Agendamento de Pousos de Aeronaves com Simulated Annealing

Eduardo Henke

Contexto do Problema

- Ao entrar no alcance do radar do controle de tráfego aéreo (ATC): o avião precisa ser atribuído um **horário de pouso** dentro de uma janela de tempo específica.
- **Objetivo:** Garantir pousos seguros mantendo a separação entre aeronaves e minimizando os custos relacionados ao desvio do horário de pouso preferido.

Restrições

- **Janela de Tempo:**
  - Cada avião i tem um horário de pouso mais cedo E_i e um mais tarde L_i.
    - E_i: Horário mais cedo se o avião voar à velocidade máxima.
    - L_i: Horário mais tarde, considerando a eficiência de combustível e o tempo máximo de espera.
    - T_i: Horário de pouso preferido.
- **Tempo de Separação:**
  - Cada par de aviões (i, j) deve manter um tempo mínimo de separação S_{ij} entre seus pousos:
    - Exemplo: Um Boeing 747 precisa de mais tempo de separação em relação a um avião menor, por conta da turbulência gerada.

Função Objetivo - Minimizar o Custo Total

- O custo é incorrido quando o avião pousa antes ou depois do horário alvo T_i:
  - Pouso antes de T_i gera um custo, pouso depois também.
  - O objetivo é minimizar a soma dos custos para todos os aviões.

Exemplo

Objetivo: Determinar os horários de pouso x_i para cada avião i de forma que:

1. $x_i \in [E_i, L_i]$, ou seja, o avião pouse dentro da janela de tempo.
2. A condição de separação $x_j \geq x_i + S_{ij}$ seja respeitada para cada par de aviões (i, j) subsequentes.

- **Exemplo Simples:**
  - O avião A tem uma janela de tempo de 10h às 10h30.
  - O avião B tem uma janela de 10h20 às 11h.
  - Se o tempo de separação for 10 minutos, o avião A deve pousar antes de B, com uma separação mínima de 10 minutos.
  - Exemplo: `[(A, 10h05), (B, 10h20)]`; `[(B, 10h20), (A, 10h30)]`

Representação da solução

- Representação da solução, é uma lista ordenada de pousos, onde um pouso é uma tupla `(ID Avião, Horário de Pouso)`.
- A estrutura de vizinhança então será alterar o horário de um pouso e reordenar a lista baseado nesse novo horário.

Estratégias de Busca

- **Randômica:** selecionar um pouso aleatório e alterar o seu horário aleatoriamente dentro da janela de tempo, não respeitando o tempo de separação.

```
procedure random_neighbor(solution) {
  i = random.choice(solution.len())
  solution[i].time = random.uniform(solution[i].earliest, solution[i].latest)
  return solution
}
```

- **First Improvement:** itera aleatoriamente sobre todos os pousos e altera o seu horário de uma forma "zig-zag" dentro da janela de tempo.
  - Pouso i com horário x_i, tenta horários $x_i; x_i - 1; x_i + 1; x_i - 2; x_i + 2; ...; E_i; L_i$.

```
procedure first_improvement_neighbor(solution) {
  for arrival in shuffle(solution) {
    for time in zigzag_range(arrival.earliest, arrival.latest, arrival.time) {
      if changing arrival with time results in a better solution {
        return changed_solution
      }
    }
  }
}
```

Implementação - Loop Principal

- **Solução Construtiva Inicial**: criar uma lista de pousos com os horários T_i de cada avião $\rightarrow$ terá vários conflitos.
- Repetir até chegar no tempo limite:
  - **Simulated Annealing**: aplicar SA na solução construtiva inicial.
  - **Busca First Improvement**: aplicar FI na solução dada pelo SA.

Implementação - Função Objetivo

Permitimos soluções inválidas, porém com penalidade para conflitos.

```
procedure cost(solution) {
  landing_cost(solution) + conflict_cost(solution)
}

procedure landing_cost(solution) {
  sum(arrival.plane.cost_for_landing(arrival.landing_time) for arrival in solution)
}

procedure conflict_cost(solution) {
  sum(CONFLICT_PENALTY * conflict.duration for conflict in solution.conflicts())
}
```

IRace

Projeto foi configurado com o IRace para realizar o *tuning* dos parâmetros para o SA, especialmente:

```
sa_max_k      ""  r (0.1, 8.0)
alpha         ""  r (0.8, 1.0)
initial_temp  ""  r (100.0, 1000000.0)
```

O resultado foi o seguinte:

```
   sa_max_k  alpha initial_temp
45   3.8272 0.9809   10599.1518
24   5.7757 0.9565  219812.3305
```

> IRace pode rodar o programa em até $10 * n^2$ iterações.

TABLE I

COMPUTATIONAL RESULTS

			Heuristic		Tree Search								
Problem Number	Number of Planes	Number of Runways	Value	Time (secs)	Number of Restarts	Solution Value at Each Restart	Time at Each Restart (secs)	Number of Variables at Final Restart	Number of Constraints at Final Restart	LP Value at Final Restart	Optimal Value	Number of Tree Nodes	Total Time (secs)
1	10	1	optimal	0.1	—	—	—	68	173	321.16	700	49	0.4
		2	optimal	0.1	—	—	—	90	158	0	90	91	0.6
		3	optimal	0.1	—	—	—	—	—	—	0	—	—
2	15	1	1500	0.1	1	1480	1.9	157	412	430.00	1480	454	5.2
		2	optimal	0.1	—	—	—	196	386	0	210	115	1.8
		3	optimal	0.1	—	—	—	—	—	—	0	—	—
3	20	1	1380	0.1	1	820	0.6	194	525	449.40	820	42	2.7
		2	optimal	0.1	—	—	—	277	489	0	60	142	3.8
		3	optimal	0.2	—	—	—	—	—	—	0	—	—
4	20	1	optimal	0.1	—	—	—	336	929	924.37	2520	20002	220.4
		2	optimal	0.1	—	—	—	389	870	0	640	193319	1919.9
		3	optimal	0.1	—	—	—	345	837	0	130	39901	2299.2
		4	optimal	0.2	—	—	—	—	—	—	0	—	—
5	20	1	5420	0.1	—	—	—	426	1181	964.83	3100	50745	922.0
		2	1070	0.1	—	—	—	452	1076	0	650	282160	11510.4
		3	240	0.1	—	—	—	361	890	0	170	20035	1655.3
		4	optimal	0.2	—	—	—	—	—	—	0	—	—
6	30	1	optimal	0.1	—	—	—	80	222	5393.25	24442	10806	33.1
		2	882	0.1	—	—	—	630	1266	0	554	25316	1568.1
		3	optimal	0.2	—	—	—	—	—	—	0	—	—
7	44	1	optimal	0.1	—	—	—	174	341	184.00	1550	2192	10.6
		2	optimal	0.2	—	—	—	—	—	—	0	—	—
8	50	1	2690	0.4	7	2480; 2285; 2245; 2080; 2040; 1990; 1950	5.4; 11.3; 16.8; 24.3; 29.9; 41.4; 51.2	1318	3785	1547.86	1950	1114	111.9
		2	255	0.2	—	—	—	1562	3154	0	135	9020	3450.6
		3	optimal	0.6	—	—	—	—	—	—	0	—	—

Resultados obtidos com >1s de execução

#	n	Optimal	Paper	Mine
1	10	700	700	700
2	15	1480	1500	1480
3	20	820	1380	820
4	20	2520	2520	2820
5	20	3100	5420	3950
6	30	24442	24442	~~20054~~*
7	44	1550	1550	1588

Extra

Gráficos de melhoria de performance:

```
file:///Users/henke/ufop/metaheuristics/airplane-landing-scheduler/img/flamegraph.svg
```

```
samply record ./target/release/airplane-landing-scheduler eval-one 3.8272 0.9809 10599.1518 data/airland4.txt 2.0
```

```
file:///Users/henke/ufop/metaheuristics/airplane-landing-scheduler/perf-cmp-1/report/index.html
```

Obrigado!