МЕТОД ОПТИМИЗАЦИИ РАССТАНОВКИ ВОЗДУШНЫХ СУДОВ ПО МЕСТАМ СТОЯНОК НА ПОВЕРХНОСТИ АЭРОДРОМА В СОСТАВЕ ИМИТАЦИОННОЙ МОДЕЛИ АЭРОДРОМА

А.В. Егоркина, К.А. Вересов (Москва)

С ростом интенсивности воздушного движения всё более важную роль стала играть пропускная способность аэропорта. Она может оказать негативное влияние на пропускную способность структуры воздушного пространства в целом. Для нормального функционирования аэропорта необходимо ежедневно решать большое число оптимизационных задач, в том числе оптимального использования имеющихся ресурсов. Одной из таких задач является задача назначения мест стоянок прибывающим воздушным судам (ВС).

Решение этой задачи экспертным путём на большом аэродроме затруднено, поэтому требуется разработка определённых математических методов. Так как многие процессы, протекающие на аэродроме, достаточно сложны и подвержены влиянию случайных факторов, разработка аналитических моделей представляется сложным, а зачастую и невозможным делом. В таких случаях целесообразно использовать имитационную модель аэродрома. Такая модель была разработана ФГУП «ГосНИИАС» в составе Комплексного исследовательского стенда полунатурного имитационного моделирования интегрированных систем управления воздушным движением (КИС УВД). Задачей стенда является моделирование аэронавигационных операций в так называемом режиме «от перрона до перрона» (from gate to gate) для отработки новых принципов управления движением, предполагающих увеличение в нём роли экипажа [1]. В исследовательском стенде моделируются все этапы движения ВС и взаимодействие пилотов ВС и диспетчеров УВД, в том числе и по поверхности аэродрома. В состав стенда входит имитационная модель аэродрома [2], которая моделирует движение ВС, наземных транспортных средств, работу систем наблюдения и работу аэродромных диспетчерских служб, включая планирование использования мест стоянок.

Техническая постановка задачи

Имеется аэродром, задан набор мест стоянок воздушных судов. Каждая стоянка характеризуется координатами, наличием/отсутствием телетрапа, возможностью размещать определённые типы ВС. Заданы правила взаимозависимости мест стоянок (некоторые стоянки не могут быть заняты одновременно, на некоторые стоянки можно проехать только через другие и т.д.). Дано расписание прилётов и вылетов на фиксированный промежуток времени. Для каждого рейса в расписании известны: авиакомпания, регистрационный номер борта, тип ВС, аэропорты и плановые времена вылета и прилёта. Авиакомпании имеют предпочтения в назначении стоянок принадлежащим им ВС. Часть стоянок обслуживает только ВС определённых авиакомпаний.

Необходимо назначить места стоянок рейсам на всю глубину заданного расписания таким образом, чтобы максимально удовлетворить запросам авиакомпаний и служб аэропорта. При этом решение должно быть нечувствительным к небольшим изменениям расписания рейсов. Также должны выполняться следующие условия: 1) на одной стоянке не может одновременно находиться более одного ВС; 2) соблюдение правил взаимозависимости стоянок.

Математическая постановка задачи

Определим функцию $f:N \rightarrow M, N=\{1,\ldots,n\}, M=\{n+1,\ldots,n+m\}$, где n – количество рейсов, m – количество стоянок. На функцию действуют несколько ограничений:

1. Для каждого рейса определено множество допустимых мест стоянок:

$$f(i) \in M(i) \subseteq M \ \ \forall i \ \in N \tag{1}$$

2. Определим симметричную матрицу T размера $n \times n$. Её элементы $t_{ij} \geq 0$ соответствуют промежутку времени между двумя ВС, и $t_{ij} < 0$, если эти промежутки перекрываются на $|t_{ij}|$. Потребуем, чтобы на одной стоянке одновременно находилось не более одного ВС, т.е.

$$f(i) \neq f(j) \ \forall t_{ij} < 0, \forall i, j \ \in N \tag{2}$$

Для каждой стоянки j существует множество связанных с ней стоянок – такое, что нельзя одновременно занимать стоянки из этого множества:

$$f(i) = j, f(k) \neq l \ \forall l \in D_j, \forall i, k < n \text{ таких, что } t_{ik} < 0. \tag{3}$$

Будем рассматривать функцию $g(f) = \alpha z_1(f) + (1 - \alpha) z_2(f)$.

1. Для каждого назначения i-го рейса на j-ю стоянку определим выигрыш от этого назначения p_{ij}. Выигрыш зависит от параметров стоянки: удалённости от терминала, оборудованности телетрапом. Величины задаются до начала процедуры назначения мест стоянок и являются входными данными. Таким образом, первое слагаемое целевой функции определяется как

$$z_1 = -\sum_{i=1}^{n} p_{if(i)}$$

2. Для того чтобы решение нашей задачи не зависело от незапланированных изменений в расписании, необходимо обеспечить заданный временной интервал между занятиями стоянки , если это возможно, иначе назначать штраф за его несоблюдение. Определим как

$$z_2 = \sum_{\{(i,j)|i \neq j, f(i)=f(j)\}} \max\{t_0 - t_{ij}, 0\}$$

Тогда математическая формулировка данной задачи будет выглядеть следующим образом:

$$\min_{\{f|f:N \to M\}} g(f) = \min_{\{f|f:N \to M\}} (\alpha z_1(f) + (1 - \alpha) z_2(f))$$

с ограничениями (1 – 3).

Вес $0<\alpha<1$ определяет приоритет выполнения одной из целей (назначение наиболее выгодной стоянки или обеспечение безопасности и гибкости полученного расписания) и может определяться службами аэродрома или предпочтениями авиакомпаний.

Такая постановка позволяет гибко адаптировать задачу под различные критерии. Так, к примеру, если необходимо максимизировать использование контактных стоянок, достаточно определить выигрыш от назначения на такую стоянку намного большим, чем выигрыш от назначения на удалённые места стоянок.

Математическая модель задачи разделения графа на клики

Используется подход к задаче назначения мест стоянок как к задаче разделения графа на клики. Такой подход показал свою перспективность и простоту адаптации под конкретные условия задачи [3].

Рассмотрим полный неориентированный взвешенный граф

$$G = (V, E, W),\ V = \{1, 2, \dots, v\}, E \subseteq V \times V,\ W = (w_{ij}),\ i, j \in V,\ w_{ij} = w_{ji} \in \mathbb{R} \cup \{-\infty\}.$$

Здесь V – набор вершин графа, E – набор рёбер, W – симметричная матрица весов размера $v \times v$. Задача разделения графа на клики (*Clique Partition Problem, CPP*) – поиск такого разделения вершин на клики так, чтобы сумма весов всех рёбер между вершинами, входящими в клики, была максимальной.

Пусть элементы пространства решений

$$x_{ij} = \begin{cases} 1, \text{если } i \text{ и } j \text{ принадлежат одной клике} \\ 0, \text{иначе} \end{cases}$$

для всех рёбер из . Тогда CPP будет выглядеть следующим образом:

$$\max_{ij} Q = \max_{ij} \sum_{1 \le i < j \le v} w_{ij} x_{ij}$$

$$x_{ij} + x_{jk} - x_{ik} \le 1, \quad 1 \le i < j \le v \tag{4}$$

$$x_{ij} - x_{jk} + x_{ik} \le 1, \quad 1 \le i < j \le v \tag{5}$$

$$-x_{ij} + x_{jk} + x_{ik} \le 1, \quad 1 \le i < j \le v \tag{6}$$

$$x_{ij} \in \{0, 1\}\ \forall i, j:\ 1 \le i < j \le v$$

Ограничения (4–6) обеспечивают выполнение следующего требования: если вершины i и j, j и k входят в одну клику, то вершины i и k также входят в ту же клику.

Применение к задаче назначения стоянок. Пусть $v := n + m$ – число вершин, где n – число рейсов, m – число стоянок. Если вершина $i \le n$, соответствующая рейсу, находится в клике с вершиной $k > n$, соответствующей стоянке, то рейс i назначен на стоянку k. Если в одной клике находятся вершины, соответствующие рейсам, то они назначены на одну стоянку. Если в клике находится больше одной вершины, соответствующей стоянкам, то это означает, что данные стоянки являются «связанными». Необходимые ограничения будут гарантироваться выбором весов рёбер.

Добавим в граф дополнительную вершину – «фальшстоянку». Объединение в клику с такой вершиной будет означать неназначение рейсу стоянки.

$$V := \{1, 2, \dots, n + m + 1\},$$

$$w_{ij} = \begin{cases} -\infty, \text{ если } t_{ij} < 0 \\ (\alpha - 1) * \max\{t_0 - t_{ij}, 0\}, \text{если } t_{ij} \ge 0 \end{cases} \forall i, j \le n$$

$$w_{ij} = \begin{cases} -P, \text{ если } j \notin M(i) \\ \alpha * p_{ij}, \text{если } j \in M(i) \end{cases} \forall i \le n, j > n$$

$$w_{i(n+m+1)} = -P\ \forall i \le n$$

$$w_{ij} = -P, \forall i, j > n \ \vee i \notin D(j)$$

Здесь $P \gg p_{ij}\ \forall\ i, j$ – штраф за неназначение ВС или за нарушение ограничений (1)–(3).

Выбор и адаптация алгоритма

Задача назначения стоянок является NP-полной задачей с дискретным пространством решений. Для таких задач высокую эффективность показали стохастические алгоритмы локального поиска [4]. Для решения задачи был использован метод имитации отжига. Основным преимуществом этого метода является свойство избегать зацикливания в локальных оптимумах и продолжать поиск глобального оптимума. Для метода имитации отжига была показана эффективность его работы для большого класса задач.

Основной идеей алгоритма имитации отжига является использование убывающей функции температуры. При этом на каждой итерации алгоритма происходит генерация нового состояния из окрестности текущего решения. Переход в это состояние осуществляется с вероятностью $p(\Delta Q,T)$. Для адаптации метода к конкретной задаче необходимо выбрать следующие параметры: 1) исходное решение; 2) функцию убывания температуры $T=T(k)$ и начальное значение T_0; 3) функцию изменения состояния $G(x,T)$ (или порождающее семейство вероятностных распределений $G(x,T)$); 4) функцию вероятности перехода в новое состояние; 5) критерий остановки.

В качестве исходного решения берётся случайное допустимое состояние. Такое решение может быть получено очень быстро (максимальное число итераций), однако, как правило, оказывается далёким от оптимального.

Функция убывания температуры имеет огромное влияние на скорость и точность работы алгоритма. В данной работе использовались схемы: 1) геометрическая прогрессия $T(k)=k*\alpha*T_0$;; 2) модель отжига Коши $T(k)= T_0/k$.

При этом T_0 было выбрано равным P, что позволяется переходить на начальных этапах работы алгоритма в существенно худшие состояния. $Q(x)$ – значение целевой функции модели в состоянии x. $\Delta Q=Q(x^\rangle)-Q(x)$. Переход осуществляется с вероятностью

$$p(\Delta Q, T) = \exp\left(\frac{\Delta Q}{T}\right).$$

При таком значении вероятности гарантируется переход в лучшее состояние.

Для генерации нового состояния были рассмотрены два варианта функций перехода: 1) детерминированная: новое состояние находится как наилучшее из проколотой окрестности текущего решения, 2) стохастическая: новое состояние генерируется из проколотой окрестности текущего решения на основе порождающего семейства вероятностных распределений . В качестве таких распределений используются распределения Коши с плотностью

$$g(x';x,T) = \frac{T}{\pi(|x'-x|^2+T^2)}.$$

Для такой модели отжига была доказана сходимость при условии использования функции Коши изменения температуры [5, 6]. Алгоритм заканчивает свою работу в следующих случаях: 1) $T<\varepsilon$, при этом найдено допустимое решение; 2) найденное лучшее решение не менялось уже в течение *end_num* итераций. Числа *end_num* и ε определяются на основе статистики. При этом выбирается достаточно малым.

Таким образом, были разработаны следующие алгоритмы.

Алгоритм 1 (выбор наилучшей стоянки). На каждой итерации алгоритма для каждой вершины графа, соответствующей рейсу, находится наилучшее перемещение из текущей клики. При этом вершины, соответствующие конфликтным рейсам, перемещаются в клику с «фальшстоянкой». Переход в новое состояние принимается с вероятностью 1, если оно лучше предыдущего, либо с вероятностью $p(\Delta Q,T)$. После того, как перемещены все вершины, соответствующие рейсам, температура уменьшается в соответствии с первой схемой (геометрическая прогрессия) и происходит переход к следующей итерации.

Алгоритм 2 (выбор случайной стоянки). На каждой итерации алгоритма для каждой вершины, соответствующей рейсу, генерируется новое состояние на основе распределения Коши. При этом в первую очередь рассматриваются вершины, объединённые в клику с «фальшстоянкой». Переход в новое состояние осуществляется по тем же правилам, что и в варианте 1. После того, как перемещены все вершины, соответствующие рейсам, температура

уменьшается в соответствии со второй схемой (схема отжига Коши), и происходит переход к следующей итерации.

Полученные результаты

При проведении исследований использовался реальный полётный план по Российской Федерации за 01.06.2013. Эксперименты проводились на исследовательском стенде КИС УВД. На основе проведённых исследований были выбраны следующие значения параметров алгоритма: *P= 5000, end_num = 5, ε=1*. Были проведены эксперименты на имитационных моделях ряда аэродромов [2]. В таблице представлены сравнительные результаты работы алгоритма. Это средние значения, полученные по результатам серии из 20 экспериментов. Прочерк в столбце «Время работы» означает, что время работы алгоритма не превышало 1 секунды. Исследования проводились на ЭВМ с процессором Intel Xeon W3520, 2,67 ГГц и оперативной памятью 3,5 Гб под управлением операционной системой Microsoft Windows XP (32-разрядная).

Результаты работы алгоритмов 1 и 2 на моделях различных аэродромов

АП	Число рейсов	Число стоянок	Результат 1	Время работы 1	Результат 2	Время работы 2
Внуково	456	216	4555	2:26	1259	0:18
Домодедово	852	200	5019,25	17:10	4270	5:21
Елизово	18	9	72,5	-	60,5	-
Пулково	439	144	4388	1:27	1294,5	0:17
Сочи	77	38	768	-	224	-
Хабаровск	70	44	349,5	-	197,5	-
Шереметьево	793	255	3946,5	12:01	2211,5	0:53

Как видно из таблицы, алгоритм 2 работает гораздо быстрее и даёт на выходе допустимое решение, которое, однако, хуже, чем результат работы алгоритма 1.

Подводя итоги исследования работы предложенных вариантов алгоритма имитации отжига для задачи расстановки воздушных судов по местам стоянок на поверхности аэродрома, можно сделать следующие выводы:

– оба рассмотренных варианта алгоритма позволяют найти допустимое решение, если оно возможно;

– при моделировании маленьких аэродромов целесообразно пользоваться алгоритмом с выбором наилучшей стоянки, как дающий лучший результат;

– для больших аэродромов предлагается использовать схему с выбором наилучшей стоянки для формирования предварительного расписания на рассматриваемый промежуток времени;

– в случаях нарушения расписания в процессе моделирования целесообразно использовать алгоритм с выбором случайной стоянки как более быстрый.

Литература

1. **Gabeydulin R., Skavinskaya D., Orlov V.** Stand-loop simulation of Air Traffic control systems // 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), 2015, pp. 1F5-1 – 1F5-11.
2. **Вересов К.А., Топин В.А., Глаговский К.А.** Имитационная модель управляемого движения на поверхности аэродрома в составе стенда полунатурного моделирования системы ОРВД // Сборник докладов шестой всероссийской научно-практической конференции «Имитационное моделирование. Теория и практика» (ИММОД-2013). Т. 2. Казань, 2013. С. 68–71.
3. **Dorndorf U., Jaehn F., Pesch E.** Modelling Robust Flight-Gate Scheduling as a Clique Partitioning Problem // Transportation Science. 2008. Vol.42, No. 3.
4. **Пантелеев А.В.** Метаэвристические алгоритмы поиска глобального экстремума. МАИ, 2009
5. **Ingber L.** Very fast simulated re-annealing // Mathl. Comput.Modelling. 1989. Vol.12. P.967–973.
6. **Лопатин А.С.** Метод отжига // Стохастическая оптимизация в информатике. 2005. Вып.1. С.133–149.