Оптимизация, системный анализ и исследование операций

© 2022 г. Е.Л. КУЛИДА, канд. техн. наук (elena-kulida@yandex.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ ПОСЛЕДОВАТЕЛЬНОСТИ И ВРЕМЕН ПОСАДОК ВОЗДУШНЫХ СУДОВ

Рассматривается NP-трудная задача оптимизации последовательности и времен посадок воздушных судов с соблюдением необходимых ограничений. В режиме реального времени получить точное решение задачи не представляется возможным из-за большого объема вычислений. Для получения приближенного решения предлагается комплексный подход: на первом этапе применяется генетический алгоритм для получения начального решения, которое затем улучшается на основе эвристического алгоритма. Предлагаемый подход позволяет получить оптимальные или близкие к оптимальным решения за приемлемое время. Для исследования разработанных алгоритмов использовалось программное средство имитационного моделирования. Обширные вычислительные эксперименты подтвердили эффективность предлагаемого подхода.

Ключевые слова: управление воздушным движением, последовательность посадок воздушных судов, оптимизация, генетический алгоритм, эвристический алгоритм.

DOI: 10.31857/S0005231022030114

1. Введение

В настоящее время управление воздушными судами (ВС) в районе аэропорта в значительной степени базируется на решениях диспетчеров, которые разрабатывают расписания движения ВС в аэропортах, основываясь на своем опыте, интуиции и некоторых правилах планирования. Однако в последнее время по мере развития научно-исследовательской программы модернизации Европейской системы организации воздушного движения SESAR (Single European Sky ATM Research Programme) возникает необходимость автоматизации систем для обеспечения координации принятия решений в аэропортах [1]. Из-за высокой интенсивности движения в зоне аэропорта и ограниченной пропускной способности взлетно-посадочных полос решения о траектории движения каждого ВС должны приниматься с высокой точностью и за минимальное время [2]. В [3] рассматривается задача планирования траекто-

рий в зоне аэропорта с учетом жестких ограничений по разделению ВС и их бесконфликтному сближению вблизи взлетно-посадочной полосы. Поскольку взлетно-посадочные полосы являются одними из самых жестких узких мест в глобальных узловых аэропортах, одним из путей повышения эффективности аэропортов является построение оптимальных очередей ВС на посадку с целью минимизации задержек рейсов, максимизации пропускной способности аэропорта и минимизации выбросов загрязняющих веществ [4].

В [5] задача формирования оптимальных очередей ВС на посадку была формализована в виде задачи смешанного целочисленного линейного или квадратичного программирования (в зависимости от выбранной целевой функции) с использованием $\sim P^2$ целочисленных переменных, принимающих значения 0 или 1, для учета необходимых ограничений на минимальное время разделения между ВС, где P – количество ВС. Применение стандартных методов оптимизации для получения точного решения приводит к экспоненциальному росту времени счета с ростом P. Для решения задачи в режиме реального времени, когда решение необходимо пересчитывать постоянно и очень быстро, в течение нескольких секунд, становится неизбежным применение эвристических алгоритмов для сокращения перебора с учетом знаний об особенностях данной задачи.

В обзоре [6] рассматриваются различные методы приближенного решения задачи оптимизации очередей ВС на посадку: метод рассеивания и бионический метод, метод искусственных иммунных систем, табу-поиск, меметические алгоритмы и др. Для решения задачи предпринимались попытки разработать эффективные генетические алгоритмы на основе перестановок [7]. Особенность заключается в том, что для перестановок стандартные операторы скрещивания неприменимы. В [8] предложен генетический алгоритм, в котором хромосома из перестановки номеров ВС преобразуется в двоичную матрицу, которая позволяет использовать равномерный оператор скрещивания. Однако этот оператор включает матричные операции, требующие значительных вычислительных затрат, при этом оператор скрещивания в процессе генетического алгоритма выполняется многократно.

В настоящей статье предлагаются генетический алгоритм, требующий меньших вычислительных затрат, с оригинальным оператором скрещивания на основе векторов времен посадок ВС и гибридный алгоритм решения задачи, который позволяет повысить эффективность генетического алгоритма на завершающем этапе. Вычислительные эксперименты показали, что предлагаемый подход позволяет получить оптимальные или близкие к оптимальным решения за приемлемое время.

2. Постановка задачи

Задача оптимизации последовательности и времен посадок прибывающих ВС заключается в оптимизации глобальной целевой функции для группы ВС, которые находятся в зоне аэропорта с целью совершить посадку, P – число ВС, ожидающих посадки.

Для каждого ВС определено временно́е окно, в течение которого ВС с номером i может совершить посадку в соответствии с его летно-техническими характеристиками, наличием топлива, длительностью полета и т.д. E_i – самое раннее возможное время приземления i-го ВС; L_i – самое позднее возможное время приземления i-го ВС. Для каждого ВС известно также время T_i – оптимальное время прибытия i-го ВС при условии свободной взлетно-посадочной полосы, $E_i < T_i < L_i$, $i = \overline{1,P}$.

Требуется определить последовательность ВС при посадке и для каждого ВС назначить x_i – время приземления i-го ВС в течение заданного интервала времени $[T_0, T_k]$.

В зависимости от решаемой задачи могут формироваться различные целевые функции. В настоящей статье рассматривается нелинейная целевая функция — сумма квадратов отклонений от оптимальных времен посадок:

$$F(X) = \sum_{i=1}^{P} (T_i - x_i)^2, \quad X = \{x_i,\ i = \overline{1,P}\}. \tag{1}$$

В связи с образованием струйно-вихревого следа необходимо обеспечить некоторый минимальный временной интервал между посадками ВС, зависящий от типов следующих друг за другом ВС.

Известна матрица S размера $P \cdot P$, где $S_{C_iC_j}$ – минимальный интервал между посадкой ВС типа c_j после ВС типа c_i, $i, j = \overline{1,P}$, $i \neq j$, c_i – тип i-го ВС.

Решение является допустимым, если для него выполнены два ограничения.

1. Приземление ВС с номером i происходит внутри временного окна $[E_i, L_i]$:

$$E_i \leq x_i \leq L_i, \quad i = \overline{1,P}. \tag{2}$$

2. Соблюдаются необходимые временны́е интервалы, зависящие от типов ВС, при посадке следующих друг за другом ВС:

$$x_j \geq x_i + S_{C_iC_j}, \quad i \neq j, \quad x_j > x_i, \quad i, j = \overline{1,P}. \tag{3}$$

3. Описание генетического алгоритма

Для задачи оптимизации последовательности и времен посадок группы из P ВС, занумерованных от 1 до P, особь в генетическом алгоритме представляется в виде двух целочисленных векторов $R = \{Y, X\}$.

Вектор $Y = \{y_i,\ i = \overline{1,P}\}$ представляет собой перестановку номеров ВС и определяет последовательность посадок ВС. Вектор $X = \{x_i,\ i = \overline{1,P}\}$ представляет собой упорядоченную по возрастанию последовательность смещений времен посадок ВС в секундах относительно начального времени T_0. Время посадки ВС с номером i равно $T_0 + x_{y_i}$.

3.1. Алгоритм формирования особи случайным образом

Шаг 1. Генерируется вектор Z длины P. Компоненты вектора генерируются при помощи датчика случайных чисел в диапазоне от T_0 до T_K. Этот вектор используется для формирования последовательности посадок ВС.

Шаг 2. Вектор $\overline{Y}$ полагается равным упорядоченной по возрастанию номеров последовательности от 1 до P: $\overline{Y} = \{1, \ldots, P\}$.

Шаг 3. Векторы Z и $\overline{Y}$ параллельно сортируются в порядке возрастания z_i, при этом формируется новый вектор Y, который является искомой перестановкой номеров ВС при посадке.

Шаг 4. Полученному вектору Y ставится в соответствие новый вектор X одним из следующих двух способов.

Способ 1. Сохранив последовательность посадок, определяемую вектором Y, вектор X формируем так:

$$x_{y_0} = \max(T_0, E_{y_0}), \quad \max\left(x_{y_{i-1}} + S_{C_{y_i}, C_{y_{i-1}}}, E_{y_i}\right), \quad i = \overline{2, P}. \tag{4}$$

В силу такого формирования вектора X, особь $R = \{Y, X\}$ будет удовлетворять ограничению (3), при этом ограничение (2) может быть нарушено. Величина нарушения рассчитывается по формуле:

$$W_1(R) = \sum_{i=1}^{P} \max(0, x_{y_i} - L_{y_i}).$$

Способ 2. Сохранив последовательность посадок, определяемую вектором Y, вектор X формируем так:

$$x_{y_0} = \max(T_0, E_{y_0}), \quad x_{y_i} = \min\left(\max\left(x_{y_{i-1}} + S_{C_{y_i}, C_{y_{i-1}}}, E_{y_i}\right) L_{y_i}\right), \quad i = \overline{2, P}.$$

В силу такого формирования вектора X особь $R = \{Y, X\}$ будет удовлетворять ограничению (2), при этом ограничение (3) может быть нарушено. Величина нарушения рассчитывается по формуле:

$$W_2(R) = \sum_{i=1}^{P} \sum_{\substack{j=1, \\ j \neq i,\ x_j > x_i}}^{P} \max\left(0, S_{C_i C_j} - (x_j - x_i)\right).$$

Если $W(R) = W_1(R) + W_2(R) = 0$, то R является решением задачи.

3.2. Создание начальной популяции

Начальная популяция $\{R\}$ состоит из N особей, созданных по алгоритму формирования особи случайным образом. Размер популяции N является параметром алгоритма.

Особенностью рассматриваемой задачи является наличие исходного решения $R = \{Y, X\}$, в котором компоненты вектора Y упорядочены в порядке

возрастания времен T_i. Такая последовательность называется *FCFS*-последовательностью (*First-Come*, *First-Served* – первым пришел, первым обслужен). Вектор X исходного решения формируется на основе *FCFS*-последовательности по процедуре, описанной в шаге 4 алгоритма формирования особи случайным образом. Для повышения эффективности алгоритма исходное решение добавляется в начальную популяцию вместо одного из случайных решений.

3.3. Оценка и упорядочение решений

Особи оцениваются и сравниваются по значениям целевой функции (1) и величинам нарушения $W(R)$.

Сравнение решений.

Решения $R_1 = \{Y_1, X_1\}$ и $R_2 = \{Y_2, X_2\}$ сравниваются следующим образом.

Если $W(R_1) < W(R_2)$, то R_1 лучше, чем R_2.

Если $W(R_1) = W(R_2)$ и $F(X_1) < F(X_2)$, то R_1 лучше, чем R_2.

Все решения популяции оцениваются и упорядочиваются от лучших к худшим.

3.4. Операторы скрещивания и мутации

Особенностью генетического алгоритма решения задачи оптимизации последовательности и времен посадок прибывающих ВС является нестандартный оператор скрещивания. Стандартный оператор скрещивания в данном случае применить нельзя, поскольку полученные в результате векторы могут не являться перестановками номеров ВС. Рисунок 1 демонстрирует это на примере одноточечного скрещивания.

Алгоритм модифицированного оператора скрещивания.

Шаг 1. На первом шаге алгоритма для выбранной пары особей-родителей $R_1 = \{Y_1, X_1\}$ и $R_2 = \{Y_2, X_2\}$ выполняется стандартный оператор скрещивания над векторами X_1 и X_2, в результате формируются вспомогательные векторы Z_1 и Z_2.

Шаг 2. Формируются векторы $\overline{\overline{Y}}_1$ и $\overline{\overline{Y}}_2$, равные упорядоченным по возрастанию последовательностям номеров от 1 до P: $\overline{\overline{Y}}_1 = \overline{\overline{Y}}_2 = \{1, \ldots, P\}$.

Шаг 3. Векторы Z_1 и $\overline{\overline{Y}}_1$ параллельно сортируются в порядке возрастания компонент вектора Z_1, при этом формируется новый вектор $\overline{Y}_1$, который является искомой перестановкой номеров ВС первой особи-потомка. Аналогично при помощи векторов Z_2 и $\overline{\overline{Y}}_2$ формируется вектор $\overline{Y}_2$, который является искомой перестановкой номеров ВС второй особи-потомка.

Рис. 1. Стандартный оператор одноточечного скрещивания.

Шаг 4. Вектору $\overline{Y}_1$ ставится в соответствие вектор $\overline{X}_1$ способом, описанным в шаге 4 алгоритма формирования особи случайным образом, приведенного в подразделе 3.1. В процессе формирования над каждым компонентом $\overline{x}_i$ особи-потомка с заданной параметром алгоритма вероятностью p_m выполняется оператор мутации. Оператор мутации заключается в замене значения $\overline{x}_i$ случайно сгенерированным числом в диапазоне от T_0 до T_K. Таким образом, формируется особь-потомок $\overline{R}_1 = \{\overline{Y}_1, \overline{X}_1\}$.

На основе вектора $\overline{Y}_2$ аналогично формируется особь-потомок $\overline{R}_2 = \{\overline{Y}_2, \overline{X}_2\}$.

3.5. Формирование новой популяции

Выбор решений-родителей выполняется с учетом значений их целевых функций следующим образом.

- Из популяции $\{R\}$ случайным образом выбирается решение $\hat{R} = \{\overline{X}, \overline{Y}\}$.
- Для выбранного решения рассчитывается величина

$$retF = 1 - F(\overline{X}) / \max_{\{R\}} F(X).$$

- Генерируется случайное вещественное число rnd в диапазоне от 0 до 1.
- Если $rnd < retF$, то решение $\overline{R} = \{\overline{X}, \overline{Y}\}$ выбирается в качестве решения-родителя, в противном случае процедура повторяется.

После выбора двух решений-родителей над ними выполняется оператор скрещивания и формируются два решения-потомка. Эта процедура повторяется до тех пор, пока не будет сформировано P решений-потомков. Затем все $2P$ решений оцениваются и упорядочиваются. Из P лучших решений формируется новая популяция.

Процесс формирования новой популяции повторяется циклически. Количество повторений определяется параметром алгоритма — количеством поколений M.

3.6. Завершение алгоритма

Работа алгоритма завершается либо после формирования заданного количества поколений M, либо в случае, если популяция стала однородной. Популяция считается однородной, если близко к единице отношение:

$$\min_{X \in \{R\}} F(X) \Big/ \operatorname*{avg}_{X \in \{R\}} F(X) > 0{,}98.$$

В качестве решения задачи выбирается лучшее решение из последней популяции.

Поскольку исходное решение добавляется в начальную популяцию, то достаточным условием существования решения задачи является условие $W(X) = 0$ для исходного решения.

Отметим, что задача может не иметь решения, если недостаточна пропускная способность взлетно-посадочной полосы. Для фиксированной группы ВС существует минимальное время $T_{\min}$, которое при посадке этих ВС с соблюдением минимальных временных интервалов уменьшить нельзя ни при какой последовательности посадок. Необходимым условием существования решения является условие $T_k - T_0 \geq T_{\min}$.

4. Организация вычислительных экспериментов

Для исследования зависимости результата от параметров алгоритма разработано программное средство имитационного моделирования.

Программное средство позволяет:

- генерировать тесты с заданными параметрами, главным из которых является интенсивность потока ВС на посадку, зависящая от P и величины временного интервала $Time = T_k - T_0$,
- выбирать целевую функцию (1) или (2),
- выбирать алгоритм решения и задавать параметры алгоритма,
- в автоматическом режиме набирать статистику по заданному количеству тестов и сохранять ее в файле.

Тесты генерируются следующим образом.

Тип ВС определяется при помощи функции генерации случайного целого числа из заданного диапазона $1, \ldots, K$: $C_i = rnd.Next\,(K)$, $i = \overline{1, P}$.

Времена $0 \leq E_i \leq T_i \leq L_i \leq Time$, $i = \overline{1, P}$, для всех ВС формируются при помощи датчика случайных вещественных чисел.

$T_i = rnd.Next\,(T_0, T_0 + Time)$ – оптимальное время прибытия i-го ВС – случайное число в диапазоне $(\overline{T_0, T_0 + Time})$, $i = \overline{1, P}$. Интенсивность потока ВС варьируется параметром $Time$.

$E_i = T_0 - t - rnd.Next(2t)$ – самое раннее возможное время приземления i-го ВС, $i = \overline{1, P}$, t – ориентировочная минимальная длительность окна посадки, например $t = 300$ с, $rnd.Next(t)$ – случайный разброс времени окна посадки в диапазоне $\overline{0, 2t}$.

$L_i = T_0 + t + rnd.Next(2t)$ – самое позднее возможное время приземления i-го ВС, $i = \overline{1, P}$.

Подробно программное средство моделирования описано в [9].

Для оценки эффективности решения, полученного при помощи генетического алгоритма, это решение на большом количестве тестов сравнивалось с исходным решением, а для задач небольшой размерности – до 20 ВС – еще и с оптимальным решением.

Рассматриваемая задача сводится к задаче смешанного целочисленного линейного или квадратичного программирования (в зависимости от выбранной целевой функции) при помощи введения дополнительных переменных [5]:

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

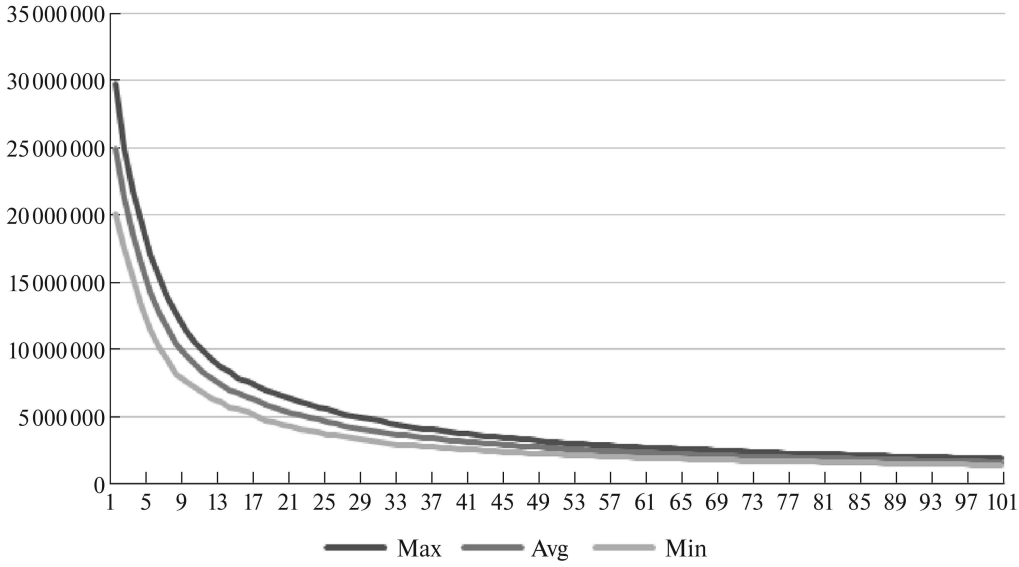

Рис. 2. Пример изменения значений целевой функции для последовательных поколений.

Для задач большой размерности получить точное решение не удается из-за большого объема вычислений. Для получения оптимального решения для задач небольшой размерности использовалась стандартная библиотека CPLEX. Возможность получения точного решения, его визуализация, анализ и сравнение с приближенными решениями задачи принесли большую пользу в процессе разработки и анализа алгоритмов.

Недостатками генетических алгоритмов являются снижение скорости сходимости по мере приближения к решению и возможное получение однородной популяции вблизи локального минимума. На рис. 2 представлены графики типичного изменения максимального Max, среднего Avg и минимального Min значений целевой функции для последовательных поколений. На нескольких первых поколениях целевая функция улучшается стремительно, затем скорость изменения резко снижается. В этой связи предлагается использовать комплексный подход: на первом этапе использовать генетический алгоритм для быстрого получения начального решения, а затем при помощи эвристического алгоритма улучшить это решение.

Параметры алгоритма – размер популяции N, количество поколений M – необходимо корректировать в зависимости от размерности задачи P. В [7] предлагается следующий способ расчета параметров:

$$N = 30 + 10\left(round\left(\frac{\max(0, P-10)}{5}\right)\right),$$

$$M = 20 + 10\left(round\left(\frac{\max(0, P-10)}{5}\right)\right).$$

5. Исследование эффективности комплексного алгоритма

Ранее был разработан и реализован эвристический алгоритм на основе перестановок для решения поставленной задачи на основе улучшения некоторого начального решения [9].

В основе алгоритма лежат сравнение решений с переставленными близко расположенными в последовательности посадок ВС разных типов и выбор лучшего решения. Пример представлен на рис. 3 — в результате трех перестановок значение целевой функции (2) уменьшилось с 589 512 до 306 912.

Первый шаг эвристического алгоритма заключается в итерационном процессе формирования, сравнения и выбора лучшего из двух решений, в которых переставлены два рядом стоящих компонента вектора Y, если им соответствуют ВС разных типов:

$$\begin{aligned} Y &= \{y_1, \ldots, y_j, y_{j+1}, \ldots, y_P\}, \\ \tilde{Y} &= \{y_1, \ldots, y_{j+1}, y_j, \ldots, y_P\}, \end{aligned} \qquad j = \overline{1, P-1}.$$

Для векторов Y и $\tilde{Y}$ строятся векторы X и $\tilde{X}$ описанным выше способом (4), полученные решения сравниваются и для дальнейших итераций выбирается лучшее из них.

Второй шаг эвристического алгоритма заключается в итерационном процессе формирования, сравнения и выбора лучшего из трех решений, в которых переставлены три рядом стоящих компонента вектора Y, если им соответствуют ВС разных типов:

$$\begin{aligned} &\{y_1, \ldots, y_j, y_{j+1}, y_{j+2}, \ldots, y_P\}, \\ &\{y_1, \ldots, y_{j+1}, y_{j+2}, y_j, \ldots, y_P\}, \\ &\{y_1, \ldots, y_{j+2}, y_j, y_{j+1}, \ldots, y_P\}, \end{aligned} \qquad j = \overline{1, P-2.}$$

На каждом шаге алгоритма для дальнейших итераций выбирается лучшее решение.

Этот алгоритм работает тем эффективнее, чем лучше взятое за основу начальное решение. Подробное описание и результаты исследования эффективности эвристического алгоритма приводятся в [10].

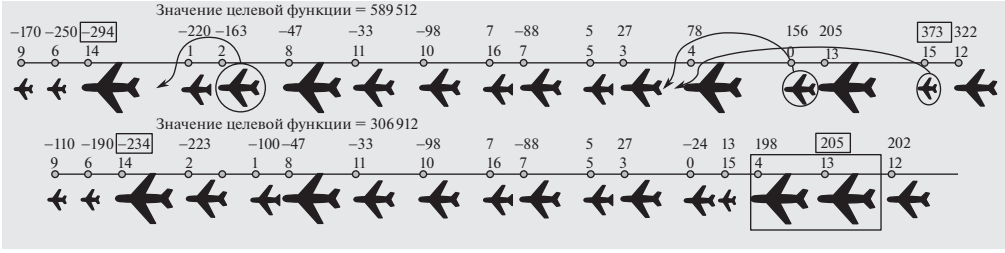

Рис. 3. Пример перестановок в последовательности посадок ВС.

Таблица 1. Значения целевой функции для разных решений для 17 ВС

Тест №	Исходное решение	Генетический алгоритм	Комплексный алгоритм, 1-й шаг	Комплексный алгоритм, 2-й шаг	Точное решение, CPLEX
1	704 400	403 068	391 788	307 564	**293 100**
2	633 057	361 836	297 801	234 696	**171 687**
3	279 885	220 277	212 469	212 469	**166 424**
4	286 196	157 848	155 688	**141 035**	**141 035**
5	544 427	244 942	**158 362**	**158 362**	**158 362**
6	261 039	109 815	**74 370**	**74 370**	**74 370**
7	233 146	127 655	**100 882**	**100 882**	**100 882**
8	232 913	212 217	210 177	175 292	**162 332**
9	162 565	162 485	**150 013**	**150 013**	**150 013**
10	211 737	**206 756**	**206 756**	**206 756**	**206 756**
11	430 268	173 584	122 384	**119 752**	**119 752**
12	323 899	259 419	243 954	243 954	**230 094**
13	462 540	322 080	266 424	266 424	**211 584**
14	220 370	181 586	135 566	**130 446**	**130 446**
15	345 061	195 525	158 253	152 916	**134 212**

Таблица 2. Значения целевой функции для разных решений для 50 ВС

Тест №	Исходное решение	Генетический алгоритм	Комплексный алгоритм, 1-й шаг	Комплексный алгоритм, 2-й шаг
1	2 904 778	1 745 359	1 039 215	1 035 355
2	1 887 645	999 401	426 044	426 044
3	2 354 880	1 231 366	1 024 269	940 472
4	1 805 502	1 768 538	1 126 340	955 292
5	3 193 848	797 364	592 548	534 562
6	1 158 859	591 509	506 445	461 345
7	2 571 624	2 417 078	1 596 856	1 364 057
8	2 310 212	589 444	427 588	403 588
9	747 182	437 120	378 117	378 117
10	3 347 467	1 150 929	878 921	775 345
11	2 462 607	2 177 331	1 619 727	1 424 103
12	3 332 179	1 353 453	648 310	629 647
13	1 989 415	1 729 610	1 070 057	961 849
14	1 395 464	1 104 507	601 173	569 085
15	2 848 916	2 219 743	1 384 726	1 375 948

Предлагается использовать комплексный алгоритм, состоящий из двух этапов.

1. Получить начальное решение при помощи предлагаемого генетического алгоритма.

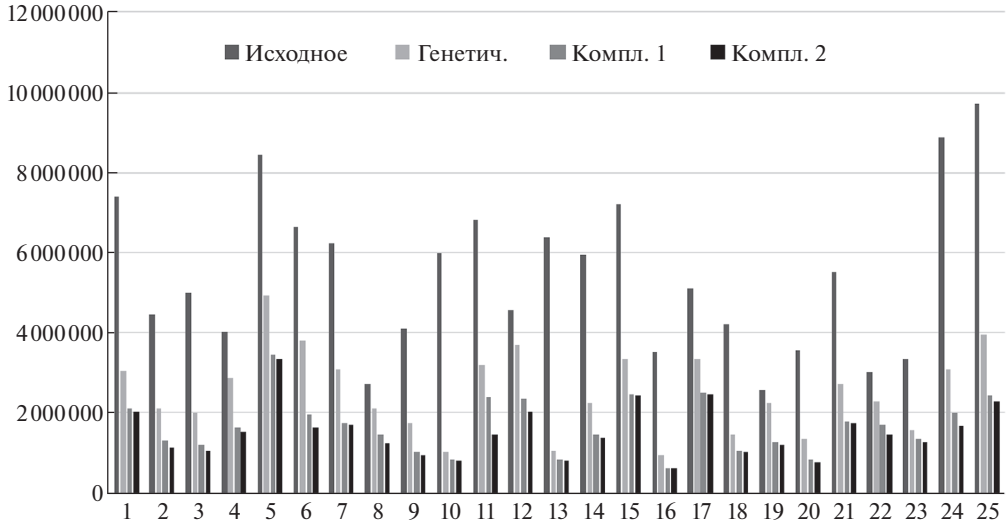

Рис. 4. Сравнение значений целевых функций, полученных на основе разных алгоритмов.

2. Получить решение при помощи эвристического алгоритма на основе перестановок, использовав начальное решение, полученное на первом этапе.

В табл. 1 приводится пример сравнения значений целевых функций (2), полученных при помощи разных алгоритмов, для 15 тестовых задач для 17 ВС. Оптимальные значения целевых функций выделены жирным шрифтом.

Для задач большой размерности получить точное решение не удается из-за вычислительной сложности. Для определения эффективности алгоритмов сравниваются значения целевой функции для исходного и полученных решений.

В табл. 2 приводится пример значений целевых функций (2), полученных при помощи разных алгоритмов, для 15 тестовых задач для 50 ВС.

На рис. 4 значения целевых функций, полученных на основе разных алгоритмов для тестов из 100 ВС, представлены в форме диаграммы.

В процессе вычислительных экспериментов выигрыш вычислялся по формуле:

$$\left(\left(F\left(X\right) - F\left(\overline{X}\right)\right)/F\left(X\right)\right)100\%,$$

где X – исходное решение, $\overline{X}$ – исследуемое решение. В табл. 3 представлены средние выигрыши, полученные после применения генетического алгоритма,

Таблица 3. Выигрыш в результате применения алгоритмов

Количество ВС	ГА	ГА + ЭА шаг 1	ГА + ЭА шаг 2
17	≈46,2%	≈54,1%	≈56,4%
50	≈51,5%	≈67,8%	≈70,1%
100	≈53,5%	≈70,2%	≈72,4%

после первого шага эвристического алгоритма и после второго шага эвристического алгоритма.

Основной вклад в уменьшение значения целевой функции вносит применение генетического алгоритма, 1-й шаг эвристического алгоритма позволяет значительно улучшить полученное решение, уменьшение значения целевой функции после 2-го шага эвристического алгоритма менее существенно.

6. Заключение

В Российской Федерации в полном объеме автоматизированное регулирование потоков воздушного движения не внедрено, поэтому развитие теории, методов и алгоритмов для поддержки систем планирования и регулирования потоков воздушного движения в условиях увеличивающейся интенсивности воздушного движения является актуальной проблемой.

Проведенные исследования подтвердили целесообразность применения в реальных условиях, для задач оптимизации последовательности и времен посадок ВС большой размерности (более 20 ВС), методов приближенного решения, которые позволяют получить за приемлемое время (за несколько секунд) хорошее, хотя не всегда оптимальное решение.

Представлены три оригинальных алгоритма формирования очередей ВС на посадку: генетический, эвристический и комплексный, в котором решение, полученное на основе генетического алгоритма, используется в качестве начального решения для эвристического алгоритма. Разработанные алгоритмы реализованы в виде библиотеки программ.

Для исследования эффективности и обоснования методов и алгоритмов построения оптимальных очередей ВС на посадку разработан инструментальный программный комплекс, который включает средства моделирования различных ситуаций воздушного движения, сбора статистического материала на основе большого числа экспериментов и анализа результатов.

Описаны вычислительные эксперименты с целью оценки эффективности и быстродействия программ, реализованных на основе разработанных алгоритмов. Для оценки эффективности использовались оптимальные результаты, полученные при помощи стандартного пакета CPLEX. Подтверждены эффективность и достаточное быстродействие реализованных программ формирования очередей ВС на посадку для задач разной размерности при интенсивном потоке ВС.

СПИСОК ЛИТЕРАТУРЫ

1. *Samà M., D'Ariano A., Corman F., Pacciarellia D.* Coordination of Scheduling Decisions in the Management of Airport Airspace and Taxiway Operations // Transportation Research. Part A: Policy and Practice. 2018. V. 114. Part B. P. 398–411.
2. *Samà M., D'Ariano A., Palagachev K., Gerdts M.* Integration Methods for Aircraft Scheduling and Trajectory Optimization at a Busy Terminal Manoeuvring Area // OR Spectrum. 2019. V. 41. P. 641–681.

3. *Ng K.K.H., Lee C.K.M., Chan F.T.S., Chen C.H., Qin Y.* A Two-stage Robust Optimisation for Terminal Traffic Flow Problem // Applied Soft Computing. 2020. V. 89. 106048.
4. *Yin J., Ma Y., Hu Y., et al.* Delay, Throughput and Emission Tradeoffs in Airport Runway Scheduling with Uncertainty Considerations // Netw Spat Econ. 2021. 21. P. 85–122.
5. *Beasley J.E., Krishnamoorthy M., Sharaiha Y.M., Abramson D.* Scheduling Aircraft Landings – the Static Case // Transportation Science. 2000. V. 34. No. 2. P. 180–197.
6. *Вересников Г.С., Егоров Н.А., Кулида Е.Л., Лебедев В.Г.* Методы построения оптимальных очередей воздушных судов на посадку. Ч. 2. Методы приближенного решения // Проблемы управления. 2018. № 5. С. 2–13.

   Veresnikov G.S., Egorov N.A., Kulida E.L., Lebedev V.G. Methods for Solving of the Aircraft Landing Problem. II. Approximate Solution Methods // Autom. Remote Control. 2019. V. 80. No. 8. P. 1502–1518.
7. *Hu X.B., Chen W.H.* Genetic Algorithm Based on Receding Horizon Control for Arrival Sequencing and Scheduling // Eng. Appl. Artif. Intell. 2005. V. 18. No. 5. P. 633–642.
8. *Hu X.B., Di Paolo E.* Binary-representation-based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling // IEEE Trans. on Intelligent Transportation Syst. 2008. V. 9. No. 2. P. 301–310.
9. *Kulida E.L.* Analysis of Algorithms for Solving the Aircraft Landing Problem // Proc. 13th Int. Conf. "Management of Large-Scale System Development" (MLSD). Moscow: IEEE, 2020. С. https://ieeexplore.ieee.org/document/9247839.
10. *Кулида Е.Л., Лебедев В.Г., Егоров Н.А.* Исследование эффективности алгоритма оптимизации потока воздушных судов на посадку // Проблемы управления. 2019. № 6. С. 63–69.

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

Поступила в редакцию 19.04.2021

После доработки 21.10.2021

Принята к публикации 20.11.2021