

## USING A SET PACKING FORMULATION TO SOLVE AIRLINE SEAT ALLOCATION/REALLOCATION PROBLEMS

Akira Tajima      Shinji Misono  
*IBM Research, Tokyo Research Laboratory*

(Received August 11, 1997; Revised October 19, 1998)

*Abstract* We report our experience in developing prototype solutions for two optimization problems faced by an airline company. Whereas previous work has focused on minimizing direct costs or maximizing direct profits, both problems in this case involve maximizing customer satisfaction. One is the seat allocation problem: given a set of groups of passengers, find the optimal assignment of passengers to seats in an aircraft so that each member of a group sits as near to the others as possible. The other is the seat reallocation problem, which arises when the aircraft is changed just before departure. It consists in finding the optimal reallocation of passengers in a new aircraft so that the original seat configuration is retained as far as possible. We formulate both as instances of the set packing problem and propose efficient methods for generating promising candidate subsets. We show through numerical experiments with real-life data that our approach is practical as regards both the computation time and the quality of solutions.

### 1. Introduction

For the past twenty years, the airline industry has been a rich field for research on optimization problems. Some representative optimization problems in the field are the crew-pairing problem [1, 2, 3, 10, 11, 12, 14, 16], the crew-rostering problem [6, 13, 17], and the seat-inventory-control problem [4, 5, 7, 8, 9, 15]. So far, most of the optimization research in this field has been directly related to airlines' costs and profits. The crew-pairing problem, for example, consists in finding the crew schedule that minimizes the crew's cost.

As competition between airlines becomes more intense, better technology is required in order to reduce cost. In addition, *customer satisfaction*, which is rather indirectly related to cost or profit, has recently attracted stronger interest as one of the key factors for differentiating a company from its competitors.

We report our experience in applying optimization technologies to two problems related to customer satisfaction that were faced by Japan Airlines (JAL). The first one is the *seat allocation problem*, which concerns one of the airline's daily operations: given a list of passenger groups, most of which have more than one member, and the seat layout of an aircraft, find a way to assign seats to passengers so that the customer satisfaction is maximized. Here, greater customer satisfaction is realized when all the members of each group can sit close to each other. The second problem is the *seat reallocation problem*, which concerns the situation when the aircraft is suddenly changed after the seat assignment has been announced to all passengers. The problem consists in reassigning seats in the new aircraft to passengers so that the seat assignment in the original aircraft is most closely reproduced.

We formulate both problems as instances of the *set packing problem* (SPP), and propose a new method for generating columns in the SPP, or candidate subsets. We show through

numerical experiments with real-life data that our approach is practical. Our prototypes have been revised and implemented by JAL, and have been in daily use since April 1997.

## 2. Seat Allocation Problem

### 2.1. Problem statement

Seat allocation is one of the airline company's daily operations before ticketing. The company accepts reservations for economy-class seats on international flights through travel agencies, but does not actually assign seats until a certain time at which the reservation process is considered to be complete. The unit of seat reservation is not for an individual passenger, but for a group, consisting of members who are supposed to travel together. After the reservation process is complete, seat assignment is performed for each group. The primary objective of the seat assignment is to allocate passengers in an aircraft so that seats for members of a group are as close to each other as possible.

When a reservation is made, the airline company receives the following group information from the travel agency: how many members of a group are male/female/infant, how many prefer non-smoking/smoking seats, and whether a group has the *privileged status*. The privileged status means that the group is a good client of the agency, and that it should be treated warmly, including seat locations. An aircraft is divided into a non-smoking zone and a smoking zone. Each zone may consist of more than one compartment, a set of rows which are separated from other rows by a central wall.

The seat allocation problem consists in finding how to allocate passengers in units of a group or a subgroup so that following preferences are maximized:

**Zone Preference 1:** If a group specifies how many members prefer non-smoking/smoking zone, this preference should be satisfied as far as possible.

**Zone Preference 2:** If a group has privileged status and no zone preference is specified, all members of the group should be assigned to the non-smoking zone as far as possible.

**Seat Preference 1:** All group members in a compartment should sit close to each other as far as possible. In particular, a member should have at least one member of the same group in the adjacent seat(s). Here an adjacent seat refers to a physically adjacent seat in the same row; this does not include a neighboring seat across an aisle.

**Seat Preference 2:** Members of a group with privileged status should be assigned to seats with lower row numbers in a zone as far as possible.

**Seat Preference 3:** Vacancy in each compartment in each zone should be concentrated into the front part of the compartment as much as possible. In other words, vacant seats between two groups and between members of the same group, which we will refer as *gaps* hereafter, should be avoided as much as possible.

Zone Preferences 1, 2 and Seat Preferences 1, 2 are to improve passengers' satisfaction, whereas Seat Preference 3 is to maximize the flexibility of the airline in handling last-minute changes in reservations. There are other conditions in addition to these preferences. For example, a group cannot be assigned to two or more compartments if the number of its members is less than the capacity of one compartment.

When a group does not have any non-smoking/smoking preference or privileged status flag, we first need to decide which zone to allocate its members to. The airline company uses the following rules for zone selection, which it believes satisfactorily reflect the statistics on non-smoking and smoking preference, and which are therefore good for customer satisfaction:

**Rule 1:** When a group has more than 15 members, allocate approximately half of them to the non-smoking zone, and the other half to the smoking zone.

**Rule 2:** Otherwise, allocate all members to either the non-smoking or smoking zone without splitting them into subgroups, following detailed rules. For example, if a group has at least one infant member, assign all its members to seats in the non-smoking zone.

These rules should be obeyed whenever possible.

## 2.2. Approach

### 2.2.1. Set packing formulation

We formulate the problem as a class of integer linear programming (ILP) known as the set packing problem (SPP). Intuitively, we first enumerate a set of *good* candidates of destinations for each group, represent the quality of each candidate as a cost, and select one destination for each group so that the selected destinations do not intersect with each other - that is, do not involve the same seat being occupied by more than one passenger - and so that their total cost is minimized.

Here, there are no constraints among groups, and the non-linearity of the problem can be encapsulated within a candidate. Thus, after the candidates have been enumerated, the formulation becomes linear, and efficient domain-independent algorithms are available. We focus solely on enumerating good candidates and evaluating them, which is the only domain-dependent procedure. The enumeration procedure is described in Section 2.2.3.

To formulate the problem as an SPP, we use a group as the unit of assignment. Suppose that we have  $p$  groups, that the  $k$ -th group has  $q_k$  candidate destinations, and that there are  $s$  available seats in the plane. The constraint matrix  $A$  has  $m$  rows and  $n$  columns, where  $m = s + p$  and  $n = \sum_{k=1}^p q_k$ .  $a_{ij}$  represents the element of  $A$  in the  $i$ -th row and  $j$ -th column.

$$\text{minimize } \sum_{j=1}^n c_j x_j \quad (2.1)$$

s.t.

$$\sum_{j=1}^n a_{ij} x_j \leq 1 \quad (1 \leq i \leq s) \quad (2.2)$$

$$\sum_{j=1}^n a_{ij} x_j = 1 \quad (s + 1 \leq i \leq m) \quad (2.3)$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n$$

where

$$x_j = \begin{cases} 1 & \text{if candidate } j \text{ is adapted in the assignment} \\ 0 & \text{otherwise} \end{cases}$$

and  $c_j$  is the cost of candidate  $j$ .

For  $1 \leq i \leq s$ ,

$$a_{ij} = \begin{cases} 1 & \text{if candidate } j \text{ occupies seat } i \\ 0 & \text{otherwise.} \end{cases}$$

Thus, equation (2.2) specifies that each seat should be occupied by at most one group.

For  $s + 1 \leq i \leq m$ ,

$$a_{ij} = \begin{cases} 1 & \sum_{k=1}^{i-s-1} q_k + 1 \leq j \leq \sum_{k=1}^{i-s} q_k \\ 0 & \text{otherwise.} \end{cases}$$

Equation (2.3) specifies that each group (the  $i - s$ -th group in the equation) has exactly one destination.

The complete procedure in our approach is as follows:

- Assign groups to the smoking or non-smoking zone so that Zone Preferences 1 and 2 are satisfied.
- For each zone,

**Step1:** Enumerate the candidates for each group.

**Step2:** Evaluate each candidate and determine its cost.

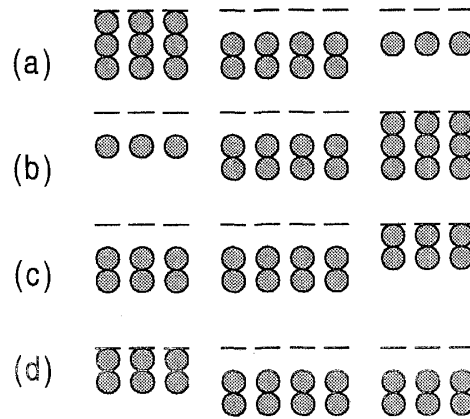


Figure 1: Four variations in shapes of candidates

**Step3:** Formulate the problem as an instance of the SPP and solve it.

The details of each step are described in the following sections.

### 2.2.2. Assigning groups to zones

We determine how many members of each group are to be assigned to the non-smoking/smoking zone so that Zone Preference 1 is strictly satisfied and Zone Preference 2 and Rules 1 and 2 are satisfied as far as possible. If no assignment that fulfills all the conditions exist, owing to the capacity limitation of each zone, we relax the conditions in the order of Rule 2, Rule 1, and Zone Preference 2.

### 2.2.3. Enumerating candidates

Here we describe how to enumerate *good* candidates. This part is highly dependent on the problem, and differs most markedly from the seat re-allocation problem in Section 3.

To reduce the search space of the optimization, we only consider candidate assignments for each group that largely satisfy Seat Preferences 1 and 3 (section 2.1). More precisely, we generate only candidates that satisfy the following conditions:

**Condition 1:** No empty seats between members of a group are allowed within a row.

**Condition 2:** When group members are assigned to more than one row,

- (a) each assigned row has to include one of window seat (A or K).
- (b) except for the *top* and *bottom assigned rows*, all assigned rows have to be filled completely by the members of a group.
- (c) all assigned rows must be adjacent to each other.

Here the *top* (resp. *bottom*)*assigned row* refers to the row with the lowest (resp. highest) row number among the rows to which the members of a group are assigned. With these restrictions, it suffices to consider the four configurations depicted in Figure 1. These configurations differ in which window seat (A or K) is included in the top assigned row and the bottom assigned row. In configuration (a) in Figure 1 both the top and bottom assigned rows start with seat A, whereas in configuration (d) the top assigned row starts with seat A and the bottom assigned row with seat K. We can easily enumerate candidates

for each group with these four variations by sweeping all seats in a time linear to the number of seats.

#### 2.2.4. Calculating costs

We consider only candidates that strictly satisfy Zone Preferences 1 and 2, which are described in subsection 2.1. The cost is then specified so that it reflects the extent to which the candidate satisfies Seat Preferences 1, 2, and 3. The following are the preferences and the corresponding costs that we considered in the prototype system:

- *Keep the members of a group close to each other* (Seat Preference 1).  
Consider  $C^D$ , whose value is inversely proportional to the density of the group, which is defined by  $N_{pop}/(N_{row} \times N_{width})$ , where  $N_{pop}$  is the population of the group,  $N_{row}$  is the number of rows in the candidate, and  $N_{width}$  is the number of seats in a row.
- *Keep groups with privileged status in front of other groups* (Seat Preference 2).  
We estimate in each zone the highest possible row number,  $R_c$ , to which a group with privileged status can be assigned without letting any members of groups without privileged status sit in front of it. We fill out seats in each zone from the rear to the front with the passengers who do not have privileged status, and adopt as  $R_c$  the highest row number in which at least one empty seat exists after the filling out is complete. We then penalize candidate assignments that occupy seats in rows whose row numbers are higher than or equal to  $R_c$ . More precisely, we define the cost  $C^F$  regarding Seat Preference 2 as

$$C^F = \begin{cases} C^f & \text{if the member belongs to one of groups with privilege status} \\ & \text{and the seat is behind } R_c. \\ 0 & \text{otherwise,} \end{cases}$$

where  $C^f$  is a fixed value sufficiently larger than  $C^P$ .

- *Minimize the number of gaps* (Seat Preference 3).  
Consider  $C^P$ , whose value is super-linear to the distance  $d$  from the tail row,  $O(d^2)$  for example. We should not set the cost to be linear, in order to avoid the situation such that one single group with a small population suffers quite a large inconvenience. This cost is assigned to the members of the group.

The cost  $C$  of a candidate for a group with  $a$  members is then calculated as

$$C = \sum_{i=1}^a (C_i^P + C_i^F) + C^D.$$

### 2.3. Numerical experiments

#### 2.3.1. Experimental setups

We built our prototype by using the IBM Optimization Subroutine Library to solve set packing problems (SPPs) and their linear programming relaxations (LPRs). In solving the SPP we use the branch-and-bound method, and terminate the calculation as soon as the first feasible solution is found. The typical size of the SPP is 300 rows and thousands of columns. All runs were made on an IBM RS/6000 model 990.

We discuss numerical experiments using four real-life data sets supplied by the airline company, whose contents are summarized in Table 1. For each set of data, the first column represents an aircraft's capacity in each zone, the second column lists the number of passengers who should be assigned to the non-smoking zone (the first row) and the smoking zone (the second row) in order to fully satisfy the conditions stated in Preferences 1 and 2. The third row in the second column is the number of passengers for whom the zone should be selected according to rules 1 and 2. The third column lists similar information to the second

column with a group as a unit. For example, in Data Set A all members of ten groups and some members of two groups should be assigned to the non-smoking zone.

Table 1: Data set statistics

	A			B			C			D		
	Cap.	Pass.	Grp	Cap.	Pass.	Grp	Cap.	Pass.	Grp	Cap.	Pass.	Grp
Non-Smoking	312	246	10+2	242	115	5+1	322	307	13+3	157	96	4+0
Smoking	142	56	2+2	128	8	0+1	138	31	1+3	141	0	0+0
Any	-	99	15	-	244	9	-	43	7	-	174	15
Total	454	401	29	370	367	15	460	381	24	298	270	19

### 2.3.2. Allocation results

Figure 2.3.2 shows the seat allocations for Data Set A. The corresponding aircraft consists of the non-smoking zone (rows 1 through 42) and the smoking zone (rows 43 through 59). The non-smoking zone is divided into three compartments: rows 1 to 18, 19 to 31, and 32 to 42.

Figure 2.3.2(a) shows the original seat allocation designed by the airline company. This seat assignment is problematic in the following four respects. First, it does not satisfy Preferences 1 and 2 in zone selection: 44 out of 201 passengers who explicitly specified either the smoking or the non-smoking zone are not seated in their requested zone. 27 out of 101 passengers who have privileged status are seated in the smoking zone. We will refer to the number of passengers who are not assigned to the zone specified by Preferences 1 or 2 as the *number of violations*. Second, there are 13 isolated group members in this assignment. For example, a member of group 09 in Seat K in row 20 has no neighbors from the same group. We add shading to isolated members (Note that seats D and G in rows 7 and 8 are physically adjacent). Third, two groups, 12 and 17, are split into subgroups in one compartment. Such splitting should be avoided as much as possible. Finally, there are as many as 41 gaps, which should be minimized, as stated in Preference 4.

Figure 2.3.2(b) shows the seat assignment when we partly apply our method; that is to say, we assume that the assignment of (sub) groups to the non-smoking and the smoking zone is given (in the original seat assignment), and solve the set packing problem. As shown in this figure, this seat assignment is completely free of isolated members, and of group splitting. Also, the number of gaps is reduced from 41 to 26. Figure 2.3.2(c) shows the seat assignment when we fully apply our method. Since we create candidate seat assignments so that Zone Preferences 1 and 2 are strictly satisfied, there are no violations in the resulting assignment; the number of violations is reduced from 71 to zero. The number of gaps is also reduced, from 41 to 29.

The results for all four data sets are summarized in Table 2. For each data set, the table lists the number of isolated members (row 4), the number of group splittings (row 5), the number of gaps (row 6), and the number of violations (row 7). The first column for each data set is for the original assignment, the second is for the assignment by our method with the same zone selection as the original one, and the third column is for the assignment by our method with our own zone selection algorithm.

### 2.3.3. CPU time

Table 3 summarizes the CPU time spent on solving the SPP and the total CPU time for data sets A, B, C, and D. The CPU time for the SPP for data set B is more than four times

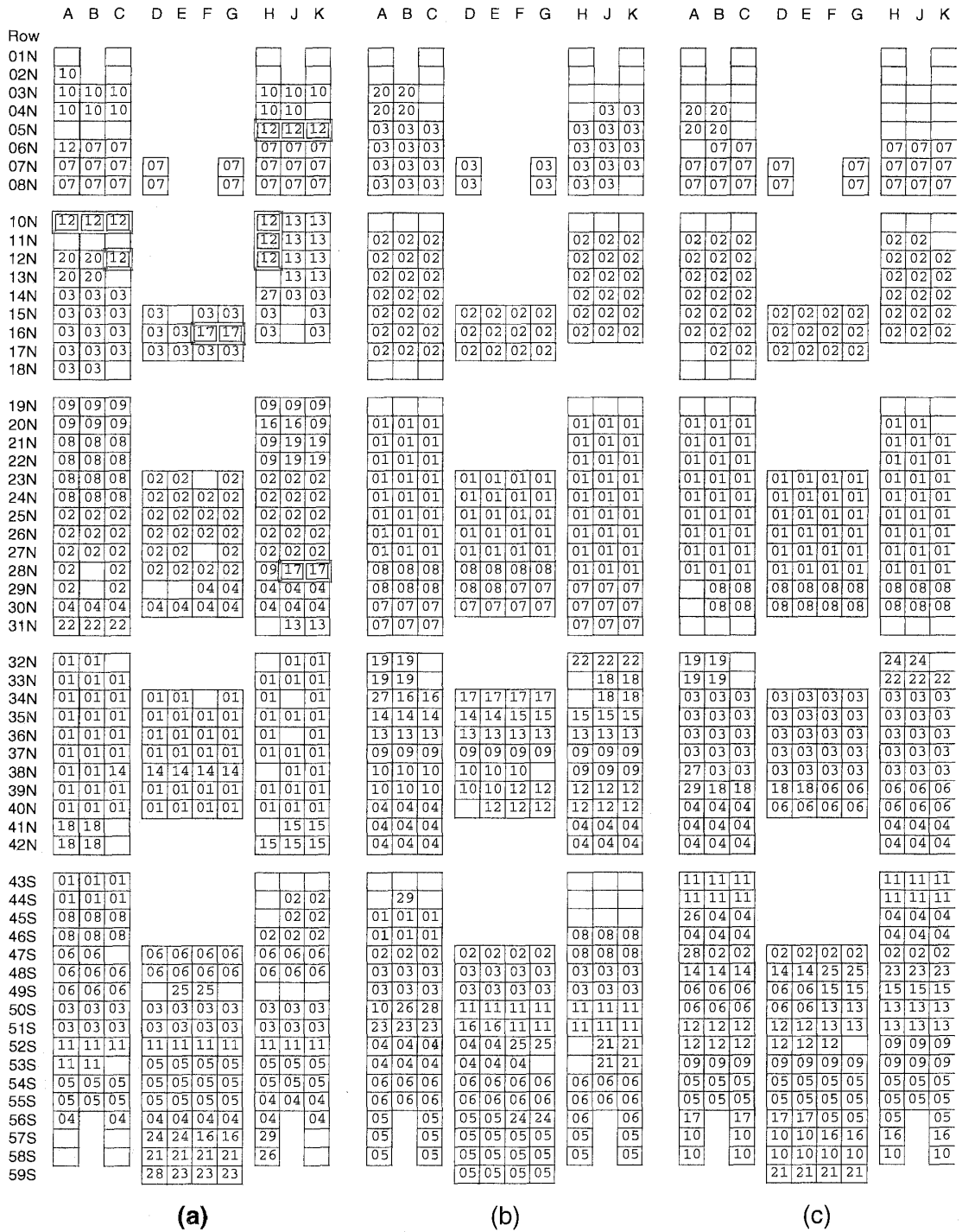


Figure 2: Example of seat allocation

Table 2: Results

	A			B			C			D		
	Orig.	Our		Orig.	Our		Orig.	Our		Orig.	Our	
		Part	Full		Part	Full		Part	Full		Part	Full
Isolation	13	0	0	11	1	1	1	0	0	2	0	0
Splitting	2	0	0	0	0	0	0	0	0	1	0	0
Gap	41	26	29	2	2	2	38	38	5	24	19	3
Violation	71	71	0	0	0	0	36	36	0	27	27	0

Table 3: Total CPU time

Dataset	A	B	C	D
SPP (sec.)	8.0	39.7	10.0	7.8
TOTAL (sec.)	9.7	43.8	12.6	9.7

larger than those for the other three data sets. This can be attributed to the difference in the number of empty seats, rather than the difference in the total number of passengers. The number of empty seats in data set B is 3 (out of 370), while for A, C, and D it is 53 (out of 454), 79 (out of 460), and 28 (out of 298), respectively.

To clarify this dependence, we artificially change the number of empty seats in the non-smoking zone and examine the corresponding CPU times. The results are shown in Table 4. When there are no empty seats, much more CPU time is necessary. If, however, we formulate the problem as set partitioning instead of set packing, that is, if we replace the inequality condition in equation (2.2) with the equality condition because every seat is occupied, the CPU time is considerably reduced.

### 3. Seat Reallocation

#### 3.1. Problem statement

The seat re-assignment problem arises when an aircraft is replaced due to mechanical trouble before departure. By that time, passengers know which seat they were allocated in the original aircraft. In order to maximize customer satisfaction, it is desirable to re-assign seats to passengers so that original seat assignment be retained as far as possible in the new aircraft. What makes the seat reassignment problem non-trivial is that the physical seat layout differs from aircraft to aircraft.

The first priority is to allocate passengers to the same class, zone (non-smoking/smoking), and deck (upper/lower) as in the original configuration, whenever possible. The next priority is to retain other properties of the original configuration as far as possible. The properties that should be retained include column attributes (window/aisle/middle, etc.) and row

Table 4: CPU time for SPP and the number of empty seats

No. of empty seats	4/322	3/322	2/322	1/322	0/322	0 (partitioning)
CPU time	25.20	34.26	39.79	38.50	147.38	59.22



numbers relative to the front seat. When two or more passengers travel together, they are assigned a group id. For each group, the relative positions of its members should be retained in the new aircraft as far as possible. Unlike the seat assignment problem in the previous section, all members of a group do not always sit close to each other. Some members may prefer to sit a few rows behind others.

The seat reassignment problem is stated as follows: given the seat assignment in the original aircraft and the seat layout in the new aircraft, find the seat assignment in the new aircraft, for each class and each zone, such that following preferences are best satisfied:

**Preference 1:** Passengers originally assigned to the upper deck should be seated in the upper deck, if any, of the new aircraft.

**Preference 2:** Changes in the column attributes of individual passengers should be minimized. Here, the column attribute indicates whether a seat is window seat (column number A or K), an aisle seat next to a window seat (C or H), an aisle seat (D or G), or a middle seat (E).

**Preference 3:** The change in the number of rows from the front seat should be minimized for each passenger.

**Preference 4:** For each group, the relative positions of its members should be retained as far as possible in the new aircraft.

The computation time available for solving the seat reassignment problem is quite limited, since the problem usually occurs unexpectedly, often just before departure, and a new assignment should be announced to passengers as soon as possible. The airline company set a limit of a few minutes for the computation time on a workstation.

Up to now, the airline company has done seat reassignment manually. Because of the complexity of the problem, the airline has limited reassignment only to the first and connoisseur classes. Reassignment for the economy class seats would lead to better customer satisfaction if an efficient method is implemented.

## 3.2. Approach

We can use the same formulation as for the case described in section 2, that is, use a group as the unit of assignment. The assignment of individuals in a group can be done easily, because the number of individuals in a group is small, and the shape of a group's destination is similar to that for the original seats, as we will describe in subsection 3.2.1.

We do not have to consider the smoking or non-smoking zone, because this is specified when the reservation is made. In this section, therefore, we will only describe how to enumerate candidates and evaluate them.

### 3.2.1. Enumerating candidates

The most important consideration is that the relative positions within a group should be preserved as far as possible. The relative positions reflect various intentions of passengers, and we cannot introduce any assumptions about the arrangement such as the *closer the better*.

We then generate two kinds of candidates, rigid ones and relaxed ones. The rigid ones are generated by parallelly shifting and/or inverting the original assignments in the new aircraft. The relaxed ones are generated by slightly modifying the relative positions within a group. That is to say, we first modify the shape of the assignment by shifting a part of the group divided by an aisle a few rows forward or backward, then generate candidates in the same way as rigid ones.

In most cases, we can obtain feasible solutions. In some cases, however, no candidates are available for some groups, because the seat arrangement of the plane has changed so

Table 5: Seat reallocation results

Dataset	Class	Groups	Adopted candidates			CPU time (sec.)
			Rigid	Relaxed	Enumerated	
A	First,non-smoking	6	6	0	0	0.2
	First,smoking	1	0	0	1	0.1
	Business,non-smoking	23	22	1	0	0.3
	Business,smoking	14	14	0	0	0.2
B	Business,Non-smoking	14	14	0	0	0.2
	Business,Smoking	11	11	0	0	0.2

much that the number of seats in a row or the number of rows are insufficient for the groups. We then have to modify the shapes of their assignments completely. We enumerate all the horizontally connected candidates as we have done for the seat allocation problem in Section 2.2.3.

### 3.2.2. Calculating costs

The passengers have already been informed of their seat positions, and the difference between the original and modified positions must be minimized. Further, changes in the attributes of the seats should be avoided. We therefore introduce two kinds of costs for individuals,  $C^p$ , super-linear to the distance from the original position, and  $C^a$ , the sum of given fixed values for each violated attribute.

The cost for groups  $C^g$  reflects the relative positions between members of the groups,

$$C^g = \begin{cases} 0 & \text{rigid candidates} \\ C^r & \text{relaxed candidates} \\ C^d & \text{enumerated candidates.} \end{cases}$$

$C^r$  is super-linear to the number of row-shifts, and  $C^d$  is defined in the same way as  $C^D$  in Section 2.2.4. Thus, the cost  $C$  of a candidate for a group with  $a$  members is then calculated as

$$C = \sum_{i=1}^a (C_i^p + C_i^a) + C^g.$$

### 3.3. Numerical experiments

We built our prototype by using the IBM Optimization Subroutine Library to solve set packing problems (SPPs) and their linear programming relaxations (LPRs). The typical size of the SPP is 70 rows and 700 columns, which is much smaller than that in the seat assignment problem. All runs were made on an IBM RS/6000 model 990.

We used four datasets provided by the airline company. Two of them are very sparse, and most of the passengers occupy the same seats in the new planes. Table 5 shows the results with the other two datasets. Rigid candidates are assigned for most groups, and the CPU times are short. The relaxed LPPs mostly have integer solutions; only a few branching procedures are needed to obtain the optimal solution.

This is because the problem size is small in comparison with that of the seat allocation problem in Section 2, and the allocation information for the original plane is quite useful.

Figure 3.3 shows an example of seat reassignment. Figure 3.3(a) shows the original assignment, in which rows 1 to 7 are first class and rows 10 to 33 are connoisseur class. Rows 15 and 16 are in the upper deck. Figure 3.3(b) shows the result of our method as well as the seat layout of the new aircraft. The number of passengers for each column attribute

Column	A	C	D	E	G	H	K
Row							
01N	a	a				a	
02N	-	-				b	b
03N	c	c				c	
04N	c	c					d
05S	e	e					
06S	e	-					
07S	f	f					
10N							
11N			g		-	j	j
12N	i	i	h		g	-	-
13N	k	k				-	-
15NU	-	-					-
16NU	l	l				l	l
24N	-	-				m	m
25N	n	n				o	o
26N	p	p				o	o
27N	n	n				-	-
28N	q	q	q	r	r	-	-
29S							
30S	n	-				-	-
31S	s	s	-	-	-	t	t
32S	-	-				-	u
33S	-	h				u	u

(a)

01N	01A (a)	01C (a)				01H (a)	02K (b)
02N	02A (-)	02C (-)				02H (b)	03A (a)
03N	04A (a)	03C (a)	04C (a)			03H (a)	04K (d)
04S	05A (e)	06A (e)	05C (e)		06C (-)	07C (f)	07A (f)
09S	30A (n)	30C (-)					
10S						32H (-)	32K (u)
11S	31A (s)	31C (s)	31D (-)	31E (-)		33H (u)	33K (u)
12S	32A (-)	32C (-)			32G (-)	30H (-)	
13S	33A (-)	33C (h)				31H (t)	31K (t)
27N	12A (b)	12C (b)	11G (-)		12H (-)	11H (j)	11K (j)
28N	13A (k)	13C (k)	11D (g)			13H (-)	15A (-)
29N	16A (l)	16C (l)	12D (h)		12G (g)	16H (l)	16K (l)
30N	26A (p)	26C (p)	24A (-)			27H (-)	15K (-)
31N	25A (n)	25C (n)	24C (-)	27K (-)	28H (-)	24H (m)	24K (m)
32N	28A (q)	28C (q)	28D (q)	28E (r)	28G (r)	25H (o)	25K (o)
33N	27A (n)	27C (n)				26H (o)	26K (o)

(b)

Figure 3: Example of seat reallocation: (a) original allocation (b) reallocation

and the corresponding capacity of the new aircraft are summarized in Table 6. This table shows a shortage of columns AK and CH in both the first and connoisseur classes.

Our result gives the smallest possible number of changes in column attributes (namely, 5). The original configuration of each group is also closely followed.

Table 6: Passengers and capacity

	First class								Connoisseur class							
	Non-smoking zone				Smoking zone				Non-smoking zone				Smoking zone			
Col.	AK	CH	DG	E	AK	CH	DG	E	AK	CH	DG	E	AK	CH	DG	E
#(Pass.)	6	7	0	0	3	3	0	0	16	17	6	1	7	8	2	1
Capacity	8	6	1	0	2	2	2	0	14	14	12	6	9	9	8	4

#### 4. Conclusions and Discussions

We have addressed two optimization problems faced by an airline company – the seat allocation and reallocation problems – to maximize customer satisfaction. We formulated each as an instance of the set packing problem (SPP) and devised an efficient method for generating promising candidates or columns in the SPP.

We showed through numerical experiments with real-life data that, under the criterion given by the airline company, our methods realize better customer satisfaction than the present allocation and reallocation, which are done manually by the airline’s operators. We also showed that our methods require only a few minutes of computation time on a workstation, and that they can therefore be put to practical use. Especially for the seat reallocation problem, our method can minimize the stress felt by the operators who have to perform seat reallocation under immense pressure from passengers.

We should notice that the airline company implemented their own systems, by revising our prototypes. One reason is that the output cannot be always accepted as it is for some practical reasons. Additional functions such as re-solving the problem after fixing a subset of the solution, or manually editing would be necessary. The other reason is that the airline company got aware of more constraints or preferences during the validation of our prototypes.

#### Acknowledgments

The authors wish to thank Mr. Ryuusuke Shinato and Mr. Masahiro Takishita, both of Japan Airlines, for supplying real data sets and offering helpful comments and suggestions.

#### References

- [1] R. Anbil, E. Gelman, B. Patty, and R. Tanga: Recent advances in crew-pairing optimization at American Airlines. *Interfaces*, **21** (1991) 62-74.
- [2] R. Anbil, R. Tanga, and E. L. Johnson: A global approach to crew-pairing optimization. *IBM Systems Journal*, **31** (1992) 71-78.
- [3] C. Barnhart, L. Hatay, and E. L. Johnson: Deadhead selection for the long-haul crew pairing problem. *Operations Research*, **43** (1995) 491-499.
- [4] P. P. Belobaba: Airline yield management: an overview of seat inventory control. *Transportation Science*, **21** (1987) 63-73.

- [5] P. P. Belobaba: Application of a probabilistic decision model to airline seat inventory control. *Operations Research*, **37** (1989) 183-197.
- [6] L. Bianco, M. Bielli, A. Mingozzi, S. Ricciardelli, and M. Spadoni: A heuristic procedure for the crew rostering problem. *European Journal of Operational Research*, **58** (1992) 272-283.
- [7] S. L. Brumelle, J. I. McGill, T. H. Oum, K. Sawaki, and M. W. Tretheway: Allocation of airline seats between stochastically dependent demands. *Transportation Science*, **24** (1990) 183-192.
- [8] S. L. Brumelle, and J. I. McGill: Airline seat allocation with multiple nested fare classes. *Operations Research*, **41** (1990) 127-137.
- [9] R. E. Curry: Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Science*, **24** (1990) 193-204.
- [10] I. Gershkoff: Optimizing flight crew schedules. *Interfaces*, **19** (1989) 29-43.
- [11] G. W. Graves, R. D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara: Flight crew scheduling. *Management Science*, **39** (1993) 736-745.
- [12] K. L. Hoffman, and M. Padberg: Solving airline crew scheduling problems by branch-and-cut. *Management Science*, **39** (1993) 657-682.
- [13] M. Kress, and B. Golany: Optimizing the assignment of aircrews to aircraft in an airlift operation. *European Journal of Operational Research*, **77** (1994) 475-485.
- [14] S. Lavoie, M. Minoux, and E. Odier: A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, **35** (1988) 45-58.
- [15] T. C. Lee, and M. Hersh: A model for dynamic airline seat inventory control with multiple seat bookings. *Transportation Science*, **27** (1993) 252-265.
- [16] J. Rubin: A technique for the solution of massive set-covering problems with application to airline crew scheduling. *Transportation Science*, **7** (1973) 34-48.
- [17] D. M. Ryan: The solution of massive generalized set partitioning problems in aircrew rostering. *Journal of Operations Research Society*, **43** (1992) 459-467.

Akira TAJIMA

IBM Research, Tokyo Research Laboratory

1623-14, Shimotsuruma, Yamato, Kanagawa 242-0001, Japan

E-mail: tajima@trl.ibm.co.jp