INTERFACES
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0092-2102 | EISSN 1526-551X | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Mathematical Programming Guides Air-Ambulance Routing at Ornge

Timothy A. Carnes
Sloan School of Management, MIT, Cambridge, MA 02139, tcarnes@mit.edu

Shane G. Henderson, David B. Shmoys
School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853
sgh9@cornell.edu david.shmoys@cornell.edu

Mahvareh Ahghari
Ornge, Mississauga, ON L4W 5H8, Canada, mahghari@ornge.ca

Russell Macdonald
Ornge, Mississauga, ON L4W 5H8, Canada
Faculty of Medicine, University of Toronto, Toronto, ON, Canada
rmacdonald@ornge.ca

Ornge provides air-ambulance services to the Province of Ontario. A major portion of their service involves pre-scheduled transports from one medical facility to another. These transports almost exclusively require fixed-wing aircraft due to the distances involved and cost considerations. The requests are received in advance, scheduled overnight, and typically executed the following day. We describe our work in developing a planning tool which determines an assignment of requests to aircraft that minimizes cost, subject to a range of complicating constraints. The tool is in use by flight planners every day at Ornge, and has resulted in substantial savings relative to the previous manual approach to devising schedules. We describe the problem, our formulation, its implementation, and the impact on operations at Ornge.

Key words: set partitioning, transport medicine, dial-a-ride
History:

Introduction

The province of Ontario, Canada has approximately thirteen million residents spread over approximately one million square kilometers. Ontario ensures a high level of medical care for all residents partly through an advanced medical transport capability. The not-for-profit company Ornge provides these transport services primarily through air-ambulance services, although it also has several land ambulances. Ornge transports approximately 19,000 patients every year. These transports can be broken down into "scene calls" and "inter-facility transports." Scene calls are those calls that immediately spring to mind when one thinks of air-ambulance service: paramedics respond to the location of an accident by

helicopter, sometimes with assistance from land ambulances when the sending or receiving facility doesn't have a heliport. Inter-facility transfers can be emergent (42%), urgent (21%) or non-urgent (37%). Emergent and urgent interfacility transports are patient transfers between medical facilities that require an immediate response. In this study we focus on non-urgent inter-facility transports that can be scheduled in advance.

Non-urgent inter-facility transports are carried out by a subset of aircraft staffed and equipped for non-urgent requests. These aircraft are occasionally seconded for emergent transports in times of overwhelming demand for patients with emergent, time-sensitive conditions, causing disruption in the non-urgent patient transfer schedules. Thus there is a limited amount of sharing of aircraft between the categories. We ignore this sharing in our current work, but future work may take the sharing into account; see the "Impact and Next Steps" section for more on this point.

Non-urgent inter-facility transports (henceforth referred to as requests) typically number 10-20 requests per day, although 30 requests is not unheard of, and are almost exclusively handled by fixed-wing aircraft that are stationed around the province.

A key problem at the core of Ornge's mandate then, is to determine how to schedule and route available aircraft to handle these requests at minimal cost. The problem is combinatorially complex, because aircraft can handle up to approximately four requests within a duty shift, the requests can be handled in any order subject to certain time restrictions, some aircraft can carry up to two patients while others can carry only one, some patients cannot be transported with another patient owing to infectiousness or other issues, and there are other complexities as well. Previously, experienced flight planners at Ornge would determine the aircraft assignments and routes manually, based on experience and personal practice.

Beginning in 2009, and in concert with several Master-of-Engineering project students and Ph.D. students, we have developed an optimization-based tool that determines the optimal assignment of requests to aircraft, and the optimal routes to fly, taking into account a number of complicating side constraints. The tool employs an intuitive Excel interface, along with a C++ implementation that assembles a set-partitioning formulation of the problem, invokes an integer-programming solver, and returns the optimization results to the Excel tool. Flight planners now use this tool to develop a plan for the next day's operations. They set up the problem in the Excel tool, obtain the solution, study the

solution for its practicality and adherence to policies that are not easily encoded in a mathematical formulation, adjust certain parameters that can help account for these, and re-solve. Often the first solution obtained is the one implemented, but in general a small number of these iterations are required.

The tool was tested and assessed for validity using retrospective data from randomly selected dates from July 2010 to February 2011 (MacDonald et al. 2011), with results guiding a live implementation. The study showed that the tool would yield an estimated 12% decrease in flying hours, 13% in distance flown, and 16% in cost. The application was then implemented in real-time on a test basis in May 2011, with a trial implementation on randomly selected days in June and August 2011. During the first 8 weeks of full implementation of the tool, the application resulted in only a 3% decrease in cost (MacDonald et al. 2011, 2012). The difference between estimated and actual was determined to be deviations between optimized routings and subsequent changes made by flight planners for aircraft diversion to patients with emergency conditions without using the application. This gap between fully optimized and actual use is being addressed by a "schedule repair" option incorporated into the new version of the application. Further details on the impact of the tool on Ornge's operations can be found in the "Impact and Next Steps" section below.

The underlying mathematical problem faced at Ornge is known in the Operations Research literature as a static "dial-a-ride" problem (Cordeau and Laporte 2007) with nonhomogeneous vehicles (aircraft) and multiple depots (aircraft bases). The problem is static in that requests are not scheduled as they are received, but rather collected and scheduled simultaneously. For comprehensive surveys on this problem and other vehicle routing problems with pickups and deliveries, see the surveys Cordeau and Laporte (2007), Cordeau et al. (2008) and Parragh et al. (2008). For an example of a challenging *dynamic* instance of a dial-a-ride problem in healthcare, see Beaudry et al. (2010).

There are a number of different approaches available to solve static dial-a-ride problems. In addition to a variety of heuristics that can scale to very large instances, Cordeau and Laporte (2007) describe two exact methods based on integer programming formulations with decision variables that determine the "arcs" traversed by vehicles, and therefore the routes that vehicles take. The formulations differ in that one determines vehicle-specific arcs while the other applies to homogenous vehicle fleets. Neither formulation can be applied in our setting because of the difficulty in effectively capturing side constraints.

The ability to capture complex constraints is viewed by Ornge as vital, and it ensures that the solution produced requires little or no further modification by the end-user. To capture side constraints, we adopt a set-partitioning formulation with enumerative pricing of feasible assignments. A similar formulation is given in Parragh et al. (2012). In contrast to that work we are able to solve our instances to optimality, perhaps mostly due to the smaller scale of our problems, but also partly by exploiting the combinatorial structure of the problem to efficiently carry out the enumeration of pricing and feasibility checking of routes. The set-partitioning formulation was also considered as one of two formulations for an air-taxi problem in Espinoza et al. (2008a,b). Even the linear-programming relaxation of the set partitioning problem could not be solved due to the size of the instances. Instead, the underlying formulation there is a multicommodity network flow with side constraints, with special heuristic techniques for scaling to large problem instances.

Certain side constraints can be captured using dynamic column generation in time-space networks as in Engineer et al. (2011), but fortunately we can avoid the associated algorithmic complexity because our problem is small enough that we can enumerate and price all feasible columns in a manner that allows real-time use of the application by flight planners. Therefore, unlike most (but not all) of the studies in the literature, we solve our instances to optimality.

The work herein is, to the best of our knowledge, a first in addressing such a problem within the air medical transport industry. Commercial airlines employ related strategies to develop flight schedules, routes, aircraft assignments and crew schedules. Our application differs because the schedule is different each day.

The remainder of this paper is organized as follows. The "Problem" section defines the problem in more detail, explaining the structure of requests, the costs associated with routes, and the primary complicating side constraints. The "Impact and Next Steps" section describes the results of an experiment to validate the tool and measure its performance relative to previous practice. It also covers the subsequent impact of this work at Ornge, and outlines ongoing work. The appendix describes the set-partitioning problem formulation, and explains the recursion used to compute column costs. It also explains how we handle complicating side constraints.

Problem

Requests

Ornge receives a number of requests on a given day which need to be scheduled the following day. Each request consists of the following information.

Origin Airport at which the transport originates.

Destination Airport at which the transport terminates.

Pickup After The earliest time a pickup can be completed.

Dropoff Before The latest time a dropoff can be completed.

Level of Care This represents the needs of a patient, and can be "primary," "advanced" or "critical" care. These levels represent increasingly constraining requirements of personnel, scope of practice, and equipment needed on a transporting aircraft.

Stretchers The number of stretchers required.

Escorts The number of escorts required. Some patients, such as children, require escorts, and these may have an impact on the number of patients an aircraft can transport.

Solitary Whether or not a patient can be transported with other patients. This field also indicates whether a patient is infectious or not, in which case extra time is required to disinfect a plane once the patient is dropped off.

Maximum Time The maximum time that the patient can remain on the aircraft from pickup to dropoff. This is included to partially account for patient "convenience." It is sometimes used by flight planners when an optimized plan calls for a patient to be kept on a plane for an inordinate amount of time.

Aircraft

Each aircraft used by Ornge has a range of identifying characteristics as follows.

Base Where the plane begins and ends each route.

Level of Care The level of care that the plane can provide. A plane can carry a patient whose level of care is at most equal to the level of care of the plane.

Stretchers Number of patients the plane can carry. This is usually 2, but for some planes is 1.

Escort Capacity The number of escorts that can be taken as a function of the number of stretcher-bound patients (1 or 2) on board.

Airspeed The cruising speed of the aircraft.

Fuel Cost Fuel cost per hour flown.

Charter Cost The charter cost per hour flown.

Advanced Charter Cost The charter cost per hour flown when the plane is carrying advanced or critical patients.

Routes and Costs

Each plane that is used in one day flies a *route* consisting of multiple *legs*, where each leg consists of a takeoff and landing with no intermediate stops. The route begins and ends at the plane's base. The cost of a route is rather complicated, but can be approximated as having three components as follows.

Fuel Cost Charged on each leg based on the time spent in the air. We assume an average fuel price per litre across the province for all aircraft.

Charter Cost Charged on each leg based on the time spent in the air and the level of care provided.

Detention Cost Charged per hour spent waiting on the ground in excess of a certain ground-holding time.

Computing the fuel and charter costs requires the time spent in the air, which primarily depends on the distance traveled. The distance traveled depends on weather systems that must be avoided, and on flight-path requirements. We ignore such complexities in our model, instead assuming that flights follow great-circle paths. The situation is complicated by wind. Even if wind speed and direction are constant over a route that begins and ends at the same place, the wind's effects do not cancel out. Instead, one must explicitly account for wind. We do not have access to detailed location-and-time-specific wind information, so instead assume a constant wind speed and direction across the province, and compute flying time accordingly.

There are two primary requirements of routes.

1. A route cannot last longer than a certain maximum length of time that is government mandated. We take this bound to be 12 hours. This time limit also, practically speaking, limits the number of requests that a plane can handle in one day. We take this limit to be 4, in accordance with current Ornge practice. As we will see, this limit on the number of requests is vital in ensuring that the optimization problems we tackle are computationally tractable.

2. A route cannot keep a patient on board for too long relative to how long the trip would have taken were the patient flown directly from origin to destination. Similarly, there

is a limit on how many legs a patient will travel on during their transport. These "soft" constraints are imposed and adjusted by flight planners to varying degrees.

The problem is to choose airplane routes, and takeoff times for each leg on the routes, that handle all requests at minimum total cost while not violating any of the constraints discussed above.

Impact and Next Steps

A tool of this form needs careful validation and indeed, we went through several phases of validation and adjusting of requirements and functionality, before reaching the current implementation. A key step in this process was a retrospective study, wherein fifty days were randomly sampled between July 2010 and February 2011 (MacDonald et al. 2011). For each day in the study, we completed the following steps.

1. Retrieve the actual schedule flown, along with the information available when the schedule was constructed.
2. Derive an optimized plan as described herein.
3. Calculate the total flying time, distance traveled, and number of legs where a plane was empty for the optimized plan.
4. Calculate the cost for the optimized plan based on data from financial records.
5. Use expert opinion to ensure the validity of the optimized plan.

Before discussing the results of the study, we should acknowledge three limitations. First, the statistics for the plans computed by flight planners incorporated effects such as the need to fly around weather systems, and other aviation factors, whereas those for the model did not. Second, the plans computed by the flight planners included any real-time disruption associated with newly arising calls that require reorganization of the schedule, whereas those for the optimized plan do not. Third, the costs modeled in the optimization tool are only an approximation for the realized costs (but these estimated costs were also used in the evaluation of the routing constructed by flight planners at Ornge).

For the study period there were a total of 838 requests for transfers, with a daily mean of 16.8 ± 5.4 requests. Based on the costs computed in this study, the optimized plans were projected to yield savings on the order of 16.5%. Further summary statistics in Table 1 provide evidence for the substantial benefits of an optimized plan. The differences in the table, when computed as averages over the 50 days in the study, are statistically significant at the 5% level of an unpaired t-test.

Table 1 Summary statistics showing the difference in plan characteristics as developed by flight planners without the optimization tool, and with the optimization tool.

	flight planners	optimized plan	difference
flights	312	305	7
hours	1417	1253	164
distance (km)	481,381	417,156	64,225
% empty legs	35.5	32.8	2.7

Given the limitations discussed above, the actual realized savings were projected to be smaller than 16.5%, but the clear benefits of the optimized plan are not in dispute, and Ornge management is extremely supportive of the tool. Ornge has subsequently adopted the tool for daily use, and the tool is now used except on days when the flight planners that are trained in its use are not available. However, Ornge recognizes the benefits of using the tool, and are now training additional staff in its use.

In terms of current work, there are two primary activities. First, disruption to the schedule is inevitable at some level, owing to weather effects, new calls that require urgent attention, and so forth. This leads one to the "schedule repair" problem, which is essentially a smaller version of the problem discussed above. Together with a team of Master of Engineering students we are adapting the tool to provide a schedule repair facility. Second, the approximate costs used in the model are recognized to have a number of limitations that lead to nontrivial discrepancies between the costs assumed in the model and those realized in practice, and we are working to reduce these discrepancies.

Other very interesting research questions have arisen as part of this study. For example, where should aircraft be based around the province to minimize expected daily cost? See Carnes (2010, Chapter 3) for an approximation algorithm that addresses this question on a stylized model of air-ambulance operations. This question is also related to ambulance-location problems (see Brotcorne et al. 2003 for a survey), although those models are designed with a view towards emergency response one request at a time, rather than through routes that cover a *set* of requests. Another natural question is how to design a plan for the scheduled requests that takes into account the potential disruptions that could arise in executing the plan. The natural modeling framework is two-stage stochastic programming, although other methods might be appropriate. This question is related to the *dynamic* dial-a-ride problem as surveyed in Cordeau and Laporte (2007).

Acknowledgments

Shane Henderson's work was partially supported by National Science Foundation grants CMMI-0758441 and CMMI-1200315. David Shmoys's work was partitionally supported by the National Science Foundation through grants CCR-0635121, DMS-0732196, CCF-0832782, and CCF-1017688. Tim Carnes's work was partially supported by the National Science Foundation through grants CCR-0635121, DMS-0732196, and CCF-0832782. We thank Master-of-Engineering students Jong-Yub Chae, Johannes Essl, Ying Xian, Ben Cheron, Anchal Dube, Lokesh Manohar, and Kevin Yu for their efforts in formulation and cost modeling, and Ph.D. student Alex Fix for programming assistance.

Appendix. Formulation

We formulate the problem using a set-partitioning integer program. In this integer program there is a binary variable associated with each combination of a plane and a set of up to 4 requests. Given that there are r requests and p planes, the number of variables n in the integer program is therefore

$$p\left(\binom{r}{1}+\binom{r}{2}+\binom{r}{3}+\binom{r}{4}\right).$$

In practice this is an upper bound because some plane/set-of-requests combinations are infeasible, and we omit such variables from the formulation. For example, for 30 requests and 40 airplanes there are 1,277,200 variables, but this typically reduced to approximately 750,000 in a number of test instances after removing infeasible combinations. Let the resulting variables be $x_1, x_2, \ldots, x_n$.

Define $A_{ij} = 1$ if Request i is included in the jth column, i.e., the jth plane/set-of-requests combination, and 0 if not. Let $B_{ij} = 1$ if Plane i is associated with the jth column, and 0 if not. Let $e^{(m)}$ denote the m-dimensional column vector with $e_i^{(m)} = 1$ for all i. Let c_j denote the cost of completing the set of requests in the jth column with the associated plane, as discussed shortly. The optimization problem is therefore a set-partitioning problem of the form

$$\begin{aligned} \min_x \ & c'x \\ \text{subject to } & Ax = e^{(r)} \\ & Bx \le e^{(p)} \\ & x \text{ binary.} \end{aligned}$$

It remains to discuss how the cost coefficient vector c is computed, and this is the heart of the problem. For now we ignore many of the side constraints on the problem, such as the need to transport solitary patients alone and so forth, and assume that all planes can carry two patients. We will discuss the additional complexities of the problem at the end of this appendix. For an amplified and more general discussion of the recursion described below, see Chapter 3 of Carnes (2010).

Fix a particular column, Column j say, in the formulation. The value c_j represents the minimal cost of handling the set of requests, $\mathcal{R}_j$ say, using the plane associated with Column j. Requests can be picked up and dropped off in any order, subject to the requirements listed in the "Problem" section. This complicates the computation of c_j. We enumerate all possible orderings of pickups and dropoffs of the requests $\mathcal{R}_j$, and choose the feasible sequence that minimizes costs. If no sequence is feasible, e.g., if the route would take too long and therefore violate the duty-day requirement, then the column is omitted from the formulation.

If $\mathcal{R}_j$ consists of a single request, then c_j is the cost of flying (if necessary) from the plane's base to the request's origin, on to the destination, and then returning to the plane's base.

Suppose now that $\mathcal{R}_j$ consists of $k > 1$ requests. We consider all $k!$ orderings of the requests in turn, where each ordering is defined by the order in which the requests are *picked up*. For each such sequence, assuming that planes can carry two patients, there are 3^{k-1} possible routes, where the route consists of the sequence of pickups and dropoffs. To see why, suppose that the first patient (the current patient) has already been picked up. The route can now either

1. drop off the current patient and pick up the next patient,
2. pick up the next patient and drop off the current patient, or
3. pick up the next patient and drop off the next patient.

In all three of these cases, we are left with one patient on board, so we can repeat the argument for each subsequent patient, obtaining a factor of three for every patient except the first one.

This argument also provides a recursion that we use to enumerate all possible routes that can be used to complete the set of requests $\mathcal{R}_j$. Each route is then costed, and tested for adherence to the many requirements and side constraints, and if feasible and cheaper than the best route so far, is retained as the incumbent route.

In performing the test for feasibility for a route, we take into account the plane's characteristics (level of care, capacity, etc). A minimum turnaround time is enforced at each stop on a route. Time on the ground is also added, e.g., to disinfect the aircraft after an infectious patient has been transported. These times are factored in when determining whether the time windows can be satisfied. We assume that the plane takes off at a time that is chosen by the flight planners, and we sequentially delay the takeoff times through the route to attempt to adhere to the time window constraints for requests. This will yield a feasible way to accommodate the time windows if one exists (Cordeau and Laporte 2007, Section 3.3), but it does not necessarily minimize the detention cost. One can apply an algorithm to minimize detention cost by delaying the initial takeoff (Savelsbergh 1992), but Ornge prefers to fix the initial takeoff times, partly for crew convenience. Once the takeoff times of the route are fixed, the cost of the route is then straightforward to compute.

We complete the above steps for every combination of the set of requests and planes. Therefore, the total computation required to assemble the set-partitioning problem is of the order

$$p \sum_{k=1}^{4} \binom{r}{k} k!\, 3^{k-1}, \tag{1}$$

where the upper bound of 4 in the summation arises due to current Ornge practice of not assigning more than this number of requests to any aircraft. It is imposed as a practical limit through Ornge policies, but it has the side benefit of ensuring that the recursion described above completes quickly, because if this bound were much larger, then the computational effort (1) for larger numbers of requests r would be formidable.

One might consider using column generation to attempt to avoid the computational limitations we face in using complete enumeration of the columns. However, this is not possible owing to the complicating side constraints that form an essential part of the formulation.

References

Beaudry, A., G. Laporte, T. Melo, S. Nickel. 2010. Dynamic transportation of patients in a hospital. *OR Spectrum* **32** 77–107.

Brotcorne, L., G. Laporte, F. Semet. 2003. Ambulance location and relocation models. *European Journal of Operational Research* **147** 451–463.

Carnes, Timothy A. 2010. Approximation algorithms via the primal-dual schema: Applications of the simple dual-ascent method to problems from logistics. Ph.D. thesis, Operations Research and Information Engineering, Cornell University, Ithaca NY.

Cordeau, Jean-François, Gilbert Laporte. 2007. The dial-a-ride problem: models and algorithms. *Annals of Operations Research* **153**(1) 29 – 46.

Cordeau, J.F., G. Laporte, S. Ropke. 2008. Recent models and algorithms for one-to-one pickup and delivery problems. *The vehicle routing problem: latest advances and new challenges* 327–357.

Engineer, F.G., G.L. Nemhauser, M.W.P. Savelsbergh. 2011. Dynamic programming-based column generation on time-expanded networks: Application to the dial-a-flight problem. *INFORMS Journal on Computing* **23**(1) 105–119.

Espinoza, D., R. Garcia, M. Goycoolea, G. L. Nemhauser, M.W.P. Savelsbergh. 2008a. Per seat, on demand air transportation part I: Problem description and an integer multicommodity flow model. *Transportation Science* **42** 263–278.

Espinoza, D., R. Garcia, M. Goycoolea, G. L. Nemhauser, M.W.P. Savelsbergh. 2008b. Per seat, on demand air transportation part II: Parallel local search. *Transportation Science* **42** 279–291.

MacDonald, Russell D., Mahvareh Ahghari, Timothy A. Carnes, Shane G. Henderson, David B. Shmoys. 2011. Use of a novel application to optimize aircraft utilization for non-urgent patient transfers. *Air Medical Journal* **30**(5) 255.

MacDonald, Russell D., L. Walker, Mahvareh Ahghari, Timothy A. Carnes, Shane G. Henderson, David B. Shmoys. 2012. Prospective, real-time use of an optimization application for non-urgent patient transfers using fixed wing aircraft. *Air Medical Journal* **31**(5) 230.

Parragh, S. N., J.-F. Cordeau, K. F. Doerner, R. F. Hartl. 2012. Models and algorithms for the heterogenous dial-a-ride problem with driver-related constraints. *OR Spectrum* **34** 593–633.

Parragh, S.N., K.F. Doerner, R.F. Hartl. 2008. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft* **58**(2) 81–117.

Savelsbergh, M. W. P. 1992. The vehicle routing problem with time windows: minimizing route duration. *ORSA Journal on Computing* **4** 146–154.