значения относительной площади \bar{S}_2 (н. п - направление полёта):

а) нормальная схема ($0 < \bar{S}_2 < 1$); б) тандем ($\bar{S}_2 \sim 1$); в) утка ($\bar{S}_2 > 1$)

При этом выделяется главная несущая поверхность: передняя, если $S_1 > S_2$ и наоборот задняя, если $S_1 < S_2$. Нормирование аэродинамических коэффициентов и относительных геометрических характеристик осуществляется относительно САХ главной поверхности.

Кроме указанных проектных переменных, внутри общего цикла оптимизации выбираются значение угла установки задней несущей поверхности относительно продольной оси δ_2 и угла атаки α ЛА с помощью алгоритма балансировки.

Совокупность средств для вычисления значения взлётной массы ЛА при определённых значениях проектных переменных и оценки уровня выполнения ограничений представляет собой информационную модель объекта оптимизации в рассматриваемых задачах НМП.

2.4 Ограничения

2.4.1 Геометрические ограничения

Это ограничения на максимальные и минимальные значения самих проектных переменных в виде $x_i^l \leq x_i \leq x_i^u$. Эти ограничения определяют диапазон значений соответствующей переменной и называются в НМП *геометрическими*, независимо от их природы. Они могут быть линейными, угловыми, удельной нагрузкой на крыло и т. п [4] (таблица 1).

Таблица 1 – Геометрические ограничения в виде диапазонов возможных значений проектных переменных

№	Обозначение	Проектная переменная	Диапазон значения
1	m_0^{es} , кг	Оценочная величина взлётной массы	[500, 3000]
2	λ_1	Удлинение передней поверхности	[4, 20]
3	$\chi_1, ^\circ$	Стреловидность передней поверхности	[0, 15]
4	η_1	Сужение передней поверхности	[1, 3]
5	$\delta_1, ^\circ$	Угол заклинения передней поверхности	[0, 5]
6	λ_2	Удлинение задней поверхности	[4, 20]
7	$\chi_2, ^\circ$	Стреловидность задней поверхности	[0, 15]
8	η_2	Сужение задней поверхности	[1, 3]
9	\bar{L}_2	Относительное расстояние между двумя поверхностями	[2, 6]
10	\bar{S}_2	Относительная площадь	[0,2, 5]
11	V , м/с	Скорость	[30, 90]
12	p_0 , кг/м ²	Удельная нагрузка на крыло	[5, 110]

2.4.2 Функциональные ограничения

Ограничения в виде неравенств $q_j(\mathbf{x}) \leq 0$ и равенств $h_k(\mathbf{x}) = 0$ определяют технические требования к функционированию ЛА [4]. Как правило, их невозможно выразить в явном виде через проектные переменные. Для оценки выполнения функциональных ограничений используются специальные алгоритмы, в ряде случаев достаточно сложные, например, МДВ в аэродинамике, статистические данные или критерий «силовой фактор» [110] в весовых расчётах и т. п.

В данной работе используются следующие [1]:

- в виде равенства - условие балансировки БПЛА в вертикальной плоскости с заданным запасом статической устойчивости [153]:

$$h_1(\mathbf{x}) = m_z(\mathbf{x}) = 0, \quad (12)$$

$$h_2(\mathbf{x}) = c_{y_{a_{\text{бал}}}}(\mathbf{x}) - c_{y_a}(\mathbf{x}) = 0; \quad (13)$$

- в виде неравенства:

- ограничение на максимальное значение коэффициента подъёмной силы:

$$q_1(\mathbf{x}) = c_{y_a}(\mathbf{x}) - c_{y_a}^* \leq 0; \quad (14)$$

- ограничение на величину коэффициента статического момента горизонтального оперения для обеспечения требуемых характеристик управляемости БПЛА, оказывающее влияние на величину плеча горизонтального оперения и его относительной площади [154]:

$$q_2(\mathbf{x}) = A_{\text{ГО}}(\mathbf{x}) - A_{\text{ГО}_{\text{макс}}} \leq 0; \quad (15)$$

$$q_3(\mathbf{x}) = A_{\text{ГО}_{\text{мин}}} - A_{\text{ГО}}(\mathbf{x}) \leq 0. \quad (16)$$

Здесь m_z - коэффициент продольного момента относительно центра масс; c_{y_a} - коэффициент подъёмной силы; $c_{y_{a_{\text{бал}}}}$ - коэффициент подъёмной силы в состоянии балансировки; $c_{y_a}^*$ - допустимый коэффициент подъёмной силы на заданном режиме полёта; $A_{\text{ГО}}$ - коэффициент статического момента горизонтального оперения.

Эти типы ограничений могут взаимодействовать друг с другом, что приводит к усложнению решения задач. Для уменьшения сложности задачи, расширения возможностей использования алгоритмов оптимизации, а также упрощения условий останковки и обеспечения устойчивости алгоритма задачи оптимизации часто преобразуются к форме без ограничений. Наиболее популярными методами являются метод штрафных функций [155] и метод обобщённых множителей Лагранжа [156].

2.5 Многодисциплинарное математическое моделирование на этапе концептуального проектирования

Уравнение существования самолёта (11) позволяет учитывать одновременно отдельные критерии аэродинамической и весовой эффективности, а рассчитанное по этому уравнению значение взлётной массы может выступать непосредственно как целевая функция или использоваться в качестве основы для расчёта других показателей.

Относительные массы в (11) зависят от многих проектных переменных. Особенно это касается $\bar{m}_{\text{топ}}$ и $\bar{m}_{\text{сy}}$, которые зависят от аэродинамического качества K и энерговооружённости \bar{N} ЛА. Поэтому, для целей данной работы предлагается специальный алгоритм вычисления уточнённой (выходной) взлётной массы $m_0^{\text{out}}(\mathbf{x})$, учитывающий требование обеспечения продольной устойчивости самолёта (рисунок 8).

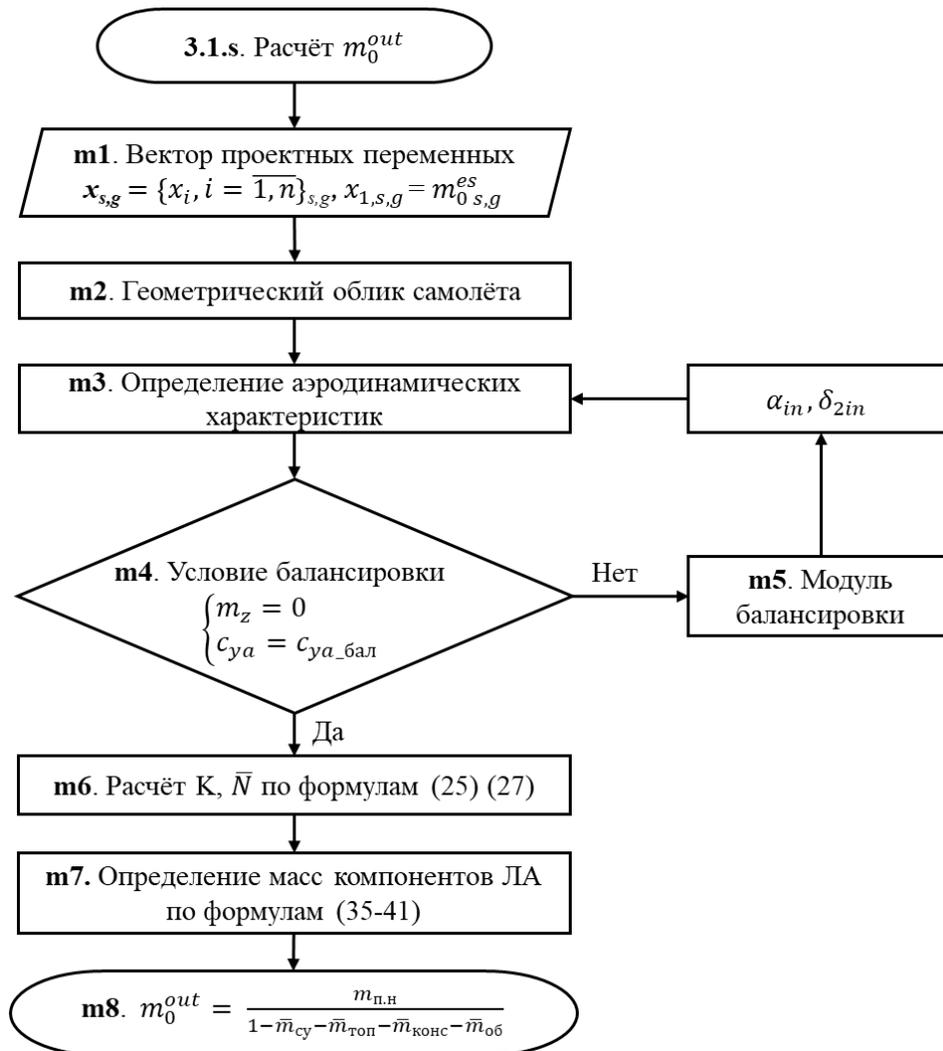


Рисунок 8 – Блок-схема алгоритма расчёта взлётной массы на основе уравнения существования (11) учёта обеспечения продольной балансировки БПЛА

Блок m_1 - получает на вход исходные данные и вектор проектных переменных особи \mathbf{x} , включая оценочную величину взлётной массы m_0^{es} .

Блок m_2 - служит для вычисления абсолютных геометрических характеристик ЛА на основе оценочной величины взлётной массы m_0^{es} и удельной нагрузки на несущую систему p_0 , значение которой находится в векторе проектных переменных рассматриваемой особи \mathbf{x} .

Геометрические характеристики используются в алгоритме для:

- автоматизированной генерации трёхмерных геометрических моделей БПЛА;
- генерации численных моделей расчёта АДХ БПЛА;
- применения инженерных формул аэродинамики учёта сжимаемости и вязкого трения;
- расчёта массы силовой конструкции планера БПЛА.

Блок m_3 - выполняет расчёт АДХ БПЛА с целью определения аэродинамического качества.

Блок m_4 - выполняет проверку условия равновесия аппарата в вертикальной плоскости.

Блок m_5 - реализует процесс балансировки БПЛА путём выбора угла атаки α и угла установки δ_2 балансирующей поверхности с учётом заданного запаса статической продольной устойчивости.

Блок m_6 - выполняет расчёт потребных энергетических характеристик силовой установки и потребного количества энергоносителя на всех этапах полёта.

Блок m_7 - выполняет расчёт масс основных частей БПЛА.

Блок m_8 - выполняет расчёт выходного значение взлётной массы $m_0^{out}(x)$ с использованием уравнения существования (11) на основе значений параметров x_i и оценочной величины взлётной массы m_0^{es} .

2.5.1 Геометрические характеристики

Вычисление геометрических характеристик (рисунок 8, Блок **m2**) особи производится на основе оценочной величины взлётной массы m_0^{es} , удельной нагрузки на несущую систему p_0 и других относительных геометрических параметров из вектора особи x .

Суммарная площадь несущих поверхностей БПЛА S_Σ определяется по формуле:

$$S_\Sigma = \frac{m_0^{es}}{p_0}, \quad (17)$$

где m_0^{es} - оценочная величина взлётной массы, кг; p_0 - удельная нагрузка на несущую систему, кг/м².

Остальные абсолютные геометрические характеристики каждой несущей поверхности находятся по соотношениям:

$$S_{\Sigma} = S_1 + S_2; \quad (18)$$

$$l_j = \sqrt{\lambda_j \cdot S_j}; \quad (19)$$

$$b_{kj} = \frac{2S_j}{l_j(1 + \eta_j)}; \quad (20)$$

$$b_{0j} = b_{kj} \cdot \eta_j; \quad (21)$$

$$b_A = \frac{2}{3} b_0 \frac{1 + \eta + \eta^2}{\eta + \eta^2}, \quad (22)$$

где S_1 - площадь передней несущей поверхности крыла, м²; S_2 - площадь задней несущей поверхности, м²; l - размах крыла, м; λ - удлинение крыла; j - индекс ($j = 1$ или 2); b_0 - корневая хорда крыла, м; b_k - концевая хорда крыла, м; η - сужение крыла; b_A - средняя аэродинамическая хорда, м.

Геометрия фюзеляжа описывается относительными и абсолютными параметрами: удлинением фюзеляжа λ_{ϕ} ; удлинением носовой и хвостовой части фюзеляжа $\lambda_{н.ч}$, $\lambda_{х.ч}$; эквивалентным диаметром миделева сечения $d_{\phi}^{\text{э}}$, м; удельной нагрузкой на мидель $p_{0м}$, кг/м².

2.5.2. Аэродинамические модели

Расчёт АДХ БПЛА (рисунок 8, Блок **м3**) осуществляется для аэродинамической компоновки БПЛА, полученной для каждой особи x_s из популяции P_g . Определение несущих свойств и индуктивной составляющей лобового сопротивления выполняется с помощью МДВ [95, 96, 157] с использованием открытого ПО AVL [102], а учет сжимаемости и расчёт сил вязкого трения выполняется на основе инженерных методов.

Основная идея метода МДВ основана на замене крыла П-образной вихревой системой, равномерно распределенной по хорде (n вихрей) и по размаху крыла (m вихрей) (рисунок 9).

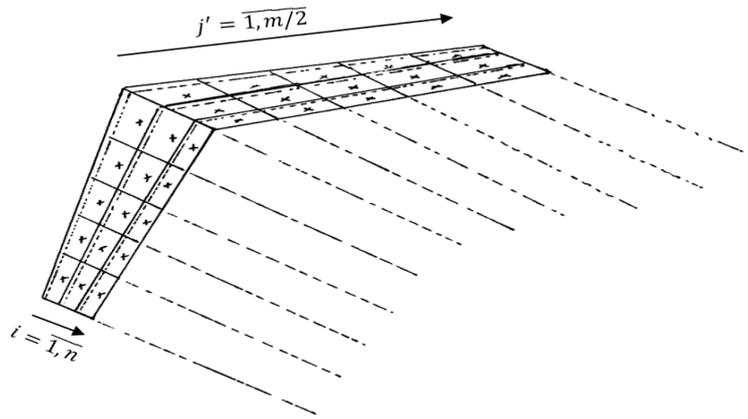


Рисунок 9 – Пример вихревой модели крыла [96]

На основе набора проектных переменных рассматриваемых задачах определяются геометрические параметры модели самолёта и с учётом задаваемых сеточных характеристик выполняется моделирование самолёта в ПО AVL (рисунок 10).

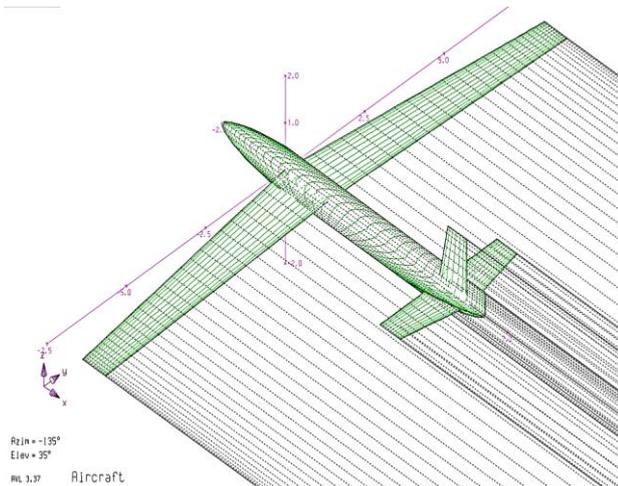


Рисунок 10 – Сеточная модель самолёта в ПО AVL

Далее в AVL определяется интенсивность циркуляций Γ_i^α по сечениям крыла из решения системы линейных алгебраических уравнений [158]:

$$\sum_{i=1}^n (w_{y i,j} + \Delta w_{y i,j}) \Gamma_{i,j} = -2\pi, \quad (j = \overline{1, m}), \quad (23)$$

где $w_{y i,j}$ – безразмерная скорость, индуцированная вихрем с номером i в контрольной точке с номером j ; $\Delta w_{y i,j}$ – дополнительная безразмерная скорость, индуцированная в контрольной точке j зеркальным вихрем, симметричным вихрю i .

На основе решения системы (23) определяются безразмерные интегральные АДХ крыла: C_y , C_{xi} , m_z .

Определение сопротивления при нулевой подъёмной силе

На дозвуковых скоростях сопротивление при нулевой подъёмной силе c_{x0} в основном создаётся профильным сопротивлением. Профильное сопротивление самолёта находят суммой сопротивлений изолированных его частей с учётом интерференции между ними, определяемая по следующей формуле [159]:

$$c_{x0} = c_{xp} = c_{xp\phi} \bar{S}_{\phi} + k_T^* c_{xp\kappa} \bar{S}_{\kappa} + k_T c_{xp\Gamma.o} \bar{S}_{\Gamma.o} + c_{xp\text{в.о}} \bar{S}_{\text{в.о}} + k_T^* \Delta c_{xp\kappa(\phi)} \bar{S}_{\kappa} + k_T \Delta c_{xp\Gamma.o(\phi)} \bar{S}_{\Gamma.o}. \quad (24)$$

Здесь $c_{xp\phi}$, $c_{xp\kappa}$, $c_{xp\Gamma.o}$, $c_{xp\text{в.о}}$ - коэффициенты профильного сопротивления изолированных фюзеляжа, крыла, горизонтального и вертикального оперения; $\Delta c_{xp\kappa(\phi)}$, $\Delta c_{xp\Gamma.o(\phi)}$ - коэффициенты дополнительного профильного сопротивления, обусловленного интерференцией крыла и горизонтального оперения с фюзеляжем; k_T^* , k_T - коэффициенты торможения потока перед крылом и горизонтальным оперением; \bar{S}_{ϕ} , \bar{S}_{κ} , $\bar{S}_{\Gamma.o}$, $\bar{S}_{\text{в.о}}$ - отношения площадей миделя фюзеляжа, изолированного крыла, горизонтального и вертикального оперения к характерной площади S . Определение компонентов в формуле (24) подробно описано в работе [159].

Определение аэродинамического качества и энерговооружённости

Основная задача используемого блока моделей аэродинамики - это определение аэродинамического качества на каждом из этапов полёта с целью оценки потребных энергетических характеристик силовой установки и количества энергоносителя, определяемых потребной энерговооружённостью на различных этапах полёта.

Аэродинамическое качество, определяющее энергетические затраты на каждом из этапов полёта, определяется соотношением соответствующих этому режиму полёта коэффициентов подъёмной силы c_{ya} и лобового сопротивления c_{xa} .

Аэродинамическое качество определяется по формуле:

$$K = \frac{Y_a}{X_a} = \frac{c_{ya}}{c_{xa}} = \frac{c_{ya}}{c_{x0} + c_{xi}}, \quad (25)$$

где c_{ya} - коэффициент подъёмной силы; c_{xi} - коэффициент индуктивного сопротивления, эти два параметра получаются с помощью ПО AVL [102]; c_{x0} - коэффициент сопротивления при нулевой подъёмной силе, который рассчитывается по эмпирическим формулам [159].

Энерговооружённость определяется через систему уравнений равновесия самолёта [160] (рисунок 11).

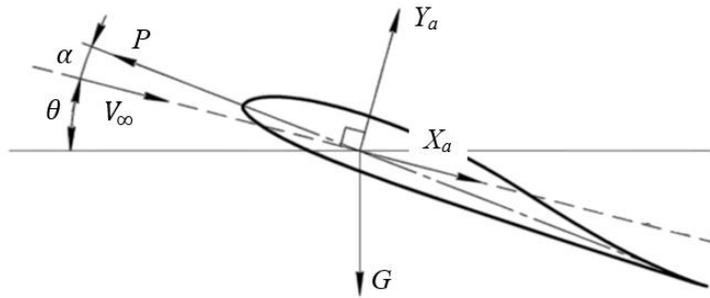


Рисунок 11 – Силы, действующие на БПЛА

$$\begin{cases} P \cdot \cos\alpha - X_a - G \cdot \sin\theta = 0 & (a) \\ Y_a - G \cdot \cos\theta + P \cdot \sin\alpha = 0 & (б) \end{cases} \quad (26)$$

где P - сила тяги силовой установки, Н; X_a - сила сопротивления, Н; G - вес БПЛА, Н; Y_a - подъёмная сила, Н; α - угол атаки, °; θ - угол наклона траектории полёта, °.

Из (26) энергетический баланс обеспечивается на каждом из этапов полёта выражением:

$$\bar{N} = \frac{V}{\eta_{\text{вин}}} \cdot \frac{K \sin\theta + \cos\theta}{\sin\alpha + K \cos\alpha}, \quad (27)$$

где V - скорость полёта, м/с; $\eta_{\text{вин}}$ - коэффициент полезного действия (КПД) винтов.

2.5.3 Динамические модели

Продольная статическая балансировка БПЛА (рис. 8, блок m5)

Определение аэродинамического качества и потребных энергетических характеристик БПЛА осуществляется при обеспечении балансировки БПЛА в вертикальной плоскости, то есть при выполнении условия:

$$\begin{cases} M_z = 0 & (a) \\ Y_a = Y_{a_{\text{бал}}} & (б) , \\ X_a = P \cos\alpha + G \sin\theta & (в) \end{cases} \quad (28)$$

где M_z - продольный аэродинамический момент относительно центра масс; Y_a - подъёмная сила; $Y_{a_{\text{бал}}}$ - подъёмная сила, необходимая для балансировки по оси y_a в скоростной системе координат; X_a - сила лобового аэродинамического сопротивления; P - сила тяги силовой установки; G - вес БПЛА.

Значение коэффициента подъёмной силы $c_{y_{a_{\text{бал}}}}$, необходимое для обеспечения балансировки по оси y_a , определяется по формуле:

$$c_{y_{a_{\text{бал}}}} = \frac{g \cdot p_0 \cdot \cos \theta}{\frac{1}{2} \cdot \rho \cdot V^2}, \quad (29)$$

где g - ускорение свободного падения, м/с²; p_0 - удельная нагрузка на крыло, кг/м²; θ - угол наклона траектории, °; ρ - плотность воздуха, кг/м³; V - скорость полёта, м/с.

Коэффициент подъёмной силы c_{y_a} БПЛА зависит от угла атаки и углового расположения несущих поверхностей относительно друг друга, а коэффициент продольного момента относительно центра масс зависит ещё и от выбранной степени продольной устойчивости.

Таким образом задача обеспечения балансировки (28) представляет собой задачу по обеспечению равенств (28а) и (28б) с двумя переменными:

$$\begin{cases} m_z(\alpha, \delta_2) = 0 & \text{(а)} \\ c_{y_a}(\alpha, \delta_2) - c_{y_{a_{\text{бал}}}} = 0 & \text{(б)} \end{cases}, \quad (30)$$

где α - угол атаки, °; δ_2 - угол установки балансирующей аэродинамической поверхности, °.

Используемые модели аэродинамики определяют линейный характер зависимостей в (30), поэтому в работе предлагается использовать следующий алгоритм с использованием аналитических методов с целью повышения быстродействия решения данной задачи по следующим шагам:

а) расчёт характеристик: подъёмной силы c_{y_a} и продольного аэродинамического момента относительно носка САХ главной несущей поверхности m_{zA} численным методом МДВ в ПО AVL относительно носка САХ главной несущей поверхности для двух разных сочетаний угла атаки α и угла установки задней несущей поверхности δ_2 ;

б) на основе полученных данных выполняется линейная аппроксимация аналитических зависимостей $c_{ya}(\alpha, \delta_2) = c_{ya}^\alpha \cdot (\alpha - \alpha_0 + \delta_2)$ и $m_{zA}(\alpha, \delta_2) = m_{z0} + m_{zA}^{c_{ya}} \cdot c_{ya}(\alpha, \delta_2)$;

в) вычисляется положение аэродинамического фокуса по углу атаки относительно носка САХ главной несущей поверхности $\bar{x}_\phi = |m_{zA}^{c_{ya}}|$ и рассчитывается требуемое положение центра масс, которое обеспечит заданный запас продольной статической устойчивости: $\bar{x}_{ц.м} = \bar{x}_\phi + (\bar{x}_{ц.м} - \bar{x}_\phi)$, где $(\bar{x}_{ц.м} - \bar{x}_\phi) = \Delta$ - требуемый запас продольной статической устойчивости;

г) производится пересчёт зависимости $m_{z_{ц.м}}(\alpha, \delta_2)$ относительно требуемого положения центра масс (рисунок 12):

$$m_{z_{ц.м}} = m_{z0} + \Delta \cdot c_{ya} = m_{z0} + \frac{m_{zA} - m_{z0}}{m_{zA}^{c_{ya}}} \cdot \Delta \quad (31)$$

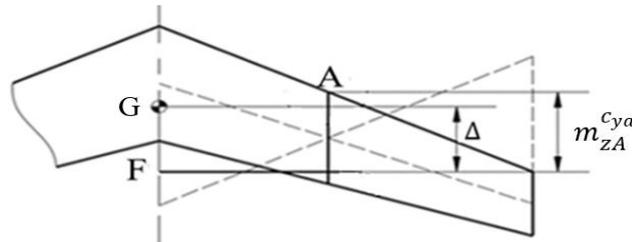


Рисунок 12 – Определение коэффициента продольного момента относительно центра масс

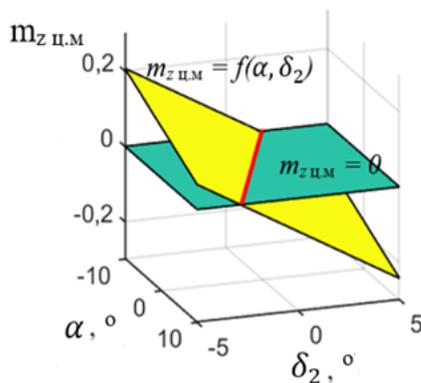


Рисунок 13 – Зависимость $m_{z_{ц.м}}$ от α и δ_2

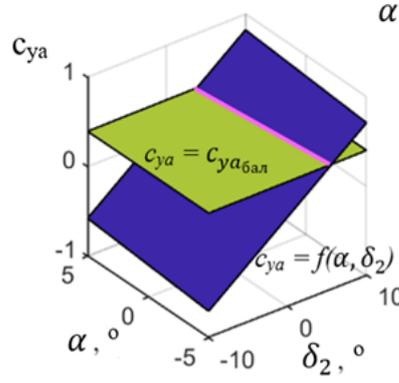


Рисунок 14 – Зависимость c_{ya} от α и δ_2

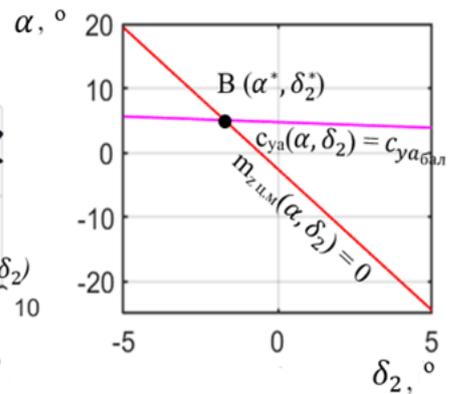


Рисунок 15 – Решение условия балансировки БПЛА

д) находятся аналитические выражения для двух прямых, полученные пересечением плоскости $c_{ya} = f(\alpha, \delta_2)$ с плоскостью $c_{ya} = c_{ya_{бал}}$ и плоскости

$m_{z_{ц.м}} = f(\alpha, \delta_2)$ с плоскостью $m_{z_{ц.м}} = 0$, то есть находятся решения (30а) и (30б) независимо друг от друга (рисунки 13-15);

е) задача (30) решается аналитически. Решением является точка $B = (\alpha^*, \delta_2^*)$ пересечения двух прямых $c_{ya}(\alpha, \delta_2) = c_{ya_{бал}}$ и $m_{z_{ц.м}}(\alpha, \delta_2) = 0$ (рисунок 15);

ж) выполняется расчёт величин $c_{ya}(\alpha^*, \delta_2^*)$ и $m_{z_{ц.м}}(\alpha^*, \delta_2^*)$ на основе моделей метода МДВ в ПО AVL для проверки условия балансировки.

Таким образом, задача (30) решается графоаналитическим методом.

Условие устойчивости

Для исследования устойчивости рассматриваются схемы сил, действующих на самолёт с нормальной схемой (рисунок 16) и схемой «утка» (рисунок 17).

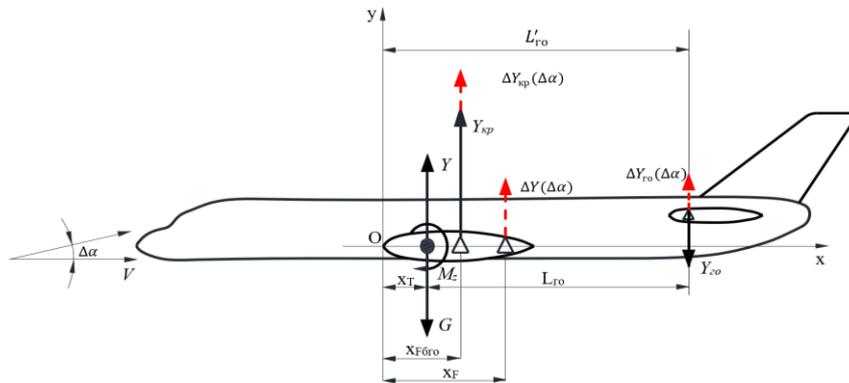


Рисунок 16 – «Нормальная» балансировочная схема самолёта с хвостовым горизонтальным оперением [45]

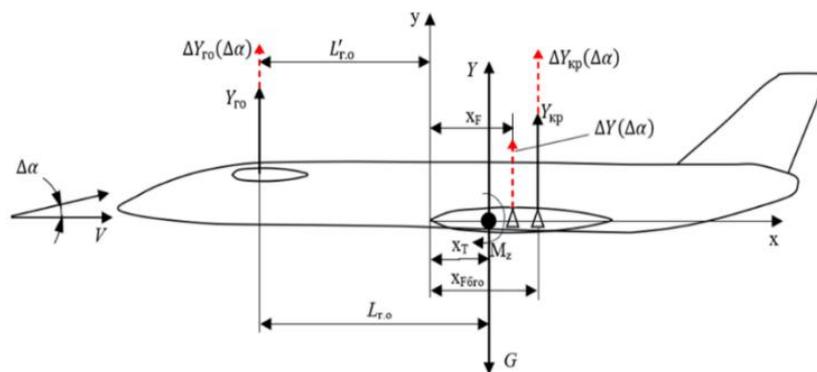


Рисунок 17 – Балансировочная схема самолёта с передним горизонтальным оперением (схема «утка») [45]

Предположим, что угол атаки увеличивается из-за какого-либо возмущения, подъёмная сила изолированного крыла и подъёмная сила изолированного горизонтальном оперения увеличиваются на величину $\Delta Y_{кр}(\Delta\alpha)$, $\Delta Y_{го}(\Delta\alpha)$ соответственно.

Изменение силы и момента представлено через соотношения:

$$\Delta Y(\Delta\alpha) = \Delta Y_{кр}(\Delta\alpha) + \Delta Y_{го}(\Delta\alpha) \quad (32)$$

$$\Delta M_z = \Delta Y(\Delta\alpha)(x_T - x_F) \quad (33)$$

Для обеспечения продольной статической устойчивости изменение продольного момента в уравнении (33) должно иметь отрицательное значение, т. е. приводить к пикированию самолёта. Это означает:

$$(x_T - x_F) < 0, \quad (34)$$

т. е. аэродинамический фокус самолёта находится позади центра тяжести самолёта.

В данной работе условие (34) обеспечивается выбором $\Delta = (\bar{x}_T - \bar{x}_F) = -0,1$ на основе рекомендаций [45, 46].

2.5.4 Весовые модели

Масса энергоносителя

В случае использования БПЛА ДВС (или другого типа двигателя на топливе) относительная масса топлива определяется по формуле [1]:

$$\bar{m}_{\text{топ}} = \bar{N}C_e t, \quad (35)$$

где \bar{N} - энерговооружённость, кВт/кг; C_e - удельный расход топлива, кг/(кВт.ч); t - время полёта, ч. Далее, для самолётов, летающих долго, предлагается более точный расчёт $\bar{m}_{\text{топ}}$ (раздел 3.4, с. 82).

В случае использования БПЛА с электрическим двигателем относительная масса аккумуляторных батарей будет равна [1]:

$$\bar{m}_{\text{акк}} = \frac{g\bar{N}t}{E\eta_{\text{cy}}}, \quad (36)$$

где η_{cy} - КПД силовой установки; E - удельная энергоёмкость аккумулятора, кВт.ч/кг; g - ускорение свободного падения, м/с².

Масса силовой установки

Масса силовой установки включает в себя массу воздушных винтов и двигателя с системами обеспечения работоспособности - топливной системой для ДВС и контроллерами электрических двигателей в случае их использования в качестве силового агрегата. Относительная масса двигателя

вычисляется по формуле:

$$\bar{m}_{cy} = k\bar{N}\gamma, \quad (37)$$

где k - коэффициент, учитывающий увеличение массы силовой установки за счёт систем; γ - удельная масса двигателя.

Масса конструкции

Предсказание в процессе проектирования ЛА абсолютных и относительных масс отдельных частей и всей конструкции в целом представляет собой достаточно трудную задачу. Так как на эти параметры влияет множество факторов различной физической природы: режимы полёта и перегрузки, удельная прочность материалов, абсолютные размеры и массы, конструктивно-силовое и технологическое совершенство. Поэтому в практическом самолетостроении находят широкое применение так называемые «*весовые формулы*» [43-53, 113, 161], полученные, как правило, на основе обработки статистических данных по построенным и успешно летающим машинам.

В данной работе в целях ускоренной разработки и испытания алгоритма МДО использовались следующие весовые формулы для вычисления относительных масс частей конструкции БПЛА рассматриваемого класса:

- масса крыла рассчиталась по (38) из [46]:

$$\bar{m}_{кр} = 1,15 \cdot 10^{-4} k_{мех} k_{кон} k_{мт} \varphi n_p \frac{\lambda \sqrt{S}}{(\cos \chi)^{1,5} \sqrt{\theta \bar{c}_0}} \frac{\eta + 4}{\eta + 1} \left(1 - \frac{\bar{c}_0 - 1}{\bar{c}_к + 3} \right), \quad (38)$$

где $k_{мех}$ - коэффициент, учитывающий механизацию крыла; $k_{кон}$ - коэффициент, учитывающий тип конструкции крыла; $k_{мт}$ - коэффициент, учитывающий марку основного материала конструкции крыла; φ - коэффициент разгрузки крыла; n_p - расчётная перегрузка; λ - удлинение крыла; S - площадь крыла, м²; χ - стреловидность крыла, °; θ - коэффициент, учитывающий эффективность работы продольных силовых элементов; \bar{c}_0 - корневая относительная толщина крыла; $\bar{c}_к$ - концевая относительная толщина крыла; η - сужение крыла.

- масса конструкции фюзеляжа определялась по формуле (39) из [45]:

$$\bar{m}_\phi = \frac{0,23}{m_0} \sqrt{V_{\max} \frac{l_\phi}{b_\phi + h_\phi} S_{\text{полн}}^{1,2}}, \quad (39)$$

где V_{\max} - предельная расчётная скорость, м/с; l_ϕ - длина фюзеляжа, м; b_ϕ - максимальная ширина фюзеляжа, м; h_ϕ - максимальная высота фюзеляжа, м; $S_{\text{полн}}$ - полная площадь внешней поверхности фюзеляжа, м².

- масса вертикального оперения определялась по формуле (40) из [161]:

$$\bar{m}_{\text{в.о}} = 6,8 \frac{S_{\text{в.о}}^{1,2}}{m_0} \left[0,4 + \frac{(V_{\text{крейс}} + 113)}{1100} \right], \quad (40)$$

где $S_{\text{в.о}}$ - площадь вертикального оперения, м²; $V_{\text{крейс}}$ - крейсерская скорость, м/с.

- масса шасси определялась путём сложения масс передней и основных стоек, которые определяются отдельно по формуле (41) из [45]:

$$\bar{m}_{\text{ш}} = k_{g\text{ш}} \left(\frac{A}{m_0} + \frac{B}{m_0^{\frac{1}{4}}} + C + D m_0^{\frac{1}{2}} \right), \quad (41)$$

где $k_{g\text{ш}}$ - коэффициент, учитывающий положение крыла относительно фюзеляжа; A, B, C, D - статистические коэффициенты и способы определения их значений можно найти в [45].

В данной работе из всего многообразия весовых формул, которые приводятся в публикациях упомянутых авторов, выбраны формулы, учитывающие явно такие параметры, как перегрузка n , площадь крыла S , взлётная масса m_0 , удельная нагрузка на крыло p_0 и безразмерные геометрические параметры, используемые в данной работе как проектные переменные.

На выбор формул (38-41) повлиял также анализ весовых формул (6-8) Комарова [84, 110, 111] для крыла (раздел 1.5, с. 30):

$$\bar{m}_{\text{кр}} = \frac{\varphi}{\sigma} C_K n g \sqrt{S} = \frac{\varphi}{\sigma} C_K n g \sqrt{\frac{m_0}{p_0}}, \quad (8)$$

которые получены с использованием методов строительной механики [121, 122]. Эти формулы учитывают в явном виде: удельную прочность основного

конструкционного материала, перегрузку n , минимальную величину безразмерного критерия силового совершенства конструкций крыла C_K , достижимую внутри выделенного объёма для её размещения [84, 110] и конструктивно-технологическое совершенство с использованием коэффициента полной массы φ . Описание и алгоритм получения этого коэффициента на основе МКЭ анализа реальных конструкций, представлено в [85, 112].

Псевдокод алгоритма (рисунок 8) представлен в модуле 1 [4].

Модуль 1: Определение взлётной массы m_0^{out}

```

1  Вводить:  $m_0^{es}$ ,  $\lambda_1$ ,  $\chi_1$ ,  $\eta_1$ ,  $\lambda_2$ ,  $\chi_2$ ,  $\eta_2$ ,  $\bar{L}_2$ ,  $\bar{S}_2$ ,  $V$ ,  $p_0$ ;
2  Создать входные файлы ПО AVL для определения АДХ;
3  Определить  $\alpha_{бал}$  и  $\delta_{2бал}$ ;
4   $c_{xв}$ ,  $c_{yв}$ ,  $m_z = runAVL(m_0^{es}, \lambda_1, \chi_1, \eta_1, \lambda_2, \chi_2, \eta_2, L, \bar{S}_2, V, p_0, \alpha_{бал}, \delta_{2бал})$ ;
5  Определить  $c_{x0}$  по эмпирическим формулам;
6  Определить качество  $K$  согласно (32);
7  Определить  $\bar{m}_{конс}$  согласно (38-41);
8  Определить  $X_a$ ;
9  Определить  $P$  согласно (28);
10 // Определить компоненты массы на этапах набора высоты, крейсера
    и спуска
11 Определить  $\bar{N}$  согласно (27);
12  $\bar{m}_{cy\_i}$  согласно (37);
13 if type_motor == 'eltr' then
14      $\bar{m}_{акк\_i}$  согласно (36);
15      $\bar{m}_{акк} = \sum_{i=1}^3 \bar{m}_{акк\_i}$ ;
16 else
17      $\bar{m}_{топ\_i}$  согласно (35);
18      $\bar{m}_{топ} = \sum_{i=1}^3 \bar{m}_{топ\_i}$ ;
19 end if;
20  $m_{вин\_i}$ ;

```

- 21 $\bar{m}_{cy} = \max(\bar{m}_{cy_i});$
- 22 $m_{вин} = \max(m_{вин_i});$
- 23 Определить m_0^{out} согласно (11);

2.6 Методика выбора оптимальных значений проектных переменных БПЛА самолётного типа

Предлагаемая методика базируется на использовании дифференциального эволюционного алгоритма оптимизации «Success-History based Adaptive Differential Evolution» (SHADE) [162] с применением штрафных функций [163] и численного математического моделирования. Используемый метод оптимизации SHADE предполагает преобразование информации в популяциях особей. SHADE - это разновидность алгоритма ДЭ, разработанная для повышения производительности путём адаптации с помощью параметров настройки. Используя показатели успешности различных параметров настроек в предыдущих расчётах, SHADE более эффективно направляет процесс поиска. Здесь можно видеть определённую аналогию с использованием *вектора памяти* в алгоритмах случайного поиска с самообучением [85]. В задачах оптимизации с ограничениями функции штрафа используются таким образом, что исходные задачи преобразуются их в задачи без ограничений. Эти функции налагают штрафы на решения, которые нарушают ограничения, направляя поиск в сторону допустимых областей. Благодаря этому процесс поиска отдаёт приоритет допустимым решениям, эффективно исследуя пространство решений с нарушением.

Методика данной работы использует алгоритм [141], но имеет ряд существенных доработок, позволяющих [4]: рассматривать различные аэродинамические схемы БПЛА с двумя несущими поверхностями (нормальная схема, утка, тандем); проектировать БПЛА разных размерностей;

- применять на БПЛА различные типы силовых установок; повышать производительность расчётов за счёт применения аналитического аппарата и параллельных вычислений.

Методика реализована на платформе Python с подключением открытого

кода AVL для аэродинамических расчётов [102]. Блок-схема методики представлена на рисунке 18 [4].

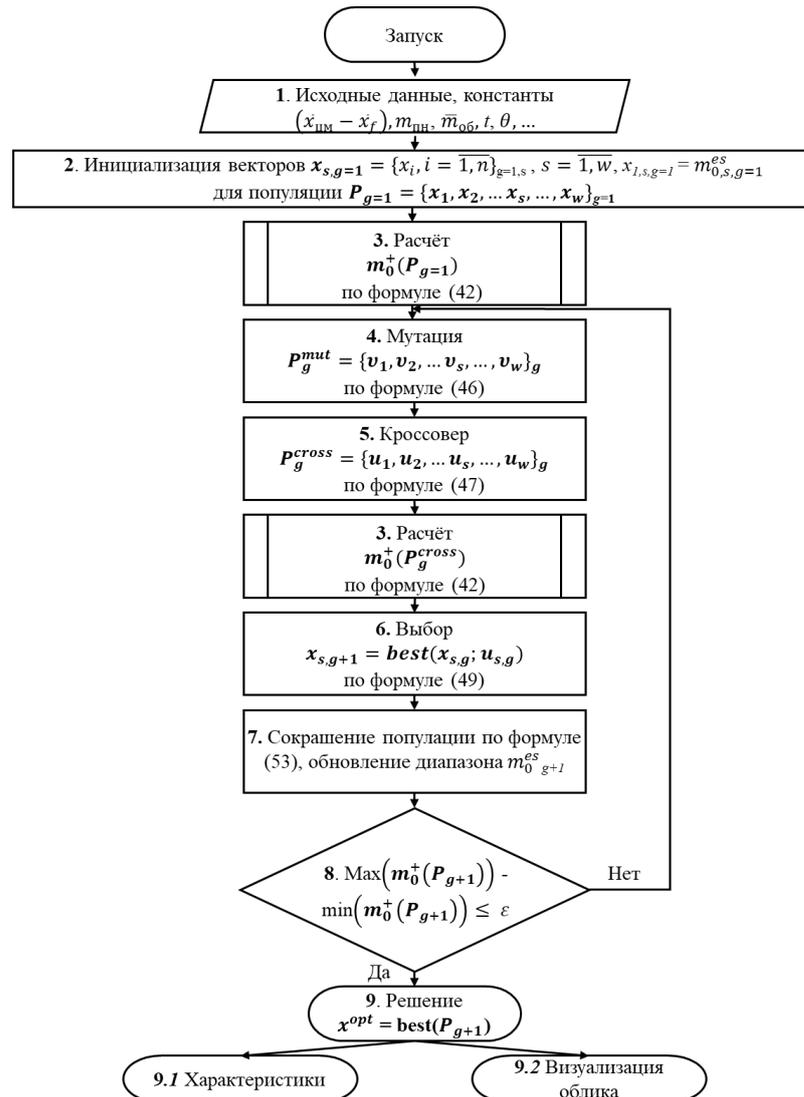


Рисунок 18 – Блок-схема алгоритма предлагаемой методики оптимизации проектных переменных БПЛА [4]

Блок 1 - процесс ввода исходных данных: полётного задания (полезной нагрузки $m_{пн}$, времени барражирования t , угла наклона траектории θ , запаса по продольной устойчивости и т.п.); введение настроек метода оптимизации (выбор размера первой популяции w , назначение величины критерия остановки ε , задание диапазонов варьирования значений проектных переменных $[x_i^l, x_i^u]$, задание значений ограничений $q_j \leq 0$, а также задание диапазона оценочной величины взлётной массы особей на первой итерации $[m_{0g=1}^{es}(\min), m_{0g=1}^{es}(\max)]$).

Блок 2 - процесс инициализации первой популяций $P_{g=1}$, состоящих из

векторов особей \mathbf{x}_s . Выбор значений проектных переменных $x_{i,g=1}$ в векторах каждой особи $\mathbf{x}_{s,g=1}$ осуществляется случайным образом из заданного пользователем диапазона значений проектных переменных $[x_i^l, x_i^u]$ с помощью метода латинского гиперкуба Latin Hypercube Sampling (LHS) [164], включая и оценочную величину взлётной массы $m_{0s,g=1}^{es}$ с учётом заданного диапазона $[m_{0g=1}^{es}(\min), m_{0g=1}^{es}(\max)]$. Размер первой популяции должен быть не менее $w_{g=1} = 10 \times n$, где n - это количество проектных переменных.

Блок 3 - выполняется расчёт взлётной массы для каждой особи $\mathbf{x}_{s,g}$ через уравнение существования [37, 110, 152] в варианте (11) с использованием итерационной процедуры (рисунок 8), учитывающей требование обеспечения продольной устойчивости самолёта.

Полученное значение взлётной массы m_0^{out} дополняется величиной штрафной функции $\psi(\mathbf{x}_{s,g})$ для особи $\mathbf{x}_{s,g}$ по (42), (43).

$$m_0^+(\mathbf{x}_{s,g}) = \begin{cases} m_0^{out}(\mathbf{x}_{s,g}) & \text{если } \psi(\mathbf{x}_{s,g}) = 0 \\ R \cdot \psi(\mathbf{x}_{s,g}) + U^* & \text{если } \psi(\mathbf{x}_{s,g}) > 0 \cap m_0^{out}(\mathbf{x}_{s,g}) \leq U^* \\ R \cdot \psi(\mathbf{x}_{s,g}) + m_0^{out}(\mathbf{x}_{s,g}) & \text{если } \psi(\mathbf{x}_{s,g}) > 0 \cap m_0^{out}(\mathbf{x}_{s,g}) > U^* \end{cases}, \quad (42)$$

где

$$\psi(\mathbf{x}_{s,g}) = \psi_1(\mathbf{x}_{s,g}) + \psi_2(\mathbf{x}_{s,g}); \quad (43)$$

$$\psi_1(\mathbf{x}_{s,g}) = \begin{cases} 0 & \text{если } c_{ya}(\mathbf{x}_{s,g}) \leq c_{ya}^* \\ c_{ya}(\mathbf{x}_{s,g}) - c_{ya}^* & \text{если } c_{ya}(\mathbf{x}_{s,g}) > c_{ya}^* \end{cases}; \quad (44)$$

$$\psi_2(\mathbf{x}_{s,g}) = \begin{cases} 0 & \text{если } A_{го}(\mathbf{x}_{s,g}) \in [A_{гоmin}, A_{гоmax}] \\ A_{го}(\mathbf{x}_{s,g}) - A_{гоmax} & \text{если } A_{го}(\mathbf{x}_{s,g}) > A_{гоmax} \\ A_{гоmin} - A_{го}(\mathbf{x}_{s,g}) & \text{если } A_{го}(\mathbf{x}_{s,g}) < A_{гоmin} \end{cases}; \quad (45)$$

ψ – безразмерная штрафная функция; U^* - верхнее ограничение возможного значения взлётной массы; R - параметр усиления штрафа, согласующий размерности и порядки величин штрафной функции с учётом класса ЛА по взлётной массе. Величина этого параметра выбирается экспертным путем с учётом взлётной массы. На основании оценок, приведённых в [163], с учётом чувствительности параметров ограничений к целевой функции были выбраны $U^* = 60000$ и $R = 100$.

Уравнения (44) и (45) анализируют выполнение ограничений по значению коэффициента подъёмной силы и коэффициента статического момента горизонтального оперения, соответственно [5].

Более подробное описание вычисления штрафной функции можно найти в [141, 163].

Результатом работы блока 3 является вектор значений целевых функций $m_0^+(P_g)$ каждой особи x_s в популяции P_g (рисунок 19).

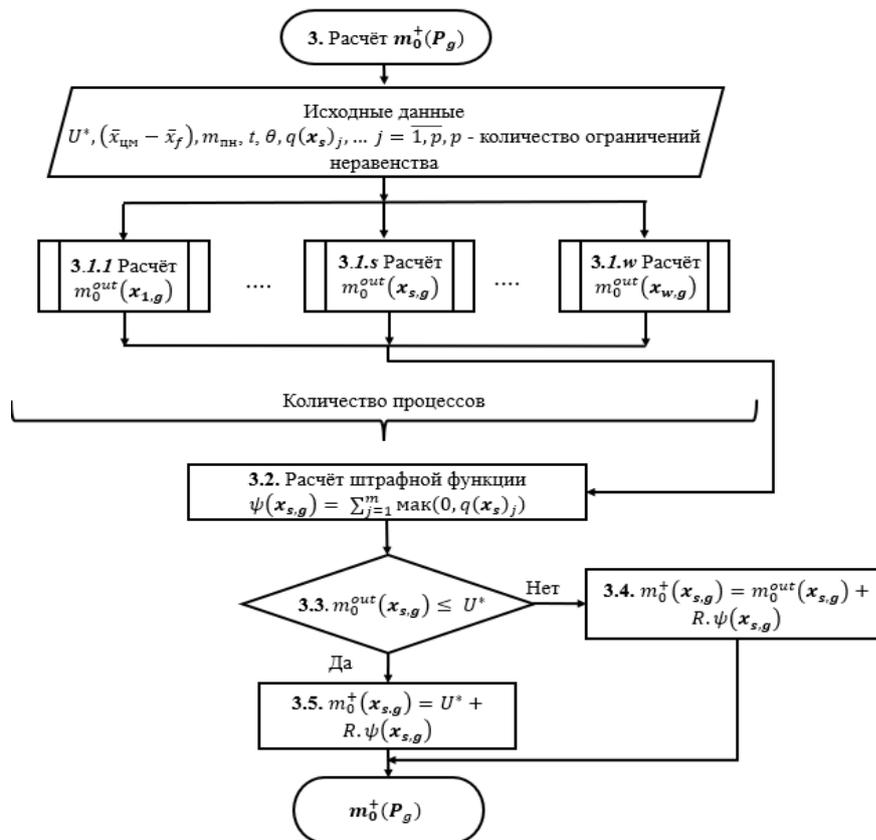


Рисунок 19 – Блок вычисления целевой функции $m_0^+(x_{s,g})$ для всех особей x_s популяции P_g

Применение метода ДЭ позволяет выполнять процесс расчёта $m_0^{\text{out}}(x_{s,g})$ с помощью параллельных вычислений (рисунок 19, блоки 3.1.s) с использованием библиотеки Joblib [165] с целью повышения производительности, так как векторы проектных переменных каждой особи не зависят друг от друга.

Блоки 4-7 отвечают за процесс генерации новых популяций. В данной работе для этих целей используются алгоритмы [162].

Новая популяция особей $x_{s,g+1}$ на каждом последующем шаге

оптимизации P_{g+1} формируется на основе отбора лучших особей $x_{s,g}$ из предыдущей популяций $P_g = \{x_1, x_2, \dots, x_s, \dots, x_w\}_g$ и популяции кроссовера $P_g^{cross} = \{u_1, u_2, \dots, u_s, \dots, u_w\}_g$, которая получается путём скрещивания популяции P_g и мутантной популяции $P_g^{mut} = \{v_1, v_2, \dots, v_s, \dots, v_w\}_g$.

Блок 4 - Мутация

Популяция P_g^{mut} включает в себя мутантные векторы особей $P_g^{mut} = \{v_1, v_2, \dots, v_s, \dots, v_w\}_g$, каждый из которых вычисляется по формуле:

$$v_s = x_s + F_s(x_{pbest} - x_s) + F_s(x_{r1} - x_{r2}) \quad (46)$$

где x_{pbest} - случайно выбранный вектор из группы лучших векторов популяции P_g . Группа лучших векторов формируется по принципу минимального значения целевой функции $m_0^+(x_{g,s})$, а размер группы лучших векторов определяется настройками алгоритма; x_{r1} - любой случайный вектор из популяции P_g , кроме самого себя x_s ; x_{r2} - любой случайный вектор из объединённой популяции P_g и архива худших решений A ; F_s - коэффициент масштабирования.

Блок 5 - Кроссовер

Векторы популяции кроссовера $P_g^{cross} = \{u_1, u_2, \dots, u_s, \dots, u_w\}_g$, $u_s = (u_1, u_2, \dots, u_i, \dots, u_n)_s$ образуется путём скрещивания значений мутантных векторов $v_s = (v_1, v_2, \dots, v_i, \dots, v_n)_s$ популяции $P_g^{mut} = \{v_1, v_2, \dots, v_s, \dots, v_w\}_g$ и векторов $x_s = (x_1, x_2, \dots, x_i, \dots, x_n)_s$ текущей популяции $P_g = \{x_1, x_2, \dots, x_s, \dots, x_w\}_g$.

Скрещивание осуществляется случайным образом по условию:

$$u_{i,s} = \begin{cases} v_{i,s} & \text{если } U[0,1] \leq CR_s \text{ или } i = i_{rand} \\ x_{i,s} & \text{наоборот} \end{cases}, \quad (47)$$

где: CR_s - значение скорости кроссовера; $U[0,1]$ - диапазон случайных чисел.

В случае, если после операции мутации и кроссовера новые значения проектных переменных какой-нибудь особи выйдут за заданные пределы варьирования, они будут переименованы по условию:

$$u_i = \begin{cases} \min(x_i) & \text{если } u_i < \min(x_i) \\ \max(x_i) & \text{если } u_i > \max(x_i) \end{cases} \quad (48)$$

Блок 6 - Выбор особей и формирование новой популяции

Выполняется расчёт значений целевых функций векторов кроссовера $m_0^+(\mathbf{u}_{s,g})$ с помощью блока 3 (рисунок 18) и сравнение их значений со значениями целевых функций $m_0^+(\mathbf{x}_{s,g})$ векторов \mathbf{x}_s текущей популяции \mathbf{P}_g . Выбор особей новой популяции \mathbf{P}_{g+1} для следующего шага оптимизации осуществляется по условию:

$$\mathbf{x}_{s,g+1} = \begin{cases} \mathbf{u}_{s,g} & \text{если } m_0^+(\mathbf{u}_{s,g}) \leq m_0^+(\mathbf{x}_{s,g}) \\ \mathbf{x}_{s,g} & \text{наоборот} \end{cases} \quad (49)$$

Соответственно значения целевых функций, штрафных функций и входных значений взлётной массы каждой особи будут получены для новой популяции по условиям:

$$m_0^{\text{out}}(\mathbf{x}_{s,g+1}) = \begin{cases} m_0^{\text{out}}(\mathbf{u}_{s,g}) & \text{если } m_0^+(\mathbf{u}_{s,g}) \leq m_0^+(\mathbf{x}_{s,g}) \\ m_0^{\text{out}}(\mathbf{x}_{s,g}) & \text{наоборот} \end{cases} \quad (50)$$

$$m_0^+(\mathbf{x}_{s,g+1}) = \begin{cases} m_0^+(\mathbf{u}_{s,g}) & \text{если } m_0^+(\mathbf{u}_{s,g}) \leq m_0^+(\mathbf{x}_{s,g}) \\ m_0^+(\mathbf{x}_{s,g}) & \text{наоборот} \end{cases} \quad (51)$$

$$\psi(\mathbf{x}_{s,g+1}) = \begin{cases} \psi(\mathbf{u}_{s,g}) & \text{если } m_0^+(\mathbf{u}_{s,g}) \leq m_0^+(\mathbf{x}_{s,g}) \\ \psi(\mathbf{x}_{s,g}) & \text{наоборот} \end{cases} \quad (52)$$

На данном этапе осуществляется также и формирование архива худших особей, которые используются при мутации на последующих итерациях оптимизации:

$$\mathbf{A}_{g+1} = \mathbf{A} + \mathbf{x}_{s,g} \text{ если } m_0^+(\mathbf{u}_{s,g}) \leq m_0^+(\mathbf{x}_{s,g})$$

При этом формируется множество векторов особей, значение штрафной функции у которых равно нулю $\psi = 0$ по следующему условию:

$$\mathbf{Mxf}_g = \emptyset + \mathbf{x}_{s,g+1} \text{ если } \psi(\mathbf{x}_{s,g+1}) = 0$$

Блок 7 - Сокращение популяции

Сходимость целевой функции обеспечивается сужением диапазона оценочной величины взлётной массы $[m_0^{\text{es}}(\text{min}) \dots m_0^{\text{es}}(\text{max})]$ на каждом последующем шаге оптимизации согласно условию:

$$m_{0g+1(\min)}^{es} = \min(\mathbf{Mxf}_g)$$

$$m_{0g+1(\max)}^{es} = \max(\mathbf{Mxf}_g)$$

То есть новые границы $[m_{0g+1(\min)}^{es}, m_{0g+1(\max)}^{es}]$ для следующего поколения определяются самыми удачными особями ($\psi = 0$).

Если же множество особей с $\psi = 0$ на предыдущем шаге оптимизации пустое $\mathbf{Mxf}_g = \emptyset$ (то есть все $\psi(x_{g,s}) > 0$), то новый диапазон $[m_{0(\min)}^{es} \dots m_{0(\max)}^{es}]$ остаётся таким же, как и в предыдущем поколении:

$$m_{0g+1(\min)}^{es} = m_{0g(\min)}^{es},$$

$$m_{0g+1(\max)}^{es} = m_{0g(\max)}^{es}.$$

При этом на каждом шаге оптимизации происходит сокращение численности популяции за счёт исключения наихудших особей с самыми большими значениями целевой функции m_0^+ ($\psi \gg 0$). Количество оставшихся особей с наименьшим значением m_0^+ на каждом новом шаге оптимизации вычисляется по формуле (53):

$$w_{g+1} = \text{round} \left[w_0 \left(\frac{w_{\min}}{w_0} \right)^{\frac{w_g}{w_{\max}}} \right], \quad (53)$$

где w_{\min} - минимальная популяция; w_0 - начальная популяция; w_{\max} - максимальное количество оцениваемых функций; w_g - текущее количество оцениваемых функций.

Если размер новой популяции будет больше, чем количество особей с $\psi = 0$, то популяция будет дополнена особями с наименьшими значениями целевой функции из тех, у которых $\psi \neq 0$.

Оценочная величина взлётной массы следующего поколения для всякой особи $m_{0s,g+1}^{es}$ является продуктом алгоритма мутации и скрещивания векторов популяции на предыдущем шаге. В случае, если эти действия привели к выходу оценочной величины взлётной массы у особи нового поколения $m_{0s,g}^{es}$ за границы нового диапазона $[m_{0g+1(\min)}^{es}, m_{0g+1(\max)}^{es}]$, то данной особи присваивается граничное значение из данного диапазона.

Таким образом особи со штрафом $\psi \neq 0$ будут постепенно устраняться в

соответствии с процессом сокращения численности популяции, а границы $[m_0^{es}(\min), m_0^{es}(\max)]$ будут сужаться до достижения сходимости уравнения существования $m_0^{es} \approx m_0^{out} = m_0^+$.

Блок 8 - проверяет условие сходимости $\max(m_0^+(P_{g+1})) - \min(m_0^+(P_{g+1})) \leq \varepsilon$. В случае невыполнения данного условия выполняются блоки алгоритма 3-7 до его сходимости.

Схематично сходимость процесса оптимизации показана на рисунке 20.

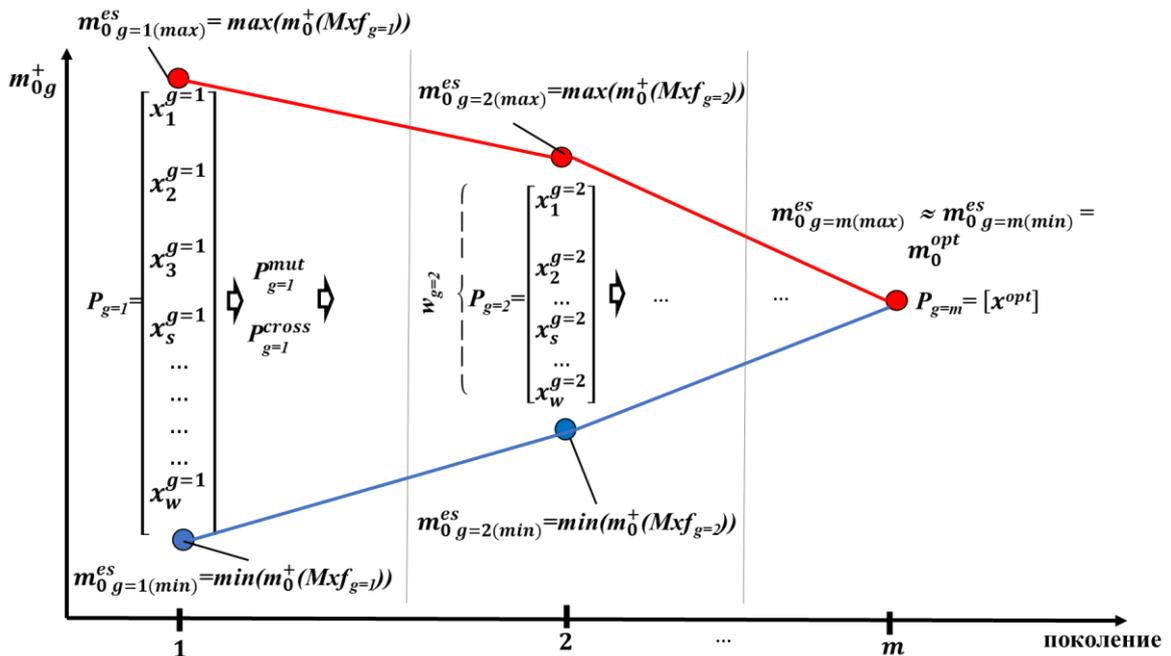


Рисунок 20 – Схема процесса сходимости целевой функции [4]

Блок 9 - выполняет выбор наилучшей особи из итоговой популяции x^{opt} и соответствующее ему значение целевой функции $m_0^+(x^{opt})$ (в итоге $m_0^+(x^{opt}) = m_0^{out^{opt}}$).

Блок 9.1 и 9.2 - дополнительные результаты

Кроме оптимальных значений проектных переменных результатом работы алгоритма является ряд дополнительных зависимых значений параметров проекта, имеющих важное значение для сравнения полученного результата с альтернативными решениями и для проведения дальнейших этапов проектирования, в том числе это массы частей БПЛА (2.5.4), лётно-технические, энергетические, аэродинамические характеристики (2.5.2, 2.5.3) и облик (рисунок 18, блоки 9.1-9.2).

Предлагаемая методика написана на языке Python на основе ПО Spyder с использованием ПО AVL.

Псевдокод алгоритма оптимизации (рисунок 18) представлен в модуле 2 [4].

Модуль 2: Алгоритм оптимизации

```

1 // Фаза инициализации
2 Вводить постоянные проектирования;
3 Вводить интервал переменных проектирования  $[x^l, x^u]$ ;
4 Вводить параметры алгоритма оптимизации  $w_0, w_{min}, w_{max}, \varepsilon, U^*, H, p, g = 1$ ;
5 Инициализировать начальную популяцию  $\mathbf{P}_{g=1}$ , используя  $LHS$ ;
6 // Параллельный цикл с помощью joblib
7 for  $s = 1$  to  $w_0$  do
10     Определить взлётную массу с модулем 1;
11 end for;
12 Определить  $\psi(\mathbf{x}_{g=1})$  согласно (43) и  $m_0^+(\mathbf{x}_{g=1})$  согласно (42);
13 Обновить  $U^*$ ;
14 Сохранить  $m_0^+(\mathbf{x}_{g=1})$ ;
15 Сохранить данные поколения  $g$ ;
16 Установить все значения в  $M_{CR}, M_F$  на 0,5;
17 Архив  $\mathbf{A} = \emptyset$ ;
18 Индексный счётчик  $g = 2$ ;
19 Индексный счётчик  $k = 1$ ;
20 // Основной цикл
21 while Критерии прекращения не соблюдены do
22      $S_{CR} = \emptyset, S_F = \emptyset$ ;
23     for  $s = 1$  to  $w$  do
24          $r_s =$  выбрать из  $[1, H]$  случайным образом;
25         Определить  $F_{s,g}$ ; определить  $CR_{s,g}$ ;
26         Определить вектор мутации  $\mathbf{v}_{s,g}$ ;
27         Определить пробный вектор  $\mathbf{u}_{s,g}$ ;
28     end for;
29     // Параллельный цикл с помощью joblib
30     for  $s = 1$  to  $w$  do
31         Определить взлётную массу с модулем 1;
32     end for;
33     Определить  $\psi(\mathbf{x}_{s,g})$  согласно (43) и  $m_0^+(\mathbf{x}_{s,g})$  согласно (49);
34     Обновить  $U^*$ ;
35     for  $s = 1$  to  $w$  do
36         if  $m_0^+(\mathbf{u}_{s,g}) \leq m_0^+(\mathbf{x}_{s,g})$  then
37              $\mathbf{x}_{s,g+1} = \mathbf{u}_{s,g}$ ;
38              $\mathbf{x}_{s,g} \rightarrow \mathbf{A}$ ;
39              $CR_{s,g} \rightarrow S_{CR}, F_{s,g} \rightarrow S_F$ ;
40         else

```

```

41          $x_{s,g+1} = x_{s,g}$ ;
42     end if;
43 end for;
44 // Обновить  $M_{CR,k}$ ,  $M_{F,k}$  на основе  $S_{CR}$ ,  $S_F$ ;
45 if  $S_{CR} = \emptyset$  and  $S_F = \emptyset$  then
46     if  $M_{CR,k,g} = -1$  or  $\max(S_{CR}) = 0$  then
47          $M_{CR,k,g+1} = -1$ ;
48     else
49          $M_{CR,k,g+1} = \text{mean}_{WL}(S_{CR})$ ;
50          $M_{F,k,g+1} = \text{mean}_{WL}(S_F)$ ;  $k++$ ;
51     end if;
52     if  $k > H$  then
53          $k = 1$ ;
54     end if;
55 else
56      $M_{CR,k,g+1} = M_{CR,k,g}$ ;
57      $M_{F,k,g+1} = M_{F,k,g}$ ;
58 end if;
59 Сохранить  $m_0^+(x_{s,g+1})$  ;
60 // Обновить диапазон  $m_{0(g+1)}^{in}$ 
61  $Mxf = []$ ;
62 For  $s = 1$  to  $w$  do
63     if  $\psi(x_{s,g+1}) = 0$  then
64          $Mxf = \text{append}(Mxf, x_{s,g})$ 
65     end if;
66 end for;
67 if  $\text{size}(Mxf) = 0$  then
68      $m_{0g+1}^{es}[0] = m_{0g}^{es}[0]$ ;
69      $m_{0g+1}^{es}[1] = m_{0g}^{es}[1]$ ;
70 else
71      $m_{0g+1}^{es}[0] = \min(Mxf)$ ;
72      $m_{0g+1}^{es}[1] = \max(Mxf)$ ;
73 end if;
74 if  $g \geq 3$  then
75     Определить  $w_{g+1}$ ;
76      $(w_g - w_{g+1})$ -й худший вектор  $\rightarrow A$ ;
77     Удалить  $(w_g - w_{g+1})$ -й худший вектор из  $P_{g+1}$ ;
78 end if;
79 Сохраните данные поколения  $(g+1)$ ;
80 if  $(\max(m_0^+(P_{g+1})) - \min(m_0^+(P_{g+1})))$  then
81     break;
82 end if;

```

```

83      $g++;$ 
84 end while;
85 Вывод  $x^{opt}$ ,  $m_0^+(x^{opt});$ 

```

2.7 Визуализация облика в виде плановой проекции и трёхмерной модели

После завершения цикла оптимизации создаётся файл в формате **.xlsx*, содержащий все геометрические параметры последней генерации.

В конце основной программы оптимизации добавляется модуль для запуска программы FreeCAD [166] через пакет Python.

Построение 3D-моделей осуществляется с помощью файлов макросов, написанных на языке Python. Этот файл макроса создаёт 3D-модель на основе геометрических параметров, содержащихся в файле **.xlsx*, сохраняет изображение модели в формате *.png*, закрывает программу FreeCAD после выполнения задачи и помещает в папку установки FreeCAD. Благодаря функции «запускать макрос при запуске» FreeCAD автоматически запускается файл макроса, и весь процесс происходит в фоновом режиме в системе.

2.8 Выбор настроечных параметров оптимизационного алгоритма

2.8.1 Контрольный параметр p (по-английски – “greediness”)

В SHADE каждая особь x_s будет связана со значением p_s , заданным уравнением:

$$p_s = rand[p_{min}, 0,2], \quad (54)$$

где p_{min} устанавливается таким образом, чтобы при выборе p_{best} особей выбиралось больше, чем 2 особи, т. е. $p_{min} = 2/w$; значение 0,2 - это максимальное значение p_s , предложенное Чжаном и Сандерсоном [167].

2.8.2 Размер глубины памяти H

Если величина H не большая, это означает, что часто используются последние значения параметров (поскольку старые значения быстро перезаписываются из-за ограниченного объёма памяти), что приводит к быстрой сходимости контрольных параметров. С другой стороны, когда величина H большая, ожидается, что скорость сходимости контрольных параметров замедлится, поскольку старые параметры будут продолжать оказывать влияние в течение более длительных периодов времени. Заметим,

что в случайном поиске с самообучением для этих целей используется вектор памяти и коэффициент забывания с величиной меньше единицы, на который умножается вектор памяти каждой итерации [148]. Как предложено в [162], выбор $H = 100$ хорошо влияет на производительность SHADE.

2.8.3 Коэффициент масштабирования F и коэффициент кроссовера CR

SHADE сохраняет историческую память M_F и M_{CR} для контрольных параметров F и CR соответственно, как показано в таблице 2 [162].

Таблица 2 – Глубина памяти для самоадаптивных контрольных параметров [162]

Индекс	1	2	...	$H - 1$	H
M_F	$M_{F,1}$	$M_{F,2}$...	$M_{F,H-1}$	$M_{F,H}$
M_{CR}	$M_{CR,1}$	$M_{CR,2}$...	$M_{CR,H-1}$	$M_{CR,H}$

H - размер исторической памяти, устанавливаемый пользователем.

В каждом поколении контрольные параметры CR_s и F_s , используемые каждой особью x_s , генерируются путём сначала выбора случайного индекса r_s из $[1; H]$, затем применяются приводимые ниже уравнения:

$$CR_s = \begin{cases} 0 & \text{если } M_{CR,r_s} = -1 \\ randN_s(M_{CR,r_s}, 0, 1) & \text{наоборот} \end{cases}, \quad (55)$$

$$F_s = randC_s(M_{F,r_s}, 0, 1), \quad (56)$$

где $randN_s(M, 0, 1)$ и $randC_s(M, 0, 1)$ - случайные величины, выбранные согласно нормальному распределению Гаусса и распределению Коши со средним значением M , дисперсия $0, 1$.

В случае, если значение сгенерированного CR_s находится за пределами $[0, 1]$, оно заменяется предельным значением (0 или 1), наиболее близким к сгенерированному значению. Когда $F_s > 1$, F_s переназначается на 1, а когда $F_s \leq 0$, уравнение (55) повторяется до тех пор, пока не будет сгенерировано допустимое значение. Значения CR_s и F_s , используемые успешными индивидами, записываются в S_{CR} и S_F , а в конце генерации память обновляется следующим образом [165]:

$$M_{CR,k,g+1} = \begin{cases} C_{WL}(S_{CR}) & \text{если } S_{CR} \neq \emptyset \\ M_{CR,k,g} & \text{наоборот} \end{cases}; \quad (57)$$

$$M_{F,k,g+1} = \begin{cases} C_{WL}(S_F) & \text{если } S_F \neq \emptyset \\ M_{F,k,g} & \text{наоборот} \end{cases}, \quad (58)$$

где

- индекс k ($1 \leq k \leq N$) определяет место в памяти, которое необходимо обновить. В начале поиска k инициализируется значением 1, k увеличивается всякий раз, когда в историю вставляется новый элемент. Если $k > N$, k устанавливается равным 1. В поколении g обновляется k -й элемент в памяти.

В уравнениях обновления (57) и (58) следует отметить, что когда всем индивидам в поколении g не удаётся создать тестируемый вектор, который лучше исходного вектора, т. е. $S_{CR} = S_F = \emptyset$, память не обновляется;

- $C_{WL}(S)$ - средневзвешенное значение Лемера, определенное по формуле (59):

$$C_{WL}(S) = \frac{\sum_{k=1}^{|S|} w_k \cdot S_k^2}{\sum_{k=1}^{|S|} w_k \cdot S_k}, \quad (59)$$

$$w_k = \frac{\Delta F_k}{\sum_{k=1}^{|S|} \Delta F_k},$$

$$\Delta F_k = |F(u_{g,k}) - F(x_{g,k})|.$$

2.8.4 Размер начальной популяции w

Как и другие популяционные алгоритмы, w играет важную роль в производительности ДЭ. Большие размеры популяций могут увеличить разнообразие популяций и помочь ДЭ одновременно отбирать образцы на большом количестве территорий. Согласно [168], если размер популяции невеликий, алгоритм может быстро сходиться; но вероятность ранней сходимости и стагнации может быть выше. Стагнация происходит практически без видимых причин. Когда наступает стагнация, популяция остаётся разнообразной и ещё не сходится, но процесс оптимизации уже не прогрессирует [169]. На ранней сходимости популяции теряют разнообразие. Вместо этого, взятие большой популяции требует больше вычислительных усилий. Согласно предложению [168] размер популяции определяется не менее $10 \times n$.

2.9 Программная реализация алгоритма оптимизации проектных переменных БПЛА

Алгоритмы и разработанные модули программы оптимизации

реализованы на языке программирования Python совместно с интерпретатором Spyder. Python - это язык программирования высокого уровня с открытым исходным кодом, совместимый с большинством различных типов операционных систем. Python отличается простым синтаксисом и широким набором встроенных библиотек, поддерживающих многие научные области.

Для удобства использования и модификации программа построена по модульному принципу. Каждый модуль имеет отдельную функцию и запускается при каждом вызове основной программы (рисунок 21).

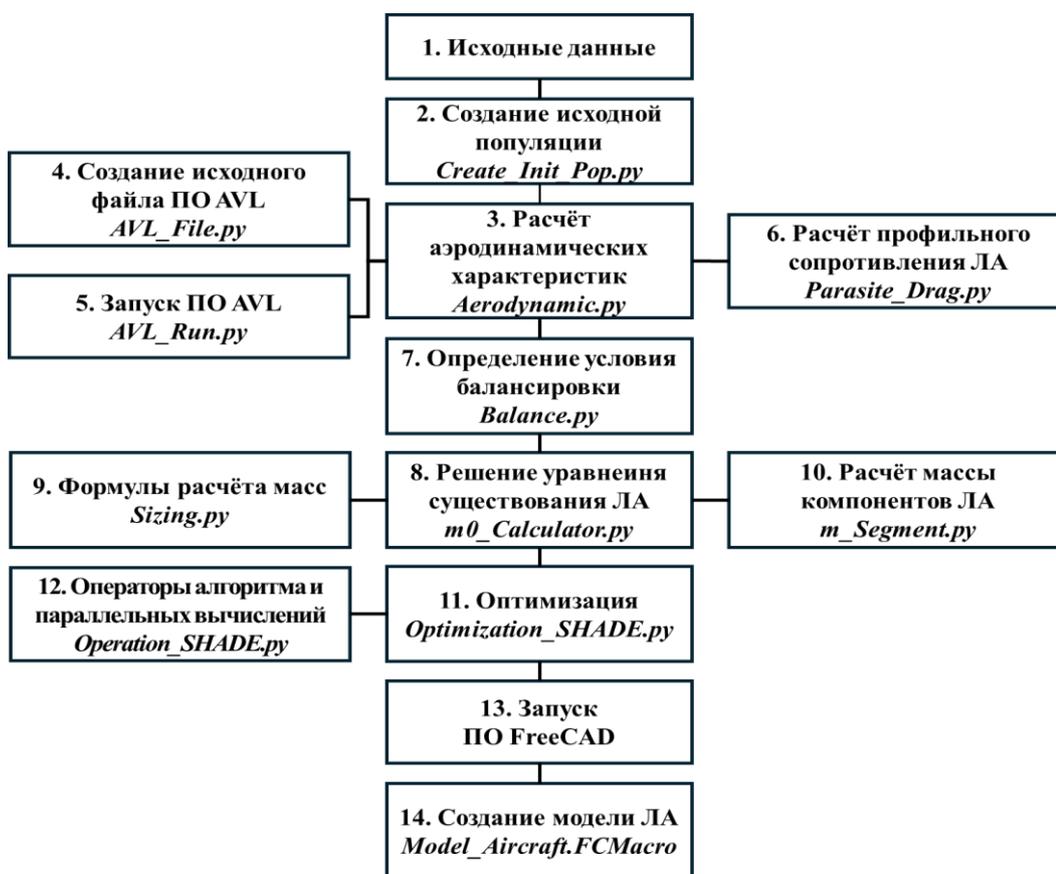


Рисунок 21 – Структура пакета программ АСП_КП

Исходные данные (написаны автором), необходимые для расчётов, включают константы расчёта характеристик полёта (время полёта, угол наклона траектории полёта), силовую установку (тип двигателя, удельный расход топлива, скорость вращения винта, удельный вес двигателя), массу (полезную нагрузку, массу оборудования), ограничения (ограничение на максимальное значение коэффициента подъёмной силы, ограничение на величину коэффициента статического момента горизонтального оперения) и диапазон расчётных переменных.

Модуль *Create_Init_Pop.py* (заимствован из открытых ресурсов [164]) инициализирует начальную популяцию с помощью метода латинского гиперкуба. Каждый вектор в популяции представляет собой определённый отдельный самолёт.

Модуль *Aerodynamic.py* (доработан автором) состоит из трех подмодулей. Подмодуль *AVL_File.py* (заимствован из открытых ресурсов [102] и доработан автором) принимает входные параметры из проектных переменных для создания файла в формате ****.avl* со структурой, необходимой ПО AVL. Этот файл моделирует все геометрические параметры компонентов самолёта как поверхности с вихревыми решётками. Подмодуль *AVL_Run.py* (заимствован из открытых ресурсов [102] и доработан автором) запускает ПО AVL, используя библиотеку *subprocess*. После запуска ПО AVL файл ****.avl* рассчитывается для определения аэродинамических коэффициентов. Чтобы не влиять на процесс параллельных вычислений, входные и выходные файлы ПО AVL автоматически удаляются после завершения расчёта. Поскольку ПО AVL не может рассчитать коэффициент профильного сопротивления, создаётся подмодуль *Parasite_Drag.py* (доработан автором) для определения коэффициента профильного сопротивления с использованием инженерных формул из [159].

Модуль *Balance.py* (написан автором) решает условие балансировки самолёта в вертикальной плоскости, используя графоаналитический метод.

Модуль *m0_Calculator.py* (написан автором) решает уравнение существования ЛА, включая 2 подмодуля. Первый подмодуль *Sizing.py* (доработан автором) реализует формулы для определения массы всех компонентов самолёта. Второй подмодуль *m_Segment.py* (доработан автором) применяет формулы из модуля *Sizing.py*. Выходными данными этого модуля являются масса компонентов самолёта, а также АДХ на каждом этапе полёта.

Модуль *Optimization_SHADE.py* (заимствован из открытых ресурсов [162] и доработан автором) является основным модулем программы. Он

выбирает проектные переменные самолёта на основе алгоритма оптимизации, упомянутого в разделе 2.6. Этот модуль включает в себя подмодуль *Operation_SHADE.py* (заимствован из открытых ресурсов [162, 163, 165] и доработан автором), содержащий операторы алгоритма, а также программу параллельных вычислений. В *Optimization_SHADE.py* интегрирована программа, которая запускает ПО FreeCAD (заимствован из открытых ресурсов [166] и доработан автором) сразу после завершения процесса оптимизации. Результаты процесса оптимизации сохраняются в папке программы, включая первый файл Excel, содержащий информацию о значениях проектных переменных, второй файл Excel, содержащий информацию о значении массы компонентов и АДХ для каждой особи каждого поколения. Кроме того, также сохраняется файл Excel, содержащий необходимую входную информацию последнего поколения для ПО FreeCAD.

Отдельный модуль *Model_Aircraft.FCMacro* (написан автором), написанный на Python, строит геометрическую модель с оптимальными геометрическими параметрами, полученными из программы оптимизации, сохраняет файл изображения в формате *.png* и завершает работу программы FreeCAD. Модуль *Model_Aircraft.FCMacro* находится в папке установки FreeCAD. При запуске ПО FreeCAD с помощью функции «запустить макрос при запуске» FreeCAD автоматически запускает файл *Model_Aircraft.FCMacro*, и весь процесс происходит скрыто внутри системы.

Файлы кода программы представлены в Приложении А.

2.10 Выводы по главе 2

1. Задача проектирования ЛА сформирована математически в терминах НМП с выбором взлётной массы ЛА в качестве целевой функции, определяемой уравнением существования с учётом выполнения условия продольной статической балансировки.

2. Разработаны алгоритм и ПО выбора облика и параметров БПЛА самолётного типа на этапе концептуального проектирования.

3. Разработанная АСП_КП состоит из 14 программных блоков (рисунок

21), из которых 6 блоков (2, 4, 5, 11, 12, 13) заимствованы из открытых ресурсов. Блоки 1, 3, 6, 7, 8, 9, 10, 14 реализуют алгоритм ДЭ с доработками, написанными автором, из которых блоки 1, 7, 8, 14 обладают существенными новизной.

4. Предложенный компактный набор из 12 проектных переменных обеспечивает возможность получения различных сбалансированных аэродинамических схем в оптимальном решении (нормальная схема, утка, тандем) за счёт определённых значений \bar{S}_2 и δ_2 .

5. С учётом специфики задачи предложено включение оценочной величины взлётной массы m_0^{es} в состав вектора проектных переменных. С помощью вычислительного эксперимента показано, что это приём даёт ускорение сходимости общего алгоритма оптимизации.

6. Визуализация особенностей оптимизации в виде плановой проекции и их трёхмерной модели для последующих этапов проектирования.

3 ДОСТОВЕРНОСТЬ И БЫСТРОДЕЙСТВИЕ РАЗРАБОТАННОЙ АСП_КП

В предыдущей главе описаны алгоритмы и реализация «АСП_КП» БПЛА самолётного типа, в которую включён ряд опубликованных программ и инновационные предложения автора с ответствующим ПО. В связи с этим в данной главе последовательно тестируются как работоспособность системы в целом, так и точность и эффективность инновационных модулей.

3.1 Валидация математических моделей аэродинамики

АДХ БПЛА рассчитываются на основе метода МДВ в соответствии с заданными параметрами, с подключением ПО AVL в комбинации с формулами инженерных методов на платформе Python. Валидация получаемых результатов расчёта осуществлена на основе сравнения с экспериментальными данными, полученными в аэродинамической трубе Самарского университета Т-3 весовым методом. Описание экспериментальной установки представлено в [170, 171]. Объектом исследования является модель с геометрическими характеристиками, представленными на рисунке 22.

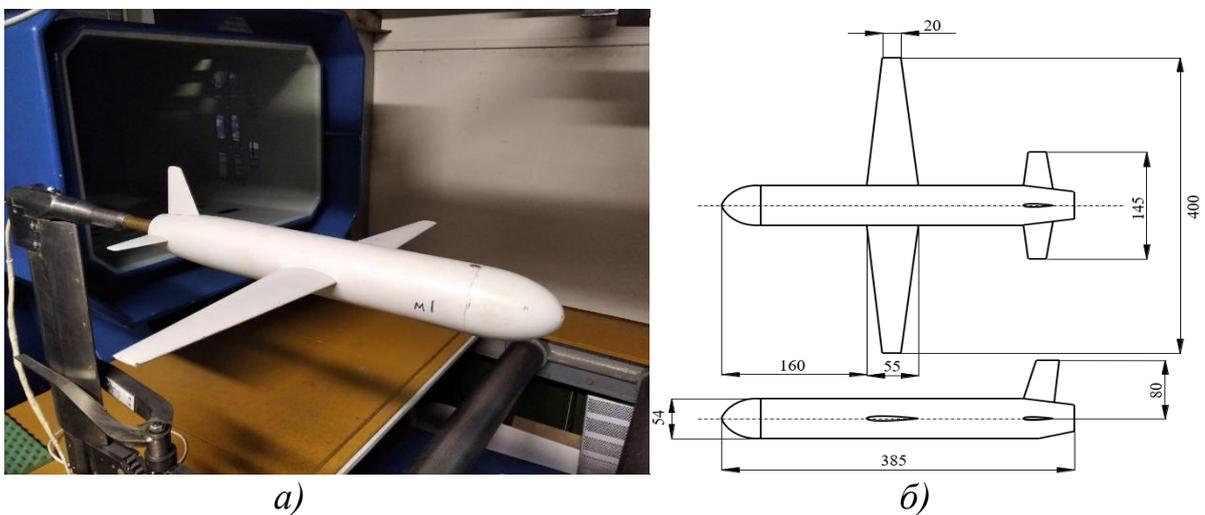


Рисунок 22 – Экспериментальная модель в аэродинамической трубе:

- а) Установка модели в аэродинамической трубе;
- б) Геометрические параметры экспериментальной модели

Эксперимент проводился в условиях окружающей среды $p = 736$ мм рт.ст., температуре $T = 23,4$ °С, скорости набегающего потока $V_\infty = 25,5$ м/с. Используемая модель имеет геометрические размеры, показанные на рисунке 22б, используемый профиль крыла - NASA 1115.

Аэродинамические коэффициенты, полученные с помощью ПО AVL, сравниваются с экспериментальными данными на рисунке 23.

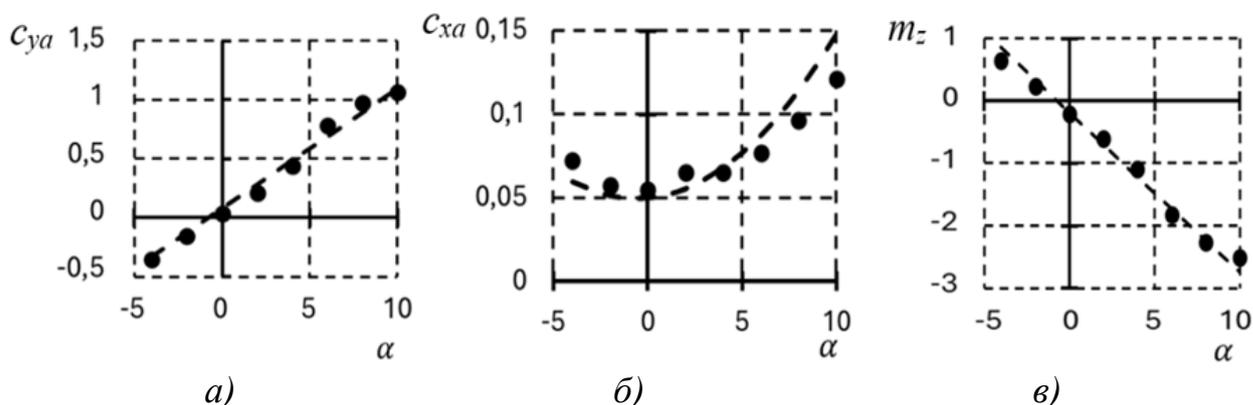


Рисунок 23 – Аэродинамические характеристики:
 а - зависимость c_{ya} от α ; б - зависимость c_{xa} от α ; в - зависимость m_z от α
 — - расчёты ПО AVL; ● экспериментальные данные

Нормализованная средняя абсолютная ошибка (по-английски - Normalized Mean Absolute Error) позволяет оценить точность данных без чрезмерного влияния величины задействованных значений и определяется по формуле [172]:

$$NMAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n \times [\max(y) - \min(y)]} \quad (60)$$

где y_i - значение АДХ, определённое экспериментально в точке i ; \hat{y}_i - значение АДХ, определённое ПО AVL в точке i ; n - количество точек сравнения.

Результаты валидации показали хорошее соответствие результатов расчёта с помощью используемых математических моделей и экспериментальных данных.

Нормализованная средняя абсолютная ошибка между данными по коэффициентам c_{ya} , c_{xa} и m_z составила 3,8%, 9,6%, 3,7% соответственно.

3.2 Оценка точности алгоритма расчёта взлётной массы

Для оценки достоверности расчёта целевой функции и ряда характеристик, получаемых по разработанной программе, выполнялся однократный расчёт важнейших характеристик двух существующих БПЛА на основе опубликованных данных БП-1 [150] и БП-2 [151].

Константы расчёта представлены в таблице 3, а основные параметры, использованные при расчётах, приведены в таблице 4.

Таблица 3 – Константы расчёта

Параметр		БП-1	БП-2
Время полёта, ч		35	24
Запас продольной статической устойчивости ($\Delta = \bar{x}_{ц,м} - \bar{x}_{\phi}$)		– 0,1	
Удельный расход топлива (C_e), кг/(кВт.ч)	набор высоты	0,285	
	крейсерский режим	0,27	
Угол траектории полёта, (θ) °	набор высоты	+ 5	
	крейсерский полёт	0	
	снижение	– 5	
КПД винтов [38]		0,76	
Полезная нагрузка ($m_{пн}$), кг		204	600
Относительная масса оборудования ($\bar{m}_{об}$)		0,08	
Удельный вес двигателя (γ), кг/кВт		0,87	

Таблица 4 – Параметры БПЛА БП-1 и БП-2

Параметр	БП-1	БП-2
Передняя несущая поверхность		
Удлинение	19	20
Стреловидность по передней кромке, °	7,22	1
Сужение	2,77	3
Угол заклинения, °	2,5	2,5
Задняя несущая поверхность		
Удлинение	6,75	20
Стреловидность по передней кромке, °	0	1
Сужение	1	3
Относительное расстояние между передними и задними крыльями ($\bar{L}_2 = L/b_{САХ}$)	4,368	5,55
Относительная площадь ($\bar{S}_2 = S_2/S_1$)	0,257	1
Лётные характеристики		
Скорость, м/с	47	55
Удельная нагрузка на крыло, кг/м ²	73,18	90
Вертикальное оперение		
Удлинение	4,4	1,3
Стреловидность по передней кромке, °	0	12,5
Сужение	1	2,3
Площадь, м ²	3,13	3
Фюзеляж		
Длина, м	8,23	11,13
Удлинение	10,7	10,6

Сходимость решения уравнения существования в предлагаемой методике концептуального проектирования обеспечивается в общем цикле оптимизации. Поэтому для решения уравнения существования в прямом

однократном расчёте целевой функции с исходными данными из таблицы 3 использован стандартный итерационный цикл. Блок-схема алгоритма определения взлётной массы представлена на рисунке 24. Рассмотрены этапы полёта: взлёт, набор высоты, крейсерский режим, снижение и посадка с учётом балансировки БПЛА в вертикальной плоскости на всех режимах полёта.

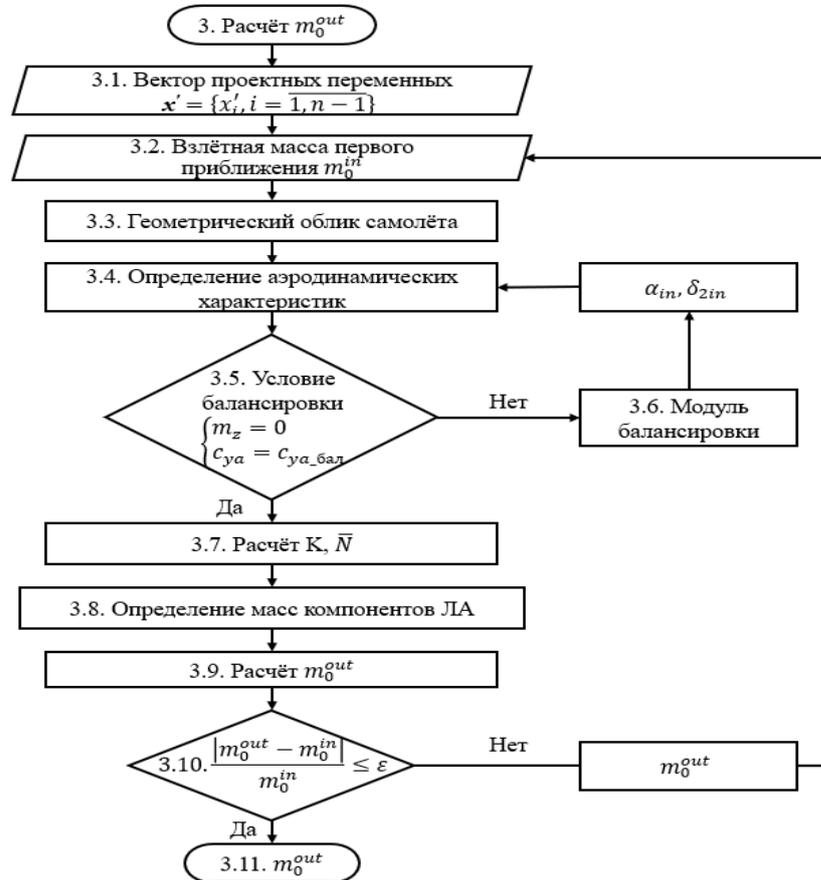


Рисунок 24 – Блок-схема алгоритма определения взлётной массы

В таблице 5 наряду с результатами однократного расчёта параметров БПЛА по разработанной программе приводятся также аналогичные характеристики указанных существующих аппаратов по опубликованным данным.

Таблица 5 – Результаты расчёта

Параметр	БП-1	Расчёт	%	БП-2	Расчёт	%
Взлётная масса, кг	1020	1015	0,49	2000	2055	2,68
Масса двигателей, кг	76	79	3,9	152	150	1,3
Масса топлива, кг	302	316	4,4	450	458	1,8
Масса пустого, кг	514	495	3,7	950	997	4,7
Макс. мощность двигателя, кВт	84,5	88,3	4,3	2×84,5	2×84,1	0,01

Отличие результатов расчёта взлётной массы, а также масс основных частей двух существующих БПЛА по предложенной методике не превысило 5%. Отличие расчётной потребной мощности силовой установки и фактической максимальной мощности силовой установки рассмотренных БПЛА находится в пределах 4-5%. Результаты сравнения данных в таблице 5 позволяют оценить точность расчёта основных характеристик БПЛА с помощью предложенной методики как достаточную для проектирования.

3.3 Оценка точности метода обеспечения балансировки самолёта

Для определения условий балансировки самолёта было проведено независимое исследование с использованием метода оптимизации «*Constrained optimization by linear approximation*» (COBYLA) в пакете OpenMDAO библиотеки Python. Результаты, полученные этим методом, сравниваются с результатами, полученными методом, предложенным в диссертации.

COBYLA - это алгоритм оптимизации без градиента, который использует линейную аппроксимацию функции в окрестности текущей точки для определения следующей точки для оценки функции. Алгоритм основан на идее «зоны доверия», что означает, что он учитывает только изменения переменных, которые находятся в пределах определённой «зоны доверия» вокруг текущей точки. COBYLA особенно полезна, когда целевую функцию трудно оценить или она не дифференцируема. Он также хорошо подходит для задач с ограничениями, поскольку может обрабатывать ограничения в качестве как равенства, так и неравенства. Применение COBYLA для решения задачи балансировки самолёта было реализовано и подтвердило свою корректность в [6, 142]. Модель задачи исходит из условия системы уравнений (30) и описывается в форме НМП следующим образом:

$$|m_z(\mathbf{x}'^{opt})| \leq |m_z(\mathbf{x}')| \quad \forall \mathbf{x}' \in \Omega_2, \\ \Omega_2 = \{\mathbf{x}': c_{ya}(\mathbf{x}') = c_{ya_{бал}}\}, \quad (61)$$

где $m_z(\mathbf{x}')$ - целевая функция (коэффициент продольного момента); $\mathbf{x}' = \{\alpha, \delta_2\}$ - вектор переменных; \mathbf{x}'^{opt} - оптимальное решение задачи; Ω_2 - область допустимых проектов.

Оба метода применены к одному и тому же набору данных, приведённому в таблицах 3 и 4, взятому с прототипа БПЛА БП-2 [151].

В таблице 6 представлены результаты, полученные двумя методами.

Таблица 6 – Результаты определения условий балансировки БПЛА БП-2

Параметр	Значение		Разница
	СОВУЛА	Предложенный метод	
Угол атаки (α), °	3,249	3,1	4,5%
Угол установки (δ_2), °	2,077	2,151	3,44%
Время (t), с	14,49	0,763	19 раз

Из результатов, представленных в таблице 6, следует, что разница между результатами двух методов составляет менее 5%. Преимущество предлагаемого метода заключается в значительном сокращении времени, необходимого для решения задачи.

3.4 Оценка точности расчёта массы топлива

Для оценки точности расчёта массы топлива в данной работе рассмотрены три метода определения массы топлива.

Первый метод предполагает, что мощность двигателя, а также масса самолёта не изменяются в течение всего полёта. Масса топлива определена по формуле (35).

Второй метод учитывает изменение массы самолёта за счёт выработки топлива во время полёта. Время полёта разбито на n равных полётных участков. Рассматривается балансировка ЛА в каждом участке и выполняется расчёт потребной энерговооружённости и взлётной массы ЛА с учётом выработки топлива. За счёт изменения полётной массы самолёт на каждом полётном участке будет иметь своё состояние балансировки и лётно-технические характеристики.

Коэффициент подъёмной силы, обеспечивающий условия балансировки на каждом участке, определяется по формуле:

$$c_{yabal_i} = g \cdot \frac{m_{0_{i-1}}}{qS} \cos(\theta), \quad (62)$$

где g - ускорение свободного падения, м/с²; q - скоростной напор, Па; S - общая площадь основных поверхностей, м²; $m_{0_{i-1}}$ - масса самолета в начале i -ого

этапа полёта, кг; θ - угол наклона траектории полёта, °; n - количество этапов полёта, $i = \overline{1, n}$.

Во время решения задачи балансировки на каждом полётном участке определяется свой угол атаки α_i и аэродинамическое качество K_i .

Потребная энерговооружённость рассчитывается по формуле:

$$\bar{N}_i = \frac{V}{\eta_{\text{вин}}} \cdot \frac{K_i \sin \theta + \cos \theta}{\sin \alpha_i + K_i \cos \alpha_i}. \quad (63)$$

Количество израсходованного топлива на каждом этапе полёта рассчитывается по формуле:

$$\bar{m}_{\text{топ}i} = \bar{N}_i \cdot C_e \cdot \Delta t, \quad (64)$$

$$m_{\text{топ}i} = \bar{m}_{\text{топ}i} \cdot m_{0i-1},$$

где $m_{\text{топ}i}$ - абсолютная масса топлива, израсходованного на i -ом этапе полёта, кг; \bar{N}_i - потребная энерговооружённость на i -ом полётном участке, кВт/даН; C_e - удельный расход топлива, кг/(кВт.ч); Δt - время полёта каждого этапа полёта, ч.

Масса самолёта по истечению полётного участка будет уменьшаться на величину $m_{\text{топ}i}$ и определяется по формуле:

$$m_{0i} = m_{0i-1} - m_{\text{топ}i}. \quad (65)$$

Масса израсходованного топлива в крейсерском режиме рассчитывается по формуле:

$$m_{\text{топ}} = \sum_1^n m_{\text{топ}i} \quad (66)$$

Третий метод основан на формуле Бреге [47-50] применительно к винтовым самолётам, масса топлива определяется по формуле:

$$\bar{m}_{\text{топ}} = 1 - e^{-\frac{V \cdot t \cdot C_e}{75 \cdot K \cdot \eta_{\text{вин}}}}, \quad (67)$$

где V - скорость полёта, м/с; t - время полёта, ч; C_e - удельный расход топлива, кг/(л.с.ч); K - аэродинамическое качество; $\eta_{\text{вин}}$ - КПД винтов.

На основе констант расчёта (таблица 3) и геометрических параметров (таблица 4) прототипа БПЛА БП-2 рассчитана масса топлива по трём методам для основного этапа полёта: крейсерского этапа полёта. В частности, для второго метода количество участков на этапе крейсерского полёта составляет

$n = 10$. Результаты представлены в таблице 7.

Таблица 7 – Результаты расчёта массы топлива БПЛА БП-2

Параметр	Метод 1	Метод 2	Метод 3
Абсолютная масса топлива крейсерского полёта, кг	385,117	355,105	338,337
Время расчёта, с	2	8	2

На рисунках 25-29 показаны изменения параметров во времени, определяемые вторым методом.

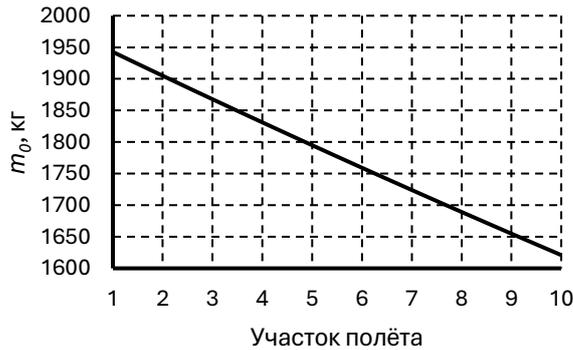


Рисунок 25 – Изменение массы самолёта в крейсерском полёте

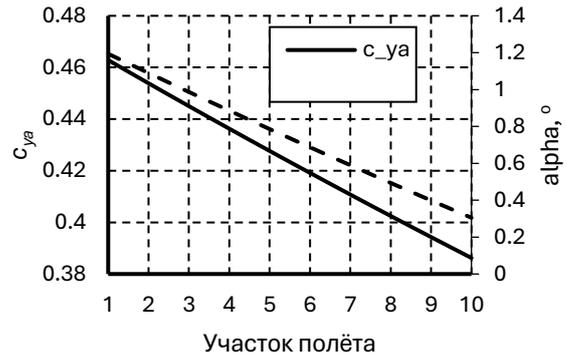


Рисунок 26 – Изменение подъёмной силы и угла атаки, обеспечивающее условия балансировки в крейсерском полёте

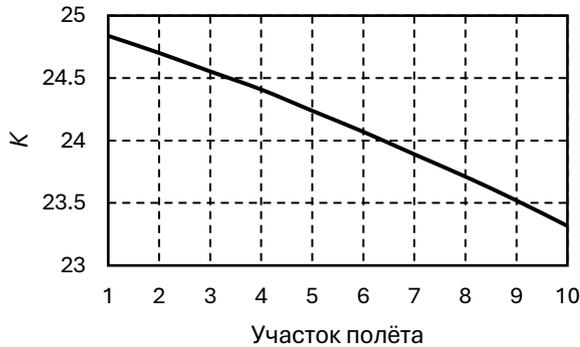


Рисунок 27 – Изменение качества в крейсерском полёте

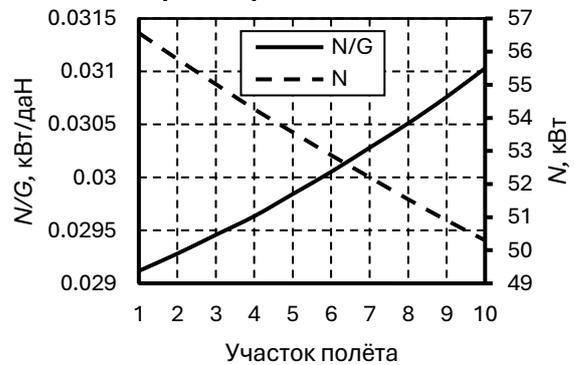


Рисунок 28 – Изменение требуемой мощности в крейсерском полёте

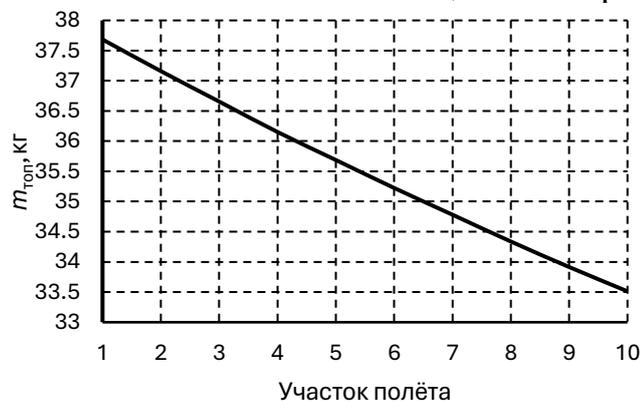


Рисунок 29 – Масса топлива, израсходованная на каждом участке полёта

Из графиков видно, что с течением времени масса самолёта уменьшается за счёт расхода топлива (рисунок 25), что приводит к уменьшению потребного коэффициента подъёмной силы и угла атаки самолёта в сбалансированном режиме (рисунок 26). При этом аэродинамическое качество также уменьшается (рисунок 27), что вызывает увеличение потребной энерговооружённости (рисунок 28). Однако в целом требуемая мощность двигателя снижается вследствие уменьшения массы самолёта (рисунок 28). Количество топлива, потребляемого за один и тот же период времени, постепенно уменьшается (рисунок 29).

В первом методе масса самолёта и мощность двигателя считаются постоянными в течение всего полёта, поэтому получаемое значение массы топлива определённо завышено. Хотя формула Бреге учитывает изменение массы самолёта но не учитывает изменения аэродинамического качества при перебалансировке. Формула Бреге использует максимальное значение K в начале этапа полёта и сохраняет его постоянным, что приводит к занижению расчётных значений требуемой массы топлива.

По результатам, представленным в таблице 7, масса топлива, полученная по методу 2 на 7,793% ниже, чем метод 1 и на 4,722 % выше, чем метод 3. Однако время расчёта метода 2 в 4 раза превышает время расчёта методов 1 и 3. Как компромисс в практических расчётах целесообразно использовать предложенный второй метод.

3.5 Выводы по главе 3

1. Проведена валидация математических моделей аэродинамики, показавшая хорошее соответствие получаемых результатов с расчётными и экспериментальными данными, в пределах 10%.

2. Однократный расчёт взлётной массы и масс компонентов двух существующих БПЛА различается не более, чем 5%. Полученные результаты подтвердили точность предложенного метода.

3. Оценена точность метода определения балансировки самолёта. Результаты, полученные с помощью метода оптимизации COBYLA

использующего пакет OpenMDAO, показывают корректность и многократно (в 19 раз) более высокую вычислительную производительность предложенного метода.

4. Выявлена недостаточная точность двух известных методов определения необходимой массы топлива для длительного крейсерского режима полёта. Предложен ступенчатый метод, дающий достаточную точность с отклонением результатов от формулы Бреге меньше 5%.

4 ДОСТОВЕРНОСТЬ И ЭФФЕКТИВНОСТЬ МЕТОДА ОПТИМИЗАЦИИ

В данной главе рассматривается применение программы «АСП_КП» к оптимизации существующих БПЛА. В качестве объектов исследования ставятся и решаются задачи по снижению взлётной массы ранее выбранных прототипов БП-1 с взлётной массой 1020 кг и БП-2 с взлётной массой 2000 кг.

4.1 Постановка и решение задач

Программа оптимизации «АСП_КП» применена при заданных диапазонах значений проектных переменных в таблицы 1, константах расчёта в таблицы 3 и настройках в таблицы 8.

Таблица 8 – Настройки алгоритма оптимизации (общие)

Параметр	Значение
Ограничение коэффициента подъёмной силы (c_{va})	$< 0,6$
Ограничения коэффициента статического момента горизонтального оперения ($A_{го}$)	$[0,2...0,6]$
Начальная величина штрафа (U^*), кг	60000
Критерий остановки оптимизации $\max(m_0^+(P_{g+1})) - \min(m_0^+(P_{g+1})) \leq \varepsilon$	0,005
Максимальное количество поколений (G)	12000
Количества проектных переменных (n)	12
Размер исходной популяции (w_0)	$10 \times n$
Минимальный размер популяции (количество особей)	12

Полученные в качестве оптимальных проектные переменные представлены в таблице 9 вместе с соответствующими данными прототипов.

Таблица 9 – Результаты оптимизации БП-1 и БП-2

Проектная переменная	БП-1		БП-2	
	Значение прототипа	Оптимальное значение	Значение прототипа	Оптимальное значение
m_0^{es} , кг	1020	914	2000	1667
λ_1	19	7,6	20	14
$\chi_1, ^\circ$	7,22	0	1	2,7
η_1	2,77	3	3	1,9
$\delta_1, ^\circ$	2,5	0	2,5	2,5
λ_2	6,75	18,7	20	4
$\chi_2, ^\circ$	0	3	1	2
η_2	1	1,62	3	1,6
\bar{L}_2	4,37	4,6	5,55	5,1
\bar{S}_2	0,26	4,9	1	0,2
V , м/с	47	45	55	50
ρ_0 , кг/м ²	73,2	75	90	92

В результате работы оптимизационного алгоритма для обоих прототипов найдены решения с меньшей взлётной массой: для БП-1 - на 10,4% и для БП-2 - на 16,7%.

4.2 Анализ результатов оптимизации

4.2.1 Анализ сходимости

Сходимость оптимизации для прототипа БП-1 достигнута после 117 генераций с затратой 57 минут, а для прототипа БП-2 после 120 генераций и с затратой 62 минут на компьютере с Windows 10, процессором Intel(R) Core i7-6700 @ 3,40 ГГц, оперативной памятью 64 ГБ. Численность последней генерации составила 23 и 22 особи, соответственно для БП-1 и БП-2. На рисунке 30 показана сходимость взлётной массы в процессе оптимизации по используемому алгоритму.

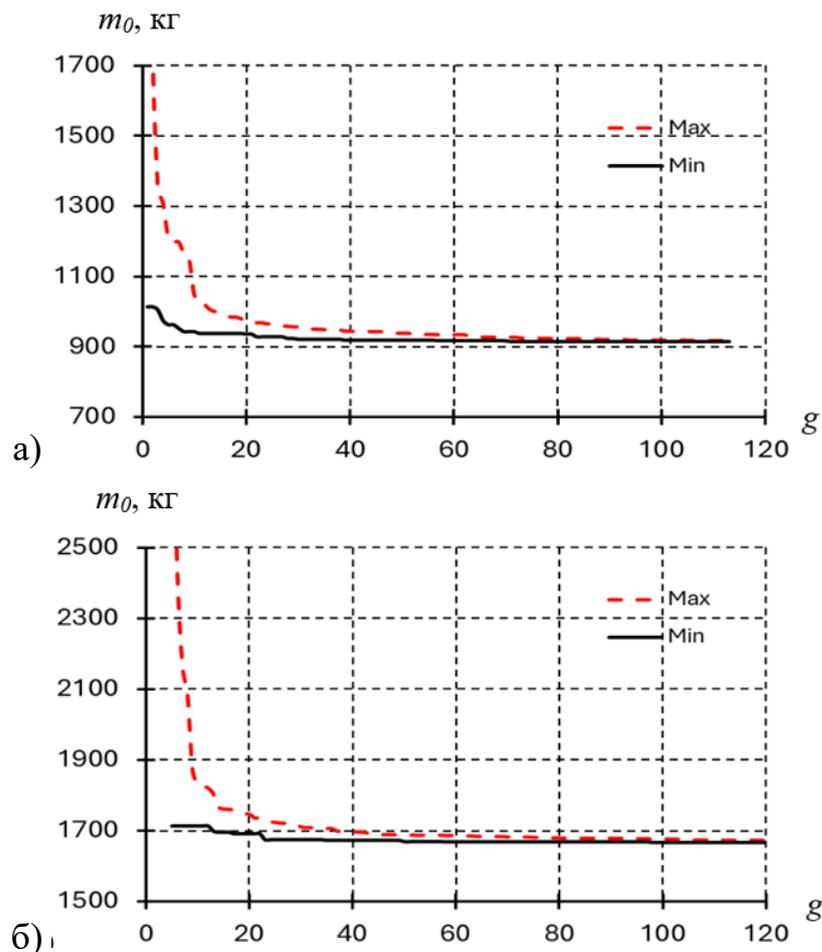


Рисунок 30 – Сходимость вычисления взлётной массы БП-1 (а), БП-2 (б) по поколениям предлагаемого алгоритма

Таким образом, в рассмотренных задачах значение целевой функции m_0 , близкое к оптимальному, получено уже на 20-ой генерации.

4.2.2 Геометрический анализ

В связи с тем, что в процессе оптимизации (таблица 9) некоторые проектные переменные изменяются от нулевых или очень малых значений до существенно больших, применить обычный подход для оценки величины изменений какой-либо переменной в процентах по отношению к начальной величине, не представляется возможным. Поэтому в данной работе для этих целей используется методика анализа такого типа данных, предложенная в [173]. В соответствии с этой методикой для того, чтобы наглядно представить степень изменения переменных, необходимо сначала нормализовать их на один и тот же интервал $[0, 1]$ по следующей формуле:

$$x'_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \quad (68)$$

где x - начальное значение переменной; x' - нормализованное значение переменной; $\min(x)$ - нижний предел переменной; $\max(x)$ - верхний предел переменной; i - индекс переменной, в данной работе $i = \overline{1,12}$.

Нормированные значения проектных переменных для прототипов и оптимальных решений представлены в таблице 10.

Таблица 10 – Нормированные значения проектных переменных

Проектная переменная	БП-1			БП-2		
	Значение прототипа	Оптимальное значение	Изменение	Значение прототипа	Оптимальное значение	Изменение
m_0^{es}	0,21	0,17	0,04	0,6	0,47	0,13
λ_1	0,94	0,23	0,71	1	0,63	0,37
χ_1	0,48	0	0,48	0,07	0,18	0,11
η_1	0,89	1	0,11	1	0,45	0,55
δ_1	0,5	0	0,5	0,5	0,5	0
λ_2	0,17	0,92	0,75	1	0	1
χ_2	0	0,2	0,2	0,07	0,13	0,06
η_2	0	0,31	0,31	1	0,3	0,7
\bar{L}_2	0,59	0,65	0,06	0,89	0,78	0,11
\bar{S}_2	0,01	0,98	0,97	0,17	0	0,17
V	0,28	0,25	0,03	0,42	0,33	0,09
p_0	0,47	0,5	0,03	0,82	0,84	0,02

Значение безразмерных нормированных проектных переменных (68) прототипов до и после оптимизации внутри заданного диапазона (таблица 1) показаны на рисунке 31.

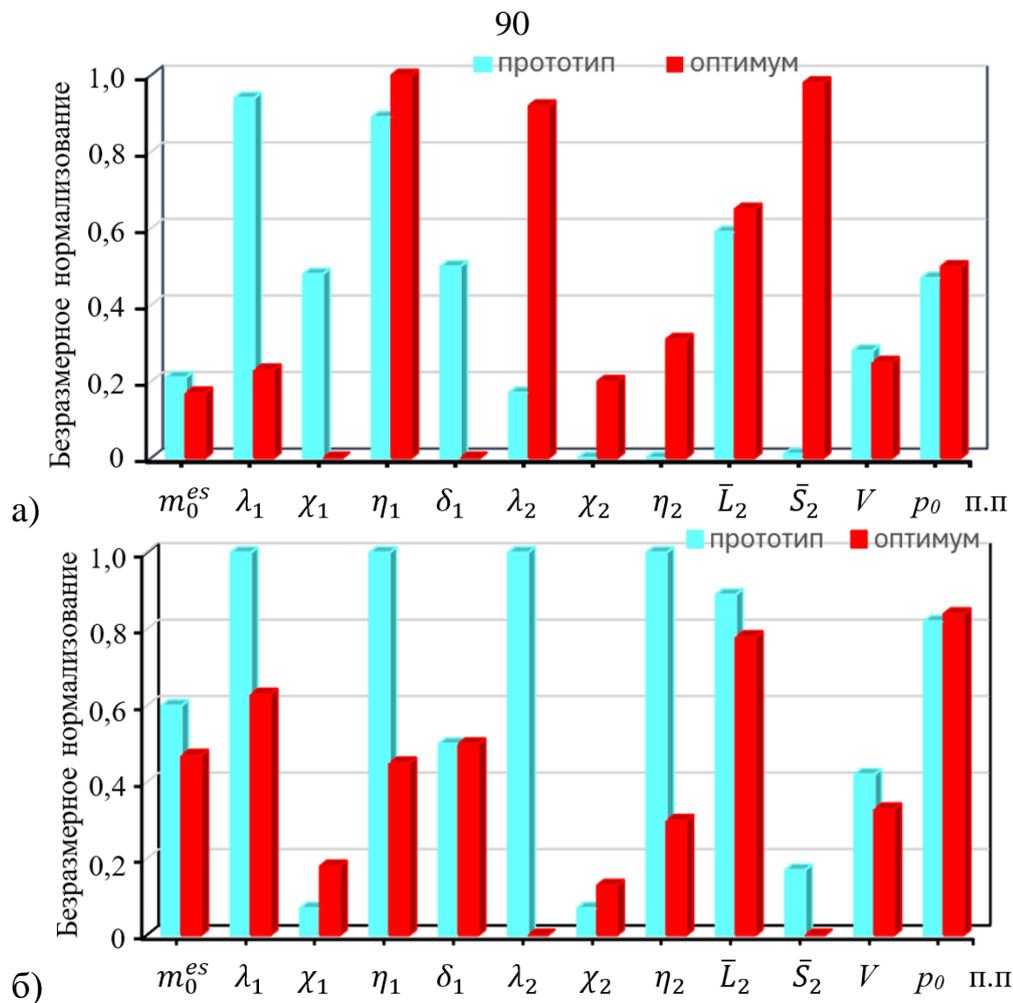


Рисунок 31 – Значения нормализованных проектных переменных БП-1 (а), БП-2 (б) в результате оптимизации

Полученные в результате оптимизации облики БПЛА (БП-1, БП-2) представлены на рисунке 32 в сравнении с прототипами.

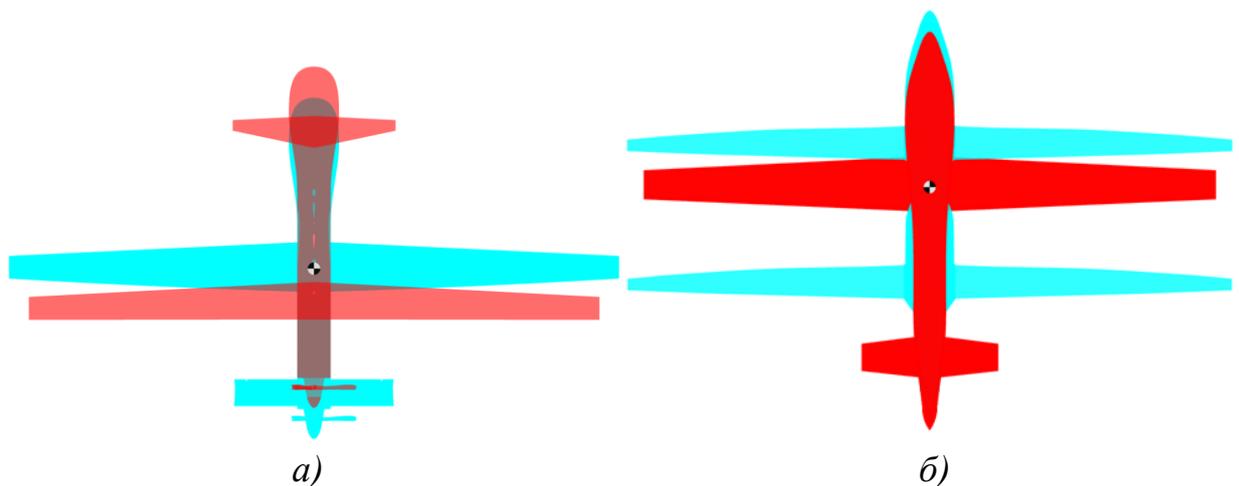


Рисунок 32 – Облики БПЛА по прототипам (синий) и после оптимизации (красный): а) БП-1; б) БП-2

Рисунки получены с помощью ПО FreeCAD, которое используется в Batch-режиме для автоматической отрисовки общего вида оптимальных особей.

Из анализа оптимальных параметров (таблица 9) следует, что им соответствуют существенные изменения облика рассмотренных аппаратов, показанные на рисунке 32, которые можно рассматривать как идеи для разработки рекомендаций по модификациям этих ЛА.

4.2.3 Весовой анализ

В таблице 11 представлены дополнительные результаты оптимизации, выдаваемый программой «АСП_КП».

Таблица 11 – Дополнительные результаты оптимизации

Параметр		БП-1			БП-2		
		Исходное значение	Оптимизация	Δ , %	Исходное значение	Оптимизация	Δ , %
Взлётная масса, кг		1020	914	10,4	2000	1667	16,7
Масса оборудования, кг		н/д	84,1	-	н/д	137	-
Массы, кг	передней несущей поверхности	н/д	22,4	-	н/д	181,7	-
	задней несущей поверхности	н/д	123,4	-	н/д	25,2	-
	вертикального оперения	н/д	19,6	-	н/д	13,5	-
	фюзеляжа	н/д	77,6	-	н/д	171,6	-
	шасси	н/д	36,5	-	н/д	84,1	-
Масса двигателей, кг		76	76	0	152	132	13,2
Масса топлива, кг		302	266,6	11,7	450	313,5	30,3
Масса винтов, кг		н/д	3,8	-	н/д	8,4	-
Мощность двигателя, кВт		1x84,5	1x80,6	4,6	2x84,5	2x62,9	25,6
Относительное положение центра масс по оси x ($\bar{x}_{\text{цм}} = \frac{x_{\text{цм}}}{b_{\text{САХ}}}$)		н/д	-0,651	-	н/д	0,54	-
Крейсерский коэффициент подъёмной силы		н/д	0,59	-	н/д	0,59	-
Крейсерское аэродинамическое качество		н/д	24,7	-	н/д	26	-

Для покомпонентного анализа составляющие взлётной массы представлены на гистограммах на рисунке 33.

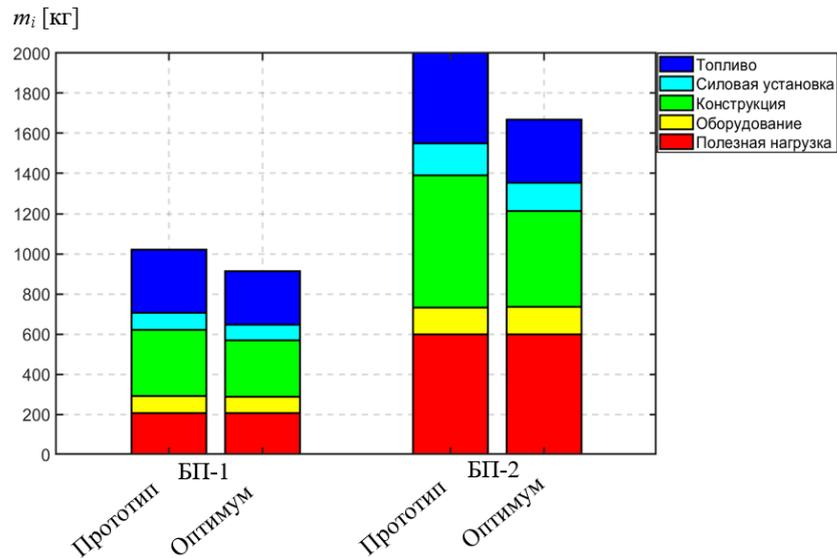


Рисунок 33 – Весовые сводки рассмотренных БПЛА до и после оптимизации

На рисунке 34 показаны потребные мощности двигателя до и после оптимизации.

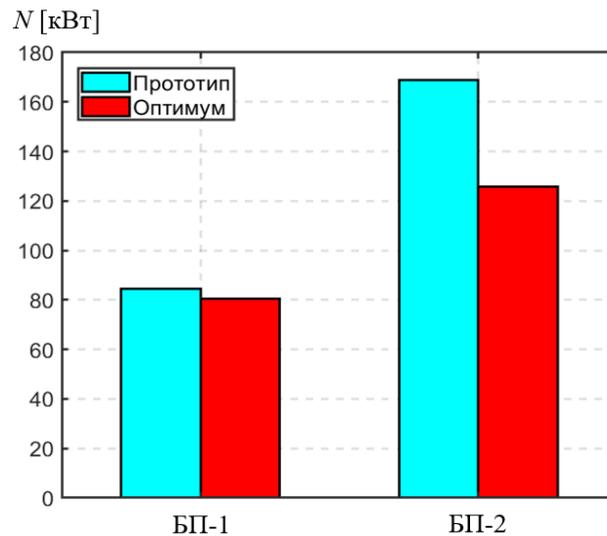


Рисунок 34 – Сравнение мощности двигателя прототипа до и после оптимизации

Из таблицы 11 следует, что оптимизация позволяет снизить взлётную массу рассмотренных БПЛА и уменьшить потребную мощность силовых установок.

Прототип БПЛА БП-1 имеет нормальную схему, размещение тяжёлой силовой установки в хвостовой части фюзеляжа сместило центр масс назад, что потребовало смещение крыла назад ближе к оперению. Для сохранения требуемого коэффициента статического момента горизонтального оперения из-за уменьшения плеча горизонтального оперения вынужденно была

увеличена его площадь и ухудшены динамические характеристики БПЛА в продольном канале. Предлагаемый алгоритм позволил получить другую компоновку - с передним горизонтальным оперением меньшей площади и большего плеча. Переднее горизонтальное оперение создаёт положительную подъёмную силу и таким образом несколько разгружает крыло. Таким образом уменьшена площадь крыла и увеличено аэродинамическое качество БПЛА. В совокупности это дало снижение взлётной массы на 10,4% и потребной мощности силовой установки на 4,6%.

БПЛА БП-2 имеет аэродинамическую схему «тандем» с равным распределением площадей между крыльями при прочих равных параметрах будет иметь самое низкое аэродинамическое качество и самую большую массу конструкции. Данное обстоятельство связано с тем, что заднее крыло в крейсерском режиме полёта (основной режим для дальних БПЛА) является «недогруженным»: в рассмотренном примере при $\bar{S}_2 = 1$ переднее крыло создаёт 64% подъёмной силы, а заднее - 36%. Попытки загрузить заднее крыло приводят к нормальной схеме или к схеме «утка». Увеличение массы конструкции БП-2 связано и с большим удлинением обоих крыльев. Предлагаемый алгоритм перераспределил воздушную нагрузку более рационально и свёл решение к нормальной схеме с параметром соотношения площадей крыльев $\bar{S}_2 = 0,2$. Данное решение позволило несколько уменьшить удлинение основного крыла и тем самым снизить массу конструкции крыла и БПЛА в целом. За счёт повышения аэродинамического качества на 3,84% и снижения взлётной массы БПЛА на 16,7% потребная мощность силовой установки была снижена на 25,6%. Это позволило бы для БПЛА БП-2 заменить имеющиеся двигатели силовой установки (Rotax 914) на менее мощные, например, Rotax 912, что дополнительно снизило бы массу силовой установки и БПЛА в целом.

4.3 Анализ значимости проектных переменных для целевой функции

Влияние проектных переменных на целевую функцию различно. Это имеет важное значение в процессе проектирования, поскольку помогает

разработчикам уточнить диапазон проектных переменных, даже уменьшить количество проектных переменных для повышения стабильности и эффективности алгоритма оптимизации, особенно на этапе концептуального проектирования.

Влияние проектных переменных на целевую функцию - взлётную массу - в данной работе предлагается определять путём анализа изменения взлётной массы (Δm_{0i}), соответствующего поочерёднему изменению каждой проектной переменной (Δx_i) на 10% при фиксированных остальных переменных. Этот расчёт может проводиться как для исходного прототипа самолёта, так и для модификации после оптимизации.

Применяя формулу расчёта частной производной функции многих переменных в общем случае [174]:

$$\left| \frac{\partial f}{\partial x_i} \right| \approx \lim_{\Delta x_i \rightarrow 0} \left| \frac{f(x_1, \dots, x_i + \Delta x_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{\Delta x_i} \right| \quad (69)$$

к изучаемым примерам в настоящей работе результаты изменения взлётной массы в окрестности оптимального решения, соответствующего каждой переменной, представлены в таблице 12.

Таблица 12 – Изменение взлётной массы, соответствующее каждому проектному переменному

Проектная переменная	Δm_0			
	БП-1		БП-2	
	Прототип	Оптимум	Прототип	Оптимум
$\Delta \lambda_1$	2,115%	0,346%	1,291%	0,51%
$\Delta \chi_1$	0,162%	0,004%	0,557%	0,15%
$\Delta \eta_1$	0,273%	0,128%	0,433%	0,04%
$\Delta \delta_1$	0,099%	0,016%	0,490%	0,04%
$\Delta \lambda_2$	2,010%	0,831%	0,230%	0,17%
$\Delta \chi_2$	0,109%	0,003%	0,554%	0,01%
$\Delta \eta_2$	0,209%	0,033%	0,519%	0,02%
$\Delta \bar{L}_2$	0,405%	0,311%	0,250%	0,14%
$\Delta \bar{S}_2$	0,798%	0,924%	0,307%	0,12%
ΔV	11,370%	4,458%	10,288%	2,43%
Δp_0	2,335%	2,041%	2,827%	1,22%

Значимость проектных переменных для модификации БП-1 и БП-2 показана на рисунке 35.

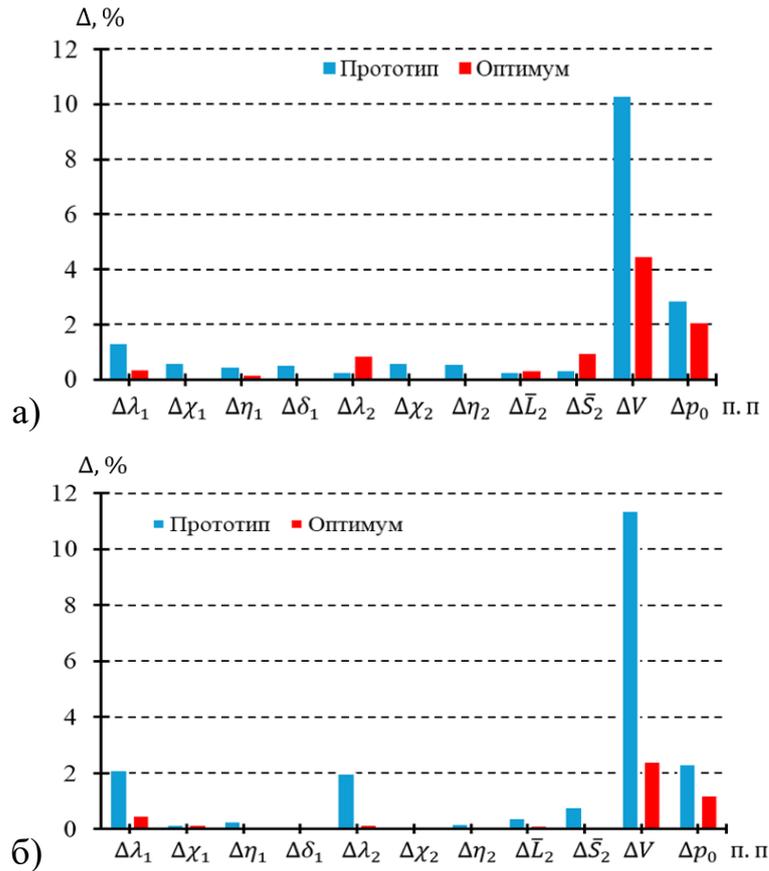


Рисунок 35 – Значимость проектных переменных для взлётной массы БПЛА БП-1 (а) и БП-2 (б)

Из рисунка 35 видно, что градиент целевой функции исходного прототипа несколько больше, чем у оптимизированной модификации. Это ещё раз подтверждает, что найденное решение близко к оптимальному.

Анализ рисунка 35 показывает, что переменные, оказывающие существенное влияние на взлётную массу, включают удлинение (λ_1, λ_2), относительную площадь (\bar{S}_2), скорость полёта (V) и удельную нагрузку на крыло (p_0). Удлинение крыла напрямую влияет на аэродинамическое качество и массу конструкции крыла, а относительная площадь определяет тип возможной аэродинамической конфигурации самолёта, что в свою очередь влияет на площадь крыла и балансировочной поверхности. Скорость полёта и удельная нагрузка на крыло ограничивают максимальный коэффициент подъёмной силы, а также непосредственно участвуют в формулах для расчёта массы конструкции.

Хотя угол заклинения передней поверхности (δ_1) и относительное расстояние (\bar{L}_2) оказывают незначительное влияние на взлётную массу, они

напрямую влияют на условия обеспечения продольной балансировки самолёта, а также ограничивают коэффициент статического момента горизонтального оперения.

Остальные проектные переменные оказывают незначительное влияние, отчасти из-за их роли в формулах расчёта массы, а также потому, что в дозвуковом режиме полёта $M < 0,6$ их влияние на АДХ также незначительно.

На основании таких оценок можно рассмотреть возможность сокращения количества проектных переменных для повышения производительности и работоспособности алгоритма оптимизации, а также расширения рассмотрения других проектных переменных.

4.4 Достоверность метода оптимизации

Оценка точности оптимизационных алгоритмов в сложных многодисциплинарных технических задачах заслуживает специального рассмотрения. В известной литературе по НМП сходимость оптимизационных алгоритмов, как правило, демонстрируется с помощью тестовых функций, тестовых моделей и специально разработанных задач с известным оптимальным решением [175].

Из анализа, приведённого в разделе 4.3, видно, что в рассматриваемых задачах проектными переменными, которые существенно влияют на целевую функцию, являются скорость полёта и удельная нагрузка на крыло.

Для оценки работоспособности предложенного алгоритма используется метод построения поверхности отклика. Выбор подходящего сочетания скорости и удельной нагрузки на крыло осуществляется при фиксированных значениях остальных проектных переменных. Значения этих переменных варьировались в диапазонах $30 \leq V \leq 90$ и $5 \leq p_0 \leq 110$. Остальные проектные переменные приняты фиксированными: удлинение передней $\lambda_1 = 14$ и задней несущей поверхности $\lambda_2 = 4$; - стреловидность по передней кромке передней $\chi_1 = 2,7^\circ$ и задней несущей поверхности $\chi_2 = 2^\circ$; сужение передней $\eta_1 = 1,9$ и задней несущей поверхности $\eta_2 = 1,6$; угол заклинения передней несущей поверхности $\delta_1 = 2,5^\circ$; относительное расстояние между несущими

поверхностями $\bar{L}_2 = 5,1$; относительная площадь задней несущей поверхности $\bar{S}_2 = 0,2$.

Этот метод определяет взлётную массу самолёта на основе решения уравнения существования самолёта при изменении параметров скорости и удельной нагрузки на крыло. На этой основе определена зависимость взлётной массы от этих двух параметров, а также наименьшее значение m_0 .

Для заданных исходных данных было установлено, что наивыгоднейшее сочетание двух параметров составило $V = 50$ м/с и $p_0 = 90$ кг/м². В этом случае взлётная масса составляет $m_0 = 1668$ кг, аэродинамическое качество $K = 25,6$, а время расчёта - 86 м.

Поверхность отклика целевой функции представлена на рисунке 36а, а на рисунке 36б показана аппроксимация полученной зависимости аэродинамического качества от скорости и удельной нагрузки на крыло.

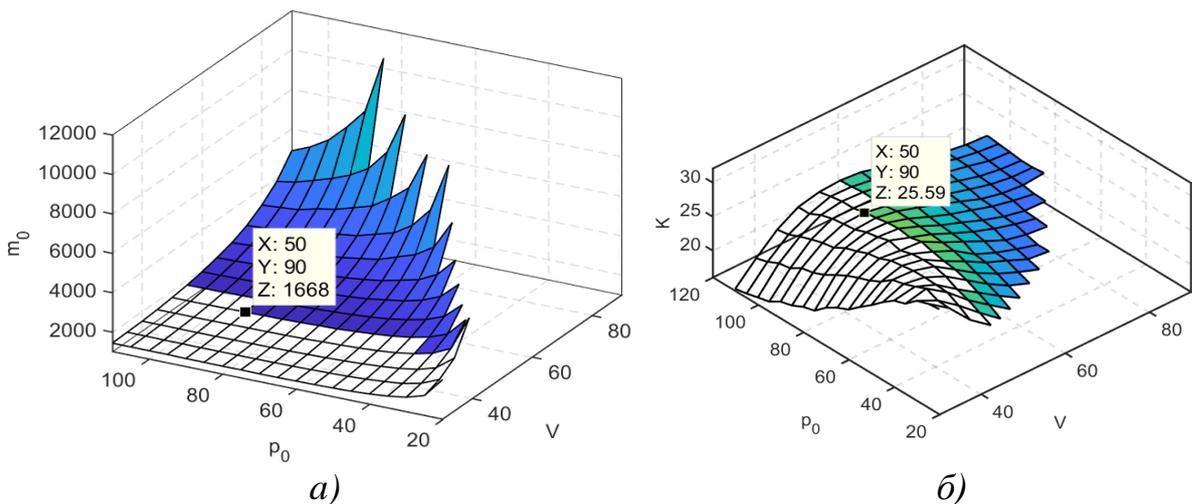


Рисунок 36 – Поверхности отклика взлётной массы (а) и аэродинамического качества (б) по скорости и удельной нагрузке на крыло

Разница значения взлётной массы между предложенным методом и методом построения поверхности отклика составляет всего 0,06%, подтверждая тем самым работоспособность предложенного алгоритма. Однако, при этом открытым остаётся теоретический вопрос об оптимальной точке в многомерном пространстве и практический вопрос о рельефе целевой функции в окрестности оптимума, в которой могут находиться интересные рациональные решения. Рассмотрим возможный путь решения этой задачи.

Совокупность средств для анализа рельефа целевой функции в многомерном пространстве проектных переменных с целью оценки близости получаемого результата оптимизации к предполагаемому экстремуму можно представить в виде *специальной разработанной методики тестирования области оптимальных проектов*, которая именуется далее методика «*test-opt*»:

1. значения полученных проектных переменных в качестве оптимальных нормализуются по (68) (таблица 10) и визуализируются в виде номограммы (рисунок 31). Эти результаты показывают расположение каждой проектной переменной в своём допустимом диапазоне;

2. значимость каждой проектной переменной анализируется по (69) (таблица 12) и представляется в виде номограммы (рисунок 35);

3. проводится исследование оптимума по значимым проектным переменным при фиксированных величинах малозначимых (рисунок 36);

4. при наличии прототипа действия 1-3 повторяются для сравнения и вынесения суждения об оптимальности полученного решения.

Замечание - В тех случаях, когда отсутствует прототип с найденными решением, его можно создать специально следующим образом:

а) формируется случайный вектор $\xi = \{\xi_1, \xi_2, \dots, \xi_n\}$ по числу проектных переменных n в рассматриваемой задаче из составляющих ξ_i ($i = \overline{1, n}$), равномерно распределённых на отрезке $[-1, 1]$;

б) случайный вектор нормируется:

$$\zeta = \frac{\xi}{\sqrt{\sum_{i=1}^n \xi_i^2}},$$

где ζ - вектор единичной длины, направленный вдоль ξ ;

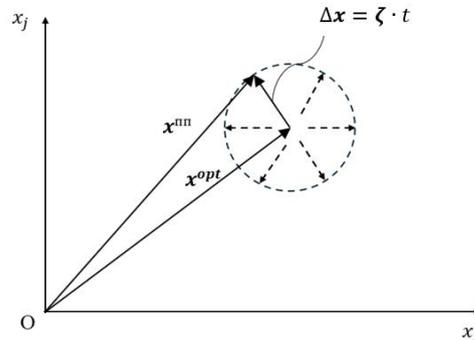
в) создаётся вектор смещения-приращений проектных переменных Δx :

$$\Delta x = \zeta t,$$

где ζ - направляющий нормированный вектор, определяющий направление движения в пространстве проектных переменных; t - скалярная величина, которая определяет длину шага смещения Δx ;

г) геометрическим суммированием исследуемого вектора x^{opt} и вектора

смещения Δx создаётся вектор проектных переменных псевдо-прототипа $x^{пп} = x^{opt} + \Delta x$, конец которого находится на гиперсфере радиуса t с центром в конце вектора x^{opt} , то есть в некоторой управляемой окрестности исследуемого оптимального решения x^{opt} (рисунок 37).

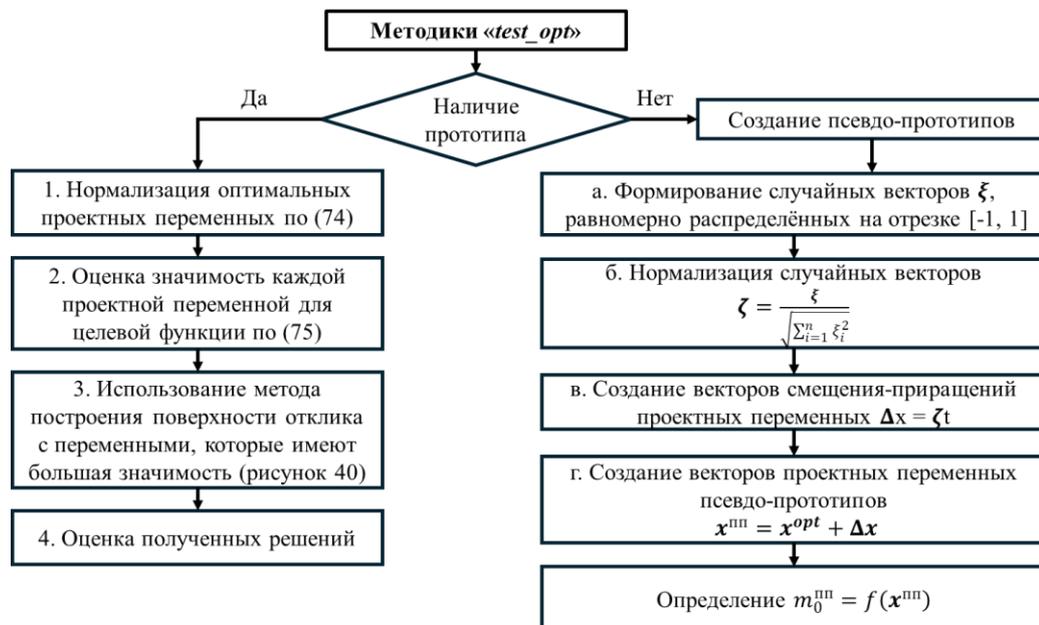


Рисунке 37 – Создание псевдо-прототипов

Малыми стрелками на рисунке 37 показаны другие возможные Δx , которые соответствуют другим наборам случайных чисел ξ_i в векторе ξ .

Полученный вектор $x^{пп}$ вводится в программу оптимизации для однократного расчёта целевой функции $f(x^{пп})$. Получаемое значение $m_0^{пп}$ псевдо-прототипа сравнивается с m_0^{opt} .

Последовательность шагов методик «test-opt» представлена на рисунке 38.



Рисунке 38 – Последовательность шагов методик «test-opt»

Пример генерации и использования псевдо-прототипа для оценки оптимального решения для БП-2 представлен в таблице 13. В таблице 13 индексом k и k' обозначены различные случайные вектора

$\xi^{(k)}$ и $\xi^{(k')}$ (случайные числа из интервала $[-1, 1]$) и относящиеся к ним единичные векторы $\zeta^{(k)}$, $\zeta^{(k')}$, шаги смещения $\Delta x^{(k)}$ и $\Delta x^{(k')}$ и наборы переменных псевдо-прототипов $x^{пп(k)}$ и $x^{пп(k')}$. Этим данным соответствует: длина вектора x^{opt} - 105,9 с результатом оптимизации $m_0^{opt} = 1667$ кг. Исходя из длины вектора x^{opt} , для исследования области оптимальных проектов выбирались шаги смещения $t = 1$ и 10 (рисунок 37), которые составляют примерно 1% и 10% от длины вектора x^{opt} .

Таблица 13 – Пример генерации и использования псевдо-прототипа при $t = 1$ и 10

Проектная переменная	Этап								
	а		б		в		г		
	$\xi^{(k)}$	$\xi^{(k')}$	$\zeta^{(k)}$	$\zeta^{(k')}$	$\Delta x^{(k)}$ ($t = 1$)	$\Delta x^{(k')}$ ($t = 10$)	$x^{пп(k)}$	$x^{пп(k')}$	x^{opt}
m_0 , кг	-	-	-	-	-	-	1710	2097	1667
λ_1	-0,5	-0,10	-0,29	-0,06	-0,29	-0,56	13,71	13,44	14
χ_1	-0,2	0,01	-0,12	0,01	-0,12	0,08	2,58	2,78	2,7
η_1	0,2	0,39	0,12	0,21	0,12	2,15	2,02	4,05	1,9
δ_1	0,7	-0,21	0,41	-0,12	0,41	-1,19	2,91	1,31	2,5
λ_2	0,3	-0,30	0,18	-0,17	0,18	-1,67	4,18	2,33	4
χ_2	-0,4	0,84	-0,24	0,47	-0,24	4,69	1,77	6,69	2
η_2	0,5	0,61	0,29	0,34	0,29	3,42	1,89	5,02	1,6
\bar{L}_2	-0,8	-0,59	-0,47	-0,33	-0,47	-3,26	4,63	1,84	5,1
\bar{S}_2	0,3	0,24	0,18	0,13	0,18	1,31	0,38	1,51	0,2
V	0,6	-0,73	0,35	-0,41	0,35	-4,05	50,35	45,95	50
p_0	-0,7	0,96	-0,41	0,53	-0,41	5,33	91,59	97,33	92

Из таблицы 13 следует, что проектным переменным псевдо-прототипа $x^{пп(k)}$ соответствует взлётная масса 1710 кг и $x^{пп(k')}$ - 2097 кг. Таким образом, смещение на 1% ($t = 1$) от оптимальной точки увеличивает взлётную массу на 3,14%, смещение на 10% ($t = 10$) от оптимальной точки увеличивает взлётную массу на 20,49%. Подобные результаты получаются при исследовании оптимума с наборами других случайных векторов на отрезке $[-1, 1]$, что свидетельствует:

- о способности рассмотренного алгоритма оптимизации находить экстремум с высокой точностью;
- в рассмотренных примерах область оптимальных решений обладает значительной чувствительностью к проектным переменным.

4.5 Оценка значимости и эффективности включения условия балансировки в расчёт и оптимизацию взлётной массы

Рассмотрим вариант алгоритма, описанного в главе 2, без включения внутреннего цикла решения с условием обеспечения балансировки ЛА. Отличие этого варианта состоит в том, что угол атаки ЛА и угол установки задней поверхности не определяются через условие обеспечения балансировки, а учитываются через проектные переменные (α , δ_2) и выбираются в процессе оптимизации. Тогда вектор проектных переменных становится $\mathbf{x}' = \{m_0^{es}, \lambda_1, \chi_1, \eta_1, \delta_1, \lambda_2, \chi_2, \eta_2, \bar{L}_2, \bar{S}_2, V, p_0, \alpha, \delta_2\}$.

Результаты, получаемые по модифицированному алгоритму (без учёта условия обеспечения балансировки), сравнивались с результатами, получаемыми по основному алгоритму главы 2 (с учётом балансировки, рисунок 8) и представлены таблице 14.

Таблица 14 – Сравнение результатов модифицированного алгоритма и основного алгоритма

С учётом балансировки		Без учёта балансировки			
Проектная переменная	Значение	Проектная переменная	Значение		
1	m_0^{es} , кг	1672	1	m_0^{es} , кг	1616
2	λ_1	14,0	2	λ_1	16,8
3	χ_1 , °	2,7	3	χ_1 , °	0,2
4	η_1	1,9	4	η_1	3
5	δ_1 , °	2,5	5	δ_1 , °	3
6	λ_2	4,0	6	λ_2	4,0
7	χ_2 , °	2,0	7	χ_2 , °	0
8	η_2	1,6	8	η_2	1,0
9	\bar{L}_2	5,1	9	\bar{L}_2	2
10	\bar{S}_2	0,2	10	\bar{S}_2	0,2
11	V , м/с	50	11	V , м/с	50
12	p_0 , кг/м ²	92	12	p_0 , кг/м ²	92,7
-			13	α , °	1,77
			14	δ_2 , °	0,0
Дополнительный результат	Значение	Дополнительный результат	Значение		
c_{xa}	0,023	c_{xa}	0,022		
c_{ya}	0,59	c_{ya}	0,59		
m_z	0	m_z	-0,1		
Поколение (g)	123	Поколение (g)	359		
Время, м	62	Время, м	114		
α , °	2,8	-			
δ_2 , °	-2,2				

Наиболее заметным отличием результатов, представленных в таблице 14, является значение угла установки горизонтального оперения δ_2 . В варианте, не учитывающем условие обеспечения балансировки, угол установки горизонтального оперения равен $0,0^\circ$, что создаёт на горизонтальном оперении положительную подъёмную силу. Это означает, что подъёмная сила, создаваемая крылом, уменьшается, так как общая подъёмная сила остаётся неизменной. Это приводит к уменьшению площади крыла, а также требуемого угла атаки полёта. Уменьшение площади крыла напрямую влияет на массу конструкции крыла, а также позволяет программе оптимизации выбирать крылья с увеличенным удлинением, снижая индуктивное сопротивление и улучшая аэродинамическое качество. Все эти факторы позволяют снизить взлётную массу самолёта на 3,34%.

Однако исключение внутреннего цикла для решения условия балансировки приводит к увеличению числа проектных переменных на два (α и δ_2), что, в свою очередь, увеличивает размерность пространства поиска. В непрерывных областях объём пространства поиска увеличивается экспоненциально с ростом размерности. Напротив, количество функций оценки, необходимых для исследования этого пространства поиска, обычно увеличивается только линейно. Это расхождение оказывает значительное влияние на параметры, используемые в алгоритме ДЭ, по мере увеличения размерности.

Изучение влияния размерности пространства поиска на быстродействие и точность алгоритма ДЭ показано следующие: пространства высокой размерности повышают сложность поиска, что часто приводит к замедлению сходимости и снижению эффективности алгоритма, так как требуется больше вычислительных ресурсов для исследования увеличенного пространства решений [176].

Предложенный автором алгоритм, хотя и включает внутренний цикл для решения условия балансировки, благодаря преимуществам аналитического метода, изложенного в разделе 3.3, значительно сокращает время,

затрачиваемое на его выполнение. Как видно из результатов, представленных в таблице 14, по сравнению с рассмотренным вариантом предлагаемый метод достигает сходимости за в 3 раза меньшее количество поколений, а время оптимизации сокращается вдвое.

4.6 Оценка влияния на быстродействие алгоритма включения оценочной величины взлётной массы в состав вектора проектных переменных

Одной из особенностей предлагаемого алгоритма является рассмотрение оценочной величины взлётной массы как проектной переменной. В процессе оптимизации процессам мутации и кроссовер подлежат не только рассматриваемые проектные переменные, но и оценочная величина взлётной массы. По итогам получения новой популяции будут выбраны особи с наименьшей величиной взлётной массы. В ходе оптимизации значение оценочной величины взлётной массы предыдущего шага оптимизации, вводимое в вектор проектных переменных после выбора, и текущее значение выходной оценочной величины взлётной массы будут приближаться, обеспечивая сходимость. Таким образом, данный подход обеспечивает сходимость уравнения существования в общем цикле оптимизации и позволяет значительно сократить время расчёта, не требуя внутреннего итерационного решения существования на каждом шаге оптимизации.

Для оценки производительности предлагаемого метода рассмотрен вариант, в котором оценочная величина взлётной массы не рассматривается как проектная переменная, уточнённая взлётная масса определяется через внутренний цикл решения уравнения существования для каждой особи (рисунок 24).

Оба метода рассчитаны с использованием данных прототипа БП-2 и на одном компьютере с Windows 10, процессором Intel(R) Core i7-6700 @ 3,40 ГГц, оперативной памятью 64 ГБ.

Результаты оптимизации двух методов сравнены в таблице 15.

Таблица 15 – Анализ эффективности включения расчётного значения оценочной величины взлётной массы в состав проектных переменных

С учётом оценочной величины взлётной массы в векторе проектных переменных			Без учёта оценочной величины взлётной массы в векторе проектных переменных		
Проектная переменная		Значение	Проектная переменная		Значение
1	m_0^{es} , кг	1672	1	-	
2	λ_1	14,0	2	λ_1	14,5
3	$\chi_1, ^\circ$	2,7	3	$\chi_1, ^\circ$	0
4	η_1	1,9	4	η_1	1,2
5	$\delta_1, ^\circ$	2,5	5	$\delta_1, ^\circ$	2,5
6	λ_2	4,0	6	λ_2	5,0
7	$\chi_2, ^\circ$	2,0	7	$\chi_2, ^\circ$	0
8	η_2	1,6	8	η_2	1,05
9	\bar{L}_2	5,1	9	\bar{L}_2	5,0
10	\bar{S}_2	0,2	10	\bar{S}_2	0,21
11	V , м/с	50	11	V , м/с	50
12	p_0 , кг/м ²	92	12	p_0 , кг/м ²	93
Дополнительный результат			С учётом оценочной величины взлётной массы в векторе проектных переменных		Без учёта оценочной величины взлётной массы в векторе проектных переменных
m_0^{out} , кг			1667		1682
Количество итераций расчёта уравнения существования на каждом шаге оптимизации для каждого индивида (среднее)			1		6
Время расчёта уравнения существования для одной особи [с] (среднее)			2,4		14
Поколение (g)			120		80
Время, м			62		209

Результаты в таблице 15 показывают, что разница значений взлётной массы двух методов составляет всего 0,89%. При этом количество итераций (поколений, генераций) до сходимости для варианта без включения оценочной величины взлётной массы в вектор проектных переменных меньше по сравнению с предложенным методом. Однако общее время оптимизации с использованием оценочной величины взлётной массы значительно меньше (в

3 раза). Это объясняется тем, что затраты на вычисление взлётной массы в каждой генерации существенно (6 раз) меньше.

На рисунке 39 показан процесс сходимости двух методов. Следует отметить, что оба метода сходимости приводят к одному и тому же значению целевой функции и одинаковому сочетанию проектных переменных.

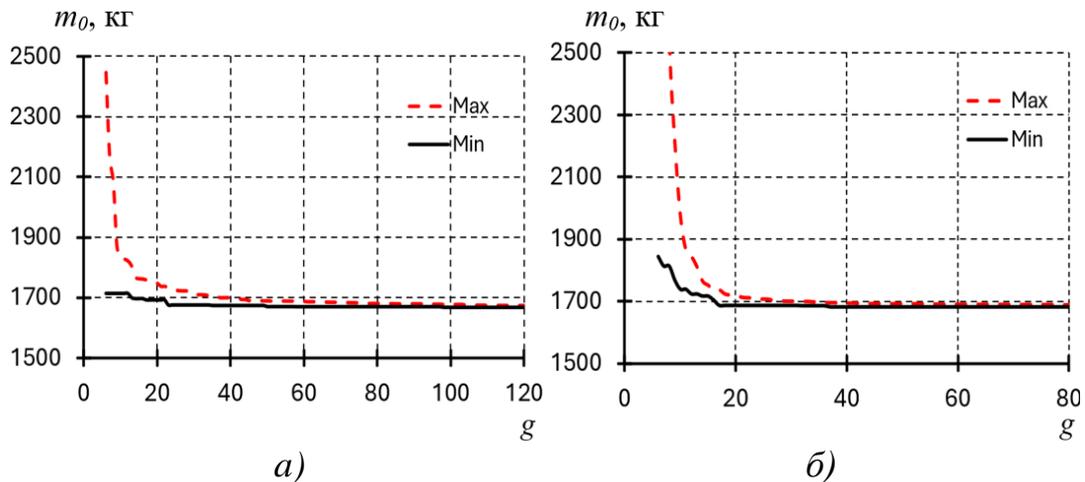


Рисунок 39 – Процесс сходимости целевой функции с учётом (а) и без учёта (б) оценочной величины взлётной массы в векторе проектных переменных

Из рисунка 39 видно, что разница минимального значения взлётной массы между началом и концом оптимизации предлагаемого метода меньше, чем у метода с внутренним циклом. Это помогает проектантам быстро провести предварительную оценку проектных решений.

4.7 Выводы по главе 4

1. Вычислительный эксперимент с оптимизацией проектных переменных БП-1 и БП-2 показал способность разработанного алгоритма и программы находить оптимальные значения 12 разнородных проектных переменных в задачах со сложными математическими моделями в приемлемое машинное время для цели концептуального проектирования - в пределах 1 часа.

2. Анализ значимости проектных переменных через величину градиента целевой функции в зависимости от расположений в пространстве проектных переменных прототипа и оптимального решения БП-1 и БП-2 (рисунок 35) показывает, что эти точки находятся в окрестности оптимума. Однако, при этом параметры оптимального решения соответствуют более эффективным проектам. В обеих задачах достигнуто существенное улучшение (уменьшение)

целевой функции m_0 соответственно на 10% и 16%.

3. Построение и анализ функций отклика m_0 и K по наиболее значимым проектным переменным - скорости (V) и удельной нагрузке на крыло (p_0) показывает, что результаты оптимизации находятся на предельных значениях, что свидетельствуют о невозможности дальнейшего улучшения проекта.

4. Из анализа результатов оптимизации следует, что предложенный набор проектных переменных позволяет находить решения с существенными изменениями облика прототипов, показанными на рисунке 32, которые можно рассматривать как идеи для разработки рекомендаций по модификациям этих ЛА.

5. Анализ включения условия балансировки в общий цикл оптимизации снижает размер пространства поиска и приводит к двукратному сокращению времени оптимизации.

6. Включение оценочной величины взлётной массы в состав вектора проектных переменных исключает внутренний цикл при решении уравнения существования ЛА и даёт дополнительное трёхкратное сокращение общего времени оптимизации.

7. Разработанный комплекс методик «*test-opt*» позволяет оценить разработанную методику выбора оптимальных параметров облика БПЛА самолётного типа как достаточно быстрый инструмент для поддержки концептуального проектирования с высокой точностью.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

1. Разработан алгоритм и ПО автоматизированной системы поддержки концептуального проектирования ЛА самолётного типа «АСП_КП» на основе варианта оптимизационного алгоритма ДЭ, отличающийся от известных решений интеграцией опубликованных ПО (AVL, FreeCAD) с вновь разработанными программами параллельных вычислений, использованием дискретного пространства проектных переменных, уточнённым расчётом массы топлива, включением оценочной величины взлётной массы в состав вектора проектных переменных.

2. Предложен компактный набор из 12 проектных переменных, способный порождать в процессе оптимизации различные аэродинамические схемы от «нормальной» до схемы «утка», включая «тандем».

3. Разработана методика и ПО уточнённого расчёта взлётной массы с внутренним циклом учёта обеспечения условия продольной устойчивости.

4. Предложено дополнительное включение оценочной величины целевой функции в состав вектора проектных переменных для сокращения затрат времени на оптимизационный процесс.

5. Выбранные и разработанные функциональные математические модели для выполнения аэродинамических, весовых и энергетических расчётов обладают компромиссным сочетанием характеристик по точности и быстродействию, а также допускают простое включение в АСП_КП через предложенные проектные переменные.

6. Разработанное ПО позволяет получать в процессе оптимизации большой объём дополнительной информации в цифровом и графическом виде, полезной для интерпретации получаемых результатов и использования на следующих этапах проектирования: лётно-технические, энергетические, аэродинамические характеристики, относительные и абсолютные массы частей самолёта, его облик, требования к двигателю и т. п.

7. Тестирование отдельных модулей АСП_КП проведено различными способами: сопоставлением с физическим экспериментом, расчётом массы

конструкции и топлива двух существующих и описанных в литературе БПЛА, существенно различных по массе полезной нагрузки и времени барражирования, показало, что расхождение величин предсказываемых критических параметров с действительными составляет менее 5%.

8. Тестирование ПО в целом на предмет способности находить лучшие результаты по сравнению с известными дало снижение взлётной массы для БП-1 и БП-2 соответственно 10% и 16%.

9. Применение комплекса методик «test-opt» к анализу проектных переменных существующих БП-1 и БП-2 и полученных в результате оптимизации показало, что рассмотренные реальные и виртуальные проекты находятся в области оптимальных решений.

10. Специальные вычислительные эксперименты показали, что учёт продольной устойчивости в расчёте взлётной массы сокращает время оптимизационных расчётов вдвое. Дополнительно к этому включение значения целевой функции в вектор проектных переменных даёт сокращение времени расчётов в 3 раза. Суммарно обе инновации сокращают время поиска оптимального проекта в 6 раз, доводя его до величины менее 1 часа в рассматриваемых задачах.

Перспективы дальнейшей разработки темы

Полученные результаты могут быть использованы для дальнейшей разработки методов оценки массы конструкций ЛА нетрадиционных компоновок на основе безразмерного критерия силового совершенства и учёта аэроупругих явлений.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

- COBYLA - Constrained optimization by linear approximation (Ограниченная оптимизация с помощью линейной аппроксимации);
- LHS - Latin hypercube sampling (Отбор образцов латинского гиперкуба);
- SHADE - Success-History based adaptive differential evolution (Адаптивная дифференциальная эволюция на основе истории успеха);
- АДХ - аэродинамическая характеристика;
- АСП_КП - Автоматизированная система поддержки концептуального проектирования;
- БАС - беспилотная авиационная система;
- БВС - беспилотное воздушное судно;
- БПЛА - беспилотный летательный аппарат;
- ДВС - двигатель внутреннего сгорания;
- ДЭ - дифференциальная эволюция;
- КПД - коэффициент полезного действия;
- ЛА - летательный аппарат;
- МДВ - метод дискретных вихрей;
- МДО - многодисциплинарная оптимизация;
- МКЭ - метод конечных элементов;
- НДС - напряженно-деформированное состояние;
- НМП - нелинейное математическое программирование;
- ПО - программное обеспечение;
- САХ - средняя аэродинамическая хорда;
- А - архив худших особей;
- $A_{го}$ - коэффициент статического момента горизонтального оперения;
- b_0 - корневая хорда поверхности;
- b_k - концевая хорда поверхности;
- b_{ϕ} - максимальная ширина фюзеляжа;
- C_e - удельный расход топлива;
- C_K - коэффициент силового фактора;

CR_S - значение скорости кроссовера;

c_{ya} - коэффициент подъёмной силы;

c_{xa} - коэффициент сопротивления;

c_{xi} - коэффициент индуктивного сопротивления;

c_{x0} - коэффициент сопротивления при нулевой подъёмной силе;

$c_{ya_{\text{бал}}}$ - коэффициент подъёмной силы в состоянии балансировки;

c_{ya}^* - допустимый коэффициент подъёмной силы;

\bar{c}_0 - корневая относительная толщина поверхности;

\bar{c}_k - концевая относительная толщина поверхности;

$d_{\text{ф}}^3$ - эквивалентный диаметр миделева сечения фюзеляжа;

E - удельная энергоёмкость аккумулятора;

F_s - коэффициент масштабирования;

G - вес ЛА;

g - ускорение свободного падения/ номер популяции;

H - размер исторической памяти;

$h_{\text{ф}}$ - максимальная высота фюзеляжа;

i - индекс проектных переменных;

K - аэродинамическое качество;

k - коэффициент, учитывающий увеличение массы силовой установки за счёт систем;

$k_{\text{мех}}$ - коэффициент, учитывающий механизацию крыла;

$k_{\text{кон}}$ - коэффициент, учитывающий тип конструкции крыла;

$k_{\text{мт}}$ - коэффициент, учитывающий марку основного материала конструкции крыла;

$k_{gш}$ - коэффициент, учитывающий положение крыла относительно фюзеляжа;

\bar{L} - относительное расстояние между несущими поверхностями;

l - размах поверхности;

$l_{\text{ф}}$ - длина фюзеляжа;

Mxf - множество векторов особей, значение штрафной функции у которых

равно нулю;

M_z - продольный аэродинамический момент относительно центра масс;

m_0 - взлётная масса;

m_0^{in} - входная взлётная масса;

m_0^{out} - выходная взлётная масса;

m_0^{opt} - взлётная масса решения оптимизации;

m_0^+ - целевая функция;

$m_0^{пп}$ - взлётная масса псевдо-прототипа;

m_z - коэффициент продольного момента относительно центра масс;

m_{zA} - коэффициент продольного момента относительно носка САХ;

$m_{п.н}$ - масса полезной нагрузки;

\bar{m}_{cy} - относительная масса силовой установки;

$\bar{m}_{топ}$ - относительная масса топлива;

$\bar{m}_{акк}$ - относительная масса аккумуляторных батарей;

$\bar{m}_{конс}$ - относительная масса конструкции;

$\bar{m}_{об}$ - относительная масса оборудования и управления;

$\bar{m}_{кр}$ - относительная масса конструкции крыла;

$\bar{m}_{ф}$ - относительная масса конструкции фюзеляжа;

$\bar{m}_{в.о}$ - относительная масса конструкции вертикального оперения;

$\bar{m}_{ш}$ - относительная масса шасси;

\bar{N} - энерговооружённость;

n - перегрузка/количество проектных переменных/количество этапов полёта;

P - популяция;

P - сила тяги силовой установки;

p - параметр жадности/количество ограничения в виде неравенств;

p_0 - удельная нагрузка на несущую систему;

q - динамическое давление/количество ограничения в виде равенств;

R - параметр усиления штрафа;

S - площадь поверхности крыла;

- $S_{\text{полн}}$ - полная площадь внешней поверхности фюзеляжа;
 S_{Σ} - суммарная площадь несущих поверхностей;
 S_1 - площадь передней несущей поверхности крыла;
 S_2 - площадь задней несущей поверхности;
 \bar{S}_2 - относительная площадь задней несущей поверхности;
 s - индекс особи;
 t - время полёта;
 U^* - верхнее ограничение возможного значения взлётной массы;
 \mathbf{u} - вектор проектных переменных популяции кроссовера;
 V - скорость полёта;
 $V_{\text{макс}}$ - предельная расчётная скорость;
 $V_{\text{крейс}}$ - крейсерская скорость;
 \mathbf{v} - вектор проектных переменных популяции мутации;
 w - количество особей в популяции;
 X_a - сила сопротивления;
 \mathbf{x} - вектор проектных переменных;
 \mathbf{x}^{opt} - оптимальное решение задачи;
 $\mathbf{x}_{\text{pbest}}$ - случайно выбранный вектор из группы лучших векторов;
 \mathbf{x}_r - любой случайный вектор;
 $\mathbf{x}^{\text{пп}}$ - вектор проектных переменных псевдо-прототипа;
 x_i - проектный переменный;
 x_i^l - нижняя граница проектного переменного;
 x_i^u - верхняя граница проектного переменного;
 Y_a - подъёмная сила;
 α - угол атаки;
 γ - удельная масса двигателя;
 Δ - требуемый запас продольной статической устойчивости;
 Δt - время полёта каждого этапа полёта;
 $\Delta \mathbf{x}$ - вектора смещения;

δ - угол установки поверхности;

ε - величина критерия остановки;

ζ - направляющий нормированный вектор, определяющий направление движения в пространстве проектных переменных;

η - сужение поверхности;

$\eta_{\text{вин}}$ - КПД винтов;

$\eta_{\text{су}}$ - КПД силовой установки;

θ - угол наклона траектории полёта;

λ - удлинение поверхности;

$\lambda_{\text{ф}}$ - удлинение фюзеляжа;

$\lambda_{\text{н.ч}}$ - удлинение носовой части фюзеляжа;

$\lambda_{\text{х.ч}}$ - удлинение хвостовой части фюзеляжа;

ξ - случайный вектор;

$\bar{\sigma}$ - коэффициент удельной прочности материала;

φ - коэффициент полной массы;

χ - стреловидность поверхности;

ψ - величина штрафной функции;

Ω - область допустимых проектов.

СПИСОК ЛИТЕРАТУРЫ

1. Лукьянов, О. Е. Снижение энергетических затрат беспилотного летательного аппарата вертикального взлёта и посадки с использованием гибридных технических решений [Текст] / О. Е. Лукьянов, В. Х. Хоанг, В. А. Комаров [и др.] // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. – 2024. – Т. 23. – №1. – С. 38–54.
2. Комаров, В. А. Автоматизация концептуального проектирования и модификации беспилотных летательных аппаратов самолётного типа с использованием многодисциплинарной оптимизации и эволюционных алгоритмов. Часть 1: Методы и модели [Текст] / В. А. Комаров, О. Е. Лукьянов, В. Х. Хоанг [и др.] // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. – 2024. – Т. 23. – №3. – С. 42–57.
3. Комаров, В. А. Автоматизация концептуального проектирования и модификации беспилотных летательных аппаратов самолётного типа с использованием многодисциплинарной оптимизации и эволюционных алгоритмов. Часть 2: Результаты и анализ [Текст] / В. А. Комаров, О. Е. Лукьянов, В. Х. Хоанг [и др.] // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. – 2024. – Т. 23. – №4. – С. 48–64.
4. Lukyanov, O. Atmospheric Aircraft Conceptual Design Based on Multidisciplinary Optimization with Differential Evolution Algorithm and Neural Networks [Текст] / V. H. Hoang, E. Kurkin, J. G. Quijada-Pioquinto // Drones. – 2024. – Vol. 8, no. 8: 388.
5. Хоанг, В. Х. Выбор параметров облика несущей системы беспилотного летательного аппарата самолётного типа с использованием алгоритма дифференциальной эволюции [Текст] / В. Х. Хоанг, О. Е. Лукьянов, В. А. Комаров // Управление движением и навигация летательных аппаратов: сб. тр. XXVI Всерос. семинара по управлению движением и навигации летат. аппаратов - Самара: Изд-во Самар. ун-та, 14 - 16 июня 2023 г. - С. 177–182.
6. Хоанг, В. Решение задачи продольной балансировки беспилотного

летательного аппарата методами нелинейного математического программирования с помощью межплатформенной связи PYTHON и AVL [Текст] / В. Хоанг, О. Е. Лукьянов, В. А. Комаров // Всероссийская молодёжная научная конференция с международным участием «XVII Королёвские чтения», посвящённая 35-летию со дня первого полёта МТКС «Энергия-Буран», Самара, Россия, 03-05 Октября, 2023 г. В 2 томах. Том 1. – С. 130.

7. Hoang, V. H. Acceleration of the convergence of the iterative solution for the aircraft's sizing equation when optimizing its parameters [Текст] / V. H. Hoang, O. E. Lukyanov // 22-st International Conference on the Methods of Aerophysical Research, Novosibirsk, Russia, July 1 – 5, 2024: Abstracts. Pt. II / Ed. E.I. Kraus; Ministry of Science and Higher Education of the Russian Federation: on [et al.]. – Novosibirsk: SB RAS, 2024. № 22 (2). – p. 75–77.

8. Воздушный кодекс Российской Федерации. ВЗК РФ Статья 32. Воздушное судно [Электронный ресурс]. URL: https://www.consultant.ru/document/cons_doc_LAW_13744/ (дата обращения 03.12.2024).

9. The Free Dictionary. "Unmanned aerial vehicle". Dictionary of Military and Associated Terms, US Department of Defense, 2005. [Электронный ресурс]. URL: <https://www.thefreedictionary.com/unmanned+aerial+vehicle> (дата обращения 03.12.2024).

10. Wagner, W. Fireflies and other UAVs [Текст] / W. Wagner, W. P. Sloan. – Texas: Aerofax, Inc., 1992. – 205 p.

11. Garg, P. K. Unmanned aerial vehicles [Текст] / P. K. Garg. – Mercury learning and information LLC., 2021. – 413 p.

12. Павлушенко, М. БЕСПИЛОТНЫЕ ЛЕТАТЕЛЬНЫЕ АППАРАТЫ: история, применение, угроза распространения и перспективы развития [Текст] / М. Павлушенко, Г. Евстафьев, И. Макаренко. – М.: Права человека, 2005. – 610 с.

13. Abdelkader, M. Unmanned Aerial Vehicles Applications: Challenges and Trends [Текст] / M. Abdelkader, A. Koubaa. – Springer, 2023. – 409 p.

14. Петушкова, В. Б. История создания беспилотных летательных аппаратов [Текст] / В. Б. Петушкова, С. О. Потапова // Пожарная безопасность: проблемы и перспективы. – 2018. – Т. 1, № 9. – С. 713-717.

15. Palik, M. Brief history of uav development [Текст] / M. Palik, M. Nagy // Repüléstudományi Közlemények. – 2019. – Vol. 31, no. 1. – pp. 155-166.

16. Liu, L. MEMS Sensor Data Anomaly Detection for the UAV Flight Control Subsystem [Текст] / L. Liu, M. Liu, Q. Guo, D. Liu, Y. Peng // 2018 IEEE SENSORS, New Delhi, India, 28-31 October, 2018. – pp. 1-4.

17. Unmanned Aerial Vehicle (UAV) Drones Market Size, Share, and Trends 2024 to 2034 [Электронный ресурс]. URL: <https://www.precedenceresearch.com/unmanned-aerial-vehicle-drones-market> (дата обращения 03.12.2024).

18. Pasandideh, F. A Review of Flying Ad Hoc Networks: Key Characteristics, Applications, and Wireless Technologies [Текст] / F. Pasandideh, J.P.J. da Costa, R. Kunst, N. Islam, W. Hardjawana, E. Pignaton de Freitas // Remote Sensing. – 2022. – Vol. 14, no. 18: 4459.

19. Arjomandi, M. Classification of unmanned aerial vehicles [Текст] / M. Arjomandi. – The University of Adelaide Australia, 2007. – 48 p.

20. Беспилотная авиационная система с БВС ВТ 45 [Электронный ресурс]. URL: <https://radar-mms.com/product/bespilotnye-aviatsionnye-sistemy/bespilotnaya-aviatsionnaya-sistema-s-bvs-vt-45/> (дата обращения 03.12.2024).

21. Многоцелевой БПЛА Ка-37 [Электронный ресурс]. URL: <http://xn--80aafy5bs.xn--p1ai/aviamuseum/aviatsiya/rf/bpla/mnogotselevoj-bpla-ka-37/> (дата обращения 03.12.2024).

22. Агрокомплекс точного земледелия «Гектор» S-80 [Электронный ресурс]. URL: <https://tb-drone.ru/developments/agrodron/> (дата обращения 03.12.2024).

23. «Карго» S-700 [Электронный ресурс]. URL: <https://tb-drone.ru/developments/s-700/> (дата обращения 03.12.2024).

24. Global Drone Market By Type (Fixed-Wing Drone, Rotary Wing Drone), By Application (Consumer, Commercial (Construction, Agriculture, Oil & Gas, Mining, Public Safety & Law Enforcement, Others), Military), By Region and Companies - Industry Segment Outlook, Market Assessment, Competition Scenario, Trends, and Forecast 2024-2033 [Электронный ресурс]. URL: <https://market.us/report/drone-market/#overview> (дата обращения 03.12.2024).

25. Беспилотный летательный аппарат "Орион" [Электронный ресурс]. URL: <https://kronshtadt.ru/products/bespilotnyij-kompleks-orion> (дата обращения 03.12.2024).

26. В Минобороны заявили об окончании испытаний тяжелого разведывательно-ударного беспилотника «Альтиус-РУ» [Электронный ресурс]. URL: <https://topwar.ru/190430-v-minoborony-zajavili-ob-okonchanii-ispytanij-tjazhelogo-razvedyvatelyno-udarnogo-bespilotnika-altius-ru.html> (дата обращения 03.12.2024).

27. Adcom Unveils Innovative UAV at Dubai Air Show [Электронный ресурс]. URL: <https://www.ainonline.com/aviation-news/defense/2011-11-13/adcom-unveils-innovative-uav-dubai-air-show> (дата обращения 03.12.2024).

28. Predator: The Secret Origins of the Drone Revolution [Электронный ресурс]. URL: https://thehumanist.com/magazine/march-april-2015/arts_entertainment/predator-the-secret-origins-of-the-drone-revolution/ (дата обращения 03.12.2024).

29. Baykar Bayraktar TB.1 / TB.2 [Электронный ресурс]. URL: <https://www.avionslegendaires.net/avion-militaire/baykar-bayraktar-tb-1-tb-2/> (дата обращения 03.12.2024).

30. Корнеев, В. М. Беспилотные летательные аппараты самолётного типа: Анализ конструкции и эксплуатации (Russian Edition) [Текст] / В. М. Корнеев. – LAP LAMBERT Academic Publishing, 2018. – 88 с.

31. Вейгелин, К. Е. Очерки по истории летного дела [Текст] / К. Е. Вейгелин. – М.: Государственное издательство оборонной промышленности, 1940. – 459 с.

32. Niccoli, R. Book of flight : from the flying machines of Leonardo Da Vinci to the conquest of space [Текст] / R. Niccoli. – New York, NY: Friedman / Fairfax, 2002.

33. Egorov, B. The first Russian airplane (PICS) [Электронный ресурс] / В. Egorov. – URL: <https://www.rbth.com/history/334850-first-russian-airplane-pics> (дата обращения 03.12.2024).

34. Соболев, Д. А. История самолетов мира [Текст] / Д. А. Соболев.– М.: Русское авиационное общество, 2001. – 680 с.

35. Lilienthal, O. Birdflight as the Basis of Aviation: A Contribution Towards a System of Aviation [Текст] / O. Lilienthal, G. Lilienthal. – Markowski International Pub., 2018. – 176 p.

36. Вейгелин, К. Е. Завоевание воздушного океана: История и современное состояние воздухоплавания [Текст] / К. Е. Вейгелин. – СПб.: Типография П.П. Сойкина, 1912. – 168 с.

37. Болховитинов, В. Ф. Пути развития летательных аппаратов [Текст] / В. Ф. Болховитинов. – М.: Оборонгиз, 1962. – 132 с.

38. Самсонов, П. Д. Проектирование и конструкции гидросамолетов [Текст]: учеб. пособие для авиационных вузов / П. Д. Самсонов. – М.: НКТП ГУАП Моск. авиац. инст., 1936. – 541 с.

39. Вуд, К. Проектирование самолетов [Текст] / К. Вуд // Пер. с англ. под ред. инж. И. И. Эскина и инж. А. Д. Калужнина. – М.: Оборонгиз, 1940. – 470 с.

40. Сутугин, Л. И. Основы проектирования самолетов [Текст] / Л. И. Сутугин. – М.: Оборонгиз, 1945. – 267 с.

41. Бедункович, А. Г. Проектирование самолетов. Ч. 1. Общие вопросы проектирования [Текст] / А. Г. Бедункович. – Ленинград: типолитограф. Ленингр. краснознам. воен.-воздуш. инж. акад, 1948. – 291 с.

42. Свищёв, Г. П. Авиация – энциклопедия [Текст] / Г. П. Свищёв. – М.: Научное издательство «большая российская энциклопедия» центральный аэрогидродинамический институт имени профессора Н. Е. Жуковского, 1994. – 733 с.

43. Фомин, Н. А. Проектирование самолетов. Определение веса.

Компоновка. Выбор схемы и основных параметров [Текст] / Н. А. Фомин. – М.: Оборонгиз, 1961. – 362с.

44. Бадягин, А. А. Проектирование самолетов [Текст] / А. А. Бадягин. – М.: Машиностроение, 1972. – 516 с.

45. Егер, С. М. Проектирование самолетов [Текст] / С. М. Егер, В. Ф. Мишин, Н. К. Лисейцев. – М.: Машиностроение, 1983. – 616 с.

46. Проектирование самолетов [Текст] / под ред. М. А. Погосяна. – 5-е изд., перераб. и доп. – М.: Инновационное машиностроение, 2018. – 864 с.

47. Anderson, J. D. Aircraft performance and design [Текст] / J. D. Anderson. – The McGraw-Hill Companies, Inc., 1999. – 580 p.

48. Howe, D. Aircraft Conceptual Design Synthesis [Текст] / D. Howe. – Professional Engineering Publ. Ltd., 2000. – 447 p.

49. Torenbeek, E. Advanced aircraft design - conceptual design, analysis and optimization of subsonic civil airplanes [Текст] / E. Torenbeek. – A John Wiley & Sons, Ltd., Publ., 2013. – 410 p.

50. Sadraey, M. H. Aircraft design - A Systems Engineering Approach [Текст] / M. H. Sadraey. – A John Wiley & Sons, Ltd., Publ., 2013. – 778 p.

51. Sforza, P. M. Commercial airplane design principles [Текст] / P. M. Sforza. – Butterworth-Heinemann, Oxford, 2014. – 623 p.

52. Roskam, J. Airplane Design Part I: Preliminary Sizing of Airplanes, Design [Текст] / J. Roskam. – Analysis and Research Corporation, Lawrence Kansas. 5-th printing, 2017. – 206 p.

53. Raymer, D. P. Aircraft design: A Conceptual Approach [Текст] / D. P. Raymer. – Washington: Reston, VA: AIAA Education Series American Institute of Aeronautics and Astronautics, Inc. 6th ed., 2018. – 1062 p.

54. Афанасьев, П. П. Беспилотные летательные аппараты [Текст] / П. П. Афанасьев [и др.]. – М.: МАИ, 2008. – 656 с.

55. Zhang, M. RDS-SUMO: from lofting to physics-based grids [Текст] / M. Zhang, A. Rizzi // Aircraft Engineering and Aerospace Technology. – 2012. – Vol. 84, no. 3. – pp. 140-150.

56. Егер, С. М. Основы автоматизированного проектирования самолётов [Текст] / С. М. Егер, Н. К. Лисейцев, О. С. Самойлович. – М.: Машиностроение, 1986. – 232 с.

57. Schmit, L. A. Structural Design by Systematic Synthesis [Текст] / L. A. Schmit // 2nd Conference on Electronic Computation, ASCE, New York, NY, 8-9 September, 1960. – pp. 105-132.

58. Schmit, L. A. Synthesis of an Airfoil at Supersonic Mach Number [Текст] / L. A. Schmit, W. A. Thornton. – National Aeronautics and Space Administration, 1965. – 92 p.

59. Haftka, R. T. Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements [Текст] / R. T. Haftka // Tech. Rep. TN D-7264, NASA Langley Research Center, Hampton, VA, 1973.

60. Haftka, R. T. Optimization of Flexible Wing Structures Subject to Strength and Induced Drag Constraints [Текст] / R. T. Haftka // AIAA Journal. – 1977. – Vol. 15, no. 8. – pp. 1101-1106.

61. Haftka, R. T. Approximate Methods for Combined Thermal/Structural Design [Текст] / R. T. Haftka, C. P. Shore // Tech. Rep. TP-1428, NASA, June 1979.

62. Grossman, B. Integrated Aerodynamic/Structural Design of a Sailplane Wing [Текст] / B. Grossman, Z. Gurdal, G. J. Strauch, W. M. Eppard, R. T. Haftka // Journal of Aircraft. – 1988. – Vol. 25, no. 9. – pp. 855-860.

63. Grossman, B. Integrated Aerodynamic-Structural Design of a Transport Wing [Текст] / B. Grossman, R. T. Haftka, P. J. Kao, D. M. Polen, M. Rais-Rohani // Journal of Aircraft. – 1990. – Vol. 27, no. 12. – pp. 1050-1056.

64. Livne, E. Towards Integrated Multidisciplinary Synthesis of Actively Controlled Fiber Composite Wings [Текст] / E. Livne, L. A. Schmit, P. Friedmann // Journal of Aircraft. – 1990. – Vol. 27, no. 12. – pp. 979-992.

65. Livne, E. Integrated Aeroservoelastic Optimization: Status and Direction [Текст] / E. Livne // Journal of Aircraft. – 1999. – Vol. 36, no. 1. – pp. 122–145.

66. He, Y. Multidisciplinary Optimization of Multibody Systems with Application to the Design of Rail Vehicles [Текст] / Y. He, J. Mcphee // Multibody

System Dynamics. – 2005. – Vol. 14, no. 2. – pp. 111-135.

67. Чернышёв С. Л. Государственный научный центр ЦАГИ: 100 лет на службе Отечеству [Текст] / С. Л. Чернышёв // Общероссийский научно-технический журнал «Полёт». – 2018. – № 11. – С. 3-13.

68. Karali, H. Advanced UAV Design Optimization Through Deep Learning-Based Surrogate Models [Текст] / H. Karali, G. Inalhan, A. Tsourdos // MPDI - Aerospace. – 2024. – Vol. 11, no. 8: 669.

69. Гайнутдинова, Т. Ю. Алгоритмы определения пространственного положения беспилотного летательного аппарата относительно посадочной платформы с использованием компьютерного зрения [Текст] / Т. Ю. Гайнутдинова, С. В. Новикова, В. Г. Гайнутдинов, М. В. Трусфус, В. М. Литвин // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. – 2023. – Т. 22, № 4. – С. 37-51.

70. Rajagopal, S. Conceptual design of UAV using Kriging based multi-objective genetic algorithm [Текст] / S. Rajagopal, R. Ganguli // Aeronautical Journal. – 2008. – Vol. 112, no. 1137. – pp. 653 - 662.

71. Aström, K. J. Feedback Systems: An Introduction for Scientists and Engineers [Текст] / K. J. Aström, R. M. Murray. – Princeton, New Jersey 08540: Princeton University Press, 2008. – 408 p.

72. Беспилотные летательные аппараты. Основы устройства и функционирования [Текст] / под ред. И. С. Голубева, И. К. Туркина. – М.: Август-принт, 2010. – 654 с.

73. Беспилотные аппараты «БПЛА – 2024» [Текст]: сборник. ст. Междунар. молодежного форума., Белорус. гос. технол. ун-т ; отв. за изд. И. В. Войтов., 22–26 апреля 2024 г.

74. Александрович, Т. Д. К вопросу проектирования беспилотного летательного аппарата для решения разведывательных задач на море [Текст] / Т. Д. Александрович, И. К. Туркин // Электронный журнал «Труды МАИ». – 2014. – № 78.

75. Вейсхаар, Т. А. Человеческий фактор в проектировании авиационных

конструкций [Текст] / Т. А. Вейсхаар, В. А. Комаров // Общероссийский научно-технический журнал «Полёт». – 1998. – № 1. – С. 17-23.

76. Ил-96 [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/%D0%98%D0%BB-96> (дата обращения 03.12.2024).

77. Komarov, V. A. Aircraft design using a variable density model [Текст] / V. A. Komarov, A. V. Boldyrev, A. S. Kuznetsov, M. Y. Lapteva // Aircraft Engineering and Aerospace Technology: An International Journal. – 2012. – Vol. 84, no. 3. – pp. 162-171.

78. Weishaar, T. A. New word vistas Air and Space Power for 21-st century [Текст] / Т. А. Weishaar // Aircraft and propulsion. – 1995. – pp. 80-91.

79. Komarov, V. A. New Approach to Improving the Aircraft Structural Design process [Текст] / V. A. Komarov, T. A. Weishaar // Journal of Aircraft. – 2002. – Vol. 39, no. 2. – pp. 227-233.

80. Комаров, В. А. Точное проектирование [Текст] / В. А. Комаров // Онтология проектирования. – 2012. – Т. 5, № 3. – С. 8-23.

81. Гуменюк, А. В. Оценка весового совершенства конструкции крыла самолёта Як-130 [Текст] / А. В. Гуменюк, А. К. Ковалевский, В. А. Комаров, К. Ф. Попович // Общероссийский научно-технический журнал «Полёт». – 2004. – № 7. – С. 22-27.

82. Болдырев, А. В. Учет статической аэроупругости на ранних стадиях проектирования [Текст] / А. В. Болдырев, В. А. Комаров, М. Ю. Лаптева, К. Ф. Попович // Общероссийский научно-технический журнал «Полёт». – 2008. – № 1. – С. 34-39.

83. Лукьянов О. Е. Разработка методики выбора облика грузовых самолётов с использованием многодисциплинарной оптимизации [Текст]: дисс. канд. техн. наук: 05.07.02 / Лукьянов Олег Евгеньевич. – Самара, 2019. – 154 с.

84. Комаров, В. А. Безразмерный критерий силового совершенства конструкций [Текст] / В. А. Комаров // Известия Российской академии наук. Механика твёрдого тела. – 2018. – № 4. – С. 34-47.

85. Лазарев, И. Б. Математические методы оптимального проектирования конструкций [Текст] / И. Б. Лазарев. – Новосибирск: Новосибирский институт инженеров железнодорожного транспорта, 1974. – 193 с.
86. Павлов, А. А. Конструкторско-технологическое проектирование деталей из слоистых тканевых композитов [Текст] / А. А. Павлов // Онтология проектирования. – 2022. – Т. 12, № 3. – С. 393-404.
87. Łukaszewicz, G. Navier–Stokes Equations: An Introduction with Applications [Текст] / G. Łukaszewicz, P. Kalita. – Springer International Publishing, 2016. – 390 p.
88. Пакет программ инженерного анализа и суперкомпьютерного моделирования [Электронный ресурс]. URL: <http://logos.vniief.ru/>
89. ANSYS CFX computational fluid dynamics (CFD) [Электронный ресурс]. URL: <https://www.ansys.com/products/fluids/ansys-cfx>
90. Программный комплекс для конструкторов, инженеров и технологов [Электронный ресурс]. URL: <https://flowvision.ru/>
91. STAR-CD [Электронный ресурс]. URL: www.cd-adapco.com
92. Numeca: a new wave in fluid dynamics [Электронный ресурс]. URL: <http://www.numeca.com/>
93. OpenFOAM [Электронный ресурс]. URL: <https://www.openfoam.com/>
94. Raymer, D. Enhancing Aircraft Conceptual Design Using Multidisciplinary Optimization [Текст]: PhD dissertation / Daniel Raymer. – Institutionen för flygteknik, Stockholm, 2002. – 166 p.
95. Белоцерковский С. М. Тонкая несущая поверхность в дозвуковом потоке газа [Текст] / С. М. Белоцерковский. – М.: Наука, 1965. – 244 с.
96. Katz, J. Low-Speed Aerodynamics, Second Edition [Текст] / J. Katz, A. Plotkin. – Cambridge: Cambridge University Press, 2000. – 126 p.
97. Гудилин, А. В. Комплекс программ «Аргон» аэропрочностного проектирования самолета [Текст] / А. В. Гудилин, Д. Д. Евсеев, Ф. З. Ишмуратов, Е. К. Липин, В. Н. Маркин, В. А. Мосунов, И. М. Пантелеев, С. В.

Сотников, В. Е. Теняева, А. С. Тимонин, В. В. Чедрик // Ученые записки ЦАГИ. – 1991. – № 5.

98. Краснов, Н. Ф. Основы прикладной аэрогазодинамики [Текст] / Н. Ф. Краснов, Е. Э. Боровский, А. И. Хлупнов. – М.: Высшая школа, 1990. – 694 с.

99. Software for Aerodynamics and Aircraft Design [Электронный ресурс]. URL: <http://tornado.redhammer.se/>

100. XFLR5 [Электронный ресурс]. URL: <https://www.xflr5.tech/xflr5.htm>

101. Лукьянов, О. Е. Программа расчета аэродинамических характеристик крыла методом дискретных вихрей "DMVWing" / О. Е. Лукьянов [и др.] // Свидетельство РФ № 2017618598; получено 15.08.2017.

102. AVL Overview [Электронный ресурс]. URL: <https://web.mit.edu/drela/Public/web/avl/>

103. OpenVSP [Электронный ресурс]. URL: <https://openvsp.org/>

104. Житомирский, Г. И. Конструкция самолётов [Текст] / Г. И. Житомирский. – М.: Машиностроение, 1995. – 415 с.

105. Кива, Д. С. Научные основы интегрированного проектирования самолетов транспортной категории. Ч. 1 [Текст] / Д. С. Кива. – Х.: Нац. аэрокосм. ун-т им. Н.Е. Жуковского «Харьк. авиац. ин-т», 2014. – 419 с.

106. Кива, Д. С. Научные основы интегрированного проектирования самолетов транспортной категории. Ч. 2 [Текст] / Д. С. Кива, А. Г. Гребеников. – Х.: Нац. аэрокосм. ун-т им. Н.Е. Жуковского «Харьк. авиац. ин-т», 2014. – 439 с.

107. Кива, Д. С. Научные основы интегрированного проектирования самолетов транспортной категории. Ч. 3 [Текст] / Д. С. Кива, А. Г. Гребеников. – Х.: Нац. аэрокосм. ун-т им. Н.Е. Жуковского «Харьк. авиац. ин-т», 2014. – 439 с.

108. Проектирование гражданских самолетов: Теории и методы [Текст] / [И. Я. Катыврев, М. С. Неймарк, В. М. Шейнин и др.]; под ред. Г. В. Новожилова. – Москва: Машиностроение, 1991. – 666 с.

109. Новожилов, Г. В. Теория и практика проектирования пассажирских самолетов [Текст] / Г. В. Новожилов. – М.: Наука, 1976. – 440 с.

110. Комаров, В. А. Весовой анализ авиационных конструкций:

теоретические основы [Текст] / В. А. Комаров // Общероссийский научно-технический журнал «Полёт». – 2000. – № 1. – С. 31-39.

111. Комаров, В. А. Оценка весовой эффективности силовых схем несущих поверхностей [Текст] / В. А. Комаров, А. В. Гуменюк // Вестник Самарского государственного аэрокосмического университета им. академика С. П. Королёва (национального исследовательского университета). – 2002. – № 1. – С. 45-54.

112. Гуменюк А. В. Прогнозирование и контроль массы авиационных конструкций с использованием критерия «силовой фактор» [Текст]: дисс. канд. техн. наук: 05.07.02 / Гуменюк Александр Викторович. – Самара, 2004. – 189 с.

113. Шейнин, В. М. Весовое проектирование и эффективность пассажирских самолётов [Текст] / В. М. Шейнин, В. И. Козловский. – М.: Машиностроение, 1977. – 343 с.

114. Лазарев, А. Ю. К поиску оптимальной компоновки летательного аппарата [Текст] / А. Ю. Лазарев, В. Н. Семенов // Вестник Самарского государственного аэрокосмического университета. – 2012. – Т. 5, № 36. – С. 66-71.

115. Семенов, В. Н. Конструкции самолетов замкнутой и изменяемой схем [Текст] / В. Н. Семенов. – М.: ЦАГИ, 2006. – 228 с.

116. Ardila-Parra, S. A. Finite Element based Redesign and Optimization of Aircraft Structural Components using Composite Materials [Текст] / S. A. Ardila-Parra, C. M. Pappalardo, O. A. G. Estrada, D. Guida // International Journal of Applied Mathematics. – 2020. – Vol. 50, no. 4: 860.

117. Иваньшин, П. Н. Метод конечных элементов [Текст]: курс лекций: учебное пособие / П. Н. Иваньшин. – Казань: Казан. ун-т, 2013. – 120 с.

118. Болдырев, А. В. Развитие методики топологической оптимизации конструкции фюзеляжа в зоне большого выреза [Текст] / А. В. Болдырев, М. В. Павельчук, Р. Н. Синельникова // Вестник Московского авиационного института. – 2019. – Т. 26, № 3. – С. 62-71.

119. Болдырев, А. В. Использование высокоточного моделирования на ранних стадиях проектирования конструкций летательных аппаратов [Текст] / А. В. Болдырев, В. А. Комаров // Вестник Самарского государственного аэрокосмического университета. Авиационная и ракетно-космическая техника. – 2012. – Т. 1, № 32. – С. 15-23.

120. Кефели, А. И. О теоретических весах сооружений [Текст] / А. И. Кефели // Тр. ЛИИЖТ. – 1927. – Т. 96. – С. 247-266.

121. Комаров, А. А. Силовое конструирование [Текст] / А. А. Комаров // Труды Куйбышев, авиац. ин-та. Куйбышев.: Куйбышев, обл. гос. изд-во. – 1952. – №. 1. – С. 36-47.

122. Комаров, А. А. Основы проектирования силовых конструкций [Текст] / А. А. Комаров. – Куйбышев: Куйбышев, книжн. изд-во, 1965. – 88 с.

123. Резниченко, Г. А. Вычисление силового веса конструкций типа крыла с использованием метода конечных элементов [Текст] / Г. А. Резниченко // Автоматизация проектирования авиационных конструкций. Межвузовский сборник. КуАИ: Куйбышев. – 1979. – С. 77-83.

124. Резниченко, Г. А. Оценка массы авиационных конструкций на основе конечно-элементных моделей [Текст]: дисс. канд. техн. наук: 05.07.02 / Резниченко Геннадий Алексеевич. – Куйбышев, 1988. – 157 с.

125. Комаров, В. А. Критерий силового совершенства конструкций крыльев [Текст] / В. А. Комаров, А. В. Гуменюк // Общероссийский научно-технический журнал «Полёт». – 2003. – № 6. – С. 24-30.

126. Gu, H. Coordinate Descent Optimization for Winged-UAV Design [Текст] / H. Gu, X. Lyu, Z. Li, F. Zhang // Intell. Robot. Syst. – 2009. – no. 97. – pp. 109-124.

127. Ming, L. A gradient-based aero-stealth optimization design method for flying wing aircraft [Текст] / L. Ming, B. Junqiang, L. Li, M. Xiaoxuan, L. Qian, C. Bao // Aerosp. Sci. Technol. – 2019. – no. 92. – pp. 156-169.

128. Safavi, E. Collaborative Multidisciplinary Design Optimization. A Framework Applied on Aircraft Conceptual System Design [Текст] / E. Safavi, M.

Tarkian, H. Gavel, J. Ölvander // *Concurr. Eng.* – 2015. – no. 23. – pp. 236-249.

129. Talbi, E.G. *Metaheuristics: from Design to Implementation* [Текст] / E. G. Talbi. – John Wiley & Sons, 2009. – 624 p.

130. Luo, J. An improved grasshopper optimization algorithm with application to financial stress prediction [Текст] / J. Luo, H. Chen, Y. Xu, H. Huang, X. Zhao // *Appl. Math. Model.* – 2018. – no. 64. – pp. 654-668.

131. Wang, M. Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses [Текст] / M. Wang, H. Chen, B. Yang, X. Zhao, L. Hu, Z. Cai, H. Huang, C. Tong // *Neurocomputing.* – 2017. – no. 267. – pp. 69-84.

132. Shen, L. Evolving support vector machines using fruit fly optimization for medical data classification [Текст] / L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, D. Liu // *Knowl.-Based Syst.* – 2016. – no. 96. – pp. 61-75.

133. Zhang, Q. Chaos enhanced bacterial foraging optimization for global optimization [Текст] / Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu, C. Li // *IEEE Access.* – 2018. – no. 6. – pp. 64905-64919.

134. Rai, D. Bio-inspired optimization techniques: a critical comparative study [Текст] / D. Rai, K. Seth // *ACM SIGSOFT Software Engineering Notes.* – 2013. – Vol. 38, no. 4. – pp. 1-7.

135. Cherniaev, A. Multistep Optimization of Composite Drive Shaft Subject to Strength, Buckling, Vibration and Manufacturing Constraints [Текст] / A. Cherniaev, V. Komarov // *Appl. Compos. Mater.* – 2015. – no. 22. – pp. 475-487.

136. Kurkin, E. Topology Optimization and Efficiency Evaluation of Short-Fiber- Reinforced Composite Structures Considering Anisotropy [Текст] / E. Kurkin, O. U. Espinosa Barcenas, E. Kishov, O. Lukyanov // *Computation.* – 2024. – no. 12: 35.

137. Kurkin, E. Gate Location Optimization of Injection Molded Aerospace Brackets Using Metaheuristic Algorithms [Текст] / E. Kurkin, E. Kishov, O. U. Espinosa Barcenas, V. Chertykovtseva // *Proceedings of the 2021 International Scientific and Technical Engine Conference (EC), Samara, Russia, 23-25 June,*

2021. – pp. 1-6.

138. Boutemedjet, A. UAV aerodynamic design involving genetic algorithm and artificial neural network for wing preliminary computation [Текст] / A. Boutemedjet, M. Samardžić, L. Rebhi, Z. Rajić, T. Mouada // *Aerosp. Sci. Technol.* – 2019. – no. 84. – pp. 464-483.

139. Simanowitsch, D. Comparison of Gradient-Based and Genetic Algorithms for Laminar Airfoil Shape Optimization [Текст] / D. Simanowitsch, A. Theiss, A. Sudhi, C. Badrya // *Proceedings of the AIAA SciTech 2022 Forum, ARC. AIAA Scitech Forum 2022, San Diego, CA, USA, 3-7 January, 2022.* – pp. 1-21.

140. Zingg, D. W. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization [Текст] / D. W. Zingg, M. Nemec, T. H. Pulliam // *Eur. J. Comput. Mech.* – 2008. – no. 17. – pp. 103-126.

141. Storn, R. Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces [Текст] / R. Storn, K. Price // *Journal of Global Optimization.* – 1997. – no. 11. – pp. 341-359.

142. Barcenas, O. U. E. Multidisciplinary Analysis and Optimization Method for Conceptually Designing of Electric Flying-Wing Unmanned Aerial Vehicles [Текст] / O. U. E. Barcenas, J. G. Q. Pioquinto, E. Kurkina, O. Lukyanov // *Drones.* – 2022. – Vol. 6, no. 10: 307.

143. Almufti, S. Comparative study between elephant herding optimization (EHO) and U-turning ant colony optimization (U-TACO) in solving symmetric traveling salesman problem (STSP) [Текст] / S. Almufti, R. Marqas, R. Asaad // *Journal Of Advanced Computer Science & Technology.* – 2019. – Vol. 8, no. 2: 32.

144. Gonzales, G. V. Comparison between differential evolution and simulated annealing algorithms applied to the constructal design of double-T shaped cavities [Текст] / G. V. Gonzales, L. A. Isoldi, L. A. O. Rocha, E. D. dos Santos, A. J. S. Neto // *Proceedings of the XXI ENMC – National Meeting on Computational Modeling and IX ECTM – Materials Science and Technology Meeting, Rio de Janeiro, Brazil, 8-11 October, 2018.*

145. Saruhan, H. Differential Evolution and Simulated Annealing Algorithms

for Mechanical Systems Design [Текст] / Н. Saruhan // Eng. Sci. Technol. Int. – 2014. – no. 17. – pp. 131-136.

146. Kurkin, E. Identification of a mathematical model of elastic-plastic behavior of short-reinforced composite materials using evolutionary algorithms [Текст] / E. Kurkin, E. Kishov, J. G. Quijada Pioquinto, A. Gavrilov, V. Chertykovtseva // Proceedings of the X International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russia, 20-24 May, 2024. – pp. 1-6.

147. Мальчевский, В. В. Матрично-топологический метод синтеза и компоновки самолета (опыт автоматизации творческой деятельности конструктора) [Текст] / В. В. Мальчевский. – М.: МАИ, 2011. – 356 с.

148. Лазарев, И. Б. Основы оптимального проектирования конструкций. Задачи и методы [Текст] / И. Б. Лазарев. – Новосибирск: Сибирская государственная академия путей сообщения, 1995. – 295 с.

149. Малков, В. П. Оптимизация упругих систем [Текст] / В. П. Малков, А. Г. Угодчиков. – М.: Наука, 1981. – 288 с.

150. The General Atomics “Predator MQ-1” UAS [Электронный ресурс]. URL: https://barnardmicrosystems.com/UAV/uav_list/predator.html (дата обращения 03.12.2024).

151. Средневысотный беспилотный летательный аппарат большой продолжительности полёта United 40 [Электронный ресурс]. URL: <https://vpk.name/library/f/united-40.html> (дата обращения 03.12.2024).

152. Корольков, О. Н. Уравнение и область существования самолёта [Текст] / О. Н. Корольков // Общероссийский научно-технический журнал «Полёт». – 2001. – № 10. – С. 45-52.

153. Стариков, Ю. Н. Основы аэродинамики летательного аппарата [Текст]: учеб. пособие / Ю. Н. Стариков, Е. Н. Коврижных. – Ульяновск: УВАУ ГА, 2004. – 151 с.

154. Балакин, В. Л. Динамика полёта самолёта: устойчивость и управляемость продольного движения [Электронный ресурс]: электрон. курс лекций / В. Л. Балакин, Ю. Н. Лазарев. – Самара: Самар. гос. аэрокосм. ун-т

им. С. П. Королёва (нац. исслед. ун-т), 2011. – 48 с.

155. Yeniay, Ö. Penalty Function Methods for Constrained Optimization with Genetic Algorithms [Текст] / Ö. Yeniay // *Mathematical and Computational Applications*. – 2005. – Vol. 10, no. 1. – pp. 45-46.

156. Olorunsola, S. A. On Application of Modified Lagrange Multipliers for Solving Optimization Problems in both Equality and Inequality Constraints [Текст] / S. A. Olorunsola, T. E. Olaosebikan, K. J. Adebayo // *IOSR Journal of Mathematics*. – 2014. – Vol. 10, no. 3. – pp. 06-09.

157. Budziak, K. Aerodynamic Analysis with Athena Vortex Lattice (AVL) [Текст] / K. Budziak. – Hamburg: Hamburg university of applied sciences, 2015. – 72 p.

158. Komarov, V. A. RE0-1386 Research of Morphing Wing Efficiency. Final Technical Report / Samara State Aerospace University, AVIKON Institute. 1 October 2003 – 1 June 2004.

159. Холявко, В. И. Расчёт аэродинамических характеристик самолёта Ч. 1 [Текст]: учеб. пособие / В. И. Холявко. – Харьков: Харьк. авиац. ин-т, 1991. – 72 с.

160. Балакин, В. Л. Динамика полета самолета. Расчет траекторий и летных характеристик: Конспект лекций [Текст] / В. Л. Балакин, Ю. Н. Лазарев. – Самара: Самар, гос. аэрокосм. ун-т, 2002. – 56 с.

161. Бадягин, А. А. Проектирование легких самолётов [Текст] / А. А. Бадягин, Ф. А. Мухамедов. – М.: Машиностроение, 1978. – 208 с.

162. Tanabe, R. Success-history based parameter adaptation for Differential Evolution [Текст] / R. Tanabe, A. Fukunaga // *2013 IEEE Congress on Evolutionary Computation*, Cancun, Mexico, 20-23 June, 2013. – pp. 71-78.

163. Ali, M. M. A penalty function-based differential evolution algorithm for constrained global optimization [Текст] / M. M. Ali, W. X. Zhu // *Computational Optimization and Applications*. – 2013. – Vol. 54, no. 3. – pp. 707-739.

164. Viana, F. A. A tutorial on Latin hypercube design of experiments [Текст] / F. A. Viana // *Quality and reliability engineering international*. – 2015. – Vol. 32,

no. 5. – pp. 1975-1985.

165. Анвар, М. Создание конвейеров данных на Python. Почему альтернатива без кода лучше? [Электронный ресурс] / М. Анвар // Astera. – 2024. – URL: <https://www.astera.com/ru/type/blog/data-pipelines-in-python/> (дата обращения 03.12.2024).

166. FreeCAD - Your own 3D parametric modeler [Электронный ресурс]. URL: <https://www.freecad.org/>

167. Zhang, J. JADE: Adaptive differential evolution with optional external archive [Текст] / J. Zhang, A. C. Sanderson // IEEE Transactions on Evolutionary Computation. – 2009. – Vol. 13, no. 5. – pp. 945-958.

168. Mallipeddi, R. Empirical Study on the Effect of Population Size on Differential Evolution Algorithm [Текст] / R. Mallipeddi, P. N. Suganthan // 2008 IEEE Congress on Evolutionary Computation, Hong Kong, China, 01-06 June, 2008. – pp. 3663-3670.

169. Lampinen, J. On Stagnation of the Differential Evolution Algorithm [Текст] / J. Lampinen, I. Zelinka // Proceedings of MENDEL 2000, 6th Int. Mendel Conf. on Soft Computing, Brno, Czech Republic, 07-09 June, 2000.

170. Лукьянов, О. Е. Удалённое управление экспериментальной установкой и автоматизация обработки экспериментальных данных [Текст] / О. Е. Лукьянов, Е. В. Тарасова, В. А. Мартынова // Известия Самарского научного центра РАН. – 2017. – Т. 19, № 1. – С. 128-132.

171. Лукьянов, О. Е. Экспериментальные исследования аэродинамических характеристик модели корпуса несущей формы [Текст] / О. Е. Лукьянов, Е. В. Тарасова, В. А. Мартынова // Известия Самарского научного центра РАН. – 2016. – Т. 18, № 1. – С. 83-89.

172. Piotrowski, P. Evaluation Metrics for Wind Power Forecasts: A Comprehensive Review and Statistical Analysis of Errors [Текст] / P. Piotrowski, I. Rutyna, D. Baczyn´ski, M. Копыт // Energies. – 2023. – Vol. 15, no. 24: 9657.

173. Patro, S. G. K. Normalization: A Preprocessing Stage [Текст] / S. G. K. Patro, K. K. Sahu // International Advanced Research Journal in Science,

Engineering and Technology. – 2015. – Vol. 2, no. 3.

174. Stewart, J. Multivariable Calculus, 7th Edition [Текст] / J. Stewart. – Cengage Learning Publ., 2011. – 600 p.

175. Козлов, Д. М. О настройке алгоритмов случайного поиска по тестовым моделям [Текст] / Д. М. Козлов, В. А. Комаров // Автоматизированное оптимальное проектирование инженерных объектов и технологических процессов: материалы всесоюзной школы-семинара. Ч.1 Горький: Горьковский государственный университет им. Н. И. Лобачевского. – 1974. – С. 67-70.

176. Chen, S. Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution [Текст] / S. Chen, J. Montgomery, A. Bolufé-Röhler // Applied Intelligence. – 2015. – no. 42. – pp. 514-526.

ПРИЛОЖЕНИЕ А

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «АСП_КП»

А.1. Создание исходной популяции***Create_Init_Pop.py***

```

import numpy as np
from smt.sampling_methods import LHS
des_var = np.array([[4, 20],           # Удлинение передней поверхности           0
                   [0, 10],          # Стреловидность передней поверхности      1
                   [1, 3],           # Сужение передней поверхности            2
                   [0, 2.5],         # Угол установки передней поверхности      3
                   [4, 20],          # Удлинение задней поверхности            4
                   [0, 10],          # Стреловидность задней поверхности        5
                   [1, 3],           # Сужение задней поверхности              6
                   [3, 6],           # Положение задней поверхности            7
                   [0.2, 5],         # Относительная площадь                 8
                   [30, 90],         # Скорость                                9
                   [5, 110],         # Удельная нагрузка                     10
                   [500, 3000]])     # Оценочная величина взлётной массы     11

sampling = LHS(xlimits=des_var); variable = des_var[:,0].size; const = 0
init_pop = 10 * (variable-const); x = sampling(init_pop)
np.save(f'initial_population_{init_pop}', x)

```

А.2. Расчёт аэродинамических характеристик***Aerodynamic.py***

```

import math
import numpy as np
def inputValuesAtmospheric(V, ro = 1.225):
    q = ro * V**2 / 2
    return q
def inp_profile(thick = 0.12, x_t = 0.3):
    return thick, x_t
def inp_f_cond(velocity = 0, angel_attack = 0, altitude = 0):
    velocity = np.around(velocity, decimals=3)
    angel_attack = np.around(angel_attack, decimals=3)
    altitude = np.around(altitude, decimals=3)
    return velocity, angel_attack, altitude

def inp_sf(lamda=0, khi=0, eta=0, S=0, location_x=0, teta=0, NC=6, NS=24):
    return lamda, khi, eta, S, location_x, teta, NC, NS
def sf_definition(sf):
    if sf[3] == 0:
        l = 0; b_0 = 0; b_k = 0
    else:
        l = np.sqrt(sf[3] * sf[0])
        b_k = 2 * sf[3] / (1 + sf[2])
        b_0 = b_k * sf[2]
    return l, b_0, b_k
def inp_fuse(lamda_n=0, lamda_c=0, lamda_t=0, hi=0, wi=0, NC=20, NS=15, x=0, y=0, z=0):
    return [lamda_n, lamda_c, lamda_t, hi, wi, NC, NS, x, y, z]
def fuse_definition(fuse):
    if fuse[0] == 0:
        d_fuse = 0; l_fuse = 0; S_fuse = 0; S_surround_fuse = 0

```

```

else:
    if fuse[3] == fuse[4]:
        d_fuse = fuse[4]
    else:
        d_fuse = 2 * np.sqrt(fuse[3] * fuse[4] / math.pi)
        l_n = fuse[0]*d_fuse
        l_c = fuse[1]*d_fuse
        l_t = fuse[2]*d_fuse
        l_fuse = l_n + l_c + l_t
        R_fuse = d_fuse/2
        S_fuse = math.pi*R_fuse**2
        S_surround_n=math.pi*R_fuse*((R_fuse**2+4*l_n**2)**(3/2)-R_fuse**3)/(6*l_n**2)
        S_surround_c=math.pi*d_fuse*l_c
        S_surround_t=math.pi*R_fuse*((R_fuse**2+4*l_t**2)**(3/2)-R_fuse**3)/(6*l_t**2)
        S_surround_fuse = S_surround_n+S_surround_c+S_surround_t
    return d_fuse, l_fuse, S_fuse, S_surround_fuse
def sf_section(sf):
    l, b_0, b_k = sf_definition(sf)
    x = 1/2* np.tan(np.radians(sf[1])); y = 1/2; z = 0
    x = [0, x]; y = [0, y]; z = [0, y]; b = [b_0, b_k]
    return x, y, z, b
def M_R_num(velocity, cax, altitude=0):
    sound_speed = np.sqrt(1.4*287*(288-0.0065*altitude)); mach_number = velocity/sound_speed
    re_number = mach_number*cax*(2.33*(1 - 0/12 + altitude**2/535)*(10**7))
    return mach_number, re_number
def sf_ref(sf):
    if sf[2] != 0:
        l, b_0, b_k = sf_definition(sf)
        cax = 2/3 * b_0 * (1 + sf[2] + sf[2]**2) / (1 + sf[2]) / sf[2]
    else:
        cax = 0
    return cax
def horizontal_statistics_volume(forward_wing, aftward_wing, x_G):
    if aftward_wing[0] == 0:
        htail_vol = 0; htail_arm = 0
    else:
        forward_wing_cax = sf_ref(forward_wing)
        forward_l, forward_b_0, forward_b_k = sf_definition(forward_wing)
        aftward_wing_cax = sf_ref(aftward_wing)
        aftward_l, aftward_b_0, aftward_b_k = sf_definition(aftward_wing)
        forward_y_cax = forward_l*(forward_wing[2]+2)/6/(forward_wing[2]+1)
        forward_x_cax = forward_y_cax * np.tan(np.radians(forward_wing[1]))
        aftward_y_cax = aftward_l*(aftward_wing[2]+2)/6/(aftward_wing[2]+1)
        aftward_x_cax = aftward_y_cax * np.tan(np.radians(aftward_wing[1]))
        if forward_wing[3] >= aftward_wing[3]:
            x_G_abs = forward_x_cax + x_G * forward_wing_cax
            htail_arm = (aftward_wing[4]*forward_wing_cax - x_G_abs + aftward_x_cax +
0.25*aftward_wing_cax)/forward_wing_cax
            htail_vol = htail_arm*aftward_wing[3]/forward_wing[3]
        else:
            x_G_abs = aftward_wing[4]*aftward_wing_cax + aftward_x_cax + x_G *
aftward_wing_cax

```

```
htail_arm = (x_G_abs - forward_x_cax - 0.25*forward_wing_cax)/aftward_wing_cax
```

```
htail_vol = htail_arm*forward_wing[3]/aftward_wing[3]
```

```
return htail_vol, htail_arm
```

A.3 Создание исходного файла ПО AVL

AVL_File.py

```
import Aerodynamic as ad
```

```
import os as hung
```

```
def file_for_software(forward_wing, aftward_wing, kk, x_G = 0):
```

```
    forward_l, forward_b_0, forward_b_k = ad.sf_definition(forward_wing)
```

```
    aftward_l, aftward_b_0, aftward_b_k = ad.sf_definition(aftward_wing)
```

```
    ref_area = forward_wing[3] + aftward_wing[3]
```

```
    forward_x, forward_y, forward_z, forward_b = ad.sf_section(forward_wing)
```

```
    aftward_x, aftward_y, aftward_z, aftward_b = ad.sf_section(aftward_wing)
```

```
    if forward_wing[3] >= aftward_wing[3]:
```

```
        cax = ad.sf_ref(forward_wing)
```

```
    else:
```

```
        cax = ad.sf_ref(aftward_wing)
```

```
    aftward_location = aftward_wing[4]*cax
```

```
    original_path = hung.path.expanduser('~')
```

```
    temple_path = hung.path.join(original_path, f'ACII_KII_{kk}/tmp/')
```

```
    file = f'Plane_{kk}.avl'
```

```
    plane_path = hung.path.join(temple_path, file)
```

```
    if hung.path.exists(plane_path):
```

```
        return 0
```

```
    with open(plane_path, 'a') as f:
```

```
        print('Plane', file=f)
```

```
        print('0      | Mach', file=f)
```

```
        print('0 0 0 | iYsym iZsym Zsym', file=f)
```

```
        if forward_wing[3] >= aftward_wing[3]:
```

```
            print(f'{ref_area} {cax} {forward_l}', file=f)
```

```
        else:
```

```
            print(f'{ref_area} {cax} {aftward_l}', file=f)
```

```
        print(f'{x_G} 0 0', file=f)
```

```
        print('#=====', file = f)
```

```
        #The fist wing
```

```
        if forward_wing[0] == 0:
```

```
            pass
```

```
        else:
```

```
            print('SURFACE', file=f)
```

```
            print('The fist Wing', file=f)
```

```
            print('#Distr_discrete_vor', file = f)
```

```
            print(f'{forward_wing[6]} 1 {forward_wing[7]} -2', file=f)
```

```
            print('YDUPLICATE', file = f)
```

```
            print('0', file = f)
```

```
            print('ANGLE', file=f)
```

```
            print(f'{forward_wing[5]}', file=f)
```

```
            print('TRANSLATE', file=f)
```

```
            print('0 0 0', file=f)
```

```
            print(' the first ', file = f)
```

```
            print('SECTION', file=f)
```

```
            print(f'{forward_x[0]} {forward_y[0]} {forward_z[0]} {forward_b[0]} {0}', file=f)
```

```
            print('NACA', file=f)
```

```

if forward_wing[3] >= aftward_wing[3]:
    print('2212', file=f)
else:
    print('0012', file=f)
print(' the second ', file = f)
print('SECTION', file=f)
print(f'{forward_x[1]} {forward_y[1]} {forward_z[1]} {forward_b[1]} {0}', file=f)
print('NACA', file=f)
if forward_wing[3] >= aftward_wing[3]:
    print('2212', file=f)
else:
    print('0012', file=f)
print('#=====', file = f)
#The second Wing
if aftward_wing[0] == 0:
    pass
else:
    print('SURFACE', file = f)
    print('The second Wing', file = f)
    print(f'{aftward_wing[6]} 1 {aftward_wing[7]}', file = f)
    print('YDUPLICATE', file = f)
    print('0', file = f)
    print('ANGLE', file=f)
    print(f'{aftward_wing[5]}', file=f)
    print('TRANSLATE', file=f)
    print(f'{aftward_location} 0 0', file=f)
    print('#', file = f)
    print(' the first ', file = f)
    print('SECTION', file=f)
    print(f'{aftward_x[0]} {aftward_y[0]} {aftward_z[0]} {aftward_b[0]} {0}', file=f)
    print('NACA', file=f)
    if forward_wing[3] < aftward_wing[3]:
        print('0012', file=f)
    else:
        print('2212', file=f)
    print(' the second ', file = f)
    print('SECTION', file=f)
    print(f'{aftward_x[1]} {aftward_y[1]} {aftward_z[1]} {aftward_b[1]} {0}', file=f)
    print('NACA', file=f)
    if forward_wing[3] < aftward_wing[3]:
        print('0012', file=f)
    else:
        print('2212', file=f)
print('Hoang Van Hung', file = f)
return 0

```

A.4 Запуск ПО AVL

AVL_Run.py

```

def run_software(f_cond, kk):
    original_path = hung.path.expanduser('~')
    temple_path = hung.path.join(original_path, f'ACII_KII_{kk}/tmp/')
    software_path = hung.path.join(original_path, 'Avl/bin/avl')
    file = hung.path.join(temple_path, f'Plane_{kk}.avl')

```

```

results_file = hung.path.join(temple_path, 'calc.txt')
if hung.path.exists(results_file):
    return results_file
sound_speed = np.sqrt(1.4*287*(288 - 0.0065* f_cond[2]))
number_mach = f_cond[0]/sound_speed
command = f'load {file}\n oper\n m\n mn\n {number_mach}\n v\n {f_cond[0]}\n \n a a
{f_cond[1]}\n x\n ft\n{results_file}\n \n \n quit'
with open(hung.devnull, 'w') as FNULL:
    try:
        process = subprocess.Popen([software_path],stdin = subprocess.PIPE, stdout = FNULL,
shell = True)
        process.communicate(bytes(command, encoding='utf8'))
    except subprocess.CalledProcessError:
        print('ERROR')
    hung.remove(file)
return results_file
def coefficients_aero(forward_wing, aftward_wing, f_cond, kk, x_G):
    forward_l, forward_b_0, forward_b_k = ad.sf_definition(forward_wing)
    aftward_l, aftward_b_0, aftward_b_k = ad.sf_definition(aftward_wing)
    if forward_wing[3] >= aftward_wing[3]:
        cax = ad.sf_ref(forward_wing)
        forward_y_cax = forward_l*(forward_wing[2]+2)/6/(forward_wing[2]+1)
        forward_x_cax = forward_y_cax * np.tan(np.radians(forward_wing[1]))
        x_G_abs = forward_x_cax + x_G * cax
    else:
        cax = ad.sf_ref(aftward_wing)
        aftward_y_cax = aftward_l*(aftward_wing[2]+2)/6/(aftward_wing[2]+1)
        aftward_x_cax = aftward_y_cax * np.tan(np.radians(aftward_wing[1]))
        x_G_abs = aftward_wing[4]*cax + aftward_x_cax + x_G * cax
avl.file_for_software(forward_wing, aftward_wing, kk,x_G_abs)
results_file = run_software(f_cond, kk)
c_y = coefficient(results_file, 'CLtot')
c_xi = coefficient(results_file, 'CDtot')
m_z = coefficient(results_file, 'Cmtot')
hung.remove(results_file)
return c_xi, c_y, m_z
def coefficient(results_file, var_iden):
    ID = '{}'.format(var_iden)
    with open(results_file,'r') as ex:
        for line in ex:
            line = line.replace(' ', '/')
            line = line.replace('|', '/')
            if ID in line:
                dong = line.replace(' ', '')
                he_so = dong.split('/')
                for du_lieu in he_so:
                    if ID in du_lieu:
                        hs = du_lieu.split('=')
                        hs = hs[1].strip('\n')
                        if hs == '*****':
                            return 0
                return float(hs)

```

A.5 Расчёт профильного сопротивления ЛА

Parasite_Drag.py

```

import Aerodynamic as ad
import numpy as np
def profile_drag_sf(sf, profile_char, velocity, altitude=0):
    if sf[0] == 0:
        drag_profile = 0
    else:
        cax = ad.sf_ref(sf)
        mach_number, re_number = ad.M_R_num(velocity, cax, altitude)
        x_t_0 = profile_char[0] * profile_char[1] / (profile_char[0] + 0.02) + 0.95 / (re_number * 10
** -6 + 2.4)
        km = 1 + 0.35 * (mach_number ** 0.5)
        k_khi = (1 - 0.6 * np.sin(np.radians(sf[1])) ** 2) * np.cos(np.radians(sf[1])) ** 2
        xt = x_t_0 * km * k_khi
        et_c = 1 + 2 * profile_char[0] * np.e ** (-2.4 * xt) + 9 * profile_char[0] ** 2 * np.e ** (-4 * xt)
        et_M = (1 / (1 + 0.2 * mach_number ** 2) ** 0.5 + 0.055 * xt ** 2 * mach_number) * (1 +
5 * profile_char[0] * mach_number)
        c_f = (0.087 / (np.log10(re_number) - 1.6) ** 2) * (1 - xt) + 1.33 / re_number ** 0.5 * xt ** 0.5
        drag_profile = 2 * c_f * et_c * et_M
    return drag_profile
def profile_drag_fuse(fuse, velocity, altitude=0):
    if fuse[1] == 0:
        drag_profile = 0
    else:
        fuse_def = ad.fuse_definition(fuse)
        bet_t = np.arctan(1/2/fuse[2])
        mach_number, re_number = ad.M_R_num(velocity, fuse_def[1])
        x_t = (fuse[0]/(fuse[0] + fuse[1] + fuse[2]) + 1.5/(5 + re_number*10** -6)) * (1 +
0.15 * mach_number ** (2/3))
        c_f = 0.087 * (1 - x_t) / ((np.log10(re_number) - 1.6) ** 2) +
1.33 * np.sqrt(x_t) / np.sqrt(re_number)
        et_lamd = 1 + 0.5 * (2 - x_t) / (fuse[0] + fuse[1] + fuse[2]) + 1.5 / ((fuse[0] + fuse[1] +
fuse[2]) ** 2)
        et_M = (1 / np.sqrt(1 + 0.2 * mach_number ** 2) + 0.055 * (x_t ** 2) * mach_number) * (1 +
2 * mach_number * (fuse[0] + fuse[1] + fuse[2]) / (1 + (fuse[0] + fuse[1] + fuse[2]) ** 2))
        profile_drag_TB = c_f * et_lamd * et_M * 3.8 * (fuse[0] + fuse[1] + fuse[2])
        del_profile_drag = (0.04 / np.sqrt(profile_drag_TB)) * (np.tan(0.5 * bet_t)) ** (3/2)
        drag_profile = profile_drag_TB + del_profile_drag
    return drag_profile
def profile_drag(forward_wing, f_profile_char, aftward_wing, a_profile_char, vertical_wing,
fuse, velocity, altitude=0):
    profile_drag_forward_wing = profile_drag_sf(forward_wing, f_profile_char, velocity, altitude)
    profile_drag_aftward_wing = profile_drag_sf(aftward_wing, a_profile_char, velocity, altitude)
    profile_drag_vertical_wing = profile_drag_sf(vertical_wing, velocity, altitude)
    profile_drag_fus = profile_drag_fuse(fuse, velocity, altitude)
    fuse_def = ad.fuse_definition(fuse)
    ref_area = forward_wing[3] + aftward_wing[3]
    drag = profile_drag_forward_wing * forward_wing[3] / ref_area + profile_drag_aftward_wing
* aftward_wing[3] / ref_area + profile_drag_vertical_wing * vertical_wing[3] / ref_area +
profile_drag_fus * fuse_def[2] / ref_area
    return drag

```

A.6. Определение условия балансировки

Balance.py

```

import numpy as np
import Aerodynamic as ad
import Parasite_Drag as par
import AVL_Run as ar
def cy_bal(specific_load, dynamic_pressure, gamma = 0, overload = 1):
    return 9.81 *overload* specific_load * np.cos(np.radians(gamma))/dynamic_pressure
def attack_need(forward_wing,aftward_wing,specific_load,velocity,gamma,kk,altitude=0):
    dynamic_pressure = ad.inputValuesAtmospheric(velocity)
    c_y_need = c_y_bal(specific_load,dynamic_pressure,gamma)
    boulder_attack = np.arange(-20,21,40)
    matrix_c_y = []; matrix_m_z = []
    for attack in boulder_attack:
        f_cond = ad.inp_f_cond(velocity, attack)
        c_xi, c_y, m_z = ad.coefficients_aero(forward_wing, aftward_wing, f_cond, kk)
        matrix_c_y.append(c_y); matrix_m_z.append(m_z)
    attack_need = np.interp(c_y_need, matrix_c_y, boulder_attack)
    derivative_mz_cy = -(matrix_m_z[-1] - matrix_m_z[0]) / (matrix_c_y[-1] - matrix_c_y[0])
    x_F = derivative_mz_cy
    return attack_need, x_F, c_y_need
def balanc(forward_wing, f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse,
specific_load, velocity, gamma, sta_mar, kk):
    install_angle = np.arange(-5,6,10); attack_angle = np.arange(-10,11,20)
    matrix_c_x = np.zeros([len(install_angle),len(attack_angle)]); matrix_c_y =
np.zeros([len(install_angle),len(attack_angle)])
    matrix_m_z = np.zeros([len(install_angle),len(attack_angle)]); matrix_m_z0 =
np.zeros([len(install_angle),1])
    matrix_mz_CG = np.zeros([len(install_angle),len(attack_angle)]); matrix_K =
np.zeros([len(install_angle),len(attack_angle)])
    matrix_x_F = np.zeros([len(install_angle),1]); matrix_x_G_new =
np.zeros([len(install_angle),1])
    c_x0 = par_dra.profile_drag(forward_wing, f_profile_char, aftward_wing, a_profile_char,
vertical_wing, fuse, velocity)
    i = 0
    for i in range(len(install_angle)):
        j = 0
        for j in range(len(attack_angle)):
            aftward_wing[5] = install_angle[i]
            f_cond = ad.inp_f_cond(velocity, attack_angle[j])
            c_xi, c_y, m_z = ar.coefficients_aero(forward_wing, aftward_wing, f_cond, kk)
            c_x = c_xi + c_x0; K_ = c_y/c_x
            matrix_c_x[i,j] = c_x; matrix_c_y[i,j] = c_y; matrix_m_z[i,j] = m_z; matrix_K[i,j] = K_
            j+=1
            matrix_x_F[i] = -(matrix_m_z[i,-1] - matrix_m_z[i,0]) / (matrix_c_y[i,-1] - matrix_c_y[i,0])
            matrix_x_G_new[i] = matrix_x_F[i] + sta_mar
            matrix_m_z0[i] = np.interp(0, [matrix_c_y[i], matrix_m_z[i]])
            i+=1
        i = 0
    for i in range(len(install_angle)):
        j = 0
        for j in range(len(attack_angle)):

```

```

matrix_mz_CG[i,j] = (matrix_m_z[i,j] - matrix_m_z0[i]) * (-sta_mar) / matrix_x_F[i] +
matrix_m_z0[i]
dynamic_pressure = ad.inputValuesAtmospheric(velocity)
c_y_b = c_y_bal(specific_load, dynamic_pressure, gamma)
HS_m_z = polyfit2D(attack_angle,install_angle,matrix_mz_CG)
HS_c_y = polyfit2D(attack_angle,install_angle,matrix_c_y)
left = np.array([[HS_m_z[1],HS_m_z[2]], [HS_c_y[1],HS_c_y[2]]])
right = np.array([-HS_m_z[0],-HS_c_y[0]+c_y_b])
attack_angle_b, install_angle_b = np.linalg.solve(left, right)
if attack_angle_b >50:
    attack_angle_b = 50
if attack_angle_b <-50:
    attack_angle_b = -50
aftward_wing[5] = install_angle_b
x_G = matrix_x_G_new[0]
x_G = x_G[0]
f_cond = ad.inp_f_cond(velocity, attack_angle_b)
c_xi_b, c_y_b, m_z_b = ar.coefficients_aero(forward_wing, aftward_wing, f_cond, kk, x_G)
c_y_b = HS_c_y[0] + HS_c_y[1]*attack_angle_b + HS_c_y[2]*install_angle_b
m_z_b = HS_m_z[0] + HS_m_z[1]*attack_angle_b + HS_m_z[2]*install_angle_b
c_x_b = c_xi_b + c_x0
K_b = c_y_b/c_x_b
return attack_angle_b, install_angle_b, x_G, c_x0, c_x_b, c_y_b, m_z_b, K_b
def poly_fit_2D(attack_angle,install_angle,m_z_CG):
    attack_angle_new, install_angle_new = np.meshgrid(attack_angle, install_angle)
    attack_angle_new = attack_angle_new.flatten()
    install_angle_new = install_angle_new.flatten()
    Hung = np.array([attack_angle_new*0+1,attack_angle_new,install_angle_new]).T
    m_z_CG_new = m_z_CG.flatten()
    HS, res, rank, s = np.linalg.lstsq(Hung, m_z_CG_new,rcond=None)
    return HS

```

A.7. Решение уравнения существования ЛА

m0_Calculator.py

```

import Aerodynamic as ad
import m_Segment as m
import Sizing as sz
import pandas as pd
def m0_calc(par, mpay, w_rpm, t, sta_mar, type_power, gamma_rotor, Ce1, Ce2, kk, altitude=0):
    S_relative = par[kk][8];
    velocity = par[kk][9];
    p_0 = par[kk][10];
    m_in = par[kk][11];
    t_cr = t
    eff = [0.76, 0.75, 0.75] # Prop, motor, batt;
    gamma = [5, 0, -5]
    forward_wing_S = m_in/(p_0*(1+S_relative))
    aftward_wing_S = S_relative*forward_wing_S
    forward_wing = ad.inp_sf(par[kk][0], par[kk][1], par[kk][2], forward_wing_S, 0, par[kk][3])
    forward_l, forward_b_0, forward_b_k = ad.sf_definition(forward_wing)
    f_profile_char = ad.inp_profile(0.12)
    aftward_wing = ad.inp_sf(par[kk][4], par[kk][5], par[kk][6], aftward_wing_S, par[kk][7], -2)
    aftward_l, aftward_b_0, aftward_b_k = ad.sf_definition(aftward_wing)

```

```

a_profile_char = ad.inp_profile(0.12)
fuse = ad.inp_fuse(2, 4.308, 2, 1., 1.)
vertical_wing = ad.inp_sf(1.2734, 12.5, 2.3273, 3)
# Cruise
m_V_DBC_cr, D_V_DBC_cr, m_pow_cr, m_fue_cr, p2w_cr, c_x0, c_x_b_cr, c_y_b_cr,
m_z_b_cr, K_b_cr, x_G_cr, alpha_b_cr, delta_b_cr, T_cr, A = m.m_cruise(forward_wing,
f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse, velocity, p_0, m_in, eff, t_cr,
sta_mar, gamma[1], w_rpm, type_power, gamma_rotor, Ce2, kk, altitude)
mV = m_V_DBC_cr; m_pu = m_pow_cr; m_fue = m_fue_cr; p2w = p2w_cr
D_V = D_V_DBC_cr; c_x = c_x_b_cr; c_y = c_y_b_cr; m_z = m_z_b_cr; K = K_b_cr
x_G = x_G_cr; alpha_b = alpha_b_cr; delta_b = delta_b_cr
# Climb
m_V_DBC_cl, D_V_DBC_cl, m_pow_cl, m_fue_cl, p2w_cl, T_cl = m.m_seg(forward_wing,
f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse, 0.9*velocity, p_0, m_in, eff,
0.05*t_cr, sta_mar, gamma[0], w_rpm, type_power, gamma_rotor, Ce1, kk, altitude)
if m_fue_cl > 0:
    m_fue = m_fue + m_fue_cl
if m_V_DBC_cl > mV:
    D_V = D_V_DBC_cl; mV = m_V_DBC_cl
if m_pow_cl > m_pu:
    m_pu = m_pow_cl
if p2w_cl > p2w:
    p2w = p2w_cl
# Declimb
m_V_DBC_dcl, D_V_DBC_dcl, m_pow_dcl, m_fue_dcl, p2w_dcl, T_dcl =
m.m_seg(forward_wing, f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse,
0.9*velocity, p_0, m_in, eff, 0.1*t_cr, sta_mar, gamma[2], w_rpm, type_power, gamma_rotor,
Ce1, kk, altitude)
if m_fue_dcl > 0:
    m_fue = m_fue + m_fue_dcl
if m_V_DBC_dcl > mV:
    D_V = D_V_DBC_dcl; mV = m_V_DBC_dcl
if m_pow_dcl > m_pu:
    m_pu = m_pow_dcl
if p2w_dcl > p2w:
    p2w = p2w_dcl;
N = p2w*m_in
m_const_f_w = sz.m_constr_sf(forward_wing, m_in, 0.12)
m_const_a_w = sz.m_constr_sf(aftward_wing, m_in, 0.12)
m_const_v_w = sz.m_constr_VT(vertical_wing, velocity)
mSS = sz.m_SS(m_in)
mF = sz.m_constr_fuse(fuse, 1.3*velocity)
m_equip = 0.08
m_out = (mpay + mV + mSS + mF + m_const_v_w) / (1 - m_fue - m_pu - m_const_f_w -
m_const_a_w - m_equip)
if m_out <=0:
    m_out = 100000
m_V = mV/m_out; m_SS = mSS/m_out; m_F = mF/m_out;
m_const_v_w = m_const_v_w/m_out
m_constr = m_SS + m_F + m_const_v_w + m_const_f_w + m_const_a_w
return m_out, m_V, D_V, m_pu, m_fue, p2w, p2w_cr, N, m_constr, m_const_f_w,
m_const_a_w, m_const_v_w, m_F, m_SS, c_x0, c_x, c_y, m_z, K, x_G, alpha_b, delta_b,

```

```

forward_wing_S, aftward_wing_S, T_cl, T_cr, A
def save_input_DataFrame(plane):
    name_columns = ['f_w_aspect', 'f_w_sweep', 'f_w_taper', 'f_w_delta',
                    'a_w_aspect', 'a_w_sweep', 'a_w_taper', 'a_x_loc',
                    'S_rel', 'V', 'p0', 'm0_in']
    data_frame = pd.DataFrame(plane, columns = name_columns)
    return data_frame
def save_output_DataFrame(info_aircraft):
    name_columns = ['mtow_out', 'm_V', 'D_V', 'm_pu', 'm_fue', 'p2w', 'p2w_cr', 'N',
                    'm_constr', 'm_const_f_w', 'm_const_a_w', 'm_const_v_w', 'm_F', 'm_SS', 'cx0', 'cx', 'cy', 'mz', 'K',
                    'center_mass', 'alpha_bal', 'delta_bal', 'f_w_area', 'a_w_area', 'T_cl', 'T_cr', 'L']
    data_frame = pd.DataFrame(info_aircraft, columns = name_columns)
    return data_frame

```

A.8 Формулы расчёта масс

Sizing.py

```

import numpy as np
import Aerodynamic as ad
def m_battery(E, p2w, t, eta_motor, SoC_op=0.7):
    SEC = p2w * t # Удельный расход энергии
    return 9.81 * SEC / E / eta_motor / SoC_op / 10
def m_constr_sf(sf, m0, thickness = 0.12):
    k_mex = 0.95; k_constr = 0.95;
    k_MT = 0.85;
    teta = 0.9; phi = 0.93; n_p = 5.5
    coef = 1.15*10**(-4)*k_mex*k_constr*k_MT*phi*n_p
    lamda = sf[0]
    khi = sf[1]
    S = sf[3]
    eta = sf[2]
    m_constr = coef * lamda * np.sqrt(S) * (eta + 4) / (((np.cos(np.radians(khi)))**1.5) / np.sqrt(teta)
    / np.sqrt(thickness)/(eta+1)
    return m_constr
def m_constr_VT(sf, velocity):
    m = 6.8*(sf[3])**1.2*(0.4+(velocity+113)/1100)
    return m
def m_constr_fuse(fuse, velocity):
    d_fuse, l_fuse, S_fuse, S_surround_fuse = ad.fuse_definition(fuse)
    m = 0.23 * (S_surround_fuse**1.2) * np.sqrt(velocity * l_fuse / (fuse[3] + fuse[4]))
    return m
def m_SS(m0):
    k = 1.08; A1 = 11.3; B1 = 0; C1 = 0.0024; D1 = 0
    m1 = k*(A1+B1*m0**(3/4)+C1*m0+D1*m0**(3/2))
    A2 = 9.1; B2 = 0.082; C2 = 0.019; D2 = 0
    m2 = k*(A2+B2*m0**(3/4)+C2*m0+D2*m0**(3/2))
    m = m1+m2
    return m
def m_power(p2w, gamma):
    m_cy = gamma * p2w
    return m_cy
def m_fuel(p2w, t_flight, Ce = 0.285):
    m_f = p2w*Ce*t_flight
    return m_f

```

```

def pwr_to_weight(V, K, alpha, efficiency = 0.75, gamma = 0):
    p2w = (np.cos(np.radians(gamma)) + np.sin(np.radians(gamma)) * K) / (K *
np.cos(np.radians(alpha)) + np.sin(np.radians(alpha))) * V / efficiency / 100
    return p2w
def mvinta_DBC(T, w_rpm, H=0):
    k_m_V = 4; k_m_V_DBC = 0.1; rho0 = 1.225; rho = rho0*(1-H/44300)**4.256
    w_rps = w_rpm/60
    D_V_DBC = (T/2/k_m_V_DBC/rho/(w_rps**2))**(1/4)
    m_v_DBC = 2*k_m_V*D_V_DBC
    return m_v_DBC, D_V_DBC

```

A.9 Расчёт массы компонентов ЛА

m_Segment.py

```

import Balance as bl
import Aerodynamic as ad
import numpy as np
import Sizing as sz
import AVL_Run as ar
import Parasite_Drag as par
def m_cruise(forward_wing, f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse,
V, p_0, m0, eff, t, star_mar, gamma, w_rpm, type_power, gamma_rotor, Ce, kk, H=0):
    alpha_b, delta_b, x_G, c_x0, c_x_b, c_y_b, m_z_b, K_b = bl.balanc(forward_wing,
f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse, p_0, V, gamma, star_mar, kk)
    A, L = ad.vol_coeff(forward_wing, aftward_wing, x_G)
    rho_0 = 1.225; rho = rho_0*(1-H/44300)**4.256
    X_b = c_x_b*rho*V**2*(forward_wing[3]+aftward_wing[3])/2
    T = (X_b + m0 * 9.81 * np.sin(np.radians(gamma))) / np.cos(np.radians(alpha_b))
    p2w = sz.pwr_to_weight(V, K_b, alpha_b, eff[0], gamma)
    m_pow = sz.m_power(p2w, gamma_rotor)
    if (type_power == 'eltr'):
        m_fue = sz.m_batery(0.2, p2w, t, eff[1], eff[2])
    else:
        m_fue = sz.m_fuel(p2w, t, Ce)
    m_V_DBC, D_V_DBC = sz.mvinta_DBC(T, w_rpm, H)
    return m_V_DBC, D_V_DBC, m_pow, m_fue, p2w, c_x0, c_x_b, c_y_b, m_z_b, K_b, x_G,
alpha_b, delta_b, T, A
def m_seg(forward_wing, f_profile_char, aftward_wing, a_profile_char, vertical_wing, fuse, V,
p_0, m0, eff, t, star_mar, gamma, w_rpm, type_power, gamma_rotor, Ce, kk, H=0):
    alpha_b, x_F, c_y_b = bl.attack_need(forward_wing, aftward_wing, p_0, V, gamma, kk, H)
    x_G = x_F + star_mar; f_cond = ad.inp_f_cond(V, alpha_b)
    c_xi, c_y, m_z = ar.coefficients_aero(forward_wing, aftward_wing, f_cond, kk, x_G)
    c_x0 = par.dra.parasite_drag(forward_wing, f_profile_char, aftward_wing, a_profile_char,
vertical_wing, fuse, V, H)
    c_x = (c_xi + c_x0); K = c_y/c_x; rho_0 = 1.225; rho = rho_0*(1-H/44300)**4.256
    X = c_x*rho*V**2*(forward_wing[3]+aftward_wing[3])/2
    T = (X+m0*9.81*np.sin(np.radians(gamma)))/np.cos(np.radians(alpha_b))
    p2w = sz.pwr_to_weight(V, K, alpha_b, eff[0], gamma)
    m_pow = sz.m_power(p2w, gamma_rotor)
    if (type_power == 'eltr'):
        m_fue = sz.m_batery(0.2, p2w, t, eff[1], eff[2])
    else:
        m_fue = sz.m_fuel(p2w, t, Ce)
    m_V_DBC, D_V_DBC = sz.mvinta_DBC(T, w_rpm, H)

```

```
return m_V_DBC, D_V_DBC, m_pow, m_fue, p2w, T
```

A.10 Оптимизация

Optimization_SHADE.py

```
import numpy as np
import m0_calc as m0
import time
import oper_ev as ev
import os as hung
import shutil
import subprocess
original_path = hung.path.expanduser('~')
general_path = hung.path.join(original_path, 'АСПИ_КП_0/gen/')
FreeCAD_path = 'C:/Program Files/FreeCAD 0.21/bin/FreeCAD'
t = 24 # Ввести время полета
star_mar = -0.1 # Ввести запас статической устойчивости
mpay = 600 # Ввести полезную нагрузку
w_rpm = 5800 # Ввести скорость вращения лопастей вентилятора двигателя.
gamma_rotor = 0.87 # Ввести удельный вес двигателя
Ce_takeoff = 0.285 # Ввести удельный расход топлива двигателя в режиме набора высоты
Ce_cruise = 0.27 # Ввести удельный расход топлива двигателя в крейсерском режиме
type_power = 'DBC' # Ввести тип двигателя
cores = 4 # Ввести количество ядер параллельного вычисления
des_var = np.array([[4, 20], # Удлинение передней поверхности 0
[0, 10], # Стреловидность передней поверхности 1
[1, 3], # Сужение передней поверхности 2
[0, 2.5], # Угол установки передней поверхности 3
[4, 20], # Удлинение задней поверхности 4
[0, 10], # Стреловидность задней поверхности 5
[1, 3], # Сужение задней поверхности 6
[3, 6], # Положение задней поверхности 7
[0.2, 5], # Относительная площадь 8
[30, 90], # Скорость 9
[5, 110], # Удельная нагрузка 10
[500, 3000]]) # Оценочная величина взлётной массы 11

variable = des_var[:,0].size
init_pop = 10 * (variable); NP = init_pop
max_eval = variable * 1000; min_pop = variable
epsilon = 0.005; H = 10; p = 0.2; rA = 2.6
### Restrictions
bias_mz = 0.001; U_mtow = mpay*100; min_alpha = -10; max_alpha = 10
min_delta = -5; max_delta = 5; min_A = 0.2; max_A = 0.6; max_cy = 0.6
### First generation
error = np.array([]); Lmin = np.array([]); Lmax = np.array([])
Pg = np.load(f'initial_population_{NP}.npy')
start = time.time()
g = 0
print('-----')
print('Generation { } '.format(g))
print('Progress: ', end="")
calc= ev.multijob(Pg, mpay, w_rpm, t, star_mar, type_power, gamma_rotor, Ce_takeoff,
Ce_cruise, cores, NP, original_path)
info_aircraft = ev.vector_info(calc)
```

```

m0_val = calc[0]; m0cns = np.ones(init_pop)*500; mz_val = abs(calc[17]); alpha_val = calc[20]
cy_val = calc[16]; delta_val = calc[21]; A_val = calc[26]
psi = ev.pen_fun(mz_val, cy_val, delta_val, alpha_val, A_val, bias_mz, max_cy, min_delta,
max_delta, min_alpha, max_alpha, min_A, max_A)
U_mtow = ev.obj_max(psi, m0_val, NP, U_mtow)
des_var[11, 0] = ev.U_obj(psi, m0cns, NP, des_var[11,0])
des_var[11, 1] = ev.obj_max(psi, m0_val, NP, des_var[11,1])
Lx = ev.fit_fun(m0_val, psi, U_mtow)
Fx_m0 = m0_val
error = np.append(error, abs(max(Lx)-min(Lx))/max(Lx))
Lmin = np.append(Lmin, min(Lx))
Lmax = np.append(Lmax, max(Lx))
print(f'\nError: {np.around(error[-1],decimals=4)} \tLmax: {np.around(Lmax[-1],decimals=4)}
\tLmin: {np.around(Lmin[-1], decimals=4)} \tU*: {np.around(U_mtow, decimals=4)}')
index_opt = np.argmin(Lx); confi_opt = Pg[index_opt]
print('Optimum index: ', index_opt)
print('Optimum desing parameters vector: ', confi_opt)
print('-----')
Pg_0 = m0.save_input_DataFrame(Pg)
Pg_0.to_excel(general_path+'Px_0.xlsx')
info_aircraft0 = m0.save_output_DataFrame(info_aircraft)
info_aircraft0.to_excel(general_path+'info_aircraft_0.xlsx')
np.save(general_path+'Lx0.npy', Lx); np.save(general_path+'psi_0.npy', psi)
### New generation
MF = 0.5 * np.ones(H); MCR = 0.5 * np.ones(H)
A = np.empty([0, variable]); new_admin_pop = round(rA * NP);
k = 0; num_eval_f = NP; new_pop = NP
g = 1
while new_pop >= min_pop:
    new_pop_per = round(p * new_pop)
    Pg_best = ev.best_indiv(new_pop_per, Pg, Lx, variable)
    Dif_fit_mut = np.array([]) # Difference between muted and non-muted
    P_mut = np.zeros((new_pop,variable)); P_cross = np.zeros((new_pop,variable))
    SF = np.array([]); SCR = np.array([])
    Pg_new = np.zeros((new_pop, variable)); Lx_new = np.zeros(new_pop)
    info_aircraft_new = np.zeros((new_pop, 27))
    Fx_m0_new = np.zeros(new_pop); psi_new = np.zeros(new_pop)
    Fg = np.array([]); CRg = np.array([])
    print('Generation {}'.format(g))
    print('Progress: ', end="")
    # Mutation operation
    for i in range(new_pop):
        Fi, CRi = ev.operators(H, MF, MCR); Fg = np.append(Fg, Fi); CRg = np.append(CRg, CRi)
        best_ind = Pg_best[np.random.choice(new_pop_per)]
        P_mut[i] = ev.mut_oper(best_ind, new_pop, Pg, A, i, Fi, variable, des_var)
        P_cross[i] = ev.cross_oper(P_cross[i], P_mut[i], Pg[i], CRi, variable)
        P_cross[i] = np.clip(P_cross[i],des_var[:, 0],des_var[:, 1])
    calc = ev.multijob(P_cross, mpay, w_rpm, t, star_mar, type_power, gamma_rotor, Ce_takeoff,
Ce_cruise, cores, new_pop, original_path)
    info_aircraft_cross = ev.vector_info(calc); m0_mut = calc[0]; mz_mut = abs(calc[17])
    alpha_mut = calc[20]; delta_mut = calc[21]; cy_mut = calc[16]; A_mut = calc[26]
    print('#',end="")

```

```

psi_mut = ev.pen_fun(mz_mut, cy_mut, delta_mut, alpha_mut, A_mut, bias_mz, max_cy,
min_delta, max_delta, min_alpha, max_alpha, min_A, max_A)
U_mtow = ev.obj_max(psi_mut, m0_mut, new_pop, U_mtow)
Lx_mut = ev.fit_fun(m0_mut, psi_mut, U_mtow)
# Selection operation
for i in range(new_pop):
    if Lx_mut[i] < Lx[i]:
        A = np.row_stack((A, Pg[i]));
        SF = np.append(SF, Fg[i]); SCR = np.append(SCR, CRg[i])
        Dif_fit_mut = np.append(Dif_fit_mut, abs(Lx_mut[i] - Lx[i]))
        info_aircraft_new[i] = info_aircraft_cross[i]
        Pg_new[i] = P_cross[i]; Lx_new[i] = Lx_mut[i]
        Fx_m0_new[i] = m0_mut[i]; psi_new[i] = psi_mut[i]
    else:
        Pg_new[i] = Pg[i]; info_aircraft_new[i] = info_aircraft[i]
        Lx_new[i] = Lx[i]; Fx_m0_new[i] = Fx_m0[i]; psi_new[i] = psi[i]
if SF.size!=0:
    mSF = ev.Lehmer_weight_average(SF, Dif_fit_mut)
    mSCR = ev.Lehmer_weight_average(SCR, Dif_fit_mut)
    MF[k] = mSF
    if MCR[k]==-1 or max(SCR)==0:
        MCR[k] = -1
    else:
        MCR[k] = mSCR
    if k>=H-1:
        k=0
    else:
        k+=1
# m0 range update
des_var[11,0] = ev.U_obj(psi_new, Fx_m0_new, new_pop, des_var[11,0])
des_var[11,1] = ev.obj_max(psi_new, Fx_m0_new, new_pop, des_var[11,1])
A = ev.population_A(new_admin_pop, A)
num_eval_f += new_pop
new_pop = ev.pop_reduction_exponential(max_eval, NP, min_pop, num_eval_f)
error = np.append(error, abs(max(Lx_new)-min(Lx_new))/max(Lx_new))
Lmin = np.append(Lmin, min(Lx_new)); Lmax = np.append(Lmax, max(Lx_new))
np.save(general_path+'Lx'+str(g)+'.npy', Lx_new)
np.save(general_path+'MF'+str(g)+'.npy', MF)
np.save(general_path+'MCR'+str(g)+'.npy', MCR)
np.save(general_path+'A'+str(g)+'.npy', A)
np.save(general_path+'psi'+str(g)+'.npy', psi_new)
print(f'\nError:{ error[-1]}\tLmax:{ Lmax[-1]}\tLmin:{ Lmin[-1]}\tU*:{ U_mtow }')
index_opt = np.argmin(Lx_new); confi_opt = Pg_new[index_opt]
print('Optimum index: ', index_opt)
print('Optimum desing parameters vector: ', confi_opt)
print('-----')
Pgnew = m0.save_input_DataFrame(Pg_new)
Pgnew.to_excel(general_path+'Px_' + str(g) + '.xlsx')
info_aircraftnew = m0.save_output_DataFrame(info_aircraft_new)
info_aircraftnew.to_excel(general_path+'info_aircraft_' + str(g) + '.xlsx')
if ((max(Lx_new) - min(Lx_new))/max(Lx_new)) <= epsilon:
    break

```

```

Pg, Lx, info_aircraft, psi, Fx_m0 = ev.new_generation(new_pop, Pg_new, Lx_new,
info_aircraft_new, psi_new, Fx_m0_new)
g+=1
for i in range(init_pop-1):
    shutil.rmtree(original_path + f'/ACП_КП_{i+1}')
end = time.time()
print("\n-----")
print("Total calculation time: {}".format(end-start))
print('-----')
with open(hung.devnull, 'w') as FNULL:
    try:
        process = subprocess.run([FreeCAD_path], stdin=subprocess.PIPE, stdout = FNULL,
shell=True)
    except subprocess.CalledProcessError:
        print('ERROR')

```

A.11 Операторы алгоритма и параллельное вычисление

Operation_SHADE.py

```

import numpy as np
from joblib import Parallel, delayed
import os
import m0_Calculator as m0
def vector_info(calc):
    v_info = np.array([calc[0], calc[1], calc[2], calc[3], calc[4], calc[5], calc[6], calc[7], calc[8],
calc[9], calc[10], calc[11], calc[12], calc[13], calc[14], calc[15], calc[16], calc[17], calc[18],
calc[19], calc[20], calc[21], calc[22], calc[23], calc[24], calc[25], calc[26]])
    v_info = v_info.transpose()
    return v_info
def multijob(des_par, mpay, w_rpm, t, margin, type_power, gamma, Ce1, Ce2, cores, init_pop,
usr_path):
    for i in range(init_pop):
        if os.path.exists(usr_path + f'/ACП_КП_{i}'):
            pass
        else:
            os.mkdir(usr_path + f'/ACП_КП_{i}')
            os.mkdir(usr_path + f'/ACП_КП_{i}/tmp/')
            h = Parallel(n_jobs=cores)(delayed(m0.m0_calc)(des_par, mpay, w_rpm, t, margin,
type_power, gamma, Ce1, Ce2, kk) for kk in range(init_pop))
            mtow_out = np.array([]); m_V = np.array([]); D_V = np.array([]); m_pu = np.array([])
            m_fue = np.array([]); p2w = np.array([]); p2w_cr = np.array([]); N = np.array([])
            m_constr = np.array([]); m_const_f_w = np.array([]); m_const_a_w = np.array([])
            m_const_v_w = np.array([]); m_F = np.array([]); m_SS = np.array([])
            cx0 = np.array([]); cx = np.array([]); cy = np.array([]); mz = np.array([])
            K = np.array([]); center_mass = np.array([]); alpha_bal = np.array([])
            delta_bal = np.array([]); f_w_area = np.array([]); a_w_area = np.array([])
            T_cl = np.array([]); T_cr = np.array([]); A = np.array([])
            for kk in range(init_pop):
                mtow_out = np.append(mtow_out, t[kk][0])
                m_V = np.append(m_V, h[kk][1])
                D_V = np.append(D_V, h[kk][2])
                m_pu = np.append(m_pu, h[kk][3])
                m_fue = np.append(m_fue, h[kk][4])
                p2w = np.append(p2w, h[kk][5])

```

```

p2w_cr = np.append(p2w_cr, h[kk][6])
N = np.append(N, h[kk][7])
m_constr = np.append(m_constr, h[kk][8])
m_const_f_w = np.append(m_const_f_w, h[kk][9])
m_const_a_w = np.append(m_const_a_w, h[kk][10])
m_const_v_w = np.append(m_const_v_w, h[kk][11])
m_F = np.append(m_F, h[kk][12])
m_SS = np.append(m_SS, h[kk][13])
cx0 = np.append(cx0, h[kk][14])
cx = np.append(cx, h[kk][15])
cy = np.append(cy, h[kk][16])
mz = np.append(mz, h[kk][17])
K = np.append(K, h[kk][18])
center_mass = np.append(center_mass, h[kk][19])
alpha_bal = np.append(alpha_bal, h[kk][20])
delta_bal = np.append(delta_bal, h[kk][21])
f_w_area = np.append(f_w_area, h[kk][22])
a_w_area = np.append(a_w_area, h[kk][23])
T_cl = np.append(T_cl, h[kk][24])
T_cr = np.append(T_cr, h[kk][25])
A = np.append(A, h[kk][26])
return mtow_out, m_V, D_V, m_pu, m_fue, p2w, p2w_cr, N, m_constr, m_const_f_w,
m_const_a_w, m_const_v_w, m_F, m_SS, cx0, cx, cy, mz, K, center_mass, alpha_bal, delta_bal,
f_w_area, a_w_area, T_cl, T_cr, A
def pen_fun(mz, cy, delta, alpha, biaz_mz, max_cy, min_delta, max_delta, min_alpha,
max_alpha):
    pop_size = len(mz); psi = np.array([])
    for i in range(pop_size):
        ep_sum = 0
        ep_sum += max([0, (mz[i] - biaz_mz)]); ep_sum += max([0, (cy[i] - max_cy)])
        ep_sum += max([0, (min_delta - delta[i])]); ep_sum += max([0, (delta[i] - max_delta)])
        ep_sum += max([0, (min_alpha - alpha[i])]); ep_sum += max([0, (alpha[i] - max_alpha)])
        psi = np.append(psi, ep_sum)
    return psi
def fit_fun(obj, pen, max_obj, R=100):
    pop_size = len(obj); L_x = np.array([])
    for i in range(pop_size):
        if pen[i] == 0:
            L_x = np.append(L_x, obj[i])
        else:
            if obj[i] <= max_obj:
                L_x = np.append(L_x, R*pen[i]+max_obj)
            else:
                L_x = np.append(L_x, R*pen[i]+obj[i])
    return L_x
def best_indiv(NPp, Pg, Fx, var_num):
    Pgp = np.zeros((NPp, var_num))
    while Fx.size > Pg.size:
        Fx = np.delete(Fx, np.argmax(Fx))
    for i in range(NPp):
        best_ind = np.argmin(Fx); Pgp[i] = Pg[best_ind];
        Pg = np.delete(Pg, best_ind, axis=0); Fx = np.delete(Fx, best_ind)

```

```

    return Pgp
def pBestSelect(Lx, NP, p):
    Np = round(NP*p); indX = np.arange(Lx.size, dtype='int')
    Xcheck = Lx.copy(); pBest = np.array([], dtype='int')
    for i in range(Np):
        iBest = np.argmin(Xcheck); pBest = np.append(pBest, indX[iBest])
        Xcheck = np.delete(Xcheck, iBest); indX = np.delete(indX, iBest)
    return pBest
def oper_CR(H, MF):
    from scipy.stats import cauchy
    rand_unif_int = np.random.randint(0, H)
    while 1==1:
        Fe = cauchy.rvs(MF[rand_unif_int], 0.1)
        if Fe>1:
            Fi = 1; break
        elif Fe<=0:
            pass
        else:
            Fi = Fe; break
    return Fi
def operators(H, MF, MCR):
    from scipy.stats import cauchy
    rand_unif_int = np.random.randint(0, H)
    while 1==1:
        Fe = cauchy.rvs(MF[rand_unif_int], 0.1)
        if Fe>1:
            Fi = 1; break
        elif Fe<=0:
            pass
        else:
            Fi = Fe; break
    if MCR[rand_unif_int]==-1:
        CRi = 0
    else:
        CRi = np.random.normal(MCR[rand_unif_int],0.1)
    return Fi, CRi
def mut_oper(best_ind, new_pop, Pg, A, i, Fi, num_var, des_var):
    if A.size == 0:
        xi = np.arange(new_pop); xi = np.delete(xi, i)
        rs = np.random.choice(xi, 2, replace=False); xr = Pg[rs]
    else:
        xr = np.zeros((2,num_var))
        Pg_A = np.concatenate((Pg, A), axis=0); new_Pg_A = np.size(Pg_A, 0)
        x0 = np.arange(new_pop); x0 = np.delete(x0, i); r0 = np.random.choice(x0)
        x1 = np.arange(new_Pg_A); x1 = np.delete(x1, [i, r0]); r1 = np.random.choice(x1)
        xr[0] = Pg[r0]; xr[1] = Pg_A[r1]
        vi = Pg[i] + Fi * (best_ind - Pg[i]) + Fi * (xr[0] - xr[1])
    return vi
def cross_oper(Pgi_cross, Pgi_mut, Pgi, CRi, variable):
    for j in range(variable):
        if np.random.rand()<=CRi or j==np.random.randint(0,variable):
            Pgi_cross[j] = Pgi_mut[j]

```

```

    else:
        Pgi_cross[j] = Pgi[j]
    return Pgi_cross
def Lehmer_weight_average(SF, diff):
    S1 = 0; S2 = 0
    for i in range(SF.size):
        S2 += diff[i] * SF[i]**2 / sum(diff)
        S1 += diff[i] * SF[i] / sum(diff)
    if S1==0:
        mS = 0
    else:
        mS = S2/S1
    return mS
def population_A(new_A_pop, A):
    actual_pop = np.size(A,0)
    if actual_pop > new_A_pop:
        r = np.random.choice(np.arange(actual_pop), new_A_pop, replace=False)
        A_new = A[r]
    else:
        A_new = A
    return A_new
def pop_reduction_exponential(max_eval_f, init_pop, min_pop, num_eval_f):
    return round(init_pop * (min_pop / init_pop) ** (num_eval_f / max_eval_f))
def new_generation(new_pop, Pg_new, Lx_new, info_wing_new, psi_new, Fx_new):
    dx = Lx_new.size - new_pop
    for i in range(dx):
        del_index = np.argmax(Lx_new)
        Pg_new = np.delete(Pg_new, del_index, axis=0)
        info_wing_new = np.delete(info_wing_new, del_index, axis=0)
        Lx_new = np.delete(Lx_new, del_index)
        psi_new = np.delete(psi_new, del_index)
        Fx_new = np.delete(Fx_new, del_index)
    return Pg_new, Lx_new, info_wing_new, psi_new, Fx_new
def U_obj(pen, Fx, num_pop, U_obj_last):
    Fxf = np.array([])
    for i in range(num_pop):
        if pen[i] == 0:
            Fxf = np.append(Fxf, Fx[i])
    if Fxf.size == 0:
        U_obj = U_obj_last
    else:
        U_obj = min(Fxf)
    return U_obj
def obj_max(pen, Fx, num_pop, U_obj_last):
    Fxf = np.array([])
    for i in range(num_pop):
        if pen[i] == 0:
            Fxf = np.append(Fxf, Fx[i])
    if Fxf.size == 0:
        U_obj = U_obj_last
    else:
        U_obj = max(Fxf)

```

```

return U_obj
A.12 Создание модели ЛА
Model_Aircraft.FCMacro
import PartDesignGui
from FreeCAD import Base
import Part
import importXLSX
import importAirfoilDAT
import os
import math
import FreeCAD
import numpy as np
import Draft
import re
#Create new document
FreeCAD.newDocument()
FreeCAD.getDocument('Unnamed').Label = "Airplane"
#Open a file excel
user_path = os.path.expanduser('~')
file_excel = os.path.join(user_path, ACII_KII_0/gen/info_to_FreeCAD.xlsx')
importXLSX.insert(file_excel,"Unnamed")
sheet = FreeCAD.ActiveDocument.getObject('Sheet1')
cell_regex = re.compile('^[A-Z]+[0-9]+$')
get_cells = lambda sheet: filter(cell_regex.search, sheet.PropertiesList)
i = 1
for cell in get_cells(sheet):
    i+=1
num_row = int(i/15)
FreeCAD.getDocument('Unnamed').getObject('Sheet1').Label = "Parameters"
iter = 2
for i in range(0, num_row-1):
    index = str(iter)
    #Assign parameters for f_wing
    f_delta1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('B'+index)
    f_delta = f_delta1
    f_root_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('C'+index)
    f_root_chord = f_root_chord1*1000
    f_tip_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('D'+index)
    f_tip_chord = f_tip_chord1*1000
    f_x_loc_root_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('E'+index)
    f_x_loc_root_chord = f_x_loc_root_chord1*1000
    f_y_loc_root_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('F'+index)
    f_y_loc_root_chord = f_y_loc_root_chord1*1000
    f_x_loc_tip_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('G'+index)
    f_x_loc_tip_chord = f_x_loc_tip_chord1*1000
    f_y_loc_tip_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('H'+index)
    f_y_loc_tip_chord = f_y_loc_tip_chord1*1000
    #Assign parameters for a_wing
    a_delta1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('I'+index)
    a_delta = a_delta1
    a_root_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('J'+index)
    a_root_chord = a_root_chord1*1000

```

```

a_tip_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('K'+index)
a_tip_chord = a_tip_chord1*1000
a_x_loc_root_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('L'+index)
a_x_loc_root_chord = a_x_loc_root_chord1*1000
a_y_loc_root_chord1 =
FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('M'+index)
a_y_loc_root_chord = a_y_loc_root_chord1*1000
a_x_loc_tip_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('N'+index)
a_x_loc_tip_chord = a_x_loc_tip_chord1*1000
a_y_loc_tip_chord1 = FreeCAD.getDocument('Unnamed').getObject('Sheet1').get('O'+index)
a_y_loc_tip_chord = a_y_loc_tip_chord1*1000
#For f_wing
#Open a file airfoil
airfoil_file = os.path.join(user_path,'ACII_KII_0/tmp/airfoil_2212.dat')
f = open(airfoil_file, 'r')
coords=[]
for line in f:
    lstData = line.split(' ')
    x = float(lstData[0])
    y = 0
    z = float(lstData[1])
    coords.append(FreeCAD.Vector(x,y,z))
#Create profile
Draft.makeBSpline(coords, closed=True)
FreeCAD.ActiveDocument.recompute()
#Change name and scale f_root_chord
FreeCAD.getDocument('Unnamed').getObject('BSpline').Label = "f_root_foil"
Draft.scale([FreeCAD.ActiveDocument.BSpline], scale=FreeCAD.Vector(f_root_chord,
f_root_chord, f_root_chord), center=FreeCAD.Vector(0.0, 0.0, 0.0), copy=True)
FreeCAD.ActiveDocument.recompute()
#Change name and scale f_tip_chord
FreeCAD.getDocument('Unnamed').getObject('BSpline').Label = "f_tip_foil"
Draft.scale([FreeCAD.ActiveDocument.BSpline], scale=FreeCAD.Vector(f_tip_chord,
f_tip_chord, f_tip_chord), center=FreeCAD.Vector(0.0, 0.0, 0.0), copy=False)
FreeCAD.ActiveDocument.recompute()
#Move and rotate f_root_chord
App.getDocument("Unnamed").BSpline001.Placement=App.Placement(App.Vector(f_x_loc_
root_chord,f_y_loc_root_chord,0), App.Rotation(App.Vector(0,1,0),f_delta), App.Vector(0,0,0))
#Move and rotate f_tip_chord
App.getDocument("Unnamed").BSpline.Placement=App.Placement(App.Vector(f_x_loc_tip_
chord,f_y_loc_tip_chord,0), App.Rotation(App.Vector(0,1,0),f_delta), App.Vector(0,0,0))
App.getDocument('Unnamed').recompute()
#Loft a half of f_wing
App.getDocument('Unnamed').addObject('Part::Loft','Loft')
App.getDocument('Unnamed').ActiveObject.Sections=[App.getDocument('Unnamed').BSplin
e001, App.getDocument('Unnamed').BSpline, ]
App.getDocument('Unnamed').ActiveObject.Solid=True
App.getDocument('Unnamed').ActiveObject.Ruled=True
App.getDocument('Unnamed').ActiveObject.Closed=True
App.getDocument('Unnamed').recompute()
#Mirror to create full of f_wing
__doc__=FreeCAD.getDocument("Unnamed")

```

```

__doc__.addObject("Part::Mirroring")
__doc__.ActiveObject.Source=__doc__.getObject("Loft")
__doc__.ActiveObject.Label=u"Loft (Mirror #1)"
__doc__.ActiveObject.Normal=(0,1,0)
__doc__.ActiveObject.Base=(0,0,0)
del __doc__
FreeCAD.getDocument('Unnamed').getObject('Part__Mirroring').Label = "forward_wing"
App.getDocument('Unnamed').recompute()
#For a_wing
Draft.makeBSpline(coords, closed=True)
FreeCAD.ActiveDocument.recompute()
#Change name and scale a_root_chord
FreeCAD.getDocument('Unnamed').getObject('BSpline002').Label = "a_root_foil"
Draft.scale([FreeCAD.ActiveDocument.BSpline002], scale=FreeCAD.Vector(a_root_chord,
a_root_chord, a_root_chord), center=FreeCAD.Vector(0.0, 0.0, 0.0), copy=True)
FreeCAD.ActiveDocument.recompute()
#Change name and scale a_tip_chord
FreeCAD.getDocument('Unnamed').getObject('BSpline002').Label = "a_tip_foil"
Draft.scale([FreeCAD.ActiveDocument.BSpline002], scale=FreeCAD.Vector(a_tip_chord,
a_tip_chord, a_tip_chord), center=FreeCAD.Vector(0.0, 0.0, 0.0), copy=False)
FreeCAD.ActiveDocument.recompute()
#Move and rotate a_root_chord
App.getDocument("Unnamed").BSpline003.Placement=App.Placement(App.Vector(a_x_loc
_root_chord,a_y_loc_root_chord,0), App.Rotation(App.Vector(0,1,0),a_delta),
App.Vector(0,0,0))
#Move and rotate a_tip_chord
App.getDocument("Unnamed").BSpline002.Placement=App.Placement(App.Vector(a_x_loc
_tip_chord,a_y_loc_tip_chord,0), App.Rotation(App.Vector(0,1,0),a_delta), App.Vector(0,0,0))
App.getDocument('Unnamed').recompute()
#Loft a half of a_wing
App.getDocument('Unnamed').addObject('Part::Loft','Loft')
App.getDocument('Unnamed').ActiveObject.Sections=[App.getDocument('Unnamed').BSplin
e003, App.getDocument('Unnamed').BSpline002, ]
App.getDocument('Unnamed').ActiveObject.Solid=True
App.getDocument('Unnamed').ActiveObject.Ruled=True
App.getDocument('Unnamed').ActiveObject.Closed=True
App.getDocument('Unnamed').recompute()
#Mirror to create full of f_wing
__doc__=FreeCAD.getDocument("Unnamed")
__doc__.addObject("Part::Mirroring")
__doc__.ActiveObject.Source=__doc__.getObject("Loft001")
__doc__.ActiveObject.Label=u"Loft001 (Mirror #2)"
__doc__.ActiveObject.Normal=(0,1,0)
__doc__.ActiveObject.Base=(0,0,0)
del __doc__
FreeCAD.getDocument('Unnamed').getObject('Part__Mirroring001').Label = "aftward_wing"
App.getDocument('Unnamed').recompute()
Gui.SendMsgToActiveView("ViewFit")
Gui.activeDocument().activeView().viewTop()
png_file = os.path.join(user_path,f'U_40_0/tmp/{i}.png')
Gui.activeDocument().activeView().saveImage(png_file,4000,2600,'Current')
App.getDocument('Unnamed').removeObject('Part__Mirroring')

```

```
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('Part__Mirroring001')
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('Loft')
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('Loft001')
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('BSpline')
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('BSpline001')
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('BSpline002')
App.getDocument('Unnamed').recompute()
App.getDocument('Unnamed').removeObject('BSpline003')
App.getDocument('Unnamed').recompute()
iter+=1
mw = FreeCADGui.getMainWindow()
mw.deleteLater()
```