Tomasz JANIK*, Piotr ŁEBKOWSKI*

RESCHEDULING OF ARRIVALS AND DEPARTURES AT MULTI-RUNWAY AIRPORTS

Abstract: The issues related to the changeability of the environment concerns primarily all the service and production systems. Proper selection of the control tools and their adjustment to the actual system operation conditions decide of the whole system effectiveness. This paper is an attempt at the development of an efficient tool supporting the control of the airplane arrivals and departures at the airports using more than one runways, with the application of the reactive approach.

Keywords: rescheduling, robust project scheduling, reactive scheduling.

1. Introduction

The environment changeability causes the occurrence of diverse disruptions during the execution of a planned service process, and, consequently, it reduces the process quality and prevents the process continuation. Such situations require taking proper action so that negative consequences of system operation would be as limited as possible. Among the various approaches that are used for solving the problem, there are the methods of reactive task sequencing created with the intention of ensuring effective production system management (Tanimizu *et al.* 2006, van de Vonder *et al.* 2007).

For the purposes of reactive sequencing, strategies, methods and tactical solutions were developed. Their selection and effectiveness depends on the features of the system in which they are planned to be applied. The strategies described in Chong *et al.* 2003, Dorn 2001, Herroelen *et al.* 2004, Kizilisik 1999, Szelke *et al.* 1994, Tanimizu *et al.* 2006 and Thomas 2000 allow to take into account interference by granting priorities to particular tasks in reaction to the changing environment, and carrying our service in accordance with those priorities, without creating new schedules (this is known as dynamic sequencing).

* AGH University of Science and Technology, Krakow, Poland

Another approach consists in the creation of original schedules, followed by changes implemented in them in view of interference that occurs during the process execution. This method is known as predictive-reactive sequencing. The method of change introduction to sequencing is determined by the accepted rescheduling methods (Van de Vonder *et al.* 2007). That can be effected by: delaying the commencement of those tasks which, as a result of other factors, cannot be carried out according to the schedule (right-hand rescheduling); developing a new schedule, taking into account the directly or indirectly disrupted tasks only (partial rescheduling); solving a new sequencing problem, taking into account all the planned tasks and disruptions (complete regeneration).

The reactive rescheduling tactical solutions (cf. Chong *et al.* 2003, Dorn 2001, Herroelen *et al.* 2004, Kizilisik 1999, Szelke *et al.* 1994, Tanimizu *et al.* 2006, Thomas 2000, Viera *et al.* 2000 and Viera *et al.* 2003) determine the frequency with which changes should be implemented in the original sequencing. Creation of a new schedule can be executed either in predetermined periods (periodic rescheduling) or as a consequence of occurrence of a particular factor (follow-up rescheduling). Also, combined tactics is allowed where a new sequencing is created every particular period and additionally each time when essential disturbing factors appear.

The environment changeability problem exists not only in production systems, but also in broadly understood service systems. The reactive approach can be adopted to solving the problems that are not directly associated with production, for example the ones related to airplane arrival and departure handling in airports. The clients of such a system are the airplanes whose safe landings and take-offs constitute the tasks that are carried out on the runways fulfilling the processor functions. A planned flight schedule is the original airport schedule.

The rescheduling strategy described in this paper is a predictive-reactive strategy. A secondary schedule is developed as a reaction to the occurrence of certain factors called rescheduling factors (Viera *et al.* 2000 and Viera *et al.* 2003); therefore, the applied reactive rescheduling tactics is follow-up rescheduling. In the arrival and departure handling systems on airports, the rescheduling factors are delays in reporting of the tasks to the system, periodic unavailability of runways (processors), occurrence of extra-schedule tasks, prolongation of task execution etc.

Due to high problem calculation complexity, it may prove impossible to take into account all the tasks during the development of new sequencing. For that reason, what is essentially important is a proper determination of a subset of planned tasks subjected to rescheduling (partial rescheduling method). When applying a reactive approach to solving other problems than the originally assumed production systems, one should additionally take into account special characteristics of the system under consideration.

In the case of airports, such characteristics include cycles (particular events taking place periodically) and priorities (the landing airplanes have a higher priority than the ones ready to take off) (van de Vonder *et al.* 2007). In addition, in the airport service systems, the most important criterion is safety, which may be taken into account by the creation of what is known as original resistant schedules (van de Vonder *et al.* 2007).

2. Proposed rescheduling solution

In this paper, we will present an optimisation model based on a classical task sequencing theory for parallel machines. This model can be applied to the development of secondary schedules in reaction to various types of disruptions occurring in the multi-runway airport service process performance. This model takes into account those consequences of disruptions and changes only which concern the airport in question. We have introduced the following symbols here.

Sets:
Zd – set of all the tasks which should be performed at the airport according to the plan,
P – set of the runways available at the airport.

Parameters:
m – number of planned tasks (number of elements in the set Zd),
n – disrupted task number,
D – actual moment of task disruption occurrence,
v_i – planned event moment $i(i \in Zd)$,
x_{ij} – preliminary assignment of tasks to runways ($x_{ij} = 1$, when the task i should take place on the runway j according to schedule, otherwise $x_{ij} = 0$) ($i \in Zd;\ j \in P$),
p_i – service time of the i-th event ($i \in Zd$),
K_i – a critical moment of commencing the i-th event ($i \in Zd$),
β – unit penalty for runway change,
$A,\ B$ – adequately large numbers.

Variables:
y_{ij} – actual moment of handling the task i on the runway $j(i \in Zd,\ j \in P)$,
G_i – actual moment of the event $i(i \in Zd)$,
w_{ij} = 1, when the task i is assigned to the runway j (otherwise, $w_{ij} = 0$),
z_{klj} = 1, when the task k precedes the task l on the runway j; when the task l precedes the task k on the runway j, the variable $z_{klj} = 0$.

Mathematical model

Objective Function f:

$$\min f = \sum\nolimits_{i \in Zd} (g_i - v_i) + \sum\nolimits_{i \in Zd} \sum\nolimits_{j \in P} (x_{ij} * (1 - w_{ij}) + (1 - x_{ij}) * w_{ij}) * \beta \quad (1)$$

Constraints:

$$\forall i \in Zd : g_i \geqslant v_i \quad (2)$$

$$\forall i \in Zd\, \forall j \in P : y_{ij} \leqslant K_i * w_{ij} \quad (3)$$

$$\forall i \in Zd : \sum\nolimits_{j \in P} w_{ij} = 1 \quad (4)$$

$$\forall i \in Zd : \sum\nolimits_{j \in P} y_{ij} = g_i \quad (5)$$

$$\forall j \in P : z_{klj} + z_{lkj} = 1; k = 1, \ldots, m;\; l = 1, \ldots, m; k \neq l \tag{6}$$

$$\forall j \in P : y_{kj} + p_{kj} \leqslant y_{lj} + (2 - w_{kj} - w_{lj}) * B + A * (1 - z_{klj}); \quad k = 1, \ldots, m; l = 1, \ldots, m;\; k \neq l \tag{7}$$

$$\forall j \in P : y_{lj} + p_l \leqslant y_{kj} + (2 - w_{kj} - w_{lj}) * B + A * z_{klj}; \quad k = 1, \ldots, m; l = 1, \ldots, m; k \neq l \tag{8}$$

$$g_n = D \tag{9}$$

$$\forall i \in Zd\forall j \in P : y_{lj} \geqslant 0, \;\; \text{is an integer} \tag{10}$$

$$\forall i \in Zd : g_i \geqslant 0, \; \text{is an integer} \tag{11}$$

$$\forall i \in Zd\, \forall j \in P : w_{ij} \in \{0, 1\} \tag{12}$$

$$\forall j \in P : z_{klj} \in \{0, 1\}; k = 1, \ldots, m; l = 1, \ldots, m; k \neq l \tag{13}$$

Model Description

The goal of our optimisation task is to minimise the changes introduced to the original schedule, occurring as a result of the operation of the rescheduling factors. The first component of the objective function (1) corresponds to the total delay value of all the planned tasks, i.e. the differences between the actual and the planned moments of particular task occurrence.

The second component introduces a system "penalty" for a change of a runway, and it is active exclusively in the event of the values of the assignment of the task i to the runway j (x_{ij}) and of the assignment variable (w_{ij}) being different. That solution provides a protection against implementation of unnecessary changes to the original sequencing. The accuracy of determining the event timing is 1 minute, and consequently, the values of respective variables are integers. The value of the parameter β should depend on the sizes of the problem under consideration and selected in such a way that the total value of the second part of the objective function is lower than 1. That assumption admits the following interpretation of the obtained objective function value: the integer part means the total delay value and the fraction part is equivalent to the penalty for the change in the assignment of the tasks to the runways. The determination of the allowed range of the parameter β values is carried out in the following equation (14):

$$0 \leqslant C * \beta < 1, \tag{14}$$

where C means the maximum number of possible task-to-runway assignment parameter changes. In an extreme case, C corresponds to the number of the assignment parameters (x_{ij}), and, therefore, the product of the number of the runways j times the number of the tasks under consideration. The situation in which $\beta = 0$

means the cancellation of the penalty for the task-to-runway assignment change. For the problem to be considered below, i.e. an airport with three runways and taking into account seven tasks at most, the allowed parameter β value is in the range $[0; 0.047)$. The objective function constructed in that way causes development of secondary schedules, with minimum total delays, and, when there is more than one solution possible, the sequencing containing a lower number of task-to-runway assignments is selected.

Constraint (2) assures that the actual moment of handling the i-th task can occur not earlier than it is specified in the original schedule. According to Constraint (3), actual moment of starting to handle each of the tasks may not exceed a certain critical value which is related to the safety conditions. That interdependence additionally assures zeroing (resetting) of the value of the variable y_{ij} for the runways j, to which the task i is not assigned. Constraint (4) forces each task to be assigned to one runway exactly. Constraint (5) determines the interdependence between the decision variables: y_{ij} which takes into account the task-to-runway assignment, and g_i which is the moment of starting to handle a single task.

The introduction of separate variables that store the above-specified data makes it easy to analyse the results obtained, on the one hand, and to get a simple calculation of the final objective function value, on the other. Constraints (6), (7) and (8) represent the sequential dependencies between particular task pairs. Dichotomous Constraints (7) and (8) prevent the situation in which more than one task is handled on the same runway at the same time.

For a task pair (k, l) in which each task is assigned to a different runway j, the constraints are not active. When two events are taking place on the same runway, only one of the constraints is activated (Constraint (7) when the event k precedes the task l, or Constraint (8), when the task l precedes the task k). The activation of the proper (exactly one) constraint is assured by equation (6) determining that, for each pair of the tasks (k, l), one of the tasks precedes the other. Constraint (9) is responsible for the implementation of disruption to the system and it determines the actual moment of starting to handle the task disturbed.

Should the need be, we can replace "=" with "?" in equation (9), which will practically cause the introduction of a delay of the moment of task readiness for handling. The constraints related to the decision variables force non-negativity of the event handling start times: y_{ij} and g_i, and the requirement that they should be integer result in an approximation introduced in order to attain a better transparency of the results obtained. The remaining variables z_{klj}, w_{ij} are binary.

3. Examples of optimum solutions

The model's operation in practice will be presented in exemplary sets of input data which correspond to various real situations. In the first example, the time range under consideration will contain, according to the flight schedule, seven planned events (m = 7) which should be handled on the airport with three runways ($P = \{a, b, c\}$). The input parameter values are as follows: the planned times of particular task handling: $v_1 = 5$, $v_2 = 20$, $v_3 = 40$, $v_4 = 10$, $v_5 = 35$, $v_6 = 15$ and $v_7 = 30$, the handling time (p_i) of each event equals 10 minute, and the critical moments of the handing start (K_i) are equivalent to 70. The Tasks 1, 2 and 3 should be carried out on the runway a, the Tasks 4 and 5 on the runway b, and the Tasks 6 and 7 on the runway c. Therefore, $x_{11} = x_{21} = x_{31} = x_{42} = x_{52} = x_{63} = x_{73} = 1$, and the remaining values of the parameter x_{ij} are equal to zero. The original schedule is presented in Figure 1. The occurrence of time buffers between the planned events on particular runways allows us to conclude that sequencing is resistant to disruptions several minutes long (not longer than 5 minutes each because this is the value of the shortest buffer).

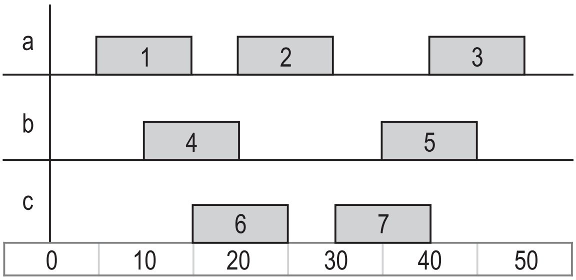

Fig. 1. *Original schedule for Example 1*

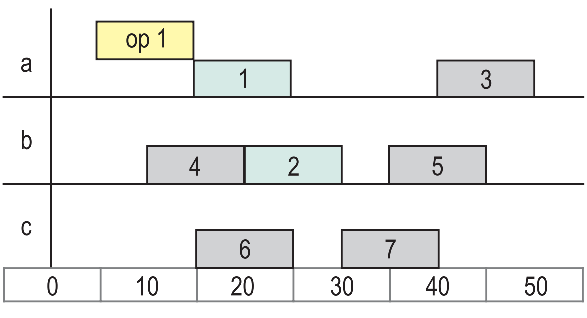

Fig. 2. *Secondary schedule for Case 1*

Let us consider the situation where the first task is late by 10 minutes ($g_1 = 15$). The solutions of Models (1) – (13) allow us to sequence the tasks as shown in Figure 2.

The handling delay time of Task 1 caused that, when Task 2 was reported to the system on Runway a, handling of the previous event still continued. Therefore, Task 2 handling would not start according to the plan. However, since Runway b remained available at the same time, it was possible to carry out Task 2 on a timely basis. The objective function value was 10.02 of which 10 describes the preliminary delay of Task 1 performance, while 0.02 represents the cost of the runway change for Task 2. The remaining events would be carried out in accordance with the preliminary assumptions.

By introducing the same disruption to a slightly different original schedule presented in Figure 3, we will obtain the secondary sequencing shown in Figure 4.

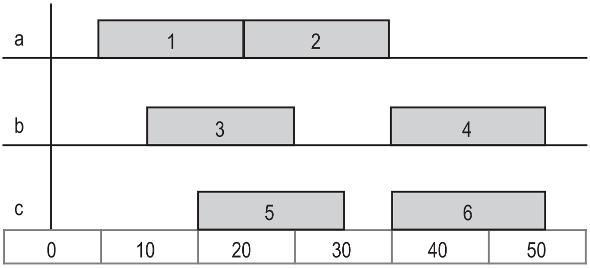

Fig. 3. *Original schedule for Example 2*

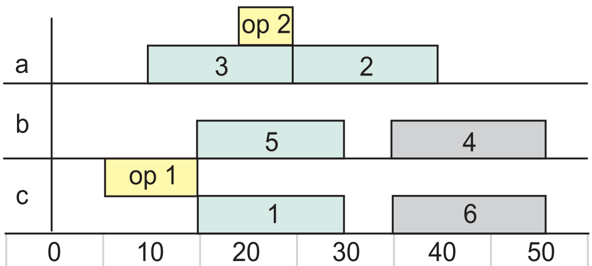

Fig. 4. *Secondary schedule for Case 2*

With reference to the previous case, the handling times of particular tasks have been extended (to 15 minutes), with the reduction of the number of changes in runway assignment. The new values of the planned times of the events v_i are 5, 25, 10, 35, 15 and 35, respectively. The values of the parameters K_i remain unchanged.

Due to the handling time prolongation, the time buffers between particular operations have been shortened, and, consequently, the disruption effects are larger. Tasks 4 and 6 are carried out according to the plan, Tasks 3 and 5 are handled on time, but on runways different from those originally planned, Task 1, after a late reporting to the system, is transferred to Runway c, and Task 2, after a timely reporting to the system must wait for the completion of Task 3 handling. The total delay value in this case amounts to 15 minutes of which 10 minutes correspond to the delay of Task 1 and 5 minutes to Task 2 waiting for handling.

By additionally implementing a critical moment of starting to handle Task 2 at Level 21 ($K_2 = 21$), or by forcing the timely performance of that task, we will obtain a different secondary schedule (Fig. 5).

According to new sequencing, Events 2, 3 and 6 will take place according to the plan, Task 4 will be received on a runway different from the originally planned one, Task 1 will be carried out directly after a late event report, and Task 5 must wait for runway availability. That will produce the total delay value of 20 minutes which will require a 5 minute increase in the total delay value, when compared with the previous case, or deterioration of the objective function value caused by criteria tightening.

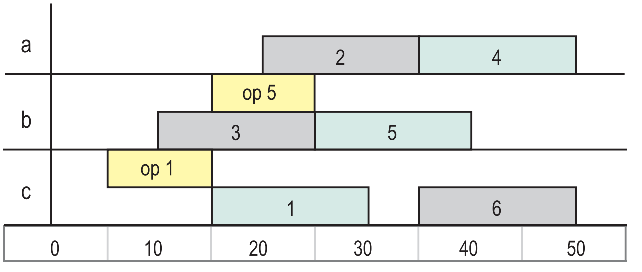

Fig. 5. *Secondary schedule for Case 3*

In the following two examples, the disruptions will consist in the occurrence of a blockade of one of the take-off runways, e.g. as a result of a failure which prevents event handling on that runway, within the time range under consideration. The influence of such a disruption on the original sequencing is presented in the secondary schedule in Figure 6.

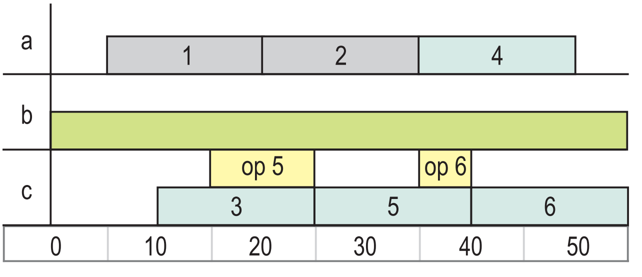

Fig. 6. *Secondary schedule for Case 4*

Unavailability of Runway *b* forces the execution of the tasks assigned to that runway on the remaining runways. Events 1 and 2 are carried out according to the plan, Events 3 and 4, the ones which were originally assigned to the unavailable runway, are carried out according to the plan but on the neighbouring runways, while Tasks 5 and 6 must wait for the availability of a runway. Owing to the use of the buffers which were placed in the original schedule, the total delay value amounts to 15 minutes only. However, we need to point out that the obtained secondary sequencing does not have any time reserves between the operations, which means that our schedule is not resistant to a disruption, even the slightest one. Each subsequent disruption (except for those related to Tasks 4 and 6 after which no planned events occur) will cause further delays.

When this case is supplemented by a critical moment of Event 5, e.g. $K_5 = 20$, we will obtain a different secondary sequencing presented in Figure 7.

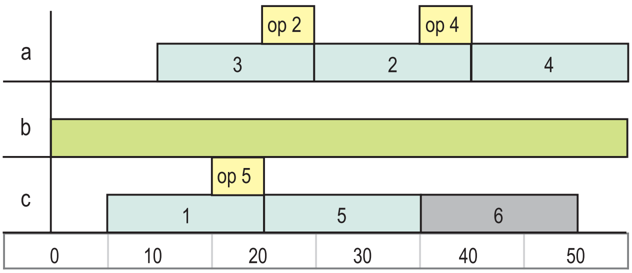

Fig. 7. *Secondary schedule for Case 5*

The total value of delays amounts to 15 minutes for each of these two cases, but the number of changes is lower in the latter case (Task 6 only takes place according to the plan), and, as a result of the imposition of penalties for the changes, the objective function value is worse as well.

The examples presented above have demonstrated the model's operational effectiveness in solving small-size problems. Actual systems, however, are characterised by much larger complexity; consequently, the calculation of optimum secondary schedules is very time consuming or even next to impossible to perform. Such situations require the application of the heuristics enabling results to be obtained quicker, even at the cost of losing solution optimality.

4. Heuristic rescheduling procedures

Proposed procedures consist in breaking up a large task into interrelated non-complex subtasks which are easier to solve. The tasks should be numbered in line with the sequence of the planned task handling times.

Heuristics 1

Step 0. Determination of the original task division points, or the times in which no task is performed according to the plan: This heuristics is applied in particular cases only, when separating points occur in the flight schedule being handled. In addition, the number of the events between neighbouring points may not be larger than the size of the task which can be solved within acceptable time.

Step 1. Disturbed event identification: Determination of which task is delayed and how long the delay is.

Step 2. Determination of subtasks and their performance sequences: The created subtasks should have the sizes that allow for prompt task handling. The subtask to be solved with priority should contain disturbing events and all the tasks whose planned handling start times are located between the nearest division point preceding the actual moment of the disturbed event occurrence and the respective actual task. In addition, when the number of events in the subtask determined in that way is small, one can also take into account the tasks which, according to the plan, follow the actual moment of the disturbed event performance.

Step 3. Subtask solution and checking the admissibility of the result obtained in the whole schedule: If the newly created sequence does not contain any collisions consisting in the assignment of the busy runways to other tasks, the obtained result will form the final secondary sequencing. In the event that the start of task performance in the schedule occurs during the performance of another task, that occurrence should be treated as an input disturbance: a delay in the starting task, followed by a solution of the subsequent subtask. Step 3 must be repeated until an admissible solution is obtained.

If the preliminary schedule is resistant, the number of iterations will be small, while, in the case of extremely non-resistant sequences, we may never obtain an admissible solution. The obtained result is an approximate one. However, in the systems which take into account such safety criteria as ground handling on the airport, prompt obtaining of an admissible solution is of essence, and the whole solution is considered only afterwards.

Heuristics 2

Step 0. To Models (1) ÷ (13), implement an additional constraint (equation (15)) of the earliest availability of particular runways, and the related parameter f_j.

$$\forall i \in Z_d \; \forall j \in P : y_{ij} \leqslant f_j * w_{ij} \tag{15}$$

Constraint (15) ensures that the events considered in the first subtask are not assigned to the runways on which previous tasks are being handled.

Step 1. Disturbed event identification: Determination of which task is delayed and how long the delay is.

Step 2. Determination of subtasks and their performance sequences: The subtasks solved as priorities should contain disturbed tasks and several other tasks whose planned start times occur after the actual start time of the disturbed task. In addition, if the performance of previous tasks continues on all the runways at the actual start time of the disturbed event, it is necessary to take into account the event whose performance started last. The number of events in a subtask should allow for a prompt task solution. It is also necessary to introduce the values of the parameters f_j, assuming, based on the original schedule, the moments of finishing of the latest events that were not considered in the first subtask.

Step 3. Subtask solution: This step is identical to Step 3 in Heuristics 1.

The operation of the above heuristics will be illustrated with the example in which the original schedule has the form shown in Figure 8.

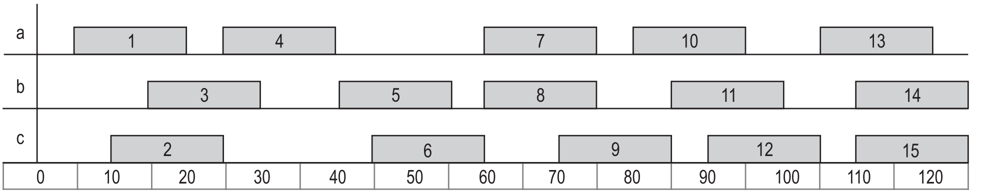

Fig. 8. *Original schedule for 15 tasks*

Proceeding according to the first heuristics, we first determine the division points in the original schedule: $t_1 = 40$, $t_2 = 60$, and $t_3 = 105$. After the notification that Task 5 is delayed by 25 minutes, we select the events requiring the development of the secondary schedule. First, we consider Task 5, the disturbed one, then Tasks 7 and 8, whose handling will start after the time $t_2 = 60$, and Tasks 9, 10 and 11 which are close to the disturbed task.

Solving such a limited problem determines the partial secondary schedule shown in Figure 9. The sequencing analysis of all the tasks presented in Figure 10 allows us to conclude that the obtained solution is admissible and it forms a secondary sequencing for the whole period under consideration.

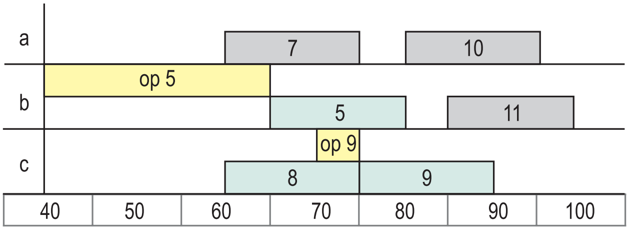

Fig. 9. *Partial secondary schedule for the case analysed (Heuristics 1)*

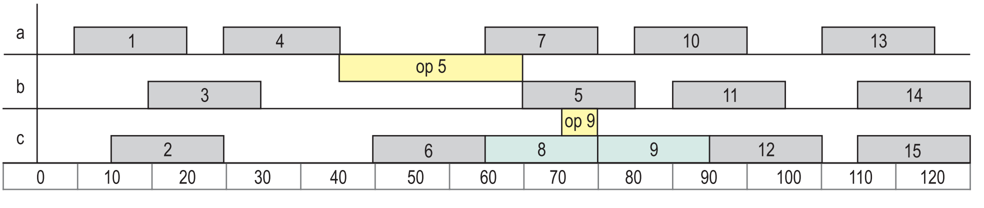

Fig. 10. *Secondary sequencing for the case analysed (Heuristics 1)*

The total value of delays is 30 minutes (of which 25 minutes is a nominal delay of Task 5, while during the remaining 5 minutes, Task 9 is waiting for runway availability).

Proceeding according to the second heuristics, we take into account the following tasks: Task 5 – the disturbed task, and Tasks 9, 10, 11, 12 and 13. We introduce the values of the parameters f_j which for the runways a, b and c take the values: 75, 75 and 60, respectively. Solving such a limited subtask leads to obtaining the partial secondary schedule presented in Figure 11. The full schedule being a secondary sequencing is presented in Figure 12.

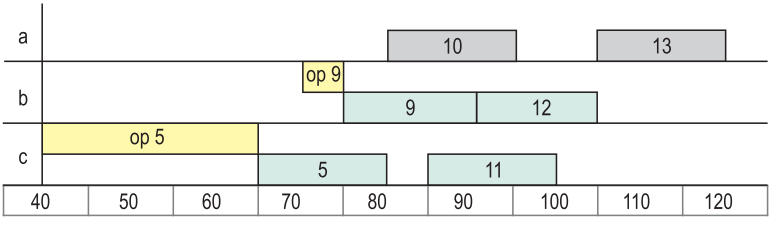

Fig. 11. *Partial secondary sequencing for the case analysed (Heuristics 2)*

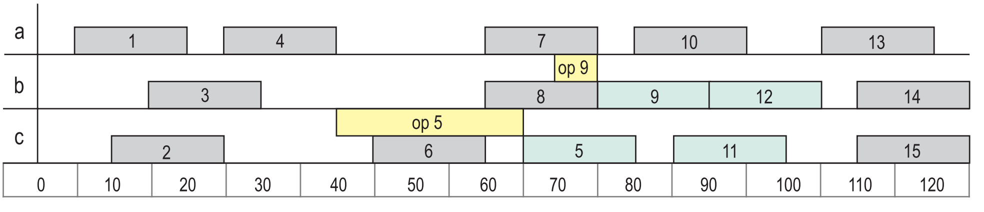

Fig. 12. *Secondary sequencing for the case analysed (Heuristics 2)*

The total value of delays is 30 minutes, of which 25 minutes is a nominal delay of Task 5, while during the remaining 5 minutes, Task 9 is waiting for runway availability.

When we compare the results obtained by the application of both heuristics, we can come to the conclusion that the total delay values are identical; however, by applying the second method, we had to introduce more changes in assigning the tasks to the runways. Heuristics 1 is designed to handle to changes in the tasks that precede the disturbed task start, although the usability of Heuristics 1 is fairly limited due to the necessity of determining the separation points. That heuristics is useful for the systems with lower demands and larger resistance to disturbance.

Heuristics 2 is designed to handle the tasks which occur after disturbance and it allows for an analysis of a broader time horizon; however, preventing introducing changes in previous tasks. It is easy to determine subsequent subtasks so that they are easy to solve. Heuristics 2 may be useful for the systems with large demands and small resistance to disturbance.

5. Conclusions

The problem of environment changeability is common and it concerns the majority of production and service systems. The development of effective tools using the reactive approach allows for quicker and better reactions to disturbance occurring during the production or service process performance. However, controlling a large system, e.g. a system of flight arrival and departure on an airport, in real time requires large computational capabilities. The safety criterion, which is essential in airport handling services, forces quick reaction to disturbance, even at the cost of rejecting optimum solutions. The proposed problem solutions presented in this paper demonstrate how to create, in a relatively simple manner, the tools which effectively support the control process. Subsequent development of computer tools supporting airport handling may support the creation of gradually more effective systems of air traffic coordination among numerous airports.

References

Chong C.S., Sivakumar A.I., Gay R. 2003. *Simulated-based scheduling for dynamic discrete manufacturing*. Proceedings of the 2003 Winter Simulation Conference, pp. 1465–1473.

Dorn J. 2001. *Evaluating Reactive Scheduling Systems*. Vienna University of Technology.

Herroelen W., Leus. R. 2004. *Robust and reactive project scheduling: a review and classification of procedures*. International Journal of Production Research, 42(8).

Kizilisik Ö.B. 1999. *Predictive and Reactive Scheduling*. IE 672, Theory of Machine Scheduling.

Szelke E., Kerr R. 1994. *Knowledge-based reactive scheduling*. Production Planning and Control, 5(2), pp. 124–145.

Tanimizu Y., Sakaguchi T., Iwamura K., Sugimura N. 2006. *Evolutional reactive scheduling for agile manufacturing systems*. International Journal of Production Research, Vol. 44, No. 18–19, pp. 3727–3742.

Thomas D.W. 2000. *Integration Strategies for the Reactive Scheduling System*. MSI Research Institute, Loughborough University.

Vonder S. van de, Ballestin F., Demeulemeester E., Herroelen W. 2007. *Heuristic procedures for reactive project scheduling*. Computers and Industrial Engineering, Vol. 52(1), pp. 11–28.

Vieira G.E., Herrmann J.W., Lin E. 2000. *Predicting the Performance of Rescheduling Strategies for Parallel Machine Systems*. Journal of Manufacturing Systems, 19(4), pp. 256–265.

Vieira G.E., Herrmann J.W., Lin E. 2003. *Rescheduling manufacturing systems: a framework of strategies, policies, and methods*. Journal of Scheduling, 6(1).