Second Year Technical Report on Case Study 4: Value Chain Optimisation
Improvements in the Solution by Mathematical Programming

Sebastian Panek          Sebastian Engell
University of Dortmund
3rd May 2004

Project acronym: AMETIST
Project full title: Advanced Methods for Timed Systems
Project no.: IST-2001-35304 Project Co-ordinator: Frits Vaandrager
Project Start Date: 1 April 02
Duration: 36 months
Project home page: http://ametist.cs.utwente.nl/
1 Introduction

The first-year report [1] described how the scheduling problem provided by Axxom was modeled and solved using mixed-integer linear programming. The contribution of the Dortmund group comprised the following topics:

- A mixed integer problem formulation of the problem with the following properties:
  - a set of binary variables describes precedence relations between operations being executed on the same machine,
  - real variables represent starting and finishing dates of operations,
  - the objective function defined as a weighted sum of makespan, accumulated tardiness, and earliness terms.

- A set of heuristic rules to reduce the complexity of the model in terms of binary variables was introduced.
  1. Preordering by earliest due date (EDD),
  2. non-overtaking of operations belonging to jobs of the same type,
  3. non-overtaking of operations whose start and finishing dates do not overlap.

- A two-step solution procedure which uses all heuristics successively to find integer solutions quickly was developed:
  1. apply heuristics 1 and 3 by fixing some binary precedence variables and solve the problem,
  2. relax the fixed variables from step 1 and apply heuristics 2 and 3 by fixing the corresponding binary variables. Solve the problem again using the result of step 1 as initial solution.

- A rolling-horizon solution procedure which decomposes the problem into a series of smaller sub-problems. These are then solved separately. The solution of the master problem is finally composed of the solutions of the sub-problems.

- An implementation of the mixed integer formulation in the GAMS modeling language.

- Numerous numerical experiments with scaled problem instances, optimal scheduling and rolling-horizon settings. All experiments were carried out using the GAMS/Cplex optimization package.

Valid schedules could be computed successfully within a few minutes. Optimal or near-to-optimal schedules with up to 20 jobs could be computed quickly (few minutes). Larger instances turned out to be difficult and schedules computed with limited computation times can be far away from the optimum. Qualitatively good results could be computed using the moving-horizon procedure with less computational effort.

During the past year further improvements on the MILP approach to the Axxom case study have been achieved. These concern both, the modeling and the solution aspects.
Table 1: Results of the optimal-scheduling experiments from the first-year report. All values are hours-based. *Time* means the solution time in seconds, *Obj* is the objective value, *LB* is the lower bound, *Tardiness* is the value of the tardiness part of the objective function in hours, *Earliness* is the value of the part that penalties early job completions. *EDD* denotes the result of the first step of the 2-step solution procedure in which the EDD heuristics is used.

<table>
<thead>
<tr>
<th># Jobs</th>
<th>Time</th>
<th>EDD</th>
<th>Time</th>
<th>Obj</th>
<th>EDD</th>
<th>Obj</th>
<th>LB</th>
<th>Tardiness</th>
<th>Earliness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>24.43</td>
<td>0.765</td>
<td>0.6612</td>
<td>0.6612</td>
<td>0.6512</td>
<td>0.0012</td>
<td>0.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1200</td>
<td>1200</td>
<td>7.2310</td>
<td>7.2310</td>
<td>1.2916</td>
<td>0.0010</td>
<td>7.2300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1200</td>
<td>1200</td>
<td>10.7110</td>
<td>10.7110</td>
<td>1.2920</td>
<td>0.0010</td>
<td>10.7100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1200</td>
<td>1200</td>
<td>13.9308</td>
<td>13.9308</td>
<td>2.0920</td>
<td>0.0008</td>
<td>13.9300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1200</td>
<td>1200</td>
<td>11.9314</td>
<td>11.9314</td>
<td>2.0924</td>
<td>0.0014</td>
<td>11.9300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1200</td>
<td>1200</td>
<td>214.0013</td>
<td>214.0013</td>
<td>2.3724</td>
<td>198.0313</td>
<td>15.9700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1200</td>
<td>1200</td>
<td>1383.8372</td>
<td>1383.8372</td>
<td>3.7830</td>
<td>1367.3172</td>
<td>16.5200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 Improvements within the MILP approach

All results presented in the following sections were computed by the two-step solution procedure as it was explained in the first-year report. The three types of heuristics were used in the same fashion as before, thus the results can be compared directly to table 1 which is taken from the first report. The computation time in each of both steps was always limited to 20 minutes. Another termination criterion was a 5% optimality gap between the lower and the upper bound in both steps.

The objective function from the first-year report was not changed, it is a weighted sum of earliness and tardiness minimization terms. The weights were chosen as follows:

- 1 for the tardiness of the last operation,
- 0.01 for the earliness of the last operation
- 0.0001 for the tardiness of all other operations
- 0 for the earliness of all other operations.

Each change of either the model or the solution method was tested on a set of scaled problem instances. All instances are derived from the original job table provided by Axxom. The table was sorted according to due dates. In order to determine how the problem size influences the solution time, jobs were removed from the end of the original table to obtain smaller problem instances. The following instances were tested: 10, 14, 16, 18, 20, 22, 29 jobs.

All tests were computed on a 2.4 GHz Athlon machine with 1 GB of memory and the Linux operating system. This is due to the fact that the old machine was not available any more. Since both machines have the same Athlon CPU, the differences in results can be attributed to different clock rates. The new machine is approximately 1.6 times faster than the old one.

2.1 Minutes-based job table

The first improvement was the use of a minutes-based job table. It contains more accurate values than the job table with rounded integer dates in hours that was used in the previous version of the model. These new objective values must be divided by 60 to be comparable with the old results.

The results of various instances with the minutes-based job table are shown in table 2. All tests were performed using GAMS 20.7 and Cplex 8.1.
The results show that up to 20 jobs can be scheduled without delay. The last problem instance with 29 jobs exhibits a large accumulated delay of approximately 26500 minutes.

2.2 Solver update

Replacing GAMS 20.7/Cplex 8.1 by GAMS 21.3 and Cplex 9.0 has led to some changes of the results as shown in table 3. All other settings are the same as in the previous section.

From the results, it is not clear which solver version performs better. While smaller problem instances mostly are solved better, for the last instance a poor result with more than 38500 minutes accumulated delay was computed within 40 minutes.

Nevertheless, the newer versions of GAMS and Cplex were chosen to carry out further experiments, as described in the following sections.

2.3 Parameter studies

The performance of Cplex strongly depends on several parameters which have a critical influence on its efficiency for particular problems. Unfortunately, it is not entirely clear which parameter setup is optimal for which classes of problems. To identify a set of best parameters for a family of problem instances, so called parameter studies are often required.

Thus, a kind of meta-optimization is necessary to determine an optimal setup of the solver. It should not only involve as many parameters as possible, but also include a large number of problem instances to avoid finding parameters which improve the solution effort only for a single or few problem instances.

Even if it is possible to identify promising parameter settings, it is difficult to predict whether this setup will still be advantageous if the problem is slightly changed. Furthermore, it is difficult to
explain why some parameters have lead to improvements or why not. This problem comes mainly from the fact that the vendor of Cplex does not publish enough information about the algorithms and details of the implementation.

Another problem is that the large number of parameters and corresponding values makes it difficult to test all possible combinations. A way to overcome this problem is to test only one parameter at once assuming orthogonality of the effects of individual parameters.

The following parameters of Cplex were tested:

- **ppriind = 0,1,2,3,4**
  Pricing algorithm for the primal Simplex method. -1 = Reduced cost pricing, 0 = Hybrid reduced-cost and Devex pricing, 1 = Devex pricing, 2 = Steepest-edge pricing, 3 = Steepest-edge pricing with slack initial norms, 4 = Full pricing.

- **dpriind = 0,1,2,3,4**
  Pricing strategy for dual Simplex method. 0 = Determined automatically, 1 = Standard dual pricing, 2 = Steepest-edge pricing, 3 = Steepest-edge pricing in slack space, 4 = Steepest-edge pricing, unit initial norms.

- **mipordtype = 0,1,2,3**
  This option is used to select the type of generic priority order to generate when no priority order is present. 0 = None (default), 1 = Decreasing cost magnitude, 2 = Increasing bound range, 3 = Increasing cost per coefficient count.

- **brdir = -1,1**
  Used to decide which branch (up or down) should be taken first at each node. -1 = Down branch selected first, 0 = Algorithm decides (default), 1 = Up branch selected first.

- **varsel = -1,1,2,3,4**
  This option is used to set the rule for selecting the branching variable at the node which has been selected for branching. -1 = Branch on variable with minimum infeasibility, 0 = Branch variable automatically selected, 1 = Branch on variable with maximum infeasibility, 2 = Branch based on pseudo costs, 3 = Strong Branching, 4 = Branch based on pseudo reduced costs.

The results of parameter studies are shown in table 4. Here, the computation time was limited to 60 minutes in the first step and 30 minutes in the second step of the solution procedure. All other parameters are set to default values. The last line shows a combination of two parameters: varsel=1 which provided a very quick first integer solution, and the best values of dpriind=1,3.

These efforts helped identifying one parameter which leads to great improvements: dpriind=1. Any other parameters did not show a clear advantage. Table 5 shows results of the new parameter setting with scaled problem instances.

The results show that the new setting leads to improved results. For the last instance the solution is one order of magnitude better than before. With extended CPU time limitations the algorithm was able to compute delay-free schedules in the first step within less than one hour.

### 2.4 Advances in modeling

Some efforts concerning the modeling of the scheduling problem have led to further improvements. In general, there are three types of jobs the structure of which is basically the same. First, the operations pre-dispersion and dispersion are started. In parallel, a mixing vessel is allocated and used until the job is completed. Afterwards, the dose spinner dk1 is invoked to mix the components
Table 4: Parameter studies with various Cplex parameters. Only the best integer objectives of both steps are presented. N.i.s. means no integer solution because both steps failed. In experiments \(dpriind=2,3\) and \(dpriind=3\) (\(varsel=1\)) the second step of the solution procedure failed.

<table>
<thead>
<tr>
<th>parameter</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppriind</td>
<td>-</td>
<td>56662.43</td>
<td>56662.43</td>
<td>56662.43</td>
<td>56662.43</td>
<td>56662.43</td>
</tr>
<tr>
<td>dpriind</td>
<td>-</td>
<td>56662.43</td>
<td>3387.01</td>
<td>22382.29</td>
<td>63300.58</td>
<td>30403.43</td>
</tr>
<tr>
<td>mipordtype</td>
<td>-</td>
<td>56662.43</td>
<td>56662.43</td>
<td>56662.43</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>brdir</td>
<td>-</td>
<td>273897.63</td>
<td>-</td>
<td>87282.94</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>varsel</td>
<td>n.i.s.</td>
<td>-</td>
<td>22248.27</td>
<td>56662.43</td>
<td>16681.94</td>
<td>43030.75</td>
</tr>
<tr>
<td>dpriind (varsel=1)</td>
<td>-</td>
<td>-</td>
<td>139408.68</td>
<td>-</td>
<td>162491.54</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Scalability tests with GAMS 21.3/Cplex 9.0 and \(dpriind=1\).

<table>
<thead>
<tr>
<th># Jobs</th>
<th>Time EDD</th>
<th>Time</th>
<th>Obj EDD</th>
<th>Obj</th>
<th>LB</th>
<th>Tardiness</th>
<th>Earliness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>27.53</td>
<td>0.42</td>
<td>30.72</td>
<td>30.72</td>
<td>30.72</td>
<td>0</td>
<td>30.72</td>
</tr>
<tr>
<td>14</td>
<td>1200</td>
<td>1200</td>
<td>232.14</td>
<td>232.14</td>
<td>65.49</td>
<td>0</td>
<td>232.14</td>
</tr>
<tr>
<td>16</td>
<td>1200</td>
<td>1200</td>
<td>339.44</td>
<td>339.44</td>
<td>65.49</td>
<td>0</td>
<td>339.44</td>
</tr>
<tr>
<td>18</td>
<td>1200</td>
<td>1200</td>
<td>546.87</td>
<td>297.67</td>
<td>108.74</td>
<td>0</td>
<td>297.67</td>
</tr>
<tr>
<td>20</td>
<td>1200</td>
<td>1200</td>
<td>597.79</td>
<td>597.79</td>
<td>108.74</td>
<td>0</td>
<td>597.79</td>
</tr>
<tr>
<td>22</td>
<td>1200</td>
<td>1200</td>
<td>788.13</td>
<td>688.26</td>
<td>121.55</td>
<td>0</td>
<td>688.26</td>
</tr>
<tr>
<td>29</td>
<td>1200</td>
<td>1200</td>
<td>5315.45</td>
<td>2969.35</td>
<td>136.2</td>
<td>1229.21</td>
<td>1740.14</td>
</tr>
</tbody>
</table>

of the lacquer. After that, a quality check is done in the laboratory (operation lab1). The quality is corrected by using the dose spinner a second time dk2. Finally, the quality of the color is checked again (lab2) and the mixing vessel is moved to the filling station (abf) and cleaned after the filling.

In the previous model the resource lab was defined as a non-critical resource. Despite this fact, all operations on it were defined within the model at thus required additional variables and appeared in the final schedule. In order to reduce the model size, the resource lab and all operations requiring it have been removed. This was motivated by the fact that lab is a non-critical resource and thus may be used whenever and as long as necessary.

In the present model, the timing constraints between the end of the first dk operation and the beginning of the first lab operation have been deleted. Furthermore, both lab operations have been replaced by a minimum delay between both dk operations and a second minimum delay between the second dk operation and the abf operation. Essential parts of both job templates, before and after removing the lab operations are shown in Fig. 1 as Pert Charts.

Removing two operations from all of 29 jobs has decreased the number of operations by 58 to 144. The results for the new model are shown in table 6. Surprisingly, the results have became slightly worse for most instances. Thus, the new model with fewer operations is not necessarily easier to solve. Again, a final run with extended CPU time limitations was performed. The first step was limited to 60 and the second to 30 minutes. In this final run, the solver was able to compute a delay-free schedule that is slightly better than the result in table 5.
Figure 1: Modeling of jobs before and after changes. $a$ and $b$ are durations of both lab operations, $H$ is the constant maximum time horizon.

Table 6: Scalability tests with GAMS 21.3/Cplex 9.0, dprind=1 without the lab resource.

<table>
<thead>
<tr>
<th># Jobs</th>
<th>Time</th>
<th>EDD</th>
<th>Time</th>
<th>Obj</th>
<th>EDD</th>
<th>Obj</th>
<th>LB</th>
<th>Tardiness</th>
<th>Earliness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>7.77</td>
<td>0.34</td>
<td>35.67</td>
<td>35.67</td>
<td>36.67</td>
<td>0.07</td>
<td></td>
<td>35.60</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1200</td>
<td>1200</td>
<td>343.93</td>
<td>343.93</td>
<td>72.87</td>
<td>0.06</td>
<td></td>
<td>343.87</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1200</td>
<td>1200</td>
<td>343.72</td>
<td>343.72</td>
<td>72.89</td>
<td>0.08</td>
<td></td>
<td>343.64</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1200</td>
<td>1200</td>
<td>508.85</td>
<td>498.47</td>
<td>118.54</td>
<td>0.08</td>
<td></td>
<td>498.39</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1200</td>
<td>1200</td>
<td>536.94</td>
<td>465.51</td>
<td>118.56</td>
<td>0.10</td>
<td></td>
<td>465.41</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1200</td>
<td>1200</td>
<td>4572.11</td>
<td>965.66</td>
<td>133.77</td>
<td>61.07</td>
<td></td>
<td>904.59</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1200</td>
<td>1200</td>
<td>6016.09</td>
<td>6016.09</td>
<td>196.20</td>
<td>4737.58</td>
<td>1278.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>3600</td>
<td>1800</td>
<td>1233.5</td>
<td>1208.47</td>
<td>196.04</td>
<td>0.084</td>
<td></td>
<td>1208.39</td>
<td></td>
</tr>
</tbody>
</table>
Table 7: Scalability tests with GAMS 21.3/Cplex 9.0, dpriind=1, lab resource and an objective function without earliness penalty.

<table>
<thead>
<tr>
<th># Jobs</th>
<th>Time</th>
<th>EDD</th>
<th>Obj</th>
<th>LB</th>
<th>Tardiness</th>
<th>Earliness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.19</td>
<td>0.36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>18.63</td>
<td>0.91</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>33.25</td>
<td>1.24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>57.78</td>
<td>1.73</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>90.66</td>
<td>2.32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>132.25</td>
<td>2.80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>1200</td>
<td>1200</td>
<td>877.06</td>
<td>877.06</td>
<td>0</td>
<td>877.06</td>
</tr>
</tbody>
</table>

Table 8: Scalability tests with GAMS 21.3/Cplex 9.0, dpriind=1, without the lab resource and with an objective function without earliness penalty.

<table>
<thead>
<tr>
<th># Jobs</th>
<th>Time</th>
<th>EDD</th>
<th>Obj</th>
<th>LB</th>
<th>Tardiness</th>
<th>Earliness</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5.33</td>
<td>0.33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>20.12</td>
<td>0.77</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>37.87</td>
<td>1.21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>57.07</td>
<td>1.64</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>126.99</td>
<td>2.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>243.71</td>
<td>2.61</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>693.78</td>
<td>5.32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

2.5 Lateness minimizing objective function

In the TA approach only tardiness minimization was considered thus far. Thus, the question arises, how the performance of the MILP approach changes, if only the tardiness of the jobs is minimized. The new objective function can be easily derived from the old one by setting weights penalizing earliness to 0. The new weights are:

- 1 for the tardiness of the last operation,
- 0.0001 for the tardiness of all other operations
- 0 for the earliness of all operations.

The results of the experiments with the new objective function and with the lab resource are shown in table 7.

The computation times have improved dramatically. In most instances the optimal solution was computed already in the first step of the solution procedure. Only for the last instance no delay-free schedule could be computed within 40 minutes.

This result is not surprising because the problem is posed such that the due dates are always two weeks after the release dates. That means that there is usually much time to complete a job if it is started as early as possible (i.e. at its release date). Thus, the problem seems to be much easier penalizing only tardiness.

The results in table 8 were obtained by the combination of the new objective function and the new model without lab (as investigated in section 2.4).
In this case the smaller model exhibits better performance. All instances could be solved without delay within 12 minutes. The second step of the solution procedure did not lead to any improvements, thus for tardiness minimization the EDD heuristic is effective, in accordance with known results on job-shop scheduling.

3 Conclusions

The advances in the MILP approach have lead to considerable improvements in terms of computation time and quality of the computed schedules:

1. The parameter studies with the Cplex parameters helped identifying a setting which leads to better schedules in less computation time for large problems.

2. Using both tardiness and earliness penalization delay-free schedules could be computed in less than 60 minutes for all examples.

3. An objective function which penalizes only the tardiness leads to problems which are much easier to solve.

4. The combination of an improved MILP model with less operations and an objective function which penalizes only the tardiness allows us computing delay-free schedules in less than 700 seconds.

5. It is not clear whether the new versions of GAMS and Cplex give a better performance for the problem instances and parameter settings discussed here.

6. Different to the results reported in the first-year report, the second step of the solution procedure often led to improvements which can be attributed to the new version of Cplex and the choice of the solver parameters.

References