EÖTVÖS LORÁND UNIVERSITY

Pilot Training Optimization for Airlines

Réka Hábel

Applied Mathematics MSc

Supervisor: Alpár Jüttner, senior researcher
Eötvös Loránd University
Department of Operations Research

Master Thesis

Budapest, 2013

Contents

Chapter 1

Introduction

Nowadays, using various planning and operating tools and software is essential for the efficient operation of an airline. The range of these so-called decision support systems is quite wide. The whole planning process, including timetable construction, fleet assignment, manpower planning, crew scheduling, load planning, flight planning etc. is supported by numerous well-developed software. Besides these systems are required to be reliable, they should provide optimal or nearly optimal plans in as short time as possible. If any disruption occurs, like bad weather, equipment failure etc. there is not much time to recalculate, a new optimal or nearly optimal solution must be provided as soon as possible.

Airlines are interested in applying these products because optimal utilization of the resources is essential for the cost-effective operation. Adapting supply to the changing passenger demand, predicting the number of pilots needed in different seats, determining the timing of trainings and vacations are some of the serious problems facing major airlines. Overcoming these difficulties occur in a complex planning process, called *manpower planning*. The goal of manpower planning is to ”provide the right number of the right personnel at the right time at minimum cost”. This thesis focuses on manpower planning, especially on pilot training optimization.

There is a huge attention towards pilot trainings because most of the accidents in the past could have been avoided if the pilots had been better-trained. Flight planning and flight control systems are getting more and more reliable, therefore, most of the time accidents happen because of human mistakes. In case any unexpected disruption occurs, pilots co-operating with the air traffic controllers must exactly know how to act. They have to be prepared to any arising unusual, sometimes dangerous situation.

The deadliest aviation accident in the past few years was the disaster of Air France Flight 447. On 1 June 2009, an Airbus A330 crashed into the Atlantic Ocean, killing all 216 passengers and 12 air crew on board. Investigations of the accident continued for the forthcoming three years. The circumstances of the crash made the research very difficult. The flight recorder was found much later, on 2 May 2011. Until then, there were only speculations - based on the already known facts - regarding the cause of the accident. The plane hit some turbulence, the autopilot disconnected and the plane started to roll to the right. The pilot tried to correct this by deflecting the side-stick to the left and made an abrupt nose-up. Then the plane nearly stalled two times and tossed in the air. The airspeed indicator showed unreliable values and the pilot continued making nose-up. Unfortunately the crew didn't manage to coordinate the aeroplane and it started to descend rapidly and moments later, hit the ocean.

The question was, among other failures, was there any human error involved? According to the Final Report of the French Civil Aviation Safety Investigation Authority [13], released in July 2012, the aeroplane was flying close to the upper limit of its envelope and neither of the crew members knew exactly how to act. The absence of any training at high altitude in manual aircraft handling contributed to the disaster. Some changes after the accident referred to crew training, for example flight simulator trainings were extended to high altitudes and to land without airspeed indicators.

That's why we put the emphasis on pilot training optimization in this thesis. The more optimal schedule is created, the more savings can be achieved, which leads to the possibility of training more pilots with more advanced training methods.

Manual solutions to the occurrent manpower planning problems are time consuming and not so cost-effective. Manpower planners gather data from several database systems and build training plans using spreadsheets. Optimal or close to optimal solutions can be reached in minutes by using mathematical models and well-designed algorithms. As soon as the concept of decision support systems (DSS) emerged in the 1980s, the need for applications of DSS's started growing. Verbeek designed a system to support strategic manpower planning of airline pilots at Royal Dutch Airlines (KLM) [2]. Another US carrier, Continental Airlines, has been using an integrated decision support system for more than 10 years, called Crew ResourceSolver to construct training plans. By using the system, Continental has savings around $10 million annually [3]. Another example is Lufthansa Technical Training GmbH (LTT), that runs training courses for Lufthansa Technik AG and for other international airlines. LTT increased its profit by using a course-scheduling decision support system. The overall profit of the schedule created by the system is about 26 percent higher than the profit of the manually constructed one [4]. Åsa Holm did similar work to Jeppesen (subsidiary of Boeing Commercial Aviation Services). She explored various areas of manpower planning in her thesis and built up mathematical models. She revealed that by using optimization methodology the potential savings are around 10 percent [1].

This thesis proposes promising solution techniques to the pilot training problem with mathematical approach. The referenced models were tested on generated - however realistic - data, optimal solutions could be reached in surprisingly short time.

Chapter 2

Business background

An airline constantly adjusts its need for pilots in different seats in response to new market opportunities, changing passenger demand etc. Shortages in some positions may be filled by transferring surplus pilots from other seats. If shortages are greater than surpluses, the airline will need new hires. Conversely, if surpluses are greater, crew will be released. Therefore, pilots change position many times during their career. They want to move from smaller to larger crafts and from *first officer* seat to *captain* seat to increase their salary and responsibility. In the cockpit there are usually two positions, captain and first officer. The captain is ultimately responsible for the operation and the safety of the flight. The first officer is the second pilot (also known as co-pilot). The control of the aircraft is shared between the captain and the first officer. Optionally there is a third person in the cockpit, a flight engineer in order to monitor the engines and other flight systems.

The way that crew members move between positions is different for each airline. One option is the traditional career ladder, when pilots change from first officer to captain at the same type of aircraft before they change to a bigger aircraft. An other alternative is when they change from first officer on one aircraft to first officer on a bigger aircraft and when they reach the top, they change to captain on the smallest one. Some airlines managed to put positions in parallel, thus the number of possible transitions was increased.

The process of deciding which pilots to be transferred from one position to another differs between carriers. This action is taken one or two times a year. The most common is *system bid award.* During system bid award, the airline offers positions to the pilots, based on forecasted needs and then they are given the opportunity to bid on these new openings they desire. Each pilot has a *seniority number*, based on the length of service within the airline, with incidental reductions. The airline then awards positions to the pilots in seniority order. We can break the order, but this is associated with cost called *pay-protection.* It means that if two pilots are awarded the same new position and the less senior pilot is advanced first, then the airline pays the more senior pilot the increased amount from the advancement of the junior pilot.

After a bidding period, pilots can be separated into four groups:

- Pilots, who are going to be transferred without training. They are already qualified for their new position.
- Pilots to be advanced after training.
- There may be pilots to be released including pilots over age 60 who should be furloughed.
- Pilots maintaining their current position.

After deciding the new positions a planning phase begins. Each newly awarded position is associated with several trainings. A training plan must be constructed that determines the timing of trainings, advancements, releases and new hires. The plan also includes detailed information about all training events for each student. When creating such a plan there are several restrictions to be considered.

- Pilots, who are awarded a new position must be trained within the required period. Promotions and releases must be carried out before a pre-defined deadline.

- Seniority rules restrict the order of promotions and releases. Pilots must be released in reverse-seniority order and advancements should be performed in seniority order. The order can be broken but it involves pay-protection.

- Pilots may have longer predetermined activities, such as vacations which must be considered when creating the schedule.

- We define *block-hours* as the number of flight hours to be staffed at each position during a given period. It is measured by the time between an aircraft is leaving the departure gate and arriving at the destination gate. Trainings must be planned in a way that it is ensured that pilots in service can satisfy the demand. If it is not possible, block-hour shortage should be minimized, especially as it is associated with high cost.

- The number of available resources is limited. In the optimal schedule the utilization of simulators, instructors and other devices is efficient.

- Pilots are assigned to different trainings based on their work experience and their newly awarded position. Each training consists of various courses. A schedule must be made so that each pilot is assigned to the required courses.

Chapter 3

Determining the timing of transitions

Because of the large size of the problem, it is solved with a two-phase process. At first, the timing of transitions is determined with only monthly precision. In the second phase a detailed assignment is created. In this chapter, the focus is on the first stage. Only limited information is available of the resources and only estimation of block-hours is known. A solution is obtained by solving a *mixed integer program (MIP)*. A schedule in monthly periods is created, which minimizes the costs (including pay-protection and expenses of block-hour shortage) while maintaining practicability.

There exist previous works regarding pilot training optimization. Yu et al. [3] built up a basic mixed integer program in order to solve pilot-transitioning. However their referred model does not consider some requirements, e.g. the order of trainings. Åsa Holm examined the problem in details and also created a MIP. In addition to the trainings she put a great emphasis on vacation planning. The MIP shown below is based on their models and adapted to the current problem. The output is of the form that the second phase requires.

3.1 Input parameters of the problem in discussion

A 6-month period in the winter season is considered because during that half of the year less flights operate, trainings can be scheduled to the pilots' agenda.

- T = time periods, in our case $T = \{1, 2, 3, 4, 5, 6\}$

Pilots under consideration are those who need to be trained, released or advanced without any training.

- I = list of pilots
- I_a = sublist of pilots to be advanced both after and without training
- I_t = sublist of pilots who need at least one training
- I_r = sublist of pilots to be released in reverse-seniority order
- $I(h)$ = pilots whose initial position is h
- $I'(h)$ = pilots whose future position is h
- IP = pairs of pilots (i, j) where pilot j may pay-protect pilot i

A position can be described with a triad of base, rank and aircraft type.

- H = set of all positions

Each training comprises various courses, which must be completed by pilots assigned to the training. There are different types of trainings, each training can be described by the courses involved. Only one-period-long trainings are considered. If a training is more than one-period-long, it is divided into several one-period-long segments. An extra constraint is added in order to ensure the correct order.

- K = set of all training types

- KP = pairs of trainings (k, l) where k must precede l
- $I(k)$ = sublist of pilots, who need training type k
- $K(i)$ = required trainings for pilot i

Each course requires resources, such as simulators, instructors etc. which are limited.

- R = set of all the necessary resources
- $K(r)$ = trainings which use resource r

Number of pilots who can start a specified training at any given time is maximized. The initial supply of each resource is given, for example in a given period the number of accesses of the simulators is determined.

- m_k = maximum number of participants in training type k
- rs_{rt} = initial supply of resource r in period t
- d_{kr} = required number of resource r for training k

Based on the relevant planned flight schedule, the required number of block-hours per position can be calculated. It is the number of hours to be staffed at each position in a given period. The number of block-hours which can be covered by the current crew can be calculated by multiplying the number of qualified pilots by the average number of flight hours in a given period (in our case in a certain month). Pilots in training are not productive, these missing flight hours must be subtracted from the initial supply of block-hours.

- bd_{ht} = demand for block-hours for position h in period t
- bi_{ht} = initial supply of block-hours for position h in period t
- u_{ht} = utilization of a pilot at position h in period t
- ul_k = utilization loss due to participate in training type k

A plan should be created, which minimizes the incurring costs. Expenses to be considered are:

- cc_k = price of holding training type k
- ca_{it} = cost of advancing pilot i in period t
- cr_{it} = cost of releasing pilot i in period t
- cp_i = pay-protection cost for pilot i
- cs_h = expenditures resulting from block-hour shortage at position h

3.2 Mixed integer program for the planning

If all the necessary information is available, then the planning process can be started. In this section, a mixed integer programming formulation of the problem is proposed. A plan can be constructed from the values of the decision variables:

$$ya_{it} = \begin{cases} 1 & \text{if pilot } i \text{ is advanced in period } t \\ 0 & \text{otherwise} \end{cases}$$

ya_{it} is defined only for pilots to be advanced, $i \in I_a$

$$yr_{it} = \begin{cases} 1 & \text{if pilot } i \text{ is released in period } t \\ 0 & \text{otherwise} \end{cases}$$

yr_{it} is defined only for pilots to be released or furloughed, $i \in I_r$
Advancements and releases happen at the beginning of the period.

$$y_{ikt} = \begin{cases} 1 & \text{if pilot } i \text{ starts training type } k \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

y_{ikt} is defined only for pilots who need training type k, $i \in I(k)$ and for periods not conflicting with long absences.

Further variables are added in order to track the number of ongoing trainings, block-hour shortage and pay-protection.

$$\begin{aligned}
x_{kt} &= \text{number of pilots starting training } k \text{ in period } t \\
z_{kt} &= \text{number of training type } k \text{ starting in period } t \\
s_{ht} &= \text{number of block-hours short for position } h \text{ in period } t \\
p_i &= \text{number of periods in pay-protection for pilot } i
\end{aligned}$$

The goal is to minimize the incurring costs.
Objective function:

$$\text{minimize} \sum_{i \in I_a} \sum_{t \in T} ca_{it} ya_{it} + \sum_{i \in I_r} \sum_{t \in T} cr_{it} yr_{it} + \sum_{k \in K} \sum_{t \in T} cc_k z_{kt} + \sum_{i \in I_a} cp_i p_i + \sum_{h \in H} \sum_{t \in T} cs_h s_{ht}$$

Subject to:
Assignments and releases must happen exactly once.

$$\sum_{t \in T} ya_{it} = 1 \qquad \forall i \in I_a$$

$$\sum_{t \in T} yr_{it} = 1 \qquad \forall i \in I_r$$

A pilot can start his/her assigned training exactly once and if he/she has more trainings, he/she can start at most one training at a given period.

$$\sum_{t \in T} y_{ikt} = 1 \qquad \forall k \in K, \forall i \in I(k)$$

$$\sum_{k \in K(i)} y_{ikt} + ya_{it} \leq 1 \qquad \forall i \in I_t, \forall t \in T$$

Tracking the number of trainings and participants.

$$x_{kt} = \sum_{i \in I(k)} y_{ikt} \qquad \forall k \in K, t \in T$$

$$\frac{x_{kt}}{m_k} \leq z_{kt} \qquad \forall k \in K, t \in T$$

A pilot is advanced after completing the required trainings.

$$\sum_{t=1}^{n} y_{ikt} - \sum_{t=1}^{n} ya_{it} \geq 0 \qquad \forall i \in I_t, n = 1..|T|, \forall k \in K(i)$$

Pilots must be released in reverse-seniority order.

$$\sum_{t=1}^{n} yr_{it} - \sum_{t=1}^{n} yr_{i-1t} \geq 0 \qquad \forall i \in I_r, n = 1..|T|$$

Some trainings have strict order which can't be broken.

$$\sum_{t=1}^{n} y_{ikt} - \sum_{t=1}^{n} y_{ilt} \geq 0 \qquad \forall (k,l) \in KP, n = 1..|T|, \forall i \in I(k) \cap I(l)$$

Capacity of the resources is limited.

$$rs_{rt} \geq \sum_{k \in K(r)} d_{kr} z_{kt} \qquad \forall t \in T, r \in R$$

Tracking the number of periods in pay-protection for each pilot.

$$p_j \geq \sum_{t \in T} t * ya_{it} - \sum_{t \in T} t * ya_{jt} \qquad \forall (i,j) \in IP$$

Tracking the number of block-hour shortages.

$$s_{ht} \geq bd_{ht} - (bi_{ht} -$$

$$(\sum_{i \in I(h) \cap I_a} \sum_{p=1}^{t} u_{ht} ya_{ip} + \sum_{i \in I(h) \cap I_r} \sum_{p=1}^{t} u_{ht} yr_{ip} + \sum_{i \in I(h) \cap I_t} \sum_{k \in K(i)} ul_k y_{ikt})$$

$$+ \sum_{i \in I'(h)} \sum_{p=1}^{t} u_{ht} ya_{ip}) \qquad \forall h \in H, t \in T$$

Restrictions to the variables:

$$ya_{it}, yr_{it}, y_{ikt} \in \{0,1\}$$

$$x_{kt}, z_{kt}, p_i \in \mathbb{Z}_+ \cup \{0\}$$

$$s_{ht} \in \mathbb{R}_+ \cup \{0\}$$

Chapter 4

Course scheduling

In the previous phase the timing of trainings are determined by solving a *mixed integer program*. In the course scheduling phase, pilots are already assigned to several trainings and trainings are already assigned to months. Each training consists of various courses which have to be scheduled in consideration of some limitations, such as working time regulations. In this phase a detailed schedule is constructed by solving an *integer program (IP)*.

Class scheduling is one of the most popular areas studied by operations researchers. To the school timetabling problem there exist numerous algorithms. A wide variety of methods are available, the problem can be formulated as graph coloring or can be solved with network flows. Genetic algorithms and tabu search also can be applied. Further techniques can be found in a technical survey of Schaerf [6]. Despite the fact, that it is a well-researched area, the existing algorithms can't be adapted to the course scheduling problem of pilot training. There are significant differences between school timetabling and the problem under consideration, e.g. we are planning for one period, courses are not repeated week by week.

Specific algorithms were also constructed but not in such large numbers. Haase et al. [4] developed a prototype course-scheduling module however the details of realization remained a secret. Qi et al. made a research on class scheduling for pilot training at the training center of Continental Airlines [5].

They proved that the problem is NP-hard in the strong sense. The idea of the proof is to reformulate as a known NP-hard problem, the 3-PARTITION. Then a branch and bound solution technique is proposed. Our solution is different from the existing ones, we use a widespread column generation approach.

4.1 Problem description

Consider a specific training. This training is defined by its courses. Each course requires some of the following elements: classroom, instructor, flight training devices, simulator etc. Most critical resources are flight training devices (FTDs) and full flight simulators (FFSs). Their complexity comes from the fact, that most of the expenses are associated with FTDs and FFSs and the number of available devices is very limited. Classrooms and instructors are not significant, therefore in this thesis we put the emphasis on the perfect utilization of FTDs and FFSs. In order to create a practicable schedule the appropriate number of resources must be ready. Each course is denoted by its related resource:

- T indicates that FTDs are needed
- S indicates that FFSs are is needed
- N indicates that neither an FTD nor an FFS is needed.

Using the previous notation a training can be described by a sequence consisting of {N, S, T}. Each course is divided into 1-day-long events. For example a basic template of a specific training is of the form

$$[N, N, N, N, N, N, T, T, T, S, S, N, N]$$

which means that the training consists of a 6-day-long classroom training, a 3-day-long course using FTDs, a 2-day-long course using FFSs and a 2-day-long classroom training again.

There is an other restriction to take in account. After a predetermined number of working days pilots are entitled to have *days off*. Various restrictions exist regarding days off such as contractual obligations and governmental regulations. In this thesis the following restrictions are supposed:

- After 7 consecutive days at least one day off must be given
- Length of days off is at most 3 days

Days off are given due to various obligations or strategic decisions. If there is a lack of resources, the pilots involved are allowed to days off, the course is shifted. Pilots in days off are paid, however they are not productive. The goal is, that only barely enough days off are given. For example days off can be added to the previous basic template in many ways to create a legal training event:

$$[N, N, N, N, N, N, X, X, T, T, T, X, S, S, N, N]$$
$$[N, N, N, N, X, N, N, T, T, T, X, S, S, N, N]$$

The difference between templates including days off is their length. The longer the training the more expensive it is. The length of a template is called its *footprint*. The number of pilots in the class multiplied by its footprint is the *weighted footprint*. The objective of the scheduling problem in question is to minimize the weighted footprints of all training events.

4.2 Input parameters of the course scheduling problem

Since trainings are already assigned to months and each training is aircraft specific, the scheduling can be performed per month and per aircraft type independently.

The planning horizon and the time units are predefined, in our case 30 days in average.

- τ = time periods, in our case $\tau = \{1, 2, ..., 30\}$

Suppose that n trainings are given to schedule, each training has a basic template from which we can generate various legal training schedules by adding days off. At first it is supposed, that all possible templates are known, including all possible start days. Using the previous example, if we start the training on the 5th day, the template looks like that:

$$[X, X, X, X, N, N, N, N, N, N, X, X, T, T, T, X, S, S, N, N, X, X, ..., X, X]$$

The following templates belong to the same training, however all of them are different.

$$[4X, N, N, N, N, N, N, X, X, T, T, T, X, S, S, N, N, 10X]$$
$$[10X, N, N, N, N, N, N, X, X, T, T, T, X, S, S, N, N, 4X]$$
$$[6X, N, N, N, N, X, N, N, T, T, T, X, S, S, N, N, 8X]$$
$$[8X, N, N, N, N, X, N, N, T, T, T, X, S, S, N, N, 6X]$$

- T_i = set of all possible templates of training i
- $T = \cup_{i=1}^{n} T_i$

Each course uses either FTDs or FFS or neither of them. The required amount of the devices is given.

- d_{ij}^{T} = number of FTDs needed for template j on day i
- d_{ij}^{S} = number of FFSs needed for template j on day i

The number of available resources is limited and determined in advance.

- r_i^T = number of available FTDs on day i
- r_i^S = number of available FFSs on day i

A further parameter is introduced in order to determine which training a template belongs to.

$$\delta_{ij} = \begin{cases} 1 & \text{if template } j \text{ is related to course } i \\ 0 & \text{otherwise} \end{cases}$$

The cost of a template is its *weighted footprint*. (The length of the training multiplied by the number of participants.)

- c_j = cost of template j

4.3 Integer program for the scheduling problem

Previously we assumed, that set T of all of the legal templates of the current trainings is known. It is have to be decided which templates to use. Decision variable of the program for all $j \in T$:

$$x_j = \begin{cases} 1 & \text{if template } j \text{ is in use} \\ 0 & \text{otherwise} \end{cases}$$

The integer programming formulation:

Objective function:

$$\text{minimize} \sum_{j \in T} c_j x_j$$

Subject to:

Resource requirements must be satisfied.

$$\sum_{j \in T} d_{ij}^{T} x_j \leq r_i^{T} \qquad \forall i \in \tau$$

$$\sum_{j \in T} d_{ij}^{S} x_j \leq r_i^{S} \qquad \forall i \in \tau$$

Each training must have exactly one schedule.

$$\sum_{j \in T} \delta_{ij} x_j = 1 \qquad \forall i = 1..n$$

Restriction to the variable:

$$x_j \in \{0,1\} \qquad \forall j \in T$$

4.4 Column generation approach

The solution obtained from the integer program corresponds to an optimal legal schedule of all trainings. Templates where the value of x_j is 1 are used. However there is a problem with solving the integer program. Size of set T of all templates is enormous because the number of all possible templates is exponential. Generating set T and solving the integer program using T takes exponential time, a solution may not be achieved.

4.4.1 Basic idea of column generation

Column generation is often used to solve problems with huge number of variables. The general problem P is of the form:

$$\begin{gathered} \text{maximize } c(x) \\ Ax \leq b \\ x \in S \\ x \text{ integer} \end{gathered}$$

Suppose that S is bounded. The case when S is unbounded and further details can be seen in [7]. The basic idea is to represent the set of the integer solutions

$$S^* = \{x \in S : x \text{ integer}\}$$

by a finite set of vectors

$$S^* = \{y_1, ..., y_p\}$$

If x is binary, S^* coincides with the extreme points of its convex hull. Any feasible point $y \in S$ can be represented as a convex combination of the vectors in S^*:

$$y = \sum_{k=1}^{p} \lambda_k y_k$$

Convexity constraints:

$$\sum_{k=1}^{p} \lambda_k = 1$$

$$\lambda_k \in \{0, 1\} \qquad k = 1..p$$

Let denote $c_k = c(y_k)$ and $a_k = Ay_k$. The column generation form of P is called *master problem (MP)* and is given by:

$$\text{maximize} \sum_{k=1}^{p} \lambda_k c_k$$

$$\sum_{k=1}^{p} \lambda_k a_k \leq b$$

$$\sum_{k=1}^{p} \lambda_k = 1$$

$$\lambda_k \in \{0, 1\} \qquad k = 1..p$$

Any solution to the linear programming relaxation of P is a feasible solution to the linear programming relaxation of the column generation form if and only if it can be represented by a convex combination of extreme points of $conv(S^*)$.

The column generation form frequently contains a huge number of columns. The idea is to work with only a subset of columns and then generate additional columns if needed. The restricted version is called *restricted master problem (RMP)*. Let (π, α) be an optimal dual solution to the RMP. The pricing problem is to identify a column with maximum reduced cost.

$$\max_{x \in S^*} c(x) - \pi Ax - \alpha$$

If the maximum is greater than 0, a column to enter the basis is identified, the augmented RMP is solved again. If the maximum is less than or equal to 0, the current solution to RMP is also optimal for MP.

If the LP relaxation is solved by column generation, the solution is not necessarily integer. In order to achieve integrality *branch and bound* or rounding techniques can be applied.

4.4.2 Linear programming relaxation and the dual linear program

The master problem (MP) is the linear programming (LP) relaxation of the IP (as a maximization problem):
Objective function:

$$\text{maximize} \sum_{j \in T} -c_j x_j$$

Subject to:

$$\sum_{j \in T} d_{ij}^T x_j \leq r_i^T \qquad \forall i \in \tau$$

$$\sum_{j \in T} d_{ij}^S x_j \leq r_i^S \qquad \forall i \in \tau$$

$$\sum_{j \in T} \delta_{ij} x_j = 1 \qquad \forall i = 1..n$$

Restriction to the variable:

$$0 \leq x_j \leq 1 \qquad \forall j \in T$$

The dual linear program:
Objective function:

$$\text{minimize} \sum_{i=1}^{|\tau|} y_i^T r_i^T + \sum_{i=1}^{|\tau|} y_i^S r_i^S + \sum_{i=1}^{n} \pi_i$$

Subject to:

$$\sum_{i=1}^{|\tau|} y_i^T d_{ij}^T + \sum_{i=1}^{|\tau|} y_i^S d_{ij}^S + \sum_{i=1}^{n} \pi_i \delta_{ij} \geq -c_j \qquad \forall j \in T$$

Restriction to the variables:

$$\begin{aligned} 0 \leq y_i^T \qquad & \forall i \in \tau \\ 0 \leq y_i^S \qquad & \forall i \in \tau \\ \pi_i \in \mathbb{R} \qquad & \forall i = 1..n \end{aligned}$$

At first consider a subset T_0 of possible templates including a feasible (not necessarily optimal) schedule for the n trainings. The reduced variant (RMP) of the LP relaxation is constructed by keeping only columns related to T_0, thus we have a smaller problem. Then we find an x solution to the RMP.

$$\text{maximize} \sum_{j \in T_0} -c_j x_j$$

$$\begin{aligned} \sum_{j \in T_0} d_{ij}^T x_j \leq r_i^T \qquad & \forall i \in \tau \\ \sum_{j \in T_0} d_{ij}^S x_j \leq r_i^S \qquad & \forall i \in \tau \\ \sum_{j \in T_0} \delta_{ij} x_j = 1 \qquad & \forall i = 1..n \\ 0 \leq x_j \leq 1 \qquad & \forall j \in T_0 \end{aligned}$$

The corresponding dual variables are y_i^T $(i \in \tau)$, y_i^S $(i \in \tau)$, π_i $(i = 1..n)$. It should be checked whether the primal optimum of the reduced problem with additional zeros is an optimum of the original one or not. If it is, an optimal schedule can be constructed. According to the *duality theorem*, it should be checked whether the dual inequalities hold for all templates. If not, we add a violator column to the RMP and iterate until the optimum is found or a certain limit is reached.

The goal is to find a column which violates if exists. Search for an r column that

$$\sum_{i=1}^{|\tau|} y_i^T d_{ir}^T + \sum_{i=1}^{|\tau|} y_i^S d_{ir}^S + \sum_{i=1}^{n} \pi_i \delta_{ir} < -c_r$$

Equivalently search for a column where the negative *reduced cost* is minimal.

$$\min_{r=1..|T|} \sum_{i=1}^{|\tau|} y_i^T d_{ir}^T + \sum_{i=1}^{|\tau|} y_i^S d_{ir}^S + \sum_{i=1}^{n} \pi_i \delta_{ir} + c_r$$

If the minimum is negative, column r - where the minimum is reached - is added to the reduced problem.

4.4.3 Integer solution

By solving the LP relaxation of the problem, the solution is not necessarily integer, although a binary solution is required. There exist several rounding methods in order to obtain an integer solution.

The naive way is to round each entry of the solution of the LP relaxation, but it may not be optimal (or even feasible) to the IP. A better process is to round the value of only one variable, fix its integer value, then solve the reduced LP again and so on. The way of selecting the variable to round can vary. A similar method to the one that is applied in [9] can be used.

Evaluate each of the K largest primal variables and compute the increase of the cost function when the variable is set to 1 and all others related to the same training is set to 0. The one resulting in the least increment is chosen and the corresponding template is fixed. Then the LP relaxation is solved again without the training which the template relates to and the process is repeated until all of the trainings have a fix template.

Rounding methods are heuristics, but can be useful in practice.

4.4.4 Column generation sub-problem

By solving the following minimization problem it can be decided, whether there exists a violator column or the primal solution is an optimum for the MP.

$$\min_{r=1..|T|} \sum_{i=1}^{|\tau|} y_i^T d_{ir}^T + \sum_{i=1}^{|\tau|} y_i^S d_{ir}^S + \sum_{i=1}^{n} \pi_i \delta_{ir} + c_r$$

An algorithm is presented to find the minimum and the corresponding column. The minimization problem can be formulated as finding a constrained shortest path in a directed graph.

An auxiliary digraph is constructed $G = (V, E)$. There is a START node with n outgoing edges. Each edge refers to a training. For each training there is a subgraph G_{t_i} connected by an edge to node END. The skeleton of the graph can be seen on Figure 4.4.1.

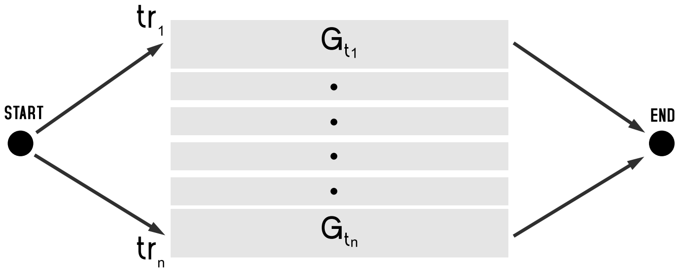

Figure 4.4.1: Skeleton of the auxiliary digraph

Each subgraph is constructed by the basic template of the training it refers to. Consider a specific training. Courses requiring different resources alternate. Each level of the subgraph corresponds to a course type. For example if the basic template of the current training is of the form

$$[N, N, N, N, N, N, T, T, T, S, S, N, N]$$

the subgraph has 4 levels. On each level there are as many nodes as time periods, currently 30. Node n_{ij} is the j-th node on level i. There are two additional nodes, a START and an END. There are 4 types of edges:

- From START to all nodes on the first level
- From a node to all of the following nodes on the same level
- From a node to all of the following nodes on the next level
- To END from all nodes on the last level

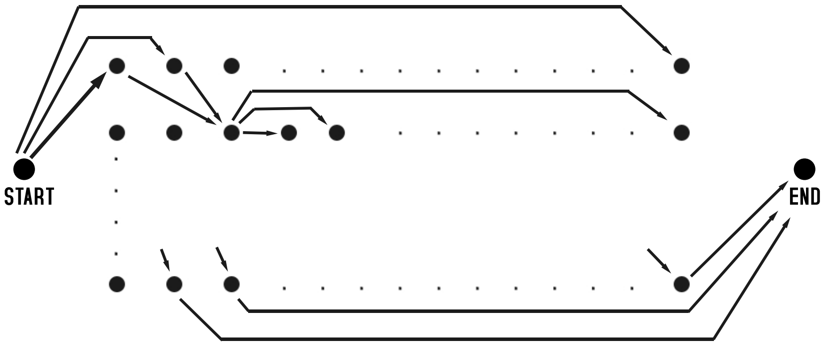

Figure 4.4.2: Sketch of a subgraph

The sketch of a subgraph can be seen on Figure 4.4.2.

We are searching for a shortest path from START to END that satisfies some conditions (see details later). Consider a feasible path, which template does it define. If the target of the arc coming from START is the j-th node on the first level, then the training starts on day j. Along the path if there is a (n_{ij}, n_{kl}) arc, a corresponding course to level k is hold on day l.

Return back to the example. If the shortest path has the form like on Figure 4.4.3, then the corresponding template is

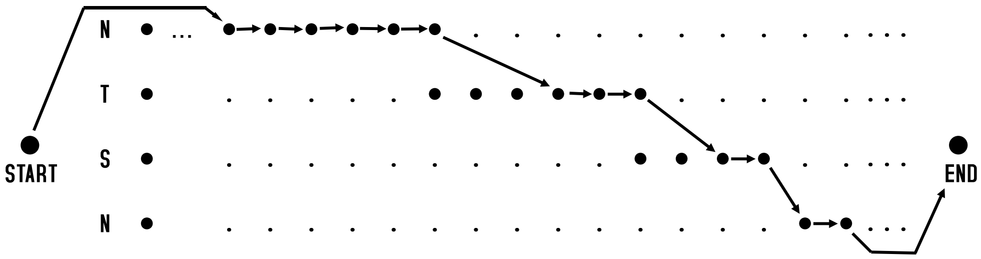

Figure 4.4.3: A possible shortest path

$$[4X, N, N, N, N, N, N, X, X, T, T, T, X, S, S, N, N, 10X]$$

There are various constraints concerning the shortest path.

- At each level the required number of nodes must be reached (It is the number of consecutive courses of the relevant type)
- After 7 consecutive days at least one day off must be given
- Length of days off is at most 3 days

The last constraint can be satisfied by the construction of the subgraph. We only consider edges from a node to the following 4 nodes. "Longer" edges can be eliminated.

Notations:

- Subgraph t refers to training t $(t = 1..n)$
- l_t is the number of levels of subgraph t
- p_t is the number of participants on training t
- s_t is the number of courses on training t
- dt^T_{qt} is the required number of resource T on course q
(0, if it is not necessary)
- dt^S_{qt} is the required number of resource S on course q
(0, if it is not necessary)

Each node has cost, $c : V \to \mathbb{R}$, coming from the dual solution of the reduced LP. The cost on the global START and END nodes is 0.

$$c(globalSTART) = 0 \qquad c(globalEND) = 0$$

Cost on the START and END node of subgraph t is:

$$c(START_t) = \pi_t \qquad c(END_t) = 0$$

On an internal n_{ij} node, $i = 1..l_t$, $j = 1..|\tau|$, its cost depends on the sequential number of the node on a specific path. If n_{ij} is the q-th internal node along path P,

$$c_P(n_{ij}) = y_j^T dt_{qt}^T + y_j^S dt_{qt}^S$$

that is,

$$c_P(n_{ij}) = \begin{cases} y_j^T dt_{qt}^T & \text{if } n_{ij} \text{ refers to a course using an FTD} \\ y_j^S dt_{qt}^S & \text{if } n_{ij} \text{ refers to a course using an FFS} \\ 0 & \text{otherwise} \end{cases}$$

Arcs are also weighted. $c' : A \rightarrow \mathbb{R}$

$$c'(e) = \begin{cases} (l-j) * p_t & \text{if } e = (n_{ij}, n_{kl}) \\ 1 * p_t & \text{if source of } e \text{ is START} \\ 0 & \text{if target of } e \text{ is END} \end{cases}$$

Using this weighting, cost of a given P path is the following:

$$c(P) = \sum_{n \in \text{ nodes of } P} c_P(n) + \sum_{e \in \text{ arcs of } P} c'(e)$$

With equivalent transformations:

$$\sum_{n \in \text{ nodes of } P} c_P(n) = c(START_t) + c(END_t) + \sum_{i=1}^{l_k} \sum_{j=1}^{|\tau|} c_P(n_{ij}) =$$

$$\pi_t + 0 + \sum_{q=1}^{s_t} c_P(n_q | n \text{ is the } q\text{-th internal node on path } P) =$$

$$\pi_t + \sum_{q=1}^{s_t} (y_j^T dt_{qt}^T + y_j^S dt_{qt}^S)$$

Denote r the template defined by path P.
Define d_{jr}^T as:

$$d_{jr}^T = \begin{cases} dt_{qt}^T & \text{if node } n_{ij} \in P \text{ is the } q\text{-th along P and refers to a course using an FTD} \\ 0 & \text{otherwise} \end{cases}$$

Define d_{jr}^{S} as:

$$d_{jr}^{S} = \begin{cases} dt_{qt}^{S} & \text{if node } n_{ij} \in P \text{ is the } q\text{-th along P and refers to a course using an FFS} \\ 0 & \text{otherwise} \end{cases}$$

With the previous notation:

$$\sum_{q=1}^{s_t} (y_j^T dt_{qt}^T + y_j^S dt_{qt}^S) = \sum_{j=1}^{|\tau|} y_j^T d_{jr}^T + \sum_{j=1}^{|\tau|} y_j^S d_{jr}^S$$

That is:

$$\sum_{n \in \text{ nodes of } P} c_P(n) = \sum_{j=1}^{|\tau|} y_j^T d_{jr}^T + \sum_{j=1}^{|\tau|} y_j^S d_{jr}^S + \sum_{i=1}^{n} \pi_i \delta_{ir}$$

With equivalent transformations:

$$\sum_{e \in \text{ arcs of } P} c'(e) = c'(START_t, n_{1f}) + \sum_{(n_{ij}, n_{kl}) \in \text{ arcs of } P} c(n_{ij}, n_{kl}) + c'(n_{l_t g}, END_t) =$$
$$1 * p_t + p_t * (l - j) + 0 = \text{weighted footprint of template } r = c_r$$

Thus, cost of a given P path in subgraph t - related to training t - can be calculated as

$$c(P) = \sum_{j=1}^{|\tau|} y_j^T d_{jr}^T + \sum_{j=1}^{|\tau|} y_j^S d_{jr}^S + \sum_{i=1}^{n} \pi_i \delta_{ir} + c_r$$

Hence finding a feasible shortest path in the auxiliary digraph from START to END is equivalent to the original minimization problem

$$\min\{c(P) | \text{ P is a feasbile path from START to END}\} =$$
$$\min_{r=1..|T|} \sum_{i=1}^{|\tau|} y_i^T d_{ir}^T + \sum_{i=1}^{|\tau|} y_i^S d_{ir}^S + \sum_{i=1}^{n} \pi_i \delta_{ir} + c_r$$

4.4.5 Dynamic program for SCP

During the algorithm for creating an optimal schedule we have to find a shortest constrained path in a well-constructed digraph. At each iteration a

SCP is found in each subgraph and then the minimum is chosen. We propose a dynamic programming method in order to find one.

$G = (V, E)$ acyclic digraph. Cost of a P path is:

$$c(P) = \sum_{n \in \text{ nodes of } P} c_P(n) + \sum_{e \in \text{ arcs of } P} c'(e)$$

The idea of dynamic programming is to determine a set of potential paths to each node. Nodes are examined in topologic order until END is reached. The solution is the path which ends in node END with minimal cost.

Let P_{ij} denote the set of paths to n_{ij}. Suppose, that the algorithm is at node n_{kl}. Paths to the previous nodes are already calculated. An arc is coming from n_{ij} to n_{kl}, further potential paths can be added to P_{kl}:

$$P_{kl} \longleftarrow \{p + (n_{ij}, n_{kl}) \quad \forall p \in P_{ij}\}$$

Paths which are invalid or more costly, than another comparable path can be eliminated from P_{kl}.
A path is invalid if:

- node n_{kl} would be the $m+1$-th node on the path on the k-th level while only m is needed
- node n_{kl} would be the first node on the path on the k-th level while on the $k-1$-th level the number of nodes is not enough.
- node n_{kl} would be the 8-th course without a day-off

Path S is better than T if:

- n_{kl} is the r-th node on level k both in S and in T
- number of consecutive days without a day off in S is less than or equal to in T
- $c(S) \leq c(T)$

The algorithm iterates through the nodes from START to END and the solution is the path which ends in node END with minimal cost.

Chapter 5

Test Cases

For the implementation we chose the C++ programming language with a template library, called LEMON. LEMON is a C++ template library providing efficient implementations of algorithms connected with graphs and optimization. It was launched by the Egerváry Research Group on Combinatorial Optimization (EGRES) at the Department of Operations Research, Eötvös Loránd University. In order to solve (mixed) integer programs we used the GLPK solver.

Since we had no opportunity to test on real data, we worked with generated, but realistic data.

5.1 Determining the timing of transitions

An illustrative example.

Consider 21 pilots who have new positions. 15 of them need several trainings, 6 can be advanced without any trainings. There are 4 additional pilots to be released. Changes affect 4 positions. Some pilots must complete several trainings, here we consider 6 types of trainings with precedences. Resource constraints are added and participants are limited to 2 or 3 on each training. The largest expenses are associated with block-hour shortage. Costs on advancing and releasing increase over time. To this input an optimal schedule is obtained by solving the previously presented mixed integer program within minutes. The output can be seen on Table 5.1 and on Table 5.2. Advances are after trainings and happen as soon as possible in order to minimize costs while maintaining resource availability. In this example block-hour shortage exist, but minimal.

We also tested with larger (but realistic) amount of data. 50 pilots are involved, 12 training types are required and 10 positions are available. A plan can be constructed within minutes if resource availability is set to large. The tighter the resource availability is, the more time it takes to solve. Even if the supply of resources is less, a solution can be obtained in 15 minutes.

	t0	t1	t2	t3	t4	t5
p0	c0	adv	0	0	0	0
p1	c0	adv	0	0	0	0
p2	c0	adv	0	0	0	0
p3	0	0	c0	adv	0	0
p4	c0	0	c1	0	0	adv
p5	c0	c1	adv	0	0	0
p6	c0	c1	adv	0	0	0
p7	0	c2	c3	adv	0	0
p8	0	c2	c3	adv	0	0
p9	0	c2	c3	adv	0	0
p10	0	c2	c3	adv	0	0
p11	0	0	0	c4	c5	adv
p12	0	0	0	c4	c5	adv
p13	c4	c5	adv	0	0	0
p14	c4	c5	adv	0	0	0

Table 5.1: Pilots' planned schedule

p15	adv	0	0	0	0	0
p16	adv	0	0	0	0	0
p17	adv	0	0	0	0	0
p18	adv	0	0	0	0	0
p19	adv	0	0	0	0	0
p20	adv	0	0	0	0	0
p21	0	0	0	0	0	rel
p22	0	0	0	0	0	rel
p23	0	0	0	0	0	rel
p24	0	0	0	0	0	rel

Table 5.2: Pilots' planned advancement or release

5.2 Course scheduling

An illustrative example.

Consider 4 trainings to be scheduled for 10 days. Basic templates are the following:

- $[NTS]$
- $[NNTSS]$
- $[NTTSS]$
- $[NNTTSSNS]$

To start with, T' includes the following not at all optimal templates.

- $[NXXXXTXXSX]$ $[XNXXXTXSXX]$
- $[XNXNXXTXSS]$ $[XXNXNXTXSS]$ $[XXNNXTXSXS]$
- $[NTXXXTSSXX]$
- $[NNXTTSSXNS]$

Then the LP relaxation is solved with T'. With the dual variables, we solve an SCP problem to generate new template. If its cost is negative it is added to T' and iterate, if not, the solution is optimal. On Table 5.3 a running of the program is shown. After 7 iteration, there is no violator template, an optimal schedule can be found by solving the LP relaxation. The process took only seconds. Fortunately, the solution is binary, no rounding process needed. The optimal schedule is the following:

- $[XXXXXXXNTS]$
- $[XXXXNNTSSX]$
- $[XXXNTTSSXX]$
- $[XNNTTSSNXS]$

y^T	y^S	π	new template	its cost
00...00	00...00	-35 -48 -56 -80	$[XXXXXNTTSS]$	-21
00...00	00...00 21	-35 -69 -56 -101	$[XXXXNNTSSX]$	-39
010...00	00...00 10 00	-45 -40 -35 -80	$[XXXXXXXNTS]$	-30
00...00	00...00 20	-35 -30 -55 -100	$[XNNTTSSNXS]$	-30
00...00	00...00 20	-35 -30 -55 -92	$[XXXXXXNTSX]$	-20
00...00	00...00 10.5 10.5	-25.5 -40.5 -56 -82.5	$[XXXNTTSSXX]$	-21
00...00	00...00	-15 -30 -35 -72		0

Table 5.3: A running of the program

We tested with larger (but realistic) amount of trainings. Scheduling 20 trainings to 30 days took only seconds, however the number of resources affect the number of iterations. During testing all of the solutions were integer but it isn't expected. This triggers further investigation whether it can be guaranteed or just most of the time.

Chapter 6

Conclusion

Pilot training optimization is a complex problem, due to the various constraints concerning the construction of a plan. In this thesis, we presented practically usable solution methods to the general problem. The proposed models are flexible, which means that more constraints can be added or unnecessary assumptions can be removed to solve airline-specific cases.

A 2-phase process were proposed as the problem consists of a planning part and a scheduling part. A training plan with minimum cost could be obtained by solving a mixed integer program. In the second phase the integer programming formulation of the scheduling problem were described. It couldn't be solved directly because of the large number of variables. Column generation method was applied, the sub-problem to solve was finding a shortest constrained path. This sub-problem was solved by dynamic programming. Using the idea of column generation, a schedule with minimal cost could be constructed.

Acknowledgements

At first I'd like to thank Lufthansa Systems the opportunity to make a research in manpower planning. Special thanks to Gyula Magyar, who made it possible for us to look into the planning process.

I'd like to thank Alpár Jüttner, who introduced me the practical problem-solving by mathematical methods. He helped me a lot with his technical expertise in formulating and solving the pilot training problem.

Thanks to my boyfriend for the programming support, I learnt a lot in this area with his help.

I dedicate this thesis to my father, who endeared me to mathematics. Boldog szülinapot Apa!

Bibliography

[1] Åsa Holm, *Manpower Planning in Airlines- Modeling and Optimization* Master thesis, Linköpings Universitet 2008

[2] Verbeek, P.J. *Decision Support Systems: An Application in Strategic Manpower Planning of Airline Pilots*
European Journal of Operational Research, Vol. 55, pages 368–381

[3] Yu, G., Pachon, J., Thengvall, B., Chandler, D. and Wilson, A. *Optimizing Pilot Planning and Training for Continental Airlines*
Interfaces, Vol. 34, pages 253–264.

[4] Haase, K., Latteier, J., Schirmer, A. *Course Planning at Lufthansa Technical Training: Constructing More Profitable Schedules*
Interfaces, Vol. 29, pages 95-109.

[5] Qi, X., Bard, J.F, Yu, G. *Class Scheduling for Pilot Training*
Operations Research, Vol. 52, pages 148–162

[6] Schaerf, A. *A Survey of Automated Timetabling*
Technical report, CWI (Centre for Mathematics and Computer Science) Amsterdam

[7] Barnhart, C., Johnson, E.L, Nemhauser, G.L., Savelsbergh, M.W.P, Vance, P.H. *Branch-and-price: Column generation for solving huge integer programs*
Operations Research, Vol. 46 no. 3, pages 316–329

[8] Borndörfer, R., Schelten, U., Schlechte, T., Weider S. *A Column Generation Approach to Airline Crew Scheduling*
Operations Research Proceedings, Vol 2005, 2006, pages 343-348

[9] Dezső B., Jüttner, A., Kovács, P., Ladányi, Á. *Contact Center Staff Scheduling with Various Constraints*

[10] Ahuja, R.K., Magnanti, T.L., Orlin, J.B. *Network Flows*

[11] Qi, X., Yang, J., Yu, G. *Scheduling Problems in the Airline Industry*
Handbook of Scheduling, Chapter 50

[12] Barnhart, C., Belobaba, P., Odoni, A. R. *Applications of Operations Research in the Air Transport Industry*
Transportation Science, Vol. 37, pages 368-391;

[13] *Final Report - On the accident on 1st June 2009 to the Airbus A330*
Bureau d'Enquêtes et d'Analyses (BEA)

[14] http://lemon.cs.elte.hu

[15] http://en.wikipedia.org