Stakeholder Cooperation for Improved Predictability and Lower Cost Remote Services

Joen Dahlberg, Tatiana Polishchuk, Valentin Polishchuk, Christiane Schmidt
Communications and Transport Systems, Linköping University
firstname.lastname@liu.se

***Abstract*—We consider the problem of adjusting flight schedules (arrival/departure slots) in order to optimize staffing in a Remote Tower Center. We explore tradeoffs between the number of affected flights, the times by which the movement slots are shifted in comparison with the original schedules, the number of airports controlled from one Remote Tower Module, and the number of modules necessary to provide air navigation services to the five Swedish airports with Remote Towers. We consider different variants of the problem (allowing different times for shifts, different numbers of airports that can be assigned to a module etc.), and prove one variant NP-complete, give polynomial algorithms for others, and formulate a general version as an integer program (IP). Our results show that cooperation between airlines, airport owners and ANSPs may help in reduction of Remote Tower Center operation costs by requiring fewer controller positions handling traffic at the airports.**

Keywords*—*Air Traffic Management, Remote Control Tower, Optimal Personnel Scheduling, Integer Programming

I. Introduction

Predictability and cost efficiency are among the most sought-after key performance indicators (KPIs) which both Air Navigation Service Providers (ANSPs) and airspace users (airlines) have strived to improve via a variety of actions. It has been generally accepted that efforts of a *single* actor (e.g., ANSP alone, or a single airline) might not have as large a potential for further improvements as a *cooperative* act of several stakeholder—cases in point are CDM (Collaborative Decision Making), UDPP (User-Driven Prioritization Process), and related initiatives. This paper explores opportunities for collaboration between airlines, airport operators and an ANSP: Is it possible to play with flight schedules in order to smooth operations and save staffing costs in a remote air navigation service center?

Remote Towers Services (RTS) are one of several technological and operational solutions that the SESAR Joint Undertaking delivers to the ATM community for deployment. Today, many small aerodromes struggle with financial difficulties, and a large cost is air traffic control. The RTS concept implementation splits the cost of Air Traffic Services (ATS) provision and staff management between several airports, providing significant cost savings for small airports (30-120 movements a day). The difference in terms of investment is also significant when comparing installation of sensors to the construction of a new tower. Maximizing the efficiency of human resources (HR) is of particular importance because labour accounts for up to 85% of air traffic service (ATS) costs [23].

Motivation

Remotely Operated Towers (ROT) are a novel approach to air navigation service provision via digitization and integration of airport functions. Within the Remote Tower concept it becomes possible to control multiple aerodromes from a single remote location. In particular, a single air traffic controller (ATCO) works with more than one airport from a single position in the Remote Tower Module (RTM). In such a setting, it is of utmost importance to ensure that movements never occur simultaneously in the jointly controlled airports, as one controller may not be able to cope with multiple movements that occur at two airports simultaneously: professional ATCOs, participating in a study on their ability to operate two airports remotely within a single RTM, stated that situations with simultaneous departures and landings are critical for safety and sometimes “impossible to handle” [14]. That is, if two airports simultaneously have movements, they cannot be controlled from the same RTM; in the extreme case, if e.g. simultaneous movements occur in all 5 Swedish airports with remote towers, then 5 RTMs will be needed, undermining the whole idea of operational cost sharing. This motivates investigation of possibilities to perturb flight schedules in order to provide smoother operations of the Remote Tower Center (RTC).

Indeed, airlines create the flight schedules looking only at the individual airports (flights origins and destinations) and their constraints, disregarding the possibility that an airport may be controlled remotely together with another one. It may thus be the case that airlines would be willing to move around their slots: airlines do not adhere to their schedules precisely anyway, so few slot shifts should not be a deal breaker. Moreover, in remote airports slotting is not an issue and hence the parties may hope that the airlines will actually arrive within the allocated slots. Last but not least, the ATCOs may anyway have to put one of the aircraft on hold (even if the aircraft are in different airports) should simultaneous movements occur.

Roadmap

In the remainder of this section we review related work. The next section gives formal definitions related to our optimization problem and settles its computational complexity. In Section III we formulate our problem as an Integer Program. In Section IV we feed the model with real flight data for five Swedish airports planned for remote operation and study how the simultaneous movements in the airports can be handled.

Section V concludes the paper and outlines future work directions.

Related work

RTC aims at providing ATS for multiple airports by air traffic controllers located remotely as defined in [15]. Researches studied various aspects of the RTS concept. Möhlenbrink et al. [12] and Papenfuss et al. [18] considered usability aspects within the novel remote control environment. Wittbrodt et al. [28] stress the role of radio communication in the context of a remote airport traffic control center. In a safety assessment of the Remotely Operated Tower (ROT) concept, Meyer et al. [11] suggest functional hazard analyses and pinpoint the issue of getting reliable probability values for the models. Oehme and Schulz-Rueckert [16] propose a sensor-based solution for aerodrome control that removes the dependency on visibility conditions and tower location. In [5], [14], [13], [10] and [17] various aspects of work organization and human performance issues related to the remote operation are considered. The authors propose several methods to control two airports from a single center. Using simulations they studied how the monitoring performance may influence the system design and behavioral strategies, and suggested several ideas on the design of novel RTC workplaces.

Distributing the total traffic load between controller positions is the subject of sectorization research—a well studied area in ATM; see, e.g., the survey [4] and references therein. While for sectorizations of the Terminal Maneuvering Area (TMA) of an airport the goal is to distribute the monitoring task from a single airport to several air traffic controllers (ATCOs), for an RTC we want to merge the monitoring task of several airports into a single Remote Tower Module (RTM) (then staffed by a single ATCO).

Assigning airport traffic to Remote Tower Modules was initially considered in [1]. That model did not take into account the possibility to switch assignments during the day (that is, an airport was assigned to the same module throughout a complete day) or load balancing between the different modules. Josefsson et al. [9] create an optimization framework with multiple objectives and additional constraints based on the model proposed in [1], and demonstrate how it enables personnel planning at RTCs on real data. In [8] Josefsson et al. take a more detailed look at the problem, and schedule the actual ATCO shifts in the RTC taking into account factors such as maximum time "in position", minimum rest times, etc. Also, for this framework, several objectives are considered, and the authors present how the framework facilitates RTC controller shift planning on real data. In particular, a limitation was placed, preventing the airports from being assigned to the same controller for the whole hour during which the potential conflict occurs. During the conflict hours more controllers are obviously needed because the airports with the potential conflicts are to be controlled by separate controllers. The resulting statistics show a noticeable increase in the lower bound on the total number of controllers for the day with the highest traffic load during the year 2016 (from eight necessary controllers without conflict avoidance to 10 necessary controllers with conflict avoidance). In [8], this constraint is ensured by postprocessing: if airports that are in conflict during a period are scheduled for the same module, local adjustments to the assignment are made to break this. The results show that there is a need to adjust initial airport schedules to optimize their scheduling within RTC, which will provide immediate HR savings. That is, one way to circumnavigate the need for more staff at the RTC could be to ask airlines for slight adjustments to their schedules, by which they would contribute to cost savings via a decreased RTC staff demand.

In general, slot (re)allocation and trading is a recurring topic in ATM [21], [26], [27], [3], [19], [20], [2], [7].

	04:00											
	0	5	10	15	20	25	30	35	40	45	50	55
AP1	0	0	1	1	0	0	0	0	0	1	0	0
AP2	0	0	1	0	1	0	0	1	0	1	1	0
AP3	0	0	0	0	0	1	0	1	0	0	0	0
AP4	0	0	0	1	0	1	0	0	0	0	1	0
AP5	0	0	1	1	0	0	0	1	0	0	1	0

	05:00											
	0	5	10	15	20	25	30	35	40	45	50	55
AP1	0	1	1	1	0	0	0	0	0	1	0	0
AP2	1	1	0	1	0	1	0	0	0	1	0	0
AP3	0	1	1	0	0	0	0	0	0	1	0	1
AP4	0	0	0	0	0	1	0	0	0	0	0	0
AP5	0	1	1	1	0	0	0	0	0	1	0	0

	06:00											
	0	5	10	15	20	25	30	35	40	45	50	55
AP1	0	0	0	1	0	0	1	0	0	0	0	0
AP2	0	0	0	1	1	0	1	0	0	0	0	0
AP3	0	0	0	1	0	1	0	0	0	0	0	0
AP4	1	0	1	1	0	0	0	0	1	1	1	0
AP5	0	0	0	1	0	0	1	0	0	0	0	0

Figure 1. An example of the input matrix F_{as}, where 1's in the matrix cells represent movements.

II. Preliminaries

We start from describing the airports and defining the considered optimization problems; we also comment on the problems computational complexity.

Input: Airports and Conflicts

Over the last years, Swedish ANSP Luftfartsverket (LFV) has been working on the deployment of the RTS concept as an alternative to traditional ATS. In 2015 and 2016 LFV and a Swedish airports operator conducted a joint feasibility study to analyze the impact of the transition from traditional tower ATS to RTS for even more airports in Sweden. The study confirmed that RTS is technically and operationally feasible, the level of risk is manageable, and that it is deemed financially advantageous to use RTS for these airports.

LFV provides remote air navigation services for Örnsköldsvik Airport since April 2015. Two additional airports, Sundsvall-Midlanda and Linköping SAAB, will be connected during 2017. Eventually, it is planned to control five additional airports from a single RTC. The general properties of the airports in consideration may be summarized as follows:

- *Airport 1* (AP1). Small airport with low traffic, few scheduled flights per hour, non-regular helicopter traffic, sometimes special testing activities.
- *Airport 2* (AP2). Low to medium-sized airport, multiple scheduled flights per hour, regular special traffic flights (usually open 24/7, with exceptions).
- *Airport 3* (AP3). Small regional airport with regular scheduled flights (usually open 24/7, with exceptions).
- *Airport 4* (AP4). Small airport with significant seasonal variations.
- *Airport 5* (AP5). Small airport with low scheduled traffic, non-regular helicopter flights.

Our input are aircraft movements at each airport, which we received from the Demand Data Repository (DDR) hosted by EUROCONTROL. We split the time into 5-min intervals, called *slots*, and put every flight into its slot (e.g., if the arrival or departure time is 0853, the movement is put into the slot 0850-0855, etc.). Formally, our input is a matrix F with 5 rows (a row per airport) and a column per each slot; the entry F_{as} in row a and column s is equal to 1 if a movement happens at airport a at time slot s, and is equal to 0 otherwise (see an example input matrix in Figure 1).

At any single airport, at most 1 movement occurs during any slot (therefore the entries in F are only 0's and 1's). However, it often happens that two movements occur during the same slot in different airports; we define this as a *conflict* (in terms of F, a conflict is two 1s in the same column). The main constraint in our airports-to-RTMs assignment problem is that conflicting airports should never be assigned to the same RTM.

Figure 2 shows the number of conflicts in the schedules for all airport pairs for the year 2016, while Figure 3 illustrates the number of days during which these conflicts occur. It can be seen that the number of conflicts is very high, and they occur almost every day for most airport pairs.

Output: Perturbed flights and Airport-to-RTM assignment

Our goal is to introduce "small" shifts to the flight schedules, leading to decreased number of required RTMs. The extent of a shift may be measured by the maximum slot shift (how far a flight is moved) and by the number of shifted flights. We denote the maximum shift by Δ and the number of shifts by S (Δ is measured in minutes and is necessarily a multiple of 5, since we shift only by whole slots).

The last piece of the problem statement is the maximum number of airports per module. In all earlier optimization frameworks for RTC staff scheduling [1], [9], [8] the number of airports that may be consolidated in a single module is a parameter (let MAP denote this bound); for all studies in Sweden MAP is set to 2 (that is, at most 2 airports can be monitored from a single RTM), while in other parts of Europe larger numbers for MAP are considered.

Conflict count	AP1	AP2	AP3	AP4	AP5
AP1		1058	621	366	339
AP2	1058		6473	3400	3021
AP3	621	6473		2603	2517
AP4	366	3400	2603		1449
AP5	339	3021	2517	1449	

Figure 2. The number of potential conflicts in schedules for each airport pair during the year 2016.

Conflict days	AP1	AP2	AP3	AP4	AP5
AP1		341	316	278	285
AP2	341		366	363	365
AP3	316	366		362	362
AP4	278	363	362		359
AP5	285	365	362	359	

Figure 3. The number of days when potential conflicts in schedules occur.

Problem Formulation

We are now ready to formulate the most generic version of our problem:

Flights Rescheduling and Airport-to-Module Assignment (FRAMA)

Given:

- flight slots in a set of airports (the matrix F)
- the maximum allowable shift of a flight, Δ
- the maximum total number of allowable shifts, S
- the maximum number of airports per RTM, MAP
- the total number of modules, M

Find: New slots for the flights and an assignment of airports to RTMs such that

- at most S flights are moved
- each flight is moved by at most Δ
- no conflicting airports are assigned to the same RTM
- at most MAP airports are assigned per module
- at most M modules are used

Note that the above formulation is a *decision* (or *feasibility*) problem: there is nothing to optimize in it, i.e., there is no objective function. As with any feasibility problem, it can be turned into an optimization problem by moving one of the constraints into the objective function. When considering the optimization version of FRAMA, our primary objective will be to minimize the number M of the used RTMs, while respecting the bounds Δ, S and MAP.

Complexity and Heuristics for FRAMA

The next section presents our integer program (IP) for FRAMA; even though solving IPs is NP-hard in general, we demonstrate in Section IV that smaller instances of the problem can be solved using commercial off-the-shelf optimization software. In the remainder of this section we discuss the computational complexity of the problem and possible polynomial-time solution approaches.

Theorem 1. *FRAMA is NP-complete, even if $\Delta = 0$ and MAP=3.*

Proof. The proof is by reduction from `Partition into Triangles (PIT)`, which was shown to be NP-complete for graphs of maximum degree four by van Rooij et al. [24]. An instance of `PIT` is given by a graph $G = (V, E)$ (of maximum degree four), and the question is whether V can be partitioned into triples $V_1, V_2, ...V_{|V|/3}$ such that each V_i forms a triangle in G (that is, such that for each triple of vertices V_i each vertex in V_i is connected to both other vertices in V_i).

Given an instance of `PIT`, that is, a graph $G = (V, E)$ with maximum degree four, we construct the matrix F, the input of FRAMA, as follows: per vertex we have an airport, that is, F has $|V|$ rows. Per non existing edge of G (that is, for each edge in G's complement) we have a time slot. Let $G^c = (V, E^c)$ be the complete graph on the vertex set V, then we have $|E^c \setminus E|$ time slots, one per edge $e^c \in E^c \setminus E$. For the time slot corresponding to $e^c = \{v, w\}$ we add two 1's to the time slot column: to the airports of v and w, all other

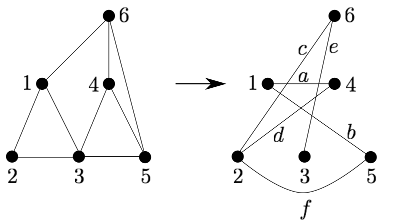

Figure 4. Left: graph G (instance of PIT), right: the complement of G.

TABLE I
THE RESULTING MATRIX F FOR THE PIT INSTANCE GIVEN BY THE GRAPH FROM FIG. 4. GROUPING THE AIRPORTS INTO THE TWO TRIPLES 1,2,3 AND 4,5,6 IS EQUIVALENT TO THE TRIANGLES OF THE SAME VERTICES IN G.

	a	b	c	d	e	f
1	1	1	0	0	0	0
2	0	0	1	1	0	1
3	0	0	0	0	1	0
4	1	0	0	1	0	0
5	0	1	0	0	0	0
6	0	0	1	0	1	1

entries in that column are 0's. See Figure 4 for a graph G and its complement, and Table I for the resulting matrix F.

Any solution to FRAMA with $\Delta = 0$ and MAP=3 then groups the airports, the vertices, into triples, such that there are no conflicts between any of the three airports in a triple, that is, such that there is an edge between any of the three vertices in the triple. Thus, we would obtain a solution to PIT.

Given a solution to FRAMA with $\Delta = 0$ (and, thus, $S = 0$) and MAP$= 3$, it can obviously be verified in polynomial time. □

On the other hand, if $\Delta = 0$ (no rescheduling) and MAP=2, then minimizing the number of modules is equivalent to finding a maximum matching in the "airport conflict graph" that has a vertex for every airport and an edge between two airports if they can be put into the same module (i.e., if they have no conflicts), see Figure 5 for an example. A maximum matching may be found in polynomial time (see e.g. [22]), implying an efficient algorithm for this (restricted) version of FRAMA ($\Delta = S = 0$, MAP=2).

For a general and more interesting case of $\Delta > 0$ (i.e., with rescheduling allowed) and MAP=2, we do not know

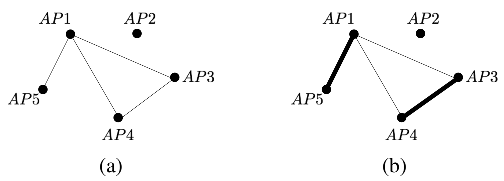

Figure 5. (a) An example for an airport conflict graph: there is no edge between airports 1 and 2, because they are in a conflict. An edge, for example, between AP1 and AP3, indicates that these two airports are not in conflict. (b) A maximum matching in the airport conflict graph: we match AP1 with AP5 and AP3 with AP4, vertices that are not matched, AP2 in this case, constitute a single module, thus, the maximum matching results in 3 modules in this case.

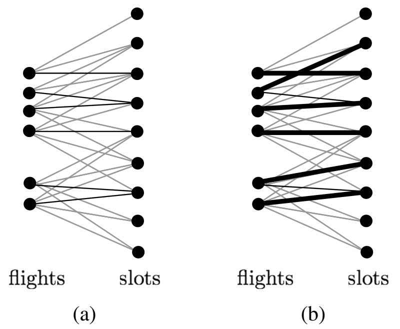

Figure 6. (a) A bipartite graph for the deconfliction problem: vertices representing flights are to the left, vertices representing slots to the right. Black edges indicate the original slot of the flight, gray edges connect a flight to slots within $\Delta = 10$, that is, each flight has two ($\Delta/5 = 2$) edges to earlier slots and two edges to later slots. Gray edges have weight 1, black edges have weight 0. (b) A minimum-weight matching in the graph, shown in bold, of cost 2.

the complexity of FRAMA. One possible approach to the problem is to first remove all the conflicts and then assign the airports to RTMs (i.e., solve the rescheduling and the assignment separately). The assignment problem is trivial in the absence of conflicts (the airports are arbitrarily packed into the RTMs, with MAP airports per module), so we now discuss how to optimally deconflict the flight schedules: The deconfliction problem is reduced to matching by forming a bipartite graph with all flights in one part and all slots in the other part; a flight f is connected to all slots within distance $\Delta/5$ from its original slot (i.e., to all slots to which f may be rescheduled). All edges have weight 1, except for the ones between f and its original slot—this edge has weight 0; see Figure 6 for an example. We now find the minimum-weight matching in the graph that matches all flights (this can be done e.g., with flow techniques, see e.g. [22]). The matching minimizes the total number S of the shifted flights. If no such matching exists, Δ must be increased. (The algorithm may be extended to minimize also the total amount of shifted minutes—just set the weight of each edge equal to the length of the shift.)

For a small number of airports (and the given five airports are feasible for this approach), we can also enumerate all pairs of airports, and completely eliminate all conflicts for the given pairs (matching as described above) with a given $\Delta > 0$, and check whether any combination of these airports leads to the minimum possible number of modules etc.

While the above heuristic of solving the rescheduling and the assignment separately runs in polynomial time, it may find suboptimal solutions to FRAMA (it is not necessary to remove all the conflicts). In the next section we give a unified approach to rescheduling and airports assignment.

III. THE INTEGER PROGRAM

We formulate FRAMA as an Integer Program (IP). We use decision variables x_{am} to indicate whether airport a is assigned to module m, z_m to indicate whether module m is used, and y_{atf} to indicate whether flight f arrives/departs at/from airport a in time slot t. Moreover, the variable w_{ab}

indicates when there is a conflict between airport a and airport b (that is, when $F_{a,t} = F_{b,t} = 1$ for some slot t).

We let A, M, T, and V_a denote the set of airports, modules, time slots, and flights at airport a, respectively. The cost to move flight f at airport a to time slot t is p_{atf}, and the scheduled time for flight f at airport a is s_{af}. We let δ denote the maximum shift distance for scheduled aircraft in terms of time slots, that is, $\delta = \Delta/5$.

$$\min c_1 \sum_{m \in M} z_m + c_2 \sum_{a \in A} \sum_{t \in T} \sum_{f \in V_a} p_{atf} y_{atf} \tag{1}$$

$$s.t. \quad x_{am} \leqslant z_m \quad \forall (a,m) \in A \times M \tag{2}$$

$$\sum_{m \in M} x_{am} = 1 \quad \forall a \in A \tag{3}$$

$$\sum_{f \in V_a} y_{atf} \leqslant 1 \quad \forall (a,t) \in A \times T \tag{4}$$

$$\sum_{t=\max(1, s_{af}-\delta)}^{\min(|T|, s_{af}+\delta)} y_{atf} = 1 \quad \forall (a,f) \in A \times V_a \tag{5}$$

$$\sum_{f \in V_a} y_{atf} + \sum_{f \in V_b} y_{btf} \leqslant 1 + w_{ab} \quad \forall (a,b,t) \in A \times A \times T, a < b \tag{6}$$

$$x_{am} + x_{bm} \leqslant 2 - w_{ab} \quad \forall (a,b,m) \in A \times A \times M, a < b \tag{7}$$

$$\sum_{a \in A} x_{am} \leqslant \text{MAP} \quad \forall m \in M \tag{8}$$

$$x, y, w, z \quad binary \tag{9}$$

The objective function (1) minimizes the number of modules used and the sum of shifts, where c_1 and c_2 are weights assigned to the two components of the objective function, number of modules and the sum of shifts. If the number of shifts is minimized, i.e., $\min S$, then $p_{atf} = 1$ if $t \neq s_{af}$ and $p_{atf} = 0$ if $t = s_{af}$. If the total amount of shifts is minimized, i.e., we minimize the total amount of shifted minutes, then $p_{atf} = |t - s_{af}|$. The constraint set (2) states that if an airport is assigned to a module, then said module is used. The constraint set (3) states that each airport must be assigned to exactly one module. The constraint set (4) states that no more than one aircraft may arrive/depart at/from each airport and time slot. The constraint set (5) states that each aircraft must arrive/depart in a time slot $\pm\delta$ from its scheduled time. The constraint set (6) states that if two aircraft arrive/depart in the same time slot at airport i and airport j respectively, then there is a conflict. The constraint set (7) states that if there is a conflict, then the two airports may not be assigned to the same module. The constraint set (8) states that at most MAP airports can be assigned to each module.

Our IP formulation of FRAMA optimizes a linear combination $c_1 M + c_2 S$ of M and S (alternatively, we could move one of the terms into constraints, giving an upper bound on it, and optimize the other term). We choose c_1 and c_2 such that minimizing the modules is the primary objective. Moreover, the IP computes new slots for flights and assigns airports to RTMs, such that each flight is moved by at most Δ (given by $\delta = \Delta/5$), no conflicting airports are assigned to the same RTM (constraint (7)), and at most MAP airports are assigned per module (constraint (8)). Thus, our IP does solve FRAMA.

IV. Experimental Study

We use traffic data from the five airports, described in Section II, on October 19, 2016. This is the day with highest traffic in 2016 and, thus, can be considered to be the most difficult day for 2016; using one day of traffic uses the underlying assumption that we do not want to move flights by more than a day. Altogether 286 flight movements were scheduled on this day for the five airports. Because of self-induced conflicts, i.e., more than one flight movement in a given slot at a single airport, we use only 233 movements for the first set of experiments.

All optimization problems are solved using the optimization solver Gurobi [6]. Python [25] scripts are used for implementing the optimization problems, as well as handling data and results. There is one optimization problem for each pair (Δ,MAP) of maximum shifts and maximum number of airports per module. The objective function weights, c_1 and c_2, are chosen such that minimizing the number of modules is the primary objective. This is done by choosing $c_1 >> c_2$.

Original traffic

For the experiments we vary MAP (and then look at the results for different δ for a fixed MAP in each case). We start with MAP= 5, see Table II for the results. If no rescheduling is allowed, we need 5 modules. For $\delta = 1$, that is, if we allow rescheduling of at most ± 5 minutes, it is sufficient to use two modules. To reduce the number of modules to one, we need $\delta = 7$, that is, we allow rescheduling of ± 35 minutes. Note that we have $12 \times 24 = 288$ slots for flight movements, that is, with sufficiently large shifts we can accommodate the 233 flight movements in a single module. In Figure 7 we present the number of shifts as a function of the maximum shifts (in minutes). This shows the tradeoffs from allowing more shifts, larger shifts (more minutes) and more APs/module (higher MAP). The results for MAP= 4, MAP= 3, and MAP= 2 are given in Tables III left, III right and IV, respectively.

In the second set of experiments, we actually use all 286 flight movements scheduled for operation on October 19, 2016: in case of a self-induced conflict, the model shifts either of them, that is, we start with possible more than one flight movement per time slot and airport, and obtain a feasible assignment if there is at most one flight movement per time slot and airport. Thus, for $\delta = 0$ this is infeasible by definition. For the results for MAP= 5 see Table V left. For the 233 movements 2 modules were sufficient with $\delta = 1$, for 286 movements, these are sufficient (and the problem at all feasible) only for $\delta = 2$. Similarly, for 233 movements

TABLE II
GIVEN δ, RESULTING NUMBER OF MODULES (M), NUMBER OF SHIFTS (S), AND THE MAXIMUM SHIFT FOR MAP= 5.

δ	# of modules	# of shifts $= S$	maximum shift (in mins) $= \Delta$
0	5	0	-
1	2	32	5
2	2	27	10
3	2	26	15
4	2	26	-
5	2	26	-
6	2	26	-
7	1	118	35
8	1	108	40
9	1	99	45
10	1	91	50
11	1	85	55
12	1	83	60
13	1	81	65
14	1	79	70
15	1	78	75
16	1	75	80
17	1	75	85
18	1	75	90
19	1	74	95
20	1	74	100
21	1	73	105

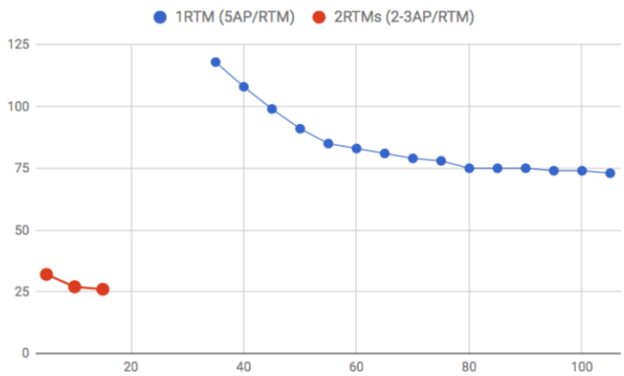

Figure 7. Number of shifts as function of the maximum shift (in minutes). Results for a single module (RTM) are shown in blue, results for 2 modules are shown in red. Note that when allowing 4 airports per ATM (MAP= 4) still 2 modules are needed.

1 module was sufficient with $\delta = 7$, for 286 movements 1 module is sufficient only for $\delta = 37$. The results for MAP= $4, 3$, and 2 are given in Tables V right, VI left and right, respectively. The results for MAP= 5 and MAP= 4 yield the same number of shifts for $\delta = 0, \ldots, 4$: the constraint related to MAP is not binding for $\delta < 37$, i.e. it has no impact.

We also evaluate the computation times for the case of solving the instances in two steps: we solve two optimization with $c_2 = 0$ and $c_1 = 0$ respectively and fix the $\sum_{k \in M} z_k$ to be equal to the optimal number of modules used when solving the second optimization problem. The computation times lie between 1,264 and 146,488 seconds for MAP= 5, see Table VII. The computation times for MAP= 4, MATP= 3, and MAP= 2 are given in Tables VIII, IX, and X, respectively.

TABLE III
GIVEN δ, RESULTING M AND S. LEFT: MAP=4, RIGHT: MAP=3.

δ	M	S
0	5	0
1	2	32
2	2	27
3	2	26

δ	M	S
0	5	0
1	2	32

TABLE IV
GIVEN δ, RESULTING M AND S FOR MAP= 2.

δ	# of modules	# of shifts
0	5	0
1	3	7

TABLE V
GIVEN δ, RESULTING M AND S WITH 286 MOVEMENTS. LEFT: MAP=5, RIGHT: MAP=4.

δ	M	S
0	infeasible	infeasible
1	infeasible	infeasible
2	2	103
3	2	80
4	2	79
36	2	79
37	1	158
38	1	154

δ	M	S
0	infeasible	infeasible
1	infeasible	infeasible
2	2	103
3	2	80
4	2	79
288	2	79

Increased traffic volume

To evaluate the behavior for the case of more traffic, we considered "2x"-traffic for October 19, 2016. That is, each of the original flight movements was duplicated and shifted randomly by plus/minus one hour, and then shifted again, randomly, by plus/minus 15 minutes. If two flight movements end up in the same slot, one of the movements is deleted. Moreover, the "2x" data was created from all data of the year 2016, that is, shifted duplicates of flights from October 18, 2016 and October 20, 2016 may now happen on October 19, 2016. Consequently, we do not end up with exactly twice the number of movements, for October 19, this data set has 416 flight movements (after deleting double movements in time slots) out of 575 flight movements (all of the movements from 2016 that the duplication and shifting process mapped to October 19, 2016). For the results see Table XI and Figure 8. For MAP= 2, we obtain the optimal number of modules of 3 for $\delta = 1$, that is, at most 5 minutes shifts, and only 33 shifts.

TABLE VI
GIVEN δ, RESULTING M AND S WITH 286 MOVEMENTS. LEFT: MAP=3, RIGHT: MAP=2.

δ	M	S
0	infeasible	infeasible
1	infeasible	infeasible
2	2	103
3	2	80
4	2	79
288	2	79

δ	M	S
0	infeasible	infeasible
1	infeasible	infeasible
2	3	61
3	3	61
4	3	60
288	3	60

TABLE VII
GIVEN δ, RESULTING M, S, AND COMPUTATION TIME FOR MAP= 5 WITH 286 MOVEMENTS (SOLVED IN TWO STEPS).

δ	# of modules	# of shifts $= S$	computation time in sec
0	infeasible	-	-
1	infeasible	-	-
2	2	103	1,40
3	2	80	1,26
4	2	79	1,79
36	2	79	7,97
37	1	158	8,42
38	1	154	9,34
39	1	151	40,84
40	1	149	46,61
41	1	147	45,12
42	1	144	38,10
43	1	141	40,20
44	1	139	43,57
45	1	137	9,24
46	1	136	106,31
47	1	135	148,79
48	1	134	100,03
49	1	133	94,08
50	1	132	479,12
51	1	130	433,79
52	1	128	348,83
53	1	126	11,65
288	1	126	46,49

TABLE VIII
GIVEN δ, RESULTING M, S, AND COMPUTATION TIME FOR MAP= 4 WITH 286 MOVEMENTS (SOLVED IN TWO STEPS).

δ	# of modules	# of shifts $= S$	computation time in sec
0	infeasible	-	-
1	infeasible	-	-
2	2	103	1,31
3	2	80	1,06
4	2	79	1,22
288	2	79	60,92

TABLE IX
GIVEN δ, RESULTING M, S, AND COMPUTATION TIME FOR MAP= 3 WITH 286 MOVEMENTS (SOLVED IN TWO STEPS).

δ	# of modules	# of shifts $= S$	computation time in sec
0	infeasible	-	-
1	infeasible	-	-
2	2	103	1,36
3	2	80	1,28
4	2	79	1,09
288	2	79	51,79

TABLE X
GIVEN δ, RESULTING M, S, AND COMPUTATION TIME FOR MAP= 2 WITH 286 MOVEMENTS (SOLVED IN TWO STEPS).

δ	# of modules	# of shifts $= S$	computation time in sec
0	infeasible	-	-
1	infeasible	-	-
2	3	61	0,55
3	3	61	1,09
4	3	60	0,98
288	3	60	100,30

TABLE XI
FOR 2X TRAFFIC.

δ	# of modules	S	Δ	S for 3RTMs (1-3AP/RTM)	S for 3RTMs (1-2AP/RTM)
0	5	0	-	-	-
1	3	30	5	30	33
2	3	24	10	24	25
3	3	23	15	23	24
4	3	23	20	-	23
5	2	111	25	-	-
6	2	101	30	-	-
7	2	96	35	-	-
8	2	92	40	-	-
9	2	88	45	-	-
10	2	87	50	-	-
11	2	84	55	-	-
12	2	81	60	-	-
13	2	81	65	-	-
14	2	81	70	-	-
15	2	81	75	-	-
16	2	80	80	-	-

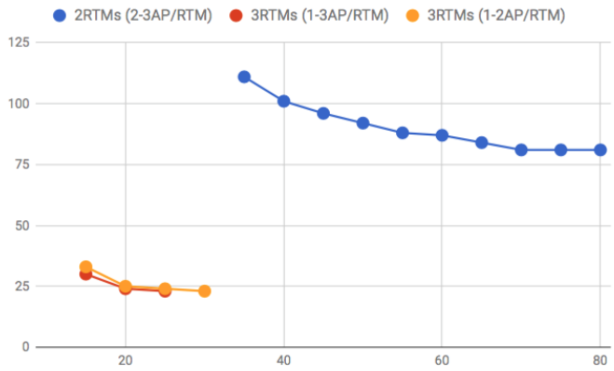

Figure 8. Number of shifts as function of the maximum shift (in minutes). Results for two modules are shown in blue, results for 3 modules with 1-3 airports per module are shown in red, results for 3 modules with 1-2 airports per module are shown in yellow.

In comparison, for the original ("1x") traffic, only 7 shifts were necessary. Figure 8 highlights the aforementioned tradeoffs from allowing more shifts, larger shifts (more minutes) and more APs/module (higher MAP) even more clearly.

V. CONCLUSION AND FUTURE WORK

In this work we considered an optimization problem for remote towers that shift flights to other, nearby, slots in order to minimize the total number of modules in the remote tower center. We discussed the computational complexity of the problem, and suggested several approaches. In particular, we present an integer program and experiments for the five Swedish airports planned for remote operation. These experiments underline the applicability of our approach and show that allowing for shifts of only a few minutes can significantly reduce the number of modules needed for operation.

Our results show that cooperation between airlines, airport owners and ANSPs may help in reduction of Remote Tower Center operation costs by requiring fewer controller positions handling traffic at the airports: already minor shifts of the movement slots may significantly reduce the number of modules necessary for operation. In the original set of flight movements, even without self-induced conflicts, 5 modules are necessary, while already 32 shifts of 5 minutes lead to only 2 necessary modules. We can observe the same trend for the increased traffic ("2x" traffic): without any shifts 5 modules are necessary, 30 flight movements shifted by 5 minutes reduce this number to 3, and 111 flight movements shifted by at most 25 minutes reduce it even to 2.

Our definition of a conflict may be too conservative and too precautionary. The discussions with operational experts on this topic will continue. One question of interest is distinguishing between arrivals and departures; another is taking uncertainty into account. It is clear that the potential conflicts can not be disregarded, and will definitely be reflected in the resulting staff planning solutions. Still, in practice, some controllers may be able to handle conflicts; elaborating other metrics of workload/complexity to quantify benefits of allowing marginal conflicts is the subject of our future research.

From the theoretical perspective, the computational complexity of FRAMA with $\Delta > 0$ and even MAP$=2$ is open.

In our current problem formulation, we do not care which airlines are affected by the reassignment of slots. That is, even if several airlines use the given airports, we might—in the worst case—reschedule flights of only a single airline. Thus, in future studies, we may take *equity* into account, as for example considered by Jacquillat and Vaze [7] for scheduling interventions in case of air traffic congestion without the option to increase airport capacity. For example, if we have three airlines, with airline 1 operating 150 flights, airline 2 operating 75, and airline 3 operating 25 flights, and we need to reassign a slot for 60 flights, we might aim for 36 new slots for airline 1, 18 new slots for airline 2, and 6 new slots for airline 3. Analogously, we could distribute a total amount of shift minutes to the airlines.

VI. Acknowledgment

We thank the anonymous reviewers for the helpful comments. This research is part of the KODIC project supported by the Swedish Transport Administration (Trafikverket) and Swedish ANSP Luftfartsverket (LFV).

The question of flights rescheduling to optimize RTC operations was asked after presentation of [1] at ICRAT 2016.

References

[1] T. Andersson Granberg, P. Axelsson, J. Petersson, T. Polishchuk, V. Polishchuk, and C. Schmidt. Configuration and Planning of the Remote Tower Modules in a Remote Tower Center. In *ICRAT*, June, 2016.

[2] T. Andersson Granberg and V. Polishchuk. Socially optimal allocation of ATM resources via truthful market-based mechanisms. In *2nd SESAR Innovation Days*, 2012.

[3] M. Ball, G. Donohue, and K. Hoffman. Auctions for the safe, efficient, and equitable allocation of airspace system resources. *Combinatorial auctions*, 1, 2006.

[4] P. Flener and J. Pearson. Automatic airspace sectorisation: A survey. *CoRR*, abs/1311.0653, 2013.

[5] M. Friedrich, S. Pichelmann, A. Papenfuss, and J. Jakobi. The evaluation of remote tower visual assistance system in preparation of two design concepts. In *HCI International 2017, Vancouver, Canada.*, 2017.

[6] I. Gurobi Optimization. Gurobi optimizer reference manual, 2015.

[7] A. Jacquillat and V. Vaze. Balancing reliability, efficiency and equity in airport scheduling interventions. In *Twelfth USA/Europe Air Traffic Management R&D Seminar*, June, 2017.

[8] B. Josefsson, T. Polishchuk, V. Polishchuk, and C. Schmidt. Scheduling Air Traffic Controllers at the Remote Tower Center. In *36th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2017.

[9] B. Josefsson, T. Polishchuk, V. Polishchuk, and C. Schmidt. A step towards remote tower center deployment: Optimizing staff schedules. In *Twelfth USA/Europe ATM R&D Seminar*, June, 2017.

[10] P. Manske and S. Schier. Visual Scanning in an Air Traffic Control Tower—A Simulation Study. *Proc. Manufacturing, Elsevier.*, 3:3274–3279, 2015.

[11] L. Meyer, M. Vogel, and H. Fricke. Functional hazard analysis of virtual control towers. In *IFAC HMS 2010*, pages 146–151.

[12] C. Möhlenbrink, M. Friedrich, A. Papenfuss, M. Rudolph, M. Schmidt, F. Morlang, and N. Furstenau. High-fidelity human-in-the-loop simulations as one step towards remote control of regional airports: A preliminary study. In *ICRAT*, 2010.

[13] C. Möhlenbrink and A. Papenfuss. ATC-monitoring when one controller operates two airports research for remote tower centres. *Proceedings of the Human Factors and Ergonomic Society annual meeting*, 55(1):76–80, 2011.

[14] C. Möhlenbrink, A. Papenfuss, and J. Jakobi. The role of workload for work organization in a remote tower control center. *Air Traffic Control Quarterly*, 20(1):5, 2012.

[15] NORACON. OSED for remote provision of ATS to aerodromes, including functional specification. Brussels: SESAR Joint Undertaking, 2013.

[16] A. Oehme and D. Schulz-Rueckert. Distant air traffic control for regional airports. In *IFAC HMS 2010*.

[17] A. Papenfuss and M. Friedrich. Head up only—a design concept to enable multiple remote tower operations. In *DASC*, 2016.

[18] A. Papenfuss, M. Friedrich, C. Möhlenbrink, M. Rudolph, S. Schier, M. Schmidt, and N. Fürstenau. Assessing operational validity of remote tower control in high-fidelity tower simulation. In *IFAC HMS 2010*, pages 117–122.

[19] P. Pellegrini, T. Bolić, L. Castelli, and R. Pesenti. Sosta: An effective model for the simultaneous optimisation of airport slot allocation. *Transportation Research Part E: Logistics and Transportation Review*, 99:34–53, 2017.

[20] A. Ranieri, N. Alsina, L. Castelli, T. Bolic, and R. Herranz. Airport slot allocation: performance of the current system and options for reform. In *3rd SESAR Innovation Days*, 2013.

[21] S. J. Rassenti, V. L. Smith, and R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, pages 402–417, 1982.

[22] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[23] G. A. Tobaruela. *A Framework to assess the ability of automation to deliver capacity targets in European airspace*. PhD thesis, Imperial College, London, March 2015.

[24] J. M. M. van Rooij, M. E. van Kooten Niekerk, and H. L. Bodlaender. Partition into triangles on bounded degree graphs. *Theory of Computing Systems*, 52(4):687–718, May 2013.

[25] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[26] T. Vossen and M. Ball. Optimization and mediated bartering models for ground delay programs. *Naval Research Logistics (NRL)*, 53(1):75–90, 2006.

[27] T. W. Vossen and M. O. Ball. Slot trading opportunities in collaborative ground delay programs. *Transportation Science*, 40(1):29–43, 2006.

[28] N. Wittbrodt, A. Gross, and M. Thüring. Challenges for the communication environment and communication concept for remote airport traffic control centres. In *IFAC HMS 2010*, pages 129–134.