Scheduling ground-handling services: a bi-objective approach

Daniel Guimarans [1], Silvia Padrón [2], Juan José Ramos [2], Salma Fitouri-Trabelsi [3] and Liana Napalkova [2]
[1] Optimisation Research Group, National ICT Australia (NICTA), 13 Garden Street, Eveleigh NSW 2015, Australia
daniel.guimarans@nicta.com.au
[2] Dpt. Telecommunication and Systems Engineering, Universitat Autònoma de Barcelona, Emprius 2, 08202 Sabadell, Barcelona, Spain
{silvia.padron, juanjose.ramos, liana.napalkova}@uab.cat
[3] Mathématiques Apliquées, Informatique et Automatique pour l'Aérien & Air Transportation Department, L'École Nationale de l'Aviation Civile (ENAC), 7 Avenue Édouard Belin, 31055 Toulouse, France
salma.fitouri-trabelsi@enac.fr

Proc. ICAOR 2015
Vienna, Austria

Keywords:
Air transportation
Ground handling
Metaheuristics
Multi-objective optimisation

Abstract

Preparing an aircraft for its next flight requires a set of interrelated services involving different types of vehicles. Planning decisions concerning each resource affect the scheduling of the other activities and the performance of the other resources. Considering the different operations and vehicles instead of scheduling each resource in isolation allows integrating decisions and contributing to the optimisation of the overall ground-handling process. This goal is defined through two objectives: (i) minimising the waiting time before an operation starts and the total reduction of corresponding time windows, and (ii) minimising the total completion time of turnarounds. We combine different technologies and techniques to solve the problem efficiently. A new method to address this bi-objective optimisation problem is also proposed. The approach has been tested using real data from a major Spanish airport, obtaining different solutions that represent a trade-off between both objectives. Experimental results permit inferring interesting criteria on how to optimise each resource, considering the effect on other operations. This outcome leads to more robust global solutions and to savings in resources utilization.

Introduction

The notable growth of air traffic in recent years has led to increasingly congested airports and significant flight delays. In 2013, approximately 36% of European flights were delayed on departure by more than 5 minutes, with an average delay of 26.7 minutes (Eurocontrol, 2013). Many aircraft delays can be attributed to overlong turnarounds due to a lack of planning integration of the different activities and an inefficient use of resources (Titan, 2010). Turnaround is defined as the period of time the aircraft is on the ramp between an inbound and outbound flight. During this time, different ground-handling operations are performed. Ground handling comprises the activities, operations procedures, equipment requirements, and personnel necessary to prepare an aircraft for the next flight. These ground tasks are very interdependent. Therefore, each operation is a potential source of delays that could be easily propagated to other ground operations and other airport processes (Fricke & Schultz, 2009; Norin *et al*, 2009).

Due to the hierarchy of overall airport planning, ground handlers are generally not included in the decision making of other scheduling processes (e.g. flight scheduling, stand allocation, etc.). This means they often must fit their planning around a set of hard constraints. These constraints include aircraft arrival, departure, scheduled turnaround time, and stand allocation, among others (Leeuwen, 2007).

In this work, we present a novel and efficient approach to tackling the ground-handling scheduling problem from a global perspective, considering all activities to be performed. To the best of our knowledge, this is the first time the problem is treated as a whole in the literature. Thus far, other approaches have been developed to optimise operations in isolation, but they do not consider the relationships between all the involved activities. Regarding ramp operations, Du *et al* (2008) proposed an Ant Colony approach to schedule fuelling vehicles based on the Vehicle Routing Problem with Tight Time Windows (VRPTTW) with multiple objectives. Clausen (2011) focused on connecting baggage transportation and proposed a greedy algorithm based on an Integer Programming model for the Vehicle Routing Problem with Time Windows (VRPTW). Norin *et al* (2009) proposed an interesting integration of a simulation model of various operations during turnaround and the scheduling of de-icing trucks obtained by a greedy optimisation algorithm. A more sophisticated solution was proposed by Ho & Leung (2010) to tackle catering operations including staff workload.

In our approach, we do explicitly consider relationships between activities to solve the problem from a global point of view. To do so, we develop a bi-objective optimisation methodology aiming at minimising waiting time before operations start and improving overall turnaround performance. We decompose the problem to apply efficient techniques. Each task can be modelled as a VRPTW, as it is associated to a particular type of vehicle. These are solved individually using the well-known Insertion Heuristics method (Solomon, 1987) and a hybrid methodology (Guimarans, 2012). Decisions made on the routing of one type of vehicle are propagated to the other tasks (i.e. VRPTWs) through reductions in the available time windows. Modifying the order in which problems are solved yields to different time windows reductions and different overall solutions. To explore different sequences in an informed manner and to address the bi-objective problem, we have developed a new method called *Sequence Iterative Method* (SIM). This process has proved to be a consistent method to solve the complete problem and provide a range of solutions representing the best trade-off between the two objectives, as we show in our results on data from Palma de Mallorca airport (PMI).

Problem description

Ramp operations take place at the aircraft parking position between the time it arrives at the stand *(In-Blocks)* and its departure *(Off-Blocks)*. Figure 1 shows an example of the main activities during a typical turnaround when the aircraft is parked at a contact point (i.e. the stand is connected to the terminal via a bridge).

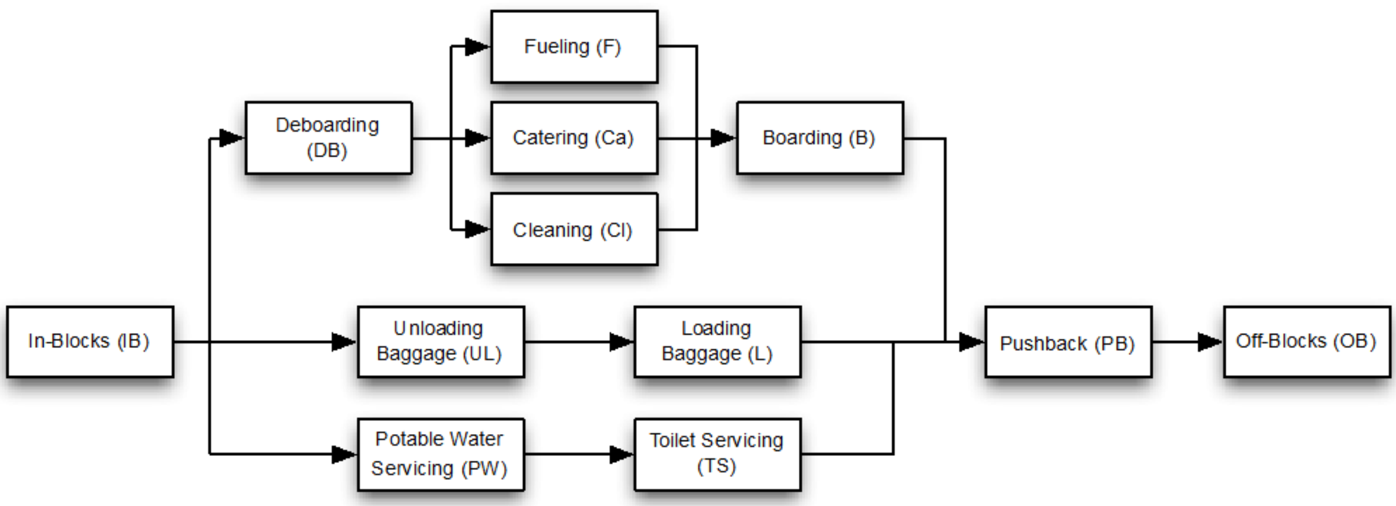

Fig. 1. Example of activity flow during a turnaround at a contact point

Because the turnaround is a very complex process, its duration depends on many different variables. These include operational variables related to the aircraft type (size, number of seats), the number of tasks, parking position at a contact or remote stand, and the service time required to carry them out (full servicing or minimum servicing). Some activities are affected by precedence constraints imposed due to security issues, space requirements or airline policies. The end of the turnaround process is determined by the off-block time, when all doors are closed, the bridge is removed, the pushback vehicle is present and the aircraft is ready for startup and push back (Fricke & Schultz, 2009).

A specific type of vehicle performs each operation. According to the task, some vehicles with a given capacity must transport some quantity of resources to the aircraft stand (e.g. catering) or collect waste from the aircraft (e.g. toilet servicing). Likewise, some vehicles do not transport any resource (e.g. pushback). To simplify the model, we selected the main activities of a full servicing turnaround on aircraft parking at a contact point. In addition, we have not considered baggage transportation. This task has special features in relation to other ground-handling activities (e.g. multiple trips, split servicing, multiple depots, etc.) and requires a specific model and solution method.

At each aircraft, operations must be performed within the defined turnaround time. Hence, a time window to begin the service is assigned to each activity, which considers the duration of each task and the precedence constraints.

Scheduling decisions made for one service affect other activities. Tasks belonging to the same aircraft are related according to precedence restrictions, as well as to their corresponding time windows. Due to these restrictions, the time when an operation begins could reduce the time windows of other activities, and consequently the performance of vehicles servicing them.

Optimising each resource while considering the effect on other operations permits an integration of planning decisions and contributes to optimising the overall ground service process. We aim to minimise the operation waiting time, i.e. accomplishing each operation as early as possible in relation to its original time window, and minimising the total reduction of time windows. This reduction may affect the number of vehicles required to service all aircraft, and therefore we implicitly minimise the number of required vehicles. Our second objective is to minimise the total completion time of ground services at each aircraft. That is, we want to balance robustness of scheduling each operation with good performance of the turnaround, using vehicles efficiently.

Our first objective aims at performing operations in a set of N aircraft as soon as possible through two arguments: minimising the total operation waiting time and the total reduction of the time windows. Let the waiting time $w_i = t_i - a_i$ be the difference between the starting time of a given operation at aircraft i (t_i) and the earliest starting time according to its associated time window (a_i). Let Δ_i denote the time window reduction of such operation at aircraft i, such as $\Delta_i = (\alpha_i - a_i) + (\beta_i - b_i)$, where α_i (β_i) is the original earliest (latest) start time and a_i (b_i) is the actual earliest (latest) start time for such operation. An aggregate function f^1 is defined to describe how early operations are performed by each vehicle $v \in V$:

$$f_v^1 = \sum_{i \in N} (\Delta_i + w_i) \quad \forall v \in V$$

The first objective function F1 is then defined as:

$$(F1)\ \min \sum_{v \in V} f_v^1$$

Let l be the last operation on each aircraft and t_i^l the start time of such operation at aircraft i. We then define the second objective function F2 in order to minimise the completion time in all N aircraft:

$$(F2)\ \min \sum_{i \in N} t_i^l$$

Solution method

We have developed a bi-objective algorithm for solving the ground-handling problem. This method is based on a work centre-based decomposition strategy (Sourirajan & Uzsoy, 2007). Most methods used to solve this decomposition derive from the Shifting Bottleneck procedure (Adams *et al*, 1988). Applying this procedure in our particular case, where each sub-problem is a VRPTW, can lead to long execution times. Thus, we followed a similar schema but combined two processes to obtain a complete solution at each iteration. In the first process, which we call *Solving Process* (SP), all sub-problems are solved one after another given a predefined order. Each time a sub-problem is solved, the time windows of the remaining sub-problems are updated to maintain consistency among sub-solutions. The SP is embedded in an iterative schema that we call *Sequence Iterative Method* (SIM), outlined in Figure 2. The goal of this second process is to improve the overall solution when dealing with the defined bi-objective optimisation problem. We modify sub-problems' solving sequence at each iteration according to the previous solution, and the SP is called again with the new sequence.

In SP, we use Constraint Programming (CP) to implement a procedure to find the time windows of each operation according to arrival and departure times and imposed precedence constraints between operations. Then, a sub-problem is identified for each task (i.e. type of vehicle) and a routing problem is solved.

Each routing problem is solved in two stages. First, we use the well-known *I3* construction heuristic (Solomon, 1987) to obtain a reasonably good initial solution. The number of vehicles obtained in this step is taken as an upper bound of the number of vehicles needed to perform the operations in all aircraft. Imposing this value as the size of the available fleet, a CP-based hybrid method (Guimarans, 2012) is applied in the second stage to improve the initial solution by minimising the operation waiting time f^1. In this methodology, the modelling and constraint propagation advantages of CP are combined with local search methods. Using the concept of operators based on Large Neighbourhood Search (LNS), the local search process is embedded in CP. These operators destroy and repair the solution to re-optimise parts of the problem. Destroy, in this case, means identifying a set of aircraft to remove from a sequence of visits. Repair refers to finding a better way to reinsert these aircraft into the partial solution. In addition, the methodology employs Variable Neighbourhood Search (VNS) to guide operators' selection, a metaheuristic often applied to VRPs with interesting results (Guimarans *et al*, 2011).

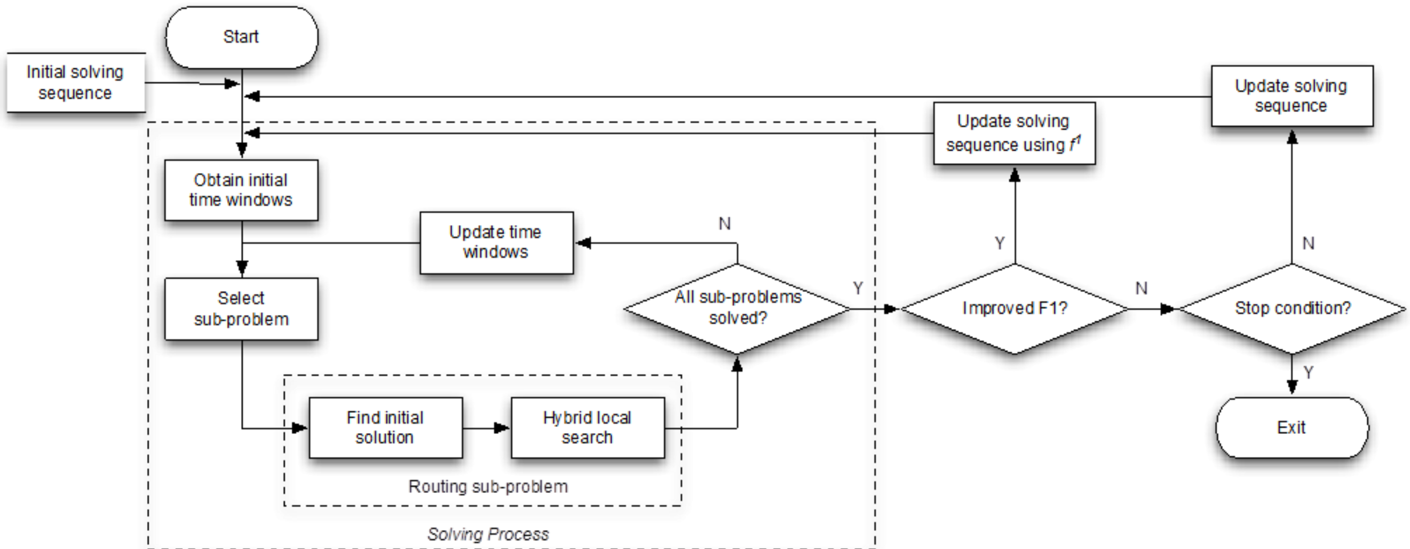

Fig. 2. Flow diagram for the *Sequence Iterative Method* (SIM)

After solving a sub-problem, an explicit process to update the remaining time windows is needed to ensure consistency with the rest of sub-problems. Once again, we take advantage of propagation features of CP to implement a simple strategy to maintain such consistency. Finally, when all sub-problems are solved, the SP is stopped.

According to Sourirajan & Uzsoy (2007), determining the next machine to be scheduled is one of the more important steps in decomposition procedures based on Shifting Bottleneck. The sequence in which machines are included in the partial schedule can reduce the re-optimisation process without loss in solution quality. For this reason, we have developed the SIM, aiming at improving the solution by modifying the order in which sub-problems are solved.

Following a *scalarization* schema for multi-objective problems (Jozefowiez *et al*, 2008), the problem is solved with respect to the first objective *F1*, and the value of the second objective *F2* is calculated from the obtained solution. At each round, sub-problems' solving sequence is modified to find a solution in the Pareto set to cover it in the best possible way. Regardless of the type of aircraft, the ground-handling service always finishes by pushing away the aircraft from its parking position (pushback). We used this information to create an initial sequence to obtain a lower bound of *F2*.

Let S be the ordered set of sub-problems where each sub-problem corresponds to each type of vehicle (i.e. task) involved, $|S|=|V|$. The order in S describes the sequence in which sub-problems are solved; s_l is the sub-problem corresponding to the last operation; B is the set of sub-problems to solve before s_l such that $B \subseteq S\backslash\{s_l\}$; and R represents the remainder of sub-problems such that $R = S - \{s_l\} - B$.

In the first step of the SIM, an initial sequence in S is created such that the s_l is the first sub-problem to solve. When a sub-problem is solved first, corresponding operations are scheduled within their original time windows. If this sub-problem is the pushback, a lower bound of *F2* is obtained. On the other hand, this reduces the original time windows of other tasks on the same aircraft, i.e. the time windows of the elements in R. Thus, a worse value of *F1* is obtained.

At first, the elements in R are sorted according to the actual order in which different tasks are carried out at each aircraft (see Figure 1). In principle, when solving the last operation first, the best value of *F2* is obtained regardless the order of the elements in R. However, solutions found should be as close as possible to the Pareto optimal set, i.e. a solution with a lower bound of *F2* with the minimum value of *F1*. Therefore, after obtaining a solution with the initial sequence by means of SP, sub-problems in R are ordered by their f^1 values. Then, we repeat the process to obtain a better sequence of R.

In a second step, SIM aims at improving the value of *F1*, planning the remainder of sub-problems before the last operation. At each round, the sub-problem with the highest value of f^1 in R is selected for inclusion in B, and solved first. Adding sub-problems to B, i.e. prioritising other operations with respect to s_l, usually leads to a decreasing *F1*. Similar to the first step, the chosen sub-problem is scheduled within its original time windows, which leads to a lower bound of its f^1. After scheduling all operations, we sort B and R in decreasing order of f^1 and use SP to solve the new sequence. We repeat the process until no further improvements on *F1* are obtained. Thus, an improvement of *F1* is reached while a new value of *F2* is found. We then select the next sub-problem to be included in B. The process is repeated until all operations are scheduled before s_l, that is, $B = S\backslash\{s_l\}$.

Computational experiments

The methods described in this paper have been implemented in Java and linked to the CP platform ECLiPSe 6.0. All tests were run on a server with an Intel Xeon processor at 2.66GHz and 16GB RAM.

To the best of our knowledge, no benchmark instances exist for the ground-handling problem. We generated a set of scenarios based on real data provided by PMI to validate the proposed approach. This data is subject to a strict confidentiality agreement and therefore cannot be disclosed. We used a flight schedule corresponding to a summer business day and considered all aircraft performed a turnaround. This dataset contains scheduled arrival and departure times, type of aircraft, and assigned parking positions. We assume constant speed to calculate vehicle travel times between these positions. In our instances, we modelled three types of aircraft with different sizes (types I, II and III, in increasing order).

For each operation and using manufacturer specifications for each aircraft type, we defined the duration, precedence restrictions and the type of vehicle used. Three sets of instances C1, C2 and C3, were generated, modifying the precedence constraints to test the algorithm. The flight schedule was divided into three eight-hour shifts scheduled separately J1, J2 and J3, servicing 42, 64 and 83 aircraft, respectively. We combined these shifts with the different sets to obtain 9 instances to test our approach. Table 1 provides a summary of results obtained with our methodology for the generated instances.

Table 1. Solutions obtained for PMI instances using SIM. Non-dominated solutions are marked in bold.

Sol.	*C1J1*		*C1J2*		*C1J3*		*C2J1*		*C2J2*		*C2J3*		*C3J1*		*C3J2*		*C3J3*	
	F1	*F2*	*F1*	*F2*	*F1*	*F2*	*F1*	*F2*	*F1*	*F2*	*F1*	*F2*	*F1*	*F2*	*F1*	*F2*	*F1*	*F2*
1	**2383**	**1594**	**3389**	**2362**	**4717**	**3009**	**2596**	**1589**	**3770**	**2331**	**4867**	**2994**	**2798**	**1584**	**3688**	**2360**	**4848**	**3029**
2	**2165**	**1613**	**4064**	**2346**	**4281**	**3057**	**2369**	**1607**	**3309**	**2357**	**4524**	**3053**	**2748**	**1600**	**4271**	**2345**	**4715**	**3068**
3	**1980**	**1621**	3509	2403	4608	3067	**2142**	**1629**	**3002**	**2381**	**4096**	**3083**	**2360**	**1635**	**3504**	**2368**	**4098**	**3110**
4	2425	1619	**2619**	**2393**	**3806**	**3112**	**2364**	**1610**	**3438**	**2346**	4394	3104	**2640**	**1613**	**3036**	**2407**	**4640**	**3087**
5	**1850**	**1655**	**2464**	**2414**	**3282**	**3130**	**1864**	**1670**	**2623**	**2422**	**3590**	**3109**	**2086**	**1659**	3832	2403	**3741**	**3161**
6	2154	1646	3162	2399	**4121**	**3100**	**2096**	**1654**	**2869**	**2385**	3940	3153	2476	1639	**2809**	**2455**	4521	3133
7	**1709**	**1695**	**2101**	**2466**	**3169**	**3185**	**1762**	**1694**	**2349**	**2464**	**3360**	**3177**	**1983**	**1670**	3548	2447	**3521**	**3181**
8	1998	1681	2904	2464	3896	3173	**1784**	**1680**	2756	2444	**3046**	**3203**	2275	1662	**2543**	**2478**	4271	3186
9	**1565**	**1715**	**1833**	**2490**	**2818**	**3207**	**1671**	**1707**	**2157**	**2488**	3428	3215	**1764**	**1714**	**2360**	**2517**	**3264**	**3218**
10	**1816**	**1687**	2754	2486	3547	3197	**1709**	**1699**	2586	2486	**2790**	**3262**	2170	1692	3067	2514	3964	3200
11	**1510**	**1736**	**1756**	**2508**	**2622**	**3264**	**1513**	**1712**	**1924**	**2513**	3279	3286	**1729**	**1734**	**2206**	**2539**	**3108**	**3280**
12	1792	1714	2499	2509	3301	3252	1676	1710	-	-	-	-	2054	1711	2863	2522	3690	3265

Vehicle utilisation is an important aspect of how scheduling decisions of a resource affect the other ones. We observed an increase in the vehicles needed to perform other operations whenever the pushback was solved first. Obtaining lower values of *F2* implies a time window reduction on other operations at the same aircraft, and consequently an increment of employed vehicles. For instance, in C1J1 baggage operations needed 19 vehicles when pushback was solved first (Sol. 1), while it uses 16 when it is solved last (Sol. 11 and 12). This might be an interesting criterion to select a solution or prioritise an operation according to the particular situation of a vehicle type, e.g. temporary unavailability or cost-based criteria.

In addition, we evaluated the performance of the proposed CP-based methodology to solve each task sub-problem. We compared our results to two modified versions of SIM: a first one using only the I3 heuristic; a second one substituting our approach by another state-of-the-art approach for the VRPTW (Woch & Lebkowski 2009). In the first case, our hybrid methodology is clearly able to improve the obtained solutions, although at a higher computational cost (Figure 3 left). In the second case, we found results were both comparable in quality and computational time (Figure 3 right).

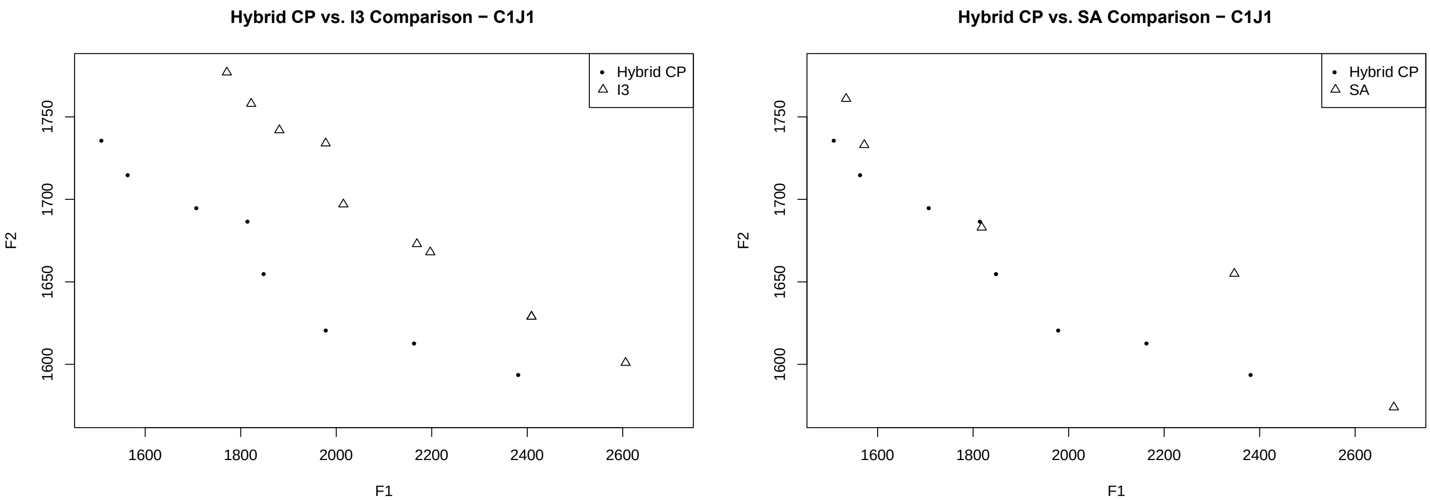

Fig. 3. Hybrid CP-based methodology vs. using only the I3 heuristic (left) or another state-of-the-art approach for the VRPTW (right)

Conclusion

In the present paper, we have presented a first approach for scheduling ground-handling vehicles at an airport. Different operations and types of vehicles have been considered to tackle this problem from a holistic perspective. We have modelled ground-handling services as a bi-objective optimisation problem, aiming to integrate the scheduling decisions about each resource and to contribute to the optimisation of the overall process. This goal is defined through two objectives: (i) minimising the operations waiting time and the total reduction of the time windows, and (ii) minimising the total completion time of the turnarounds. The problem has been decomposed to allow the model and the solution method to be simplified without losing the global approach of the proposal. Decisions are propagated between different sub-problems to ensure that local solutions can be integrated to obtain a feasible global solution. A new method called Sequence Iterative Method has been developed to improve the global solution when dealing with the bi-objective optimisation problem.

Our approach has been tested using real data from Palma de Mallorca airport and specifications from aircraft manufacturers. Results show that different solutions representing a trade-off between objectives were found by modifying the order in which operations are scheduled. Moreover, the number of vehicles needed to perform operations can change according to this order. This might be an important criterion to select between two solutions with similar values of the objective functions. Different aspects remain for further development of the presented work. The inclusion of baggage transportation or passenger transfer to remote stands will further enrich this study. Furthermore, we have assumed a homogeneous fleet of each type of vehicle. Considering a heterogeneous fleet and including this constraint in the model is another topic for future research.

Acknowledgments—NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

This work is partially funded by the European Commission FP7 project INnovative TEchnologies and Researches for a new Airport Concept towards Turnaround coordinatION (INTERACTION project). Web page: http://www.interaction-aero.eu/.

References

Adams, J., Balas, E., & Zawack, D. (1988). The Shifting Bottleneck procedure for job shop scheduling. Management Science, 3, 391-401.

Clausen, T. (2011). Airport ground staff scheduling. PhD Thesis, Technical University of Denmark, Denmark.

Du, Y., Zhang, Q., & Chen, Q. (2008). ACO-IH: An improved ant colony optimization algorithm for airport ground service scheduling. IEEE International Conference on Industrial Technology, 1-6.

Eurocontrol (2013). CODA Digest. Delays to air transport in Europe – Annual 2013.

Fricke, H., & Schultz, M. (2009). Delay impacts onto turnaround performance. 8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009), Napa, CA.

Guimarans, D. (2012). Hybrid algorithms for solving routing problems. PhD Thesis, Autonomous University of Barcelona, Spain.

Guimarans, D., Herrero, R., Riera, D., Juan, A. A., & Ramos, J. J. (2011). Combining probabilistic algorithms, constraint programming and Lagrangian relaxation to solve the vehicle routing problem. Annals of Mathematics and Artificial Intelligence, 62(3-4), 299-315.

Ho, S. C., & Leung, J. M. Y. (2010). Solving a manpower scheduling problem for airline catering using metaheuristics. European Journal of Operational Research, 202(3), 903-921.

Josefowiez, N., Semet, F., Talbi, E. G. (2008). Multi-objective vehicle routing problems. European Journal of Operational Research, 189(2), 293-309.

Leeuwen, P. van (2007). CAED D2: Modelling the turnaround process - CARE INO III: The coordinated airport through extreme decoupling. Technical Report NLR-CR-2007-200, Technical University Delft, Netherlands.

Norin, A., Granberg, T. A., Värbrand, P., & Yuan, D. (2009). Integrating optimization and simulation to gain more eficient airport logistics. 8th USA/Europe Air Traffic Management Research and Development Seminar (ATM2009), Napa, CA.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35(2), 254-265.

Sourirajan, K., & Uzsoy, R. (2007). Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication. Journal of Scheduling, 10(1), 41-65.

Titan (2010). Turnaround Integration in Trajectory and Network Project – Analysis of the current situation.

Woch, M. & Lebkowski, P. (2009). Sequential simulated annealing for the vehicle routing problem with time windows. Decision Making in Manufacturing and Services, 3(1-2), 87-100.