Crew Management Information Systems

Diego Klabjan
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL (d-klabjan@northwestern.edu)
Yu-Ching Lee
Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL (ylee77@uiuc.edu)
Goran Stojković
Jeppesen, Englewood, CO (Goran.Stojkovic@jeppesen.com)

One of the early successes of cost cutting decision support systems in the airline industry was in crew management. Due to high crew costs incurred by the airlines, deploying efficient crew management information systems became a necessity to thrive. For decades, research and development groups have been developing sophisticated algorithms and solution methodologies for solving complex crew management problems. We overview early developments and recent state-of-the-art algorithms. These algorithms form optimization engines behind modern crew management information systems. Common components of commercial systems are detailed and we discuss key software vendors.

1 Introduction

The airline industry is notorious for large swings between profitability and losses, following closely economic cycles. While economic cycles are getting less pronounced, to the contrary, profitability and losses of the airline industry are increasing in magnitude. These up and down swings were particularly observed in the last decade following the dot com bubble. On top of the recession in 2000, the airlines were hit hard by the terrorist events of September eleven. This was a defying moment in the industry.

Many large U.S. carriers have drastically cut their costs through changes in business processes and renegotiating labor contracts (some through bankruptcy protection while others barely stayed out of it). Standard flight banks at hub airports were de-peaked into a more spread out schedule. Pricing was substantially simplified to become more appealing to customers. Important cost reductions were obtained through labor renegotiations. Pilots are the highest paid unionized employees[1] and were the first target of such negotiations. Crew management information systems, including the underlying decision support, played an important role since they provide both parties with an accurate and prompt assessment of negotiated rules. Such systems became indis-

[1] http://www.bls.gov/oco/cg/cgs016.htm

pensable. In the past they were mostly used for traditional planning, but today they bear a much higher importance. Unions are becoming a more respected partner in an airline's decision making process. At selected airlines the pilot unions are even involved in the selection process of components of crew management information systems[2].

Crew cost remains the second largest expenditure behind fuel cost and thus the airlines spend significant resources for maintaining low crew cost. Manual decision making in the crew scheduling process such as constructing pairings or crew itineraries and later in the timeline rosters or bidlines has been mostly replaced by sophisticated and automated computer based management information systems. Due to the complex nature of the airline business, it is a very challenging and daunting task to produce satisfactory decision support systems for constructing pairings and other entities. Solution methodologies for this type of problems have started several decades ago and they keep evolving since then. Academicians, software vendors, and research and development groups within airlines frequently produce new methodologies to satisfy the ever growing needs. With network capacity expansions and fleet consolidations, the problem sizes keep increasing. On the other hand, to further decrease the overall cost, the airlines started having a more holistic view of their entire operations and processes. The interest for systems that consider the crew cost beyond its historical boundary of crew is growing, e.g., taking into consideration the impact of crew scheduling to aircraft routings and fleeting. Such calls also require new models and the underlying solution methodologies with a substantial crew component.

This chapter has two main sections, each one drilling into the aforementioned aspects. We start by describing basic models and the underlying state-of-the-art algorithms and solution methodologies. In the second part, we discuss the underlying information systems and the software arena. We finish the introduction with an overview of the various business processes employed by many airlines.

1.1 Business Processes

While every airline has its own processes and organizational names, most of the airlines follow the depicted processes and terminology outlined in Figure 1. Long term fleet and manpower planning consists of making strategic decisions with respect to the number of aircraft and the fleet decomposition, and cockpit crew manpower planning.

In fleet planning, considerations such as the airline's mission (e.g., Southwest has a single fleet that allows relatively simple and efficient operations), aircraft utilization, route structure, cargo/passenger mix, etc., are taken into account. Long term crew decisions revolve around the adequate staffing level in each fleet and requirements of crew training resources such as instructors, training facilities, and simulators.

[2] Panel at AGIFORS Crew Management Study Group, New York City (2003): Preferential Bidding Task Force; Brett Wilkie, Manager Crew Planning, IT, US Airways and Charles Mayer, ALPA

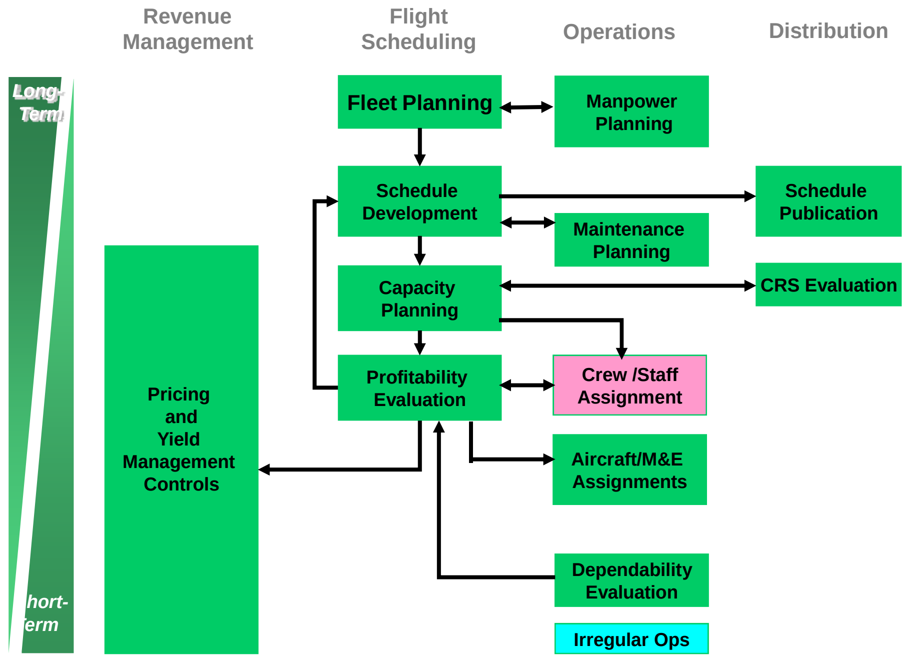

Figure 1: Typical Business Processes (M&E=maintenance and engineering, CRS=customer reservation system)

The schedule development phase typically starts 12 months before the day of operations and it lasts up to 9 months. In the first phase the airline establishes the service plan, which is the set of flights to operate in a given market. The service plan is either daily for domestic operations or weekly for international, long-haul service. The marketing group considers several factors such as traffic forecasts, status of competing carriers, internal resources, and marketing initiatives. Marketing initiatives are approved by upper management and involve decisions such as entering a new market. The designed service plan typically does not divert substantially from the current schedule. Following the service plan, the scheduling group generates a detailed flight schedule, i.e., a flight departure and arrival time. The flight schedule has to obey a set of operating constraints, e.g., maintenance planning, and given generic resources such as the number of aircraft. The schedule is then published. Next is capacity planning or fleeting. In fleeting an equipment type is assigned to each flight subject to available resources such as the number of aircraft. The goal of the fleet assignment model is to maximize profit. The schedule together with the seat capacity is then input to the computer reservation system. The produced fleeting solution is also evaluated with a profitability evaluation model and potential improvements are fed back to the schedule development group for possible minor adjustments. Once the equipment types are assigned, aircraft routing and crew scheduling follow. In aircraft routing, called also maintenance routing, a specific tail number or aircraft is assigned to each flight subject to maintenance constraints. The objectives are usually incentives such as through revenue and robustness. The goal of crew scheduling is to assign crew members to individual flights in order to minimize the crew cost and maximize various objectives related to contractual obligations, quality of life, and crew satisfaction. Crew schedules have to satisfy complex regulatory and contractual rules. Potentially, crew planners detect unfavorable connections and provide feedback to schedule and fleet planners. The crew scheduling process typically starts three months before the day of operation and it is constantly updated until a few weeks before the day of operations.

Only minor changes to fleeting, aircraft routes and crew schedules are made during the last few weeks before the day of operations. If preferential bidding is used, approximately one month before the day of operations, crews bid for their monthly crew assignments and only minor changes such as two-way trip swaps are performed in the last few weeks. An alternative form of creating monthly work schedules of an individual crew member is rostering, where rosters are constructed simultaneously for all crew members by considering individual preferences, regulatory rules, and the direct cost.

Throughout the strategic planning processes, pricing and yield or revenue management are actively involved. In revenue management, the airline controls the seat inventory by adjusting fare prices, setting overbooking limits, and making decisions at any given time about selling particular fare classes on a given passenger itinerary.

The actual day of operations scheduling, called also execution management, consists of making final minor adjustments to the flight schedule (e.g., adjust arrival times based on the daily wind forecast), executing the pre-planned schedule (e.g., file the flight plan) and rescheduling for irregular operations or disruption management. The latter is carried out by operations controllers, which are typically located in the airline operations control center. Most frequent sources of irregular operations are weather, unscheduled maintenance, congestion, crew unavailability, security problems, etc. Disruption management is composed of three processes. When an irregular operation occurs, first the aircraft are rerouted, which is called aircraft recovery. In this stage in addition to rerouting the aircraft, decisions on delaying and canceling flights are made. Next is the crew recovery process, where crews are assigned new crew itineraries. The controllers can use original, standby (crews available at the airport but not on an active duty), and reserve crews (crews on-call but performing off-work activities such as being at home). At the end is the passenger reaccommodation process, where passengers are rerouted to alternative itineraries. Clearly the new schedule must conform to all regulatory and contractual rules. While the airlines often impose more stringent rules in planning, in operations they typically use precise rules. Contractual rules for operations are usually different from those in planning.

The basic crew processes on the timeline are summarized in Figure 2.

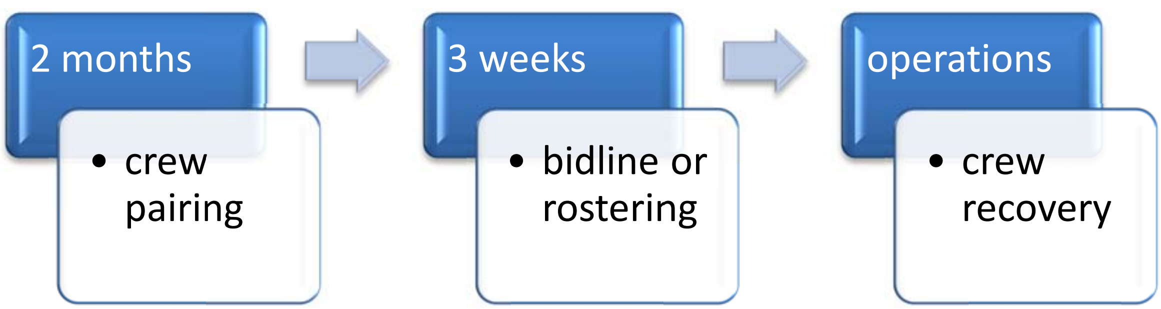

Figure 2: Main Crew Processes

2 Optimization in Crew Management

Limited and costly crew resources make optimization within an airline's system crucial. From the crew standpoint, a fleet is a group of aircraft that can be operated by the same qualified cockpit crew. On short-haul flights, modern aircraft must typically carry two cockpit crew members: a captain and a first officer. Some older aircraft must also carry a flight engineer.

As pointed out earlier, in a typical planning process, the fleet or capacity assignment problem is performed before any crew assignments. Once a fleet is assigned to each flight – mainly with the objective of satisfying market demand – crews are then scheduled. The cockpit crew problems are fleet specific, i.e., a pilot can only be assigned to a certain fleet that he or she is qualified to fly. Compared to the cockpit crew, the cabin crews are more flexible because they are cross-fleet trained. Due to their large numbers and wider range of qualified flights, the cabin crew management problems are much more elaborate; however, their total cost represents a lower portion of overall expenses.

The crew assignment problems have always defied the capabilities of our algorithmic knowledge and software science. They had and still do define state-of-the-art. The first difficulty results from the sheer size and nature of airline operations. A large carrier employs hundreds of costly crew members and it operates more than a thousand flights per day. The second challenge comes from complicated work-load rules governing crew operations. Statutory rules are defined, for example, by the Federal Aviation Administration (FAA) in the U.S., to ensure flying safety. On top of these rules, an airline usually imposes its own rules for crew management and operations. Some of them result from crew union requirements, while others are imposed to provide additional operational safety and flexibility. These additional rules are generally more restricted, yet they can be violated in emergency or exceptional cases in operations (but not in tactical planning), while the statutory rules of the government agencies do not have such flexibilities. The work-load rules are multidimensional. To explain the complexities of these rules, we start by introducing basic concepts of duty periods and pairings or crew itineraries.

A duty period is formed by a succession of flights and the required *sit time*s between two consecutive flights. Duty periods form the basis of pairings, which are sequences of duty periods with a required overnight rest or *layover* between two consecutive duty periods. A multi-duty pairing typically spans several days. A crew member departs from his or her home base airport or domicile to start the pairing, and later comes back to the same home base after several successive duty periods to complete the pairing. Each pairing is followed in a monthly assignment by a rest period of several days.

Several work-load rules pertain to each one of the following entities: flight, duty period, or pairing. For example, there are rules for the maximum numbers of duties in a pairing, the minimum time of rest between two consecutive duties, the maximum number of flying in a duty, and the maximum elapsed time of a duty and pairing. Some rules are expressed by the flying and rest time. For example, the complicated 8-in-24 rule imposed in the U.S. states that whenever there are 8 continuous hours of flying time within a 24-hour period, extra rest time called the compensatory rest is required.

The cost of a duty and pairing is traced to the contractual salary obligations. In the U.S., crew members are not paid by a fixed monthly salary; but instead, are paid with respect to the total flying time, total elapsed time, and the minimum guaranteed pay. The crew cost is usually measured in minutes. The following expression is often used to calculate the crew cost associated with duty period d:

$$b_d = max\left\{ f_d \cdot elapsed, fly, guarantee \right\},$$

where

b_d : the cost of duty period d in minutes,

f_d : a constant between 0 and 1 to give a discount to the total elapsed time of d,

$elapse$: total duty elapsed time,

fly : total flying time, and

$guarantee$: a minimum guarantee per duty in minutes.

Similarly, the crew cost associated with pairing *p* is usually computed as follows:

$$c_p = max\left\{ f_p \cdot TAFB,\ n \cdot guarantee,\ \sum_{d \in p} b_d \right\} + \sum_{\substack{\hat{d} \in p, \bar{d} \in p \\ \hat{d} \to \bar{d}}} e\left(\hat{d},\ \bar{d}\right),$$

where

c_p : the cost of pairing *p* in minutes,

f_p : a constant between 0 and 1 to give a discount to the total elapsed time of pairing *p*,

$TAFB$: time-away-from-base (TAFB) time or the total elapsed time of the pairing,

n : numbers of the duty periods in pairing *p*,

$d, \hat{d}, \bar{d}$: duty periods comprised in pairing *p*; notation $\hat{d} \to \bar{d}$ represents that duty period $\hat{d}$ is followed by duty period $\bar{d}$, and

$e\left(\hat{d}, \bar{d}\right)$: extra cost for the layover between $\hat{d}$ and $\bar{d}$, e.g., per diem and lodging expenses.

An efficient crew management system should not only aim to fulfill the work-load rules, but also assign crew members in an economical way to avoid unnecessary expenses. Because of the relatively high payment of crews, especially the cockpit crews, an efficient crew management system can significantly increase the profits of an airline. On the other hand, well designed scheduling requirements and suitable work load can also increase the welfare of the employees.

2.1 Models

In this section, prevailing mathematical models applied in crew management are surveyed. Based on their functional requirements within the business processes, the models are categorized into strategic, tactical, and operational or execution models. Strategic and tactical planning models, as the names suggest, are for long to medium term planning to optimize the use of all resources under a predicted most likely situation. They are usually employed from a few months to a few weeks before the day of the operation. On the other hand, operational planning is short term planning mostly resulting from irregular events (the cancellation or delay of flights). Operational planning is to smooth out the schedule on the day of operation, to handle contingencies, and to recover the schedule. The main concept behind operational planning is not to additionally disrupt the original schedule too much, but to quickly return back to the original tactical plan. Optimization is particularly challenging in this case since the solutions need to be provided instantaneously, i.e., in real time.

2.1.1 Strategic and Tactical Planning

The strategic and tactical planning regarding crew management is comprised of the crew pairing and crew assignment problems. In a typical crew management process, the pairing problem is solved first and then the assignment follows. The segregation of these two stages makes the problems easier to solve even though it might produce suboptimal solutions. Both crew pairing and crew assignment model formulations are of the set partitioning type. In this type of models, each element in the set must appear in one, and only one, subset. The model formulations are similar, yet the goals and the scope of problems do not overlap. We next introduce the crew pairing model and two types of crew assignment models: rostering and preferential bidding.

Crew Pairing

In the crew pairing problem, each flight has to be covered by a specific pairing. The process of selecting duty periods to form a pairing is referred to as pairing generation. The complicated work-load rules and the pay structure yield many constraints and restrictions on pairings. Due to the excessive number of all pairings, it is impractical to enumerate all of them. One way to overcome this difficulty is to only enumerate selected pairings rather than all of them. The methodologies for generation are further discussed later.

The crew pairing problem differs slightly in the U.S. from a typical problem in the rest of the world. The flights in the U.S. are usually flown every day, except on weekends or selected special days. Therefore, the pairing problem is solved first for the *daily* problem under the assumption that all flights are operated every day. In the solution, some pairings contain flights, which are not operated on some days. These are referred to as *broken* pairings. To mend the broken pairings, the *weekly* exceptions problem is solved. In this step, the flights which are operated less than or equal to four days per week are covered together with all flights from broken pairings. At the end of this process, flights in the entire weekly cycle are assigned exactly to one pairing.

Outside of the U.S. there is a larger variability of flights from day to day. As a result, the above two stage process is not amenable. In such a situation, the entire weekly problem including all of the flights in a week needs to be solved at once.

In addition, airline companies tend to modify the flight schedule every season. Whenever a new flight schedule is constructed, a transition pairing problem between the two cycles of flight schedules is required. This problem is called the *dated* problem because each flight in such a problem is associated with a particular date.

Sometimes the weekly exceptions problem is infeasible. In this case it is allowed to send the crews as passengers on additional flights, usually to allow them to go back to their crew base for later dispatch. This action is referred to as *deadheading*.

The objective of the crew pairing problem is to minimize the pay-and-credit, which captures the cost above the cost accounting for the flying time. The model formulation is as follows:

$$\begin{aligned} \min \ & \sum_{p \in P} c_p y_p & & \\ & \sum_{p: i \in p} y_p = 1 & & i \in F \\ & y_p \in \{0,1\} & & p \in P \end{aligned} \tag{1}$$

where

P : the set of all possible pairings,

F : the set of all flights,

c_P : the cost of pairing p, and

$i \in p$: represents that flight i is included in pairing p.

The decision variables are

$$y_P = \begin{cases} 1 & \text{if pairing } p \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

The only constraints present impose that each flight must be covered by exactly one pairing. In practice several additional constraints are included such as balancing pairings across crew bases. The model is also different for long-haul flights due to different crew requirements for certain flights, e.g., some flights require a single captain and two first officers, while others require a single captain and three first officers.

Rostering

The rostering problem is to assign individual crew members to a set of pairings. These pairings are obtained in the crew pairing phase and span a given time horizon, typically corresponding to a month. The output of rostering is a month of work or *roster* for each individual crew member. Each pairing in the time horizon has to be covered by a group of crew members as specified by the underlying fleet. Typical requirements and constraints with respect to a roster include the rest time interval between two consecutive pairings, the training and vacation periods, and the crew members' personal desires. Each roster is explicitly constructed for each individual crew member and it takes into account as many individual preferences as possible. Similarly to pairings, regulatory bodies and union rules impose additional restrictions on a roster such as the total flying time within the time horizon.

In rostering, all rosters are constructed simultaneously. The following is a formulation of rostering, (Gamache & Soumis, 1998):

$$\begin{aligned}
&\min \sum_{k\in K}\sum_{s\in S^k} c_s^k x_s^k && \\
&\sum_{k\in K}\sum_{s\in S^k} \gamma_p^s x_s^k \geq n_p && p\in P \\
&\sum_{s\in S^k} x_s^k = 1 && k\in K \\
&x_s^k \in \{0,1\} && s\in S^k, k\in K
\end{aligned} \tag{2}$$

where

K : the set of all crew members,

S^k : the set of rosters compatible with crew member k,

P : the set of pairings (output from crew pairing optimization), training assignments, and open times to be covered,

c_s^k : the cost of assigning roster s to crew member k,

$$\gamma_p^s = \begin{cases} 1 & \text{if pairing } p \text{ is covered by roster } s \\ 0 & \text{otherwise,} \end{cases}$$

n_p : the required numbers of crew members for pairing p,

and the decision variables read

$$x_s^k = \begin{cases} 1 & \text{if roster } s \text{ is assigned to crew member } k \\ 0 & \text{otherwise.} \end{cases}$$

The first constraint ensures adequate coverage for each specific pairing, while the second constraint ensures that each crew member is assigned to exactly one roster. The objective cost coefficients c_s^k capture the individual preferences, seniority, and possible direct or indirect costs incurred by the airline.

Preferential Bidding

An alternative business process to rostering is preferential bidding. In rostering all rosters are constructed at once and then handed over to the crew members. On the other hand, in preferential bidding, individual lines are constructed one by one again by taking personal preferences into account and guaranteeing that subsequent bidlines can be constructed (solely based on feasibility requirements). The process is usually driven by seniority, i.e., the most senior crew members are the first to specify their preferences and bids. Nowadays, the bidding process is usually carried out online. An individual crew member states his or her own preferences, such as the desire for specific pairings, specific time intervals in a day, particular regions, or expected work loads, through an interface of the preferential bidding system.

Let us assume there are m crew members labeled based on their seniority. The first crew member has the highest priority, while the last one has the lowest seniority. Given crew member k, the lines for all higher seniority crew members $1,\ldots,k$-1 have already been constructed. The remaining available tasks (pairings, other open time, training) for crew member k are not the entire set of tasks but they correspond to the residual tasks. After all crew members receive specific lines, there might be some tasks that remain uncovered. We refer to the leftovers tasks, which include uncovered pairings, as the open time line, or simply open time. We also introduce a fictitious crew member labeled m+1, which corresponds to covering open tasks, if such tasks exist. The open line is always assigned to crew member m+1. The open time might actually require more than one crew member in which case several lines consisting of open time are being assigned to crew member m+1. This is interpreted and implemented as each line of open time requiring a different crew member (not in the original pool of crew members). The main idea behind the model is to consider only the personal preferences of the current crew member k, while guaranteeing that all remaining tasks can be covered by yet to be assigned crew members $k+1,\ldots,m,m+1$. The following formulation for preferential bidding for crew member k is given by Gamache, Soumis, Villeneuve, Desrosiers, & Gelinas, 1998:

$$
\begin{aligned}
\max \quad & \sum_{s\in\Omega_k^k} c_s^k y_s & & \\
& \sum_{s\in\Omega^k} a_{ps} y_s = 1 & & p \in P^k \\
& \sum_{s\in\Omega_e^k} y_s = 1 & & e=k,\ k+1,\ldots,m \\
& \sum_{s\in\Omega_{m+1}^k} y_s \le N & & \\
& \sum_{s\in\Omega_{m+1}^k} d_s y_s \le H & & \\
& y_s \in \{0,1\} & & s \in \Omega^k
\end{aligned}
\tag{3}
$$

where

Ω_e^k : the set of residual tasks available for crew member e, $k \le e \le m+1$,

$\Omega^k = \bigcup_{e=k}^{m+1} \Omega_e^k$,

P^k : the set of residual tasks,

c_s^k : the preference of line s when assigned to crew member k,

$$a_{ps} = \begin{cases} 1 & \text{if line } s \text{ includes pairing } p \\ 0 & \text{otherwise,} \end{cases}$$

d_s : the duration of open line s,

N : the maximum number of open time lines, and

H : the maximum duration of open time lines,

and with decision variables

$$y_s = \begin{cases} 1 & \text{if line } s \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

This model is applied m times for every crew member following the seniority list and the corresponding residual set is updated each time. The objective of the residual problem is to maximize the score, which corresponds to maximizing the employee's satisfaction. The first constraint ensures that exactly one line covers each residual task. The second constraint ensures that every crew member receives exactly one line. The third and fourth constraints concern the open time lines by restricting the total number and durations of such lines, respectively. The purpose of these constraints is to limit the number of open tasks and open lines. At the very end, open lines are assigned to reserve crews.

In the U.S., yet another process is often employed, called biding. Under biding, first all of the bidlines are constructed simultaneously, similarly to rostering. All of them are then presented to the crew members, who bid on them. The bidding process is based on seniority, i.e., the most senior crew member bids first.

Additional Considerations

Other than the high level models given above, additional important considerations need to be taken into account in a decision support system.

Long-haul vs. Short-haul Networks

There are important differences in operations and the underlying network between long- and short-haul legs. Typically, international flights are long-haul, while domestic are short-haul legs. First, the network structure is different. In the U.S., domestic flights are usually operated under a *hub-and-spoke* network. Big market airports with high passenger flow act as *hubs*, while smaller airports serve as *spokes*. The hubs are connected with each other, while a typical spoke is connected with an adjacent hub. Most passenger itineraries connect at hubs and thus the network exhibits many connections. International, long-haul flights, on the other hand, usually operate back-and-forth. In such a network there are fewer possible crew (and passenger) connections, which has pros and cons. On the one hand, it implies fewer pairings and rosters or bidlines. On the other hand, due to fewer opportunities to create pairings, deadheads are required, leading at the same time to many more overall flights to consider.

Cockpit crews have different regulatory restrictions on long-haul flights. Crews sometimes are required to be upgraded on a long-haul flight. For example, an original two-member cockpit crew on a domestic flight is upgraded to a four-member team for safety reasons due to a much longer airborne time on long-haul flights. Crew members alternate on duty with one another during long flights.

Finally, the cost structure of a long-haul pairing is often dominated by the time-away-from-base time, which is considered to be the overall cost of a pairing.

Crew Splitting

The rostering model already exhibited typically applies separately for cockpit and cabin crews. For each one of them, a separate optimization model is solved. Cockpit and cabin crews are usually treated separately due to the large discrepancy of the underlying requirements. Even pairing schedules are usually distinct. Cockpit crew members are rarely treated in these systems as individuals, but instead as a team or crew. This convention is not only beneficial for cooperation in the cockpit, i.e., from the business processes standpoint, but it also reduces the size of the problem. Moreover, cockpit crews are grouped by the fleet family due to the training requirements, while cabin crew members are grouped by aircraft configurations, e.g., number of aisles, and geographical regions.

Balanced Crew Schedules

When constructing crew schedules, it is desirable that the work loads are balanced among the crew members. A situation where one crew member is assigned a long overtime duty, while the remaining crew members are assigned merely minimum guaranteed duties, is not desirable. To construct balanced schedules, we can either penalize in the objective function unbalanced assignment or impose additional balancing constraints.

Moreover, it is desirable for the crew schedules to also be equally balanced with respect to crew bases. The number of pairings originating at one crew base must be almost equal to the number of pairings originating at a different crew base. Sometimes the balance is not with respect to the number of pairings, but, e.g., the flying time. To reach this goal, crew base balancing constraints are often added to the crew pairing model. They impose that the total work load comprised of the pairings originating from each crew base must fall within an upper and lower bound.

Robustness

Besides the actual direct crew cost, robustness is gaining importance in modern crew software. The main scope of robustness is to consider the actual day of operations and the fact that disruptions occur, necessitating schedule changes. The aforementioned models assume that the schedules will be operated as planned, but, this is seldom the case. The goal of robust crew schedules is to potentially produce suboptimal schedules with respect to the planned flight schedule, but, the crew schedule would fare better in case of disruptions, (Shebalov & Klabjan, 2006, Yen & Birge, 2006, Ehrgott & Ryan, 2002). A non-robust schedule might yield highly increased crew cost at the end of a month mostly due to irregular operations. For example, a crew originally assigned to a cancelled flight in operations must be rescheduled. On the other hand, it is extremely undesirable to cancel a flight due to the crew's unavailability resulting from its late inbound flight.

It is also risky to push work-load and safety rules to their absolute limits in planning because such a strategy would not allow any flexibility in case these bounds are reached in operations. The crew schedule would not have any buffers and thus it would be very fragile. A common strategy is to penalize in the models the patterns and bounds with a high probability of breaking during irregular operations. Other issues such as the allocation of reserve crews and the establishment of work-load rules used in planning are also relevant to robustness and should be carefully considered.

2.1.2 Operational or Execution Planning

The main part of operational planning related to crew is the so-called recovery problem. Recovery is invoked in the case of unscheduled events such as a cancellation or delay of a flight in the case of inclement weather. In many situations, the underlying flight schedule is changed from the original one. The effects of a cancellation or delay are not only applicable to the underlying flight, but they can easily cascade throughout the flight network. For larger disruptions it is not uncommon that the snowball effects roll over to the operations during the next day. In general, it is not desirable to modify unaffected flights, but often it has to be done in order to obtain an acceptable and implementable low cost solution. The general strategy of the recovery problem is to reschedule the part being directly affected by disruptions while maintaining the remaining portion of the original schedule as much as possible.

An important distinguishing feature of the recovery problem is the fact that it is an execution problem and thus solutions have to be produced in real time. As a result, different solution methodologies are usually employed. From the prospect of modeling, there are still rules confining the work load and connections. Labor contracts also dictate a different cost, which is usually the maximum of the planned pairing cost and the cost of the actual realized pairing. The constraints regarding safety or work-load rules may be looser than those in strategic and tactical planning. Different airline companies may have different policies confronting an emergency situation. From the prospect of human resource, the reserve crews are available and should be included in the model.

The crew pay in the recovery problem will be recalculated for the new schedule based on the strategy mentioned before, which accounts for the flying time, elapsed time, etc. On the other hand, it requires that the crew payment for that month is at least the same as that of the original schedule. In addition, extra costs associated with cancelled flights including passengers' compensation or extra operational costs to return crew back to the crew bases, must be taken into account. One of the most important components in the recovery crew cost is to return all crews back on time as soon as possible. Several airlines drive their crew recovery decisions exclusively based on the back-on-time performance indicator while imposing the maximum number of reserved crews and deadheads.

The following is the pairing recovery model for fleet type e, (Lettovsky, 1997; Lettovsky, Johnson, & Nemhauser, 2000):

$$\begin{aligned}
&min \quad \sum_{k\in K_e}\sum_{p\in P_k} c_p x_p + \sum_{l\in L_e} f_l y_l + \sum_{k\in K_e} q_k v_k \\
&\sum_{k\in K_e}\sum_{p\in P_k} \beta_{pl} x_p + y_l = 1 \qquad\qquad l\in L_e \\
&\sum_{p\in P_k} x_p + v_k = 1 \qquad\qquad k\in K_e \qquad\qquad (4)\\
&x \text{ binary}, y \text{ binary} \\
&v \geq 0
\end{aligned}$$

where

K_e : the set of crews available for fleet type e including reserve crews,

P_k : the set of pairings that can be served by crew $k \in K_e$,

L_e : the set of flights to be covered by crews available for fleet type e,

c_p : the cost of pairing p,

q_k : the cost of returning crew k to the crew base (or, in other words, not assigning a pairing to crew k)

f_l : the cost of canceling flight l,

$$\beta_{pl} = \begin{cases} 1 & \text{if flight } l \text{ is covered by pairing } p \\ 0 & \text{otherwise,} \end{cases}$$

The decision variables are

$$x_p = \begin{cases} 1 & \text{if pairing } p \text{ is assigned to a crew} \\ 0 & \text{otherwise,} \end{cases}$$

$$v_k = \begin{cases} 1 & \text{if no pairing is assigned to crew } k \\ 0 & \text{otherwise,} \end{cases}$$

$$y_l = \begin{cases} 1 & \text{if flight } l \text{ is canceled} \\ 0 & \text{otherwise,} \end{cases}$$

The objective of the recovery model is composed of three parts: the cost of rescheduling and reassigning crew, the cost of canceling a flight, and the cost of returning a crew back to the crew base. The first constraint ensures that each flight is either covered by a crew or cancelled. The second constraint captures the fact that crew cannot 'multi-task.' A crew is either reassigned to a single pairing or returned to the crew base for a possible later dispatch.

Other than the basic concepts described before, there are some other issues worth noting regarding the recovery problem.

The Original Schedule and 'Back-on-Time' Criteria

In the presented recovery model, a feasible solution is comprised of possible new pairings, i.e., pairings not originally in the monthly schedule. During recovery, it is desirable to retain the original schedule as much as possible. To take care of this point, in practice, a penalty measure determining the difference between the original and recovered schedules is often added to the objective function. A crew schedule that is less perturbed has a higher priority in being selected.

A deviation of this employed by some airlines is to recover the schedule as soon as possible. To this end, a new variable T is introduced capturing the last point in time where the crew schedule is changed. The objective function then simplifies to $\min T$.

Evaluating a Solution

In modeling, the goal is to capture as many details as possible while remaining tractable. Nevertheless, it is virtually impossible to capture all possible events and circumstances such as the ever changing operating environment. For this reason a human decision maker must interact with a recovery system. Due to this relation between the system and the decision makers, it is hard to evaluate the final solution. Unpredictability such as the exact arrival of a snow storm or its duration poses a major challenge in the evaluation effort. On a different front, estimation of cost coefficients such as the cost of canceling a flight, typically involves capturing passenger goodwill, and the reputation of the carrier. In addition, the network effects are hard to isolate and quantify. The resulting solution is sensitive to such estimations and thus accurate cost estimations are difficult to obtain.

Crew Splitting

In practice, to gain additional flexibility, most airlines allow crews to be split in recovery. A crew arriving on a flight can be split so that a first officer continuous on one flight while the captain resumes his or her duties on a different flight. The aforementioned model does not allow crew splitting since it deals with pairings, which are building blocks of crews. A similar model can be designed by capturing individual rosters of each crew member instead of pairings.

2.2 Solution Methodologies

Behind the hood of an optimization-based crew management system are mathematical optimization models with sophisticated solution methodologies. The crew management models yield instances with large numbers of variables and substantial complexity. Developing robust solution methodologies for these models is a continuing research topic. An acceptable solution methodology must excel in solution quality, accuracy, ease of use, robustness, and efficiency.

In the framework of crew management, rows and columns have associated meanings. The rows correspond to subjects, which in crew management are the flights, duty periods, or pairings. The subjects must be contained in containers, e.g., the flights, duty periods, or pairings must be covered. The columns are associated with feasible containers. The goal is to choose from all possible containers so that the selected containers include all of the subjects. We have to choose selected combinations of flights, which are duty periods, to cover all the flights; or we have to choose from combinations of duty periods, which are pairings, to cover all the duty periods. The process of finding possible containers or columns is called *column generation*.

The most popular algorithm to solve this problem consists of the following steps. Firstly, we solve the *restricted master problem*, which considers only of a portion of the columns rather than all of them. Next, we solve the *subproblem*, which aims to identify or generate the columns that could improve the current solution. The basic idea applied in this process is to find the columns with the lowest reduced cost based on the dual variables (known also as shadow prices in economics). The columns obtained from the subproblem are added to the restricted master problem. The previous steps are repeated until we obtain a satisfactory solution or we have a provably optimal solution. We refer to this algorithm as *delayed column generation*.

It is worth noting that there is a difference between the so-called *master problem* and the restricted master problem. The master problem contains all possible variables, while the restricted master problem only contains a subset of them. Moreover, there are challenges in the step of subproblem solving because it involves sifting through a huge number of columns. Typically the subproblem is modeled as a network problem.

To illustrate the network concept, let us focus on pairing generation. The network is constructed in such a way that every possible pairing is a path in the network (but every path does not necessarily correspond to a pairing due to the various rules governing the feasibility of a pairing). Then the pairing generation problem becomes the shortest path problem, or a more complicated variant of the multi-constrained or multi-label shortest path problem. The nodes in the network could be the origins or destinations of flights or duty periods, called the flight or duty period network, respectively. In a flight network, the first family of arcs in the network links the departure node with the destination node of each flight. The origin node of a flight is also connected from an artificial source node of a crew base if the flight departs from the crew base. Similarly, the destination node of a flight is connected to an artificial sink node of a crew base if the flight arrives at the crew base. The remaining connecting arcs allow crew connections between two flights. They must capture either sit connections within a duty or rest connections between two consecutive duties in a pairing.

In a duty period network, flights are substituted by duty periods. Each duty period yields two nodes and the associated arc linking them. Connection arcs connect two duties that can be subsequent in a pairing.

There is a trade-off between the two networks. The duty period network has many more nodes and arcs than the flight-based network. It could provide a severe strain on computing resources. The flight network can be easily represented even on low-end computers, yet it does not capture as many rules as the duty period network. The duty period network by construction embeds all of the rules directly pertaining to duties and thus it has a smaller number of paths that do not lead to a pairing.

In roster or bidline generation, nodes correspond to origin and destinations of pairings and arcs to legal rest periods between two consecutive pairings.

To capture legality requirements and for proper cost accounting, we introduce a label to track various resources at each node. The label, which is a multi-dimensional vector, contains attributes, such as the total elapsed time, the number of accumulated flights, or the cost that has been accumulated up to the node under consideration. The goal is to find the shortest (cheapest) path with all of the constraints satisfied. At an intermediate node, we usually keep track of several partial paths that connect to the node. Each path is associated with exactly one label. Technical details of the multi-label shortest path method are described in Section 2.2.2.

Similar to many other large-scale models, the methodologies used to solve the crew management optimization problems must be aligned with the development of computer architectures. For example, consider two different solution methodologies A and B. Methodology A takes 2 days to get an acceptable solution while methodology B requires 30 days to get an optimal solution using today's computing hardware. At this stage, methodology A is preferable for its efficiency. The developments in computer hardware such as multi core processing units may lead to methodology B becoming preferable over methodology A. This conceptual example illustrates that the resources at hand (here, the computing architecture technology) play an important role in determining the solution methodology.

On the other hand, different methodologies are more appropriate for solving different problems. Despite the superficial similarities of the presented models (they are all set covering problems with possible side constraints), each problem has its own characteristics stemming from column feasibility rules and the underlying scope. Before several models are to be solved, structures and characteristics of the models should be inspected and an adequate methodology must be carefully selected. In the following sections we present methodologies and techniques used in solving large-scale crew related models. We also link them to the crew management problems outlined in the previous sections.

2.2.1 Basic Techniques in Large-scale Optimization

Solving mixed integer programs can be a daunting task, especially when the problems are large-scale. *Branch-and-price*, *Lagrangian decomposition*, and *Benders decomposition* are three prevailing techniques used in large-scale optimization. Each technique treats the problem from a different viewpoint. We introduce the basic ideas of these three techniques in this section, but explain the details pertaining to crew management in the subsequent sections.

Branch-and-Price

Integer programs are intriguing to solve since an explicit representation of the underlying feasibility set is impossible in a reasonable computational time. To circumvent this, we often relax the integrality restriction of selected variables and solve the resulting linear program instead, which is referred to as the linear programming *relaxation*. After the relaxation is solved, its feasible region is partitioned into two non-overlapping regions, or *branches*. The two branches are created based on a fractional variable in the optimal solution of the linear programming relaxation. This procedure, when applied repeatedly, is often denoted by a tree structure, where each node represents a linear programming relaxation and each child node represents the resulting branch. If an integer solution is obtained at a node, we record this solution, if it provides a better lower bound of the objective value, if we are maximizing, or upper bound, if we are minimizing.

The above method is called branch-and-bound. The disadvantage of the branch-and-bound algorithm, especially with a large-scale problem, is in the fact that often we have to visit a huge number of nodes and solve millions of linear programming relaxations. It is a method that, in the worst case, somehow enumerates all of the possible integer solutions and finds the best one among them. A variant of the branch-and-bound algorithm solves the linear programming relaxations using column generation at each node, which is preferable in large-scale optimization. This method is called *branch-and-price* because we often refer to the step of solving the subproblem as *pricing* during the column generation process. The branch-and-price method focuses on efficient ways to solving the linear programming relaxations, while the next two methods take advantage of the special structure of the problem in a completely different way.

Lagrangian Decomposition

The Lagrangian decomposition method (Fisher, 1985) decomposes the problem horizontally. All of the constraints are categorized into *hard* and *easy constraints*. Instead of solving the original problem with hard constraints, these constraints are relaxed and a penalty in the objective function is assigned, if they are violated. The problem we are now facing is called the Lagrangian relaxation problem. It has no hard constraints; but an additional penalty is introduced to the objective function, where each penalty coefficient or Lagrangian multiplier corresponds to a hard constraint. Lagrangian decomposition is an iterative procedure that aims to find a tighter bound at each iteration by adjusting the values of the multipliers. For example, if the original problem is a maximization problem, the objective value obtained from the relaxed problem is actually an upper bound of the true optimum. The Lagrangian dual problem is to find the lowest upper bound among all possible multipliers. In rare instances, the value of the Lagrangian dual equals to the optimal value, but more often the Lagrangian dual bound is larger than the optimal value.

A sub-gradient method is often used to find the smallest upper bound of the relaxed problem. It is an iterative minimization problem over a piecewise linear convex function. The multipliers and step size are updated until a satisfactory upper bound is reached or a very small in norm descending direction is obtained. The encountered solutions may not be feasible and thus a separate heuristic needs to be designed that forces the solutions to satisfying the hard constraints (Lagrangian multipliers are used as guidance in this process). Lagrangian decomposition is especially powerful for the problems with several hard constraints linking together different parts of the model, while other constraints are relatively easy to capture.

Benders Decomposition

The Benders decomposition method (Benders, 1962) decomposes the problem vertically. We categorized all of the variables into *linking* and *complicating variables*. This method takes advantage of the block diagonal structure of the problem. In the framework of Benders decomposition, the focus is on the so-called Benders master problem optimization. The Benders master problem is formed by projecting the complicated variables through the introduction of additional constraints in the original problem. At each iteration, first the Benders master problem is solved, which includes only the linking variables. Next we solve a problem, called the Benders subproblem, over the complicated variables while the linking variables are fixed. The optimal solution of the Benders subproblem is an upper bound, if the sense of the objective function is minimization, to the original problem. On the other hand, the solution of the Benders master problem is actually a lower bound. By adding a constraint resulting from the optimal solution of the Benders subproblem to the Benders master problem, the gap between the upper and lower bound does not increase. The constraints added to the Benders master problem are called *Benders cuts*. The algorithm stops as soon as the gap becomes small enough.

A key requirement for using Benders decomposition is that the Benders subproblem has to be a linear program. The Benders decomposition methodology is especially powerful for mixed integer problems with integer linking variables.

2.2.2 Algorithms for Crew Management

In this section, we dive into the algorithmic techniques used in crew management. The already presented algorithms out-of-the-box yield unsatisfactory solutions and thus several custom-based strategies must be employed, e.g., branching decisions, pricing, etc. Elegant algorithms tailored to crew management focus mainly on two aspects. The first aspect is not to enumerate all the possible pairings, rosters, or bidlines, but merely enumerate those that have potentials to be in an optimal solution. The other aspect is to find an optimal (or near optimal) solution quickly. The techniques that follow focus on at least one of these two aspects.

Local Improvement Heuristics

We outline local search heuristics only from a historical perspective. These were one of the first solution approaches employed by computer based systems. These days they are outdated in comparison with more sophisticated algorithms with a more global view.

The algorithms require a starting feasible solution, which is then improved in subsequent iterations. In a typical local search algorithm for crew pairing optimization, a 'spot' corresponding to a subset of pairings of the whole problem at a given iteration is identified and the solution is improved spot by spot. For the selected spot of the incumbent solution, such as 10 pairings among the 100 pairings in the entire solution, we enumerate all possible pairings covering the flights contained in these 10 pairings. Next a much smaller pairing problem is solved over newly generated pairings. The selected original pairings are replaced by the pairings in the solution of the smaller problem. If, for example, two spots are selected with non overlapping pairings, then two smaller problems can be solved concurrently in a given iteration.

A spot might be selected based on a time interval in the planning horizon, or based on stations, or even picked randomly. It is possible that the spot at one iteration overlaps with a subsequent spot in the next iteration. The local improvement heuristics have the advantage of solving much smaller integer programs even though they have to be solved many times.

Usually it is not difficult to find a feasible initial solution (at least for large-scale instances). The pairing schedule operated in the previous season is an excellent reference point since the airlines do not change too many flights from season to season and year to year.

Branch-and-Price

Branch-and-price algorithms have a holistic view of the problem. They do not single out a smaller problem, but instead new columns are generated based on the entire network used in pricing.

To implement the branch-and-price algorithm for crew pairing, we first need to specify a suitable branching rule. A valid branching rule requires that the two branching decisions are mutually disjoint and compatible with the pricing algorithm. For example, standard variable dichotomy requires in one branch a pairing to be selected in the solution, while the other branch must disallow this pairing. This simple rule is intuitive but impractical because it is much more difficult to get solutions with the rules of "must not have" than the rules of "must have." In preventing a pairing to be selected, the pricing problem now because the problem of finding the second shortest path, which is a much more intriguing problem than the vanilla shortest path problem. A different branching rule (Ryan & Foster, 1981) considers two flights simultaneously. On one branch, the two selected flights must be covered together by the same pairing and on the other branch, the two specific flights must be in two distinct pairings.

In the crew pairing application, we do not only consider two specific flights, but two consecutive flights in a pairing to make the strategy even more efficient. On one branch, the specific flights r and s must be covered consecutively by a pairing (the follow-on branch), and on the other branch, r and s must appear non-consecutively in pairings (they can either be in two distinct pairings or in the same pairing but not following each other). This strategy nicely balances the branch-and-price tree, yet it preserves the structure of the pricing algorithm. This branching rule is often called branch-on-follow-on.

In the follow-on branch, the two flights can be merged together and considered as a single entity. Within the flight network, this can be achieved by eliminating all connecting arcs from flight r except to flight s. In the duty period network, the same effect can be achieved by removing select duty periods and then fixing rest connections similarly as in the flight network. In terms of the constraint matrix, the two rows become identical and thus one of them can be eliminated. In the other branch we can take the same steps except that we remove the arc connecting the two flights. In this branch the constraint matrix remains intact, i.e., no reduction is possible.

If all flights depart at different times, the departure time becomes a distinguishing characteristic of flights. The branching rule based on departure times (Klabjan, Johnson, Nemhauser, Gelman, & Ramaswamy, 2001) is called the timeline branching. Based on this rule, we branch upon a specific flight r and a specific connection time t. On one branch, the flights immediately following r with the connection time of less than or equal to t are covered by the same pairing, and on the other branch, the flights immediately following r with the connection time longer than t are covered by the same pairing. For a flight schedule with equal departure times, we can either perturb the times slightly without harming the structure of the schedule or employ branch-on-follow-on when a 'fractional' r and t cannot be found. The timeline branching even more balanced the tree and it is also easily embeddable within the pricing network.

Besides creating balanced trees and compatibility with pricing, the branching decisions must guide the search of the tree toward finding a feasible integer solution resulting in an improved lower bound. A recent technique of selecting a good branching variable is called strong branching, and is today used in several commercial mathematical programming solvers. The basic idea is as follows. For a possible branching variable i and the two resulting branches, we perform k dual simplex iterations on each of the two branches, where k is a parameter. This yields two objective values f_i^0 and f_i^1 of the dual problems for the two branches. If f_i^0 and f_i^1 are large (for a maximization problem), the variable has a high priority for being selected for branching. We can use a combination of f_i^0 and f_i^1 as the score to evaluate a suitable variable. We perform the above steps on a set of candidate variables, and then choose the highest score variable within the set to branch on. The strong branching strategy should be combined with existing branching rules for crew pairing optimization (branch-on-follow-on or timeline branching).

Multi-label Shortest Path

We have already discussed that the subproblem or pricing is usually modeled as a network problem. Due to nonlinearities in the cost and restrictions on paths, the standard shortest path algorithms cannot be employed in pricing. Instead, a more complicated multi-label or resource constrained shortest path algorithm needs to be employed.

The multi-label shortest path algorithm relies on the network structure described at the beginning of Section 2.2. We demonstrate the basic principles by the example shown in Figure 3. In the flight network case, nodes represent the origination/destination of flights and two types of arcs represent connections. The solid arcs represent the actual flights, while the dashed arcs represent ground connections, which is either the sit or layover connection between flights. The goal is to generate a pairing that begins and ends at the base station and has low reduced cost. To capture the dual prices, which correspond to flights, each flight arc in the network is assigned the matching dual value. The main idea is to proceed from a node to another, updating the labels along the way. Each node has several labels, each one associated with an underlying partial pairing. In our example, labels have four attributes, which are the reduced cost, total flying time, TAFB, and the number of flights. We have label (25, 3, 3, 1) at the destination node of flight *a* to represent the accumulated reduced cost of 25 units, total flying time of 3 units, TAFB of 3 units, and 1 operated flight. This label is obtained by traversing flight *a* from the label associated with the origin node of flight *a* (this label is not shown in the figure). After proceeding to the origin node of flight *b*, the accumulated reduced cost and the TAFB are updated and the label attached at this node becomes (28, 3, 9, 1). Such an update is based on the connection time between these two flights of 6 time units and the additional reduced cost contribution of 3 time units. Similarly, after proceeding to the destination node of flight *b*, the label is updated to (49, 6, 12, 2), which implies that the flying time of this flight is 3 time units. In addition, the crew cost related contribution of this flight minus the associated dual price (the reduced cost contribution) of flight *b* equals to 21 time units.

This procedure is then continued. The origin node of flight *d* yields label (69,8,19,3), which is obtained after traversing 3 flights. Without additional enhancements, as presented, the method enumerates all possible pairings, which would result into computational intractability. If an attribute of a label exceeds one of the resource limits related to feasibility rules, then the label is discarded. In our example, if the maximum TAFB is 16 time units, then label (69,8,19,3) is discarded. Some rules such as the non-replicated-flights-in-a-pairing rule, which is present in daily problems, are computationally challenging and should be checked only at the end after the pairing 'survived' all of the other feasibility rules.

An additional important enhancement is based on the notion of dominance. Upon visiting an intermediate node with several labels, one of them showing 'no competitiveness', we can discard this label since the path associated with it will never be in the optimal solution. The noncompetitive label is any label with all of its values dominated in every attribute by another label. In our example in Figure 3, there are two possible legal paths at the origin of flight *d* and thus two labels. The first path consists of fights $a \rightarrow b \rightarrow c \rightarrow d$ and the second one of flights $e \rightarrow f \rightarrow g \rightarrow h \rightarrow d$. None of the two paths can be eliminated because one path has a longer flying time, a larger numbers of flights, and higher reduced cost, but the other path has a longer TAFB. If label (72,9,18,4) is changed to (72,9,20,4), then it is dominated by the other one and thus it can be discarded. This is based on the observation that in this case every sequence of flights extending the bottom path can be appended to the top path to yield a feasible path with a lower reduced cost. It is important to note that the dominance strategy is not applicable to any rule. If there is a rule requiring that the total flying time in a pairing must exceed a certain value, then dominance with respect to the flying time is no longer applicable.

Our example relies on the flight network, but the application of multi-label shortest path in pricing is also applicable on the duty period network (see Barnhart, Johnson, Anbil, & Hatay, 1994; Lavoie, Minoux, & Odier, 1988; Vance, Barnhart, Johnson, & Nemhauser, 1997).

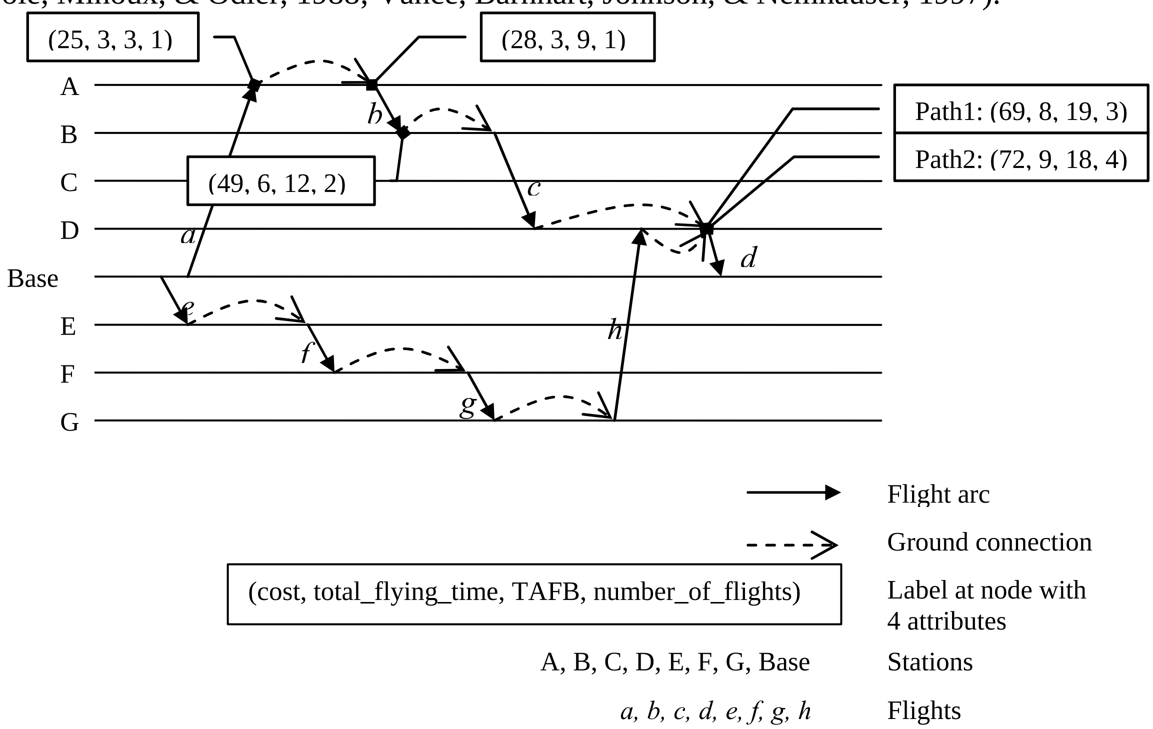

Figure 3: Flight Network and Multi-label Shortest Path

Lagrangian Based Methodology

While the crew problems are not really fitted for Lagrangian decomposition, they found success in some commercial implementations. For set partitioning problems usually all of the rows are relaxed and thus the selection of a Lagrangian multiplier plays an important role. The goal is to find multipliers u that lead to the unique solution of the Lagrangian relaxation problem, and this solution is also feasible to the original integer programming problem. We may take advantage of the special structure of the crew pairing problems with only 0 or 1 entries in the constraint matrix and the fact that the right-hand side is integral. With these characteristics, it is easy to obtain a solution to the Lagrangian relaxation problem. If the reduced cost of a variable is positive, the variable's value is 0; if the reduced cost is negative, its value is1. A difficulty arises in the case when the reduced cost is zero, and thus x can either be 0 or 1, and thus it creates ambiguity in finding a solution to the original integer programming problem. As a conclusion, the goal is to find multipliers u that lead to nonzero reduced costs.

We can apply a coordinate search algorithm and check feasibility of the original integer programming problem for every u iteratively. Unfortunately, a multiplier with desired properties described above may not exist. One way to resolve this difficulty is by using the iterative approximation algorithm (Wedelin, 1995). The main idea is to slightly perturb the coefficients of the objective function so that the optimality set does not substantially change and the uniqueness property holds. Instead of maintaining a single multiplier with each row, at each iteration, a constraint has two possible values u^+ and u^- corresponding to the multipliers. If the reduced cost of a variable in the iteration is positive, we increase u^+; if the reduced cost is negative, we decrease u^-. Values u^+ and u^- are controlled by two parameters κ and δ, where κ is the parameter controlling the level of distortion of cost coefficients, and δ ensures that the reduced cost is nonzero. If κ equals 0, there is no approximation or distortion.

To demonstrate the algorithm, let us assume that we are handling the minimizing problem of the form

$$\begin{aligned} min \quad & cx \\ & Ax=b \\ & x_j \in \{0,1\} \end{aligned} \tag{7}$$

where b is integral. The Lagrangian relaxation of this problem is

$$\min_{0 \le x \le 1} \; cx + u(b - x). \tag{8}$$

We define the reduced cost vector $\bar{c} = c - uA$, the index of rows by i, the index of columns by j, and an array s^i. In array s^i each entry corresponds to a variable. The steps of the approximation algorithm are as follows, where we start with a random row.

Step 1: At current row (constraint) i, let

$$\begin{aligned} & r^i = \bar{c} - s^i, \\ & r^+ = \text{the } b_i\text{'th smallest element of } r^i, \\ & r^- = \text{the } (\mathrm{b}_i + 1)\text{'th smallest element of } \mathrm{r}^i. \end{aligned}$$

Step 2: Choose $0 \le \kappa \le 1$ and δ based on empirical observations. Let

$$u_i^+ = u_i + \frac{\kappa}{1-\kappa}\left(r^+ - r^-\right) + \delta$$

$$u_i^- = u_i - \frac{\kappa}{1-\kappa}\left(r^+ - r^-\right) - \delta$$

Step 3: Update s^i by setting

$$s_j^i = \begin{cases} -u_i^- & \text{if } r_j^i \le r^- \\ -u_i^+ & \text{if } r_j^i \ge r^+. \end{cases}$$

Step 4: Update $\bar{c}$ by $\bar{c} = r^i + s^i$

Step 5: If all of the rows are visited, stop. Otherwise, go to the next row and repeat Steps 1 to 5.

These steps are repeated with κ gradually increasing from 0. Each time we loop through all of the rows, the corresponding x is computed based on the incumbent reduced cost $\bar{c}$. This corresponds to the reduced cost with respect to the Lagrangian multipliers $u_i = {(r^+ + r^-)}/{2}$. It is desirable that in each iteration the order of the rows is randomly selected.

In Table 1, computational results on real-world crew pairing problems are summarized, and a performance comparison between two algorithms is listed. From the experimental test, compared with the performance of the commercial solver CPLEX, the approximation Lagrangian algorithm performs very well in terms of the computational time on larger problem instances. Since crew management problems are large-scale, this makes the algorithm suitable for such problems. On the other hand, the approximation algorithm provides an objective value that is often optimal or very close to being optimal. The approximation algorithm presented is the core of the optimization engine in the software provided by Jeppesen[3] (see Andersson, Housos, Kohl, & Wedelin, 1998).

	Size		CPLEX		Approximation algorithm	
Problem	Rows	Columns	obj.	time	obj.	time
B727scratch	29	157	92,800	0.2s	92,800	1.4s
ALITALIA	118	1,165	5,017,500	1.0s	5,017,500	11.0s
A320	199	6,931	529,250	23.0s	529,250	1m4s
A320coc	235	18,753	565,000	2m12s	565,000	5m2s
SASjump	742	10,370	4,737,768	23m	4,737,892	3m58s
SASD9imp2	1,366	25,032	4,333,450	5h53m	4,335,780	7m46s
SASD9imp1	1,585	105,804	-	-	4,329,750	36m10s

Table 1: Computational results for crew pairing problems using CPLEX and the approximate Lagrangian algorithm (Wedelin, 1995)

2.2.3 Recent Advances

Three recent solution methodologies appearing in the last decade with substantial success in crew management are introduced in this section. They are the volume algorithm, primal-dual subproblem simplex, and constraint programming.

Volume Algorithm

The volume algorithm (Barahona & Anbil, 2000, Barahona & Anbil, 2002), is an extension of the subgradient method. It can be used to solve a linear programming Lagrangian relaxation (8). In this section we focus on the case $b=1$ since this is the usual set partitioning problem that is often confronted in the crew scheduling problems.

In the subgradient method, the current incumbent value is improved through an iterate search process. We update multiplier $\bar{u}$ in the direction of a subgradient, or based on the volume algorithm in the direction of a combination of a subgradient and the previous directions. The standard subgradient algorithm is simple, however, only the objective value and the value of the multipliers are obtained without providing a corresponding primal solution x. To obtain x, extra efforts are required, which are part of the volume algorithm. The main idea of the volume algorithm, which is given next, is to obtain an approximate primal value $\bar{x}$ by an updating process within the subgradient method. We denote the current objective value by z.

Step 1: Let $0<w<1$ be a parameter. Find an initial multiplier $\bar{u}$.
Solve (8) to obtain the current objective value $\bar{z}$ and solution $\bar{x}$.
Step 2: Make a step in the direction of a subgradient at $\bar{u}$.

[3] Carmen Systems was acquired in 2006 by Jeppesen.

Let the subgradient be $v^t = 1 - A\bar{x}$, the step size $s = \lambda \frac{UB - \bar{z}}{\|v^t\|^2}$, where $0 \le \lambda \le 2$, and UB is an upper bound of the optimal objective value.

Compute $u^t = \bar{u} + sv^t$.

Step 3: Solve (8) with u^t to obtain the new objective value z^t and solution x^t.

Step 4: Update the primal solution $\bar{x}$.

Compute $\bar{x} = \alpha x^t + (1-\alpha)\bar{x}$, where α solves the one-dimensional problem:

$$\min_{\frac{w}{10} \le \alpha \le w} \left\| 1 - A\left(\alpha x^t + (1-\alpha)\bar{x}\right) \right\|.$$

It is suggested to set w=0.1, and if z^t is not increased by at least 0.1%, let $w=w/2$.

In addition, if z^t does not increase enough, let $\bar{u} = u^t$.

Step 5: If $z^t > \bar{z}$, let $\bar{u} = u^t$ and $\bar{z} = z^t$.

Step 6: Set $t = t+1$. Go to Step 2.

The volume algorithm provides an easier stopping criterion than the out-of-the-box subgradient methods whose stopping criterion is not robust and cumbersome to maintain. Since an approximate primal objective value is easily obtainable from the approximate primal solution $\bar{x}$, the gap between this approximate primal objective value and the lower bound $\bar{z}$ is used as a stopping criteria. If this gap is less than a given threshold, the procedure is stopped.

It is worth mentioning that if we unwind the approximate primal solution $\bar{x}$ in Step 4, it is a convex combination of $\{x^0, \ldots, x^t\}$ if α is stationary:

$$\bar{x} = \alpha x^t + (1-\alpha)\alpha x^{t-1} + \ldots + (1-\alpha)^t x^0. \tag{9}$$

The most recent x^t has the largest contribution toward $\bar{x}$ while the previous vectors have lower contributions. To interpret the meaning of the primal solution $\bar{x}$ geometrically, we consider the space of $\bar{z}$ and $\bar{u}$. Given vector $(\bar{z}, \bar{u})$, the polytope in the space (z,u) is defined by all active constraints (at equality) with respect to $(\bar{z}, \bar{u})$ of $z + a_i u \le b_i$ and $z \ge \bar{z}$. Vector $\bar{x}$ turns out to be the ratio of the volume between the active faces $z + a_i u = b_i$ and their projection at $z = \bar{z}$, to the total volume of the entire polytope in a neighborhood of $(\bar{z}, \bar{u})$ – this statement is considered coordinate-wise with respect to coordinate i of $\bar{x}$. This result implies the name of the volume algorithm.

The volume algorithm has been used in a crew pairing system developed by IBM for US Airways and Southwest Airlines. In a crew scheduling set partitioning experiment with two instances, as Table 2 shows, the volume algorithm outperforms both the dual simplex and the primal-dual barrier algorithm. Moreover, a combination of the volume algorithm and dual simplex, which embeds the volume algorithm in column generation, reduces half of the computational time of the pure volume algorithm. In this combined algorithm, the dual values obtained from the volume algorithm are used to compute reduced costs and to select columns added to the restricted master problem, while dual simplex is applied to solve the restricted master problems.

Problem:	size		Volume algorithm	Volume algorithm + Dual simplex	Dual simplex	Primal-dual barrier algorithm
	rows	columns	Time (sec)	Time (sec)	Time (sec)	Time (sec)
sp6	2,504	50,722	440	135	3,283	1,299
sp7	2,991	46,450	895	428	5,753	2,048

Table 2: Computational results of the volume algorithm, a combination of the volume algorithm and dual simplex, the pure dual simplex, and the primal-dual barrier algorithm on set partitioning problems (Barahona & Anbil, 2000)

Primal-dual Subproblem Simplex Method

The primal-dual algorithm dates back to 1956 (Dantzig, Ford, & Fulkerson, 1956). In early days it was primarily used to exhibit polynomial algorithms for certain combinatorial problem. Later, the basic idea of the primal-dual algorithm has been modified and embedded into many solution methodologies. The variant we present here is a primal-dual simplex algorithm combined with column generation. It is well fitted for solving large-scale linear programs.

To demonstrate the primal-dual simplex we assume the following primal-dual pair:

(Primal problem) $\quad \min\ cx \quad Ax=b \quad x \geq 0$

(Dual problem) $\quad \max\ \pi b \quad \pi A \leq c.$

In column generation for large-scale problems, only a subset of columns is considered at any point in time. Solution $\tilde{x}$ of the restricted master problem is always primal feasible (to obtain a complete primal solution x we simply append 0's to $\tilde{x}$) . On the other hand, the corresponding dual solution ρ may not be feasible to the original dual problem. The key idea of the primal-dual simplex method (Hu & Johnson, 1999) is to improve the infeasible dual solution ρ by considering a convex combination of ρ and $\bar{\pi}$, where $\bar{\pi}$ is a given feasible dual solution to the original problem. We denote the new improved feasible dual solution as π' , which is also used to select the desired columns that are included into the restricted master problem in the next iteration. The steps of the primal-dual simplex method are as follows

Step 1: Obtain an initial feasible dual solution $\bar{\pi}$ of the original problem and initialize the restricted master problem with a subset of columns, potentially including artificial variables to make the restricted master problem feasible.

Step 2: Solve the restricted master problem with a subset of selected columns by any simplex type linear programming algorithm.

Let $\tilde{x}$, ρ be the corresponding optimal primal, dual solutions, respectively.

Step 3: If ρ is feasible to the dual problem, we stop. Pair ($\tilde{x}$, ρ) is optimal.

Step 4: Let $\pi' = \theta\bar{\pi} + (1-\theta)\rho$, where $0 \leq \theta \leq 1$ is such that $\pi' A \leq c$ and $\pi' b$ is as large as possible. Value θ is expressed as

$$\theta = \max_{j,\bar{c}_j^{\rho}<0} \left\{ 0, \frac{-\bar{c}_j^{\rho}}{\bar{c}_j^{\pi} - \bar{c}_j^{\rho}} \right\}$$

where $\bar{c}_j^{\pi} = c_j - \pi A^j$, $\bar{c}_j^{\rho} = c_j - \rho A^j$ for column j and A^j is column j of the constraint matrix. These are the underlying reduced costs.

Step 5: Construct the new restricted master problem. Preserve the basis columns in the restricted master problem but remove all other columns. Add the columns associated with the lowest reduced costs $c_j - \pi' A^j$ to the new restricted master problem.

Step 6: Set $\overline{\pi} = \pi'$ and go to Step 2.

The primal-dual simplex method is efficient when the number of columns is much larger than the number of rows. In the computational experiment results summarized in Table 3, the primal-dual simplex reduces by half the time it takes using standard linear programming algorithms to solve linear programming relaxations of the crew pairing problems. If an incumbent primal solution is known in advance, the computation time for the primal-dual simplex can be further reduced albeit with some technical tricks. The primal-dual simplex method can also be parallelized by distributing the columns across various computational resources to solve several restricted master problems concurrently and dramatically reducing the overall execution time (see Klabjan, Johnson, & Nemhauser, 2000). Another variant of primal-dual simplex (Barnes, Chen, Gopalakrishnan, & Johnson, 2002) uses a non-negative least squares problem to find a direction of the steepest ascent. This method shows its merit in extremely degenerate problems.

Problem	No. flights	No. pairings	P-D simplex without an incumbent solution	P-D simplex with an incumbent solution	P-D simplex with an incumbent solution and improved initial dual	Primal simplex with 1000 columns added at a time
			Time (sec)	Time (sec)	Time (sec)	Time (sec)
No.1	61	3,754	0.8	0.9	0.7	-
No.2	144	48,574	7.6	7.4	5.85	12.6
No.3	174	209,935	30.0	25.7	20.3	36.0
No.4	202	2,335,782	155.6	-	-	-
No.5	382	2,000,000	58.7	39.5	-	65.0
No.6	930	2,000,000	277.0	194.0	-	-

Table 3: Computational results for the primal-dual (P-D) simplex and primal simplex on linear programming relaxations of crew pairing (Hu & Johnson, 1999)

Constraint Programming Based Column Generation

In a very dynamic world of crew management, the decision support components and the underlying algorithms must be very robust. Regulatory rules and, in particular, labor agreements often change. Each such change requires modifications to the business rules embedded in solution methodologies. The algorithms presented so far pose major challenges in this direction. It is very hard to design a very robust branch-and-price algorithm. Pricing is the 'sticking' point since it relies heavily on rules specifying legality of a pairing. Constraint programming is an interesting direction mostly focusing on ease of incorporating legality rules.

In a general mathematical program, the goal is usually to find an optimal solution within a feasible set. In constraint programming, the focus is not on optimality, but rather feasibility. In pricing, this amounts to switching from the goal of finding a column with the lowest reduced cost to the problem of finding a column with a reduced cost below a certain parameter. This can be casted as a feasibility problem: Does there exist a column with the reduced cost below the parameter? The main concept behind constraint programming is to find solutions by confining the relations between variables as long as the output solutions satisfy some desired properties. This type of a problem is also called a *constraint satisfaction problem* (CSP). A CSP is defined by a triple (X,D,C), where $x=\{x_1,x_2,...,x_n\}$ is a set of variables, C is a set of constraints over variables in $x \in X$ each with a relation R, and D is a set of values referred to as the *domain* that variables must belong to. A solution, denoted by v, which satisfies constraints in C, is a mapping, i.e., it is a value assignment to the variables. Furthermore, a solution v satisfies a constraint in C if and only if $v(x_i)$ is an element in the set of relation R of the constraint. Formally, the relation between v and (X,D,C) can be expressed as follows:

$$v: X \to D \text{ such that } v(x_i) \in R \subseteq C.$$

In order to solve large-scale crew management problems, the methodology recently developed by Fahle, Junker, Karisch, Kohl, Sellman, & Vaaben, 2002 and Junker, Karisch, Kohl, Vaaben, Fahle, & Sellman, 1999 embeds constraint programming within column generation. Constraint programming has no effect on the overall branch-and-price scheme (or Lagrangian), it is merely incorporated as a pricing algorithm. The general framework of the *constraint programming based column generation algorithm* (Junker, Karisch, Kohl, Vaaben, Fahle, & Sellman, 1999) is as follows.

- *The subproblem* - A subproblem SP is a CSP where the set of the variables is denoted as (y, s, b). There is a one-to-one correspondence between the elements in y and those in the master problem. Let i denote a row and $y \in X$ a column. A solution $s \in \{\le, =, \ge\}$ to the subproblem corresponds to a column (variable) b of the master problem. The constraint programming variable y_i in a subproblem is introduced to correspond to coefficient a_{ij} of variable x_j in constraint i in the master constraints, and $v_j(y_i)$ gives the value of a_{ij}. Variable z is introduced to correspond to the cost coefficient c_j in the objective function of the master problem, and $v_j(z)$ gives the value of c_j.
- *The master problem* - Given a set of solutions S to SP, for each solution $v \in S$ to the subproblem SP, which is a CSP, there is a corresponding variable x_v in the master problem. A constraint in the master problem for SP is specified by triple (y, s, b). A master problem, which is a mixed integer program, is specified by triple (X,D,C), where $M=\{mc_1,...,mc_n\}$ is a set of master constraints of SP, (Y, D, C) is a CSP, and $z \in X$ is a variable in SP. The objective function of the master problem is given as :
$$min \sum_{v \in S} v(z) \cdot x_v.$$
- *Negative reduced cost constraint in the subproblem* - The goal of selecting columns with negative reduced cost can be achieved by introducing a negative reduced cost constraint in CSP. Let λ_i be the dual value for each master constraint $mc_i = (y_i, s_i, b_i)$. The goal is to

search for a value of $\overline{y} = \left[\overline{y}_1, \ldots, \overline{y}_n\right]^T$ such that the negative reduced cost constraint specified by

$$\overline{z} - \sum_{i=1}^{n} \lambda_i \cdot \overline{y}_i \leq 0,$$

is satisfied, where $\overline{z}$ is the value of $\overline{y} = \left[\overline{y}_1, \ldots, \overline{y}_n\right]$.

- *The master constraint* - A master constraint $mc_i = \left(y_i,\ s_i,\ b_i\right)$ can be in one of the following forms

$$\begin{aligned} &\sum_{v \in S} v\left(y_i\right) \cdot x_v \leq b_i && \text{if } s_i = \text{"}\leq\text{"} \\ &\sum_{v \in S} v\left(y_i\right) \cdot x_v = b_i && \text{if } s_i = \text{"}=\text{"}, \\ &\sum_{v \in S} v\left(y_i\right) \cdot x_v \geq b_i && \text{if } s_i = \text{"}\geq\text{"}, \end{aligned}$$

where i represents a row index and variable x_v represents a selected column.

CSP can be embedded in a network-based shortest path problem to solve a subproblem. The application of CP on the rostering subproblem was proposed by by Fahle, Junker, Karisch, Kohl, Sellman, & Vaaben, 2002 and is outlined next. In the shortest path CSP, we search for the shortest path in a directed acyclic network. It requires the network to be topologically ordered, which means that the nodes are ordered and labeled based on the embedded time. If there exists an arc from node *A* to node *B*, it therefore implies that *B* is higher in the order than *A*, i.e., it 'occurs' later in time. For crew rostering, the nodes in the network represent the tasks (pairings, open time, etc) ordered based on the starting time. If the completion time of task t is smaller than the start time of task t', we add arc $\left(t,\ t'\right)$ to the network. The network also has a sink s' and a source s_c for each crew member *c*. Each path that starts at one of the source nodes and ends at s', and it satisfies all legality rules, represents a roster. Each arc has an associated cost denoted by $w_{t,t'}$ and representing the cost of assigning task t' immediately after task t. The assumption here is that the cost contribution of a certain task t is related to task t itself and the task t' that directly follows t. We also let $w_{s_c,t} = w_{t,s'} = 0$ be the cost associated with the arcs associated with the sources and sink.

The goal is to obtain the rosters (paths) with negative reduced costs in this acyclic network. Recall that there are two families of constraints in the master problem of rostering (see (2)). The first family of constraints ensures that tasks are covered by an adequate number of crew members, while the second type of constraints ensure that exactly one roster is assigned to each crew member. Let λ_t denote the dual value of the first family for task t, and let μ_c denote the dual value of the second family for crew member c. To capture the reduced cost, the cost associated with arcs is then substituted by $c'_{s_c,t} = -\mu_c$ for arcs $\left(s_c,\ t\right)$ and $c'_{t,t'} = w_{t,t'} - \lambda_t$ for arcs $\left(t,\ t'\right)$. The negative reduced cost constraint becomes

$$-\lambda_c - \sum_{t \in X, t' \in X} \left(w_{t,t'} - \lambda_t\right) < 0,$$

where X is the set of all tasks that can possibly be included in a roster for crew member c.

Efficient computational techniques for embedding this negative reduced cost constraint into the constraint programming based column generation can be found in by Fahle, Junker, Karisch, Kohl, Sellman, & Vaaben, 2002. An experiment based on real-world data shows that with constraint programming the search space is reduced, and so is the computation time. In an instance consisting of 65 crew members, 165 pre-assigned tasks, and 250 tasks to be assigned, it takes 3,162 seconds for constraint programming based column generation to solve the problem to optimality.

3 State-of-the-art in Information Systems

In the previous section we discussed the models and solution methodologies for various crew management problems. In this part we focus on the underlying data management systems and we review the offerings of the leading software providers.

3.1 Data Management Information Systems

The airline industry is constantly undergoing changes and the ability to react and adjust swiftly is imperative. The volume of collected and stored data is rapidly increasing and data management capabilities become not only a key for success but also of survival. Superior optimization applications and durable airline's data management information systems are key success factors in such an environment. Many legacy and startup airlines are realizing the importance of data management and resource planning, which is reflected in recent investment in these fields despite, or maybe because of current financial difficulties. Controlling the large number of vital data and its updates is the key to reducing costly data redundancies and misleading analytics, especially within enterprises like airlines that have multiple systems depending on the same information. Accordingly, data management is evolving from scattered databases toward consolidated data warehouses and lately toward integrated enterprise data management systems.

3.1.1 Data Warehouse

A modern data warehouse is more than just a big database. It also includes information management software to extend the value of the data by collecting and distributing necessary information and moving real-time insights closer to all interested users. By consolidating information as-sets and applying real-time analytics to turn information into intelligence, it is possible to deliver information in a selective and transparent way. Data warehouses are becoming much more real-time-oriented. Access to the data warehouse is becoming more and more a part of front-end transactions and not anymore a traditional backend transaction. Traditionally, an airline data storage system consists of several databases or data marts organized around specific departments that are poorly connected with non-standardized and non-efficient data interfaces. Substantial customization, code changes and testing across the entire system are required whenever even a small modification is needed in a department. The basic requirement of a modern airline data management system is that legacy data storage system needs to move toward a more efficient and better structured data warehouse. Such a system requires a good architecture that allows:

- Modularity - enhancement of one department's domain without impacting other domains,
- Migration - secure conversion from legacy,
- Integration - combining selected in-house and external systems and applications,
- Harmonization – data synchronization across different operational systems and data sources.

3.1.2 Integrated Enterprise Data Management System

A data management information system is quite often thought of as part of a data warehouse, but it actually sits on top of it. An integrated enterprise data management system also includes enterprise application integration capabilities as well as other services like user interfaces, data mining, re-porting capabilities, etc. Solid raw data and strong analyses are essential for making timely and appropriate business decisions and maintaining a competitive edge. An integrated solution, as the one shown in Figure 4, should collect data from multiple systems into a consistent, accurate data warehouse.

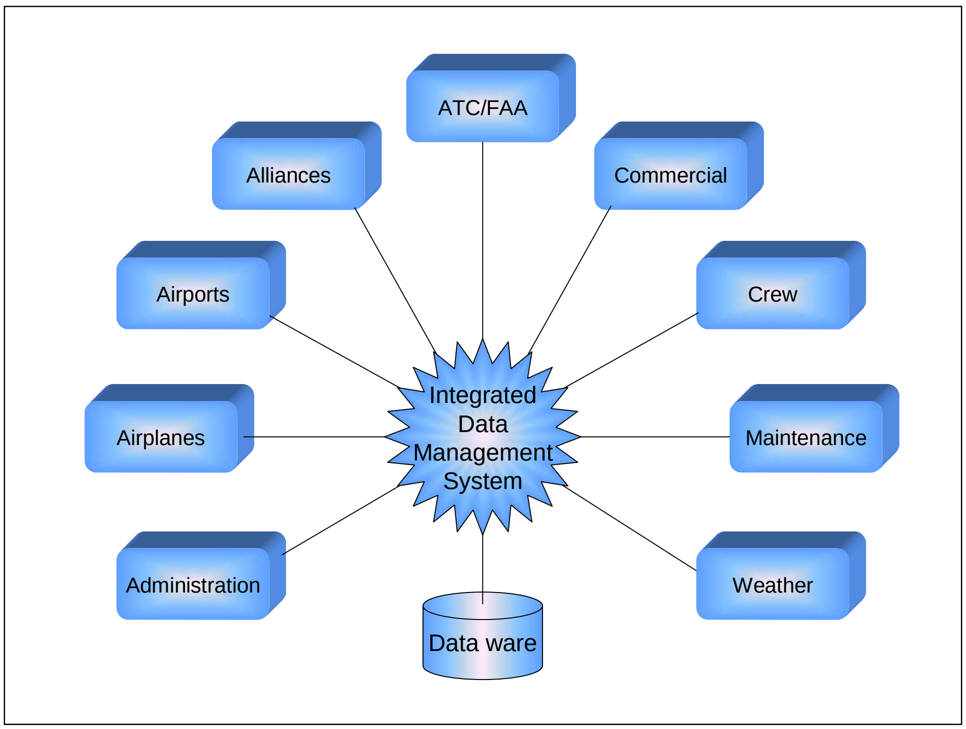

Figure 4: An Integrated Enterprise Data Management System

Selected data is then presented in multiple business views, such as views for dispatchers, ATC managers, crew managers and maintenance coordinators to enable corresponding planners and managers to rapidly retrieve information during planning and operations and make more timely and informed decisions. Accounting and finance, route profitability, human resource, materials management, enterprise management, and customer relationship management should not be forgotten as a part of a comprehensive integrated data management system landscape.

An integrated data management system provides organizations also with the flexibility, reliability, and agility to respond to different vendor's software consolidation needs, to facilitate development of one integrated solution, and to reduce the overall cost of existing and potentially new data integration projects. Such a system needs to provide the following functionalities.

- *Connectivity*: Leverage all data, regardless of its source. With the "connected world" piping in data from all areas, connectivity is the foundation of enterprise data integration. For example, connectivity helps a crew tracking application to get in-flight aircraft positions by capturing and correlating data received from both airline systems and FAA sources.
- *Monitoring*: Profile, cleanse, and monitor data to assure standardized, consistent, and reliable information.
- *Capturing*: Capture and propagate data changes in real-time to assure data integrity, consistency and credibility.
- *Processing*: Extract, transform, and load data from across the enterprise to create consistent and accurate information.
- *Data caching*: Enables different decision support tools a quick access to critical data. For example, data caching is used to enable crew managers in airline operations centers to quickly assemble views of the overall system status, delivering responsiveness that simply was not available from its relational data marts.
- *Data mining*: Helps users to discover patterns, dependencies and consistencies of huge data sets. It also allows queries and use data across multiple systems without the physical movement of source data.
- *Master data management*: Quickly and reliably create a unified view of enterprise data from multiple sources.

3.1.3 Crew Data Management Information Systems

A fast access to the most current operational data drives the success of an airline's crew disruption management system. A crew data management system needs to be able to deal with such a dynamic environment in real-time. At the same time, this system should support the entire planning process, which is divided into long term planning, short term planning (scheduling and rostering), tracking and pre-operations crew schedule repairing. While speed is not crucial for planning data management except short term planning, it requires the storage and distribution of huge amounts of crew related data.

It is believed that a global recovery plan for the three resources (aircraft, crew, and passengers) should be obtained simultaneously due to their related nature. For example, if a flight is canceled and the next flight is not covered by a crew or an airplane, a recovery plan for a crew would be of no use, if an aircraft is not recovered and positioned at the right place. The combination of sophisticated operations research techniques and an integrated data management system substantially increases the quality and accuracy of a crew decision support system. A comprehensive data management information system, in order to support crew planning and operations, needs to provide, among others, the following functionalities:

- manage all flight schedules, crew assignments, crew training, and crew time tracking,
- manage all key data including aircraft, airports, bases, limitations, and payroll policies,
- maintain all relevant crew data including contact information, qualifications, and payroll data,
- define and use multiple regulation templates,
- generate pairings and rosters for flight and cabin crew,
- use actual and future scheduled flights to check for crew feasibility,
- modify or swap pairings and rosters to clear feasibility conflicts, and
- drag-and-drop environment.

3.1.4 Case Studies

To name just a few, the following are examples of successful implementations and future plans for data warehouse upgrades and development of an integrated data management system.

Continental Airlines

Continental Airlines was proclaimed a winner in the 2003 Data Warehousing Award competition organized by the Data Warehousing Institute (TDWI) for their effort to streamline their data management and reporting practices across their entire enterprise. In the past, each department at Continental had its own approach to data management and reporting. The airline lacked a corporate data infrastructure that a broad range of employees could use for quick access to key insights about its business. Information was inconsistent across the different areas, and root causes were hard to identify.

In 1998 a Teradata-based data warehouse was implemented to bring consistency to data management and reporting at Continental. Five years later the warehouse consisted of 27 source systems including schedules, inventory, reservations, airline tickets, airline revenue flown information SOCC (airline operational system), One Pass (frequent flyer program), customer profiles and demographics, aircraft maintenance, alliance data, employee/crew payroll, and customer care. In addition, business rules are applied to the data at an enterprise level to derive additional information such as the true origin and destination of a trip. Even further, more than 1,000 users have fast and easy access to enterprise information for strategic and tactical decision support and full-spectrum business intelligence. The enterprise data warehouse demonstrates best practices in a number of ways.

- *Architecture*: Continental developed a model that simplifies joining different subject areas in real time and allows a single view of information.
- *Training*: Users receive extensive training so they have the skills to find the data they need and come up with enterprise-wide questions that could not have been answered before.
- *Transparency*: Profitability and loss are associated with any decision no matter how small.
- *Standardization*: A data warehouse steering committee works to standardize definitions and data metrics so that all users can use data from across subject areas.
- *Automation*: All transformation and loading processes are fully automated and automatically monitored. Only exceptional situations require human attention and intervention.

Continental reports multiple millions in cost savings and estimates that revenue has increased by multiple millions of U.S. dollars in 2002 through just four applications running on the enterprise data warehouse: revenue management, fraud detection, crew payroll, and customer relationship management.

Air France

At the AGIFORS 2006 Crew Management conference in Honolulu, Air France presented a plan to phase out their crew data management legacy system. The ambitious plan contains stable steps to assure a secure and smooth migration process through a period that is estimated to be approximately ten years. The proposed migration suggests a modular transition process in order to allow stability during ups and downs of the airline industry. The goal was to define separate migration projects that are not dependent to each other. Each migration project is designed to bring value in order to pay for the phase-out. Projects may be jointly accomplished together with their sister company KLM, as well as with other airlines and providers. This is an ongoing initiative at Air France that has a high business priority, which becomes even more important with new airline operations center plans that are starting to shapeup. The company's gradual transformation approach has already taken several steps toward this direction:

- continuous slimming down of the legacy data management systems,
- stop adding new data into the legacy data management systems,
- removing concepts that are better served by new data management systems,
- developing real time pairings tracking,
- centralizing rules,
- defining business intelligence and production key performance indicators,
- developing manpower planning.

Air France believes that the integrated enterprise data management system should be owned and maintained by the company because it allows the evolution of the company's IT and represents a foundation of all business processes and decisions. Additionally, it facilitates the inclusion of decision support tools such as internally and externally developed operations research applications.

Delta Airlines

The airline crew data management system contains critical business and safety information and usu-ally airlines are not ready to share such an asset, as seen in the above examples. However, Delta recently asked the U.S. bankruptcy court for permission to strike an outsourcing deal. According to the plan, Delta Air Lines will turn over the bulk of its back-office computer operations to IBM under a seven-year outsourcing agreement that Delta hopes will contribute to the savings it needs to successfully emerge from Chapter 11 bankruptcy. The airline wants IBM to assume operation of the computers and software that support Delta's customer reservations, business record-keeping, flight management, and maintenance tracking systems. Delta's agreement with IBM is the second major technology outsourcing deal involving a major U.S. airline. At the beginning of 2006, United Airlines inked a ten-year desktop and infrastructure management deal with Electronic Data Systems.

3.2 The Software Market

IT infrastructure and the corresponding complex software applications are an indispensable part of the airline industry today. Computerization is one of the key business areas for each airline. In the early days computerization was mainly oriented toward the automation of manual work and maintenance of basic databases, primarily the reservation system. Today, the main goals are optimization and integrated enterprise data management systems. Trends in the airline industry have created a market for firms capable of using technology to increase productivity and fully integrate operations management, while gradually phasing out existing legacy systems.

For more than forty years, the software providers have been offering solutions in areas such as revenue management, pricing, flight scheduling, cargo, flight operations and crew scheduling. In the early sixties, Sabre Holdings, at that time part of American Airlines, has transformed the airline industry through technological advancement by creating the first passenger reservations system. In these early days, airline software was mostly developed by airlines' IT departments them-selves and by external consultants mostly coming from academia. Probably the first independent software provider with an established list of clients was SBS International, founded in the early seventies. The SBS' launch product was a rostering application with bidding capabilities, which is where the name SBS originated – Schedule Bidding System.

Depending on an airline or a software provider, crew management applications may have different structures. Two types of crew management applications are considered in this chapter in order to support two different processes, planning and operations. The main planning tools are manpower planning in the form of pairing and rostering applications, while the main operations tools are crew tracking and crew recovery applications. The following analysis of major crew software providers is focused around these main offerings and their optimization capabilities.

3.2.1 In-house Crew Management Software

In the past, many airlines had created large operation research and IT teams who developed, and are still developing their own crew management software applications. These were mostly big legacy carriers like Air France, American Airlines, British Airways, Delta Airlines, Emirates, and United Airlines, to name just a few. Operation research resources are scarce and expensive and it is hard to keep a large operation research group in the environment where profit margins are very thin. Recently, some airlines decided to de-crease their internal operations research and IT departments to steadily grow outsourcing and to look toward the best-of-the-breed operation research applications available on the market. There are many reasons behind this tendency, but one of most important is to become lean and to cut costs through outsourcing. Delta Airlines recently even dissolved its own operation research group, Delta Technologies, in order to cut costs and get quickly out of bankruptcy protection. However, to be competitive, a critical mass of more highly skilled operation research people able to cope with technology advancements is still required to be an integral part of a carrier.

Smaller airlines usually prefer buying a product that is close to satisfying their needs rather than of investing time and money in developing an application that will exactly fit their needs. Some airlines, like Northwest (acquired by Delta Airlines in 2008) and Southwest, are deciding to buy the source codes or pay contractors to develop for them strategically important applications that they have insufficient resources to develop and then employ their small operation research teams to support them. However, there are still some airlines like American, Emirates, and United that are growing their operation research teams.

While operation research team landscape is still shifting and adjusting, the most probable scenario for the future is that most airlines will continue buying well established and proven crew planning products like pairing and rostering, as well as the cutting edge technology required for disruption management applications. There are many atypical airline specific problems that will still need to be addressed using internal operation research expertise. The role of internal operation research resources is also crucial to performing data analysis, to analyzing planning and operations business flow, to understanding the external applications and to helping with their selection, as well as to adjusting these applications to deal with changes, which are inevitable in the airline industry.

3.2.2 Software Providers and Their Products

Today, there are several crew scheduling and rostering software providers offering a large number of crew management software packages, ranging from the long term manpower planning through pairing and rostering, to the day of operations decision support systems. Airline crew management applications provide many benefits to their users. However, the quality and complexity of these offerings are not always the same and may include one or more of the following: graphical user interfaces, database services, reporting capabilities, web services, and simulation and optimization tools. Optimization tools may consist of simple heuristics or complex advanced mathematical methods. These days, more powerful computers and sophisticated operation research methods enable significant improvements of the quality and speed of the software applications. Consequently, the airline crew management process becomes much easier and even modified in a way that obtained results are of a better quality and more quickly achieved. The user may play with different scenarios and solve them much closer to the actual operations. Making planning decisions closer to operations has indirect impact on the quality and robustness of these decisions by allowing for more accurate information to be taken into account.

In the following section we list, in the lexicographical order, the major software vendors on the market and their main offerings. Some of them are major airline information technology software companies while others are specialized crew management software divisions within a bigger aviation or non-aviation company. Finally, there are a couple of small, independent companies that offer some components of an integrated crew management solution. While planning software packages have been available for quite some time and offered by most providers, more recent research and development efforts are focused on the operations.

AIMS Inc.

AIMS Inc. offering covers a wide range of applications including crew management tools like manpower planning, pairing, and tracking. The crew pairing product is probably one of the most installed AIMS crew management products. Optimization capabilities of their crew management tools are debatable, but lower benchmark results are often compensated by lower price of their products. They developed a solid customer base but it seems that it has been shrinking lately. Their web site is www.aims.aero with the head office registered in United Kingdom.

Advanced Optimization Systems, Inc. (AOS)
AOS has been developing and implementing crew optimization software and solutions for the airlines since 1992. This small company has established decent reputation and developed sophisticated optimization tools for pairing and line optimization. Despite very limited offering, they managed to develop a respectable customer base that includes small, medium, and large air-lines. The company is based in New Jersey, US and their web site is www.aos.us.

Avient, an IBS Group Company
Avient provides solutions, services and products to the global travel, transportation, and logistics industry. The company has over 15 years of experience in the design, development, and implementation of software solutions worldwide. Avient has built separate offerings for an established airline (Avient Velocity) and for a start-up airline (Avient Take-Off). The Velocity product contains the pairing, rostering, tracking, and recovery modules. While tracking is their key product, it seems that their offerings do not include sophisticated optimization tools. This is probably the main reason for the modest airline customer base of Avient. The company is based in Berks, UK and their web site is www.avientsolutions.com.

Jeppesen, A Boeing Company
Elrey Jeppesen founded the navigation chart company in early thirties. Today, the company is part of Boeing and represents one of the most comprehensive aviation software companies. Thanks to the recent acquisition of Carmen Systems AB and SBS International, now called Jeppesen AB, Jeppesen offers a comprehensive line of crew management products consisting of long and short term planning tools, operations tracking and decision support tools, administrative tools and web access services supported by an advanced rule engine and database services. With the acquisition of SBS the company acquired strong data management while Carmen's optimization capabilities are one of the best on the market. Combined Jeppesen airline customer base is one of the largest, including very small but also almost all of the largest airlines in the world. The range of operation research techniques implemented in their products spans from single business rule based heuristics, to more advanced heuristics and meta heuristics, and to highly sophisticated mathematical programming methods. Also, as a result of the collaboration with academia – mainly the Massachusetts Institute of Technology and Georgia Institute of Technology – an advanced simulation engine is used in concert with their optimization tools in order to evaluate different scenarios and calibrate optimization parameters. Jeppesen is a global company with the corporate headquarter in Colorado, US. The airline crew management offices are based in New York, US and Gothenburg, Sweden. The company's official web site is www.jeppesen.com.

Lufthansa Systems
Lufthansa Systems, part of Lufthansa Group, is one of the world's leading IT service providers for the airline and aviation industry. It has around 4,550 employees in several locations in Germany and offices in 17 countries. Its portfolio addresses all airline business processes and includes services for passenger and cargo handling, flight operations, and aircraft maintenance and repair. As a system integrator with one of the state-of-the-art data centers in Europe, Lufthansa Systems covers the entire spectrum of IT services including consulting, application development and implementation, and reliable 24-hour operation. The offering also includes a complete crew management product suite with crew tracking as the key module. Their optimization capabilities seem modest and they are mostly based on heuristics rather than on advanced mathematical pro-

gramming methods. The company's headquarter is in Kelsterbach, Germany and their web site is www.lhsystems.com.

Kronos Incorporated
Kronos provides general workforce management applications. These tools help organizations staff, develop, deploy, track, and reward their workforce. As part of an acquisition of AD OPT Technologies in 2004, Kronos obtained Altitude division specialized in airline crew management. Kronos/Altitude's most known applications are the crew pairing and preferential bidding system and lately the bidline and manpower planning. The acquisition of Mercury in 2003 by AD OPT brought to the company a crew tracking application, but it seems that it did not generate too much interest in the market. The strongest point of Altitude's crew planning applications is the quality of the optimization engines. They are the first provider in this field that introduced and commercialized the branch-and-price optimization approach. Once the leading resource optimization provider, Altitude is losing lately its competitive advantage due to the lack of investment in research and development. Kronos headquarter is in Minneapolis, US while the airline crew management software office is in Montreal, Canada. Their web site is www.kronos.com.

Navitaire
Navitaire Inc. provides reservations, direct distribution, revenue protection, decision-support and passenger revenue accounting services to the airline industry through its hosted delivery model. With the latest acquisitions of Caleb Technologies Corporation and Forte, the product portfolio is significantly expanded by the crew management products. Among the key crew management software applications, acquired from Caleb, is the recovery suite system. Other applications comprise manpower, pairing, rostering and tracking tools. Additional crew planning and tracking tools are supplied by the acquisition of Forte. The company's solid customer base includes several large airlines and some leading low-fare and midsize airlines. A wholly owned subsidiary of Accenture, Navitaire is headquartered in Minneapolis, US with the crew management office in Austin, US. Its homepage is www.navitaire.com.

Navtech, Inc.
Navtech, Inc. was originally incorporated in the State of New York in 1981 and then reincorporated in the State of Delaware in 1987. Since then, together with their subsidiaries, Navtech developed a wide range of airline software offerings. While their focus is on charting and flight planning, over the last couple of years they have been investing in the development of crew planning products. The crew planning offering includes a pairing optimizer as well as preferential bidding system. Today, Navtech is an important airline software provider with a large customer base. However, their current optimization capabilities seem to be limited and their crew planning software customer base is still modest. Navtech head office is in Monterey, US while principle operations are based in Waterloo, Canada. Their web site is www.navtechinc.com.

Sabre Airline Solutions
Sabre, an ex-subsidiary of American Airlines, has been in business for more than 40 years. Today, it is a leading global company that provides airline products and services, reservations and departure control tools, as well as consulting services. Their airline crew management suite is an integrated end-to-end resource management solution designed to handle the airlines' needs. The suite addresses all phases of crew management operations from long-term crew resource planning to day-of-operations, more precisely manpower planning, pairing, rostering, web access, crew tracking, and recovery. Sabre's applications are very data processing oriented, but they have also developed a wide range of sophisticated operation research tools. Their optimization

tools are state-of-the-art and they are planning to stay a big player in this game. Their commitment to this objective is proven by their willingness to invest in its improvement, especially following the recent Boeing acquisitions. This Texas, US based company has the largest share of the airline resource management market and is definitely the major airline information technology provider. Their web site is www.sabresolutions.com.

3.2.3 Providers Consolidation

Many industries are facing an increasingly competitive environment due to globalization, deregulation, increased complexity and consolidation. Consolidation through mergers and acquisitions is a major trend in many industries. A central goal of most mergers has been to improve investment returns through cost cutting, productivity gains, and economies of scale. The airlines industry is not an exception; the latest examples are the mergers of America West and US Airways, Delta and Northwester Airlines, and the looming merger of United and Continental Airlines. Their objectives are expanded routes, increased fares through reduced capacity, improved service and a more stable operating environment. In order to have more stable operating environments, airlines are looking more toward an integrated solution from one provider, which offers multiple modules instead of the best-of-the-breed isolated solutions that are hard to integrate. It is likely that software providers will follow this tendency. The latest acquisition trend confirms such a direction and it will not be surprising to see more acquisitions in the future.

A merger and acquisition activity should target a market presence increase, and fixed and operations costs improvements. Increased presence may be achieved by removing a competitor from the market, increasing the customer base, and offering a more complete portfolio of products. Costs can be reduced by removing redundant overheads, product overlaps, and reorganizing research and development to better use resources and reduce development costs. In recent years, several important crew software provider acquisitions took place. Boeing acquired a number of companies as part of its vision to offer airlines a complete suite of products and services in support of the aircraft they buy from them. In 1999, 2000, and 2001 Boeing acquired The Preston Group, Jeppesen Sanderson and SBS International, respectively. In 2005 Carmen Systems AB acquired OpCom and just a year later Carmen was acquired by Boeing as well. Today, they are all part of one single family - Jeppesen, A Boeing Company. In 2001, Sabre Airline Solutions acquired David R. Bornemann Associates as part of its strategy to penetrate the small and medium sized airline software solutions market. In 2004 they acquired the Stockholm-based company RM Rocade as well. In 2003 Adopt Technologies Inc. acquired Mercury Scheduling Inc. and in 2004 Adopt itself was acquired by Kronos Incorporated. In 2004 Navitaire acquired Caleb and in 2006 they added Forte to their list.

3.3 Conclusions

What all these acquisitions mean and how successful they are is hard to say and predict. It is hard to find any public data about success of the above acquisitions. Available information shows mostly the market aspect of an acquisition and may create the wrong impression about the real financial outcome. The process of consolidation is beneficial if it facilitates increased efficiency on both sides, acquired and acquirer organizations, or at least for one that survives. While many of these transactions have a potential for efficiency gains, it seems that it is not systematically exploited. However, it looks that current trend unveils a typical big-fish-eats-small-fish pattern where small independent providers get absorbed by larger companies. The feeling is that the ultimate objective of larger corporations such as Boeing, Lufthansa, and Sabre is not necessarily increased efficiency, but rather an expansion of the product and service portfolios in order to offer an enterprise crew management solution.

4 References

Andersson, E., Housos, E., Kohl, N., & Wedelin, D. (1998). Crew Pairing Optimization. In G. Yu (Ed.), *Operations Research in the Airline Industry* (pp. 228-258): Kluwer Academic Publishers.

Barahona, F., & Anbil, R. (2000). The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming, 87*(3), 385.

Barahona, F., & Anbil, R. (2002). On some difficult linear programs coming from set partitioning. *Discrete Applied Mathematics, 118*(1/2), 3.

Barnes, E., Chen, V., Gopalakrishnan, B., & Johnson, E. L. (2002). A least-squares primal-dual algorithm for solving linear programming problems. *Operations Research Letters, 30*(5), 289-294.

Barnhart, C., Johnson, E. L., Anbil, R., & Hatay, L. (1994). A Column Generation Technique for the Long-Haul Crew Assignment Problem. In T. A. Cirani & R. C. Leachman (Eds.), *Optimization in Industry II*: Wiley.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research, 46*(3), 316-329.

Benders, J. (1962). Partitioning procedures for solving mixed-integer programming models. *Numerische Mathematik, 4*, 238-252.

Dantzig, G., Ford, L., & Fulkerson, D. (1956). A Primal-dual Algorithm for Linear Programs. In H. Kuhn & A. Tucker (Eds.), *Linear Inequalities and Related Systems* (pp. 171-181): Princeton University Press.

Ehrgott, M., & Ryan, D. (2002). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis, 11,* 139–150.

Fahle, T., Junker, U., Karisch, S. E., Kohl, N., Sellmann, M., & Vaaben, B. (2002). Constraint programming based column generation for crew assignment. *Journal of Heuristics, 8*(1), 59-81.

Fisher, M. (1985). An applications oriented guide to Lagrangian relaxation. *Interfaces, 15*, 10-21.

Gamache, M., & Soumis, F. (1998). A Method for Optimally Solving the Rostering Problem. In G. Yu (Ed.), *Operations Research in the Airline Industry* (pp. 124-157): Kluwer Academic Publishers.

Gamache, M., Soumis, F., Villeneuve, D., Desrosiers, J., & Gelinas, E. (1998). The preferential bidding system at Air Canada. *Transportation Science, 32*(3), 246-255.

Hu, J., & Johnson, E. L. (1999). Computational results with a primal-dual subproblem simplex method. *Operations Research Letters, 25*(4), 149-157.

Junker, U., Karisch, S. E., Kohl, N., Vaaben, B., Fahle, T., & Sellmann, M. (1999). A Framework for Constraint Programming Based Column Generation. In *Principles and Practice of Constraint Programming-CP'99* (Vol. 1713, pp. 261-274).

Klabjan, D., Johnson, E. L., & Nemhauser, G. L. (2000). A parallel primal-dual simplex algorithm. *Operations Research Letters, 27*(2), 47-55.

Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E., & Ramaswamy, S. (2001). Solving large airline crew scheduling problems: Random pairing generation and strong branching. *Computational Optimization and Applications, 20*(1), 73-91.

Lavoie, S., Minoux, M., & Odier, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research, 35*, 45-58.

Lettovsky, L. (1997). *Airline Operations Recovery: An Optimization Approach.* Ph.D. Thesis, Georgia Institute of Technology, Atlanta.

Lettovsky, L., Johnson, E., & Nemhauser, G. (2000). Airline crew recovery. *Transportation Science, 34*, 337-348.

Ryan, D., & Foster, B. (1981). An Integer Programming Approach to Scheduling. In A. Wren (Ed.), *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling* (pp. 269-280): Elsevier Science B.V.

Shebalov, S., & Klabjan, D. (2006). Robust airline crew pairing: Move-up crews. *Transportation Science, 40*, 300-312.

Vance, P., Barnhart, C., Johnson, E., & Nemhauser, G. (1997). Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research, 45*, 188-200.

Wedelin, D. (1995). An Algorithm for Large-scale 0-1-integer Programming With Application to Airline Crew Scheduling. *Annals of Operations Research, 57*, 283-301.

Yen, J., & Birge, J. (2006). A stochastic programming approach to the airline crew scheduling problem. *Transportation Scien*ce, 40, 3-14.