# Chapter A

# Test Instances

Similar to e. g. Derstroff (1995) and Sürie (2005) we generated various differing instances as to diminish the effect of the characteristics of the test set on the results.

In the context of dynamic multi-level capacitated lotsizing problems with setup times, the relevant distinguishing characteristics are:[1]

1. the size of the problem (i. e. the number of items, the number of resources, the number of production levels and the length of the planning horizon),
2. the product and the process structures,
3. the development of demand in time,
4. the length and distribution of setup times,
5. the ratio of setup and holding costs,
6. the capacity utilization profiles and finally
7. the number of parallel machines.

For our test sample, these characteristics will be described in the remainder of this chapter.

1 Compare Derstroff (1995), page 90.

## A.1 Problem Sizes

In the following, the number of production levels will equal the number of resources. Hence, the problem size is made up of the number of items, the number of resources and the number of periods. We will consider the six combinations depicted in table A.1.

| Class | ♯ Products | ♯ Resources | ♯ Periods | ♯ Instances |
|---|---|---|---|---|
| 1 | 10 | 3 | 4 | 480 |
| 2 | 10 | 3 | 8 | 480 |
| 3 | 20 | 6 | 8 | 240 |
| 4 | 20 | 6 | 16 | 240 |
| 5 | 40 | 6 | 8 | 240 |
| 6 | 40 | 6 | 16 | 240 |

*Table A.1: Problem Sizes of the test instances*

For each instance class, a folder was created, which contains all files necessary to generate the complete instance set.

## A.2 Product and Process Structures

Four types of product structures are generally distinguished: linear, divergent, assembly and general.[2] In serial product structures, every item has at most one predecessor and at most one successor. In divergent (assembly) product structures, each item has at most one predecessor (successor), but can have an unlimited number of successors (predecessors). In general product structures finally, there is neither a limit on the number of predecessors nor on the number of successors. Hence, the general product structure is the most complex and the other three can be seen as simplifications of it.

For our tests, we will consider assembly and general product structures. Furthermore, we choose to set the production coefficient $a_{kj}$ to 1 for all items $k$ and $j$, where $j$ is a direct successor of $k$.

As regards the process structures, cyclic and acyclic production processes can be distinguished. In the latter case, items are produced on a different resource than their predecessors and their successors. In cyclic process structures, however, some parents are produced on the same resource as their component. In the following, we will consider both types of production processes.

This leads to four possible combinations of product and process structures.

- general and acyclic

2 Compare Derstroff (1995), page 21.

- general and cyclic
- assembly and acyclic
- assembly and cyclic

The product and process structures are given for 10 items and 3 resources in figure A.1.

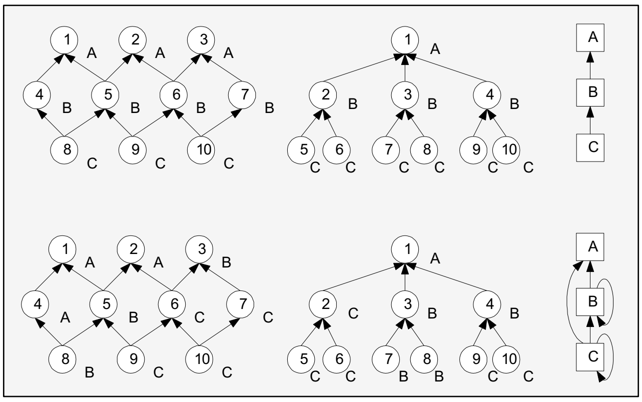


*Figure A.1: Product and process structures for 10 items and 3 resources*

The product and process structures for 20 items and 6 resources are depicted in figure A.2.

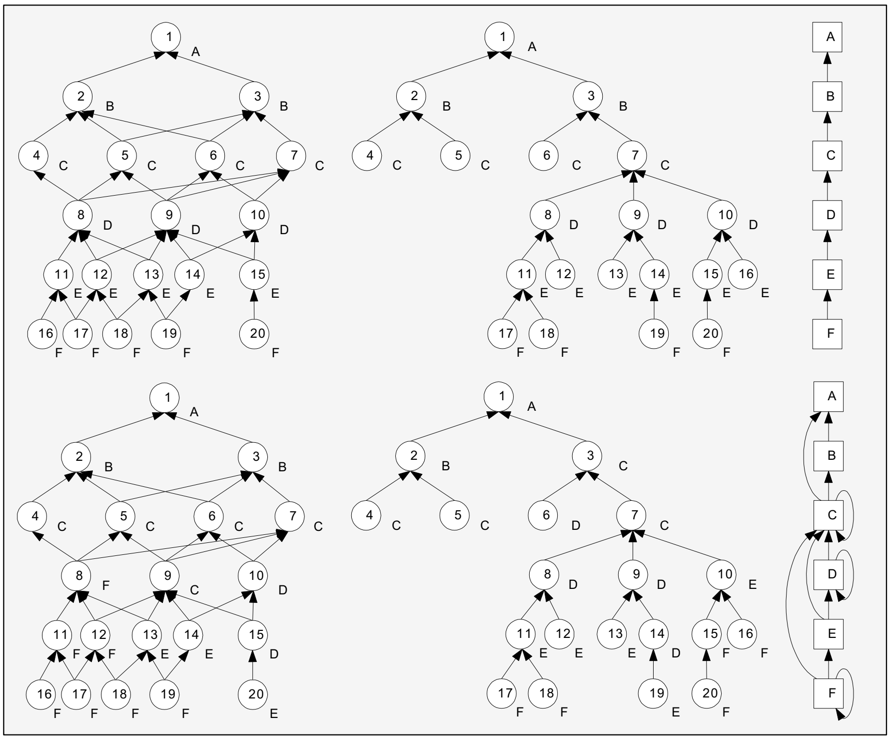


*Figure A.2: Product and process structures for 20 items and 6 resources*

Finally, the process and product structures for 40 items and 6 resources are given in figure A.3.

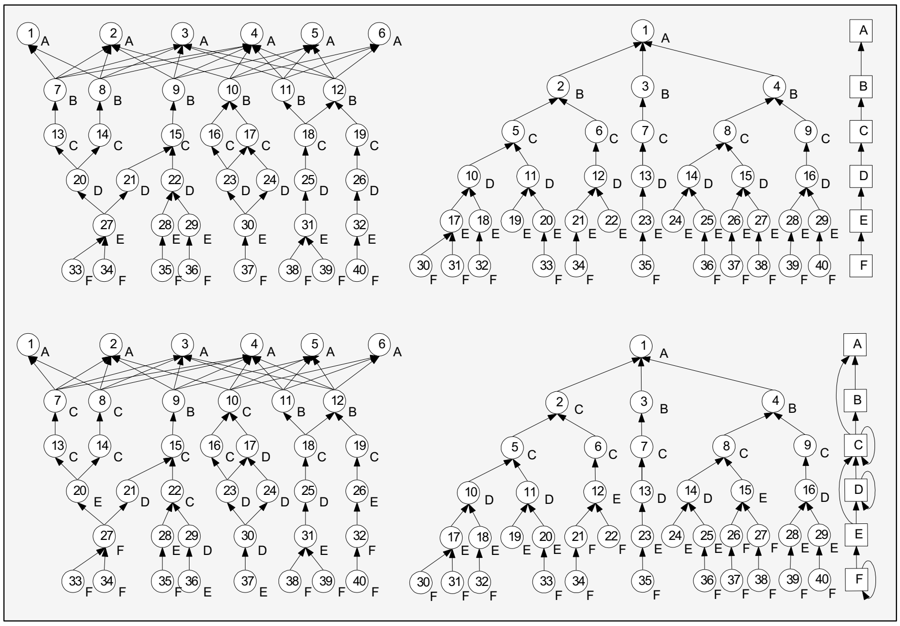

*Figure A.3: Product structures and process structures for 40 items and 6 resources*

The product and process structures are respectively contained in the '*.dat' files. Hence, there is one such file for every structure presented in the figures above. Each '*.dat' file contains first the size of the problem, i. e. the number of items, the number of resources and the number of periods. Then for each item the total holding costs $h_k$ are given. They were computet by assuming an echelon holding costs coefficient of $e_k = 1$ for all items. Then follow the production time per unit, which was assumed to be $tb_k = 1$ for all items and the number of the resource that produces the item. Finally all parents and components are paired and the according production coefficient $a_{kj}$ and the lead time are given. Both were again assumed to equal 1 for all pairs.

## A.3 Demand Profiles

External demand is assumed to be normally distributed. The distribution is truncated, such that negative values were set to zero. To generate demand, we used the 'Mersenne Twister' as described in Matsumoto and Nishimura

(1998). The means for the respective end items and the components are given in table A.2.

Respectively one demand series was computed for coefficients of variation of 0.2, 0.5 and 0.8. They were chosen as to reflect the influence of the volatility of demand.

| ♯ Items | Mean demand of end items | Mean demand of components |
|---|---|---|
| 10 (general) | 30 (1), 70 (2), 100 (3) | 20 |
| 10 (assembly) | 100 | 20 |
| 20 (general) | 100 | 30 |
| 20 (assembly) | 80 | 30 |
| 40 (general) | 50 (1), 60 (2), 70 (3), 80 (4), 90 (5), 100 (6) | 15 |
| 40 (assembly) | 90 | 15 |

*Table A.2: Mean demand*

For ease of representation and to ensure feasibility, we set the initial inventory to 20 for all items. Furthermore, for all predecessor items, preproduction $x_{k0}$ was computed to match the corresponding demand from all direct successors in the first period:

$$x_{k0} = \sum_{j \in N_k} a_{kj} \cdot D_{j1} \tag{A.1}$$

The total demand $D_{kt}$ is recursively[3] computed as:

$$D_{kt} = d_{kt} + \sum_{j \in N_k} a_{kj} \cdot D_{j,t+1} \tag{A.2}$$

The demand series together with initial inventory and preproduction are given in the '*.dem' files.

## A.4 Setup Time Profiles

Setup times are set to 5, 10 and 20 time units in order to reflect the different impacts of setup time. They are allocated to the different items such that respectively two different setup time profiles are created for each class. The individual setup time profiles are shown in table A.3.

3 from the end items down to the components

| ♯ Items | Profile | $tr_k = 5$ | $tr_k = 10$ | $tr_k = 20$ |
|---|---|---|---|---|
| 10 | 1 | $1, 2, 3$ | $4, 5, 6, 7$ | $8, 9, 10$ |
| 10 | 2 | $8, 9, 10$ | $4, 5, 6, 7$ | $1, 2, 3$ |
| 20 | 1 | $1, \ldots, 6$ | $7, \ldots, 14$ | $15, \ldots, 20$ |
| 20 | 2 | $15, \ldots, 20$ | $7, \ldots, 14$ | $1, \ldots, 6$ |
| 40 | 1 | $1, \ldots, 13$ | $14, \ldots, 27$ | $28, \ldots, 40$ |
| 40 | 2 | $28, \ldots, 40$ | $14, \ldots, 27$ | $1, \ldots, 13$ |

*Table A.3: Setup time profiles*

The setup time profiles are given in the '*.set' files.

## A.5 Setup and Holding Cost Ratios

When setup and holding costs are considered, the lotsizing problem arises from the opposing nature of these two cost components. While setup costs decrease with the lotsize, the holding costs increase and vice versa. Thus, for the nature of the problem, not the absolute values of the cost components are decisive, but their ratio. For ease of representation, we set the echelon holding cost factor $e_k = 1$ for all $k$. Then, the setup costs result from the target ratio.

We will follow Helber (1994) and Derstroff (1995) in expressing the target cost ratio via the time between orders (TBO), which results from a given ratio in the static lotsizing model. With the total demand given in equation (A.2), the average demand of an item $k$ is computed as:

$$\overline{D}_k = \sum_{t=1}^{T} D_{kt} \tag{A.3}$$

The time between orders is calculated as:

$$TBO_k = \sqrt{\frac{2 \cdot s_k}{\overline{D}_k \cdot e_k}} \tag{A.4}$$

Therefore, for a given TBO profile, the setup costs for item $k$ are calculated as:

$$s_k = 0.5 \cdot e_k \cdot \overline{D}_k \cdot (TBO_k)^2 \tag{A.5}$$

Time between orders are set to 1, 2 and 4. For instances with 10 items (classes 1 and 2) four different TBO profiles are applied. Three profiles consist of the same time between orders for all items. The fourth sets the same time between orders for all items on the same production level. For the instances with 20 and 40 items, we generated two TBO profiles, respectively. The first consists of a TBO of 2 for all items. The second again consists of different TBOs for items produced on different levels. The TBO profiles are given in table A.4.

| ♯ Items | Profile | TBO = 1 | TBO = 2 | TBO = 4 |
|---|---|---|---|---|
| 10 | 1 | $1,\ldots,10$ | | |
| 10 | 2 | | $1,\ldots,10$ | |
| 10 | 3 | | | $1,\ldots,10$ |
| 10 (assembly) | 4 | 1 | $2,\ldots,4$ | $5,\ldots,10$ |
| 10 (general) | 4 | $1,\ldots,3$ | $4,\ldots,7$ | $8,\ldots,10$ |
| 20 | 1 | | $1,\ldots,20$ | |
| 20 | 2 | $1,\ldots,3$ | $4,\ldots,10$ | $11,\ldots,20$ |
| 40 | 1 | | $1,\ldots,40$ | |
| 40 (assembly) | 2 | $1,\ldots,4$ | $5,\ldots,16$ | $17,\ldots,40$ |
| 40 (general) | 2 | $1,\ldots,6$ | $7,\ldots,19$ | $20,\ldots,40$ |

*Table A.4: TBO profiles*

The setup costs are given in the '*.sco' files.

## A.6 Capacity Utilization Profiles

To vary the characteristics of the test instances, we install different capacity utilization profiles. The absolute capacity is hence computed as to achieve the desired target utilization. We define capacity utilization, $U_m$, of any resource $m$ as:

$$U_m = \frac{\sum_{k \in K_m} \sum_{t=1}^{T} (tr_k \cdot \gamma_{kt} + tb_k \cdot q_{kt})}{\sum_{t=1}^{T} b_t^m} \tag{A.6}$$

The problem is, that neither the production quantities $q_{kt}$ nor the setup decisions $\gamma_{kt}$ are known in advance. Therefore, they have to be approximated based on the exogenous data.

For ease of representation, the production time is set to $tb_k = 1$ for all $k$.

As Kimms (1997) points out, it is not a good idea to calculate an overall capacity average. The reason id that neglecting the dynamic nature of demand may lead to feasibility problems in the early periods. Thus, the minimum average capacity consumption through production times on resource $m$ in periods $\hat{t}$ to $t$ is computed as:[4]

$$\overline{D}_{m\hat{t}t} = \frac{\sum\limits_{k \in K_m} \sum\limits_{\tau=\hat{t}}^{t} tb_k \cdot D_{k\tau}}{t - \hat{t} + 1} \tag{A.7}$$

During the cumulated lead time $L_m = \max\limits_{k \in K_m} L_k$ the minimum average capacity might not be sufficient as lotsizing is limited by predecessor item availability. Hence, in these first periods, capacity is computed in each period individually.

The resulting capacity limit matrix is lined out it algorithm 1.[5]

4 The total demand $D_{k\tau}$ is again computed according to equation (A.2).
5 See Kimms (1997), page 84.

**Algorithm 1** Generation of the capacity matrix

**for** $m \in M$ **do**

$t = 1$

**while** $t \leq L_m$ **do**

$b_t^m = \left\lceil \sum_{k \in K_m} (tb_k \cdot D_{k\tau} + tr_k) \cdot \frac{1}{U_m} \right\rceil$

$t = t + 1$

**wend**

**while** $t \leq T$ **do**

$\hat{t} = t$

$D_m = 0$

**while** $D_m \leq \overline{D}_{m\hat{t}t}$ and $t \leq T$ **do**

$D_m = \overline{D}_{m\hat{t}t}$

$t = t + 1$

**wend**

$\tau = \hat{t}$

**while** $\tau \leq t - 1$ **do**

$b_\tau^m := \left\lceil \left( D_m + \sum_{k \in K_m} tr_k \right) \cdot \frac{1}{U_m} \right\rceil$

$\tau = \tau + 1$

**wend**

**wend**

**end for**

Like Derstroff (1995), we consider five capacity profiles, which result from different combinations of the possible capacity utilizations of 50%, 70% and 90% (see table A.5).

| ♯ Resources | Profile | $U_m = 50\%$ | $U_m = 70\%$ | $U_m = 90\%$ |
|---|---|---|---|---|
| 3 | 1 | A, B, C | | |
| 3 | 2 | | A, B, C | |
| 3 | 3 | | | A, B, C |
| 3 | 4 | A | B | C |
| 3 | 5 | C | B | A |
| | | | | |
| 6 | 1 | A,. . ., F | | |
| 6 | 2 | | A,. . ., F | |
| 6 | 3 | | | A,. . ., F |
| 6 | 4 | A, D | B, E | C, F |
| 6 | 5 | A, B | C, D | E, F |

*Table A.5: Capacity profiles*

The capacities are given in the '*.cap' files.

# Bibliography


Derstroff, M. C. (1995). *Mehrstufige Losgrößenplanung mit Kapazitätsbeschränkungen*. Produktion und Logistik. Heidelberg: Physica-Verlag.

Helber, S. (1994). *Kapazitätsorientierte Losgrößenplanung in PPS-Sytemen*. Stuttgart: M und P.

Kimms, A. (1997). *Multi-Level Lot Sizing and Scheduling*. Production and Logistics. Heidelberg: Physica-Verlag.

Matsumoto, M. and T. Nishimura (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation 8*(1), 3–30.

Sürie, C. (2005). *Time continuity in discrete time models*, Volume 552 of *Lecture Notes in Economics and Mathematical Systems*. Berlin: Springer.