The Report Committee for Zhaowei Liang
Certifies that this is the approved version of the following Report:

A Comparative Analysis of Flight Crew Vacation Allocation Models

APPROVED BY
SUPERVISING COMMITTEE:

Erhan Kutanoglu, Supervisor

Yogesh Dashora

A Comparative Analysis of Flight Crew Vacation Allocation Models

by

Zhaowei Liang

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2019

Abstract

A Comparative Analysis of Flight Crew Vacation Allocation Models

Zhaowei Liang, MSE

The University of Texas at Austin, 2019

Supervisor: Erhan Kutanoglu

The airline industry has a long lasting history of using operations research for complex problems like crew scheduling, crew pairing, and aircraft tail assignment. However, the use of the optimization and operations research on crew vacation planning is not widespread. One of the most popular ways of assigning vacations currently is to let crew members bid for vacations using a heuristic preferential bidding system (PBS). This report will overview the existing problems in the crew vacation allocation domain. Then, it will introduce and compare an optimization based vacation allocation algorithm, an improved heuristic PBS model, and the original heuristic PBS model. Models will be compared using three performance measures as the number of unassigned vacation blocks, the number of crew members without any assigned vacation blocks, and the rank order of the preferences that are awarded to the crew members. This report also conducts a sensitivity analysis for the improved PBS model using the same performance measures.

CONTENTS

List of Tables

List of Figures

1. Introduction

Airlines throughout the world face similar problems with industry operating on low margins attempting to balance the primary expenses of fuel costs, aircraft maintenance, and flight crew salaries. To make matters worse, most large airlines operate within a world of little to no boundaries and a perspective of time that never stops with 24-hour-a-day operations. The multitude of constraints includes a vast number of workers, several airports, changing government rules, strict union regulations, unique crew qualifications, and required aircraft maintenance. All of these constraints provide an insight into the complex planning considerations for the airlines. Out of all the challenges that airlines planners face, workforce planning is one of the most difficult to manage due to the size and structure of the problem. Besides attempting to appease pilots through crew satisfaction, the airlines must meet several statutory rules governed by the Federal Aviation Administration (FAA) or other applicable government agency.

The crew scheduling process is the most challenging aspect of the workforce planning problem and encompasses many challenging and unique sub-problems. The first aspect of the crew scheduling process is the crew pairing problem. This short-term problem focuses on-duty periods, or a succession of flights while accounting for required crew rest, and usually only spans the length of one week. The next aspect is the creation of a crew/flight roster or flight schedule which usually spans the length of one month. One overlooked aspect of the crew scheduling process is vacation allocation as it affects everything else within the model. This is the reason why this report focuses on various models and methods of allocating vacations.

The use of bidding systems integrates into the crew rostering problem and may be extended to the vacation allocation problem. Although there are many bidding systems available, the preferential bidding system (PBS) is a crew rostering approach that is becoming widely adopted by airlines all over the world. Using an online system or an interface, crew members bid on flight pairings, and the model relies on a seniority-based algorithm to award schedules to each. The model objective is to satisfy crew preferences while ensuring feasible assignments to the remaining crews. The constraints in the model ensure schedule feasibility, ensure that all flight times are covered by crew members, and each crew member has only one awarded schedule.

Much like the PBS, airlines use a relatively simple heuristic model that considers seniority, preference, and staffing requirements in a points-based bidding system to enable planners to assign leaves and vacation days. Usually, these systems allow planners to forecast the next 6 to 12 months, and the points-based bidding system incorporates software to fill slots using the pre-programmed model on pilot input. The primary drawback of these systems is that the inherent simplicity of the heuristic model places too much emphasis on seniority. As a result, there is not a heuristic points-based approach that will maximize the preferences of all pilots while considering seniority and staffing. Therefore, an optimization model provides the flexibility to consider multiple approaches to determine what may be best to use based on the preferences of the airline planners.

Details of the optimization model are covered in a separate student project. Hence, the comparative analysis of various types of crew vacation allocation algorithm and scenarios is the contribution to this report. Meanwhile, the comparison between different versions of heuristic PBS models will also be presented. Currently, the PBS model is tightly regulated by federal regulations and company rules. This report will also assume different scenarios by assuming changes in this regulation to analyze the effects of relaxing one or more constraints. The second chapter of this report will review the literature that relates to the current phases of crew vacation planning. The third chapter will briefly go over the structure of all three models mentioned above. Then, the fourth chapter will compare the improved PBS heuristic model with the optimization model by using three performance measures: unassigned vacation slots, pilots without awards, and the average preference awarded to the pilots. The fifth chapter is a sensitivity analysis of PBS under different regulations and policies. The sixth chapter will briefly conclude the findings in this report and introduce future works.

2. Literature Review

Besides being widely used in the airline industry, the development of personnel assignment models applies to various other settings such as healthcare organizations and industrial systems. The nurse scheduling problem is almost as widely studied, debated, and questioned as to the traveling salesman problem in the operations research community. Scholars and analysts have developed multiple ways to solve the nurse scheduling problem even though its complexity may not rival that of other medical specialties. For example, pathology can cover services that span multiple sub-specialties. In some cases, it can be as many as 26 sub-specialties in divisional pathology headquarters. To adequately prioritize and process all pathology requests, clinical managers must develop monthly assignment schedules for doctors. The Ottawa Hospital is one of the first to publish the mixed integer linear program (MILP) used for its Division of Anatomical Pathology (Montazeri, Patrick, Michalowski, and Banerjee, 2015). This model uses hard constraints like the pathologist's sub-specialty, pathologist availability, and service weight (subjective weighting system based on the type of specimen) along with soft constraints like consistent assignments, rotation through specialties, and prioritization of no-fail sub-specialties to optimize an assignment schedule. Besides providing the flexibility to revise model parameters to reflect changes in the system, MILP significantly improves upon the laborious method of manual assignment from the clinical manager. The most basic concepts of Montazeri's paper help to establish the building blocks for the analysis in this paper.

Constraint programming is another application to parse and solve complex problems like scheduling and assignment. The purpose of constraint programming is to identify feasible solutions out of a large set of candidate solutions, where the problem can be modeled in terms of arbitrary constraints. Therefore, the main difference between constraint programming and linear or integer programming lies in the distinction between feasibility versus optimization. Constraint programming may not include an objective function; the goal is to add constraints to the problem until all possible solutions become a more manageable subset of solutions. Kinnunen (2016) uses constraint programming to focus on the specific problem of vacation planning for train drivers in Finland. Similar to airline pilots, train drivers must possess a particular skill set or qualification. Therefore, vacation planning requires more detailed scrutiny in the transit industry as the demand for the service is immediate and there is no shelf-life for it, even though vacation planning is often haphazardly consolidated into overall workforce planning. Kinnunen details 13 constraints that make his problem very unique to the country and the industry, but assist with finding a quicker solution to the model. He compares constraint programming to a MILP solution, and constraint programming outperforms MILP significantly (Kinnunen, 2016).

To make constraint programming feasible in his approach, Kinnunen borrows a concept from another paper that utilizes conditional time-interval variables (Kinnunen, 2016). These variables represent tasks that may be included in the final schedule but are not necessary. The utility of the variables is that they simplify modeling since they intrinsically embed conditionally into the model. In Kinnunen's model, conditional time-intervals include the vacation start time, end time, duration, and a binary variable to indicate if the interval variable is present or not (Kinnunen, 2016).

The use of constraint programming does provide utility in certain situations especially if the model is too large to solve via linear or integer programming. However, even though this method may work well with the problem of assigning vacations to train drivers in Finland, it does not fit well for a points-based bidding system for airline pilots. Besides the degree of problem specificity in Kinnunen's methodology, the main difference that makes it unusable in vacation planning for airlines is the consecutive approach to vacation planning. Ideally, pilots should not take more than three consecutive weeks off as it may require additional resources for refresher training to meet specific requirements. Additionally, it does not consider seniority, nor does it account for any points-based bidding system.

When views through a business analysis perspective, workforce planning is just another opportunity to mitigate business risk through adequate preparation. Therefore, one can formally create a definition of a business risk function for airline crew staffing by detailing the probability of failure with a designated safety margin to capture when demand exceeds supply. Fisher determines business risk by first defining system load (Fisher, 2018). It is a linear equation encompassing the stochastic nature of sick days and no-shows by pilots with the deterministic nature of crew requirements and vacation days. Since it is not feasible to reduce flights or predict stochastic events, the only way to plan to decrease system load is to manage pilot vacations adequately. Fisher (2018) uses a vacation grid optimization technique utilizing aspects of linear programming to determine safety, or reliability, index. From this, he is able to determine how to balance the needs of the business with crew satisfaction base on a designated reliability level.

In summary, Fisher is able to apply the idea of system risk to the airline industry and use linear programming to control the reliability of the crew staffing process. His model output enables the creation of the vacation grid that upholds the interests of both the business and the crew members. However, even though Fisher's model is useful in determining the number of slots allocated per vacation period, it does little to determine who should be awarded vacation. In other words, Fisher's model provides a useful approach of how to allocate vacation, but it does not prescribe who should be awarded a vacation as with our points-based bidding system problem.

Besides being an industry that operates almost nonstop every day of the year, airlines rely on relatively small margins to generate profit. Therefore, mathematical programming is an optimization technique used quite often in a multitude of applications in the airline industry. Holm uses linear programming to solve several problems for Scandinavian airlines (Holm, 2008). His first model attempts to solve the "seat ranking" problem, or how to influence the career ladder of airline pilots by finding a salary distribution that may minimize total cost when accounting for things like training time and utilization. His second model attempts to solve the "crew grouping" problem, or how to find the proper allocation of crew members when they have more than one qualification. Although pilots are promoted to larger aircraft and rarely go back to flying the smaller aircraft, flight attendants usually carry several qualifications. The "crew grouping" problem attempts to answer whether or not more qualifications are better or not in the long run for the airline. His third model is the "reserve crew" problem, and it attempts to solve how to assign adequate reserve crew to cover unforeseen circumstances like weather and maintenance. Finally, his last model covers the "training and vacation" problem, as it attempts to minimize the costs associated with transitioning pilots or pilots not in a queue to flying daily operations. As the model considers transitions to different timeframes, it encompasses all costs such as simulator costs, instructor pilot costs, and shortage requirement costs, while attempting to account for vacations in the model. The solution technique uses the branch and bound method for the MILP, but the model incorporates too little vacation analysis to be of any use for the points-based bidding system problem.

3. Model

3.1 Introduction

As introduced earlier, the goal of the models in this report is solving pilot vacation allocation problem in the airline industry. This section briefly introduces the problem. Pilots bid within their team which is called an instance in this report. Each pilot accumulates points over time. Then, within each instance, the pilots are prioritized or ranked by the number of points that they have accumulated. The highest number of points is associated with the top-ranked pilot in the team. At the beginning of the bidding period, which occurs once or twice a year depending on the airlines, pilots submit several bidsheets to the scheduling manager. Each bidsheet contains a limited amount of bids that include the preferred vacation slots that the respective pilot wishes to take. The goal of the problem is to devise a high-quality schedule to grant pilots their preferred leaves as much as possible. The bidding system is designed to give high priority pilots their preferred bids while obeying specific limitations such as concessions to the federally mandated laws and to the budgeted parameters set by the company. Some terms used in this report can be confusing for readers without an airline industry background. Hence, a short glossary is provided in table 1. Also, a sample bidsheet is provided in figure 1 to show how a bidsheet looks like.

Table 1 Model Glossary

Block	A vacation week
Capacity	The maximum number of pilots that can take a vacation in a block
Preference	A set of 6 or fewer blocks that pilots submit to identify the blocks they would like to bid on
Bidsheet	A collection of 3 preferences
Pass	Each run of the algorithm counts as a pass. The manager can use as many passes as possible.
Awards	A set of blocks assigned to a pilot.
Instance	A group of pilots that will bid and receive vacation allocation results together

	Pilot Num	xxxx	Bidsheet	2nd		
	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Week	5	6	7	8	9	10
Optional?	Y	Y	N	N	N	N
	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Week						
Optional?						
	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Week						
Optional?						

Figure 1 Sample Bidsheet

Certain rules are implemented by the airlines to regulate the vacation assignments. Even though heuristic models and the optimization model utilize different methods to assign leaves, they all need to comply with similar federal regulations and company rules. Below is a list of rules on assigning vacations currently adopted by the major airline companies. The source of this information will remain anonymous due to an established non-disclosure agreement.

1. Pilots will accumulate points according to years of service, and the points will be used for vacation bidding.
2. Pilots will be prioritized by seniority, and the seniority will be determined by points on hand at the time of bidding.
3. Each vacation can last at most three consecutive weeks.
4. The number of pilots taking a vacation in a week cannot exceed the vacation capacity of that week.

Although these regulations seem straightforward, the implementation can be challenging. Hence, both the heuristic model and the optimization model need assumptions. Since it is not possible to ascertain the structure of the bidsheet from the raw data, one primary assumption is that there are only three preferences per bidsheet, and a preference can consist of up to 6 blocks of vacation. A pilot can submit a maximum of 60 preferences before the system blocks him or her.

Another assumption is that the pilots act independently regardless of the instance size and do not attempt to achieve a consensus on vacation allocation before bidding. This assumption becomes essential for those small instances consisting of 7, 11, and 14 pilots. One primary assumption is that one round of bidding followed by forced assignment is sufficient to model and analyze the entire process. It is commonly understood that there are usually 2-3 rounds of bidding before forced assignment. Force assignment is a process to assign blocks to pilots without considering their personal preference. The discussion on force assignment model is beyond the scope of this report. Another assumption is that pilots are bidding for entire weeks/blocks rather than portions of weeks. The formulation of the mixed integer linear program accounts for a variable period, but calculations and optimization solve for entire week periods. One concept omitted from the model is the idea of the couple's bidding or bidding with a partner, as there is insufficient information. Last, all other exogenous factors remain constant, including promotions, retirements, and annual training. These factors will not affect these vacation allocation models.

3.2 Optimization Model (OVA)

This section will introduce the optimization based vacation allocation model (OVA). Below are the notation and integer programming model portrayed.

3.2.1 Single Pass Optimization Model

The single pass optimization model analyzes all of the pilots' bids at once. An essential factor of this model is that pilot and block priorities are incorporated through the use of rank coefficients $R_{ijk}$and r_l. Below is the one pass optimization notation and model:

Notation:

Indices：

I ：The set of pilots

J_i : The set of bidsheets for a given pilot $i \in I$

K_j : The set of bids for a given bidsheet $j \in J_i$

L ：The set of blocks within a bidding period

L_k ：The set of blocks for a given bid $k \in K_j$

N_{ik}:The set of non-optional blocks from bid k by pilot i

Parameters：

A_i ：The total points prior to bidding that pilot $i \in I$ has accumulated

a_l ：The total points it costs to receive block $l \in L$

C ：The maximum number of consecutive blocks to be granted

M_l : The maximum number of pilots who can be awarded leave during block $l \in L$

V_i ：The maximum amount of leave pilot i can have each year

R_{ijk} ：Rank coefficient to incorporate pilot and bid priority

r_l ：Rank vector for block l to incorporate the company's block priority

Decision Variables

x_{ijkl}: Binary decision variable equals 1 if pilot i is awarded block l from sheet j and bid k

y_{ijk} : Binary decision variable equals 1 if bid k is awarded from sheet l for pilot i

w_{il}: Binary decision variable equals 1 if pilot i s awarded block l

u_l : Total number of unassigned slots in a block

Model

$$max\ z = \sum_i \sum_j \sum_k \sum_l R_{ijk} x_{ijkl} - \sum_l r_l u$$

s.t.

(1) $$\sum_j \sum_k \sum_l x_{ijkl} \leq V_i \ \forall\ i$$

(2) $$\sum_j \sum_k \sum_l a_l x_{ijkl} \ \leq \ A_i \ \forall\ i$$

(3) $$\sum_{m=0}^{|L|-C} w_{i(l+m)} \leq C\ \forall\ i, l$$

(4) $$\sum_k y_{ijk} \ \leq \ 1\ \forall\ i, j$$

(5) $$x_{ijkl} = \ y_{ijk}\ \forall i, j, k, l\ \in O_{ik}$$

(6) $$\sum_j \sum_k x_{ijkl} \ \leq 1\ \forall\ i, l$$

(7) $$x_{ijkl} \ \leq y_{ijk}\ \forall\ i, j, k, l$$

(8) $$\sum_l x_{ijkl} \geq y_{ijk}\ \forall\ i, j, k$$

(9) $$\sum_j \sum_k x_{ijkl} \geq w_{il}\ \forall\ \mathrm{i,l}$$

(10) $$x_{ijkl} \leq w_{il}\ \forall\ i, l$$

(11) $$u_l + \sum_i \sum_j \sum_k x_{ijkl} = \ M_l\ \forall\ l$$

(12) $$x_{ijkl}, \mathrm{y_{ijk}}, \mathrm{w_{il}}\ \in \{1,0\}\ \forall\ i, j, k, l$$
(13) $$u_i \geq 0\ \forall\ \mathrm{i, j, k, l}$$

This MIP proceeds to find an optimal solution that maximizes the objective function consisting of two parts: pilot satisfaction and unassigned slots. The pilot satisfaction consists on pilots getting their most preferred bids and the system awards pilots based on their seniority. The first five constraints are to limit the decision variables to adhere to the companies' parameter requests. Constraint (1) ensures that a pilot cannot be granted more vacation than the company allows. Constraint (2) verifies that a pilot will not get blocks that they cannot afford. In other words, a pilot can only get awarded a block if it is in their points budget. Constraint (3) demands that a pilot cannot be on vacation too many blocks in a row.

The following three constraints pertain to the bidding process. Constraint (4) ensures that a bid on a bidsheet is only awarded once. Constraint (5) requires that if a bid is awarded, then all of the non-optional blocks on that bid must also be awarded. Constraint (6) guarantees that a pilot will not be awarded the same block twice.

The last few constraints define the relationships between the decision variables. Constraints (7) and (8) state that if a block is awarded from bid k for a given pilot, then that corresponding bid must also be awarded for the pilot. Then, constraints (9) and (10) links the blocks awarded from a bid with the blocks awarded to a pilot. Finally, constraint (11) defines the unassigned slots variable. Constraint (12) restricts the binary decision variables, while constraint (13) restricts the non-negativity.

3.2.2 Multi-pass Optimization Model

The Single Pass Optimization model assigns every block of vacation awarded to each pilot simultaneously. However, to develop a model that is more closely related to the heuristic, it may be preferred by a company to implement a multi-pass system. This second optimization model grants one bid to a pilot in each pass. Then, after each pass, the points that the pilot used on a given block is deducted from their total points. This results in a shifted ranking of the pilots between each pass. A company may prefer to do this so that the bidding system appears more fair to the employees. The additional notation needed for this model is mentioned below, followed by model formulation.

Added Notation

Sets

Q_{il}: The set of previously awarded blocks granted to pilot $i \in I$

Q_{ijk}: The set of previously awarded bids granted to pilot $i \in I$

Parameters

${A_i}^s$: The total points prior to round s of bidding that pilot $i \in I$ has accumulated

${M_l}^s$: The maximum number of pilots who can be awarded leave during block l in pass s

${V_i}^s$: The maximum number of leaves in pass s that pilot $i \in I$ can be awarded each year

${p^s}_{il}$: Binary parameter equal to 1 if pilot i was awarded block l in a previous pass

${q^s}_{ijk}$: Binary parameter equal to 1 if pilot i was awarded bid k in a previous pass

S : The pass number

Decision Variables

All of the decision variables remain the same as in the single pass model.

Model

$$max\ z = \sum_i \sum_j \sum_k \sum_l R_{ijk} x_{ijkl} - \sum_l r_l u_l$$

s.t.

(1) $$\sum_j \sum_k \sum_l (x_{ijkl} + p_{il}^s) \leq V_i^s \ \forall\, i$$

(2) $$\sum_j \sum_k \sum_l a_l x_{ijkl} \leq A_i^s \ \forall\, i$$

(3) $$\sum_{m=0}^{|L|-C} w_{i(l+m)} \leq C \ \forall\, i, l$$

(4) $$\sum_k (y_{ijk} + q_{ijk}^s) \leq 1 \ \forall\, i, j$$

(5) $$x_{ijkl} = y_{ijk} \ \forall i, j, k, l \in O_{ik}$$

$$\sum_j \sum_k \mathrm{x_{ijkl}} + p_{il}^s \leq 1 \;\forall\, i, l \tag{6}$$

$$x_{ijkl} \leq y_{ijk} \;\forall\, i, j, k, l \tag{7}$$

$$\sum_l x_{ijkl} \geq y_{ijk} \;\forall\, i, j, k \tag{8}$$

$$\sum_j \sum_k x_{ijkl} \geq w_{il} \;\forall\, i, l \tag{9}$$

$$x_{ijkl} \leq w_{il} \;\forall\, i, l \tag{10}$$

$$u_l + \sum_i \sum_j \sum_k x_{ijkl} = M_l^s \;\forall\, l \tag{11}$$

$$x_{ijkl} = 0 \;\forall\, i, j, k < S^s, l \tag{12}$$

$$q_{ijk}^{\mathrm{s}} > y_{ijk} \;\forall\, i, j, k, q_{ijk}^{\mathrm{s}} \in Q_{ijk} \tag{13}$$

$$p_{il}^{\mathrm{s}} > x_{ijkl} \;\forall\, i, j, k, l, p_{il}^{\mathrm{s}} \in Q_{il} \tag{14}$$

$$\sum_j \sum_k y_{ijk} \leq 1 \;\forall\, i \tag{15}$$

$$x_{ijkl}, \mathrm{y_{ijk}}, \mathrm{w_{il}} \in \{1,0\} \;\forall\, i, j, k, l \tag{16}$$

$$u_i \geq 0 \;\forall\, \mathrm{i, j, k, l} \tag{17}$$

This objective of the Multi-Pass Model is the same as the single pass Model. However, a few constraints change because of the implementation of the “pass system”. First, three of the constraints are modified to include the previously awarded vacations to pilots. Constraints (1) and (6) must now take into account the previously awarded blocks, and constraint (4) needs to include the previously awarded bids. Secondly, constraints (12-15) are added to the model. When implementing passes, the most preferred bidsheet shifts between each pass. For example, in pass 1, the highest ranked bidsheet is sheet 1. However, in pass 2, the highest ranked bidsheet is Sheet 2, and all bids from Sheet 1 can no longer be awarded. Constraint (12) forces the bids on the sheets that can no longer be awarded. Next, constraints (13) and (14) declare that previously awarded bids and blocks cannot be re-awarded in a later pass. Finally, constraint (15) ensures only one bid is awarded per pass. All of the other constraints from the single pass model remains same except for a few parameters that are updated between each pass. These parameters and their updates are written in the post processing below:

$$(1)\ A_i{}^s = A_i{}^{s-1} - \sum_j \sum_k \sum_l a_i x_{ijkl} \quad \forall\, i, s > 0$$
$$(2)\ V_i{}^s = V_i{}^{s-1} - \sum_j \sum_k \sum_l x_{ijkl} \quad \forall\, i, s > 0$$
$$(3)\ M_l{}^s = M_l{}^{s-1} - \sum_i \sum_j \sum_k x_{ijkl} \quad \forall\, l, s > 0$$
$$(4)\ p_{il}{}^s = p_{il}{}^{s-1} + \sum_j \sum_k x_{ijkl} \quad \forall\, i, k, s > 0$$
$$(5)\ q_{ijk}{}^s = q_{i\,jk}{}^{s-1} + \sum_j \sum_k y_{ijk} \quad \forall\, i, s > 0$$

The first pass $S^0 = 0$. Then, the initial parameters $A_l{}^0, V_i{}^0, M_l{}^0$ are all determined by the company's scheduling manager. Also, $p_{il}{}^0$and $q_{ijk}{}^0$are all equal to zero to begin because before the first pass, pilots have not received any vacation time yet.

3.3 Heuristic Method (PBS)

This subsection will cover two types of PBS models. The first one is the original PBS model, which is currently used by the airline industry for vacation allocation. The second one is an improved version of the original PBS model. These two models use a heuristic method to allocate vacations and have a similar approach to go through bidsheets and preferences. However, they are different in evaluating individual preferences. Both the original PBS and the improved PBS can be run for multiple passes by feeding the output from the previous pass as the input to the next pass. The concept of the multi-pass heuristic model will be introduced in section 3.3.3.

3.3.1 Original Heuristic PBS Model (OPBS)

This section will explain the original heuristic model first. The flow chart of the original heuristic algorithm can be found in figure 2. The original heuristic model utilizes a simple greedy algorithm. A bidding process starts from the pilot with most points. Starting from the first block on the first preference in each bidsheet by this pilot, the algorithm evaluates this block using constraints mentioned in section 3.2. If the block passes all constraints, the algorithm will move to the next available block in that preference. Meanwhile, If the block fails to pass any of the constraints and it is an optional block, the algorithm will continue to move to the next block and neglect the optional block. However, if a block fails any of the constraints and it is not an optional block, the entire preference will be voided.

If a preference fails the test of this algorithm, the next available preference on the same bidsheet will be evaluated by the algorithm. If all preferences on a bidsheet fail to yield any feasible solutions, the algorithm will move to the next available bidsheet. This process will happen repetitively until the algorithm finds a feasible solution. The algorithm will move to the next pilots if either all the bidsheet from a pilot has been evaluated or the algorithm successfully assigned blocks to a pilot. If the OPBS runs on single pass, the algorithm will be terminated as it traversed all pilots. If the OPBS runs on multi-pass, it will go through the multi-pass algorithm as introduced in section 3.3.3. The IPBS which is introduced in the next section uses the same process to navigate through bidsheets and pilots.

The original PBS uses a straightforward algorithm; however, due to its greedy nature, pilots may not get the longest or the highest preferable solution from their preferences. For example, if a pilot can be awarded a better vacation plan by skipping the first week on his or her preference, the original PBS algorithm will not be able to choose that allocation. This drawback brought on the improved heuristic model that is introduced in the next section.

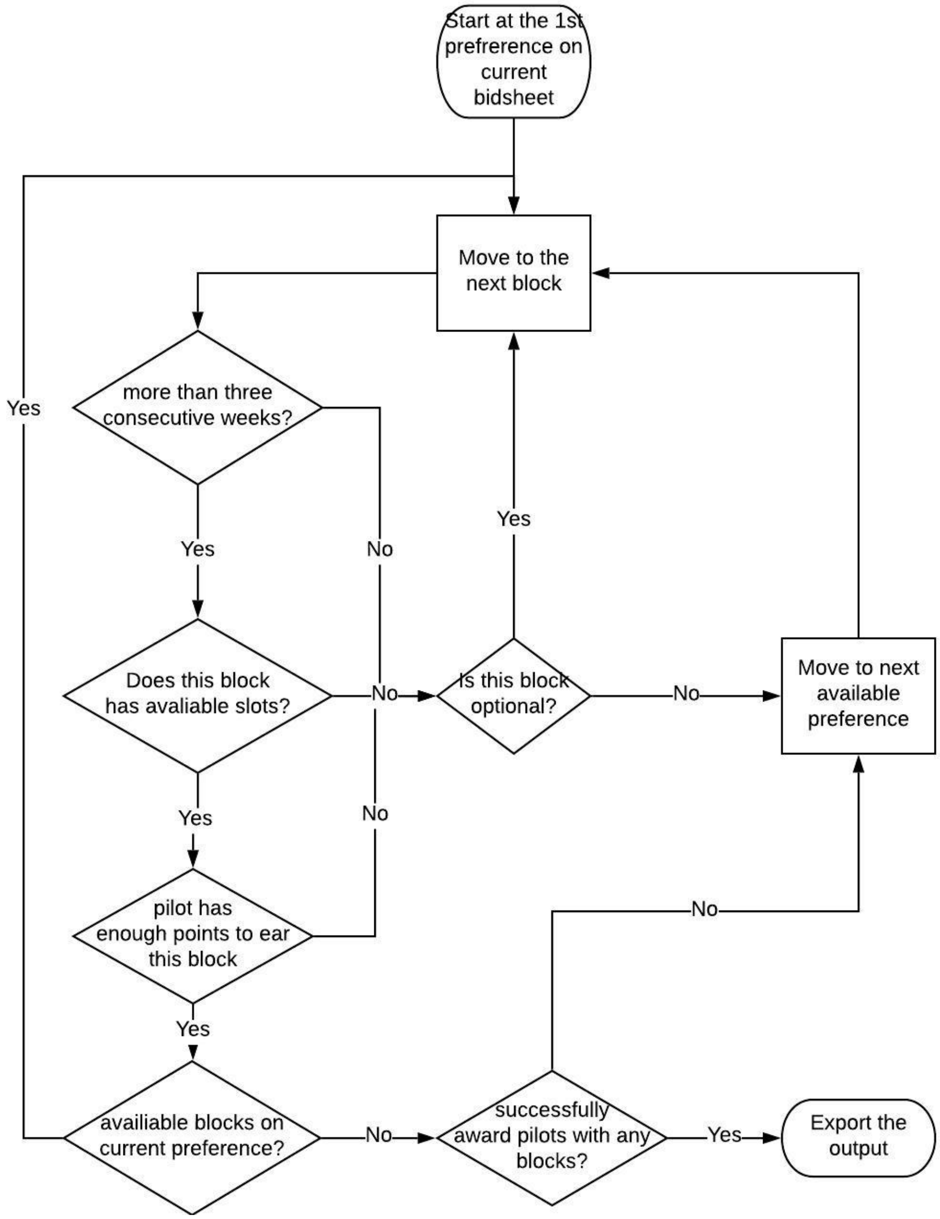

Figure 2 Flow Chart for the Original PBS Model

3.3.2 Improved Heuristic PBS Model (IPBS)

The IPBS algorithm utilizes a brute force search algorithm to find all feasible solutions heuristically. On the preference level, this algorithm will attempt to traverse all available solutions within a preference first and then move down to solutions with fewer blocks. Each solution is guaranteed to contain all non-optional blocks. The pseudocode for this algorithm is displayed in figure 3. The algorithm flowchart is shown in figure 4.

```
for each Pilot in Ballots:
    for each Bidsheet of Pilot:
        for each Preference of each Bidsheet:
            for i in range(1, length of Preference(L), -1):
                CombinationList = all possible C(L, i)
                for Combination in CombinationList:
                    if all Non-Optional blocks is in Combination:
                        PossibleSolution = Current Combination + Awarded Combination
                        if LongestConsecutiveWeeks in PossibleSolution < 3:
                        if all weeks in combination has open slots:
                        if points of Pilot > total cost of the bid:
                            SolutionList += Combination
                if SolutionList is not empty:
                    delete current Bidsheet
                    Break
    if SolutionList is longer than 1:
        sort Solutions in SolutionList by TotalPoints of the Solution
        AwardList[Pilot] += Solution[1]
    else:
        AwardList[Pilot] += Solution
```

Figure 3 Pseudocode for the Improved IPBS Model

This IPBS model is superior to the OPBS model for two reasons. This model does not utilize backtracking techniques like OPBS where the blocks are selected incrementally, starting from the first feasible block in a preference. By listing out all possible combinations, this IPBS model will pick the most extended vacation possible for the pilot. If multiple feasible vacation combinations tied at the same vacation length, this algorithm will break the tie by picking the preferable vacation combination based on choosing the vacation combination with the highest value in points. There are two reasons that IPBS uses the total points value of a vacation to break the tie. Firstly, a set of vacations with a higher points value contains more popular weeks like holiday weeks. Secondly, since pilots with higher seniority rank will be evaluated first, deducting more points from their hands could balance the point distribution.

The IPBS model is proved to have better overall performance using the performance measures that will be introduced in chapter 4. The comparison results between the IPBS and OPBS will also be presented in section 4.1.

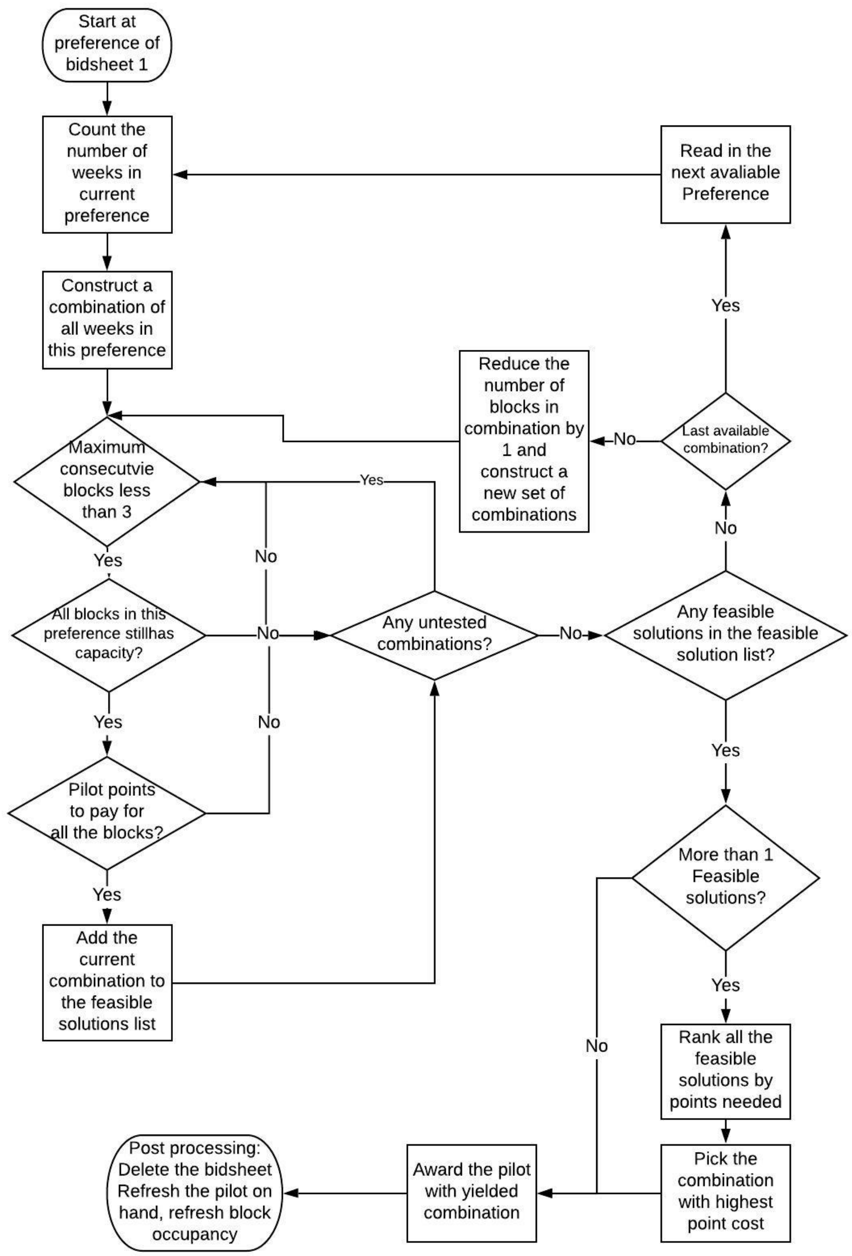

Figure 4 Flow Chart for Improved PBS Model

3.3.3 Multi-pass

Airline companies usually run PBS algorithms recursively multiple times to get the best result. Figure 2 shows how the multi-pass algorithm works. The output from the previous pass will be used to update the crew points on hand and the vacation availability for different weeks. Then this updated input will be put back into the model to run for the second time or the third time.

At the end of a bidding pass, the algorithm will first calculate the number of points each pilot spent in the current pass and delete the respective points from these pilots. Then, another function will calculate the remaining capacity, which is the original capacity of the block minus the number of pilots that awarded with that block. Then, with the new points and capacity data, the multi-pass algorithm will call the single pass algorithm again. Figure 2 summarizes the concept of the multi-pass heuristic PBS model.

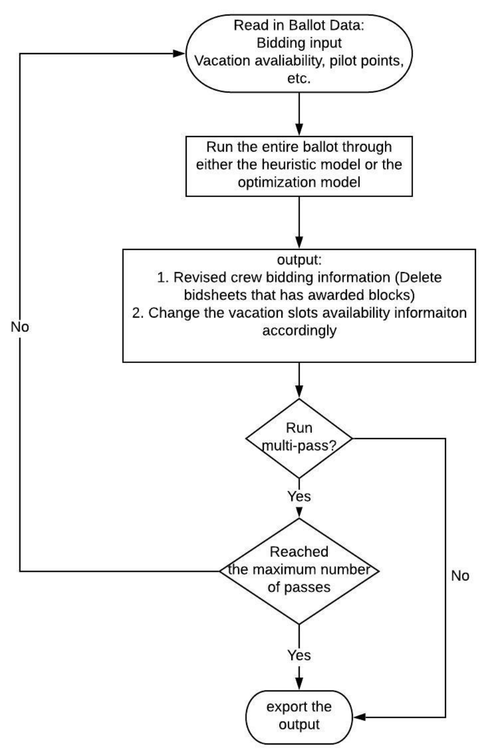

Figure 5 Flow chart to show the difference between signal and multi-pass

4. Comparison Between OVA and IPBS

4.1 Instances and Data

This report relies upon one year of historical bidding data from one particular airline to create the model and measure the performance. The original data comes in the form of multiple Excel files and needs requisite data cleaning to be useful for numeric analysis. Each instance in this dataset comes to form a specific pilot group and does not interact with other instances. Each instance will also bid for vacation weeks separately and will be analyzed individually. Four separate Excel files contain the bidding history for every instance, the leave slots available for each instance, the number of points for each pilot, and the designated points required per block for each instance.

The data contains 21 complete instances and amounted to 24,574 unique pilot-blocks combinations. In other words, each preference has an average of 5.3445 weeks chosen out of the six available in the system. The average length of preference is calculated by dividing the sum of the length of all preferences over the number of preferences. Once the data had been consolidated and cleaned, it is parsed out as instances for use in the model.

The summary statistics from the consolidated data provided some vital information and useful insights into the overall distribution of bids and pilot behavior. First, the current allocation of pilot points followed very closely to a normal distribution with a minimum of 300, a maximum of 4988, a mean of 2591, and a median of 2629 points. The distribution of points spent per pilot followed more of a gamma distribution with a mean of 469 and a standard deviation of 335. The statistic detail is generated from raw crew bid information dataset. Next, the number of preferences submitted by each pilot corresponds to his or her points. Pilots with many points submit few preferences knowing that they will get their top choices. Pilots with very little points also submit a few preferences knowing that they are at the bottom of the priority list. The pilots in the interquartile range of points submit the most bids. On average, almost 75% of all bids are non-optional, and this did not change much from block 1 to block 6 for each preference, rather pilots chose to forgo blocks 5 and 6 instead of labeling the blocks as optional. Last, the bid distribution for the popular and unpopular weeks generally followed a normal distribution, with the only caveat being that pilots with more points generally listed popular weeks (e.g. Christmas) in their first three preferences. Pilots with fewer points would list favorite weeks farther down in their preference lists in the hopes that they would be awarded.

In this report, the number of pilots in each instance is called the instance size. The instance size ranged from 7 to 97 from the dataset, and the difference between instance sizes ranged from 0 to 33. The horizontal axis in the comparison graph will be the instance number which is also the rank of the instance size. The comparison graph will reveal the general trend of performance measures instead of showing the exact relationship between instance sizes and performance measures.

4.2 PERFORMANCE MEASURES

This comparison analysis section utilizes three performance measures: the number of unassigned slots (UAS), the number of pilots without bids or unassigned pilots (UAP), and the average preference awarded (APA). These performance measures will also be used later in section 5 to compare the IPBS results under different scenarios. Chapter 4 will focus on comparing the IPBS model with the optimization model. The difference between the results from different models will be marked as Δ. Unless otherwise notified, the difference between performance measures mentioned in this chapter will be calculated by subtracting the results of the IPBS model from the OVA model (E.g. $\Delta UAS = UAS_{OVA} - UAS_{IPBS}$). For all three performance measures, smaller values indicate better performance.

The UAS value is a sum all remaining capacity from each week at the end of all bidding passes. High UAS value implies a lower utilization of the vacation capacity of the blocks. As a result, low UAS means the algorithm is effectively using all available vacation slots. As mentioned in previous sections, the number of pilots can take a vacation in the same week of the year is limited by the given capacity of that week. Although favorite weeks (e.g., Christmas week) will be filled out quickly, other unpopular weeks can remain low after the bidding process. The UAP value is the number of pilots who have no vacation assigned at the end of the bidding process. These pilots remain unassigned due to various reasons. It can either be due to low seniority compared to other team members in the instance or due to the pilot's aggressive bidding strategy (e.g., set all blocks as non-optional). By measuring how many pilots are unassigned at the end, it is possible to measure how well the model is taking care of seniority and pilot preferences. Lastly, the APA value is the average of preferences awarded. Intuitively, a higher APA means the model is trying to award low preference bids to pilots. This measurement will show the overall pilot satisfaction inside an instance.

A table of the acronyms is presented in table 2.

Table 2 List of Acronyms

PBS	Preferential Bidding System
IPBS	Improved Preferential Bidding System
OPBS	Original Preferential Bidding System
UAS	Unassigned Slots
UAP	Unassigned Pilots
APA	Average Preference Awarded
OVA	Optimization based Vacation Allocation model

4.3 IPBS vs. OPBS

As mentioned in previous sections, the IPBS is proven to have better overall performance than the OPBS. OPBS seems to outperform the IPBS in some instances in UAS and APA. However, the differences are not significant. Among all 21 instances, the IPBS model has better performance in 16 instances on APA or UAS and 18 instances on UAP. In order to keep the consistency of the comparison, this report will use the IPBS model to conduct the comparison between the heuristic model and the OVA model. The detailed comparison results are listed in table 3. the difference (Δ) values are calculated by subtracting the performance value of OPBS from IPBS which means that a positive number indicates an improvement.

Table 3 IPBS vs. OPBS Comprehensive Comparison Table

Instance	Instance Size	ΔUAS	Δ UAP	Δ APA
1	7	0	0	0.2
2	11	-3	-1	-0.07143
3	14	1	0	0
4	17	-1	0	0
5	17	4	1	-0.02727
6	17	3	1	-0.03846
7	17	4	0	0.153846
8	18	1	0	0
9	21	0	0	0
10	23	0	0	0.071429
11	25	2	0	0
12	27	-6	-1	-0.26667
13	29	4	1	-0.09486
14	32	11	1	0.786946
15	38	-4	0	0
16	71	16	3	-0.24702
17	75	7	2	0.393548
18	76	-3	-1	0.235027
19	78	11	2	0.058548
20	83	11	1	0.199437
21	97	15	1	-0.03499

4.4 Unassigned Slots (UAS)

As shown in table 4, although instances 1 and 4 have less UAS in the IPBS model, it is evident that the OVA model has lower UAS in the rest of the instances. The OVA can reduce the number of UAS compared to the IPBS model. Figure 5 shows that as the instance size increases, the difference in UAS between the IPBS model and the OVA model also increases. The most significant difference happens at the largest instance which is instance 21. The difference is 65 in slots and 20.78% in percentage.

Table 4 IPBS vs. OVA UAS Comparison Table

Instance	Instance Size	IPBS UAS	OVA UAS	Δ UAS	Δ UAS (%)
1	7	38	39	1	2.63%
2	11	40	33	-7	-17.50%
3	14	53	52	-1	-1.89%
4	17	70	71	1	1.43%
5	17	71	70	-1	-1.41%
6	17	65	61	-4	-6.15%
7	17	57	55	-2	-3.51%
8	18	66	62	-4	-6.06%
9	21	76	73	-3	-3.95%
10	23	84	75	-9	-10.71%
11	25	93	92	-1	-1.08%
12	27	118	107	-11	-9.32%
13	29	84	72	-12	-14.29%
14	32	123	122	-1	-0.81%
15	38	127	102	-25	-19.69%
16	71	227	201	-26	-11.45%
17	75	216	202	-14	-6.48%
18	76	218	185	-33	-15.14%
19	78	255	231	-24	-9.41%
20	83	285	254	-31	-10.88%
21	97	313	248	-65	-20.77%

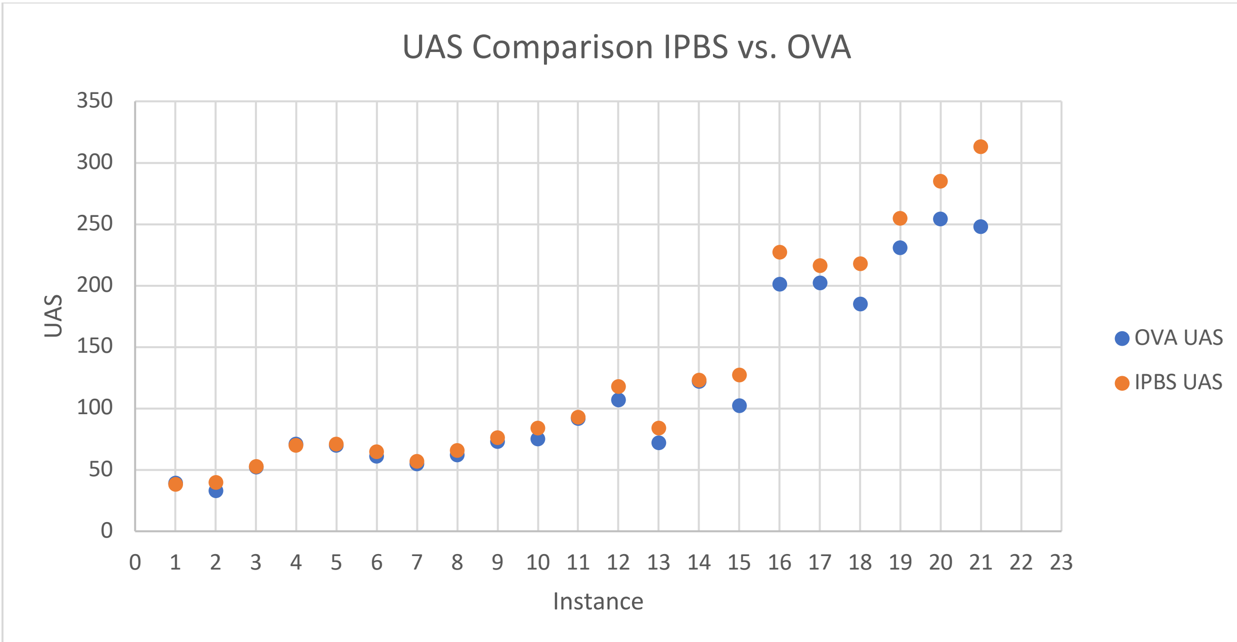

Figure 6 UAS Comparison Line Chart IPBS vs. OVA

4.5 Unassigned Pilots (UAP)

Similar to the UAS comparison, the comparison of UAP over IPBS and OVA yields a similar result. The OVA model can reduce the number of unassigned pilots. A steady reduction in UAP starts to happen with an instance sizes of over 30. Instance one which has only seven pilots has more UAP in OVA than it is in IPBS. The difference between the IPBS model and the OVA model are even wider in UAP. The most substantial difference can go up to 50%. 5 out of 21 instances show the same result between OVA and IPBS. This means the difference between IPBS and OVA are more evident in UAP when the difference exists.

Table 5 UAP Comparison Table IPBS vs. OVA

Instance	Instance Size	IPBS UAP	OVA UAP	Δ UAP	Δ UAP (%)
1	7	2	3	1	50.00%
2	11	5	3	-2	-40.00%
3	14	7	6	-1	-14.29%
4	17	4	4	0	0.00%
5	17	6	6	0	0.00%
6	17	4	4	0	0.00%
7	17	4	2	-2	-50.00%
8	18	7	6	-1	-14.29%
9	21	5	5	0	0.00%
10	23	9	6	-3	-33.33%
11	25	9	8	-1	-11.11%
12	27	12	9	-3	-25.00%
13	29	6	4	-2	-33.33%
14	32	3	3	0	0.00%
15	38	13	8	-5	-38.46%
16	71	13	10	-3	-23.08%
17	75	13	11	-2	-15.38%
18	76	19	13	-6	-31.58%
19	78	15	10	-5	-33.33%
20	83	20	14	-6	-30.00%
21	97	27	19	-8	-29.63%

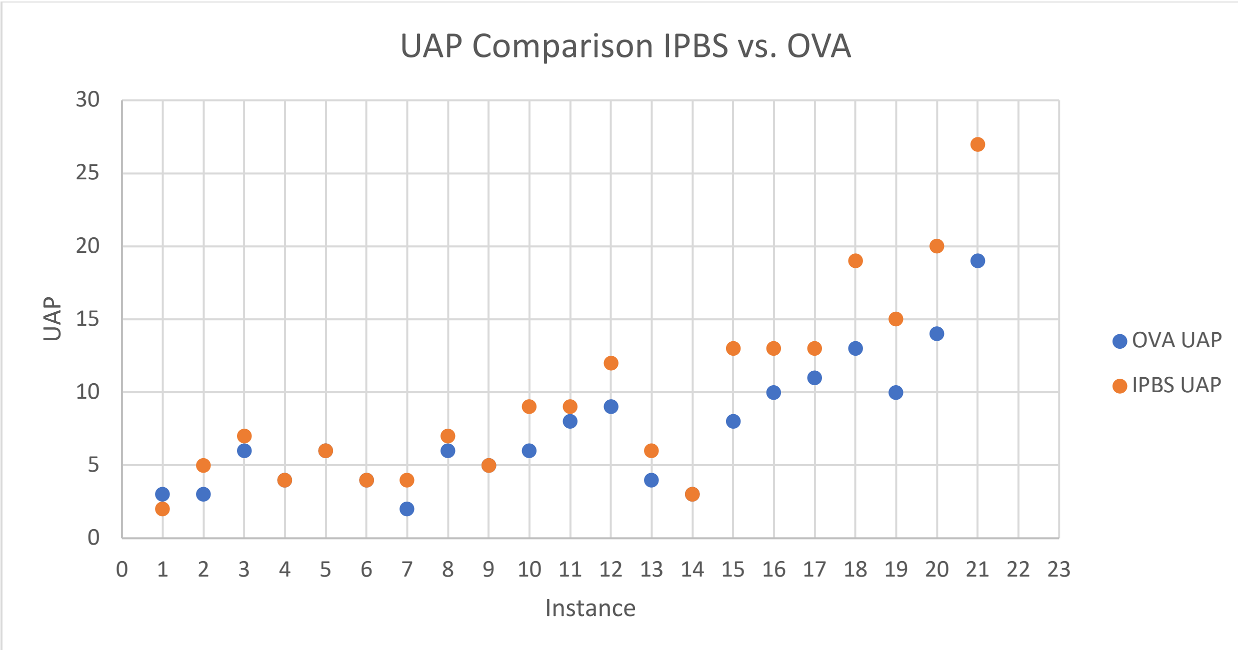

Figure 7 UAP Comparison Plot IPBS vs. OVA

4.6 AVERAGE PREFERENCE AWARDED (APA)

The average preference is calculated by dividing the sum of the highest preference a pilot ever gets awarded over the number of awarded pilots. As shown in table 7 and figure 6, in contrary to previous comparison analysis, although the APA for OVA model seems higher in instances 1, 8, and 14, the IPBS model is outperforming the OVA model in the rest of instances. Since the APA model is an average value, it can easily be affected by outliers. Defining a better performance measure for the preference awarded can be one of the future works for OVA and IPBS comparison. Personal bidding preference will also influence the result. The seniority situation of the entire ballot might also affect the final average preference.

Table 6 APA Comparison Table IPBS vs. OVA

Instance	Instance Size	IPBS APA	OVA APA	ΔAPA	ΔAPA (%)
1	7	2.643	1.385	-1.258	-47.61%
2	11	2.656	3.205	0.549	20.66%
3	14	2.688	2.576	-0.112	-4.16%
4	17	1.343	1.727	0.384	28.59%
5	17	1.800	1.882	0.082	4.58%
6	17	2.184	2.226	0.043	1.96%
7	17	1.833	2.613	0.780	42.52%
8	18	11.552	7.113	-4.439	-38.43%
9	21	2.108	2.279	0.172	8.15%
10	23	1.726	1.915	0.189	10.93%
11	25	2.304	2.175	-0.129	-5.59%
12	27	6.080	6.279	0.199	3.27%
13	29	1.964	2.311	0.348	17.71%
14	32	3.360	2.796	-0.565	-16.80%
15	38	1.696	4.127	2.431	143.32%
16	71	2.130	2.881	0.751	35.27%
17	75	1.778	2.122	0.344	19.34%
18	76	2.353	2.972	0.618	26.27%
19	78	2.671	3.081	0.410	15.33%
20	83	1.428	2.068	0.640	44.84%
21	97	2.524	3.429	0.905	35.86%

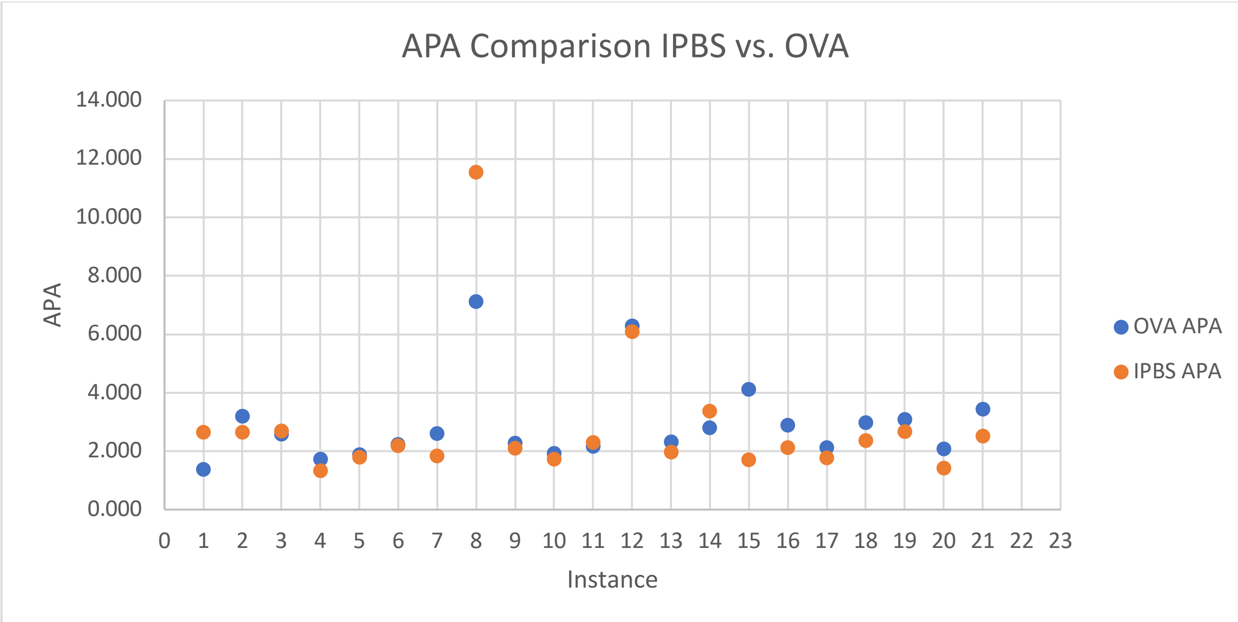

Figure 8 APA Comparison Plot IPBA vs. OVA

4.7 Computational Performance Analysis

Generally, the traditional heuristic models are considered faster than optimization solvers like CPLEX. This section of the report conducts a performance comparison between the IPBS model and the OVA model.

As shown in figure 7, the computation time for both the OVA model and the IPBS model increases as the instance size increases. However, the computation time for the IPBS model and the OVA model increases at a different rate. The smallest instance from the dataset contains seven pilots while the largest instance has 97 pilots. The heuristic model spends approximately 0.09s seconds on calculating the smallest instance, and it spends 0.37s on calculating the largest instance. The computation time for the heuristic model increases from the smallest instance to the largest instance by around 300%. Meanwhile, the optimization model increased from 0.17s for the smallest ballot to 1.97s for the largest ballot which is more than 1000%. However, since all of the instance sizes are less than 100 from the sample dataset provided, it may not be worth considering the computation time when choosing which model to use. These computation tests are done by a 2.4 GHz Intel Core i5 processor with 8 GB of 1600 MHz DDR3 memory.

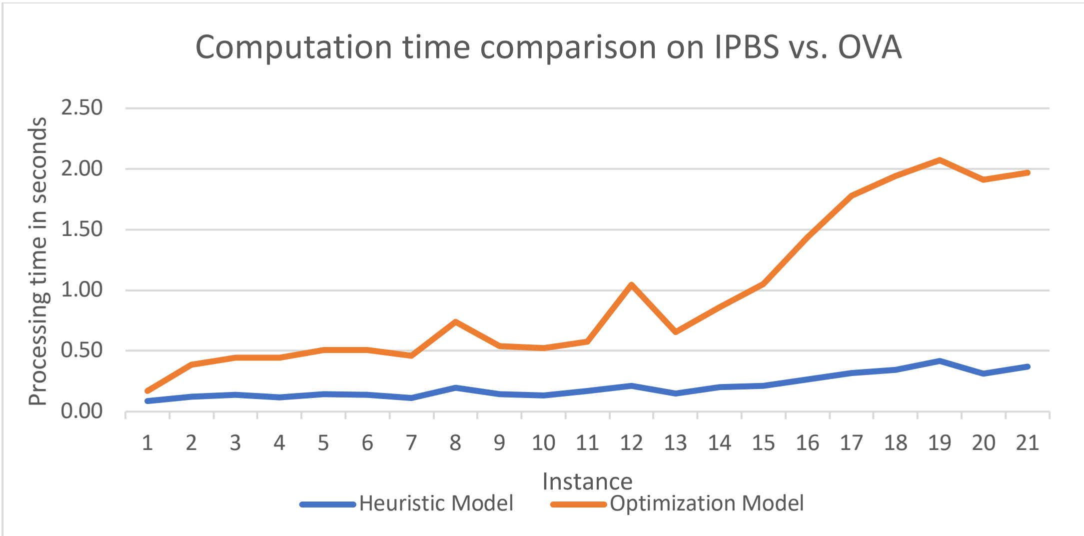

Figure 9 Processing Time Comparison on The IPBS vs. OVA

5. Sensitivity Analysis of the Heuristic PBS Model

In previous sections, the IPBS model and the OVA model are compared by running them for three passes with three maximum consecutive blocks and six weeks of the maximum amount of leaves. Changing the existing constraints of the IPBS model can potentially improve the IPBS model. The purpose of this chapter is to measure the performance improvement of the IPBS model under a different number of passes, a maximum number of consecutive blocks (parameter C in OVA model), and a maximum amount of leaves (parameter V in OVA model). This analysis will utilize the same data instances and performance measures from previous sections.

The plot type used in these three sensitivity analyses is heat map scatter plot. Darker dots represent larger instances, and smaller instances are in lighter colors. The x-axis ticks are different scenarios, and the y-axis is performance measures. The benefit of this plot is to simultaneously show the change of performance measure changes as parameters changes.

5.1 PASSES

Changing the number of passes is one of the easiest ways to alter the original IPBS model since it does not require adjustments in the model itself. The UAP number does not show any improvement as the number of passes increases. The changes in APA over the passes are also minimal. As a result, the comparison table for UAP and APA are omitted.

The UAS value comparison reveals some interesting findings. As shown in Figure 8, although the UAS value significantly decreases due to running the second pass, the improvement due to running the third pass is not apparent. Although the fourth and fifth passes are tested, they do not assign any more blocks to pilots. In other words, only the first three passes are useful regarding assigning more blocks. A similar behavior also happens in the OVA model. The OVA model will terminate when the algorithm stops to assign more blocks to pilots. In all 21 instances, the OVA model terminates after the second pass for all instances. The detailed comparison result is listed in table 8.

Table 7 UAS Comparison Table with Different Passes

Instance	Instance Size	UAS 1 pass	UAS 2 passes	UAS 3 passes	$\Delta\ UAS\ 2\ passes$	$\Delta\ UAS\ 3\ passes$
1	7	38	38	38	0	0
2	11	43	40	40	-3	0
3	14	59	53	53	-6	0
4	17	72	70	70	-2	0
5	17	71	71	71	0	0
6	17	65	65	65	0	0
7	17	61	57	57	-4	0
8	18	69	66	66	-3	0
9	21	78	76	76	-2	0
10	23	86	84	84	-2	0
11	25	96	93	93	-3	0
12	27	123	121	118	-2	-3
13	29	88	84	84	-4	0
14	32	129	123	123	-6	0
15	38	127	127	127	0	0
16	71	238	227	227	-11	0
17	75	221	216	216	-5	0
18	76	225	218	218	-7	0
19	78	275	258	255	-17	-3
20	83	291	285	285	-6	0
21	97	320	314	313	-6	-1

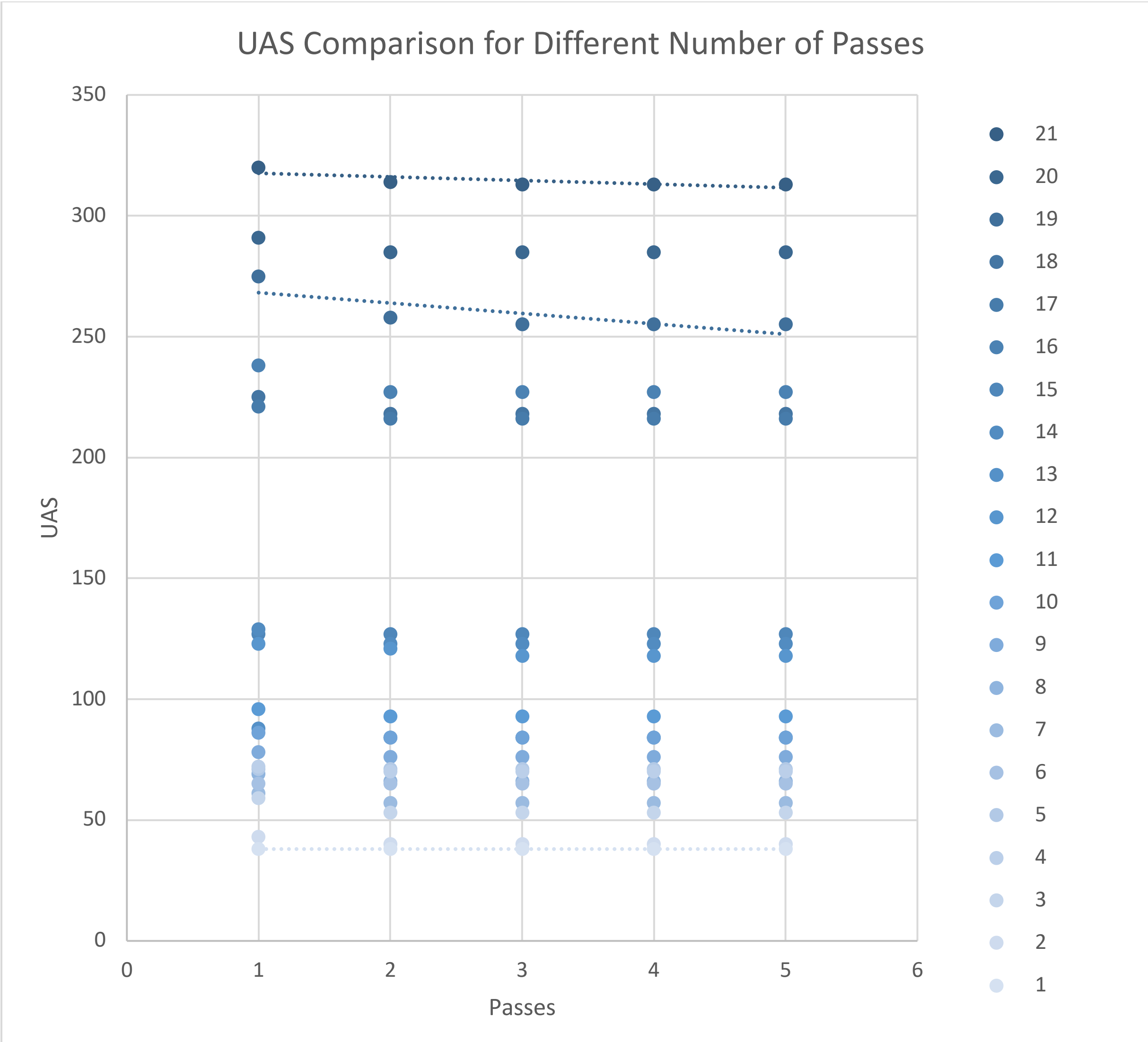

Figure 10 Comparison scatter plot of UAS among different number of passes

5.2 NUMBER OF CONSECUTIVE BLOCKS (C)

As shown in figure 9, the number of pilots without an award and the number of unassigned slots decreases dramatically as the maximum consecutive blocks constraint is relaxed. However, the average preference seems to remain steady. Figure 9 shows that as the number of pilots in an instance rises, the difference in UAP will also increase. In other words, bigger instances will not only have higher UAP value, but they will also have a more considerable difference between results using different maximum consecutive blocks. The phenomenon similarly occurs for the number of unassigned slots but on a smaller scale. The average preference, on the contrary, does not reveal any pattern as the maximum consecutive blocks change.

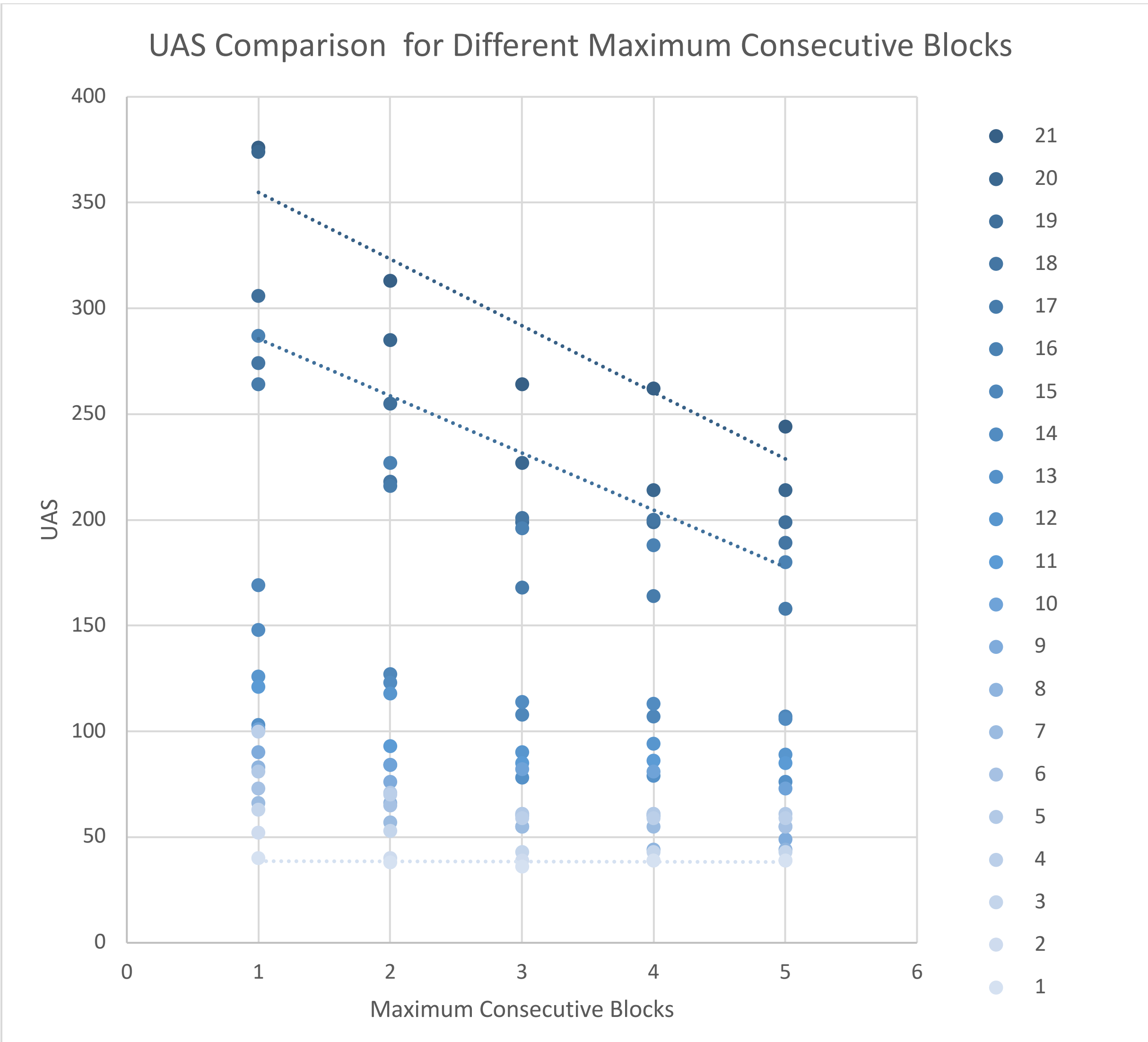

Figure 11 Comparison Scatter Plot UAS Using Consecutive Block

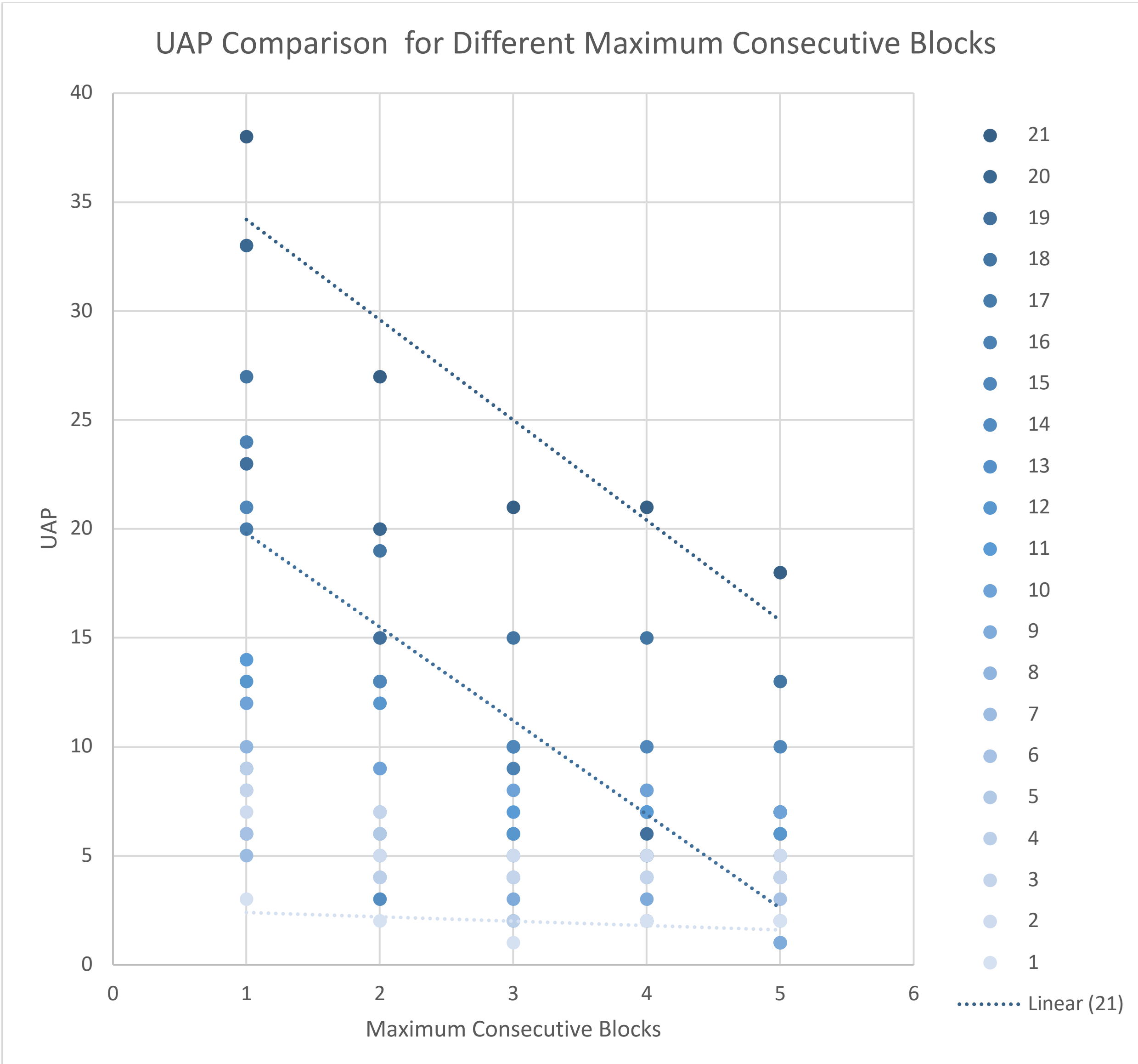

Figure 12 UAP Over a Different Number of Consecutive Blocks

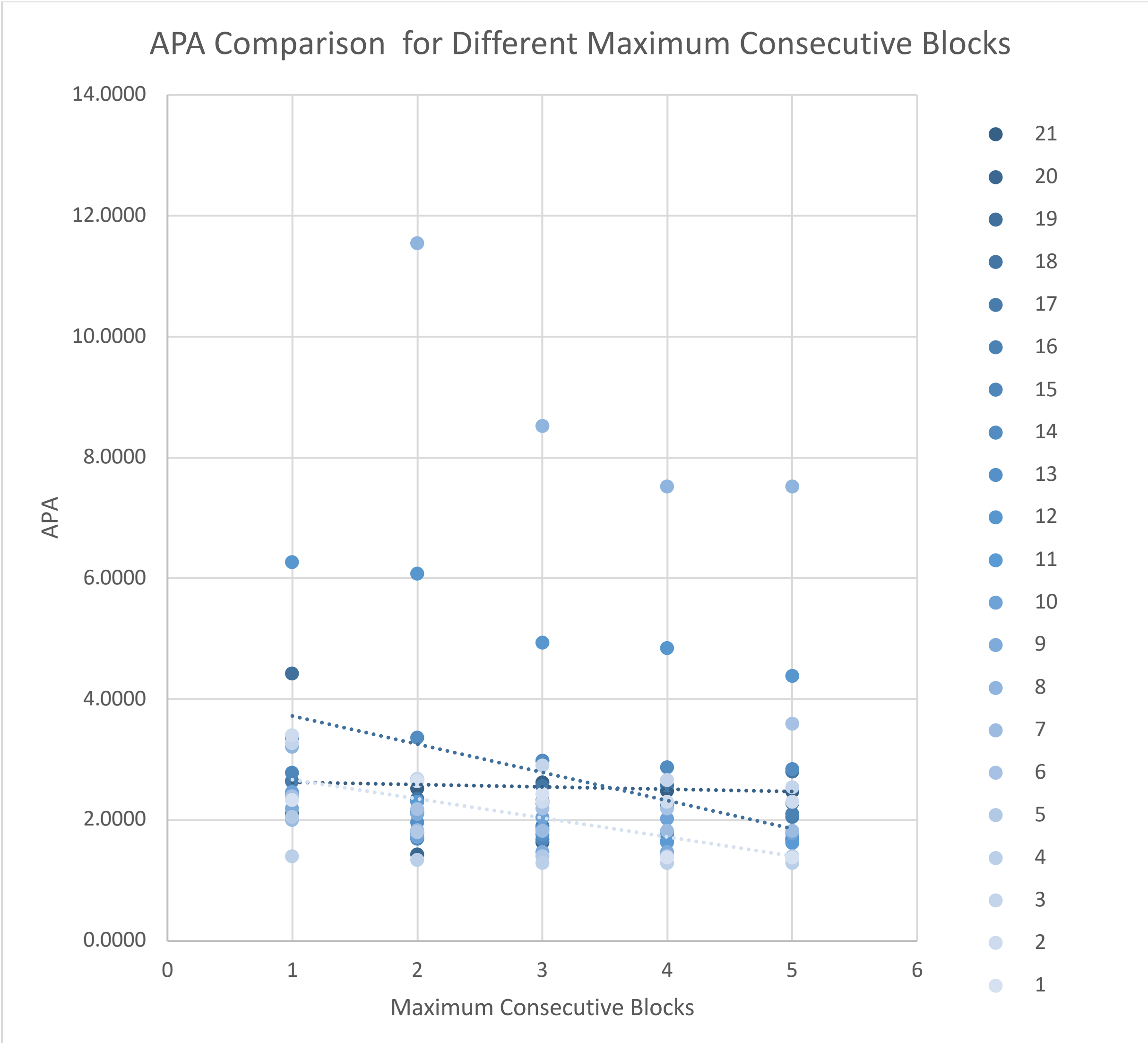

Figure 13 Comparison of Average Preference for Difference Maximum of Consecutive Blocks

5.3 Maximum Amount of Leaves (V)

This comparison may be confusing at the beginning because pilots are only allowed to bid for six weeks per preference. Since all the comparisons conducted are based on the multi-pass IPBS algorithm, the PBS algorithm can award a pilot with more than six blocks, but it is stopped by the maximum number of blocks constraint. Hence, this section tests the effect of the different maximum amounts of leaves.

Interestingly, as the maximum amount of leaves increases, the average preference value gradually increases, but the number of unassigned slots will decrease. This result can possibly be explained by the blocks later awarded after the first six weeks are filled with a higher overall preference value. Hence, increasing the maximum amount of leaves for each pilot will not significantly improve the average preference value nor decrease the number of unassigned slots. The maximum reduction is in instance 12 where the UAS value reduces from 118 to 112. Figure 15 shows that the reduction on UAS by increasing the maximum allowed leaves is not significant.

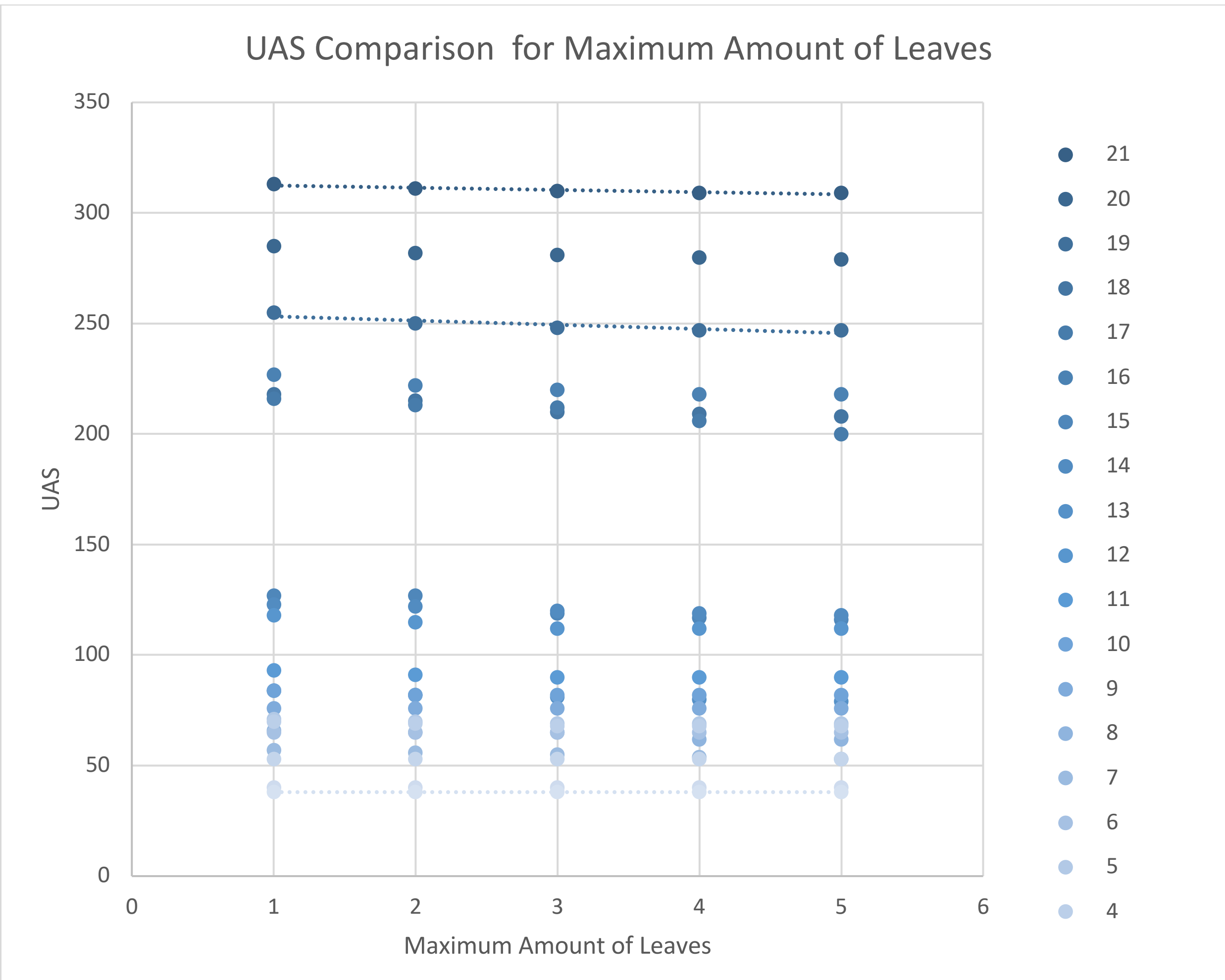

Figure 14 Comparison of UAS Using the Different Maximum Amount of Leaves

6. Conclusions and Future Works

The heuristic PBS model is widely used in the airline industry for the crew vacation allocation problem. With the newly engineered optimization vacation allocation model, this report compared the heuristic PBS model and the optimization model and compared several models of PBS under different scenarios. The optimization model outperforms the heuristic model when using unassigned slots and the number of pilots without bids. However, the heuristic model seems to have better performance on average preference. This might be due to the fact that the heuristic model awards pilots with fewer blocks but awards them with their top preferences. The IPBS runs faster than the OVA model, but the difference is negligible considering the instance size never goes over 100. Increasing the maximum number of the consecutive weeks will decrease the number of unassigned slots and the number of pilots without awards. However, the average preference values do not show a clear pattern as the maximum number of consecutive blocks changes. The maximum amount of leaves does not improve the results significantly because the data instance only has six weeks on each preference. However, as the maximum amount of leaves increase, the number of unassigned slots decreases and the average preference increases. Adding more passes to the algorithm seems not to improve UAP and APA effectively, but the number of unassigned slots drops in the first three passes.

Future works on comparative analysis could include a sensitivity analysis of the OVA model by changing the maximum number of consecutive blocks and the maximum amount of leaves. Analyzing the effect of changing the capacity on individual weeks can also be an exciting project. Collecting more data instances from a broader range of airline companies and the broader time scale will also benefit the sensitivity analysis.

The natural extension of the optimization based model would be applications to other industries like healthcare, air traffic controllers, train engineers, and other highly-specific jobs. It should be relatively easy to integrate this methodology into a vacation planning allocation for the reserve crews for the airlines. Due to the somewhat low priority and cost of the reserve pilots, there are not many current efforts to optimize their vacation allocation. However, this model should fit seamlessly into the problem and provide an extra layer of protection for the airlines to hedge against other stochastic factors like sick days or maintenance delays.

BIBLIOGRAPHY

Fisher, T., Clough, M., Gel, E., & Jacobs, T. (2015). A reliability driven model for airline crew vacation grid optimization. In AGIFORS 55th Annual Symposium: Analytics for Efficiency and Customer Centric Optimization Airline Group of the International Federation of Operations Research Societies (AGIFORS).

Holm, Å. (2008). Manpower Planning in Airlines: Modeling and Optimization (Dissertation). Linköping University, Department of Mathematics. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-14757

Kinnunen, T. (2016). Cost-efficient vacation planning with variable workforce demand and manpower (Thesis). Aalto University. Retrieved from http://urn.fi/URN:NBN:fi:aalto-201611025356

Montazeri, A., Patrick, J., Michalowski, W., & Banerjee, D. (2015). Developing the Pathologists' Monthly Assignment Schedule: A Case Study at the Division of Anatomical Pathology of The Ottawa Hospital. *AMIA Annual Symposium proceedings. AMIA Symposium*, *2015*, 933–942.