СРАВНЕНИЕ ДВУХ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ ПОСЛЕДОВАТЕЛЬНОСТИ И ВРЕМЕН ПОСАДОК ВОЗДУШНЫХ СУДОВ

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

Институт проблем управления им. В.А. Трапезникова,
Россия, г. Москва, ул. Профсоюзная, д.65
elena-kulida@yndex.ru, lebedev@ipu.ru, egorov@ipu.ru

Аннотация: В докладе предлагается генетический алгоритм с оригинальным оператором скрещивания и гибридный алгоритм, который позволяет повысить эффективность генетического алгоритма на завершающем этапе решения задачи. Сравнивается эффективность предлагаемого гибридного алгоритма и известного генетического алгоритма с представлением хромосом в виде двоичных матриц.

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

Введение

Доклад посвящен задаче оптимизации очереди воздушных судов (ВС) на посадку, которая становится все более актуальной в связи с недостаточной пропускной способностью взлетно-посадочных полос крупных аэропортов при увеличивающемся трафике воздушного движения.

Задача была сформулирована уже достаточно давно. Известна ее математическая постановка в виде задачи смешанного целочисленного линейного или квадратичного программирования (в зависимости от выбранной целевой функции) с использованием $O(P^2)$ целочисленных 0-1 переменных для учета необходимых ограничений на минимальное время разделения между ВС, где P – количество ВС [1]. Это *NP*-полная задача, поэтому для достаточно больших P в режиме реального времени решение невозможно из-за большого времени счета. Для решения задачи предпринимались попытки разработать более быстрые алгоритмы приближенного решения, в том числе генетические алгоритмы на основе перестановок [2]. Особенность заключается в том, что для перестановок стандартные операторы скрещивания неприменимы. В [3] предложен генетический алгоритм, в котором хромосомы представляются в виде бинарных матриц, и оператор скрещивания реализуется на основе выполнения матричных операций.

В докладе предлагается генетический алгоритм, требующий меньших вычислительных затрат, с оригинальным оператором скрещивания на основе векторов времен посадок и гибридный алгоритм решения задачи, который позволяет повысить эффективность генетического алгоритма на завершающем этапе. Вычислительные эксперименты показали, что предлагаемый подход позволяет получить лучшее решение, чем генетический алгоритм с представлением хромосом в виде двоичных матриц.

1 Постановка задачи

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

Для каждого ВС определено временное окно, в течение которого ВС с номером i может совершить посадку.

E_i – самое раннее возможное время приземления i-го ВС.

L_i – самое позднее возможное время приземления i-го ВС.

$$E_i \le x_i \le L_i,\ i = \overline{1,P}. \quad (1)$$

Ti – оптимальное время приземления i-го ВС при отсутствии других ВС, $E_i < T_i < L_i,\ i = \overline{1,P}$.

Согласно правилам ИКАО нужно обеспечить необходимые временные интервалы, зависящие от типов ВС, при посадке следующих друг за другом ВС.

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

Должны выполняться условия:

$$x_j \ge x_i + S_{C_i C_j},\ \text{если } i \ne j,\ x_j > x_i,\ i,j = \overline{1,P}. \quad (2)$$

Задача заключается в определении последовательности ВС при посадке и их времен приземлений, доставляющих минимум заданной целевой функции F(X).

Предполагается, что ВС занумерованы от 1 до P.

Решение задачи включает перестановку $Y = \{y_i, i = \overline{1,P}\}$ чисел от 1 до P, задающую последовательность номеров ВС в очереди на посадку, и упорядоченный по возрастанию вектор времен посадок $X = \{x_{y_i}, i = \overline{1,P}\}$.

Детальное описание постановки задачи рассмотрено в [4].

2 Описание генетических алгоритмов

2.1 Создание начальной популяции

Начальная популяция {R} состоит из N решений, где N – размер популяции, являющийся параметром алгоритма. Все решения начальной популяции, кроме одного, генерируются случайным образом. Для этого с помощью датчика случайных чисел формируются компоненты вектора X, затем вектора X и Y параллельно сортируются в порядке возрастания компонент вектора X. Полученная перестановка номеров Y является искомой последовательностью посадок ВС, а вектор X формируется следующим образом:

$$\tilde{x}_{y_0} = \max(T_0, E_{y_0}), \tilde{x}_{y_i} = \max\left(\tilde{x}_{y_{i-1}} + S_{C_{y_i}, C_{y_{i-1}}}, E_{y_i}\right), i = \overline{2,P}. \quad (3)$$

Кроме того, к начальной популяции добавляется исходное решение, в котором в очереди на посадку ВС упорядочены в порядке возрастания времен Ti, а вектор X сформирован согласно (3).

2.2 Оценка и упорядочение решений

Решения являются допустимыми, если выполнены ограничения (1) и (2).

Величина нарушения ограничений рассчитывается по формуле:

$$W(X) = \sum_{i=1}^{P} \max(0, x_{y_i} - L_{y_i}) + \sum_{i=1}^{P} \sum_{\substack{j=1, \\ j \neq i,\ x_j > x_i}}^{P} \max\left(0, S_{C_i C_j} - (x_j - x_i)\right). \quad (4)$$

Решения $\{Y_1, X_1\}$ и $\{Y_2, X_2\}$ оцениваются и сравниваются по величинам нарушения ограничений и значениям целевых функций.

В докладе рассматривается целевая функция – сумма квадратов отклонений от оптимальных времен посадок:

$$F(X) = \sum_{i=1}^{P} (T_i - x_i)^2. \quad (5)$$

Решения сравниваются следующим образом.

Если $W(X_1) < W(X_2)$, то $\{Y_1, X_1\}$ лучше, чем $\{Y_2, X_2\}$.

Если $W(X_1) = W(X_2)$, и $F(X_1) < F(X_2)$, то $\{Y_1, X_1\}$ лучше, чем $\{Y_2, X_2\}$.

Все решения популяции оцениваются и упорядочиваются от лучших к худшим.

2.3 Операторы скрещивания и мутации

Особенностью генетического алгоритма решения задачи оптимизации последовательности посадок прибывающих ВС является нестандартный оператор скрещивания. Стандартный оператор скрещивания к перестановкам применить нельзя, поскольку полученные в результате вектора-потомки могут не являться перестановками.

Различие рассматриваемых двух генетических алгоритмов заключается в использовании различных операторов скрещивания. Первый оператор скрещивания предложен в работе [3], в его основе лежит преобразование перестановок Y в бинарные матрицы и выполнении оператора скрещивания над матрицами. Соответствующий генетический алгоритм будем называть матричным. Второй генетический алгоритм основан на операторе скрещивания векторов времен X и затем формировании по ним соответствующих перестановок Y. Этот генетический алгоритм будем называть перестановочным.

Оператор скрещивания на основе бинарных матриц.

Хромосома в матричном генетическом алгоритме представляет собой бинарную матрицу размера $P \times P$, основанную на отношении соседства между парами ВС в последовательности прибытия. Если i-ое ВС в последовательности прибытия является первым, то значение $g(i,i)$

матрицы устанавливается равным 1. Если в последовательности прибытия за i-м воздушным судном в последовательности прибытия следует j-ое ВС, то значение $g(i,j)$ матрицы устанавливается равным 1. Все остальные значения матрицы равны 0.

Например, для перестановки, представленной на рис. 1, соответствующая бинарная матрица-хромосома, представлена на рис. 2.

1	3	2	4	6	5	7	8

Рис. 1. Исходная перестановка

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

Рис. 2. Бинарная матрица-хромосома

Элементы матрицы называются генами хромосомы. В соответствии с лежащим в основе физическим смыслом последовательности прибытия каждая хромосома при $P > 1$ должна удовлетворять следующим ограничениям:

$$\sum_{i=1}^{P}\sum_{j=1}^{P} g(i,j) = P \quad (6)$$

$$\sum_{i=1}^{P} g(i,i) = 1 \quad (7)$$

$$\sum_{j=1}^{P} g(i,j) = \begin{cases} = 2, \text{если } g(i,i) = 1 \\ \leq 1, \text{если } g(i,i) = 0 \end{cases} \quad (8)$$

$$\sum_{i=1}^{P} g(i,j) = 1 \quad (9)$$

которые гарантируют, что каждое ВС появляется один и только один раз в последовательности прибытия, и каждому ВС присваивается уникальный номер в последовательности прибытия. Из ограничений (6) – (9) можно заключить, что в матрице всегда есть пустая строка. Предположим, что i-я строка пуста, т. е. $g(i,j) = 0$ для всех $j = \overline{1,P}$. Это означает, что i-ое ВС является последним в последовательности прибытия.

Алгоритм генерации матрицы-хромосомы

Шаг 1. Создаем матрицу P×P со всеми элементами, равными 0. Пусть U = {1,...,P} представляет последовательность прибытия.

Шаг 2. Случайным образом выбираем $i \in U$ и устанавливаем $g(i,i) = 1$. Удаляем i из U, т. е. пусть $U = U - \{i\}$.

Шаг 3. Пока $U \neq \emptyset$, делаем

Шаг 3.1. Случайным образом выбираем $j \in U$ и устанавливаем $g(i,j) = 1$.

Шаг 3.2. Удаляем j из U, т. е. полагаем $U = U - \{j\}$. Пусть i=j.

Конец алгоритма.

Алгоритм скрещивания на основе бинарных матриц-хромосом

Как показано на рис. 3, существуют две категории общих секций генов: первая связана с теми ВС, которым назначены одни и те же номера в двух последовательностях прибытия, вторая связана с идентичными подпоследовательностями ВС в двух последовательностях прибытия. С помощью представления перестановок вычислительно сложно идентифицировать вторую категорию общих участков генов, чтобы их наследовать. В двоичном представлении общие участки генов определяются как те строки (за исключением диагональных элементов $g(i,i)$, в строках) и те столбцы

в двух родительских хромосомах, которые имеют одинаковое значение, а также столбец и диагональ в двух родительских хромосомах, которые имеют одинаковое значение $g(i,i) = 1$.

Другими словами, если две родительские хромосомы имеют одинаковые $g(i,j) = 1, i \neq j$, то $g(m,j)$ и $g(i,m)$, за исключением $g(i,i)$, для всех $m = \overline{1,P}$ являются общими генами; если две родительские хромосомы имеют одинаковые $g(i,i) = 1$, то $g(m,i)$ и $g(m,m)$ для всех $m = \overline{1,P}$ являются общими генами. На рис. 3 (a) показаны общие участки генов двух хромосом на основе бинарного представления, где общие гены выделены серым цветом. Чтобы идентифицировать эти общие гены, нужно применить операцию "&" к двум родителям.

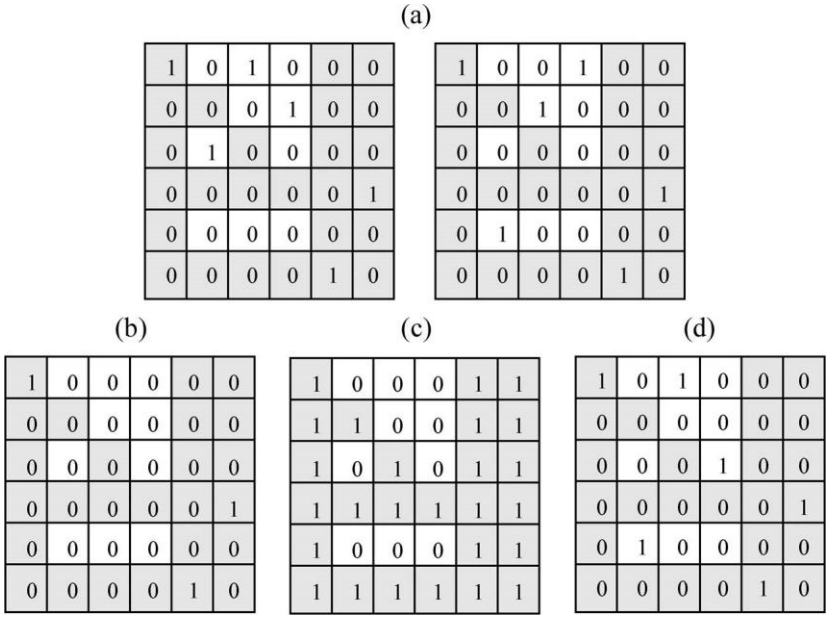

Рис. 3. Скрещивание на основе матричного представления. (a) Родительские хромосомы. (b) Оператор "&" для идентификации общих участков генов. (c) Выделение общих участков генов. (d) Рекомбинация необщих генов.

Шаг 1. Построить матрицу

$$g_3(i,j) = g_1(i,j) \,\&\, g_2(i,j) \;\; i = \overline{1,P};\; j = \overline{1,P}; \qquad (10)$$

где g_1 и g_2 – родительские хромосомы, а g_3 – результат операции "&".

На рис. 3 (b) приведен пример выполнения операции "&".

Шаг 2. На основе g_3 построить g_4, где все общие гены равны 1, а все не общие гены равны 0, как показано на рис. 3 (c).

Если $g_3(i,j) = 1$, то установить значения $g_4(m,j) = 1$ для $m = \overline{1,P}$, и значения $g_4(i,m) = 1$ для $m = \overline{1,P}, m \neq i$. Если $g_3(i,i) = 1$, установить значения $g_4(m,i) = 1$ и $g_4(m,m) = 1$ для $m = \overline{1,P}$. Значение элемента $g_4(i,j) = 1$ означает, что это общий ген.

С помощью матриц g_3 и g_4 можно эффективно рекомбинировать необщие гены следующим образом:

Шаг 3. Определим матрицу $g_5 = g_3$. Если существует i такое, что $g_3(i,i) = 1$, то перейти к шагу 2. В противном случае случайным образом выбрать i с $g_4(i,i) = 0$, установить значения $g_5(i,i) = 1$, $g_4(m,i) = 1$ и $g_4(m,m) = 1$ для $m = \overline{1,P}$.

Шаг. 4. До тех пор пока матрица g_4 не станет единичной матрицей, выполнять

Шаг 4.1. Если существуют элементы $g_4(i,j) = 0$ для некоторых значений j, то случайным образом выбрать одно из таких j, установить значения $g_5(i,j) = 1$ и $g_4(m,j) = 1$ для $m = \overline{1,P}$ и $g_4(i,m) = 1$ для $m = \overline{1,P}, m \neq i$.

В противном случае выбрать j такое, что $g_5(i,j) = 1$ и $i \neq j$.

Шаг 4.2. Установить i = j.

Пример результирующей матрицы g_5 представлен на рис. 3 (d).

Конец алгоритма.

В результате данного алгоритма ограничения (6) - (9) не нарушаются. Все общие участки генов идентифицируются и наследуются; могут быть использованы все возможные рекомбинации необщих генов.

Оператор скрещивания на основе векторов времен посадок.

Для выбранной пары решений-родителей выполняется стандартный оператор одноточечного скрещивания над векторами X1 и X2. В процессе скрещивания векторов X1 и X2 над каждым компонентом $\tilde{x}_i$ вектора-потомка с заданной параметром алгоритма вероятностью pm выполняется оператор мутации. Оператор мутации заключается в замене значения $\tilde{x}_i$ случайно сгенерированным числом в диапазоне от T_0 до T_K. Затем полученные вектора $\widetilde{X_1}$ и $\widetilde{X_2}$ упорядочиваются по возрастанию и формируются соответствующие перестановки номеров ВС $\widetilde{Y_1}$ и $\widetilde{Y_2}$. Два решения-потомка формируются по перестановкам $\widetilde{Y_1}$ и $\widetilde{Y_2}$ согласно (3).

2.4 Формирование новой популяции

После выбора двух решений-родителей над ними выполняется оператор скрещивания и формируются решения-потомки. В матричном генетическом алгоритме формируется одно решение-потомок, в перестановочном генетическом алгоритме формируется два решения-потомка. Эта процедура повторяется до тех пор, пока не будет сформировано N решений-потомков. Затем все $2N$ решений оцениваются и упорядочиваются. Из N лучших решений формируется новая популяция.

2.5 Завершение алгоритма

Если не сформировано заданное количество поколений M, то переход к пункту 2.2 – оценке и упорядочению решений. В противном случае работа алгоритма завершается.

В качестве решения выбирается лучшее решение из последней популяции.

Поскольку перестановочный генетический алгоритм существенно проще матричного и требует меньшего объема вычислений, то на завершающем этапе решения задачи для повышения эффективности он может быть дополнен эвристическим алгоритмом, предложенным в [5]. Особенностью генетических алгоритмов является быстрая сходимость на начальном этапе, в результате быстро получается не оптимальное, но достаточно хорошее решение. Предлагаемый эвристический алгоритм, наоборот, работает тем лучше, чем лучше имеющееся исходное решение. Поэтому гибридный алгоритм показывает хорошую эффективность.

3 Оценка эффективности алгоритмов

Для исследования эффективности и сравнения генетических алгоритмов проводились обширные вычислительные эксперименты. Использовалось средство имитационного моделирования [6], позволяющее генерировать задачи с заданной интенсивностью потока ВС, выбирать целевую функцию, решать задачу при помощи выбранного алгоритма, формировать статистические данные по большому количеству экспериментов.

Окно программы моделирования представлено на рис. 4.

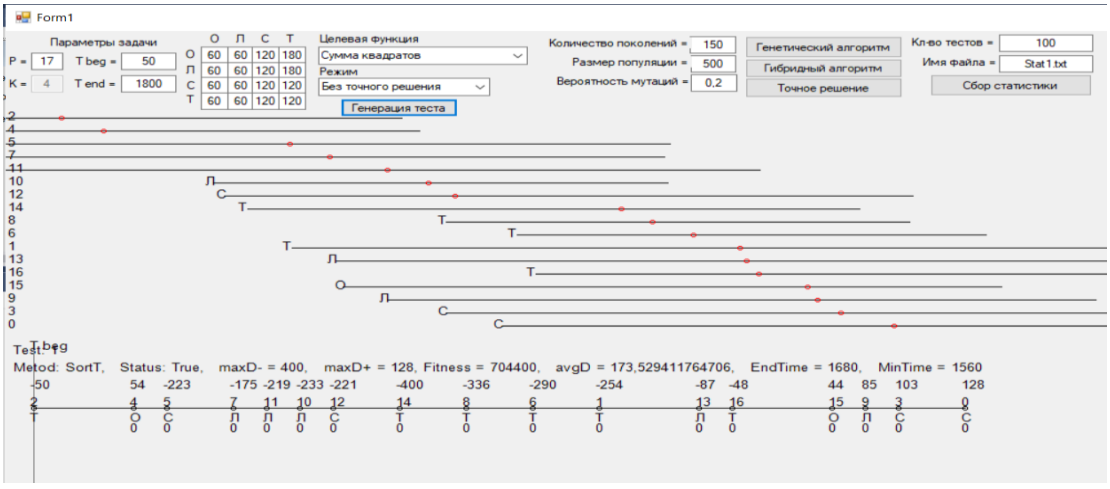

Рис. 4. Окно программы моделирования

В окне программы моделирования можно задать:

- параметры задачи: P – число ВС, T_{beg} и T_{end} – время начала и окончания посадок, матрицу S для 4 типов ВС (О – очень легкое, Л – легкое, С – среднее, Т – тяжелое), выбрать целевую функцию;
- параметры решения: количество поколений, размер популяции, вероятность мутации;
- алгоритм решения;
- количество тестов для сбора статистических данных.

Подробнее программа анализа алгоритмов оптимизации последовательности и времен посадок прибывающих воздушных судов описана в [7].

В таблице 1 сравниваются значения целевых функций (5) для задач с 50 ВС для исходного решения и решений, полученных с помощью генетического алгоритма с оператором скрещивания на основе бинарных матриц и предлагаемого гибридного алгоритма, являющегося комбинацией генетического алгоритма с оператором скрещивания на основе векторов времен посадок на первом этапе и эвристического алгоритма на втором этапе.

Таблица 1. Значения целевых функций

Тест №	**Исходное решение**	**Генетический алгоритм**	**Гибридный алгоритм**
1	2904778	1745359	1035355
2	1887645	999401	426044
3	2354880	1231366	940472
4	1805502	1768538	955292
5	3193848	797364	534562
6	1158859	591509	461345
7	2571624	2417078	1364057
8	2310212	589444	403588
9	747182	437120	378117
10	3347467	1150929	775345
11	2462607	2177331	1424103
12	3332179	1353453	629647
13	1989415	1729610	961849
14	1395464	1104507	569085
15	2848916	2219743	1375948
16	1729181	887379	472055
17	4866850	3439264	1536576
18	4467518	775564	607058
19	4065740	990716	988885
20	3138618	1289883	666822
21	2137618	909220	814770
22	2606264	1884571	1138356
23	3310856	570075	487557
24	1686094	943862	910370
25	3552100	2270444	1464992
26	4511945	1119732	530713
27	2642279	1536065	968757
28	960086	774951	532263
29	3456466	762634	456744
30	2315605	2154793	916277
31	2176322	1253555	911765
32	1638012	1201798	620830
33	2797008	979967	798266
34	994132	471412	354275
35	2938200	839467	497097
36	2436639	667246	491146
37	1491510	964342	574522
38	2510324	1958378	1165218

Для сравнения значения целевых функций решений, полученных на основе разных алгоритмов, представлены на рис. 5.

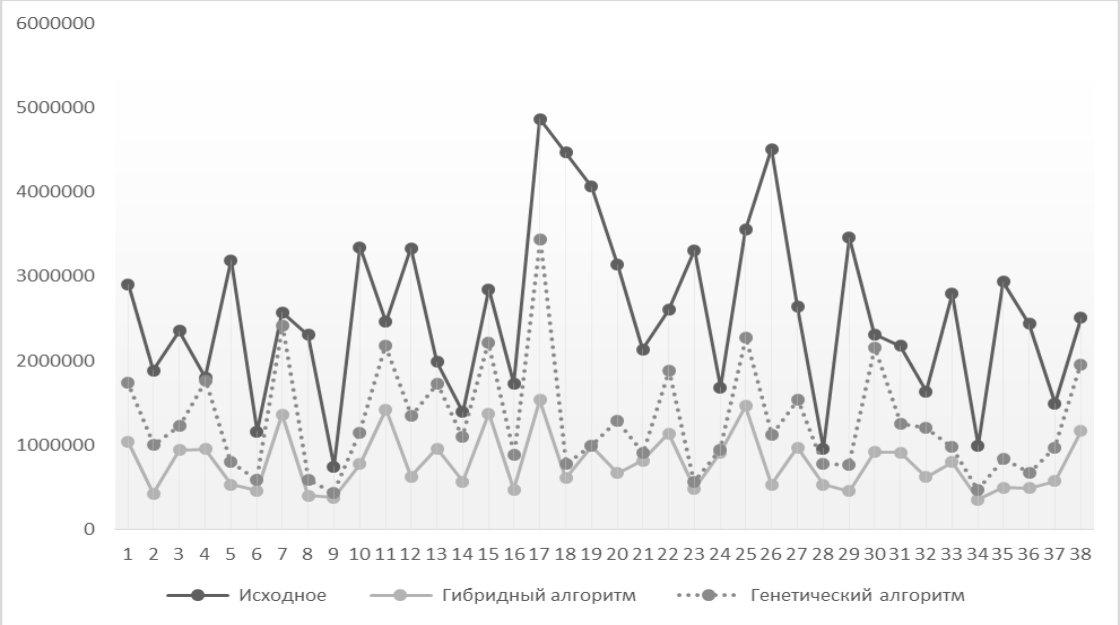

Рис. 5. Сравнение значений целевых функций для решений, полученных разными алгоритмами

Из таблицы 1 и рис. 5 видно, что предлагаемый гибридный алгоритм в большинстве случаев позволяет получить лучшее решение с меньшим значением целевой функции.

Заключение

Проведенные вычислительные эксперименты доказывают большую эффективность предлагаемого гибридного алгоритма по сравнению с генетическим алгоритмом, основанным на представлении перестановок в виде двоичных матриц. В дальнейшем предполагается модифицировать предлагаемый алгоритм для решения динамической задачи оптимизации последовательности и времен посадок ВС, когда набор группы ВС изменяется по мере посадки одних ВС и появлении в зоне аэропорта новых ВС.

Литература

1. *Beasley J.E., Krishnamoorthy M., Sharaiha Y.M., and Abramson D.* Scheduling aircraft landings – the static case // Transportation Science. Vol. 34. 2000, N 2. – P. 180-197.
2. *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.
3. *Hu X.B., Di Paolo E.* Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling // IEEE Trans. on Intelligent Transportation Systems. Vol. 9. 2008, N 2. – P. 301-310.
4. *Вересников Г.С., Егоров Н.А., Кулида Е.Л., Лебедев В.Г.* Методы построения оптимальных очередей воздушных судов на посадку. Ч. 1. Методы точного решения // Проблемы управления. 2018, № 4. – С. 2-13.
5. *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.
6. *Кулида Е.Л.* Инструментальное средство для исследования алгоритмов построения оптимальных очередей воздушных судов на посадку: Свидетельство о государственной регистрации программы для ЭВМ № 2018615221 РФ; Зарег. 03.05.2018.
7. *Кулида Е.Л.* Система анализа алгоритмов оптимизации последовательности и времен посадок прибывающих воздушных судов // Труды 13-й Международной конференции «Управление развитием крупномасштабных систем» (MLSD'2020, Москва) , под общей редакцией С.Н.Васильева, А.Д.Цвиркуна, М.: ИПУ РАН, 2020. С. 1528-1534.