Expert Systems With Applications 55 (2016) 133–144

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A novel column generation strategy for large scale airline crew pairing problems

Bahadır Zeren[a,b], Ibrahim Özkol[a,*]

[a] Istanbul Technical University, Faculty of Aeronautics and Astronautics, Istanbul, Turkey
[b] Turkish Airlines, Directorate of Corporate Development and Information Technologies Department, Istanbul, Turkey

ARTICLE INFO

Keywords:
Integer programming
Column generation
Set-covering
Crew pairing
Heuristics

ABSTRACT

A crew pairing is a sequence of flight legs beginning and ending at the same crew domicile. Crew pairing planning is the primary cost-determining phase in airline crew scheduling. Optimizing crew pairings of an airline timetable is an extremely important process which helps to minimize operational crew costs and to maximize crew utilization.

There are various restrictions imposed by regulations or company policies that must be considered and satisfied in crew pairing generation process. Keeping these restrictions and regulations in mind, the main goal of the optimization is the generation of low cost sets of valid crew pairings which cover all flights in airline's timetable.

For this research study, already existing works related to crew pairing optimization are examined and a new column generation strategy, a pricing network design and a pairing elimination heuristic are developed as a contribution to the previous studies. In the proposed strategy, the main problem is modeled and solved as a set-covering problem and the pricing sub problem is modeled as a shortest-path problem which is efficiently solved over a duty-flight overnight connection graph by the combined usage of heuristic and exact algorithms. The proposed strategy has been tested with real world data obtained from Turkish Airlines and it is seen that it is capable of generating very competitive solutions compared to current practices in Turkish Airlines. It is also observed that there are various advantages of proposed solution approach such as sensitivity to penalty coefficients, generating less deadheads, very close solution times with a single threaded software and light weight hardware.

1. Introduction

The ultimate aim of airline crew scheduling is assignment of crew members to flights so as to satisfy all crew need of all flights in airline's timetable. There are some important key performance indicators (KPI) of outputs of crew scheduling processes like total man day, number of overnights, deadhead time, ground time etc. It is always desired to reduce these KPI values as much as possible. By doing this company can increase the crew utilization and basically keep crew on air as much as possible and avoid losing time because of other reasons except flights.

There are two main sub processes of strategic crew scheduling. First one is crew pairing generation which constitutes the main topic of this study. In crew pairing generation process, flight sequences which constitute flight duties and then crew pairings are generated so as to cover all flights in airline's timetable. Optimization methods play very important role in crew pairing generation process because of critical cost factors which must be minimized. Crew costs are the second biggest cost after fuel costs for airline companies. This situation makes crew scheduling and especially crew pairing optimization significantly important operation which must be carried out carefully. Even small amount of percentages can cost millions of dollars.

The second sub process is crew rostering process. In this phase, crew assignment to all crew pairings generated in previous crew pairing generation phase is done. Crew rostering has less impact on total crew costs compared to crew pairing generation process. It mainly affects workload balance between crew members and aims to increase fairness of the plan for crew members. Some US companies does not carry out rostering process. Instead, crew assignment is done by a seniority based bidding process and then a market style system that provides possibility of trip trading.

Eventually the importance of optimum crew scheduling cannot be overstated as all airlines operate in a very complex environment. In addition, with increasing competitiveness in the marketplace, airline companies are in a position to better manage their

* Corresponding author. Tel.: +90 212 2853111.
E-mail addresses: bzeren@gmail.com (B. Zeren), ozkol@itu.edu.tr (I. Özkol).

http://dx.doi.org/10.1016/j.eswa.2016.01.045
0957-4174/

Table 1
Sample crew pairing.

Flight duty nr	Duty start	Duty end	Dep. time	Arr. time	Dep.	Arr.
1	25.03.15 10:00		25.03.15 11:15	25.03.15 13:20	IST	TLV
			25.03.15 14:20	25.03.15 16:25	TLV	IST
		25.03.15 23:05	25.03.15 17:25	25.03.15 22:35	IST	ALA
Rest period	–	–	–	–	–	–
2	26.03.15 19:45	27.03.15 03:25	26.03.15 20:45	27.03.15 02:55	ALA	IST
Rest period	–	–	–	–	–	–

expenses using effective flight and crew scheduling techniques (Zeren & Özkol, 2012).

1.1. Fundamental definitions and concepts

Flight (flight leg or leg) is the primary data unit for the crew scheduling processes. A *duty* is the time period consisting of one or more flight(s). Between two flights in a duty, there is always a *connection time* which has minimum and maximum limits. After end of a duty, regardless of the location, the crew must have a predefined *rest period*. A *pairing* is a sequence of flights which begins and ends at a particular domicile and consists of one or more duty(s). If a crew member flies as a passenger instead of as a cockpit or as a cabin attendant, this flight is called as dead-head flight for that crew member. Dead-head flight is a factor that reduces passenger transport capacity and efficiency of crew utilization. Therefore minimization of the number of dead-head flights is always desirable in the crew scheduling process. A sample crew pairing which consists of two duties and four flights can be seen in Table 1.

In terms of rules and regulations, durations of duties constitute one of the fundamental constraints in the airline crew scheduling process. Maximum duration of a duty changes between 11-14 hours depending on duty's start time. Other implemented fundamental constraints; minimum-maximum connection time which can be between 30-240 minutes, minimum rest time which can be between 8-14 hours depending on the duration of the duty, maximum pairing time which is 4 days etc. All these rules can be found in Republic of Turkey Ministry of Transport, Maritime Affairs and Communication, 2014.

1.2. Related works

In literature, airline crew pairing problem is mostly modeled as set-covering or set-partitioning problems. These models perfectly fit almost all optimization modeling needs that arise in airline crew scheduling. Both of them are combinatorial and are proven to be NP-complete (Garey & Johnson, 1979).

There are two main approaches to solve these kinds of problems: 1- Heuristic methods. 2- Mathematical (LP based) methods. The most widely used heuristic method in this field is genetic algorithms which is basically a search heuristic that mimics the process of natural selection. There are two main column generation strategies used in GA approaches. The first strategy is offline column generation which is based on generating huge amount of columns (crew pairings) just before optimization phase. Then optimization phase takes place and the best subset of these pre-generated columns is selected (Beasley & Chu, 1996; Kornilakis & Stamatopoulos, 2002; Zeren & Ozkol, 2012). The other strategy is based on generating columns during optimization phase. In this strategy columns are generated from choromosomes (represent solutions in GA context) using special heuristic (Aickelin, 2002; Ozdemir & Mohan, 2001).

Nevertheless metaheuristics can give high quality solutions for only small sized problems. In study of Zeren & Ozkol, 2012 it is shown that GA could generate solutions around 0.04 % far from global optima for an aircraft fleet which has around 710 monthly flights. Further experiences showed that increasing the number of flights reduces the solution quality and GA starts having difficulties on decreasing deadhead time efficiently. For instance, the same GA code generated outputs with 7% gap for a monthly problem which has around 7000 flights.

On the mathematical side, different column generation strategies exist as is seen in heuristic approaches. Some earlier studies which rely on mathematical methods implemented partial enumeration for large scale problems. Thus they increased performance by reducing problem size (Baker, Bodin, Finnegan, & Ponder, 1979; Klabjan, Johnson, Nemhauser, Gelman, & Ramaswamy, 2001). The second strategy which is called "column generation" or "dynamic column generation" is the most widely used and studied technic in last two decades. It provides a very efficient way to generate only necessary columns to improve objective value further. Column generation method divides whole problem into two parts: Master problem and sub (pricing) problem. In this study master problem is modeled as set-covering problem and the dual variable information (shadow price) of each row (flight leg) is calculated by solving it. The pricing sub problem is modeled as shortest-path problem and new columns (crew pairings) are generated by finding columns with least negative reduced cost using dual information obtained from master problem. For a comprehensive details about column generation readers may refer to Desrosiers & Lübbecke, 2005; Lübbecke, 2011; Lübbecke & Desrosiers, 2005; Vanderbeck, 2005.

Even though set-covering and set-partitioning problems require integer solutions at the end, the core mechanism of column generation relies on LP. Calculation of dual variables and reduced costs etc. require LP solutions. There are a few strategies that are used to obtain integer solutions for such problems. The first approach is based on using some special heuristic which eliminate or select pairings based on their linear variables and reduced costs like in study (Anbil, Tanga, & Johnson, 1992). In reference (Andersson, Housos, Kohl, & Wedelin, 1998; Gustafsson, 1999; Medard & Sawhney, 2007; Wedelin, 1995) a sort of rounding heuristic is used in a system which uses Lagrangian relaxation and sub-gradient optimization to solve master problem.

Another strategy is based on using branch and bound algorithm to get integer solution at the end of column generation phase. But this implementation fails for large problems if there is no special performance increasing heuristic.

The most popular strategy for last two decades is the branch and price approach which is based on generating columns on a branch and bound tree (Barnhart, Johnson, Nemhauser, Savelsbergh, & Vance, 1998). Branch and price is an exact method that guarantees finding global optima but requires employment of special heuristic for performance improvements with a little loss in solution quality. For a heuristic implementation of branch and price it can be refered to Vance et al., 1997. And also for a more updated study and computational results please see Desaulniers & Soumis, 2010. As can be seen from the computational results, even for a fleet with around 7000 monthly flights, branch and price framework needs around 10 hours to get the final solution. It also requires parallel programming technics which are more sophisticated and relatively difficult to implement.

In this study a column generation strategy is proposed which is based on a combined usage of heuristic and exact column generation methods over a duty-flight overnight connection network. The proposed method additionally uses a special duty and pairing elimination heuristic called connection fixing heuristic to improve the performance. As can be seen from the computational results the proposed method is capable of generating high quality solutions compared to current practices in Turkish Airlines.

2. Master model

In literature airline crew pairing problem is modeled with set-covering or set-partitioning models. From these two classical options set-covering model is used in this study. The main reason is that there is a clear analogy between set-covering model and crew pairing problem. Set-covering model perfectly fits and provides all the representation need of the crew pairing problem.

Analogically, in set-covering model, each row represents a flight in airline time table and each column represents a generated crew pairing. Each flight has to be covered by minimum one crew pairing and over covered flights represent deadhead flights.

If one considers that F is the set of flights which appear in the flight schedule and P is the set of crew pairings, then the set-covering model for crew pairing problem can be formulated as follows (Barnhart et al., 2003):

$$Min \sum_{p \in P} c_p x_p \tag{1a}$$

$$\sum_{p:i \in p} x_p \geq 1 \quad i \in F \tag{1b}$$

$$x_p \in \{0,\ 1\} \quad p \in P \tag{1c}$$

In the Eq. (1a), c_p indicates cost value for each crew pairing in P and x_p is the decision variable which indicates whether the crew pairing is in the solution set. Hence, Eq. (1a) gives the total cost of the solution. Eq. (1b) is an inequality constraint which guarantees the full covering of all flights in F. If Eq. (1b) gives a value greater than 1 for a flight in F, that flight is a dead-head flight. And Eq. (1c) represents standard constraints for the problem. There is only one crew domicile for Turkish Airlines so no additional term is required related to crew domicile workload distribution.

2.1. Stabilization

There are several drawbacks emerge from implementation of standard column generation. Some of them are slow convergence (tailing-off effect), instability in the dual variables that are jumping from one extreme point to another (bang-bang effect), producing irrelevant columns especially in early iterations due to poor dual information (heading-in effect) etc. To cope with these problems a stabilized column generation implementation is necessary. In this study a stabilization approach presented in (Du Merle, Villeneuve, Desrosiers, & Hansen, 1999) is followed. This approach adds new variables and constraints to the Eq. 1a and Eq. (1b) which penalize extreme dual variable oscillations while remaining within the linear programming framework. While overcoming aforementioned problems, a considerable increase in acceleration and therefore decrease in total number of iterations are observed. Stabilization approaches and their details are out of this paper's scope. For more comprehensive information, the interested reader is referred to the works by Ben Amor, Frangioni, & Desrosiers, 2009; Du Merle et al., 1999; Lübbecke, 2011; Lübbecke & Desrosiers, 2005.

3. Initialization

The phase which includes routines that prepares system and data for main column generation loop is called initialization phase. The very first step in this phase is the duty enumeration phase which is critical to build proposed pricing network.

3.1. Duty enumeration

Before building the proposed duty-flight overnight connection network, all possible duties for a given timetable must be generated. For duty enumeration, as can be seen from Fig. 1 a depth-first search code is developed. In the code, G*enerate_Duties* is the main recursive function which allows algorithm to search for all flight connections possible. On line 2 and 3 suitable flight legs are searched and appended to *currentDuty*. On line 4 and 5 *currentDuty* is validated and saved. On line 6 and 7 recursive procedure is recalled to proceed in search space further if length of *currentDuty* allows to append more flight legs.

In the code above *dutyList* is the list of all valid duties generated by the recursive code, *currentDuty* is the set of flight legs that constitute a duty which is validated on line 4.

Size of *dutyList* completely depends on the size of the timetable data and the ruleset used to validate duties. In this study, ruleset obtained from Turkish Airlines crew planning department is used and roughly 420000 duties are generated over around 17000 flight legs.

3.2. Duty-flight overnight connection network

After the generation of all valid flight duties, the pricing network which will be used in solving pricing sub-problem is built. There are two common network models used for crew pairing generation (Vance et al., 1997). First one is flight network model which uses arcs/nodes to represent flights and nodes/arcs to represent departures, arrivals, sources and sinks of the network. This model is more suitable for daily crew pairing problems and also in this model, it is too much difficult to express complicated rule implementation needs requested by airline company.

The second model is duty network model which uses arcs/nodes to represent duties and nodes/arcs to represent departures, arrivals, sources and sinks of the network. This model is more suitable for weekly or monthly problems and it requires pre-generated valid duties. In this model, all kind of rules and validations applied to a single duty period can be done in a straight-forward way using a programming language during duty generation phase. Nevertheless for large size monthly problems this model produces huge amount of duty to duty connections which will produce performance problems and huge amount of memory need. To overcome this issue a relaxed duty network model is proposed in ref (Gustafsson, 1999). It successfully reduces the size of the network but illegal pairings can be generated because, rules applying to duty to duty connections are not explicitly represented in the model. To handle illegalities, a network modification strategy which will prevent regeneration of the illegal pairing again is offered.

In this study, the proposed network model for the pricing sub-problem is called *duty-flight overnight connection* network. To build network, a straight-forward code is developed which checks all valid duty to duty connections by applying the relevant rules. In this network model, two types of arcs and two types of nodes exist. First type of nodes are composed of duties that are starting points of the network and they will be the first duties of new pairings that will be generated. First type of arcs are composed of duty-flight pairs that represent overnight connections and duty nodes are followed by these arcs. The second type of nodes that

```
procedure Generate_Duties(currentDuty, dutyList)
    for each leg that departs from the station that currentDuty
    arrives and departs after the arrival time of currentDuty do
          Append leg to currentDuty;
          if currentDuty is valid
                Append currentDuty to dutyList;
          if currentDuty can have more flights
                Generate_Duties(currentDuty, dutyList);
          Remove leg from currentDuty;
```

Fig. 1. Pseudocode for duty enumeration.

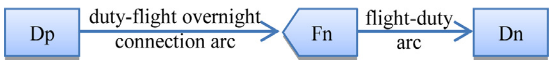

Fig. 2. Nodes and arcs of duty-flight overnight connection network.

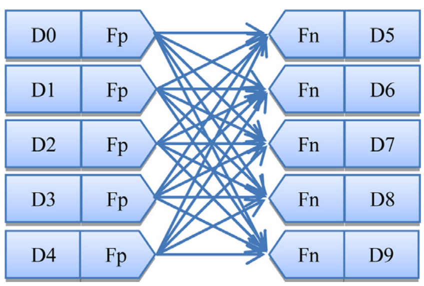

Fig. 3. Fully connected duty network sample.

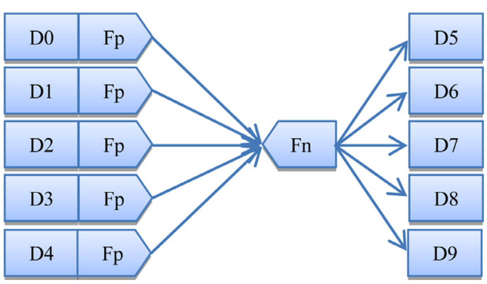

Fig. 4. Duty-flight overnight connection network sample.

follow duty-flight arcs are composed of flights that will be the first flight of the next duty after an overnighting period. And the second type of arcs are flight-duty pairs which follow flight nodes and represents duties which start with that particular flight.

As can be seen from Fig. 2, d_p is a duty node and the starting point of the network. Following duty-flight overnight connection arc represents a valid layover which will end with a new duty that will start with the flight leg f_n. And f_n is a flight node that represent the connection point to all duties that start with flight leg f_n. This network model offers a more compact representation than fully connected duty network model by preventing data repetitions especially for problems which has too many common overnight flight connections. Therefore it allows to reduce the amount of memory needed thus increases algorithm's performance.

For a better comparison of the proposed model to the full connected duty network, assume that there are 5 flight duties which end with the same flight leg f_p and another 5 flight duties which has valid overnight connections to the previous 5 flight duties and start with the same flight leg f_n. A full connected duty network for this scenario can be depicted like below and as can be seen from Fig. 3 there are 25 duty to duty connections in the network. This means that 25 variables needed to build shortest path IP model of the problem over a fully connected duty network.

The same problem can be represented using duty-flight overnight connection network model as shown in Fig. 4. As can be seen from the Fig. 4 there are only 10 arcs are needed to represent the same data with the proposed network model. And also, in the proposed model, rules applying to duty to duty connections are explicitly represented and therefore generating illegal pairings because of illegal duty to duty connections is not possible unlike the study in reference (Gustafsson, 1999).

3.3. Initial pairing generation

Initial pairing generation is done with a depth-first search algorithm that is very similar to the mechanism implemented in duty enumeration phase. The set of flight legs that were used as input is replaced by the set of duties and the algorithm generates valid pairings instead of generating set of valid duties. The only major difference between duty enumeration and this implementation is that the algorithm does not generate all possible crew pairings. Instead it just tries to cover all flights in the timetable and does not try to cover a flight again if it has already been covered. It generates around 8200 crew pairings for a timetable with around 17000 flight legs. After the completion of initial pairing generation phase, a master LP run is started and initial primal and dual information are obtained for all crew pairings and flights in the timetable therefore system and data preparation for the main column generation loop is completed.

4. Partial solution generation approach

One of the most important decision on column generation approaches is choosing the flight or date to generate a pairing for. The column generation strategy proposed in this study is called *partial solution generation approach*. It uses flight connection information obtained from crew pairings generated in previous iterations to create new crew pairings which will constitute a meaningful partial solution to a part of the whole problem. Generated partial solutions do not necessarily cover all flights in airline's timetable. So a single partial solution which is generated at the end of a column generation iteration can not represent a whole solution domain alone. Instead, all crew pairings from all partial solutions are collected in the main crew pairing pool of solution domain which will be the source of the master LP problem.

In each column generation iteration, a partial solution is generated and crew pairings belong to that partial solution are inserted to the main crew pairing pool at the end of the iteration. During the process newly generated crew pairings are compared to the crew pairings from previous iterations to find new flights, meaningful directions to generate new pairings for. To be able to do this a comparison, at least one new crew pairing, is needed to initiate the procedure. To generate this first crew pairings, column generation iterations are started with a depth-first search code which generates one or more new crew pairings heuristically. Then exact pricing model is solved repeatedly until no new flight found to generate new pairings for.

During the process, duties that include flights that are previously covered by the partial solution are omitted for next pricing iterations until the partial solution is fully generated and active column generation iteration is completed. This approach increases

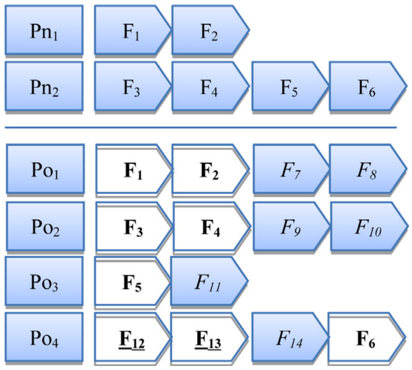

Fig. 5. Crew pairing comparison between iterations.

the algorithm's performance because of possible reduction in the network size for pricing iterations and considerably reduces deadhead time emerges in final solution.

An example of identifying new flights, that new crew pairings will be generated for, is depicted below:

In Fig. 5, there are two new crew pairings (p_{n1}, p_{n2}) that are assumed to be generated during the ongoing column generation iteration. Then the algorithm searches for old crew pairings that include any of the flights covered by new pairings. Old pairings found also must exist in the last LP solution and this decision is done by comparing their LP values with a predefined threshold. After identification of these four old crew pairings (p_{o1}, p_{o2}, p_{o3}, p_{o4}), pricing problem is resolved for flights f_7, f_8, f_9, f_{10}, f_{11}, f_{14} and new crew pairings which will cover these flights are generated and new flights are identified to rerun the pricing problem and so on. f_{12} and f_{13} are omitted because they are assumed to have been covered by one or more previously generated crew pairings during the active iteration. The partial solution generation process lasts until no new flight is found to generate new pairings for and, until its completion, many crew pairings are generated. Generating more crew pairings in each iteration has positive effect on the stabilization of the dual variables obtained.

A pseudocode of partial solution generation approach can be seen in Fig. 6. As can be seen from the line 1, algorithm gets dual variable information of last LP solution and global pairing list as inputs. On line 2, initial pairing generation heuristic is run to get initial pairings of the iteration. On line 3, initial pairings are added to the pairing list and on line 4, new flight legs are identified by comparing newly generated pairings with the pairings from previous iterations. From line 6 to 8, a similar procedure is repeated with the exact pairing generation method as long as there were new flight legs identified.

This approach constitutes main part of column generation iterations of the proposed column generation strategy presented in this study. LP runs and connection fixing heuristic which contributes on obtaining integer solutions by eliminating fractions follow this process.

A solution approach that solves crew pairing problem for similar size of data using pool of pairings is also worth to consult in reference (Erdoğan, Haouari, Örmeci, & Özener, 2015). The study uses set-partitioning model and optimization-driven heuristics (combined usage of mathematical methods and metaheuristics) to solve the problem.

4.1. Heuristic solution to the pricing problem

Each column generation iteration starts with a heuristic search which is used to find initial crew pairings of that iteration. It is done by a depth-first search code which tries to find crew pairings with negative reduced costs. For theoretical background of reduced cost calculation in column generation please refer to (Desrosiers & Lübbecke, 2005; Lübbecke, 2011; Lübbecke & Desrosiers, 2005; Vanderbeck, 2005). This heuristic search does not guarantee to find global minimum. Instead, its mission is initializing partial solution generation process by finding new crew pairings of good quality as fast as possible. Usage of an exact algorithm is difficult for this step because of missing knowledge about the flights that new crew pairings will be generated for. Without this knowledge whole pricing network would have to be used and it is practically difficult because of the large size of the data.

Heuristic search procedure is depicted in Fig. 7. Between lines 1-6, depth-search mechanism is triggered for each flight and duty separately. On line 2, flights that new crew pairings will be generated for are searched. On line 3, duties that contain selected flight are searched and filtered. And on line 6, crew pairing search for the selected flight is triggered with the selected duty.

Between line 8 and 18 recursive depth-search code is depicted. On line 9 rule validation is done. On line 10 *currentPairing* is checked to identify whether it is a complete pairing or not. On line 11, reduced cost of *currentPairing* is compared (quality check) with a threshold value. This threshold value is started from zero and during the process, it is dynamically updated according to reduced cost value of *curentPairing*. If the quality of *curentPairing* is sufficient, it is saved on line 12. Between lines 14 and 18 duties which are connectable to *curentPairing* are searched and recursive function is recalled.

New crew pairings in *pairList* will be subjected to a last heuristic operation at the end of depth-search process. This last operation eliminates some of the new crew pairings as to have no over-covered flight leg exists in pairings in *pairList*. For elimination again the reduced cost metric is used. For a flight leg, crew pairing with the lowest reduced cost survives and others are discarded. By doing this a better distributed set of crew pairings is obtained throughout the timetable. And also duties that include flights covered by pairings in *pairList* are omitted in following exact pricing iterations. Therefore size of the exact shortest-path model that will be built using duty-flight overnight connection network in following iterations is reduced and considerable performance increase is gained.

4.2. Exact solution to the pricing problem

After the initialization of the partial solution generation process by the usage of heuristic search described in Section 5.1, exact model for the pricing sub-problem is repeatedly built and run until no new flight leg exists to generate new crew pairings for.

```
1 procedure Generate_Partial_Solution(duals, pairList)
2     pairs = Generate_Pairings_Using_Heuristic_Method(duals);
3     Append pairs to pairList;
4     legs = Identify_New_Legs_For_Pairing_Generation(pairList, pairs);
5     while legs is not empty do
6             pairs = Generate_Pairings_Using_Exact_Method(duals, legs);
7             Append pairs to pairList;
8             legs = Identify_New_Legs_For_Pairing_Generation(pairList, pairs);
```

Fig. 6. Pseudocode of partial solution generation approach.

```
procedure Generate_Pairings_Using_Heuristic_Method(legList, pairList)
    for each leg in legList do
        for each duty that covers leg and departs from the domicile do
            currentPair = Generate a_new pairing;
            Append duty to currentPair;
            Search_for_Pairings(currentPair, pairList)

subprocedure Search_for_Pairings(currentPair, pairList)
    if currentPair is valid
        if currentPair is complete
            if quality of currentPair is sufficient
                Append currentPair to pairList;
        else
            if currentPair can have more duties
                for each duty that departs from the station that currentPair arrives and departs after the arrival time of currentPair do
                    Append duty to currentPair;
                    Search_for_Pairings(currentPair, pairList);
                    Remove duty from currentPair;
```

Fig. 7. Pseudocode for heuristic solution to pricing problem.

Identification of these new flight legs is done by the comparison of the crew pairings from previous solution and the newly generated crew pairings for the partial solution as described in Section 5.

Exact pricing sub-problem is modeled as a shortest path IP problem and can generate more than one crew pairing per run. Before each run of the exact model, new flights are identified by the aforementioned comparison mechanism and a straight-forward duty-flight connection search is done over the proposed duty-flight overnight connection network. This search basically finds duties that cover new flight legs that crew pairings will be generated for and other duties that has legal connections to these pre-found duties. The search is done in forward or backward directions according to the departure or arrival airports of the duties in a time frame that is set by maximum pairing length parameter. The search omits duties that include flight legs that are covered by the previously generated crew pairings in the partial solution.

After each run, newly generated crew pairings are validated. Because of the complete representation of the duty to duty connection rules by the proposed duty-flight overnight connection network, the only possibility is the generation of illegal pairings in the pairing level like exceeding maximum pairing length or having more than one early duties in a pairing etc. If such an illegal pairing is generated, the network model built for that run is changed by removing some connections in order to prevent generating that illegal crew pairing again and model is rerun. A simple heuristic that is depicted below is developed to identify connections that will be removed.

This heuristic (Fig. 8) gets list of duty-flight connections (solution) of the last pricing run as input and sets two output parameters for later use. First one is a boolean value (*violationExists*) that is used to store violation status and a pairing list (*pairs*) that is used to store generated new pairings of the last pricing solution. On lines 3 and 4, it searches for duties that depart from the domicile and this main loop is repeated until no domicile departure duty is found. If a domicile departure duty could not be found, this will mean search is completed and algorithm exists from the heuristic and continues to perform subsequent iterations. If a domicile departure duty is found, on line 8 a new *currentPair* is generated, on line 10 and 11 duty is removed from the input list and added to the *currentPair*. On line 12 *currentPair* is validated after each duty add and if there is a rule violated or maximum pairing length is going to be exceeded, on line 19 and 20 heuristic removes the last duty-flight connection from the pricing network and sets the boolean output *violationExists* as true. If there is no rules violated, depending on the completeness status of the *currentPair*, algorithm adds *currentPair* to *pairs* and goes to line 3 or searches for new duty-flight connections to add more duties to *currentPair* to have a complete pairing.

```
1  procedure Generate_And_Validate_New_Pairings(dutyflightConns, violationExists, pairs)
2      violationExists = false;
3      while true do
4          [duty, leg] = Find the first domicile departed duty and its associated connection flight from dutyflightConns;
5          if no domicile departed duty found
6              Exit from the procedure and continue_to subsequent iterations;
7          else
8              currentPair = Generate a_new pairing;
9          while true do
10             Remove duty from dutyflightConns;
11             Append duty to currentPair;
12             if currentPair is valid
13                 if currentPair is complete
14                     Append currentPair to pairs;
15                     Return to main loop;
16                 else
17                     [duty, leg] = Find the duty that starts with the leg and its associated connection flight from dutyflightConns;
18             else
19                 Remove associated duty-flight overnight connection from the pricing network;
20                 violationExists = true;
```

Fig. 8. Pseudocode for pairing validation and pricing network update heuristic.

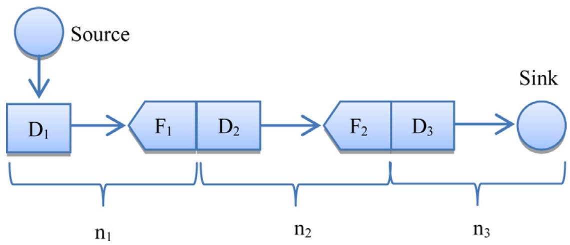

Fig. 9. Sample crew pairing.

However this heuristic does not completely guarantee to prevent generation of illegal pairings after pricing network update. An additional parameter is used to limit number of repetitions of this heuristic and this parameter is set to 3. If there is still an illegal pairing after three iterations, algorithm accepts the *pairs* of the last heuristic and adds to the global pairing pool.

Pairing level rules constitute the minor part of the whole rule implementation of the system and therefore experiences showed that 3 repetitions is enough to solve most of the illegalities faced. And also partial solution generation approach allows leaving some flights uncovered for a few extreme cases.

4.2.1. Exact pairing generation model

In the proposed model each variable represents a duty-flight overnight connection. According to this model, a sample crew pairing with three duties can be depicted as follows:

As can be seen from the Fig. 9 the depicted crew pairing has three duty-flight connections (n_1, n_2, n_3). First two duties has overnight connections to flight legs f_1 and f_2. The last duty d_3 arrives to crew domicile and thus it has a sink connection instead of a flight leg connection.

Assume that N is the set of indices for the variables (duty-flight connections) of the pricing sub-problem and x_n are the binary decision variables. The model for the pricing sub-problem is as follows:

$$x_n = \begin{cases} 1, & if\ duty - flight\ connection\ n\ is\ selected, \\ 0, & otherwise, \end{cases}$$

$$a(d_n, f) = \begin{cases} 1, & if\ duty\ d_n\ includes\ flight\ f, \\ 0, & otherwise, \end{cases}$$

$$b(n, f) = \begin{cases} -1, & if\ flight\ f\ is\ an\ overnight\ connection \\ & flight\ leg\ in\ duty - flight\ connection\ n, \\ 0, & otherwise, \end{cases}$$

$$c(d_n, f) = \begin{cases} 1, & if\ flight\ f\ is\ the\ first\ flight\ of\ duty\ d_n, \\ 0, & otherwise, \end{cases}$$

$$src(d_n) = \begin{cases} -1, & if\ duty\ d_n\ is\ a\ crew\ domicile\ departed, \\ 0, & otherwise, \end{cases}$$

$$sink(d_n) = \begin{cases} 1, & if\ duty\ d_n\ is\ a\ crew\ domicile\ arrival, \\ 0, & otherwise, \end{cases}$$

$$Min \sum_{n \in N} r_n\ x_n \tag{2a}$$

$$\sum_{n \in N_f} a(d_n, f)\ x_n \le 1, \qquad \forall f \in F \tag{2b}$$

$$\sum_{n \in N_f} (b(n, f) + c(d_n, f)) x_n = 0, \qquad \forall f \in F \tag{2c}$$

$$\sum_{n \in N} (src(d_n) + sink(d_n))\ x_n = 0 \tag{2d}$$

$$x_n \in \{0,\ 1\}, \qquad \forall n \in N \tag{2e}$$

In Eq. (2a), r_n is the reduced cost value of duty-flight connection n. Duties with overnight connections also includes the overnight cost in term r_n. In Eq. (2b) d_n represents the duty that belong to duty-flight connection n and N_f represents set of duty-flight connections that covers flight f. This equation guarantees the coverage of each flight maximum one time. Thus over-coverings (deadheads) are not allowed in pricing model. Eq. (2c) is flow conservation constraint that guarantees the continuity of crew pairings through flight overnight connections. Eq. (2d) guarantees that each crew pairing will end properly and allows generating more than one crew pairings per run. By doing this, algorithm does not search for a single crew pairing with minimum reduced cost, instead it generates meaningful combinations of more than one crew pairings that can constitute a partial solution to the whole problem.

As it is described in Section 5, exact pricing sub-problem is solved repeatedly until no new flight is found to generated crew pairing for.

4.3. Connection fixing heuristic

The last step of each iteration is the connection fixing heuristic which is used to eliminate duties and pairings for performance increase in master LP, pricing and final IP runs. In standard column generation implementations, solving the main problem relies on LP even though the output is desired to be integer. If the problem has a symmetric structure, it will cause the branch and bound algorithm to perform poorly that will generate integer solution to the problem at the final step. It is because the problem barely changes after branching in branch and bound tree and a reformulation is mandatory to eliminate this symmetry (Barnhart et al., 1998).

Especially for large instances of crew pairing problems, serious performance problems emerge during the branch and bound run. To cope with this problem a special heuristic called connection fixing heuristic is developed. The code runs after each master LP run and it basically searches for fractional crew pairings that share common flights which has different connections in different crew pairings. Algorithm stores these flight to flight connections with their primal LP variable value that indicates the importance of the flight connection. In each iteration, these connections are ordered according to their importance and connections with high values of importance are fixed. After fixing, crew pairings and duties are removed from the system if they have the same flights with different flight connections. The idea is very close to the implementation in reference (Ryan & Foster, 1981). This idea is used for branching mechanism in branch and price framework in reference (Vance et al., 1997). Instead of branching, our implementation is based on fixing connections and removing duties and crew pairings that violate fixed connections.

A simple formula is used to decide how many duty and crew pairings that will be removed in each iteration.

$$\sqrt{\frac{fracs}{pairings}}\ \frac{\left(\frac{flights}{days}\right)^2\ \ln(itr)^2\ duties}{flights}\ impactCoeff \tag{3}$$

In formula above, *fracs* is the total number of fractional variables (crew pairings) obtained from the last master LP run. *Pairings* is the total number of active (non-removed) crew pairings, *flights* is the total number of flight legs and days is the total number of days in the schedule. *Itr* is the current iteration and the *duties* is the total number of active (non-removed) duties in the system. *ImpactCoeff* is an arbitrary coefficient that can be used to adjust impact of the heuristic and thus the total run time. For *impactCoeff*,

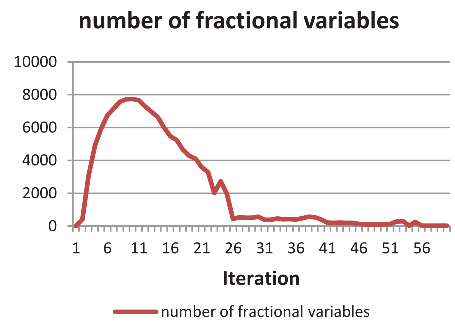

Fig. 10. Change of number of fractional variables during iterations.

0.00035 is used for our test cases. Higher values would terminate the optimization run earlier with higher cost values. Lower values would cause longer optimization runs with higher quality. It is important to set an optimum value that would yield satisfactory solution quality in reasonable time.

Since most of the airline companies have different size of fleets, this formula is designed as to be applicable to all size of fleets without any inside change. For larger fleets, *flights* and *duties* will be greater and therefore formula will eliminate more duties and pairings. It is also adaptive to changes in numbers during the optimization run. Greater values of *fracs/pairings* ratio, *itr* number will eliminate more duties and pairings during the optimization. As can be seen from the Fig. 10, after around 11th iteration, *fracs* decreases significantly and therefore formula will let the heuristic to eliminate less duties in time. Especially after around 26th iteration, the value of *fracs* becomes almost zero and very a few pairings and duties are eliminated in following iterations. It is possible to design and use alternative formulas that might better fit to different size of fleets and timetable of various densities.

After removing certain amount of duties according the formula above, crew pairings that have these duties and invalid connections are also removed from the system.

By applying this heuristic, in time, number of fractional variables in master LP decreases and it approximates to IP. During our tests, some master LP runs did not generated any fractional variable at the last step and therefore even master IP run became meaningless. This behavior of the algorithm allows branch and bound algorithm to perform very fast and finds integer solution very quickly.

In the Fig. 10, change of the number of fractional variables during an optimization run of a timetable with around 17000 flight legs is presented. It is clearly seen that in early iterations the number of fractional variables increases. After some threshold value it decreases to very low levels that branch and bound would easily solve.

Another important note about connection fixing heuristic is that instead of taking connection fixing decisions according to values of only one single run, consideration of two or more runs will show more clearly that the existence of the flight connection would continue and it would be more safe to fix it.

The final schema of the complete solution approach to crew pairing optimization problems is depicted in (Fig. 11). As can be seen from the figure, algorithm takes timetable data *legs* as input. On line 2, it generates all possible duties from input legs; on line 3, duty-flight overnight connection network is built and on line 4, initial pairing generation is done. On line 5, the main loop of the solution approach starts. On line 6 main LP problem is solved and primal and dual information of the last run is stored. On line 7, solution quality of the last run is evaluated and if solution quality is satisfactory, algorithm solves the main IP model, saves the solution and terminates the optimization run. If the solution quality is not satisfied, on line 12, connection fixing heuristic is applied and on line 13, partial solution generation is run to generate new pairings.

Table 2
Test data.

Month	Flight legs	Duties	Initial pairings
Jan	17268	416153	8190
Feb	15738	393285	7458
Mar	17318	390973	8200
Apr	16833	439661	7961
May	16127	406233	7733
Jun	15656	401527	7437

5. Experimental results

The presented test results were generated via the code developed under Java 7 platform. (ILOG, Cplex) library was used as exact solver. Test hardware was an Intel Core i5 1.9 GHz single processor laptop.

The test data used were taken from Turkish Airlines and it covers timetables of first six months of 2014. Turkish Airlines has 4 cockpit fleet and 3 cabin fleets. Among them we have chosen cockpit 320 fleet which is the largest cockpit fleet that requires complex rule implementations. And also it is a typical fleet that has very close properties to the cabin narrow body fleet.

In Table 2 these six datasets are summarized. Each line includes whole flight legs of the planning month plus 10 days from next month. Because each month's schedule will need to cover some flight legs from next month because of the possibility of having crew pairings that will end next month. Because of this, solutions presented in this study includes flight legs less than the datasets used.

In table above, Duties column has total number of duties generated for each month during duty enumeration phase. These duties are also used to build duty-flight connection network for the problems. Initial pairings columns has total number of crew pairings that are generated during initial crew pairing generation phase just before column generation iterations.

There is a complicated objective function structure that is used in current practices in Turkish Airlines. The same objective function model is implemented which has too many terms with various importance. Because of the minor effects of the most of these terms and to be able to focus more on the scope of this study, three of the most important cost coefficients are used in solution comparison tables. To do that three important cost parameters which have the high influence on the total objective value has been chosen. These values obtained from popular parameter sets that are actively in use.

As can be seen from Table 3, these cost parameters are applied to each duty day, each deadhead hour and each hotel overnight. In current practices, even though there is a final objective value, evaluations of the solutions generated are done by using outputs of these parameters as KPI (Key performance indicators) values. Unfortunately true financial cost comparison is not performed because of operational difficulties. Most of the time crew planning department focuses on reducing total deadhead time especially in

Table 3
Parameter sets.

	Duty day	Deadhead hour	Overnight
Set 1	60000	500000	10000
Set 2	200000	100000	10000
Set 3	60000	100000	100000

Table 4
KPI-Cost report for parameter set 1.

KPI	Jan TK	Jan New Impl	Feb TK	Feb New Impl	Mar TK	Mar New Impl	Apr TK	Apr New Impl	May TK	May New Impl	Jun TK	Jun New Impl
# of pairs	5053	5052	4468	4483	4978	4975	4755	4743	4545	4563	4520	4481
# of duties	5886	5884	5199	5203	5862	5860	5702	5704	5636	5627	5474	5449
# of flight legs	13811	13799	12307	12307	13783	13775	13346	13351	12916	12915	12449	12432
# of deadheads	66	64	56	54	84	80	56	54	76	72	72	74
# of duty days	14824	14826	13102	13112	14726	14730	14406	14412	14358	14358	14028	13954
# of nights in domestic overnights	786	808	692	680	718	720	580	628	722	754	668	706
# of nights in international overnights	1338	1336	1170	1170	1518	1524	1792	1790	1936	1892	1670	1660
Total active block time (h)	68518.40	68487.40	60598.40	60599.30	67966.20	67938.40	65634.10	65655.00	64493.00	64483.30	62745.10	62668.00
Total duty time (h)	107528.10	107268.30	95520.30	95227.20	107066.50	106803.20	103385.40	103436.10	101678.20	101368.30	98086.00	97662.20
Total overnight duration (h)	30531.30	30761.10	27030.50	26761.20	33080.40	33509.20	35401.50	36199.30	41844.00	41149.50	38104.00	38444.30
Total deadhead time (h)	162.00	157.20	142.50	137.00	216.50	212.40	157.50	153.40	203.40	190.50	195.10	194.20
Total pairing duration (h)	138059.40	138029.40	122551.20	121988.40	140147.30	140312.40	138787.30	139635.40	143522.20	142518.20	136190.00	136106.50
Active block time per duty day	4.37	4.37	4.37	4.37	4.36	4.36	4.33	4.33	4.29	4.29	4.28	4.29
Total run time (mins)	23.59	25.18	33.35	29.31	29.52	26.23	53.37	30.02	57.12	31.30	35.59	33.13
COST												
Duty day cost	44844000	44850000	39642000	39672000	44514000	44526000	43374000	43392000	43074000	43074000	42108000	41886000
Hotel cost	2094000	2114000	1840000	1829000	2206000	2216000	2352000	2398000	2637000	2626000	2310000	2337000
Hotel transportation cost	7904	7928	6936	6856	8368	8376	8920	8376	9848	9624	8544	8560
Deadhead cost	8099996	7866663	7141663	6849997	10841663	10633329	7891665	7683332	10183331	9541665	9758331	9716664
Soft cost	54124801	54202312	47920076	47952197	53528965	53667513	51059411	51249799	51329634	51411712	50304289	50092492
Total cost	109170701	109040903	96550675	96310050	111098996	111051218	104685996	104731507	107233813	106663001	104489164	104040716

Table 5
KPI-Cost report for parameter set 2.

KPI	Jan TK	Jan New Impl	Feb TK	Feb New Impl	Mar TK	Mar New Impl	Apr TK	Apr New Impl	May TK	May New Impl	Jun TK	Jun New Impl
# of pairs	4456	4392	3932	3909	4500	4437	4337	4352	4312	4305	4199	4195
# of duties	5697	5678	5044	5029	5710	5689	5548	5549	5529	5508	5335	5326
# of flight legs	13874	13871	12393	12377	13803	13812	13404	13401	12955	12945	12463	12458
# of deadheads	170	186	154	144	170	220	158	152	236	202	222	220
# of duty days	14396	14362	12730	12718	14350	14312	14024	14038	14072	14054	13680	13690
# of nights in domestic overnights	1310	1454	1176	1284	1232	1372	1030	998	896	978	984	1064
# of nights in international overnights	1762	1782	1574	1560	1798	1802	1916	1918	1998	1932	1760	1720
Total active block time (h)	68838.00	68792.40	60996.10	60906.30	68061.20	68086.30	65901.40	65900.30	64669.10	64620.20	62820.10	62805.20
Total duty time (h)	108438.40	108136.20	96660.20	96155.10	107682.00	107614.50	103495.40	103356.10	101858.10	101633.10	98019.50	98004.40
Total overnight duration (h)	48369.30	50667.10	43107.50	44191.50	46719.40	48656.40	46500.10	46266.50	46402.40	46331.30	45397.10	45714.20
Total deadhead time (h)	310.20	326.50	299.00	276.50	346.30	405.00	316.10	283.50	444.50	382.50	398.30	401.50
Total pairing duration (h)	156808.10	158803.30	139768.10	140347.00	154401.40	156271.30	149995.50	149623.00	148260.50	147964.40	143417.00	143719.00
Active block time per duty day	4.46	4.47	4.47	4.47	4.44	4.45	4.41	4.41	4.35	4.35	4.35	4.35
Total run time (mins)	24.54	34.13	35.45	37.27	20.55	29.52	27.13	33.05	31.25	33.48	23.48	35.55
COST												
Duty day cost	145200000	144860000	128420000	128300000	144620000	144240000	140760000	140900000	140720000	140540000	136880000	136980000
Hotel cost	3019000	3184000	2696000	2795000	2973000	3117000	2923000	2894000	2872000	2889000	2717000	2757000
Hotel transportation cost	10872	11256	9792	9864	10720	11024	10376	10240	10520	10432	9664	9648
Deadhead cost	3103284	3268280	2989956	2768294	3464950	4049934	3161636	2838302	4448274	3828284	3984948	4018284
Soft cost	160950778	160948695	142381286	142514994	159525007	159583148	153974277	154172687	153511336	153481931	149856798	150153961
Total cost	312283934	312272231	276497034	276388152	310593677	311001106	300829289	300815229	301562130	300749647	293448410	293918893

Table 6
KPI-Cost report for parameter set 3.

KPI	Jan TK	Jan New Impl	Feb TK	Feb New Impl	Mar TK	Mar New Impl	Apr TK	Apr New Impl	May TK	May New Impl	Jun TK	Jun New Impl
# of pairs	5276	5294	4672	4683	5296	5304	4840	4851	4765	4747	4661	4656
# of duties	5926	5932	5240	5248	5942	5939	5707	5724	5657	5633	5485	5463
# of flight legs	13786	13779	12289	12291	13766	13758	13332	13342	12905	12898	12426	12422
# of deadheads	112	104	88	86	136	136	102	90	152	162	148	148
# of duty days	15038	15062	13294	13310	15012	15022	14428	14472	14448	14420	14108	14098
# of nights in domestic overnights	496	494	416	418	432	444	438	460	388	458	446	488
# of nights in international overnights	1290	1280	1138	1138	1350	1334	1752	1760	1844	1794	1608	1574
Total active block time (h)	68426.40	68406.20	60509.00	60509.30	67888.20	67851.30	65576.50	65631.20	64441.40	64408.40	62654.20	62641.40
Total duty time (h)	107239.50	107144.00	95197.20	95001.50	106758.00	106711.10	103060.10	103168.20	101140.00	101188.50	97630.00	97587.40
Total overnight duration (h)	24674.00	24615.00	21735.40	21813.10	24553.10	24440.30	32032.20	32574.10	33769.10	33711.10	33124.10	33101.40
Total deadhead time (h)	218.10	204.40	180.50	174.50	283.50	283.10	243.30	217.50	331.40	340.10	283.10	287.10
Total pairing duration (h)	131913.50	131759.00	116933.00	116815.00	131311.10	131151.40	135092.30	135742.30	134909.10	134900.00	130754.10	130689.20
Active block time per duty day	4.33	4.32	4.33	4.32	4.31	4.31	4.32	4.32	4.27	4.27	4.26	4.26
Total run time (mins)	17.52	34.01	19.28	22.34	29.26	17.32	46.20	25.25	31.59	26.54	34.50	20.20
COST												
Duty day cost	45486000	45558000	40218000	40266000	45372000	45402000	43440000	43572000	43344000	43260000	42348000	42318000
Hotel cost	17810000	17740000	15500000	15530000	17770000	17770000	21720000	22040000	22170000	22390000	20380000	20490000
Hotel transportation cost	6800	6704	5888	5872	6792	6712	8280	8328	8336	8288	7520	7384
Deadhead cost	2181628	2046632	1808304	1748304	2838292	2831630	2434978	2178312	3316626	3401630	2831638	2871638
Soft cost	54598642	54779993	48335351	48437415	53847830	53955006	51019136	51318641	50975177	50967367	50298788	50405386
Total cost	120083070	120131329	105867543	105987591	119834914	119965348	118622394	119117281	119814139	120027285	115865946	116092408

```
procedure Solve_Crew_Pairing_Problem(legs)
     dutyList = Generate_Duties(legs);
     pricingNetwork = Build_Pricing_Network(dutyList);
     pairList = Generate_Initial_Pairings(legs, pricingNetwork);
     while true do
             [primals, duals] = Solve_Master_LP(pairList, legs);
             if solution quality is satisfied
                     ipSolution = Solve_Master_IP(pairList, legs);
                     Save_The_Solution(ipSolution);
                     Exit;
             else
                     Apply_Connection_Fixing_Heuristics(pairList, dutyList, primals);
                     Run_Partial_Solution_Generation_Approach(pairList, legs,
pricingNetwork, duals);
```

Fig. 11. Overview of the algorithm.

Table 7
Summary report with persentage improvement.

		Jan	Feb	Mar	Apr	May	Jun
Param Set 1	Total deadhead time (h)	2.96%	3.86%	1.89%	2.60%	6.34%	0.46%
	# of duty days	-0.01%	-0.08%	-0.03%	-0.04%	0.00%	0.53%
	# of nights in domestic overnights	-2.80%	1.73%	-0.28%	-8.28%	-4.43%	-5.69%
	# of nights in international overnights	0.15%	0.00%	-0.40%	0.11%	2.27%	0.60%
	Total cost	0.12%	0.25%	0.04%	-0.04%	0.53%	0.43%
Param Set 2	Total deadhead time (h)	-5.25%	7.53%	-16.95%	10.31%	13.95%	-0.80%
	# of duty days	0.24%	0.09%	0.26%	-0.10%	0.13%	-0.07%
	# of nights in domestic overnights	-10.99%	-9.18%	-11.36%	3.11%	-9.15%	-8.13%
	# of nights in international overnights	-1.14%	0.89%	-0.22%	-0.10%	3.30%	2.27%
	Total cost	0.00%	0.04%	-0.13%	0.00%	0.27%	-0.16%
Param Set 3	Total deadhead time (h)	6.28%	3.32%	0.14%	10.60%	-2.63%	-1.41%
	# of duty days	-0.16%	-0.12%	-0.07%	-0.30%	0.19%	0.07%
	# of nights in domestic overnights	0.40%	-0.48%	-2.78%	-5.02%	-18.04%	-9.42%
	# of nights in international overnights	0.78%	0.00%	1.19%	-0.46%	2.71%	2.11%
	Total cost	-0.04%	-0.11%	-0.11%	-0.42%	-0.18%	-0.20%

high density seasons and number of duty days especially in seasons when crew utilization is important because of lack of available crew.

The detailed solution comparison tables between current practices (TK) and proposed approach is like below:

It is obvious from the detailed comparison tables above that the proposed strategy is capable of generating very competitive results compared to current practices (TK) that uses state of the art solvers and integer programming algorithms.

Table 7 is a summary table that shows critical schedule assessment parameters for crew pairing solutions. Positive values (with green background) indicate percentage improvement gained with the usage of the proposed approach. As can be seen from the summary table, the proposed approach is more successful on reducing *total deadhead time* which is a really important KPI value especially for busy seasons when occupying passenger seats by the crew will decrease company's income. Especially with the *param set 1* the proposed approach generated superior results for all months in regards to *total deadhead time. Number of duty days* is another important KPI value which is critical especially for periods that company had difficulties to find enough available pilots. This situation is mostly seen at the end of each year in December when some of pilots reached their yearly flying time limits. In terms of *number of duty days*, the proposed approach generated very comparable results. In terms of *number of overnights*, the proposed approach generated very close results except for *number of domestic overnights* which is relatively, operationally easier and less costly than *international overnights*.

The *total cost* values indicate overall optimization performance. From the results it is seen that the proposed approach generated superior *total cost* values except for the *param set 3* which has higher overnighting penalty values. The main reason for this is generating higher number of domestic overnights by the proposed approach. This situation is the only drawback of the proposed approach. Nevertheless, all the parameter sets have very close total cost values to each other. Therefore it is more sensible to compare

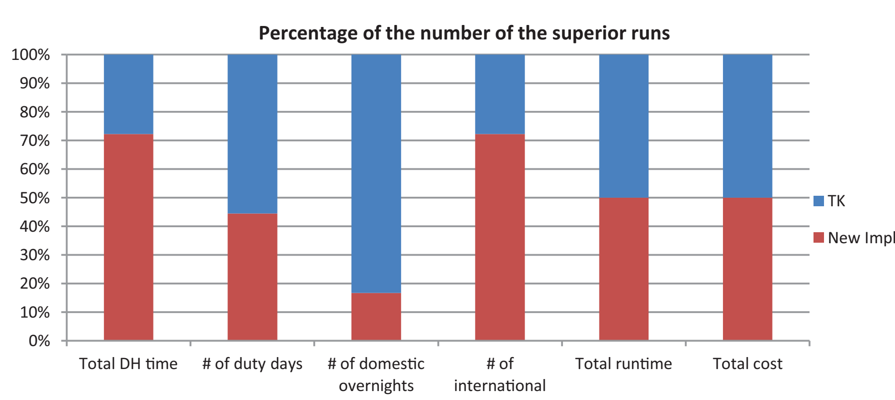

Fig. 12. Percentage of the number of the superior runs.

KPI values of solutions which will show their operational suitability to current circumstances.

All of the run time values are reasonable. Because crew scheduling department can generate solutions for many different scenarios that they want to see and to evaluate in a work day. For half of the optimization runs, the proposed approach generated better runtime values. But it also should be noted that the proposed approach is a single threaded code while the other system has strong parallel processing features and the presented solutions of the current practices (TK) in tables above are generated via 4 parallel threaded runs.

And from the Fig. 12 below it is clearly seen that the proposed approach generated solutions with better deadhead and international overnight values for over 70% of the total runs. In terms of number of duty days, total runtime and total cost values; both algorithms generated better solutions for around half of the total runs. In terms of domestic overnights, current practices (TK) generated better values for around 80% of the total runs.

6. Conclusion and future work

In the presented work, airline crew pairing problem which has been heavily studied in the literature is revisited and a column generation strategy to the problem is proposed. Since crew pairing is the main cost-determining phase of the crew scheduling processes, generating well-arranged crew pairings is highly critical to decrease crew costs and to increase crew utilization. Besides real financial costs, performing crew pairing optimization over some of the other metrics like number of duty days, deadhead time etc. is significantly important to overcome operational difficulties that companies face in real world problems like presented in this study.

During the development of the presented solution procedure; existing approaches were examined, adapted; unique models and strategies were developed. In order to perform healthy comparisons between two systems, completely the same rules and objective function model that the compared system has are implemented in the code of the proposed strategy. As can be seen from the experimental results section, the proposed strategy is capable of generating very competitive results compared to current practices where state of the art solvers and algorithms are employed. Proposed approach is more successful especially on reducing deadhead time and number of international overnights. Its run time performance is very close to the performance values of current practices even though the compared system has parallel processing capabilities and its presented test results were obtained from an advanced server structure with 4 threaded runs.

Besides the successful reports, there are three main contributions of this study:

- Duty-flight overnight connection network
- Partial solution generation approach
- Connection fixing heuristic

The first one is the unique pricing network design called duty-flight overnight connection network. As can be seen from the Sections 4.2 and 5.2.1 the proposed network design provides a memory efficient compact representation to the pricing network need of the presented problems. The presented network model also has the capability of explicit representation of duty to duty connection rules and therefore it provides generation of less illegal crew pairings and requires less modifications on the network model than other approaches. This situation makes it performance efficient as well.

The second contribution is the partial solution generation approach which provides a clear strategy to build pricing network and generate crew pairings and therefore plays the main role in producing high quality results. As it is explained in Section 5, the proposed approach uses crew pairing information from previous iterations and provides generation of better distributed set of crew pairings throughout the timetable. With this regular distribution, the approach also helps to increase the efficiency of the dual variable stabilization scheme of the system.

Connection fixing heuristic is another important contribution that is explained in Section 5.3 steers direction of optimization runs on a tree by fixing flight connections according to their importance and eliminates duties and pairings. It provides significant increase on the performance of the branch and bound algorithm. For some samples, it even completely removes the need of branch and bound run at the end. It also increases the performance of master LP and sub-problems because of the reduction in size of the pricing network during optimization.

Benchmark results obtained show that the proposed approach opens a door for further mathematical and/or heuristical developments and improvements which may lead us generating better, more suitable and robust schedules that will better fit company's needs and operations. And also these initial results were a successful step that encourage us to work on other airline optimization related studies and developments.

As a future work, parallelism in the crew pairing generation part of the code is an important development candidate that will yield better run time values. Improving algorithm's performance especially for generating less domestic overnights would be a good development point as well. This study is also an initial study for the crew rostering phase that is carried out after crew pairing phase. Success of the crew rostering phase is completely depend on the quality of the results of the crew pairing phase.

Acknowledgement

We would like to thank Turkish Airlines Ceo Dr. Temel Kotil and Evp Dr. Ahmet Bolat for their invaluable support to this study. And also we would like to thank Turkish Airlines Crew Planning Department for their help in this research project.

References

Aickelin, U. (2002). An indirect genetic algorithm for set covering problems. *Journal of the Operational Research Society, 53*(10), 1118–1126.
Ben Amor, H., Frangioni, A., & Desrosiers, J. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics, 157*(6), 1167–1184.
Anbil, R., Tanga, R., & Johnson, E. (1992). A global approach to crew-pairing optimization. *IBM Syst. J., 31*(1), 71–78.
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). Norwell, MA: Kluwer Academic.
Baker, E., Bodin, L., Finnegan, W., & Ponder, R. (1979). Efficient heuristic solutions to an airline crew scheduling problem. *AIIE Transactions, 11*(2), 79–85.
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.
Barnhart, C., Cohn, A. M., Johnson, E. L., Klabjan, D., Nemhauser, G. L., & Vance, P. H. (2003). Airline Crew Scheduling. *Handbook of Transportation Science*. New York: Springer US (Chapter 14).
Beasley, J. E., & Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research, 94*(2), 392–404.
Desaulniers, G., & Soumis, F. (2010). Airline Crew Scheduling by Column Generation. CIRRELT Spring School on Combinatorial Optimization in Logistics. Montréal Canada: CIRRELT Spring School.
Desrosiers, J., & Lübbecke, M. E. (2005). A primer in column generation. In Column Generation (pp. 1–32).
Erdoğan, G., Haouari, M., Örmeci, M., & Özener, O. Ö. (2015). Solving a large-scale crew pairing problem. *Journal of the Operational Research Society, 66*(10), 1742–1754.
Garey, M. R., & Johnson, D. S. (1979). Computers and Intractibility: A Guide to the Theory of NP-Completeness. *Books in the Mathematical Sciences*: vol. 44. New York: W. H. Freeman & Company.
Gustafsson, T. (1999). *A heuristic approach to column generation for airline crew scheduling*. Gothenburg, Sweden: (Thesis) Department of Mathematics, Chalmers University of Technology.

ILOG Cplex Optimization Studio, http://www-01.ibm.com/software/info/ilog/.

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.

Kornilakis, H., & Stamatopoulos, P. (2002). Crew Pairing Optimization with Genetic Algorithms. *Lecture Notes In Computer Science, 2308*(2002), 109–120.

Lübbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. *Operations Research, 53*(6), 1007–1023.

Lübbecke, M. E. (2011). Column generation. In *Wiley Encyclopedia of Operations Research and Management Science* (pp. 1–19). John Wiley & Sons, Inc.

Medard, C. P., & Sawhney, N. (2007). Airline crew scheduling from planning to operations. *European Journal of Operational Research, 183*(3), 1013–1027.

Du Merle, O., Villeneuve, D., Desrosiers, J., & Hansen, P. (1999). Stabilized column generation. *Discrete Math, 194*, 229–237.

Ozdemir, H. T., & Mohan, C. K. (2001). Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences, 133*(3-4), 165–173.

Republic of Turkey Ministry of Transport, Maritime Affairs and Communication (2014). Instruction on Flying Team's Task and Resting Terms and Principles of Application, http://web.shgm.gov.tr/documents/sivilhavacilik/files/mevzuat/sektorel/talimatlar/sht_6a_50_rev06.pdf.

Ryan, D. M., & Foster, B. A. (1981). An integer programming approach to scheduling. In *Computer Scheduling of Public Transport* (pp. 269–280). North-Holland: Amsterdam.

Vance, P., Atamtürk, A., Barnhart, C., Gelman, E., Johnson, E., Krishna, A., Mahidhara, D., Nemhauser, G., & Rebello, R. (1997). *A heuristic branch-and-price approach for the airline crew pairing problem* Technical Report LEC-97-06, Atlanta. Georgia Institute of Technology.

Vanderbeck, F. (2005). Implementing mixed integer column generation. In *Column Generation* (pp. 331–358). Springer US.

Wedelin, D. (1995). An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research, 57*(1), 283–301.

Zeren, B., & Ozkol, I. (2012). An improved genetic algorithm for crew pairing optimization. *Journal of Intelligent Learning Systems and Applications, 4*(1), 70–80.