

SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-7501

TECHNICAL REPORT NO. 889

February 1990

THE SOLUTION OF MASSIVE GENERALIZED
SET PARTITIONING PROBLEMS
IN AIRCREW ROSTERING

by

D.M. Ryan*

*Department of Engineering Science, University of Auckland, Auckland, New Zealand.

This paper was prepared while the author was a Visiting Scientist in the Center for Applied Mathematics at Cornell University and was partially supported by NSF Grant #DMS-8706133.

The Solution of Massive Generalized Set Partitioning

Problems in Aircrew Rostering

D. M. Ryan

Department of Engineering Science
University of Auckland
Auckland, New Zealand

October 1989

ABSTRACT: The scheduling and rostering of personnel is a problem that occurs in many organizations. Aircrew scheduling has attracted considerable attention with many heuristic methods being proposed but in recent times set partitioning optimization methods have become more popular. The aircrew rostering problem is discussed and formulated as a generalized set partitioning model.

Because of the extremely large optimization models which are generated in practical situations, some special computational techniques have been developed to produce solutions efficiently. These techniques are used to solve problems arising from an airline application in which set partitioning models with more than 650 constraints and 200,000 binary variables are generated. The solutions are produced on a Motorola 68020 microprocessor in little more than three hours.

This paper was prepared while the author was a Visiting Scientist in the Centre for Applied Mathematics at Cornell University and was partially supported by NSF Grant #DMS-8706133.

§ 1 Introduction

The scheduling of air crews is a problem that has attracted considerable attention from both airlines and the mathematical community (see for example Lavoie et al (1988) and Cook (1989)). Air crews are amongst the most valuable of airline resources and efficient utilization of crews is obviously an important consideration in airline operations. From a mathematical point of view the problems of air crew scheduling give rise to a number of challenging problems in combinatorial optimization.

The air crew scheduling problem is usually partitioned into two distinct subproblems or stages. The first problem, called the Planning or Pairings Problem, involves the construction of sequences of flights or sectors which are variously described as duties (tours of duty) or trips (or pairings or rotations). In some situations these duties or trips can be daily periods of work but in other applications they can involve sequences of flights and periods of rest over many days. The important feature of a duty or trip is that it must be "feasible" with respect to all the rules, regulations and conditions of awards and agreements. While many of these conditions are common to most airlines the details can vary greatly among airlines. Most duties or trips originate and terminate at a crew base (hence the term "rotation"). The underlying objective in the planning problem is to produce a schedule of feasible duties or trips which cover all the timetabled flights in the most efficient manner possible. Efficiency is often measured in terms of the number of duties or trips but there are other obvious criteria such as total duty time and the number of sectors on which crews are positioned (or deadheaded) without working.

Many airlines continue to produce planning schedules using heuristic computer methods of which Rubin's algorithm (Rubin (1973)) has probably been the most widely used. However, a number of airlines have developed or are developing optimization-based techniques to solve the pairings problem. Marsten et al (1979) discuss the use of optimization techniques in crew planning for Flying Tiger, and Marsten and Shepardson (1981) describe the application of similar techniques to the solution of aircrew scheduling problems for three other airlines. Crainic and Rousseau (1987) and Lavoie et al (1988) discuss the use of column generation techniques to solve set partitioning problems arising in the airline industry and Gershkoff (1987) describes American Airlines' use of optimization techniques in the solution of their aircrew pairing problems. During the past three years a set partitioning based optimization system has been used by Air New Zealand (Ryan and Garner (1985)) to prepare tours of duty for the domestic services of the airline. This system now plays an important role in the crew planning functions of Air New Zealand.

The second problem associated with air crew scheduling is referred to as Rostering. The Rostering phase involves the allocation of the planned duties or trips from the first stage to individual crew members to form a "line of work" (LoW) over the rostering period. In this paper we will consider the rostering period to be four weeks. Each trip originates and terminates at a crew base and consists of a sequence of flights (sectors) and rest periods. Trips range in length from one-day two-sector trips to fifteen or sixteen-day multisector trips. Typically a trip will be followed by a stand-down rest period of a length related to the work content of the trip. The internal work content of the trip is however of little importance in the rostering problem. Trip destinations such as Europe, the United States, Japan and Australia do influence the attractiveness of a LoW but the details of component sectors and airports visited in the trips are irrelevant. The full Rostering problem can usually be broken into smaller independent subproblems corresponding to groups of crew members of the same rank.

The approach to the "solution" of the Rostering Problem used by most North American airlines is referred to as the Bidline System. The pairings produced as the solution of the pairings problem are published and the crews then bid for their preferred line or alternative lines of work. Bids are usually accepted in order of decreasing seniority in the crew rank. The bidline system can result in considerable inequity within a crew rank with many junior crews seldom being allocated their preferred line of work. Most airlines outside North America (especially the smaller airlines) attempt to produce rosters which satisfy some measure of equity to ensure that all crew members of the same rank are allocated lines of work with similar work content. While the rosters produced are certainly more equitable, the equity rostering problem is invariably far more difficult to solve than the planning problem because of its increased combinatorial complexity. For this reason most airlines requiring equitable rosters adopt sequential heuristic methods to allocate duties or trips to individual crews. Such methods invariably lead to some inequity at least for a subset of the crew rank. In this paper we outline an extension of the mathematical optimization techniques developed for the planning problem to solve the equity rostering problem.

§ 2 A Mathematical Model for Rostering

It is clear that each individual crew member must be allocated exactly one LoW and each trip must be covered by sufficient crew members. For each crew member in a given rank we can generate a set of many LoWs from which exactly one must be chosen. Each LoW in the set can be costed according to some measure of its desirability. The generation process and cost measures will be discussed in § 3.

The rostersing problem can be modelled mathematically using a generalized version of the set partitioning model. Assuming there are p crew members and t trips, the model is naturally partitioned into a set of p crew constraints, one for each crew member in the rank, and a set of t trip constraints corresponding to each trip which must be covered. The variables of the problem can also be partitioned to correspond to the feasible LoWs for each individual crew member. The rostersing set partitioning problem can be written

$$\min z = \mathbf{c}^T \mathbf{x}, \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \quad x_i = 0 \text{ or } 1$$

where A is a 0-1 matrix partitioned as

$$A = \begin{bmatrix} C_1 & C_2 & C_3 & \dots & C_p \\ L_1 & L_2 & L_3 & \dots & L_p \end{bmatrix}$$

and $C_i = \mathbf{e}_i \mathbf{e}^T$ is a $(p \times n_i)$ matrix with \mathbf{e}_i the i^{th} unit vector and $\mathbf{e}^T = (1, 1, \dots, 1)$. The n_i LoWs for crew member i form the columns of the $(t \times n_i)$ matrix L_i with elements l_{jk} defined as $l_{jk} = 1$ if the k^{th} LoW for crew member i covers the j^{th} trip and $l_{jk} = 0$ otherwise. The matrix A has total dimensions of $m \times \sum_{i=1}^p n_i$ where $m = p + t$. The right-hand-side vector \mathbf{b} is given by $b_i = 1, i =$

$1, \dots, p$ and $b_{p+i} = r_i, i = 1, \dots, t$ where r_i is the number of crews required to cover the i^{th} trip. In some circumstances certain trips of shorter duration can be overcovered (ie more than r_i crews allocated) thus implying the inclusion of corresponding surplus variables in the model to create equality constraints. After an optimal roster has been constructed, the overcovered trip can be removed from the LoWs of any surplus crews in the optimal solution and replaced with extra days off. Surplus crews can be selected from amongst those crew members allocated the overcovered trip by taking into account the past workload and also the nature of the optimal LoWs for those crew members. It is clear that overcover should not be permitted for trips of longer duration since

the replacement of such a trip by extra days off would create an inequitable LoW with too many days off.

The cost vector \mathbf{c} should be chosen to reflect the relative "cost" of each LoW. Since most airlines are pleased to construct one feasible solution to the rostering problem there is no obvious or traditional measure which is used to discriminate among feasible solutions. We discuss one possible measure in § 3.

The rostering model has a special structure which deviates from pure set partitioning in that the right-hand-side vector is not unit valued and some constraints need not be equalities. The crew constraints of the A matrix also exhibit a generalized upper bounded structure which is not commonly found in set partitioning. It can be shown (see Ryan and Falkner (1988)) that constraints of this type have a particularly important and beneficial effect on the occurrence of integer basic feasible solutions in the LP relaxation derived from the set partitioning model.

§ 3 Generation and Costing of LoWs

The generation of LoWs is performed by a straightforward enumeration process for each crew member considered in turn. For a particular crew member the process begins with the construction of a skeleton LoW which takes into account the carryover activity from the previous roster period and includes preassigned activities such as drills and training, requested trips and requested days off and periods of call and leave. All possible legal and desirable sequences of trips are then generated by adding trips to the skeleton until no further additions are possible. The last added activity is then deleted and further additions are attempted. This enumeration process continues until all possible LoWs have been considered. There are more than thirty rules and conditions which a legal and desirable LoW must satisfy. Some of these are mandatory but others reflect the preferences of the airline or the individual crew member. One particularly important restriction involves a requirement that every LoW must have at least ten days off at home base in each twenty-eight day roster period.

For a practical crew rostering problem, the generalized set partitioning model proposed in § 2 has dimensions in the order of between 200 and 700 constraints and tens or perhaps even hundreds of millions of variables since some crew members can have many thousands of alternative LoWs. Fortunately it is possible to reduce the number of variables first by limited subsequence filtering

techniques similar to those described by Ryan and Falkner (1988) and secondly by a number of implicit restrictions on the types of LoWs which will be considered feasible.

The limited subsequence filtering techniques are based on the premise that each crew member should be allocated his or her next trip as soon as possible after completing his or her previous trip and the mandatory rest. On completion of each trip, a limited number of subsequent trips are chosen as alternative following trips. The choice of alternatives is not critical. It can be based on least possible idle time between trips but can also include consideration of aspects related to the costing of LoWs. Typically as few as one or two and as many as ten to twenty subsequences are selected. Besides reducing the number of variables, limited subsequence filtering also improves the natural integer properties of the set partitioning problem and reduces the fractioning potential of the optimal solutions since the resulting constraint matrix has a more "balanced" form (see Ryan and Falkner (1988)). The implicit restrictions reflect the practical application and could include for example an upper limit on the number of days off in any feasible LoW and the elimination of certain undesirable sequences or combinations of trips within a LoW for some crew members. The reduced problems have between 20,000 and 200,000 variables with each crew member having on average between 400 and 500 alternative LoWs. Variation can range from a single LoW for some crew members up to 2000 for others.

The definition of an objective function is not obvious. Because airlines usually operate the first roster they can construct, there is no traditional objective measure with which to compare alternative rosters. In fact feasibility plays a much more important role than optimality when rosters are being constructed. However, in order to construct equitable rosters we need to develop an objective which reflects the interests of both management and the crews. From an airline's point of view it is obviously important to minimize the number of crews required to cover all trips and also to preserve a certain tolerance of disruption in the solution. For example, reasonable flight delays should not make it impossible for a crew member to complete the allocated LoW. From the crew member's point of view one particularly important preference is that trips of the same type (often related to destination) should be spaced apart by a sufficient number of days. Given the number of crew members required for each trip of a certain type, the frequency with which such trips occur in the roster period and the number of crew members in the rank, the ideal or expected separation between trips of that type can easily be calculated. When constructing a LoW, if a trip is allocated before the ideal separation interval has elapsed since the last allocation of a trip of that type, the number of days short of the desired separation interval is referred to as the number of "history days violated". One measure of quality of a LoW, at least from the crew perspective, is given by the sum of history days violated for each trip in the LoW. For a given rank then, the total

history violated can be thought of as a possible objective measure. The objective value can also reflect some preference for LoWs which have fewer days off (ie closer to the average days off across the rank) and fewer trips (ie closer to the average number of trips required to be performed by each crew member). Both of these average values can be easily calculated before the generation process begins. Such preferences are consistent with the notion that the most desirable roster is one in which each crew member performs a similar amount of work.

It is also possible to utilize the model to minimize first the number of crews and second the equitability rostering objective. This can be accomplished by including a null LoW (a LoW which represents the crew member as performing no trips in the roster period) priced at zero cost and adding a "small" constant penalty to each actual LoW generated for that crew member. The SPP will then naturally tend to select null LoWs in preference to actual LoWs provided that all the trips can be covered without the services of the inactive crew members. This device is particularly useful in identifying the need for promotions to increase the rank size. A small number of fictitious crew members can be added to the rank but will only be used if the problem has no feasible solution without the use of extra crew members.

In generating LoWs for each crew member it is also possible to take into account an individual's preferences for LoWs with certain work content. For example some crew members may prefer LoWs made up of shorter duration trips while other crew members may prefer longer trips. Such preferences can influence the filters and tests for feasibility during the generation process and they can also influence the calculation of the objective coefficient for each LoW. Preferences are not allowed to cause infeasibility in the rostering model but they can, at least in the longer term, produce more attractive and satisfying solutions.

§ 4 Solution Method

The solution of SPPs has been surveyed by Balas and Padberg (1979) (see also Bausch (1982)) and recent results in the development of set partitioning and covering techniques suggest that problems with several thousand variables can now be solved (Bodin et al (1983)). Marsten and Shepardson (1981) reported the solution of larger set partitioning problems with up to 30,000 variables but in some cases solutions were not found. Since 1981 a number of authors have described Marsten's SETPAR code (Marsten (1974)) as the most efficient algorithm for set partitioning. In a recent paper Fisher and Kedia (1988) have reported the solution of randomly

generated set partitioning problems with up to 100 constraints and 10,000 variables and although direct comparisons with SETPAR were not performed, Fisher and Kedia suggest that their algorithm, based on Lagrangian Relaxation, "is at least an order of magnitude faster than Marsten's". However the limited and approximate nature of the comparisons do not provide a clearly superior approach especially when the problem dimensions are increased significantly. Similar claims, based on comparisons with SETPAR, have been made by Chan and Yano (1988) for their method which uses a dual multiplier adjustment algorithm to produce bounds. Again problems with fewer than 10,000 variables were considered in these comparisons.

Because of our previous successful experience in solving set partitioning problems with up to 30,000 variables in the Planning phase (but with foolhardy optimism) we have attempted to solve the massive generalized SPPs generated in the rostering application by a conventional integer programming approach. The LP relaxation is solved first using a primal simplex algorithm and then a branch and bound procedure based on a constraint branch (Ryan and Foster (1981)) is applied to produce integer solutions. Brown et al (1987) also describe the successful application of a similar conventional IP approach in the shipping industry although the LP solution method and the branching mechanism were different.

§ 4.1 Solution of the LP Relaxation.

The solution of the LP relaxation has always proved to be a computational bottleneck in solving SPPs. Two main difficulties arise. The first is due to the very large number of variables (LoWs) in the LP formulation. Even with the use of generation filters as discussed in § 3, the number of variables can often exceed 200,000. However it is possible to partition the variables into classes which rank in decreasing order of attractiveness. The ranking need not be based directly on the magnitudes of the objective coefficients although they provide one obvious means of ranking. Ranking could also be based on numbers of trips or days off which vary significantly from the expected average number of trips or days off. During the primal convergence process each class of variables can be treated in a different way. For example, some classes of variables may not be considered in the entering variable pricing until late in the primal convergence, while variables from other classes may be considered as entering variables only if the LP remains primal infeasible. With careful use of such techniques and the use of partial pricing, it is possible to significantly reduce the problems associated with very large numbers of variables in the primal LP convergence. The solution of the relaxed LP has also been improved by the use of a version of the elastic LP strategies of Brown and Graves (1975). During the convergence of the RSM, primal feasibility is not forced quickly through a conventional phase-1 procedure but is achieved much more slowly by gradually increasing the costs on the (artificial) slack and surplus variables to drive them from the

basis. The intention here is to choose costs for slacks and surplus variables which just bound the true optimal dual variable values.

The second difficulty in solving the LP relaxation of SPPs is due to the presence of gross degeneracy at near-integer basic feasible solutions. Because of the use of limited subsequence matrix reduction techniques in the generation of the constraint matrix, many such near-integer basic feasible solutions can be visited during the LP convergence and solutions with as many as 80% of the basic variables at zero value are common. At grossly degenerate primal bases, the RSM tends to stall sometimes for many thousands of degenerate pivots even when conventional techniques such as maximizing the pivot element are used to determine the leaving variable. This degeneracy phenomenon in the SPP has been discussed by a number of authors (see for example Albers (1980), Falkner (1988), Marsten and Shepardson (1981)) and to avoid the difficulties, Marsten (1974) solved the relaxed LP using a dual algorithm in his SETPAR code. For problems with very large numbers of variables this is not an attractive option. Other authors have avoided the problems of degeneracy by developing alternative bounding strategies based on Lagrangian relaxation or dual variable adjustments.

The problems of degeneracy have been overcome in the solution of the rostering model by the use of Wolfe's method (Wolfe (1963), Ryan and Osborne (1988)) and by a carefully chosen right-hand-side perturbation scheme. Wolfe's method provides a guaranteed termination of the stall and is used only when a sequence of degenerate pivots is observed. The perturbation scheme involving $b_i = 1 + \epsilon$, $i = 1, \dots, p$ creates a tension between the crew constraints of the model and the trip constraints which still have integer right-hand-sides. For a small value of ϵ such as 10^{-7} , the values of basic variables are perturbed sufficiently to avoid degeneracy and in practice few truly degenerate pivots are observed during the convergence of the RSM. Typically, as few as one or two hundred degenerate pivots occur in RSM convergences of more than 10,000 iterations and Wolfe's method is seldom required.

§ 4.2 An Efficient Branching Strategy

It is well known that the conventional variable branch is particularly ineffective in the resolution of fractional solutions which arise at the optimal solution of the relaxed SPP linear programme. The 1-branch (forcing the branched variable to the value one) imposes significant structure on the solution of the problem and often causes the minimized objective function to increase. The 0-branch however has little effect when there are many thousands of alternative variables which can

enter the basis producing a new fractional feasible solution without increasing the objective function value. The branch and bound tree then develops in a very unbalanced way so that the bounding mechanism is seldom effective especially on the zero branch. As an alternative, the constraint branch (Ryan and Foster (1981)) has proven to be particularly effective in resolving fractional solutions in the set partitioning problem. This type of branch was also discussed in the context of set covering by Etcheberry (1977). At an optimal fractional solution of any node of the branch and bound tree, a pair of constraints, say s and t , can be identified such that

$$0 < \sum_{j \in J(s,t)} x_j < 1$$

where $J(s,t) = \{ j \mid a_{sj} = 1 \text{ and } a_{tj} = 1 \}$. It is easy to show that in any fractional solution at least one such pair of constraints can be identified. In this application the pair can be chosen to involve a crew member s and a trip t only partially performed by that crew member. The branch can then be imposed by requiring on the 1-branch

$$\sum_{j \in J(s,t)} x_j = 1$$

(ie crew member s must perform trip t) and on the 0-branch

$$\sum_{j \in J(s,t)} x_j = 0$$

(ie crew member s must not perform trip t). The zero branch is easy to implement by forcing all variables in $J(s,t)$ to take the value zero. The Dual Simplex Method can be used to remove the resulting infeasibility which results from imposing the branch. In the SPP, the 1-branch can also be easily implemented by requiring

$$\sum_{j \in J'(s,t)} x_j = 0$$

where $J'(s,t) = \{ j \mid a_{sj} = 1 \text{ or } a_{tj} = 1, a_{sj} \neq a_{tj} \}$ is the complementary set to $J(s,t)$ and, as for the 0-branch, the resulting infeasibility is removed using the DSM. It should be noted that enforcing the 1-branch for crew member s and trip t involves forcing to zero those variables for crew member s which **do not** cover trip t and provided trip t requires a single additional crew member, also all variables for other crew members which **do** cover trip t .

The constraint branch, in contrast to the variable branch, can involve the simultaneous elimination of many hundreds and sometimes thousands of variables on each side of the branch. Because

many variables are eliminated on each side of the branch, the branch and bound tree develops in a much more balanced fashion.

In practice if s and t are chosen so that $\sum_{j \in J(s,t)} x_j$ is maximized (ie as close to 1 as possible)

then the branch and bound can be implemented with a depth-first 1-branch and the 0-branch can safely be left unfathomed since the 1-branch reflects the preference of crew member s for trip t already implied by the LP optimal but fractional solution. This choice of branch has the added benefit that few DSM iterations are required to remove the infeasibilities due to other crew members performing a small fractional part of trip t . Integer solutions are invariably produced by a sequence of 1-branches and the algorithm has been implemented to terminate as soon as an integer solution is found. On a few occasions towards the bottom of a sequence of 1-branches and especially when

$$\sum_{j \in J(s,t)} x_j = 0.5$$

(ie the LP is indicating no preference for the 1-branch or the 0-branch) the problem can become infeasible on the 1-branch. In such situations the 0-branch from that node of the tree is evaluated and if necessary the 0-branches of nodes even further back up the tree are evaluated to find an integer solution.

§ 5 Some Results and Computational Experience

The computations reported in this paper have been performed using the ZIP package (Ryan (1980)). ZIP, a suite of Fortran 77 routines, incorporates sparse matrix techniques (Reid (1976,1982)) to handle the sparse basis factorizations used in the primal and dual simplex algorithms. All the results have been generated on an IBM PC/AT with a Definicon DSI-780/4 accelerator board which is based on a Motorola 68020 microprocessor.

We report here results obtained from the rostering of five ranks of crews (identified by labels CH, PU, AP, SE and JU) during three consecutive roster periods. The roster periods, each requiring a separate optimization, will be identified by numbers 1, 2 and 3. The CH problems, identified as CH1, CH2 and CH3 in Table 1, involve approximately 55 crew members and 120 trips which are each to be covered once during each of three separate 28 day roster periods.

For ranks AP, SE and JU, the required trip coverage varies between 1 and 8 depending on the crew rank and aircraft type (Boeing 747 or 767) of the trip. Within each rank small natural variations occur between periods in the number of crews and trips. The results in Table 1 give the sizes of the set partitioning problems generated and show summary information about the RSM convergence including the number of iterations, the total 68020 processor time in minutes, the optimal LP objective (z-value) and the number of variables with integer (ie unit) and fractional values at the relaxed LP optimal solution. Information about the branch and bound computation is also included. Table 1 also gives the total minimized history violation in days for each problem. It should be noted that these history violation figures exclude history violations which result from a crew member requesting a trip which violates that crew member's own history. Given that requested trips should be allocated except where no feasible solutions can be produced, such requested history violations cannot be eliminated by the rostering objective.

The results of Table 1 illustrate a number of interesting points about the solution of the extremely large and structured set partitioning problems arising from the rostering model.

1. Generalized set partitioning models with more than 650 constraints and nearly 200,000 variables can be solved on a microcomputer provided the models have sparse column form and special structure. These problems are significantly larger than SPPs previously reported as being solved. Problems with as many as 250,000 variables have been solved to optimality on the microcomputer and problems with more than 300,000 variables have been solved on a mainframe machine.
2. Although there appears to be a relatively large number of revised simplex iterations required at least in the JU rank, the basis factorization updates and the entering variable pricing are fast because of the sparse 0-1 constraint matrix structure.
3. The limited subsequence matrix reductions can result in the generation of a suboptimal solution. However provided the subsequences are chosen with care, the LoWs which are excluded are less attractive in that they contain longer periods of idle time. The loss of optimality can be estimated by introducing further excluded LoWs and permitting the RSM to reconverge from the previous optimal basis. It has been found that the objective is seldom improved even by small amounts but when the objective is decreased the optimal solutions are invariably more fractional. It should also be recalled that the rostering objective reflects an ill-defined measure of equitability. The results show that the generated integer solutions have a

| Problem | Problem Size | | | RSM convergence | | | Branch & Bound | | | History violation | | | |
|---------|--------------|-----|-----|-----------------|-------|--------|----------------|----------|-------|-------------------|---------|-------|---------|
| | p | t | m | $\sum_i n_i$ | itns | mins | z-value | int/frac | nodes | mins | z-value | total | average |
| CH1 | 56 | 116 | 172 | 21826 | 2049 | 4.58 | 566.5 | 23/33 | 27 | 2.36 | 573 | 381 | 6.8 |
| CH2 | 56 | 118 | 174 | 23731 | 1987 | 4.73 | 617 | 42/14 | 12 | 1.28 | 622 | 415 | 7.4 |
| CH3 | 55 | 124 | 179 | 27772 | 2179 | 5.35 | 515 | 55/0 | 0 | 0 | 515 | 299 | 5.4 |
| PU1 | 59 | 128 | 187 | 29115 | 3405 | 9.19 | 364 | 22/37 | 32 | 8.66 | 364 | 118 | 2.0 |
| PU2 | 59 | 132 | 191 | 23192 | 1693 | 4.51 | 359 | 39/20 | 7 | 0.62 | 364 | 90 | 1.5 |
| PU3 | 61 | 144 | 205 | 53727 | 2445 | 9.28 | 570.38 | 23/38 | 25 | 8.27 | 589 | 313 | 5.1 |
| AP1 | 166 | 148 | 314 | 63893 | 4683 | 20.10 | 1163 | 149/17 | 11 | 1.82 | 1163 | 617 | 3.7 |
| AP2 | 165 | 147 | 312 | 64421 | 8486 | 38.07 | 2352.29 | 99/66 | 38 | 11.16 | 2479 | 1596 | 9.6 |
| AP3 | 171 | 157 | 328 | 63528 | 3720 | 16.81 | 1209 | 107/64 | 48 | 14.65 | 1231 | 306 | 1.8 |
| SE1 | 169 | 147 | 316 | 61363 | 5647 | 22.51 | 1780.67 | 113/56 | 25 | 12.8 | 1866 | 966 | 5.7 |
| SE2 | 170 | 147 | 317 | 63618 | 5196 | 20.83 | 1516 | 120/50 | 24 | 6.49 | 1516 | 700 | 4.1 |
| SE3 | 169 | 156 | 325 | 61282 | 3668 | 17.23 | 1156 | 103/66 | 60 | 18.37 | 1156 | 87 | 0.5 |
| JU1 | 473 | 184 | 657 | 128104 | 14033 | 99.45 | 3330.5 | 395/78 | 35 | 26.79 | 3332 | 856 | 1.8 |
| JU2 | 473 | 183 | 656 | 186444 | 13665 | 103.10 | 2791.19 | 379/94 | 36 | 34.36 | 2792 | 353 | 0.7 |
| JU3 | 461 | 200 | 661 | 145827 | 13301 | 105.30 | 2834.44 | 368/93 | 51 | 104.88 | 2867 | 85 | 0.2 |

Table 1 Results for five ranks over three roster periods

very small average history violation per crew member. In fact the total history violation is much less than the history violation generated by previously used heuristic methods. This is especially true for the larger ranks where the optimization is able to produce excellent solutions by taking advantage of the extra flexibility in a way that is not possible with a heuristic method.

4. The relaxed linear programme solutions exhibit strong integer properties. This can be attributed to the limited subsequence filtering techniques and also to the action of the crew constraints which tend to be of a "perfect" type (see Ryan and Falkner (1988)). Such dominant constraints effectively prevent fractions from occurring within the LoWs for a particular crew member. Any fractions which do occur in an optimal solution of any node of the branch and bound tree must involve more than one crew member.
5. The constraint branch based on a crew member and a trip is most effective in resolving fractional solutions. Relatively few nodes of the branch and bound tree are evaluated before an integer solution is found and the problems towards the bottom of the sequence of 1-branches usually involve few variables since most variables have already been branched to zero. The resulting DSM convergences are usually simple and feasibility is quickly restored.
6. Total CPU times to produce integer solutions on a microcomputer range from six or seven minutes for the CH rank to two to three hours for the JU rank.

One particularly important benefit of the optimization approach to the solution of the rostering problem, at least in comparison with heuristic methods, is that infeasibility can be identified with certainty. Often in situations where the crew rank is close to the minimum rank size and there are days during the roster period when insufficient crews are available to perform all the trips, the optimization will identify such trips and days. By carefully setting the costs of trip slack variables, it is possible to produce infeasible solutions which leave uncovered the fewest trips of shortest duration. This information can then be used to "create" feasibility by modifying the skeleton activities such as drills and requested trips and days off for some crew members to free further crew resources on the tight days of the roster period. In contrast, when using heuristic methods, one is never certain whether the infeasibility is due to the inadequate nature of the heuristic or whether it truly is an infeasible problem. It was commonly thought that the rostering problems were in fact always close to infeasibility but it is now clear that there exist many alternative feasible solutions which can be constructed and explored using the optimization techniques.

§ 6 Acknowledgements

I wish to acknowledge the many helpful discussions I have had with Dr Julie Falkner of the Dept of Engineering Science, University of Auckland. I also wish to thank Jeff Meyer, Brett Wilkie, Dale Bradford and other staff of Air New Zealand Ltd for their support and enthusiasm in providing data and developing a production interface to the crew and company databases.

§ 5 References

- Albers, S. (1980). Implicit enumeration algorithms for the set-partitioning problem. OR Spektrum 2, pp 23-32.
- Balas, E. and Padberg, M.W. (1979). Set partitioning: a survey. In: Christofides, N. ed. Combinatorial Optimization, Wiley.
- Bausch, D.O. (1982). Computational advances in the solution of large-scale set covering and set partitioning problems. Masters Thesis, Naval Postgraduate School, Monterey, California.
- Bodin, L.D., Golden, B., Assad, A. and Ball, M. (1983). Routing and scheduling of vehicles and crews: the state of the art. Computers and Operations Research 10, pp 63-211.
- Brown, G.G. and Graves, G.W. (1975). Elastic programming: a new approach to large scale mixed integer optimization. Paper presented at ORSA/TIMS meeting, Las Vegas.
- Brown, G.G., Graves, G.W. and Ronen, D. (1987). Scheduling ocean transportation of crude oil. Management Science 33, pp 335-346.
- Chan, T.C. and Yano, C.A. (1988). A multiplier adjustment approach for the set partitioning problem. Unpublished manuscript.
- Cook, T.M. (1989). Ed Special Issue. Airline Operations Research. In: Interfaces 19(4), pp 1-74.
- Crainic, T.G. and Rousseau, J-M. (1987). The column generation principle and the airline crew scheduling problem. INFOR 25, pp 136-151.
- Etcheberry, J. (1977). The set-covering problem: a new implicit enumeration algorithm. Operations Research 25, pp 760-772.
- Falkner, J.C. (1988). Bus crew scheduling and the set partitioning model. PhD Thesis, University of Auckland, New Zealand.
- Fisher, M.L. and Kedia, P.K. (1988). Optimal solution of set covering/partitioning problems using dual heuristics. Working Paper: 88-11, Northeastern University.
- Gershkoff, I (1987) American's system for building crew pairings. Airline Executive, September, pp 20-22.
- Lavoie, S., Minoux, M. and Odier, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. European Journal of Operational Research 35, pp 45-58.
- Marsten, R.E. (1974). An algorithm for large set partitioning problems. Management Science 20, pp 774-787.
- Marsten, R.E., Muller, M.R. and Killion, C.L. (1979). Crew planning at Flying Tiger: a successful application of integer programming. Management Science 25, pp 1175-1183.
- Marsten, R.E. and Shepardson, F. (1981). Exact solution of crew scheduling problems using the set partitioning model: recent successful applications. Networks 11, pp 165-177.

- Reid, J.K. (1976). Fortran subroutines for handling sparse linear programming bases. A.E.R.E. Report R-8269, A.E.R.E. Harwell, Oxfordshire.
- Reid, J.K. (1982). A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases. Mathematical Programming **24**, pp 55-69.
- Rubin, J. (1973). A technique for the solution of massive set covering problems with applications to airline crew scheduling. Transportation Science **7**, pp 34-48.
- Ryan, D.M. (1980). ZIP - a zero-one integer programming package for scheduling. Report C.S.S. 85, A.E.R.E. Harwell, Oxfordshire.
- Ryan, D.M. and Falkner, J.C. (1988). On the integer properties of scheduling set partitioning models. European Journal of Operational Research **35**, pp 442-456.
- Ryan, D.M. and Foster, B.A. (1981). An integer programming approach to scheduling. In: Wren, A. ed. Computer Scheduling of Public Transport North-Holland, Amsterdam, pp 269-280.
- Ryan D.M. and Garner, K.M. (1985). The solution of air-crew scheduling problems for Air New Zealand. Proceedings of the 21st Annual Conference of O.R.S.N.Z., Wellington, pp 42-48.
- Ryan, D.M. and Osborne, M.R. (1988). On the solution of highly degenerate linear programmes. Mathematical Programming **41**, pp 385-392.
- Wolfe, P. (1963). A technique for resolving degeneracy in linear programming. SIAM Journal **11**, pp 205-211.

