Scheduling and Routing Workers Teams for Ground Handling at Airports with Column Generation

Giacomo Dall'Olio[1], Rainer Kolisch[1]

TUM School of Management, Technical University of Munich, Germany
giacomo.dallolio, rainer.kolisch@tum.de

Keywords: Ground Handling Scheduling, Airport Opertations, Column Generation, Vehicle Routing.

1 Introduction

Air traffic has been constantly increasing over the past decades, and its annual growth for the next ten years is estimated to 4.6% (Boeing Commercial Airplanes 2019). An efficient management of the airport infrastructures is therefore crucial in order to avoid congestions and delays which are causes for high costs and customer dissatisfaction. Poor planning of ground handling is one of the main sources of delays (Oreschko *et. al.* 2011). Ground handling consists of those services which are necessary to prepare the aircraft for its next flight and are performed at the gates or at parking positions. Such services include baggage loading and unloading, interior cleaning of the aircraft and refueling. Aircrafts are kept on the ground for a limited amount of time, which causes ground handling tasks to have restricted time windows within they can be performed. It is desirable to get the ground handling tasks done as soon as possible, to make sure that the aircrafts are ready before the scheduled take-off time. Since ground handling tasks, from now on simply denoted as tasks, are interdependent, any delay could propagate to other tasks. Missing the due date of a task might lead to a flight delay, which translates to penalty costs and reduced quality service for the ground handler. Specialized workforce, the ground personell, is responsible for performing the tasks. Each ground worker has a qualification level, which allows her/him to perform tasks with a requirement equal or lower to her/his own. The planner has to assign the workers to the tasks according to their qualification level, and schedule the tasks avoiding workforce shortage and meeting the due dates.

In this paper, we propose a solution method, for the mentioned problem, based on the branch and price framework, where column generation is used to find a lower bound.

2 Problem Definition

Planning ground handling is a combination of routing, assignment and scheduling problems. The tasks are performed by teams of workers. The workers are grouped into teams making sure they have an adequate qualification to perform the assigned tasks. The qualifications are definded as hierarchical skill levels. Workers can perform a task of a certain level only if their skill level is equal or higher. Since the tasks are located at different parking positions, we have to plan a route for the workers, so that they are present at the locations of the tasks in time to carry them out. A schedule for all the tasks has to be found, so that the tasks are performed as soon as possible.

Some of the tasks can be performed in more than one execution mode. An execution mode defines the number of workers needed to carry out the task in a certain amount of time. Modes which require more workers to perform a task also require less time. Teams can only perform tasks which entail a mode requiring a number of workers equal to the number of members of the team. In order to avoid complex synchronizing interactions, the workers leave the depot in teams, reach one or more task locations at which they perform the corresponding tasks and return back to the depot. Teams are not fixed for the whole time horizon, since the workers are free to form new ones as they come back to the depot.

Let us define $\mathcal{I}$ as the set of tasks and $\mathcal{K}$ as the set of possible skill levels. Each task i has to be performed within its time window $[ES_i, LF_i]$. The set $\mathcal{M}_i = \{m_i^{min}, ..., m_i^{max}\}$ represents the different modes in which task i can be peformed. When task i is performed in a certain mode, the number of workers needed corresponds to m_i, while $p_{i,m}$ is the corresponding execution time; notice that $p_{i,m} > p_{i,m+1}$. The earliest finish time is therefore$EF_i = ES_i + p_{i,m^{\max}}$. We define a tour r as the sequence of tasks carried out by a team composed by f_r members with skill levels equal or higher than q_r. During its tour, the team can perform tasks with a skill level requirement equal or lower than q_r which entails an execution mode equal to f_r. The tour r specifies the start time S_i^r and end time F_i^r for each performed task. A feasible tour r must therefore be compliant with the following:

$$
\begin{aligned}
&S_i^r \geq ES_i \\
&F_i^r \leq LF_i \\
&S_i^r + p_{i,m} = F_i \\
&S_j^r \geq F_i^r + d_{i,j}
\end{aligned}
$$

where i and j represent two consecutive tasks in the tour sequence and $d_{i,j}$ is the time needed to go from i to j. Since our goal is to complete the tasks as soon as possible, we introduce a penalty for each scheduled task, that is the difference between its earliest finish time and its actual finish time. We can therefore define the cost c^r of a tour r as

$$c^r = \sum_{i \in \mathcal{I}_r} (F_i^r - EF_i) \tag{1}$$

where $\mathcal{I}_r$ is the set of tasks performed during the tour. Supposing we can generate all possible team tours, we can write down the following formulation:

$$
\begin{aligned}
\textbf{min} \quad & \sum_{k=1}^{K} \sum_{r \in \Omega_k} c_k^r \lambda_k^r && && (2)\\
\textbf{s.t.} \quad & \sum_{r \in \Omega_k} a_{k,i}^r \lambda_k^r \geq 1 && \forall k \in \mathcal{K}, \forall i \in \mathcal{I} && (3)\\
& \sum_{k'=k}^{K} \sum_{r \in \Omega_{k'}} b_{k',t}^r \lambda_{k'}^r \leq \sum_{k=k'}^{K} N_{k'} && \forall k \in \mathcal{K}, \forall t \in \mathcal{T} && (4)\\
& \lambda_k^r \in [0,1] && \forall k \in \mathcal{K}, \forall r \in \Omega_k && (5)
\end{aligned}
$$

The tours are grouped by skill level k in order to simplify the notation. The set of tours of skill level k is Ω_k and λ_k^r is the binary variable which is 1 if tour r of skill level k is selected in the solution, 0 otherwise. The parameter $a_{k,i}^r$ is equal to 1 if the team from tour r performs task i, 0 otherwise. The parameter $b_{k,t}^r$ is equal to the number f of team members (which need to own a skill level of at least k) for those instants t when the team is operating, 0 otherwise. The overall number of available workers of skill level k is denoted as N_k. Constraint (3) enforces that every task is performed. Constraint (4) ensures that the number of workers is not exceeded in any time instant.

3 Literature Review

Given its strategic importance, ground handling has been considerably investigated in the literature. Nevertheless, not many publications tackle the problem from a combined scheduling and routing point of view. In Dohn *et. al.* (2009) teams are fixed before they are routed across the tasks. There is no schedule time optimization since the focus is to maximize the number of tasks performed. Fink *et. al.* (2019) focus on the Abstract VRP with Workers and Vehicle Synchronization (AVRPWVS), which they apply to ground handling. This problem, however, does not include any kind of qualifications or skills, which are necessary in such a setting. In the AVRPWVS, workers need to be synchronized in time and space at the task locations. This dramatically increases the complexity, making it hard to solve real-world instances. Dohn *et. al.* (2009) as well as Fink *et. al.* (2019) use a column generation approach, which is known to have good performances in solving vehicle routing problems with time windows (see Desrochers *et. al.* (1992)). Manpower allocation with hierarchical skill levels has been investigated, on a general level, by Bellenguez-Morineau and Néron (2007). Practical applications can be found in Cordeau *et. al.* (2010) and Firat and Hurkens (2012). In these papers, the travel time needed to move from the location of a task to another one is neglected, differently from our problem setting. The multi-mode RCPSP has been solved to optimality by Sprecher and Drexl (1998). In Hartmann and Briskorn (2010) a survey on the topic can be found.

4 Proposed Solution Approach

We propose the use of column generation to find lower bounds, and branch and price to find the optimal integer solution. We define the *continuous master problem* (MP) as the linear relaxation of the model proposed in Section 2. The value of an optimal solution of the MP is therefore a lower bound for the original problem. Furthermore, we introduce the *restricted master problem* (RMP), which has exactly the same structure of the MP, but is defined over a subset of tours $\Psi \subset \Omega$. Column generation consists of an iterative process where RMP is solved and the values of the dual variables are used to generate new promising tours. A new tour can improve the current RMP solution only if its reduced cost is negative. If it possibile to generate a new feasible tour with a negative reduced cost, the tour is added to Ψ and a new iteration of the column generation starts. Otherwise, the current solution of the RMP cannot be improved, therefore it is optimal for the MP and its value is a valid lower bound for the original problem. The reduced cost of a tour r with f team members working at level k is the following:

$$c_k^r - \sum_{i \in \mathcal{I}} a_{k,i}^r \mu_{k,i} + \sum_{t \in \mathcal{T}} b_{k,t}^r \delta_{k,t} \tag{6}$$

where μ and δ are respectively the values of the dual variables corresponding to constraints (3) and (4). The reduced cost of a team tour can be interpreted as follows. For each task performed during the tour, a penalty has to be paid if the end time is subsequent to the earliest finish time (c_k^r). The first summation is a reward obtained for every performed task while the second summation is a penalty paid for using limited resources (i.e. workers) at specific instants in time. The pricing problem is the problem of finding a tour of minmum reduced cost. Since a tour has a predefined number of team members f who work at a maximum skill level q, we have to solve the pricing problem multiple times with different settings. When solving a pricing problem for a team of f workers working at level q, only the tasks involved are those which entail an execution mode with f workers and whose skill level requirement equal or less than q. We can model the pricing problem with a time expanded network, where we have two types of nodes for each task: *start nodes* and *end nodes*. For each suitable task i, the network encompasses a start node for each possible start time of i, and a leave node for each instant from EF_i until the end of the time horizon. Each start node has one outgoing *execution arc* connecting it to an end node according to the execution time. The weight of an execution arc from node (i, t_S) to (i, t_F) corresponds to the reward for performing task i, plus the penalty for ending i at t_F (if any) and the penalty for keeping f workers busy from t_S to t_F. An end node (i, t_i) is connected to a start node (j, t_j) with a *travel arc* if $i \neq j$ and $t_j = t_i + d_{i,j}$. Travel arcs also connect the source node to all start nodes and all end nodes to the sink node. Start and sink nodes represent respectivly the leaving from and the returning to the depot. The weights of the travel arcs represent the penalty for keeping the workers busy from t_i to t_j. If the origin of the arc is the source node, $t_i = t_j - d_{\text{depot},i}$ while if if the destination of the arc is the sink node, $t_j = t_i + d_{i,\text{depot}}$. Eventually, two consecutive end nodes (i, t_i) and $(i, t_i + 1)$ referring to the same task i are connected with a *waiting arc*. The weight of a waiting arc corresponds to the penalty for keeping workers busy, therefore it follows the rule for travel arcs. Given the described network, the pricing problem can be solved finding a shortest path from the source node to the sink node.

5 Experimental Study

In order to verify the quality of our approach we will test the proposed algorithm on data from a major European airport. Based on these data, we generated various realistic test instances. The instances cover from 30 minutes up to 4 hours of a working day. Since the flights are not equally distributed during the day, we differentiate the instances in *low*, *medium* and *high* workload. The final results of the experimental study will be presented in the conference.

References

Bellenguez-Morineau, O., and Néron, E. , 2007, “A branch-and-bound method for solving multi-skill project scheduling problem", *RAIRO-operations Research*, Vol. 41(2), 155-170.

Boeing Commercial Airplanes, 2019, *Current Market Outlook 2019-2038*, Internet.

Cordeau, J. F., Laporte, G., Pasin, F., and Ropke, S., 2010, “Scheduling technicians and tasks in a telecommunications company.", *Journal of Scheduling*, Vol. 13(4), 393-409.

Desrochers, M., Desrosiers, J., and Solomon, M., 1992, “A new optimization algorithm for the vehicle routing problem with time windows.", *Operations research*, Vol. 40(2), 342-354.

Dohn, A., Kolind, E. and Clausen, J, 2009, “The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach.", *Computers & Operations Research*, Vol. 36(4), 1145-1157.

Frey, M., Desaulniers, G., Kiermaier, F., Kolisch, R., and Soumis, 2019, “Column generation for vehicle routing problems with multiple synchronization constraints.", *European Journal of Operational Research*, Vol. 272(2), 699-711.

Firat, M., and Hurkens, C. A. J., 2012, “An improved MIP-based approach for a multi-skill workforce scheduling problem.", *Journal of Scheduling*, Vol. 15(3), 363-380.

Hartmann S. and Briskorn D., 2010, “A survey of variants and extensions of the resource-constrained project scheduling problem.", *European Journal of Operational Research*, Vol. 207(1), 1-14.

Oreschko, B., Schultz, M., and Fricke, H., 2011, “Skill analysis of ground handling staff and delay impacts for turnaround modeling", *Air Transp. Oper*, pp. 310-318.

Sprecher A. and Drexl A., 1998, “Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm.", *European Journal of Operational Research*, Vol. 107(2), 431-450.