

Aircraft landing problem

Москвитин Максим

Руководитель: Служаев Евгений, SimLabs

Составления расписания посадки самолетов

Актуальность:

- ▶ Многие крупные аэропорты перегружены
- ▶ Трафик растет приблизительно на 5% в год
- ▶ Справляться с этим ростом расширением аэропортов слишком дорого

Цели:

- ▶ Максимизация пропускной способности аэропорта
- ▶ Минимизация расходов на топливо
- ▶ Минимизация задержек относительно запланированного времени посадки

Постановка задачи

Дано множество из N самолетов.

Для каждого известны: E_i , T_i , L_i - минимальное, целевое, максимальное время посадки.

Между посадками самолетов должен быть выдержан временной интервал S_{ij} , определяемый классами турбулентности самолетов.

Задача - определить время посадки t_i каждого самолета, такое, что минимизируется некоторая функция штрафа:

$$Cost_{total} = \sum_{i=0}^N Cost_i(t_i)$$

Постановка задачи. Функция штрафа

$$Cost_i(t) = \begin{cases} \alpha_i(T_i - t_i) & : t_i \leq T_i \\ \beta_i(t_i - T_i) & : t_i > T_i \end{cases}$$

Адекватность такой модели:

- ▶ Расход топлива обратно пропорционален скорости
- ▶ Скорость ограничена снизу, для дальнейшего увеличения задержки нужно увеличивать расстояние, что также приводит к линейному увеличению расхода топлива
- ▶ Штраф за опоздания не определен, важна возможность его регулировать (кол-во пассажиров, приоритет самолета)
- ▶ Пропускная способность максимизируется естественным образом - в первую очередь решение должно удовлетворять всем ограничениям и приводить к посадке всех самолетов

Классические подходы к решению

- ▶ First-Come-First-Served - самолеты садятся в порядке прилета.
- ▶ Динамическое программирование по подмножествам самолетов. Приемлемое время работы достигается за счет ограничения максимального сдвига самолета в итоговой перестановке относительно FCFS.

Стохастические алгоритмы

Идея: для фиксированное перестановки самолетов составление расписания и его оценка не является NP-полной задачей.

- ▶ Генетические алгоритмы. Довольно долго являлись основным направлением решения задачи, но имеют сложность - отсутствие хорошего оператора скрещивания.
- ▶ Имитация отжига. Наш выбор.
 - ▶ Работаем с пространством перестановок
 - ▶ Оператор - swar двух случайных элементов

Составление расписания при фиксированном порядке

- ▶ Жадный подход: каждый самолет садится в минимально возможное время. Сложность $O(n)$. Не оптимальное решение (не учитывает расход топлива)
- ▶ Сведение к задаче линейного программирования. Точный, но не очень быстрый метод.

Составление расписания при фиксированном порядке

Утверждение. При переходе от решения для первых k самолетов к решению для первых $(k + 1)$:

- ▶ Не потребуется сдвиг на более позднее время.
- ▶ Временной интервал между парой соседних самолетов не увеличится

Используя эти два утверждения можно получить быстрое и точное решение.

$O(n \log(n))$ - при использовании декартовых деревьев.

$O(n(\log(n))^2)$ - при использовании очереди с приоритетами.

Результаты

N	FCFS	DP		BKR		SA	
	Cost	Cost	Time	Cost	Time	Cost	Time
10	1280	700	0	700	0	700	0
20	4890	3100	1.5	2520	2	2520	1
44	1550	1550	150	1550	3	1550	8
100	16876			6092	11.5	6175.47	42
250	42836			16209	219	16775	229
500	92810			41448	567	39790	717

- ▶ Получили решение на уровне публикаций последних лет
- ▶ Простота отжига компенсируется скоростью оценки перестановки, позволяющей увеличить число итераций
- ▶ Реализация предложенного алгоритма составления расписания имеет потенциал для дальнейшей оптимизации

Итоги

- ▶ Порешали NP-полную задачу
- ▶ +1 проект на C++
- ▶ Только RAII
- ▶ boost