Constraint-Based Production Planning for Simulation of Real Supply Nets

Hartwig Baumgaertel and Ulrich John

DaimlerChrysler AG, Research Information & Communication,
email: {Hartwig.Baumgaertel, Ulrich.John}@daimlerchrysler.com

Abstract. Simulation is a proper approach for supporting the goal-oriented design and redesign of real supply nets.
In this paper we present our constraint-based production planning component which is integrated in our simulation tool SNS for realistic emulation of intelligent/ advanced planning behaviour inside companies of a supply net.
Due to the generation of several types of redundant constraints inside the planning model, the planning component is able to produce batch-oriented plans that fullfil given customer demands under simultaneous consideration of part-availability constraints, storage-capacity constraints and production-capacity constraints. Additionally, several optimization objectives can be taken into account.

1 Introduction

Nowadays, the goal-oriented design/ redesign of planning processes for industrial supply nets becomes more and more important. On the one hand we can observe the continuing trend to outsource parts of the own production which results in larger and closely coupled supply networks. On the other hand the proper assessment and operation of the supply nets is essential to assure and enlarge the market position.

Because of the high complexity and dynamics inherent to supply nets, the design/ redesign of supply nets is rather complicated. It is general accepted that especially the using of powerful simulation approaches is a suitable way to takle the problem.

Planning and Control tasks in real supply nets are done by humans and/or by planning systems. In large companies, the used planning systems are recently more often *advanced planning systems* which use constraint technology to takle also more advanced planning problems and to obtain better plans.

So we can find a lot of different planning behaviours in real supply nets. A crucial issue regarding the quality of supply-net simulation software is wheter the software is able to emulate the different types of planning.

In other words, software for supply-net simulation which should be usable as universal as possible must contain also an advanced planning component that allow the modeling/emulation of different (intelligent) planning behaviours which can be found in real companies (cf. e.g. [6], [2]).

We have developed a prototypical advanced planning component (SNS-APS) which is responsible for the emulation of advanced planning methods inside our simulation software *SNS* that was only be able to emulate the traditional production planning methods, like MRPII, before. The realization of the planning component was done using *ILOG OPL* (cf. [3]). That is, we used *ILOG OPL Studio* (cf. [4]) as development environment and the *ILOG OPL Studio Component Libraries* Java API (cf. [5]) to connect the solver to *SNS* which is written in Java. Because of the general nature of our approach, the proposed constraint model can also be realized with other finite domain constraint software.

In the next section we describe the class of production planning problems which is adressed by the advanced planning component developed by us. Then we outline the basic constraint model for the solution of the adressed problem types together with some used search heuristics. As expected, the basic solution model is not sufficient for the general case. So we had to deduce redundant constraints with the intention to achieve a further significant reduction of search space. These redundant constraints are described in Section 3. A simulation example using the presented planning component is given in Section 4. Finaly, we will give a conclusion and some remarks on future work.

2 Constrained Batch-Oriented Production Planning

The initial intention for the planning component was to enable the generation of production output plans (and production release plans) for producing companies which fullfil given problem constraints. Associated with that intention, we had the objective to ensure a high degree of universality regarding the class of processable problems (see below). In the case that a given problem is recognized as unsolvable (inconsistent constraint set), the system inference should compute reasons for the inconsistency if possible. The identified bottle necks can be used as input for the communication to the planning systems of regarding supplier companies or customers in order to find a proper relaxation of the planning problem. Intrinsically, this has the potential for simulation and realization of novel distributed planning processes.

2.1 Planning Problem

Let us consider the general case of an arbitrary producing enterprise (see Figure 1) which has to plan its output over a given planning horizon (τ) which consists of several time slices/ periods ($t1, \ldots, tn$). The enterprise is able to produce a set of different product types (*Variants*).

For each product type g, a finished good inventory ($inventory(g)$) may exist which can be limited by a given storage capacity and by a demanded minimal inventory (*inventory constraints*). For the production of a product of type p, a set of parts ($parts(p)$) is required[1] . That way, for each enterprise, a set $Parts$

[1] The amounts of needed parts is stored in a bill of material (BOM).

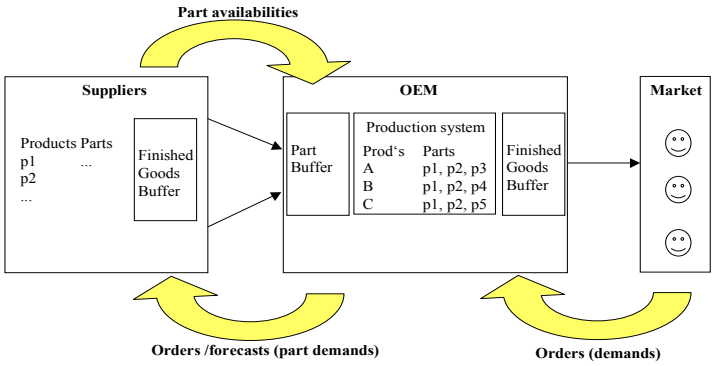

Fig. 1. Scheme of a very simple supply chain

exists that contains all identifiers of parts which can be used in production. For each part type p, a part inventory ($part\_inventory(p)$) may exist. Like in the case of finished goods inventories, the part inventories are also restricted by given capacity constraints (*part inventory constraints*) and lower limits. Regarding the refilling of the part inventories by the suppliers, some *availability information* (*availability constraints*) exist depending on the current part inventory level, the amount of parts in transit, and the capacity of the corresponding supplier company. In the normal case, the amount of production which can be done in a time slice t is restricted by existing *production capacitiy constraints* ($capacity(t)$)[2].

Coarsly said, the **core planning problem** consists in planning the production of demanded products in such a way that they can be delivered to the customers with the demanded amount at the desired delivery time/ delivery period while taking into account all given constraints.

The complexity of the problem could be increased by the fact that the production of a given product type p is only possible in fixed batchsizes ($batchsize(p)$) due to technical restrictions. In the following, we call production planning problems which contain batch-size constraints *batch-oriented production planning problems*.

In addition, several optimization objective – like minimizing the inventory deviations from given desired inventory values – can be part of the planning problem.

We can sum up the customer demands that way that we have for each product type p and for each time slice t of the planning horizon a tupel $[p, amount, t]$ which represents the overall demand regarding product type p for time slice t.

2.2 Frame Model

It can be quite easily recognized that the described planning problem can be modeled in a rather declarative way by using finite domain variables and con-

[2] For reasons of simplicity, we restricted us for the first approach to cases where each producible product type needs one unit of production capacity.

straints between the variables. Now we want to present the essentials of the constraint-based frame model which was used as the basis of our planning component. The main variables and main constraints of the frame model are schematized in Figure 2.

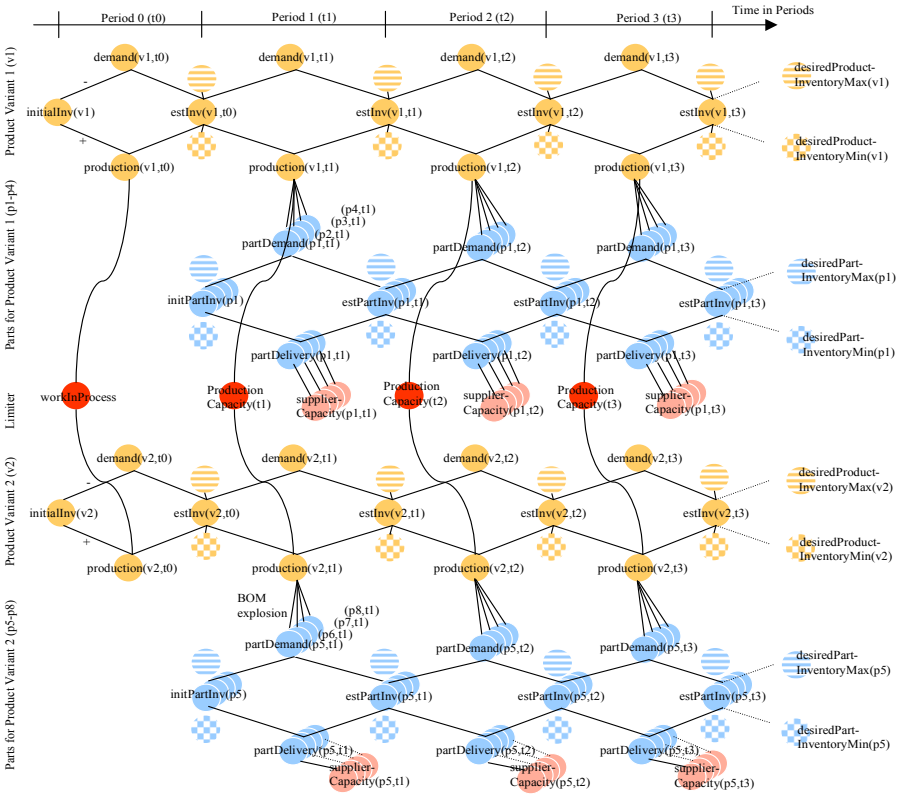

Fig. 2. Frame Model Variables and Constraints

Regarding the planning horizon, we assume that it consists of the connected time slices $t1, \ldots, tn$ and is represented by $\tau = t1, \ldots, tn$. In the real application, we differentiate between the *output-planning horizon* and a *release-planning horizon*. For reasons of simplicity, we will restrict us to the consideration of the output-planning horizon whenever we can.

Model Constants We can identify several constants which occur in problems of the described problem class. So we have an initial inventory for each producible variant v at the begin of the planning horizont ($initialInv(v)$), initial part inventories for each part p ($initPartInv(p)$), fixed over-all production capacities

($productionCapacity(t1), \ldots, productionCapacitity(tn)$), the *work in process* – differentiated regarding the different variants ($wip(v1), \ldots, wip(vs)$) – and limits for each product inventory ($desiredProductInvMin$, $desiredProductInvMax$) and part inventory ($desiredPartInvMin$, $desiredPartInvMax$).

The delivery of parts p in period i is restricted by given supplier capacities ($supplierCapacity(p, i)$) which are also constants of our model.

The production batchsizes ($batchsize(v1), \ldots, batchsize(vn)$) are treated as time-independent model constants.

In addition to the allready mentioned model constants, we have a data structure which stores the bill of materials (BOM) for the producable product types. For a product type v and a part p, $bom(v, p)$ is the amount of p which is needed to produce one pice of v.

For each period i, we have one over-all demand per variant v (see above) which is denoted by $demand(v, i)$.

Model Varibles Besides constants, we can identify several variables which will be represented by finite domain variables of the frame model: The planned production output for a product variant $v \in Variants$ in period $i \in \tau$ is represented by the FD-variable $production(v, i)$. That way, we have $card(Variants) \times card(\tau)$ production-output variables in the frame model. For each period i and part p, we have a variable $partDemand(p, i)$ which represents the amount of part p needed for production in period i. The respected part amounts can be taken from the corresponding part inventories or from just arriving part deliveries if allowed.

For each period i and for each variant v, we have a FD-variable $estInv(v, i)$ which represents the estimated inventory at the end of period i. In analogy, we have a variable $estPartInv(p, i)$ for each $p \in Parts$ and $i \in \tau$.

For realizing the batchsize constraints (see below), we introduce for each pair $[v, i]$ with $v \in Variants$ and $i \in \tau$, the batchsize-related constraint variables $batchproduction(v, i)$, $batchnumber(v, i)$, $carryover(v, i)$ and $carryoverflag(v, i)$.

The respective variable $batchproduction(v, i)$ is the amount of product type p which is planed to be produced in batches which are started inside period i. The number of full batches is $batchnumber(v, i)$. $carryover(v, i)$ is respectively the amount of product type p which is planned to be produced in period $i + 1$ but belongs to a batch which is planned to be started in period i. If this amount is greater then zero, the boolean constraint variable $carryoverflag[v, i]$ gets the value *one*, otherwise it gets the value *zero* (see below).

Following our intention to create a frame model where each assignment of the model variables represents a soltuion, we have to introduce now the essential problem constraints into the model.

Demand-originated Constraints The following constraints must be fullfiled between the finished-goods inventories, the production output and the demand:

– $\forall v \in Variants : estInv(v, 1) = initialInv(v) + wip(v) - demand(v, 0)$, [3]

[3] 0 is the actual time period with running production.

– $\forall v \in Variants \forall i \in \tau :$
$i > 1 \rightarrow estInv(v,i) = estInv(v,i-1) + production(v,i) - demand(v,i).$

Capacity Constraints Regarding the production capacity, the following constraints have to be fullfiled:

– $\forall i \in \tau : \sum_{v \in Variants} production(v,i) \leq productionCapacity(i).$

To ensure the the fullfilment of the inventory-capacity constraints, the following constraints have to be integrated into the frame model:

– $\forall v \in Variants \forall i \in \tau : estInv(v,i) \leq desiredProductInventoryMax(v),$
– $\forall v \in Variants \forall i \in \tau : estInv(v,i) \geq desiredProductInventoryMin(v).$

In analogy, for the part inventories, the constraints

– $\forall p \in Parts \forall i \in \tau : estPartInv(v,i) \leq desiredPartInventoryMax(v),$
– $\forall p \in Parts \forall i \in \tau : estPartInv(v,i) \geq desiredPartInventoryMin(v)$

have to be realized.

Availability Constraints Regarding the part availability, the following constraints are essential:

1. $\forall p \in Parts : estPartInv(p,1) =$
$min(initPartInv(p) - \sum_{v \in Variants} production(v,1) \cdot bom(v,p)$
$+ supplierCapacity(p,1), desiredPartInventoryMax(p)),$
2. $\forall p \in Parts \forall i \in \tau : i > 1 \rightarrow estPartInv(p,i) =$
$min(estPartInv(p,i-1) - \sum_{v \in Variants} production(v,i) \cdot bom(v,p)$
$+ supplierCapacity(p,i), desiredPartInventoryMax(p)),$
3. $\forall p \in Parts :$
$\sum_{v \in Variants} production(v,1) \cdot bom(v,p)$
$\leq initPartInv(p) - desiredPartInventoryMin(p)),$
4. $\forall p \in Parts \forall i \in \tau : i > 1 \rightarrow$
$\sum_{v \in Variants} production(v,i) \cdot bom(v,p)$
$\leq estPartInv(p,i-1) - desiredPartInventoryMin(p)).$

The minimum constraint (min) is used in constraint 1 and constraint 2, because it is possible that the supplier capacity can not be used exhaustively because of the existing limits of a part inventory. Constraint 3 and 4 ensure that enough parts are available for the planned production. They may be relaxed depending on the used storage policy.

Batchsize Constraints The batchsize constraints have to ensure that each sucessful assignment of the model variables represents a production output plan that takes into account the demanded production batchsizes of the different product types. This is guaranteed by introducing the following constraints into our model:

1. $\forall v \in \mathit{Variants} \forall i \in \tau : i > 1$
   $\rightarrow \mathit{production}(v,i) = \mathit{batchproduction}(v,i) + \mathit{carryover}(v,i-1) - \mathit{carryover}(v,i)$,
2. $\forall v \in \mathit{Variants} \forall i \in \tau : \mathit{carryover}(v,i) < \mathit{batchsize}(v)$,
3. $\forall v \in \mathit{Variants} \forall i \in \tau : \mathit{batchoutput}(v,i) = \mathit{batchsize}(v) \cdot \mathit{batchnumber}(v,i)$.

Constraint 1 embodies the dependencies between the production in a period and the part of it which is produced in full batches inside the period. Constraint 2 fixes the fact that a batch can overlap to the next period at most with $\mathit{batchsize}-1$ units. Constraint 3 finally ensures that batchoutput is in each case an undivided multiple of the corresponding batchsize.

Because we wanted to allow also the representation of problems where the number of overlapping batches is restricted per period, we have integrated into the model the constraints

- $\forall v \in \mathit{Variants} \forall i \in \tau : \mathit{carryoverflag}(v,i) = 1 \leftrightarrow \mathit{carryover}(v,i) > 0$

for the boolean constraint variables *carryoverflag*.

2.3 Problem Solving within the Frame Model

So far, we have introduced all essential constraints of the frame model. That means, if it succeeds to generate an assignment for all variables of the model, a solution for the modelled problem is found.

As expected we have found, that a lot of problems could not be solved in an acceptable time although we have integrated several search heuristics.

The basic idea of the search strategy follows the iteration approach of the MRP algorithm. The iteration runs along the planning periods. The output plans and estimated part inventories for a period i are computed based on corresponding values at period i-1. Our variable choice works in the same way: it chooses the plan and estimated inventory variables along the planning periods. Within each period, we first label the variables with the smallest domain, namely the *carryoverflags* . Then we label the variables for the production output plan. The most other variables within a period will be derived by constraints as soon as the output plan variables are ground.

For labeling of the output variables we make use of the production batch sizes and of the PreDemand (see below). We try the values for the output variables starting with a value derived from the respective desired inventory value and the *preDemand*. If this value violates the current constraint set, we jump with step width *batchsize* alternating up and down around the start value.

If values for early chosen variables cause a conflict in later steps, backtracking to the earlier choice points will occur. This requires a heavy computational effort. To overcome this problem we need to have a stronger propagation inside the model. One possible way to achieve this is the logical deduction and model integration of proper redundant constraints.

In the following section we describe several types of redundant constraints which we have developed and which we have succesfully used to speed up the problem solution process for a lot of problems due to the achived substantial search-space reduction.

3 Integrating Redundant Constraints

Redundand constraints are constraints that not explicitely occur in the "natural specification" of a problem. The addition of redundant constraints to a constraint model do not cut the solution space, but often the search space is reduced drastically.

During analysing problem models of the given problem class, we could identify a couple of dependencies which were not explicitely given in the frame model and which have the potential to cause an additional search-space reduction in early phases of search. In the most cases, the identified dependencies have concern with cumulations of constraint variables that are part of the frame model.

In the following we want to present the most important of them.

3.1 Global Constants

There are several constants implicitely given in a specified planning problem which can help for better characterizing the planning problem. The dependencies given in the initial specification have mostly local character. In contrast, the deducable constants reflect a more global view.

Capacity Potential The constant $CapaPot$ is the maximal capacity amount which can be let unused in every resulting plan regarding a given planning problem. $CapaPot$ can be calculated by the following formula:

$$\begin{aligned} CapaPot = & \sum_{i \in \tau} productionCapacity(i) \\ & + \sum_{v \in Variants} (initialInv(v) - desiredProductInvMin(v) + wip(v)) \\ & - \sum_{v \in Variants} \sum_{i \in \tau} demand(v,i) \end{aligned}$$

Global Predemand The constants $GPred(i)$ with $i \in \tau$ is a meassure for the minimum amount of products that must be taken from the stock in period i to fullfil the constraints of the given planning problem. For each $x \in \tau$,a constant $GPred(x)$ can be calculated in following way:

$$\begin{aligned} GPred(x) = & \sum_{v \in Variants} \sum_{i \in \tau} demand(v,i) - \sum_{v \in Variants} wip(v) \\ & - \sum_{i \in \{1 \ldots i-1\}} \sum_{v \in Variants} demand(v,i) + \sum_{i \in \{1 \ldots i-1\}} productionCapacity(i) \end{aligned}$$

Quite similar to the constants $GPred(1), \ldots, GPred(n)$ but stronger in some aspects of the target propagation are the constants $minProdSumForFuture(x)$ for $x \in \tau$.

$$\begin{aligned} & minProdSumForFuture(x) = \\ & \quad maxl(0, max_{q \in \tau : q > x} (\sum_{r \in \tau : r > x \& r <= q} \sum_{v \in Variants} demand(v,r) \\ & \qquad - \sum_{r \in \tau : r > x \& r <= q} ProductionCapacity(r)) \end{aligned}$$

Other Sums of constants Other constants ($availPartsCum$) that can be used to reduce the search space are the maximal amounts of available parts of each part type cumulated over the planning horizon. They are calculated on basis of the known supplier capacities, but they do not consider the fact that the part

stocks are limited in the normal case (see below). That's why, they are upper bounds for the amounts of parts that are planned to be used in each producible plan.

Besides the listed constants, we use other constants in our realization which are sums of existing frame-model constants:

- $demandCum(v,i) = \sum_{x \in \{1 \ldots i\}} demand(v,x)$,
- $demandSumCum(i) = \sum_{v \in Variants} demandCum(v,i)$,
- $prodCapaCum(i) = \sum_{x \in \{1 \ldots i\}} ProductionCapacity(x)$.

3.2 Cumulated Variables/ Cumulating Constraints

We introduce in the following several FD-variables which are set equial by constraints to cumulations of respective variables of the frame model. Together with the introduced constants, they bulit the basis for the formulation of redundant constraints. Doing the realization of the presented cumulation constraints, it is wise to use *recursive constructions* because only that way, we will gain a stronger propagation inside the solution model.

Cumulated Production and Cumulated Batch Production Regarding the production, cumulations of production variables with respect to the different product types are interesting for us on the one hand. On the other hand also the cumulation of the sums of production in the periods can be useful. So we introduce the following constraints variables together with the constraints for the calculation:

- $productionCum(v,i) = \sum_{x \in \{1 \ldots i\}} production(v,x)$,
- $productionSumCum(i) = \sum_{v \in Variants} productionCum(v,i)$.

In analogy, we introduce

- $batchproductionCum(v,i) = \sum_{x \in \{1 \ldots i\}} batchproduction(v,x)$,
- $batchproductionSumCum(i) = \sum_{v \in Variants} batchproductionCum(v,i)$.
- $batchnumberCum(v,i) = \sum_{x \in \{1 \ldots i\}} batchnumber(v,x)$,

for the production under batch view.

Difference Variables Other important variables for restricting the search in our model (see below) are $DiffCapaProductionCum(i)$, whereat $i \in \tau$:

- $DiffCapaProductionCum(i) = prodCapaCum(i) - prodSumCum(i)$

Accumulated Inventory Variables To facilitate the description of relations between demands, production capacities and inventory levels, we introduce for each period the constraint variable/ constraint:

- $estInvSum(i) = sum_{v \in Variants} estInv(v, i)$.

Besides the listed cumulated variables and cumulating constraints, there are others in our model realization which are constructed using the same principle. Among them, the variables $partsUsableCum(p, i)$ with $p \in Parts$ and $i \in \tau$ represent the amount of parts of type p which is available till the end of period i. Thereby, the given upper limits for the inventories are considered. Note, this and the fact that $partsUsableCum(p, i)$ are constraint variables ($i \in \tau, p \in Parts$) instead of constants are the main differences in comparsion with $availpartCum(p, i)$ (see above).

To limit the domains of $partsUsableCum(p, i)$ ($i \in \tau, p \in Parts$) in the initialization phase of the solution model, it is useful to set the following constraints:

- $\forall p \in Parts \forall i \in \tau : partsUsableCum(p, i) \leq availpartCum(p, i)$

In the folowing section, we give an overview of the most important redundant constraints which we have integrated into the frame model. They give a significant contribution for the propagation realized in SNS-APS.

3.3 Redundant Constraints

Copy Frame Model Constraints to Cumulation Variables Several constraints which are part of the frame model are valid in a similar manner for their cumulations. So we can introduce for instance the following constraints which describe, basically, the dependencies between the cumulated number of produced batches and the cumulated planned production output for all periods and for all product types:

- $\forall i \in \tau \forall v \in Variants$:
  $carryoverflag(v, i) = 0 \leftrightarrow productionCum(v, i) \bmod batchsize(v) = 0$,
- $\forall i \in \tau \forall v \in Variants$:
  $batchnumberCum(v, i) \cdot batchsize(v) \geq productionCum(v, i)$,
- $\forall i \in \tau \forall v \in Variants$:
  $(batchnumberCum(v, i) - 1) \cdot batchsize(v) \leq productionCum(v, i)$,
- $batchproductionCum(v, p) = batchsize(v) \cdot batchnumberCum(v, p)$.

Constraints Between Cumulated Production Outputs and Demands It is obviously, that – if the planning problem should be solved successfully – the cumulated demand for a product type must be smaler or equal than the cumulated production output of the product plus the available part of the inventory for each period. Otherwise, production can only be procecced in an amount which is smaler or equal than the sum of demands plus the usable free storage. These facts can be used for further reduction of the respecting search spaces:

- $\forall v \in \mathit{Variants} \forall i \in \tau$:
  $\mathit{productionCum}(v,i) + \mathit{wip}(v) + \mathit{initialInv}(v) - \mathit{desiredProductInvMin}(v)$
  $\geq \mathit{demandCum}(v,i)$,
- $\forall v \in \mathit{Variants} \forall i \in \tau$:
  $\mathit{productionCum}(v,i) \leq \mathit{demandCum}(v,i) + \mathit{desiredProductInvMax}$
  $- \mathit{desiredProductInvMin}$

If we take into consideration in addition the production capacities and the constants $\mathit{GPred}(1), \ldots, \mathit{GPred}(n)$, we can integrate into our model the constraints:

1. $\mathit{DiffCapaProductionCum}(n) \leq \mathit{CapaPot}$, where n is the last period of the planning horizon,
2. $\forall i \in \tau : \mathit{estInvSum}(i) \geq \mathit{GPred}(i) + \sum_{x \in \{i+1,\ldots,x\}} \mathit{DiffCapaOutput}(i)$
3. $\forall i \in \tau : \mathit{productionSumCum}(i) \geq \sum_{v \in \mathit{Variants}} \mathit{demandCum}(v,i)$
   $+ \mathit{minProdSumForFuture}(i) + \sum_{i \in \tau : i > p} \mathit{DiffCapaProductionCum}$,
4. $\forall i \in \tau : \mathit{productionSumCum}(i) \leq \sum_{x \in \{1,\ldots,i\}} \mathit{minl}(\mathit{desiredProductInvMaxSum}$
   $- \sum_{v \in \mathit{Variants}} \mathit{estInv}(v, x-1) + \sum_{v \in \mathit{Variants}} \mathit{demand}(v,x), \mathit{ProductionCapacity}(x))$

Briefly said, the constraints 2, 3 and 4 ensure that enough production capacity will be used if it is already known that in later periods not enought capacity is available on the one hand or if – on the other hand – it is already clear, that some existing capacity in later periods will not be used.

Constraints Between Production and Availability After realizing the additional constraints so far as described, we found a drastically improved runtime behaviour of the planning component. Some planning problems kept their bad run times. A new quality stage regarding this problems was reached by integrating redundant constraints between the production output and the availability respectively between their cumulation variables. Two of them are the following:

1. $\forall p \in \mathit{Parts} \forall i \in \tau$:
   $\mathit{partDemandCum}(p,i) = \sum_{v \in \mathit{Variants}} \mathit{productionCum}(v,i) \cdot \mathit{bom}(v,p)$,
2. $\forall i \in \tau \forall p \in \mathit{Parts} : \mathit{partDemandCum}(p,i) \leq \mathit{partsUsableCum}(p,i)$.

4 Sample Scenario

The simulation experiments for the proof of concept were based on an artificial, but realistic scenario. An OEM[4] produces three products (A, B, and C). Each product consists of three parts. While parts $p1$ and $p2$ are contained in each product type, parts $p3$, $p4$ and $p5$ occur only in products A, B, and C, respectively. The OEM has five suppliers, one for each part type. Further, a competitor of the OEM belongs to the supply network. This competitor produces a product X which also contains part $p5$. Unfortunately, the competitor also obtains $p5$

[4] original equipment manufacturer

from *supplier 5*. *Supplier 5* has aligned it's capacity with the demand from it's customers in a way that it's production system is almost completely utilized. To increase capacity, *supplier 5* would need to build a new production line. *Supplier 5* would not hesitate to do this if it would recognize a stable demand increase. But installing a new production facility lasts several weeks.

For arbitrary reason, competitors demand for $p5$ increases rapidly and suddenly. *Supplier 5* starts to build the new production line, but the demand increase starts six weeks before the capacity is adjusted. During this period, *supplier 5* cannot fulfill all orders. *Supplier 5* has to distribute it's production amount between it's two customers. A usual policy is to calculate delivery amounts in proportion of demand. Since the competitor increased demand, the proportion of OEMs demand decreased. The delivery amount decreases too. As a consequence, the inventory level of $p5$ at OEM falls, and in case of constant production rate released production orders for product C cannot start due to missing parts. This may lead to unused production capacity and to falling product inventories. In worst case, customer orders cannot be fulfilled in time. We investigated many different simulation experiments based on that scenario. Mainly we compared the question if use of an APS could help the OEM to mitigate the effects of this situation. We assumed that a supplier would inform the OEM in case of such demand increase, or any other kind of capacity shortage. That is, we defined the supplier capacity parameter of the OEM according to the expected supply situation [5]. This parameter is used by the OEM's APS system to calculate part availability.

4.1 Comparison of Simulations with and without APS System

The main result of the experiments is that the APS system was able to cope with the situation in a way that no customer order became late, no capacity was lost and part orders remained relatively stable. The APS computed a production plan like an experienced human planner. As soon as the material for product C was not available completely, the remaining production capacity was planned to be used to produce other products which were not affected by the shortage. These products were produced to stock, i.e. the inventory level was increased. As soon as *supplier 5* had delivered the missing parts, the production ratio between the products was changed. Then, the production of the stocked products was reduced and orders were filled from stock with reducing the inventory level to the initial value. In doing so, the remaining capacity was used to produce product C and to fill the product inventory up to optimal level. The sum over all products in product inventory remained constant. Despite this is a very important result from a simulation and process design point of view, it is not the most interesting point regarding the focus of this paper. That's why, we want to point out in the next section a comparison of two different APS system versions, namely one which uses cumulating constraints and a former version without these constraints.

[5] Later we will develop automated communication between the APS systems of different companies to enable automated, collaborative problem solving processes.

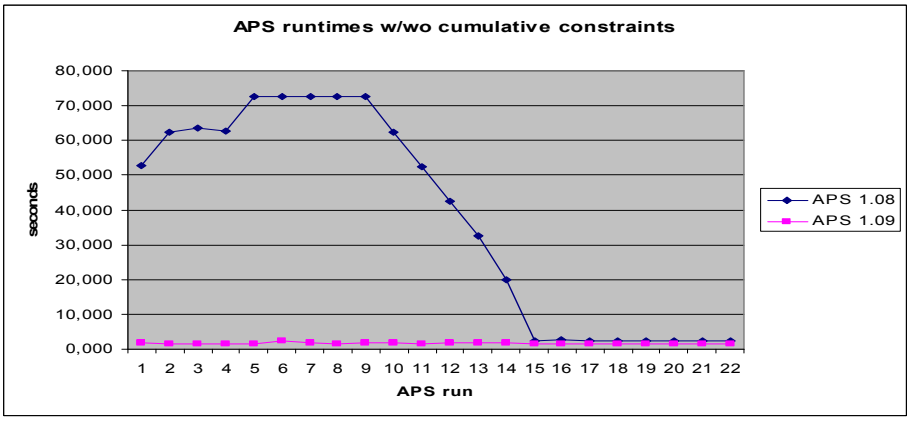

Fig. 3. APS runtimes with and without cumulating constraints

4.2 Comparison of Simulations by APS With and Without Cumulating Constraints

The scenario described above was simulated with several versions of the APS. As long as we did not use the cumulating constraints, the runtime of the APS was dissatisfactory. Even if the APS computed a plan in a time of few minutes, this is not sufficient for the application as planning node in a supply net simulation: during a simulation run with 10 companies using an APS once a week and a simulation time of one year, 520 APS runs occur. If one APS run would need only one minute, the over-all simulation time would be around 9 hours, which is not acceptable for an interactive process design approach. Hence, a runtime of few seconds instead of minutes is required.

The APS with cumulating constraints reached this goal for the given scenario as we show in Figure 3. The runtime of the APS without cumulating constraints is for the difficult planning problems greater than 60 seconds, with cumulating constraints the runtime is stable between one and two seconds.

Figure 4 provides a specific view to the planning results. The 3D charts show for each APS run and for each planning period of the planning horizon the deviation of the planned finished goods inventory from the optimal value. That is, the x-axis shows the several planning system runs the OEM performed. At each APS run, a plan for the production output and the estimated finished goods inventory will be created. The deviation of the estimated inventory from the optimal value is a measure for the quality of the plan. The higher the value is, the worse is the plan. The plan quality can vary over the periods of the planning horizon. Hence, along the y-axis for each APS run, the quality of the plan is indicated by the z-value.

The graphics show the problem which comes sooner and sooner to our OEM. At the first planning runs, the problem is foreseeable in late planning periods. With each planning run, the problem becomes more evident. At the time when APS run number 17 is executed, the problem is present. After that, it is solved,

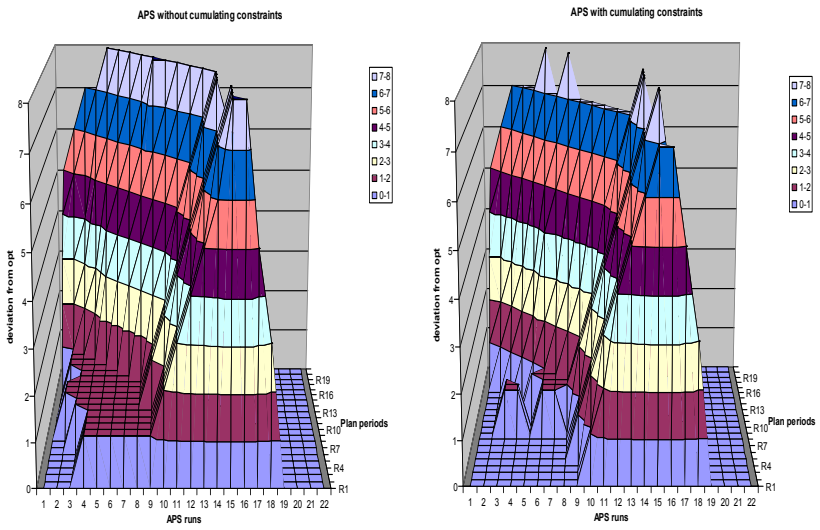

Fig. 4. APS planning results: deviation of planned inventory from optimal level

i.e. the supplier has adapted it's capacity and deliveres the missing parts. From this time, the quality of the plans is perfect, the deviation value is equal to zero.

The quality charts show another important effect of the redundant constraints: the solution quality could be slightly improved. The peaks of the deviations are not constantly as high as in the case without cumulating constraints, and in the left corner more optimal values are reached.

5 Conclusion

In this paper, we have introduced model and realization of a constraint-based planning system for solving batch-oriented production planning problems. This planning system is part of a supply network simulation tool which is dedicated to support effective designs and redesigns of information flow and planning processes in real supply networks.

Due to the integration to the simulation approach, the planning system had to fulfill strong execution efficiency requirements. On the one hand, runtimes of few seconds are the ultimate goal to ensure interactive simulation. On the other hand, the planner has to solve complex planning problems to fulfill customer demands while considering production capacity, warehouse capacity and part availability constraints simultaneously.

To reach the goal, we build a constraint model not only with basic constraints between production output, estimated inventories, production releases and part demands, but also with a lot of redundant constraints. The most important class of these redundant constraints are the cumulating constraints which bring

information about bottlenecks in later planning periods to the decision on plan values for early plan periods.

These constraints improved the efficiency of the constraint solving dramatically. The large number of possible scenarios does not allow to state a general ratio of runtimes, but for one specific simulation scenario including 22 runs of the planning systems, the runtime for difficult planning situations could be improved by factor 25.

Despite this enjoyable stage of the APS development, a lot of work is still open. One area for further improvements is for instance the extension and the more comprehensive embedding of cumulating constraints regarding the part availability. Our presented simulation results were still produced without comprehensive integration/realization of cumulating availability constraints. A second area for improvement is the search strategy. We will investigate if it is possible to replace the described sequential planning along the planning periods by a more general approach if enough redundant and cumulating constraints are introduced. For the case that this will be feasible, we expect a further improved quality of the plans.

Beside the further improvement of the planning behaviour, a lot of work remains regarding conflict analysis and conflict handling. Now, the APS is able to analyze first conflict situations and to communicate the analysis results to it's company agent in the simulation model. The company agent needs to be enhanced by dedicated communication and reaction mechanisms to solve the identified problems in an appropriate way. For example, it can activate it's capacity adjustment mechanism or it can look for alternative suppliers. Such mechanisms are described, for example, in [1].

References

1. Baumgaertel, H.: Distributed Constraint Processing for Production Logistics. In IEEE Intelligent Systems, 15(1): S. 40-48, January/February 2000.
2. Baumgaertel, H., John, U.: Combining Agent-Based Supply Net Simulation and Constraint Technology for Highly Efficient Simulation of Supply Networks using APS Systems. Winter Simulation Conference 2003. (2003)
3. Van Hentenryck, P.: The OPL Optimization Programming Language, MIT Press, 1999
4. ILOG: ILOG OPL Studio 3.6 User's Manual, May 2002, www.ilog.com. (2002)
5. ILOG: ILOG OPL Studio 3.6 Component Libraries Reference Manual, May 2002, www.ilog.com. (2002)
6. Lendermann, P., Gan, B.P., McGinnis, L.F.: Distributed Simulation with Incorporated APS Procedures for High-Fidelity Supply Chain Optimization in: Peters, B.A. et. al. (eds.) - Proc. of the 2001 Winter Simulation Conference, http://www.wintersim.org. (2001)