

# МЕТОДЫ ПОСТРОЕНИЯ ОПТИМАЛЬНЫХ ОЧЕРЕДЕЙ ВОЗДУШНЫХ СУДОВ НА ПОСАДКУ<sup>1</sup>.

## Ч. 1. Методы точного решения

Г.С. Вересников, Н.А. Егоров, Е.Л. Кулида, В.Г. Лебедев

Отмечено, что задача формирования оптимальных очередей воздушных судов на посадку одна из основных задач при планировании и регулировании потоков воздушного движения, для решения которой требуется использование достаточно серьезного математического аппарата. Представлен краткий обзор основных подходов к постановке и решению этой задачи.

**Ключевые слова:** воздушное движение, критерий оптимизации, оптимальная очередь на посадку, линейное программирование, метод ветвей и границ.

### ВВЕДЕНИЕ

Системы планирования и регулирования потоков воздушного движения (ВД) предназначены для обеспечения безопасного, упорядоченного и ускоренного ВД, в том числе в районах аэропортов и аэроузлов, уменьшения издержек авиакомпаний и пассажиров, уменьшения негативного влияния на окружающую среду с учетом неопределенности прогнозирования ВД.

Сложность планирования и регулирования потоков ВД в районе аэродрома и загрузка диспетчеров возрастают с увеличением интенсивности ВД. При этом падает эффективность управления, возрастают задержки воздушных судов (ВС), увеличивается расход топлива. Для предотвращения подобных ситуаций разрабатываются специальные средства поддержки работы диспетчера при планировании и регулировании потоков ВД, которые выполняют часть функций диспетчера и позволяют более эффективно использовать пропускную способность взлетно-посадочной полосы (ВПП) и воздушного пространства. В Европе такие средства получили название Arrival Manager (AMAN) и Departure Manager (DMAN) — менеджеры прилета и вылета.

<sup>1</sup> Работа выполнена при частичной финансовой поддержке РФФИ (проект № 18-08-00822) и Программы I.30 Президиума РАН.

Системы AMAN и их аналоги применяются в Европе и США более 25 лет, постоянно обновляясь и совершенствуясь. Современные системы активно внедряются по всему миру, а программы развития системы организации ВД SESAR и NextGen ведут обширные проекты по дальнейшему их совершенствованию.

В Российской Федерации есть несколько разработчиков систем планирования и регулирования прибывающих потоков ВД, относительно недавно начавших работы в данном направлении. Большинство из отечественных систем AMAN находятся на этапе разработки и тестирования, их внедрение пока только планируется. Однако в рамках модернизации Единой системы организации ВД Российской Федерации ведутся исследования ведущими научно-исследовательскими институтами и производителями систем автоматизации управления ВД, среди которых ГосНИИ гражданской авиации и филиал «Аэронавигация» [1], ГосНИИАС [2], Московский государственный технический университет гражданской авиации [3], Государственный университет гражданской авиации в Санкт-Петербурге [4], Московский авиационный институт [5], Ульяновское высшее авиационное училище [6, 7] и др.

«Построение оптимальных очередей на прилет является одной из основных задач при планировании и регулировании потоков ВД в районе аэродрома и аэроузла, для решения которой требуется



использование достаточно серьезного математического аппарата.

Самое простое решение такой задачи, по сути, не требующее математических вычислений, — это стратегия «первый пришел — первый обслужился» (first come — first served (FCFS)). При данном подходе не требуются перестановки в естественной очереди, и, казалось бы, посадка в первую очередь раньше прибывших ВС должна обеспечить минимальные задержки. Но разные летно-технические характеристики ВС, особенности структуры STAR (Standard Terminal Arrival Route) и интенсивное воздушное движение делают такой подход несостоятельным, в то время как некоторые перестановки в естественной последовательности (например, перестановка медленнее ВС за быстрым) могут существенно улучшить результат. Поэтому в большинстве случаев при формировании оптимальной очереди на посадку приходится отступать от стратегии FCFS и каким-либо образом выбирать новый порядок в очереди, применяя при этом более сложные алгоритмы» [8].

Задача формирования оптимальной очереди на посадку (ФООП) стоит на повестке дня уже несколько десятилетий, но до сих пор не утратила своей актуальности. Первые публикации, посвященные задаче ФООП, появились в 1970-х гг. Однако сложность проблемы, новые требования к организации ВД, новые методы решения оптимизационных задач делают задачу актуальной и в настоящее время, о чем свидетельствуют многочисленные новые публикации на эту тему. В силу сложности и важности ФООП все еще остается одной из основных проблем в исследовании проблем управления потоками ВД [9].

В литературе рассматривается большое число различных постановок задачи ФООП [10]. Различаются статическая задача [11], которая решается для фиксированного набора ВС, и динамическая задача [12] для переменного набора ВС, когда часть ВС появляется в зоне регулирования, а другая их часть эту зону покидает. Различаются постановки задачи для посадки на одну или несколько ВПП. Рассматриваются различные критерии оптимизации, среди которых можно выделить два основных. Один из них применяется, когда плотность воздушного движения высока и необходимо эффективно использовать пропускную способность ВПП. Другой критерий применяется для того, чтобы все ВС приземлились как можно ближе к своему оптимальному времени посадки. Он обеспечивает наиболее эффективное и безопасное ВД, уменьшение издержек авиакомпаний и пассажиров, уменьшение негативного влияния на окружающую среду и может применяться при разумной плотности трафика.

В литературе представлены точные и приближенные методы решения задачи ФООП, Основными из них — это методы линейного программирования, метод ветвей и границ, эвристические и метаэвристические методы.

Точные алгоритмы основаны на анализе всех возможных перестановок ВС в очереди посадок, поэтому время работы экспоненциально растет с размером проблемы (числом ВС). Поэтому в точных алгоритмах решения задачи применяются различные приемы для сокращения перебора. Тем не менее, не удается получить оптимальное решение задач большой размерности в течение приемлемого времени. В литературе представлены примеры оптимальных решений задачи ФООП не более чем для 50 ВС.

Для получения приближенных решений задачи ФООП в литературе представлено большое число эвристических, метаэвристических, гибридных и прочих алгоритмов, в том числе с приведенными вычислительными результатами для 500 ВС и 5 ВПП, полученными за приемлемое время. Особенно успешными для приближенного решения задачи ФООП оказались меметические алгоритмы.

Обзор состоит из двух частей.

В первой части подробно рассматривается представленная в работе [11] классическая постановка статической задачи ФООП на основе линейного целочисленного программирования с булевыми переменными, которая лежит в основе множества других работ. Приводится обзор различных модификаций этой постановки.

Приводится краткий обзор в историческом разрезе методов решения задачи ФООП, начиная с самых ранних работ. Затем подробно рассматривается подход к решению поставленной задачи линейного программирования с помощью линейной релаксации [11] и решение задачи на основе метода ветвей и границ, дополненного методом, называемым «генерацией столбцов» [13].

Вторая часть обзора посвящена предлагаемому в мировой научной литературе перспективным алгоритмам оптимизации для приближенного решения задачи построения оптимальных очередей посадок ВС на одну или несколько ВПП.

---

### **1. ПОСТАНОВКА СТАТИЧЕСКОЙ ЗАДАЧИ ФОРМИРОВАНИЯ ОПТИМАЛЬНОЙ ОЧЕРЕДИ ВОЗДУШНЫХ СУДОВ НА ПОСАДКУ**

---

В настоящее время классической постановкой для задачи формирования оптимальной очереди на посадку, которая лежит в основе множества других работ, считается постановка, представленная в работе [11]. В ее основе лежит задача линейного це-

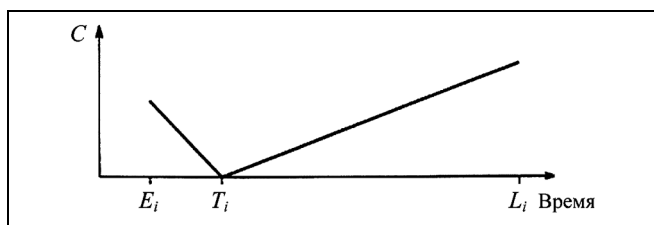


Рис. 1. Изменение затрат внутри временного окна посадки воздушных судов

лочисленного программирования с булевыми переменными.

### 1.1. Основные обозначения

Введем обозначения:  $P$  — число ВС, ожидающих посадки;  $E_i$  — самое раннее возможное время прибытия  $i$ -го ВС,  $i = 1, \dots, P$ ;  $L_i$  — самое позднее возможное время прибытия  $i$ -го ВС,  $i = 1, \dots, P$ ;  $T_i$  — оптимальное время прибытия  $i$ -го ВС,  $i = 1, \dots, P$ .

На рис. 1 представлена функция изменения затрат  $C$  в зависимости от времени посадки ВС в течение временного окна. Она нелинейная, однако, она состоит из двух линейных участков, что позволяет сформулировать задачу с линейной целевой функцией. Далее обозначим:  $S_{ij}$  — минимальный интервал между посадкой ВС  $j$  за ВС  $i$ ,  $j, i = 1, \dots, P$ ;  $g_i$  — штраф за единицу времени приземления  $i$ -го ВС,  $i = 1, \dots, P$ , до времени  $T_i$ ;  $h_i$  — штраф за единицу времени приземления  $i$ -го ВС,  $i = 1, \dots, P$ , после времени  $T_i$ .

Переменные:  $x_i$  — время приземления  $i$ -го ВС,  $i = 1, \dots, P$ ;  $\alpha_i$  — на сколько  $i$ -е ВС,  $i = 1, \dots, P$ , приземляется раньше времени  $T_i$ ;  $\beta_i$  — на сколько  $i$ -е ВС,  $i = 1, \dots, P$ , приземляется позднее времени  $T_i$ ;

$$\delta_{ij} = \begin{cases} 1, & \text{если ВС } i \text{ приземляется раньше ВС } j \\ i, j = 1, \dots, P, \\ 0 & \text{в противном случае.} \end{cases}$$

### 1.2. Ограничения, которым должно удовлетворять решение задачи

1. Практически во всех работах на время приземления налагаются ограничения, обусловленные летно-техническими характеристиками ВС и параметрами зоны приземления: ограничения на ускорения, конечные размеры зоны приземления, ограничения по топливу и др. Все эти факторы определяют фиксированный интервал допустимого времени приземления: ВС с номером  $i$  должно приземляться внутри временного окна  $[E_i, L_i]$ :  $E_i \leq x_i \leq L_i$ ,  $i = 1, \dots, P$ , [11, 14–18].

2. Либо ВС  $i$  приземляется раньше ВС  $j$  ( $\delta_{ij} = 1$ ), либо ВС  $j$  приземляется раньше ВС  $i$  ( $\delta_{ji} = 1$ ):  $\delta_{ij} + \delta_{ji} = 1$ ,  $i, j = 1, \dots, P$ .

На основе информации о временных окнах все пары  $(i, j)$ ,  $(i, j = 1, \dots, P)$  можно разделить на множества:  $U$  — множество пар  $(i, j)$ , для которых на основе анализа временных окон и ограничений задержки между последовательными посадками ВС (ограничения разделения) нельзя установить, какое ВС приземляется раньше;  $V$  — множество пар  $(i, j)$ , в которых ВС  $i$  должно приземлиться раньше ВС  $j$ , но при этом ограничения разделения не выполняются автоматически;  $W$  — множество пар  $(i, j)$ , в которых ВС  $i$  должно приземлиться раньше ВС  $j$ , при этом ограничения разделения автоматически выполняются:  $\delta_{ij} = 1 \forall (i, j) \in W \cup V$ .

3. Одно из наиболее распространенных ограничений в большинстве задач ФООП состоит в минимальной допустимой из соображений безопасности задержке между последовательной посадкой двух ВС [11, 14–16].

По длительности этой задержки ВС разделяются на ряд типов. Пример стандартной минимальной допустимой задержки между последовательными приземлениями ВС, приведенный в работе [14], представлен в табл. 1.

Эти ограничения разделения можно представить в виде:

$$x_j \geq x_i + S_{ij} \quad \forall (i, j) \in V,$$

$$x_j \geq x_i + S_{ij}\delta_{ij} - (L_i - E_j)\delta_{ji} \quad \forall (i, j) \in U.$$

Возможны два случая:

— если  $\delta_{ij} = 1$ , тогда из ограничения 2 следует  $\delta_{ji} = 0$ , и ограничение принимает вид:  $x_j \geq x_i + S_{ij}$  для всех пар ВС, в которых ВС  $i$  приземляется раньше ВС  $j$ ;

— если  $\delta_{ij} = 0$ , тогда из ограничения 2 следует  $\delta_{ji} = 1$ , и ограничение принимает вид:  $x_j \geq x_i + (L_i - E_j)$  или  $x_j - E_j \geq x_i - L_i$ , которое всегда выполняется, потому что левая часть больше или рав-

Таблица 1

Минимальная допустимая задержка между приземлениями, с

		Тип следующего приземляющегося ВС			
		1	2	3	4
Тип предыдущего приземляющегося ВС	1	96	200	181	228
	2	72	80	70	110
	3	72	100	70	130
	4	72	80	70	90

**Примечание.** 1 — Boeing 747, 2 — Boeing 727, 3 — Boeing 707, 4 — McDonnellDouglasDC 9.



на 0, а правая часть меньше или равна 0. Таким образом, в этом случае ограничение эффективно деактивируется.

4. Ограничения для связи переменных  $\alpha_i, \beta_i, x_i$ :

$$\begin{aligned} \alpha_i &\geq T_i - x_i, \quad i = 1, \dots, P; \\ 0 &\leq \alpha_i \leq T_i - E_i, \quad i = 1, \dots, P; \\ \beta_i &\geq x_i - T_i, \quad i = 1, \dots, P; \\ 0 &\leq \beta_i \leq L_i - T_i, \quad i = 1, \dots, P; \\ x_i &= T_i - \alpha_i + \beta_i, \quad i = 1, \dots, P. \end{aligned}$$

Введение дополнительных ограничений в модель позволяет учитывать ряд практически важных моментов, например, ограничения приоритета, балансировка рабочей нагрузки на ВПП, ограничение на число посадок в течение определенного периода времени. Последнее ограничение, при числе посадок, равном 0, позволяет обеспечить временные интервалы, в течение которых ВПП недоступна для посадок.

Предположим, например, что должно быть не более  $Q$  посадок за период времени.

Пусть  $G = \{i | [t_1, t_2] \cap [E_i, L_i] \neq \emptyset, i = 1, \dots, P\}$ , т. е.  $G$  — это множество ВС, которые могут приземлиться в течение периода времени  $[t_1, t_2]$ . Для простоты предположим, что все времена целочисленные. Воздушное судно  $i$  должно приземлиться в один из трех временных отрезков:  $[E_i, \max(E_i, t_1) - 1]$ ,  $[\max(E_i, t_1), \min(L_i, t_2)]$ ,  $[\min(L_i, t_2) + 1, L_i]$ . Заметим, что некоторые из этих временных отрезков могут не существовать.

Пусть

$$\gamma_{ij} = \begin{cases} 1, & \text{если ВС } i \text{ приземляется во время} \\ & \text{временного отрезка } j; \\ 0 & \text{в противном случае.} \end{cases}$$

Для несуществующих отрезков значение  $\gamma_{ij}$  устанавливается равным нулю.

После этого нужно добавить к модели уравнение

$$\sum_{i \in G} \gamma_{i2} = Q,$$

которое ограничивает число посадок за период времени  $[t_1, t_2]$ .

$$\gamma_{i1} + \gamma_{i2} + \gamma_{i3} = 1 \quad \forall i \in G.$$

Это уравнение гарантирует посадку ВС  $i$  только в один из трех временных отрезков:

$$x_i \leq [\max(E_i, t_1) - 1] \gamma_{i1} + L_i(1 - \gamma_{i1}) \quad \forall i \in G;$$

$$\max(E_i, t_1) \gamma_{i2} \leq x_i \leq \min(L_i, t_2) \gamma_{i2} + L_i(1 - \gamma_{i2}) \quad \forall i \in G;$$

$$x_i \leq [\min(L_i, t_2) + 1] \gamma_{i3} \quad \forall i \in G.$$

Последние три уравнения соотносят время посадки ВС  $i$  с соответствующим временным отрезком.

Отмечается, что эти ограничения можно применить к любому подмножеству ВС (например, можно ограничить число самолетов от определенной авиакомпании, которые приземляются в определенный период времени).

Условие  $\sum_{i \in G} \gamma_{i2} = 0$  гарантирует отсутствие посадок в период времени  $[t_1, t_2]$ , и поэтому приведенные ограничения позволяют реализовать закрытие ВПП в указанный период времени [19].

На времена посадки могут налагаться и другие ограничения, например, недопустимость опережения ВС планового времени приземления. Подобные ограничения рассматриваются, в частности, в работах [9, 20–23].

В ряде работ делается попытка учета того, что ВС идут с нескольких различных маршрутов и происходит их слияние в некоторый единый маршрут. При этом порядок приземления ВС из одного маршрута должен совпадать с исходным порядком ВС в этом маршруте. Последующие ВС маршрута не допускаются к посадке до впереди идущих ВС. Сторонники подобного подхода обосновывают его вопросами полетной безопасности, а также принципами справедливости [24].

### 1.3. Постановка статической задачи формирования оптимальной очереди воздушных судов на посадку на несколько взлетно-посадочных полос

В большинстве оживленных международных аэропортов несколько ВПП. Постановка задачи формирования оптимальной очереди на посадку может быть расширена для нескольких ВПП введением дополнительных переменных и ограничений. Рассмотрим их.

Число ВПП равно  $R$ ,

$$z_{ij} = \begin{cases} 1, & \text{если ВС } i \text{ и ВС } j \text{ приземляются} \\ & \text{на одну ВПП,} \\ 0 & \text{в противном случае, } i, j = 1, \dots, P, i \neq j, \end{cases}$$

$$y_{jr} = \begin{cases} 1, & \text{если ВС } j \text{ приземляется на ВПП } r, \\ 0 & \text{в противном случае,} \\ j = 1, \dots, P, \quad r = 1, \dots, R, \end{cases}$$

$$\sum_{r=1}^R y_{ir} = 1, \quad i = 1, \dots, P.$$

Это уравнение означает, что каждое ВС приземляется точно на одну ВПП.

$$z_{ij} = z_{ji}, \quad i, j = 1, \dots, P; \quad j > i.$$

Это симметричное ограничение (если ВС  $i$  и ВС  $j$  приземляются на одну ВПП, то это верно и для ВС  $j$  и ВС  $i$ ). Далее

$$z_{ij} \geq y_{ir} + y_{jr} - 1, \quad i, j = 1, \dots, P; \quad j > i; \\ r = 1, \dots, R.$$

Последнее уравнение гарантирует, что, если ВС  $i$  и ВС  $j$  приземляются на ВПП  $r$ , то  $z_{ij} = 1$ . Если  $z_{ij} = 0$ , то уравнение гарантирует, что ВС  $i$  и ВС  $j$  не могут приземлиться на одну ВПП.

Кроме того, должны соблюдаться ограничения, о которых говорилось в п. 1.2. В некоторых аэропортах ВПП могут располагаться таким образом, что необходимы дополнительные ограничения разделения между ВС, приземляющимися на разные ВПП. В этом случае вводятся дополнительные переменные  $s_{ij}$  — минимальный интервал между посадкой ВС  $j$  за ВС  $i$  на разные ВПП,  $0 \leq s_{ij} \leq S_{ij}$ ,  $i, j = 1, \dots, P; i \neq j$ . Ограничения разделения в этом случае можно представить в виде:

$$x_j \geq x_i + S_{ij}z_{ij} + s_{ij}(1 - z_{ij}) \quad \forall (i, j) \in V, \\ x_j \geq x_i + S_{ij}\delta_{ij} + s_{ij}(1 - z_{ij}) - (L_i + S_{ij} - E_j)\delta_{ji} \\ \forall (i, j) \in U.$$

#### 1.4. Критерии оптимизации

В настоящее время ведутся многочисленные исследования задач ФООП для различных целей оптимизации. В частности, модели ФООП могут разрабатываться с целью либо минимизации общей задержки в воздухе (общего времени задержки всех прибывающих ВС) [20, 22], либо дополнительных затрат (издержек, вызванных ранней или поздней посадкой) [11, 14, 15], либо времени обработки процесса приземления (времени, необходимого для всех посадок) [20, 21] и др.

В работе [14] проведен анализ некоторых наиболее распространенных целевых функций, используемых в различных моделях ФООП. Среди них отметим:

$$\min \sum_{i=1}^P [\max(x_i - T_i, 0)]$$

— минимум суммарной задержки всех ВС в условиях запрета опережения планового времени приземления, где  $x_i$  и  $T_i$  — фактическое и плановое времена приземления  $i$ -го ВС соответственно.

Большинство исследователей полагают, однако, что условие запрета опережения планового времени приземления является искусственным требованием, и рассматривают в силу этого более общую функцию минимизации суммарных затрат всех ВС.

В работе [11] приводятся два основных критерия оптимизации. Первый из них служит для ми-

нимизации стоимости отклонения от целевого времени посадки:

$$\min \sum_{i=1}^P (g_i\alpha_i + h_i\beta_i),$$

т. е. минимум суммарных затрат всех ВС, вызванных отклонениями фактического времени посадки от планового, где  $g_i$  и  $h_i$  — штрафы  $i$ -го ВС за опережение и задержку фактического времени приземления относительно планового. Этот критерий обеспечивает наиболее эффективное и безопасное ВД, уменьшение издержек авиакомпаний и пассажиров, уменьшение негативного влияния на окружающую среду и может применяться при разумной плотности трафика.

Второй критерий оптимизации предназначен для максимизации пропускной способности ВПП. Такая цель возникает при высокой плотности ВД и необходимости эффективного использования пропускной способности ВПП.

Для достижения этой цели рассматривается целевая функция

$$\min \max[x_i | i = 1, \dots, P],$$

т. е. последнее ВС должно приземлиться как можно быстрее. Эта целевая функция легко линейризуется введением дополнительных переменных.

Рассматриваются и некоторые другие виды целевых функций:

$$\min(\max[x_1, \dots, x_p] - \min[x_1, \dots, x_p])$$

— минимум времени приземления всей группы ВС и др.

## 2. МЕТОДЫ ПОСТРОЕНИЯ ОПТИМАЛЬНЫХ ОЧЕРЕДЕЙ ПОСАДОК ВОЗДУШНЫХ СУДОВ НА ОДНУ ИЛИ НЕСКОЛЬКО ВЗЛЕТНО-ПОСАДОЧНЫХ ПОЛОС

Приведем краткий обзор в историческом разрезе исследования задачи ФООП, начиная с ранних работ [25], в которых исследована сильно упрощенная версия задачи ФООП. В этих работах предполагалось отсутствие задержек между последовательными приземлениями ВС.

В работе [26] представлена дискретно-событийная имитационная модель для оценки различных стратегий упорядочения ВС. Приведены вычислительные результаты по ряду моделируемых сценариев.

В работах [27, 28] обсуждаются статическая и динамическая задачи посадки ВС и представлен эвристический алгоритм ФООП на одну ВПП, основанный на технике, называемой скоординированным сдвигом позиции. Он включает в себя поиск для небольшого набора ВС наилучших возможных позиций для них в очереди посадки с





ограничением сдвига, так что ВС не может быть перемещено больше, чем на заданное число позиций от положения, которое оно имело бы при посадке на основе FCFS [29]. Представлены вычислительные результаты для трех моделируемых сценариев с участием 500 ВС. Ограничение разделения распространяется только на последовательные посадки ВС. Другими словами, если ВС  $i$ ,  $j$  и  $k$  приземляются один за другим так, что ВС  $i$  приземляется перед ВС  $j$ , который приземляется перед ВС  $k$ , то неравенства

$$x_j \geq x_i + S_{ij} \text{ и } x_k \geq x_j + S_{jk}$$

гарантированно выполняются, но может быть, что  $x_k < x_i + S_{ik}$ .

Такая ситуация называется последовательным разделением. Другой случай — когда разделение гарантируется между всеми парами ВС (такая ситуация называется полным разделением). Отметим, что, если неравенство треугольника выполняется:

$$S_{ik} \leq S_{ij} + S_{jk} (\forall i, \forall k \neq i, \forall j \neq i, k),$$

то последовательное разделение достаточно для обеспечения полного разделения.

Применяется также ограничение сдвига в статической задаче ФООП на одну ВПП [25]. Рассмотрена задача посадки группы одинаковых ВС. Вычислительные результаты представлены для трех групп ВС (всего 15 ВС). В дальнейшем рассматривались постановки, где все ВС (потенциально) разные.

Алгоритм поиска по дереву в глубину, основанный на перечислении всех возможных последовательностей ВС представлен в работе [30]. Ветви дерева отбрасываются, когда стоимость частично построенной последовательности превышает известное возможное решение. По вычислительным причинам предложено применять переборный алгоритм в эвристическом варианте со скользящим окном. Последовательность ВС упорядочивается, подпоследовательность ВС длины  $F$  (длина скользящего окна) размораживается, и для нее решается подзадача, после этого первое ВС из подпоследовательности замораживается, затем процедура повторяется.

В работе [31] представлена целочисленная постановка с булевыми переменными задачи ФООП на одну ВПП и эвристика на основе генетического алгоритма. Приведены вычислительные результаты по ряду задач, включающих до 20 ВС.

В работе [32] на основе теории массового обслуживания рассмотрена задача назначения приоритетов для ВС, ожидающих захода на посадку.

Специализированный симплекс-алгоритм, который с высокой вычислительной производи-

тельностью оценивает время посадки, основываясь на некоторой частичной информации, представлен в работе [33]. Используется предварительная обработка, предусматривающая сжатие временных окон и частичное упорядочение на основе данных задачи или верхней границы решения. Алгоритм тестируется на данных из OR-библиотеки [34], включающих до 50 ВС и 4 ВПП.

В дальнейших исследованиях учитывалась окружающая обстановка и допускалось определенное ограниченное изменение позиции приземления в изначальной последовательности ВС. Предлагался алгоритм оптимизации последовательности приземления ВС, основанный на методах динамического программирования.

Задача ФООП также рассматривалась как задача календарного планирования работ в цеху. Для ее решения были предприняты значительные усилия по применению математического программирования [18, 35–37]. В статье [38] представлен подход со скользящим горизонтом, который разбивает последовательность ВС на части и решает задачу ФООП отдельно для каждой из этих частей. Предложены способы разбиения заданной последовательности ВС и исследовано их влияние на качество решения.

В результате анализа существующих методов к наиболее гибким к постановке задачи и критериям оптимизации и дающим хорошие результаты следует отнести метод ветвей и границ, в котором для вычисления оценок применяется метод линейного программирования [11, 13]. Метод ветвей и границ сочетает в себе преимущества полного перебора, при этом, на основе различных техник сокращения перебора, позволяет отбрасывать большие ветви вычислений. Далее этот подход рассматривается более подробно.

### 2.1. Линейная релаксация задачи

Представленная выше постановка показывает, что целочисленное программирование с булевыми переменными достаточно гибко для того, чтобы учесть большое число ограничений, например, допустимое окно времени посадки и минимальное допустимое время между последовательными приземлениями. Для численного решения в работе [11] предлагается линейная релаксация постановки задачи путем замены булевых переменных непрерывными переменными, т. е. вместо булевых переменных, принимающих значения 0 и 1, рассматриваются непрерывные переменные в диапазоне  $[0, 1]$ . Проблема в том, что в этом случае решение также может оказаться нецелочисленным. Для улучшения решения в непрерывном пространстве вводятся дополнительные ограничения, которые в целочисленном случае избыточны.

### 2.1.1. Дополнительные ограничения

- $\delta_{ij} \geq (x_j - x_i)/(L_j - E_i)$

Если  $(x_j - x_i) > 0$ , то ВС  $j$  приземляется после ВС  $i$  и, следовательно, необходимо обеспечить условие  $\delta_{ij} = 1$ . В приведенном ограничении член  $(L_j - E_i)$  является коэффициентом масштабирования и представляет максимальное значение, которое может принять разность  $(x_j - x_i)$ .

- Рассмотрим переменные  $\delta_{ij}$ . Первое ВС приземляется перед  $(P - 1)$  другими ВС, второе ВС приземляется перед  $(P - 2)$  другими ВС и т. д. Таким образом, сумма всех переменных  $\delta_{ij}$  должна быть равна  $(P - 1) + (P - 2) + \dots + 1$ , т. е.  $P(P - 1)/2$ .

Следовательно,

$$\sum_{i=1}^P \sum_{j=1, j \neq i}^P \delta_{ij} = P(P - 1)/2.$$

- Предположим, ВС  $j$  приземляется после ВС  $i$ , и  $T_i < T_j$ . Тогда промежуток между целевыми временами посадки  $(T_j - T_i)$  может быть закрыт суммой отклонений от оптимального времени посадки  $(\beta_i + \alpha_j)$  (рис. 2). Следовательно, мы имеем, что

$$\delta_{ij} \geq 1 - \frac{\beta_i + \alpha_j}{T_j - T_i}$$

поскольку, если  $(\beta_i + \alpha_j)$  недостаточно для перекрытия промежутка, то ВС  $i$  должно приземлиться раньше ВС  $j$ .

- Предположим, что есть два ВС, таких что  $T_i < T_j$  и  $(T_j - T_i) < S_{ij}$ , т. е. ВС  $i$  предпочтительнее приземлиться перед ВС  $j$ , но их оптимальные времена прибытия не удовлетворяют ограничению разделения.

Как видно из рис. 2, если ВС  $i$  приземляется перед ВС  $j$  ( $\delta_{ij} = 1$ ) на одну ВПП, то отклонение должно быть не меньше, чем  $S_{ij} - (T_j - T_i)$ , а если ВС  $j$  приземляется перед ВС  $i$  ( $\delta_{ji} = 1$ ), то отклонение должно быть не меньше, чем  $(T_j - T_i) + S_{ji}$ . Следовательно,

$$\begin{aligned} (\alpha_i + \beta_j) + (\alpha_j + \beta_i) &\geq [S_{ij} - (T_j - T_i)]\delta_{ij} + \\ &+ [(T_j - T_i) + S_{ji}]\delta_{ji} - \max\{[S_{ij} - (T_j - T_i)], \\ &[(T_j - T_i) + S_{ji}]\}(1 - z_{ij}) \quad \forall (i, j) \in U, \\ &\text{если } T_i < T_j \text{ \& } (T_j - T_i) < S_{ij}. \end{aligned}$$

Слагаемое, содержащее  $(1 - z_{ij})$ , становится неактивным, если ВС  $i$  и ВС  $j$  приземляются на одну ВПП и  $z_{ij} = 1$ , т. е. для случая с одной ВПП это слагаемое можно игнорировать.

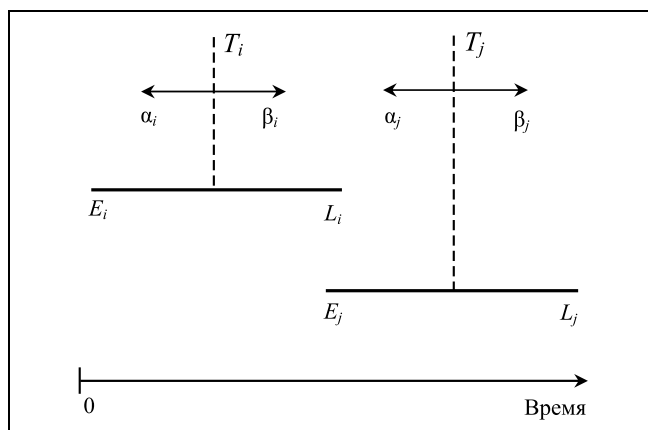


Рис. 2. Пример двух перекрывающихся временных окон посадки

Аналогичное ограничение может быть получено для ВС  $i$  и ВС  $j$ , если  $T_i < T_j$  и  $(T_j - T_i) < s_{ij}$ .

### 2.2. Эвристический алгоритм решения задачи

Для решения задачи предложен эвристический алгоритм [11], основанный на нескольких перезапусках по мере улучшения найденного решения. Он заключается в следующем.

**Шаг 1.** Применить эвристику для создания верхней границы  $Z_{UB}$ .

**Шаг 2.** Воспользоваться значением  $Z_{UB}$  для сжатия временных окон.

**Шаг 3.** Решить получившуюся линейную релаксацию задачи:

- если будет найдено улучшенное возможное решение в течение  $P$  секунд, то прекратить его поиск, обновить значение  $Z_{UB}$  и перейти к шагу 2;

- в противном случае продолжить поиск решения до завершения.

Решение может перезапускаться несколько раз, каждый раз с улучшенным возможным решением.

Для решения смешанной целочисленно-бинарной задачи с помощью линейной релаксации авторы [11] воспользовались программным пакетом CPLEX.

#### 2.2.1. Поиск верхней границы

Если найдена верхняя граница  $Z_{UB}$  оптимального решения задачи (значение целевой функции для некоторого известного допустимого решения), то можно сжать временные окна для каждого ВС. Кроме того, как будет описано ниже, верхней границей  $Z_{UB}$  можно воспользоваться для сокращения перебора в методе ветвей и границ.

Для нахождения верхней границы решения задачи применяется следующая эвристика.



*Шаг 1.* Пусть  $A_r$  ( $r = 1, \dots, R$ ) — упорядоченное множество ВС, приземляющихся на ВПП  $r$ , где первоначально  $A_r = \emptyset \forall r$ .

*Шаг 2.* Рассмотрим ВС в порядке возрастания целевого времени  $T_j$ , и для каждого такого ВС  $j$ :

— вычислим для каждой ВПП  $r$  значение

$$B_r = \max[T_j, \max[x_k + S_{kj} | \forall k \in A_r], \max[x_k + s_{kj} | u = 1, \dots, R \ u \neq r \ \forall k \in A_u]];$$

это выражение дает наилучшее (наименьшее) время  $B_r$ , в которое ВС  $j$  может приземлиться на ВПП  $r$ , учитывая более ранние посадки на эту ВПП и более ранние посадки на все другие ВПП;

— пусть  $\gamma$  — ВПП, для которой соответствующее значение  $B_\gamma = \min[B_r | r = 1, \dots, R]$ ; добавим  $j$  в множество  $A_\gamma$  и положим  $x_j = B_\gamma$ .

*Шаг 3.* Одна из особенностей шагов 1 и 2 заключается в том, что ВС никогда не приземляется до целевого времени, всегда вовремя или после него.

Чтобы улучшить полученное решение, на данном шаге пересчитываются времена посадок с помощью решения задачи линейного программирования. Это может быть сделано потому, что на шагах 1 и 2 сделан выбор всех целочисленных переменных задачи линейного программирования — определены номера ВПП для каждого ВС и порядок посадок ВС на каждую ВПП.

### 2.2.2. Сжатие временных окон

Если известна любая верхняя граница оптимального решения  $Z_{UB}$ , то исходные временные окна можно сжать. Поясним это для случая целевой функции вида:

$$\min \sum_{i=1}^P (g_i \alpha_i + h_i \beta_i).$$

Для ВС  $i$  можно ослабить ограничение  $E_i$  (см. п. 1.1), воспользовавшись формулой

$$E_i = \max \left[ E_i, T_i - \frac{Z_{UB}}{g_i} \right], \quad i = 1, \dots, P.$$

Это можно сделать на основе предположения, что когда все остальные ВС вносят нулевой вклад в целевую функцию, то если ВС  $i$  отклонится от целевого времени сильнее, то мы получим решение хуже, чем  $Z_{UB}$ .

Аналогично можно ослабить ограничение  $L_i$ :

$$L_i = \min \left[ L_i, T_i - \frac{Z_{UB}}{h_i} \right], \quad i = 1, \dots, P.$$

### 2.3. Модель генерации столбцов для задачи построения оптимальной очереди на посадку

Автор работы [13] считает, что рассмотренная линейная релаксация [11] обеспечивает плохую оценку оптимального значения для исходной смешанной целочисленной модели для задачи ФООП на несколько ВПП. Для протестированных примеров наихудшее решение составило около 77 % от оптимального. Для улучшения модели [11] предлагается переформулировать задачу как задачу генерации столбцов [13].

Обозначения:  $P$  — число ВС;  $S$  — множество допустимых последовательностей посадки, предполагается, что ВС из одной последовательности приземляются на одну ВПП;

$$c_s = \sum_{i=1}^P (g_i \alpha_i a_i^s + h_i \beta_i a_i^s),$$

$c_s$  — стоимость последовательности.

Будем называть столбцом такой вектор  $a^s$  — вектор размерности  $P$ , в котором

$$a_i^s = \begin{cases} 1, & \text{если ВС } i \text{ принадлежит} \\ & \text{последовательности } s, s \in S, \\ 0 & \text{в противном случае.} \end{cases}$$

Результирующая модель задачи генерации столбцов:

минимизируется целевая функция, равная суммарной стоимости для всех последовательностей из  $S$ :

$$\min \sum_{s \in S} c_s z_s,$$

где  $z_s = \begin{cases} 1, & \text{если последовательность } s \\ & \text{включается в решение,} \\ 0 & \text{в противном случае.} \end{cases}$

Ограничения: каждое ВС приземляется строго на одну ВПП:

$$\sum_{s \in S} a_i^s z_s = 1 \quad \forall i \in P \tag{1}$$

предел численности ВПП

$$\sum_{s \in S} z_s = R. \tag{2}$$

Поясним постановку задачи на примере, представленном в табл. 2, для трех ВС, которые приземляются на две ВПП. Время разделения (минимальный безопасный интервал) между любыми двумя ВС предполагается равным 10 [13].



Допустимые последовательности должны удовлетворять заданным временным окнам и ограничениям разделения. Например, последовательность  $\{2 \rightarrow 1\}$  не является допустимой, поскольку  $88 + 10 > 95$  ( $E_2 + S_{21} > L_1$ ), это означает, что второе ВС не может приземлиться перед первым ВС на одну ВПП, и поэтому последовательность  $\{2 \rightarrow 1\}$  не включена во множество допустимых последовательностей посадки  $S$ .

В табл. 3 представлены допустимые последовательности для примера, соответствующие времена приземления и соответствующие стоимости.

Далее рассматривается задача минимизации стоимости и нахождения оптимальных времен посадок для допустимой последовательности из  $S$ . Она может рассматриваться как задача ФООП для одной ВПП. Обозначим  $P_S$  множество ВС последовательности  $S$ , а  $U_S$  — множество упорядоченных пар последовательности множества  $S$ . Например, для последовательности 10 (см. табл. 3):  $\{1 \rightarrow 2 \rightarrow 3\}$   $P_{10} = \{1, 2, 3\}$  и  $U_{10} = \{(1, 3), (1, 2), (3, 2)\}$ .

Формулировка задачи: найти

$$\min \sum_{i \in P_S} (g_i \alpha_i + h_i \beta_i)$$

Таблица 2

Пример задачи для трех ВС

ВС	$E_i$	$T_i$	$L_i$	$g_i$	$h_i$
1	50	88	95	3	1
2	88	95	105	3	1
3	75	100	120	3	1

Таблица 3

Допустимые последовательности для примера (см. табл. 2)

Последовательность	Допустимые последовательности	Времена приземления	Общая стоимость
1	{1}	{88}	0
2	{2}	{95}	0
3	{3}	{100}	0
4	{1 → 2}	{88 98}	3
5	{1 → 3}	{88 100}	0
6	{3 → 1}	{85 95}	52
7	{2 → 3}	{95 105}	5
8	{3 → 2}	{95 105}	25
9	{1 → 2 → 3}	{88 98 108}	11
10	{1 → 3 → 2}	{85 95 105}	34

при ограничениях:

$$\begin{aligned} E_i &\leq x_i \leq L_i \quad \forall i \in P_S; \\ x_j &\geq x_i + S_{ij} \quad \forall (i, j) \in U_S; \\ \alpha_i &\geq T_i - x_i \quad \forall i \in P_S; \\ 0 &\leq \alpha_i \leq T_i - E_i \quad \forall i \in P_S; \\ \beta_i &\geq x_i - T_i \quad \forall i \in P_S; \\ 0 &\leq \beta_i \leq L_i - T_i \quad \forall i \in P_S; \\ x_i, \alpha_i, \beta_i &\geq 0 \quad \forall i \in P_S. \end{aligned}$$

Это линейная задача, она позволяет находить оптимальное решение для любой заданной последовательности посадки.

Столбец в постановке не содержит информации о порядке посадок. Например, для столбца [1, 1, 1] существуют две допустимые последовательности посадок: последовательность 9: {1 → 2 → 3} со стоимостью 11 и последовательность 10: {1 → 3 → 2} со стоимостью 34 (см. табл. 3). Поскольку нужно минимизировать общую стоимость, то для столбца [1, 1, 1] коэффициент в целевой функции равен минимальной стоимости из всех стоимостей, соответствующих этому столбцу допустимых последовательностей.

Формулируется математическая модель определения минимальной стоимости для столбца. Эта задача оптимизации отличается от задачи минимизации стоимости для заданной последовательности посадок, поскольку порядок посадок не известен.

Вводятся дополнительные ограничения:

$$\begin{aligned} \delta_{ij} + \delta_{ji} &= 1 \quad \forall i, j \in P_a, \quad i \neq j; \\ x_j &\geq x_i + S_{ij} \delta_{ij} - (L_i - E_j) \delta_{ji} \quad \forall i, j \in P_a, \quad i \neq j, \end{aligned}$$

где  $P_a$  — множество ВС, имеющихся в столбце  $a$ .

В результате получается смешанная целочисленная постановка для определения минимальной стоимости, последовательности посадок и времени посадок для столбцов в задаче генерации столбцов (табл. 4).

Имея информацию о допустимых последовательностях посадок и соответствующих стоимостях, задачу ФООП можно сформулировать как задачу генерации столбцов. Соответствующая модель для рассматриваемого примера:

$$\min(0z_1 + 0z_2 + 0z_3 + 3z_4 + 0z_5 + 5z_6 + 11z_7)$$

при ограничениях (1):

$$\begin{aligned} z_1 + z_4 + z_5 + z_7 &= 1; \\ z_2 + z_4 + z_6 + z_7 &= 1; \\ z_3 + z_6 + z_7 &= 1, \end{aligned}$$



Таблица 4

Полная информация для задачи генерации столбцов

Переменные	$c_s$	$a'_s$	Допустимые последовательности	Времена приземления
$z_1$	0	[1 0 0]	{1}	{88}
$z_2$	0	[0 1 0]	{2}	{95}
$z_3$	0	[0 0 1]	{3}	{100}
$z_4$	3	[1 1 0]	{1 → 2}	{88 98}
$z_5$	0	[1 0 1]	{1 → 3}	{88 100}
$z_6$	5	[0 1 1]	{2 → 3}	{95 105}
$z_7$	11	[1 1 1]	{1 → 2 → 3}	{88 98 108}

и ограничении (2):

$$z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 = 2.$$

Для рассматриваемого примера оптимальное решение имеет вид:  $Z^* = [0, 1, 0, 0, 1, 0, 0]$ ;

ВВП	Столбец	Последовательность	Стоимость
1	[0 1 0]	{2}	0
2	[1 0 1]	{1 → 3}	0

2.4. Метод branch-and-price

В работе [13] для решения задачи ФООП предлагается применять метод branch-and-price — метод ветвей и границ, дополненный алгоритмом генерации столбцов. Авторы утверждают, что метод branch-and-price известен как эффективный метод при решении задач большой размерности в различных областях, но для решения задачи ФООП ранее не применялся.

2.4.1. Генерация столбцов

Для небольшого примера, включающего три ВС (см. табл. 2) можно перечислить все возможности последовательностей посадок, как показано в табл. 3. Однако для задачи с 50 ВС существует уже

$$\sum_{k=1}^{50} \binom{50}{k} \approx 1,1259 \times 10^{15}$$

допустимых столбцов и перечислять все эти столбцы неэффективно. Решение начинается с ограниченной задачи генерации столбцов (которая называется master problem — МР) с небольшим подмножеством переменных, полученных в процессе инициализации. При этом значения остальных переменных фиксируются равными нулю. Затем проверяется, может ли добавление одной или нескольких переменных улучшить решение. Согласно симплекс алгоритму, это может быть достигнуто путем добавления переменной с отрицательной

приведенной стоимостью. Задача нахождения переменной с отрицательной приведенной стоимостью называется подзадачей (subproblem — SUB). Приведенная стоимость определяется так:

$$\tilde{c}_j = c_j - \pi a \quad \forall j \in S,$$

где  $c_j$  — коэффициент стоимости для столбца  $a$ ,  $\pi$  — это двойственная переменная, соответствующая каждому ограничению линейной системы, представленному в оптимальном базовом решении МР.

Математическая постановка задачи генерации столбцов SUB: найти

$$\min \left( \sum_{i=1}^P (g_i \alpha_i + h_i \beta_i) - \sum_{i=1}^P \pi_i a_i - \lambda \right)$$

при ограничениях:

$$\begin{aligned} a_i E_i &\leq x_i \leq a_i L_i \quad \forall i \in P; \\ \delta_{ij} &\leq a_i \quad \forall i, j \in P, i \neq j, \\ \delta_{ij} &\leq a_j \quad \forall i, j \in P, i \neq j, \\ 1 &\geq \delta_{ij} + \delta_{ji} \geq a_i + a_j - 1 \quad \forall i, j \in P, i \neq j, \\ x_j &\geq x_i + S_{ij} \delta_{ij} - (L_i - E_j) \delta_{ji} - (a_i L_i - a_j E_j) + \\ &\quad + (L_i - E_j) (\delta_{ij} + \delta_{ji}) \quad \forall i, j \in P, i \neq j, \\ \alpha_i &\geq a_i T_i - x_i \quad \forall i \in P; \\ 0 &\leq \alpha_i \leq T_i - E_i \quad \forall i \in P; \\ \beta_i &\geq x_i - a_i T_i \quad \forall i \in P; \\ 0 &\leq \beta_i \leq L_i - T_i \quad \forall i \in P; \\ x_i &= a_i T_i - \alpha_i + \beta_i \quad \forall i \in P, \end{aligned}$$

$$\delta_{ij}, a_i \text{ — бинарные переменные } \forall i, j \in P,$$

где  $\pi_i$  — двойственная переменная, соответствующая  $i$ -му ограничению (1),  $i = 1, \dots, P$ ,  $\lambda$  — двойственная переменная, соответствующая ограничению (2).

$$a_i = \begin{cases} 1, & \text{если ВС } i \text{ присутствует} \\ & \text{в последовательности посадки,} \\ 0 & \text{в противном случае.} \end{cases}$$

Эту постановку можно рассматривать как задачу ФООП с одной ВПП. Если ВС не присутствует в последовательности посадки, то оно не влияет на целевую функцию.

Если в этой постановке значение целевой функции неотрицательно, то задача решена, в противном случае соответствующий столбец добавляется в задачу МР, и задача решается снова.

Общая структура метода генерации столбцов: сгенерировать начальное допустимое решение, например, с помощью эвристического алгоритма, предложенного в работе [11];

*повторять:*

— решить задачу МР с помощью линейного программирования;

— найти столбцы с отрицательной приведенной стоимостью;

— добавить найденные столбцы в МР;

*завершить*, когда не существует столбцов с отрицательной приведенной стоимостью.

#### 2.4.2. Метод ветвей и границ

В большинстве случаев решение задачи генерации столбцов с помощью метода линейной релаксации является дробным. Чтобы получить целочисленное решение, применяется известный метод ветвей и границ, в котором нижняя граница для каждой вершины оценивается с помощью метода генерации столбцов (так называемый метод branch-and-price).

Решение начинается с рассмотрения исходной задачи на множестве допустимых значений, которая называется корневой задачей. К корневой задаче применяются процедуры поиска нижней и верхней границ решения. Если эти границы совпадают, то найдено оптимальное решение, и процедура прекращается. Иначе множество допустимых значений делится на два или более подмножеств. Эти подзадачи (для подмножеств) становятся дочерними вершинами корневой задачи. Алгоритм применяется рекурсивно к подзадачам, генерируя дерево подзадач.

Верхняя граница  $Z_{UB}$  является значением целевой функции, соответствующим лучшему текущему целочисленному решению. Первое такое решение получается в процессе инициализации с помощью эвристического алгоритма. На каждой итерации для одной подзадачи (вершины) ищется решение  $Z_{SV}$  с помощью описанного выше метода генерации столбцов. Возможно три случая.

*Случай 1.* Если получено целочисленное решение  $Z_{SV}$ , оно сравнивается с текущей верхней границей решения  $Z_{UB}$ . Если  $Z_{SV} < Z_{UB}$ , то эта вершина становится лучшей целочисленной вершиной, и верхняя граница решения задачи обновляется:  $Z_{UB} = Z_{SV}$ , а нижняя граница каждой активной вершины дерева сравнивается с этой новой верхней границей. Те вершины, нижняя граница  $Z_{SV}$  которых больше, чем  $Z_{UB}$ , удаляются из дерева.

*Случай 2.* Если решение дробное и значение  $Z_{SV} \geq Z_{UB}$ , то вершина удаляется из дерева, так как целочисленные решения ее потомков не будут лучше полученного дробного решения.

*Случай 3.* Если решение дробное и значение  $Z_{SV} < Z_{UB}$ , то принимается решение о создании двух дочерних узлов.

Решение продолжается до тех пор, пока для всех порожденных вершин либо получено решение, либо вершины исключены из рассмотрения. Реализация операции ветвления подробно описана в работе [13].

Алгоритм протестирован на открытых данных из OR-библиотеки, включающих до 50 ВС и 4 ВПП. Авторы утверждают, что все задачи были решены с использованием менее 450 столбцов. Это указывает на потенциальную возможность применения предложенного алгоритма для решения задачи ФООП с фиксированным числом ВПП.

Процедура построения оптимальных очередей на посадку на основе метода ветвей и границ была реализована в составе комплексного исследовательского стенда полунатурного имитационного моделирования интегрированных систем управления воздушным движением, разрабатываемом в ГосНИИАС [2]. Особенность предложенного метода состоит в учете ограничений не только на минимальные интервалы между посадками ВС, но и минимальные интервалы между пролетом некоторых точек воздушного пространства.

#### ЗАКЛЮЧЕНИЕ

В точных алгоритмах применяется метод ветвей и границ для поиска оптимального решения для задачи ФООП. Однако время работы экспоненциально растет с размером задачи (числом ВС). Следовательно, его нельзя применить для оптимального решения задач большой размерности в течение приемлемого времени. Из литературы известно, что задача ФООП решается оптимально до 50 ВС.

Для получения приближенных решений задачи ФООП в литературе представлено большое число эвристических, метаэвристических, гибридных и прочих алгоритмов, в том числе с приведенными вычислительными результатами для 500 ВС и 5 ВПП, полученными за приемлемое время.

Перспективным методом построения приближенных решений задачи ФООП на одну или несколько ВПП посвящена вторая часть обзора.

#### ЛИТЕРАТУРА

1. Кумков С.И., Пятко С.Г. Задача обнаружения и разрешения конфликтных ситуаций в автоматизированной системе управления воздушным движением // Научный вестник ГосНИИ «Аэронавигация». — 2013. — № 12. — С. 35–46.
2. Дегтярев О.В., Сикачев В.Ю., Ляценко А.В. Оптимальное управление прибывающим на аэродром потоком воздушных судов // Тр. девятого междунар. симпозиума / ГосНИИАС. — М., 2010. — С. 430–435.
3. Аleshин А.В., Аleshин В.И., Бабаев Н.В., Крыжановский Г.А. Методика имитационного моделирования процессов управления потоками прилетающих и вылетающих воздушных судов при УВД в районе аэродрома // Научный вестник МГТУ ГА. — 2011. — Т. 171, № 9. — С. 121–128.



4. *Алешин А.В.* Алгоритмы формирования очередности захода на посадку при маневрировании в зоне ожидания // Вестник Санкт-Петербургского гос. ун-та гражданской авиации. — 2013. — Т. 1 (5). — С. 73–84.
5. *Лебедев Г.Н., Малыгин В.Б., Нечаев Е.Е., Тин Пхон Чжо.* Использование системы приоритетного обслуживания при внедрении автоматизированного управления прилетом-вылетом в воздушном пространстве московского аэроузла // Научный вестник МГТУ ГА. — 2012. — Т. 180. — № 6. — С. 90–98.
6. *Казakov В.А.* Планирование и организация потоков воздушного движения: учеб. пособ. — Ульяновск: УВАУ ГА, 2008. — 89 с.
7. *Стионов М.В.* Управление воздушным движением в районе аэродрома: учеб. пособ. — Ульяновск: УВАУ ГА(И), 2011. — 103 с.
8. *Методы и алгоритмы планирования и регулирования потоков воздушного движения с учетом требований по безопасности и эффективности выполнения полетов и различных систем регулирования / Аналитический обзор.* — М.: ГосНИИАС, 2016.
9. *Harikopoulos D., Neogi N.* Polynomial-Time Feasibility Condition for Multiclass Aircraft Sequencing on a Single-Runway Airport // IEEE Trans. on Intelligent Transportation Systems. — March 2011. — Vol. 12, N 1. — P. 2–14.
10. *Bennel J.A., Messgarpour M., and Potts C.N.* Airport runway scheduling, 4OR-Quarterly // Journal of Operations Research. — 2011. — Vol. 9. — P. 115–138.
11. *Beasley J.E., Krishnamoorthy M., Sharaiha Y.M., and Abramson D.* Scheduling aircraft landings — the static case // Transportation Science. — 2000. — Vol. 34, N 2. — P. 180–197.
12. *Beasley J.E., Krishnamoorthy M., Sharaiha Y.M., and Abramson D.* Displacement problem and dynamically scheduling aircraft landings // Journal of the Operational Research Society. — 2004. — Vol. 55, N 1. — P. 54–64.
13. *Wen M.* Algorithms of Scheduling Aircraft Landing Problem. — Technical University of Denmark, 2005.
14. *Xiao-Peng Ji, Xian-Bin Cao, and Ke Tang.* Sequence searching and evaluation: a unified approach for aircraft arrival sequencing and scheduling problems // Memetic Computing. — 2016. — Vol. 8, N 2. — P. 109–123.
15. *Beasley J.E., Sonander J., and Havelock P.* Scheduling aircraft landings at London Heathrow using a population heuristic // Journal of the Operational Research Society. — 2001. — Vol. 52, N 5. — P. 483–493.
16. *Yu S.P., Cao X.B., and Zhang J.* A real-time schedule method for Aircraft Landing Scheduling problem based on Cellular Automation // Applied Soft Computing. — 2011. — Vol. 11, N 4. — P. 3485–3493.
17. *Soomer M.J., Franx G.J.* Scheduling aircraft landings using airlines' preferences // European Journal of Operational Research. — 2008. — Vol. 190, N 1. — P. 277–291.
18. *Faye A.* Solving the Aircraft Landing Problem with time discretization approach // European Journal of Operational Research. — 2015. — Vol. 242, N 3. — P. 1028–1038.
19. *Erzberger H.* Design principles and algorithms for automated air traffic management. / In: Knowledge-based Functions in Aerospace Systems. — 1995. — P. 7:1–7:31.
20. *Hu X.B., Chen W.H.* Genetic Algorithm Based on Receding Horizon Control for Arrival Sequencing and Scheduling // Engineering Applications of Artificial Intelligence. — 2005. — Vol. 18, N 5. — P. 633–642.
21. *Hu X.B., Chen W.H.* Receding Horizon Control for Aircraft Arrival Sequencing // IEEE Trans. on Intelligent Transportation Systems. — 2005. — Vol. 6, N 2. — P. 189–197.
22. *Hu X.B., Di Paolo E.* Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling // IEEE Trans. on Intelligent Transportation Systems. — 2008. — Vol. 9, N 2. — P. 301–310.
23. *Zhan Z.H., Zhang J., Li Y., et al.* An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem // IEEE Trans. on Intelligent Transportation Systems. — 2010. — Vol. 11, N 2. — P. 399–412.
24. *Tang K., Wang Z., Cao X., and Zhang J.* A multi-objective evolutionary approach to aircraft landing scheduling problems // IEEE World Congress on Computational Intelligence, 2008. — P. 3650–3656.
25. *Psaraftis H.N.* A Dynamic Programming Approach for Sequencing Groups of Identical Jobs // Operations Research. — 1980. — Vol. 23, N 6. — P. 1347–1359.
26. *Andreucci A., Bianco L., and Ricciardelli S.* A simulation model for aircraft sequencing in the near terminal area // European Journal of Operational Research. — 1981. — Vol. 8, N 4. — P. 345–354.
27. *Dear R.G., Sherif Y.S.* The Dynamic Scheduling of Aircraft in High Density Terminal Areas // Microelectronics Reliability. — 1989. — Vol. 29, N 5. — P. 743749.
28. *Dear R.G., Sherif Y.S.* An Algorithm for Computer Assisted Sequencing and Scheduling of Terminal Area Operations // Transportation Research Part A: General. — 1991. — Vol. 25, N 2-3. — P. 129–139.
29. *Dear R.G.* The Dynamic Scheduling of Aircraft in the Near Terminal Area / M.I.T., Flight Transportation Laboratory report R76-9, 1976.
30. *Brinton C.R.* An Implicit Enumeration Algorithm for Arrival Aircraft Scheduling // Proc. of the IEEE/AIAA 11th Digital Avionics Systems Conference. Seattle, 1992. — P. 268–274.
31. *Abela J., Abramson D., Krishnamoorthy M., et al.* Computing Optimal Schedules for Landing Aircraft // Proc. of the 12th national conference of the Australian Society for Operations Research. Adelaide, 1993. — P. 71–90.
32. *Milan J.* The Flow Management Problem in Air Traffic Control: A Model of Assigning Priorities for Landings at a Congested Airport // Transportation Planning and Technology. — 1997. — Vol. 20, N 2. — P. 131–162.
33. *Ernst A.T., Krishnamoorthy M., and Storer T.H.* Heuristic and Exact Algorithms for Scheduling Aircraft Landings // Networks. — 1999. — Vol. 34, N 3. — P. 229–241.
34. *Beasley J.E.* OR-Library: distributing test problems by electronic mail // Journal of the Operational Research Society. — 1990. — Vol. 41, N 11. — P. 1069–1072.
35. *Tavakkoli-Moghaddam R.* Scheduling the sequence of aircraft landings for a single runway using a fuzzy programming approach // Journal of Air Transport Management. — 2012. — Vol. 25. — P. 15–18.
36. *Briskorn D., Stolletz R.* Aircraft landing problems with aircraft classes // Journal of Scheduling. — 2014. — Vol. 17, N 1. — P. 31–45.
37. *Lieder A., Briskorn D., and Stolletz R.* A dynamic programming approach for the aircraft landing problem with aircraft classes // European Journal of Operational Research. — 2015. — Vol. 243, N 1. — P. 61–69.
38. *Furini F., Kidd M.P., Persiani C.A., and Toth P.* Improved rolling horizon approaches to the aircraft sequencing // Journal of Scheduling. — 2015. — Vol. 18, N 5. — P. 435–447.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

**Вересников Георгий Сергеевич** — канд. техн. наук, ст. науч. сотрудник, ✉ veresnikov@mail.ru,

**Кулида Елена Львовна** — канд. техн. наук, ст. науч. сотрудник, ✉ lenak@ipu.ru,

**Егоров Николай Александрович** — канд. техн. наук, ст. науч. сотрудник, ✉ egorov@ipu.ru,

**Лебедев Валентин Григорьевич** — д-р техн. наук, уч. секретарь, ✉ lebedev@ipu.ru,

Институт проблем управления им. В.А. Трапезникова РАН, г. Москва.