RESEARCH IN LOGISTICS & PRODUCTION

OPTIMIZATION OF SUGAR DISPATCHING PROCESS

Radosław Pytlak* and Wojciech Stecz**

* Institute of Automatic Control and Robotics, Warsaw Univesrity of Technology, ul. Sw. Andrzeja Boboli 8, 02-525 Warsaw, Poland, r.pytlak@mchtr.pw.edu.pl
** Faculty of Cybernetics, Military University of Technology, ul. Kaliskiego 2, Warsaw, Poland, wojciech.stecz@wat.edu.pl

Abstract We consider the sugar dispatching process at a sugar mill. The main goal of our work was to check an efficiency of the logistics system in the mill, find and correct the bottlenecks. Some methods for logistics processes optimization are presented. We base on the heuristics and CLP techniques for solving the scheduling problems. Some additional remarks about possibility of using the optimization methods in scheduling and logistics optimizations are presented too.

Paper type: Research Paper

Published online: 30 April 2014
Vol. 4, No. 2, pp. 105-118

ISSN 2083-4942 (Print)
ISSN 2083-4950 (Online)

Keywords: *job shop scheduling, optimization in logistics, CLP in scheduling, constraint programming*

1. INTRODUCTION

We consider the sugar dispatching process at a sugar mill. The problem we consider arose when some sugar mill decided to increase twofold its sugar production and there were concerns that its current dispatching facilities would not handle the extra amount of sugar. They decided to tackle the problem by solving optimization problem related to the sugar dispatching process. By solving the optimization problem we could answer two questions: 1) how many dispatching trucks could be handled, during some specified period of time, by using the current facilities; 2) how the dispatching process could be re-organized in order to handle the extra amount of sugar. It was assumed that the sugar mill would use a professional tool for scheduling tasks associated with the dispatching process. The scheduling problem associated with the sugar dispatching process can be described as follows. We have some amount of sugar which we have to dispatch from a sugar mill. We know that the sugar can be dispatched by cisterns (without sugar packing) -in that case sugar is sent to a large industrial client, or by trucks-in that case sugar must first go through the packing process. Therefore, we have many tasks, each of them has precisely described associated jobs, and scheduling of these tasks must answer the question what is the minimal aggregated time during which all these tasks can be executed. If the aggregated time obtained by solving the scheduling problem is too big one can think of upgrading sugar mill facilities in such a way that the new aggregated time obtained through solving the new scheduling problem (in which parameters of the dispatching process are different due to upgraded facilities) could meet the time target.

By solving the scheduling problem we also find how many trucks and cisterns can be loaded during 24 hours at every day of a week. Furthermore, the scheduling algorithms indicates the bottlenecks of the dispatching process. Eliminating these bottlenecks is the way of upgrading sugar mill facilities with the aim of meeting sugar delivery times. Optimization process goal described in this article was also linked with a need of improving effectiveness of the packing lines (and the whole dispatching system). Sugar dispatching process is rather complicated not only because of scheduling the tasks of the packing lines where sugar is packed in the 0.5 kg, 1 kg, 2 kg, or bigger bags. The packing job is followed by a testing job carried out by the sugar mill laboratory. Every batch of sugar has to be tested. Furthermore, when sugar is loaded into cistern it has to be tested once again by the laboratory. Due to the limited number of laboratory workers and testing devices the dispatching process by cisterns can be significantly disrupted. Additionally we have to take into account the parking place for cisterns and trucks waiting for loading. We have to bear in mind that each truck driver has precisely planned working hours which cannot be exceeded. When these hours are exceeded he must rest for some hours. When the delay in the loading process is greater than 2 hours the driver must stop driving after travelling for no more than 300 km. It implies that each truck must be loaded within 1 hour.

All these problems show that the loading process is a highly complicated task. To improve the loading system we need to model the loading process and schedule some possible variants – test cases – for the assumed number of trucks and cisterns.

The sales department of the sugar mill provided data for the reference case. The information concerning the number of trucks and cisterns which must be served during the next 30, or 45 days was available. Under the current sugar mill practice the specified number of cisterns and trucks must be server during a week. If, for some reasons, some delays occurred during the working days they must have been tackled during the coming weekend. That requirement must have been taking into account in the scheduling process. Moreover, any delay in sugar delivery was linked with penalty fee put on the sugar mill and serious delays could lead to the contract cancellation.

2. SUGAR DISPATCHING AS OPTIMIZATION PROBLEM

2.1. Parameters and variables

As described in the previous case the scheduling problem we consider is related to several tasks and each of them is related to a different product. For example, we have in the scheduling problem tasks associated with the packed sugar sent to warehouses (packed on pallets), sugar powder that is sent in cisterns to industrial counterparties, or tasks associated with some additional products – by-products made during production processes such as molasses.

A loading task consists of several operations, among which are weighing at entrance and exiting gates (with the control of the dispatching documents), loading a cistern with sugar or sugar powder from silos (in the case when sugar is dispatched without packing), packing sugar in the warehouses (sugar packed on pallets), or control tests at laboratories. Each of these operations requires assignment of the execution time. These times can be obtained from the ERP information system, or can be determined during audit of the loading processes.

Each operation defined as j is assigned to the task i. We denote the set of all tasks as I and the set of operation belonging to the task i as J^i. Each operation is described by parameters which play important role in scheduling (these parameters decide on possible algorithms one can use for scheduling (Pinedo, 2001)).

These parameters are: execution time p_{ij} of operation j of task i, time window in which the operation should be scheduled $[r_{ij}, d_{ij}]$ (r_{ij} - release time, minimal time when the operation can be started; d_{ij} - due time, maximal time at which the operation should be finished to avoid consequences such as penalty fees). We have to underline that time windows play crucial role in the scheduling.

In our work we assume penalty fees but exclude the posibility of discarding the commodity sent with a delay (that case is very rare and the clients rather give up a contract with the sugar mill than discard the commodity once sent). Additionaly after finding the feasible schedule each operation with execution time p_{ij} has got its start time and the end time defined as s_{ij}, e_{ij}.

The described problem of scheduling tasks at sugar mill isn't different from other scheduling problems at food manufacturing plants, where dispatching process involves not only packing but also testing steps. For example in the case of pharmaceutical companies one has to take into account the GMP regulations. In the case of food manufacturers one has to deal with the HACCP regulations.

2.2. Optimization function

The main objective in our scheduling problem is to minimize the time truck drivers spent at the sugar mill. This means that we have to speed up the process of loading the sugar into the trucks and the process of laboratory tests and dispatching documents preparations. When a truck crosses a plant gate the truck the loading starts and it can't be stopped according to the law regulation. We attempt at minimizing the WIP (Work in Progress) which means that we try to complete all necessary tasks as soon as possible in order to send the truck to a client in the shortest possible time (the more detailed description of WIP is given in Section 2.4). It means that our objective function is:

$$F = \sum_{i \in I} \left\{ \left(e_{ik} - s_{i1} \right) \right\} \tag{1}$$

e_{ik} - finish time of the last operation of i-th task

s_{i1} - start time of the first operation of the i-th task.

At this point we must stress that for some scheduling algorithms the above objective function is not adequate. For example, if one uses scheduling algorithms from the *nondely* class (Brucker, 2007) (Brucker, 2006) (Grabowski, Nowicki & Smutnicki, 2003) (Pinedo, 2001) he can get results such that most of the operations of the task start as soon as possible but tasks last too long and the work in progress is unacceptable. To avoid these cases some data preprocessing must be made. That preprocessing reduces to equal distribution of trucks appearance during the work day. However, such reformulation of the model is not enough to avoid the problem mentioned above – the objective function has to be changed also. The objective function modification is done by adding to it the penalty term. The function with the penalty component is as follows:

$$F = \sum_{i \in I} \{(e_{ik} - s_{i1}) + w_i \cdot \max(0, e_{ik} - d_{ik})|\} \quad (2)$$

d_{ik} - the due time corresponding to the last operation of the i-th task.

2.3. Sugar dispatching model

The system used in the sugar mill for sugar dispatching described earlier can be modeled as a triple (we assume that the plant sends all its production to the external client's warehouses):

$$(\alpha, \beta, \gamma) \quad (3)$$

α - machine environment (in our case it is a job shop system)

β - tasks and the operation parameters (for a job shop system)

γ - cost function (we minimize WIP).

The systems model forms the standard job shop system, where in each service point there are some servicing machines (in our case these machines are: packing machines, scales, laboratory testing devices etc.). For the convenience of analysis we do not consider additional resources needed for scheduling. We assume that the service lines are supported by the proper staff if that is needed. We do that simplification in the case when the efficiency of the system is tested. In general case, we can assume that additional resources needed for scheduling are provided.

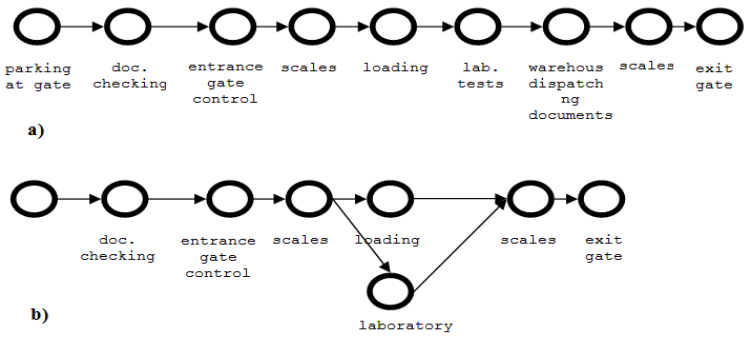

Fig. 1 Job shop system in sugar mill. Tasks (loading truck and cistern) consisting of the operations are presented

The base tasks and operation parameters used to model the problem are:

- the time windows of each operation (and the time window of the whole task – it depends on the optimization function taken) which are modeled as a pair $[r_{ik}, d_{ik}]$

- order dependencies among operations i and k depicted as $i \to k$ (it means that operation i has to be executed before the operation k).

In the case when penalty fees are levied (under the condition that sugar dispatch is delayed) the parameters depicted as w_i or w_{ij} (depending on whether delay for the task as a whole is taking into account, or for all or its operations) have to be introduced. When an optimal (for a job shop system usually rather suboptimal) schedule is found one checks, for each task in that schedule, the values of the variables:

s_{ij} - start time of the operation j which belongs to the task i

e_{ij} - finish time of the operation j which belongs to the task i .

In the case of sugar dispatching proces the order dependencies are shown in **Błąd! Nie można odnaleźć źródła odwołania.**.

2.4. Heuristic methods in tasks scheduling

The most commonly used techniques for scheduling are the heuristics methods: for example the shifting bottleneck algorithm (Pinedo, 2001), (Monch & Driefel, 2005). Another popular technique, the optimal block method (Brucker, 2006), isn't effective when one has to schedule more than 10 tasks consisting of more than 10 operations.

The complete description of heuristics methods used for scheduling cannot be presented in the paper due to the lack of space, so only important remarks related to these techniques are outlined. First of all, while considering a particular technique, one has to know whether a particular scheduling problem with a job shop system and a given objective function can be tackled by the considered technique. In the case of our scheduling problem we must look at techniques which are appropriate for job shop systems with the objective function defined by the WIP. The precise definition of the WIP function is given below.

Assume that material that has entered the production process but is not yet a finished product. Work in Progress (WIP) refers to all materials and partly finished products that are at various stages of the production process. WIP excludes inventory of raw materials at the start of the production cycle and finished products inventory at the end of the production cycle. When using WIP definition for scheduling the dispatching process we can treat the trucks (cisterns) and dispatched sugar as materials till they are present at a sugar mill area.

2.5. CLP methods used for solving scheduling problems

Constraint Logic Programming (CLP) (Fruwirth & Abdennadher, 2003) methods are used for solving complicated optimization tasks when the constraints consist not only of the algebraic equations. CLP methods are mainly applied to combinatirial

problems, scheduling problems, problems from the areas of biotechlonogy, or biology. The most important feature of CLP models is their declarativity – the possibility of modeling the problem without showing the associated control flows. CLP techniques are the lan-guage based on the CSP (Constraint Satisfation Problems) methods. The constraints depend on their domains. The most common used domains are integer and real numbers. The basic searching methods of CLP are based on the constraint propagation and the most common CLP languages are derived from Prolog language (CHIP, Eclipse, SICStus Prolog). Some modeling aspects of CLP are described in (Wallace, 2005), (Wallace & Schimpf, 2002). In this paper we apply mechanisms of CLP of the ILOG constraint programming solver. ILOG uses the OPL language which is the native format.

Below we present an example of the script for scheduling tasks of a job shop system. The first part of the code is a declaration of the variables and parameters. The solver instance initialization is also included in the script.

```
using CP;

int NbTasksTruck  = ...;
range TasksTruck  = 1..NbTasksTruck;
int NbTasksSilo  = ...;
range TasksSilo  = 1..NbTasksSilo;
{string} OperationNamesTruck   = ...;
{string} OperationNamesSilo   = ...;
int DurationTruck [t in OperationNamesTruck] = ...;
int DurationSilo [t in OperationNamesSilo] = ...;
int Workers [t in OperationNamesTruck] = ...;

tuple PrecedenceTruck {
   string pre;
   string post;
};
tuple PrecedenceSilo {
   string pre;
   string post;
};
{PrecedenceTruck} PrecedencesTruck = ...;
{PrecedenceSilo} PrecedencesSilo = ...;

int ReleaseDateTruck[TasksTruck] = ...;
int ReleaseDateSilo[TasksSilo] = ...;

dvar interval itvsTruckWhole[h in TasksTruck]
  in 0..maxint div 4;
dvar interval itvsTruck[h in TasksTruck][t in OperationNamesTruck]
in ReleaseDateTruck[h]..(maxint div 2)-1 size DurationTruck[t];

dvar interval itvsSiloWhole[h in TasksSilo]
  in 0..maxint div 4;
dvar interval itvsSilo[h in TasksSilo] [t in OperationNamesSilo] in
ReleaseDateSilo[h]..(maxint div 2)-1 size DurationSilo[t];
```

```
cumulFunction workersUsage[t in OperationNamesTruck] =
  sum(h in TasksTruck) pulse(itvsTruck[h][t],1);
cumulFunction workersUsageSilo[t in OperationNamesSilo] =
  sum(h in TasksSilo) pulse(itvsSilo[h][t],1);

dexpr int goalFnt1 = max(h in TasksTruck)
  endOf(itvsTruck[h]["Gate Out"]);
dexpr int goalFnt3 = max(h in TasksSilo)
  endOf(itvsSilo[h]["Gate Out"]);

minimize staticLex(goalFnt1,goalFnt3);

subject to {
  forall(h in TasksTruck)
    forall(p in PrecedencesTruck)
    endBeforeStart(itvsTruck[h][p.pre],itvsTruck[h][p.post]);

  forall(h in TasksSilo)
    forall(p in PrecedencesSilo : p.post != "Laboratory")
      endBeforeStart(itvsSilo[h][p.pre],itvsSilo[h][p.post]);

  forall(h in TasksSilo)
    forall(p in PrecedencesSilo : p.post == "Laboratory")
      startAtEnd(itvsSilo[h][p.post],itvsSilo[h][p.pre], 10);

  forall(i,j in OperationNamesTruck : i == j &&
    j in {"From queue","Gate In","Scales In", "Loading",
          "Laboratory","Scales Out","Gate Out"})
    workersUsage[i] + workersUsageSilo[j] <= Workers[i];

  forall(o in OperationNamesTruck : o in {"Warehouse"})
    workersUsage[o] <= Workers[o];

  forall(h in TasksTruck)
    span(itvsTruckWhole[h],
    all(t in OperationNamesTruck) itvsTruck[h][t]);

  forall(h in TasksSilo)
    span(itvsSiloWhole[h],
    all(t in OperationNamesSilo) itvsSilo[h][t]);
}
```

3. OPTIMIZATION RESULTS FOR SCHEDULING SUGAR DISPATCHING PROCESS

3.1. Scheduling with heuristic procedures

The scheduling problem with the minimization of the WIP was accomplished with the Preactor APS scheduling system. Preactor uses the TOC (Theory of Constraints) as a standard technique for its heuristic scheduling procedures. The TOC is very often used for scheduling tasks in IT projects. Other heuristics can be found in (Liverani, 2002), (Pinedo, 2001). Preactor APS features are described in (Preactor, 2005).

Below we show the schedule of sugar dispatching tasks which covers the truck and cistern dispatching processes. The schedule depicted covers horizon of several days and one can see that the sugar mill dispatches sugar 24 hours per day and 7 days per week.

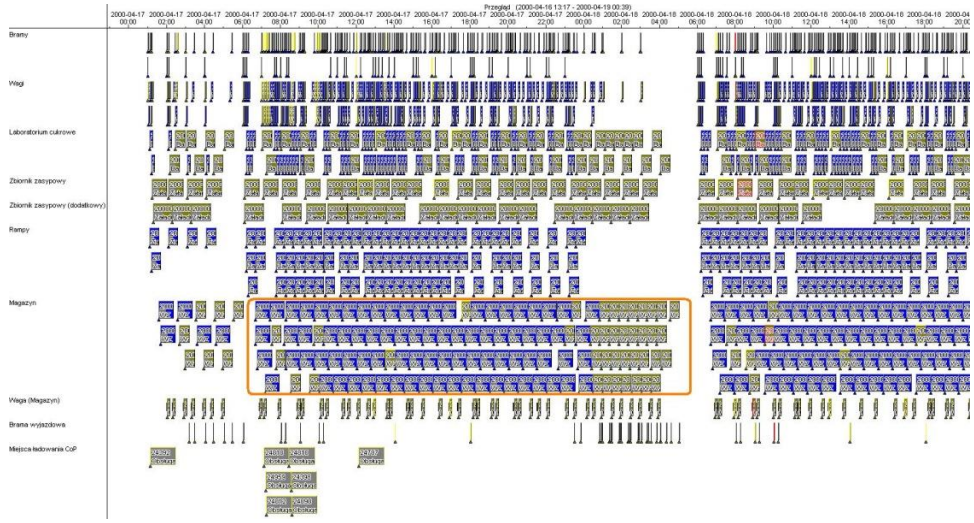

Fig. 1 Result of the tasks scheduling process for a few working days in a sugarplant

The part of the schedule constrained by the red square indicates the highest tasks accumulation – this is a potential bottleneck of the dispatching system. In our case it was a sugar warehouse where the packing processes were conducted and sugar was sent to laboratories for testing. In this warehouse some additional dispatching documents were prepared too. The dispatching process is slowed also by loading the trucks with pallets (loading the truck is complicated when one takes into account the fact that there are some constraints in the pallets packing process, for example the pallets with the sugar powder mustn't cannot be put below the other pallets.

In Figure 3 we show the schedule for the horizon of 3 hours. We indicate the order of the operations execution. One can see the operations workload of a warehouse. What is more important one can also see the workload of the scales. These

devices were the real bottleneck in the tested system because they were used any time the truck was loaded.

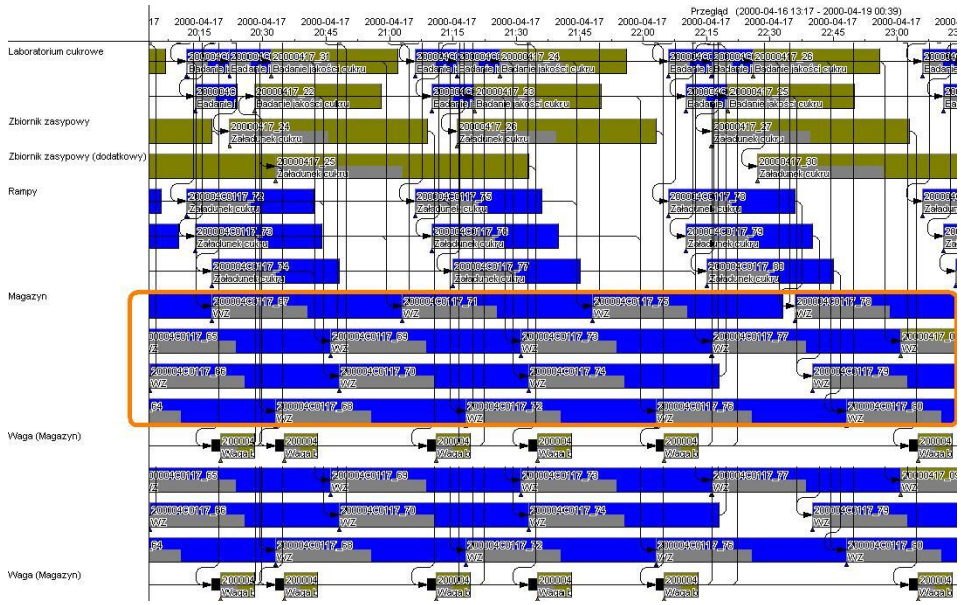

Fig. 2 Result of scheduling for 3 hours – one can see the bottlenecks of the system

The complete result for a week is depicted in Fig. 3 where the Gantt's chart of all the task's operations is shown. One can see that the WIP minimization algorithm forces the system to complete started tasks as quickly as possible and this is the expected result.

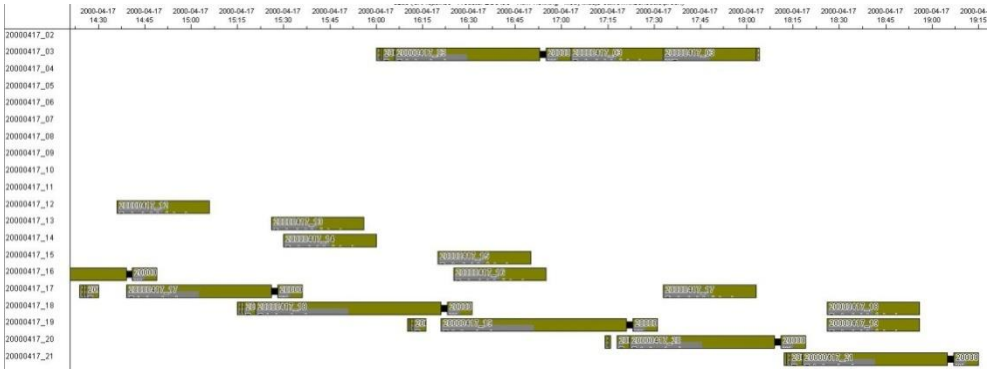

Fig. 3 Schedule for a few tasks – WiP algorithm used

Fig. 4 presents a larger set of scheduled tasks – the schedule for a few days. The point sets are the operations of the tasks completed by the loading system. One can see that there are no tasks with the operations delayed for several hours.

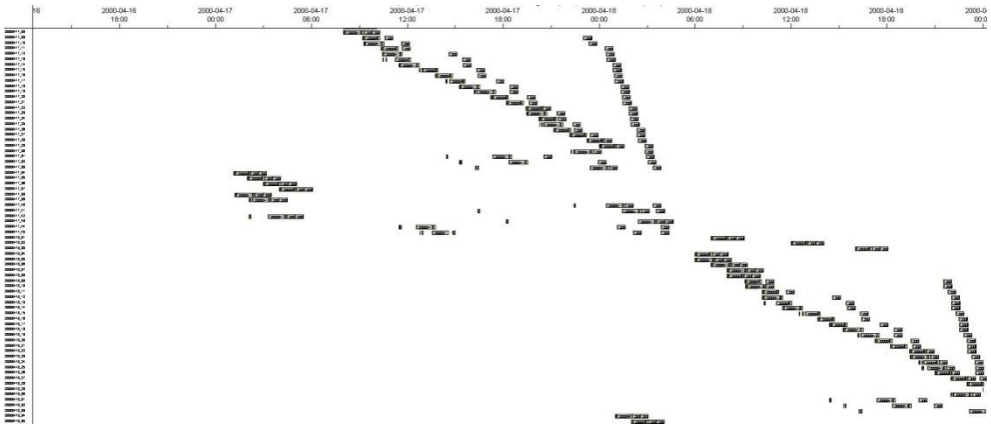

Fig. 4 Result of WiP algorithm for a few hundred tasks

The WIP heuristics allows us to schedule many hundreds of tasks consisting of several operations in a few minutes. What is more important the results presented are very good for the job shop systems for which there is no optimal scheduling algorithm. (A block method (Grabowki, Nowicki & Smutnicki, 2003) is the only optimal but strictly theoretical algorithm). Other methods that can be used for scheduling are constraint logic programming algorithms (CLP) discussed in the previous section.

3.2. Scheduling with CLP methods

The results of CLP algorithms are presented on the problem with the reduced test data – we assume 8 tasks (5 related to the truck loading and 3 to the cistern loading). Test data in ILOG CLP (IBM ILOG,2009) format are described below.

Test data:

```
NbTasksTruck  = 5;
NbTasksSilo   = 3;

OperationNamesTruck = {"From queue","Gate In","Scales
In","Loading","Laboratory","Warehouse","Scales Out",
"Gate Out"};
OperationNamesSilo = {"From queue","Gate In","Scales In", "Load-
ing","Laboratory","Scales Out","Gate Out"};

Workers = [02,02,02,02,02,02,02,02];
DurationTruck = [01,01,04,30,10,45,06,01];
ReleaseDateTruck = [ 1, 0, 9, 12, 9];
```

```
DurationSilo = [01,01,04,120,10,06,01];
ReleaseDateSilo = [ 1, 0, 10];

PrecedencesTruck = {
  <"From queue","Gate In">,
  <"Gate In","Scales In">,
  <"Scales In","Loading">,
  <"Loading","Laboratory">,
  <"Laboratory","Warehouse">,
  <"Warehouse","Scales Out">,
  <"Scales Out","Gate Out">
 };
 PrecedencesSilo = {
  <"From queue","Gate In">,
  <"Gate In","Scales In">,
  <"Scales In","Loading">,
  <"Scales In","Laboratory">,
  <"Loading","Scales Out">,
  <"Laboratory","Scales Out">,
  <"Scales Out","Gate Out">
 };
```

The cost function and scheduling results:

```
// solution with objective 188; 284
```

Scheduling results for the test data:

```
itvsTruckWhole =
[<1 1 143 142> <1 0 98 98> <1 9 128 119> <1 12 188 176>
<1 9 173 164>];

itvsSiloWhole =
[<1 1 253 252> <1 0 284 284> <1 10 174 164>];

itvsTruck = [
[<1 1 2 1> <1 2 3 1> <1 3 7 4> <1 7 37 30> <1 37 47 10>
 <1 91 136 45> <1 136 142 6> <1 142 143 1>]
[<1 0 1 1> <1 1 2 1> <1 2 6 4> <1 6 36 30> <1 36 46 10>
 <1 46 91 45> <1 91 97 6> <1 97 98 1>]
[<1 9 10 1> <1 10 11 1> <1 11 15 4> <1 36 66 30> <1 66 76 10>
 <1 76 121 45> <1 121 127 6> <1 127 128 1>]
[<1 12 13 1> <1 13 14 1> <1 14 18 4> <1 96 126 30>
 <1 126 136 10> <1 136 181 45> <1 181 187 6> <1 187 188 1>]
[<1 9 10 1> <1 10 11 1> <1 18 22 4> <1 66 96 30>
 <1 96 106 10> <1 121 166 45> <1 166 172 6> <1 172 173 1>]];

itvsSilo = [
[<1 1 2 1> <1 2 3 1> <1 7 11 4> <1 126 246 120>
 <1 1 11 10> <1 246 252 6> <1 252 253 1>]
[<1 0 1 1> <1 1 2 1> <1 6 10 4> <1 157 277 120> <1 0 10 10>
 <1 277 283 6> <1 283 284 1>]
[<1 10 11 1> <1 11 12 1> <1 16 20 4> <1 37 157 120>
 <1 10 20 10> <1 157 163 6> <1 173 174 1>]];
```

For the task described by vector variable itvsTruckWhole <1 1 143 142> the first value describes the mandatory task, the last value describes the task processing time, the second value is the start time and the third value is the finish time of the task.

The variable itvsTruckWhole gathers the aggregated values of all the task operations scheduled. The operations of the first task are the following operations of the variable itvsTruck: [<1 1 2 1> <1 2 3 1> <1 3 7 4> <1 7 37 30> <1 37 47 10> <1 91 136 45> <1 136 142 6> <1 142 143 1>].

The start time of the last operation (exiting the sugar mill by the truck through the exit gate) is equal to 142, the operation lasts for 1 time and finishes at time 143.

3. CONCLUSION

Optimization of the sugar mill dispatching process was carried out to test the effectiveness of the sugar dispatching system. Because of the complicated structure of the dispatching system and the size of the scheduling problem it is not possible to find bottlenecks of this system manually. The problem became more important when the dispatching system had to be modified to serve a larger number of the clients. That situation trigged the project described in the paper. The realization of the project helped to identify and eliminate all bottlenecks of the sugar dispatching system. The effectiveness of the system was almost doubled.

REFERENCES

Brucker P., (2007), Scheduling algorithms, Berlin, Springer.

Brucker P. & Knust S., (2006), Complex scheduling, Berlin, Springer.

Fruwirth T. & Abdennadher S., (2003), Essentials of constraint programming, Springer

Grabowski J., Nowicki E. & Smutnicki Cz., (2003), Metoda blokowa w zagadnieniach szeregowania zadań, Akademicka Oficyna Wydawnicza EXIT.

IBM ILOG, (2009), ILOG OPL Language User's manual ver.6.3.

IBM ILOG, (2009), User's manual for CPLEX ver.12.2.

Liverani A. (2002): Using Genetic Algorithms To Optimise Kanban-Based Production System, The European Applied Business Research Conference, Germany.

Monch, L. & Driefel, R., (2005), A distributed shifting bottleneck heuristic for complex job shops, Computers and Industrial Engineering, pp. 363-380.

Nemhauser G. L. & Wolsey L. A., (1988), "Integer and Combinatorial Optimization", John Wiley & Sons, New York.

Neumann K., Schwindt Ch. & Zimmermann J., (203), Project scheduling with time Windows and scarce resources, Berlin, Springer.

Nocedal J. & Wright S.J., (1999), "Numerical optimization", Springer–Verlag, New York.

Pinedo, M., (2001), Scheduling: Theory, Algorithms, and Systems, Prentice Hall.

Preactor Int., (2005), Introduction to Scheduling, Preactor International.

Pytlak R. & Stecz W., (2010), „Optimizing routes for logistics operators", [in:] Zeszyty Naukowe Politechniki Poznańskiej, Poznań.

Stecz W. others, (2009), "Sugar plan transportation problem reconstruction using optimization software" [in:] Modelling of modern enterprises logistics, M. Fertsch, K. Grzybowska, A. Stachowiak (eds.) Publishing House of Poznan University of Technology, Poznań, pp. 25-37.

Wallace M., (2005), Hybrid algorithms, local search and ECLiPSe, CP Summer School

Wallace M. & Schimpf J., (2002), Finding the right hybrid algorithm – A combinatorial meta-problem, Annals of Mathematics and Artificial Intelligence 34: pp. 259–269.

Wolsey L.A. & Neumhauser N.L., (1999), "Integer and combinatorial optimization", Wiley-Interscience.

BIOGRAPHICAL NOTES

Radosław Pytlak is a Professor at Institute of Automatic Control and Robotics of Warsaw University of Technology. He teaches: modeling, simulation and optimization of dynamical systems; numerical methods; theory and methods of nonlinear programming; scheduling. His interests lie mainly in dynamic optimization and in algorithms for large scale optimization problems. He is the author of two monographs published in Springer-Verlag: Numerical methods for optimal control problems with state constraints; Conjugate gradient algorithms in nonconvex optimization. He published several papers in leading journals on optimization such as: SIAM J. on Optimization; SIAM J. on Control and Optimization; Journal of Optimization Theory and Applications; Numerische Mathematik.

Wojciech Stecz is an assistant professor at Faculty of Cybernetics Military University of Technology. He teaches: modeling and simulation of systems; numerical methods; theory and methods of nonlinear programming and scheduling.