Scenario-Examples for IBP-Optimization

Examples of business scenarios solved by IBP Optimization
for Time Series and order-based Planning Model

Version 1.1
Jan 2021

www.sap.com

© 2016 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark for additional trademark information and notices. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.
National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

Contents

ABOUT THIS DOCUMENT – COMPARE SCENARIOS

This document helps you to classify scenarios for Supply Optimization in SAP IBP
Very often there are questions like “can supply optimization handle such a volume?”, or “what will be the runtime to solve a specific scenario?”
An answer to such questions is very difficult. The final behaviour of the optimizer depends on a lot of details. For example, a very large scenario can be solved very fast, if the constraints are easy to satisfy. But if we have very challenging constraints, a smaller scenario could need much more runtime.
As these details often are not known at the beginning of a project. There is no one, who can answer questions like above upfront in an honest manner. Estimations which are too conservative (leaving a large buffer for example for the runtime) are also not very helpful. And setting up prototypes is either inaccurate or causes too much effort.
The best and most convincing alternative is to compare the current project with already realized projects in IBP. On this level you don’t need to investigate all details, but you get very soon an impression about the planning of your scenarios, if you look into some similar scenarios. This is the approach we want to establish with this document.
The following scenarios and numbers are based on the experiences we made with different customers, partners, and projects. The information of this document was collected during a common phase of cooperation. The descriptions here do not cover the latest state and cover only a small subset of the IBP customers. The figures are rounded so that you get a good impression without sharing critical details of the customer scenarios.

If you are not experienced in optimization of supply chains, we recommend to look first into the first two chapters, before studying the different scenarios. In these opening chapters we describe what is characterizing a scenarios and which attributes of a scenario drive its complexity and the runtime to solve it.

FUNDAMENTALS ABOUT CLASSIFYING PLANNING SCENARIOS

If you want to classify a scenario of supply planning, there are a lot of dimensions to consider. But not all decisions of a specific configuration have the same impact on the overall performance and planning quality.
The following selection will highlight some important trade-offs to consider. If you compare your scenario try to define your needs and compare it against the description of the published scenarios.

Details vs. Runtime

If you set up a model of your supply chain, there is always the question, which details you want to model. Whenever you add more details to your model, you will need more runtime to find good plans. Examples for such decision are:

- Do I plan in days or weeks?
- Do I model all products, or can I concentrate on the important ones?
- Do I need to consider lot sizes, because they are computation extensive?

If we compare different scenarios for the optimizer, the question is if you can work with the same level of details or if you need more details, or can you even work on a more aggregated level.Some decisions about details have a direct impact on the model, which you configure in IBP, for example which periods you use for planning. For some other decision you can control the optimizer, if it shall care about certain details. You can control that, by assigning no or low cost. In general, you can also define a remaining gap for the optimization run, to express that certain decisions are not relevant for the optimizer. All these ways can significantly increase the performance by reducing the detail-level for planning.

Runtime vs. Quality

The optimizer can in principal compute the global best solution. At the beginning of its runtime you will see the first constructed solution and some major improvements. But with more and more runtime you will see only rare and small improvements. The most runtime will be needed to check, that other options will not improve the plan.
For different scenarios it might be different, how many runtimes you want to invest to improve your plans further. This might especially depend on questions like

- Will I automatically transfer to the result to the execution systems?
- Will I check the plan and manually correct it?
- Do I only need a rough simulation?

The optimizer offers the option to define the maximal runtime and the remaining gap to enforce a stop of the planning, when the planning quality is sufficient. In the presented scenarios you see some very long runtime. Often the user decided to use the complete available time interval for planning, to benefit even from the last little improvement of the planning progress.
If you use the optimizer not only in batch, but also as interactive planning tool, shorter runtimes become much more essential. In this case the runtime limit is usually much shorter.
We want to mention, that you carefully should set the runtime and check the solution progress of your scenarios. In many cases the optimizer finds a lot of improvements in the beginning. But the longer runtime is more and more used to prove the optimality of the solution and is not leading to further improvement over time.

Global planning vs. decomposed planning

The last criterion to characterize a scenario, is perhaps the most important one: Can you split the model into sub-problems, or do you need to solve it in one global planning run?

There are several reasons why a split into sup-problems can be superior

- The first reason is the time needed to optimize the plans. In nearly all cases it is much faster to compute the sub-problems one after the other and putting the global solution together than solving the complete global model
- The smaller sub-problems allow you to model more details. A good example are lot-sizes. In the sub-problems you can solve them very detailed, whereas for a large global scenario they might be too complex.
- Splitting up the overall scenario allows you also to solve the sub-problems with different settings or at different times. For example, you might solve a sub-problem several times a day, perhaps the deployment. But you want to plan the production only during the night.
- Defining a schedule, when to solve which sub-problems allows you also to define what is considered a constraint from the surrounding sub-problems, when you solve separate sub-problems. With this technique you can push certain alerts like missing product quantities in a certain direction, for example to the supplier or to the forecasts.

When deciding if and how to split the planning problems into several steps, you should think about which details of the model need to be solved when. A very successful approach is to look at the planning situation, which was in place before the introduction of SAP IBP. Usually you can see the natural breakdown into different sub-planning-scenarios if you identify the different responsibilities of different planners in the company. We see usually two types of organisation:

- Different planners are responsible for different regions or different products
- The different planners care about the different distribution-levels of planning

These approaches are very good hints to set up a similar process for automatic planning, if you additionally tackle the pain points of the existing process. If you stick to this former breakdown this is usually much more successful than starting with a global automatic planning process, which would fail because of the inherent complexity.

Beside of the business criteria, we already mentioned the performance as reason for a split. The usual recommendations are, that you should think of decomposition, if the global scenario would exceed 10 continuous million variables, 30,000 binary or 20,000 discrete variables.

In IBP you have the possibility

- To define the decomposition explicitly by sub-networks or
- To let the optimizer internally decompose the given scenario into product families.

If you choose the automatic decomposition by the optimizer, you can specify if you want to start with a global pre-allocation run to guarantee a global balancing. before you solve the details in the sub-problems. scenario in one planning run.

DRIVERS OF COMPLEXITY

The following chapters give a short overview about the main drivers for complexity in supply optimization and how you could handle these challenges. These explanations will also help you to understand the numbers in the descriptions of scenarios.

Scenario Size – Horizons & Planning Periods

The optimizer is internally mapping the business scenarios into mathematical variables and formulas (constraints). The number of variables is our main indicator for the size of the scenario. Increasing the number of variables usually also implies a longer runtime to optimize the scenario. Therefore, it is crucial to reduce the number of variables whenever possible, without losing the required details of the planning model.

Without explaining details about the transformation from business to maths, I want to mention, that we need some variables for every period. For example, for every combination of product and location we need a variable to express the projected stock at the end of the period. Similar we need for every option to transport, produce or purchase a variable to express how much of a certain product we transport, produce or purchase. Given that, the number of planning periods has the highest impact on the number of variables.
There are mainly two options to restrict the number of planning periods:

- In the Planning profile you can specify the number of periods (horizon). Beyond this horizon the optimizer is not planning at all and is ignoring all kind of demands and constraints.
- Alternatively, you can define the granularity of your planning periods: The optimizer can plan in days, weeks, technical weeks or month[1]. But even if you decide for a small time period granularity, you can use Time Aggregation & Telescopic Plg. to reduce the number of the planning periods for the optimizer. This feature allows you also to plan the beginning of your planning horizon in days, to switch to weeks and finally to plan the end on monthly periods.

Beside the number of periods, we often see a significant number of customers maintained in the system. If we have on all these customer locations demands, then we usually allow for all these demands to delay the delivery. Internally this means that we need to introduce additional variables to express the delay. But it is not only one variable for delay. For every demand in every period we have to add a variable to express a delay of one day, another variable to express a delay of two days, a third….. and so on. Because of that it is very helpful, if you can define the optimizer an upper limit, how much delay you allow. Don't plan with the largest seen delay ever but use a number after which you usually don't want to see a delay.

Finally, fair share is a feature which drives the number of variables significantly. Internally the fair share is controlled by a lot of variables which indicate which part of a quantity is already satisfied. With the option in how many steps you fair share, you have an easy control, to reduce the number of variables to the required amount.

[1] Currently OBP is only offering days. Also, the telescopic planning is only available for Time Series Model at the moment.

Lot-Sizes – Mixed Integer Programming

To plan supply chain scenarios the engine is creating and solving a linear program. With this kind of representation all variables are continuous. They can have any real value and any fraction. For many use-cases that's perfect. But often you want to express that you handle pieces or that you want to transport, produce or purchase products in batches of a certain size, you define lot sizes.
To solve such scenarios with lot-sizes the optimizer is still starting with the linear program and computing a solution with continuous real numbers. In the following steps the optimizer is "rounding" the solutions to integer numbers representing the lot sizes. But this is not a single easy step. Instead you need to check if the nearest rounding is the best solution or not. Assume the example we have a fractional solution of 4,3 and we must round it to pieces (rounding value 1). Then the optimizer might come up with three options: 4 as the nearest solution; all solution >= 5 and all solutions <=3. To see, what is the best value, the optimizer has for each three alternatives to evaluate, what is the best possible solution under the additional constraint =4, >=5 or <=3. To do so the optimizer must consider all the other lot sizes and must check all combination of cases, until the best solution is found. This leads to large search tees, which usually need much more time to solve than the first linear solution with fractional numbers. Modern solvers like Gurobi in IBP have many techniques to speed up this search process as good as possible, nevertheless the number of integer variables for lot sizes dominates usually the runtime.

In IBP you can define on the planning profiles for how many periods you want to consider the minimal and the rounding lot sizes to control the runtime of the solving process.
To show the impact of the lot sizes we show in the following scenario descriptions, which time is needed to find the linear fractional solution and how long the customer was running the optimizer to find the best-rounded solution.

If lot sizes are important for a large horizon, usually the product-decomposition is used to split the overall scenario in small pieces and to do a perfect rounding for these small sup-problems. Here is the idea to focus on this decision locally, without computing every little effect of the lot sizes on other parts of the supply chain, even if it might exist.

Beside the pure number of variables, the following scenario descriptions will also contain the number of binary and integer variables needed to control the lot sizes. Try to keep the horizon in which lot sizes are considered as short as possible. Especially if you do midterm calculations, rough simulations or S&OP-Planning try out the performance without lot sizes and think about the importance of the lot sizes on your planning granularity.

Algorithm

To solve the linear part of the program and the fractional solution, there are alternative algorithms available. In most cases, especially if we must consider lot-sizes we recommend the dual simplex method, which also computes information which is used for solving the integer variables of lot sizes. In some scenarios without lot sizes at all the Barrier algorithm might be a better choice for better performance.
The following description of the scenarios will mention, if barrier is used.

Numeric

There is one last important aspect reimaging for optimization performance. It is perhaps the most difficult one to understand and to judge: All the optimization is done on CPU's and is using their floating-point representation to compute numbers as fast as possible. In our cases that's double

numbers (usually 64-bit floating-point numbers). Often the precision of this numbers is not enough to do all computation steps to solve a supply chain scenario[2]. The optimizer is detecting and trying to solve situations. It is even repairing intermediate results, where possible. But as soon as there are a lot of such numerical issues, they might have an impact on the overall performance of the optimization.
In most cases it is sufficient to follow the warnings and instructions in the log of the optimization run, to switch on the recommended additional features to transform your supply chain in numerically more robust optimization model. Sometimes it might also be required to adapt parts of your business model, for example to measure a product like coal in tons as units instead of grams.
The following descriptions of the scenarios will also contain information, if special numerical features are enabled, and other numerical remarks as well.

Time Series and Order-Based based Planning

The optimizer is available for Time Series based planning and for Order-Based Planning. It is the same optimization engine and it is behaving in the same manner. Depending on the features, there are a few differences, how features large implemented. For example, the way of explaining planning results (Explanation tables versus Gating-Orders).
As we focus in this document on a rough classification of scenarios, the minor differences of such features don't matter. The approximate complexity, the number of variables and the runtime are very similar, independent of using Time Series or Order based Planning.

[2] For more background information and options to influence the numerical behavior have a look to https://launchpad.support.sap.com/#/notes/2922352

SCENARIO: AUTOMOTIVE

The following figures are from a long-term automotive production planning in months. The most important characteristics were modelled as separate products. Lot sizes were not considered for the rough capacity planning. The different number result from different version with different granularity of constraints.
As remarkable feature the extension capacity was used. That means the optimizer has to decide, if and when additional capacities required, usually additional shifts.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Automotive 1	30	38	22,600	38		6,800,000			-	-	-	-	-			1:33 h	
Automotive 2	30	38	22,600	42		7,400,000			-	-	-	-	-			1:06 h	
Automotive 3	25	38	15,000	49		5,900,000			-	-	-	-	-			7 min	

SCENARIO: BEVERAGE

In this scenario we combine a unique production plant with a large distribution network. Focus was on the service level of the DC's for a very long planning horizon. To reduce the planning volume the time-aggregation is used
Because of the limited time window for planning the lot sizes are currently not, which is in this case an essential drawback.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg. Time Aggregation	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Beverage	101		7,300	800		100,000,000	-	-		x				53 min		53 min	

SCENARIO: CHEMICAL INDUSTRY

In this scenario of chemical industry, the optimizer has to solve a multi-level production and distribution scenario. IBP is used for the tactical and the operational planning level as well.
Because of the size of scenario, the first try would be to split it into different sub-networks, for example for different product groups. But many shared components and semi-finished goods made it hard to define such a split into sub-networks. Because of that dependencies the decision was mad to have one optimization run without sub-networks and to decompose the scenario inside the optimization engine by the Product Decomposition. The engine is splitting the overall scenario internally into 18 different sup-problems, which partly overlap in common components. Because of performance and the complex dependencies, it was necessary to run the product decomposition without a pre-allocation run. (With the general availability of the product decomposition, also for such scenarios a linear pre-allocation shall be available, which would also in such scenarios allow a better balancing and propagation of adjusted values.
To model the scenario also very different units of measures are required and lead to many numerical challenges. Both parameters for the numerical focus and for the scaling, were set to the maximal value to take care of the numerical properties. Together with SAP Expert Consulting some parts of the model were reworked to finally come up with a numerical stable behavior.

	Business Model				Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Chemical production & distribution	464	250	44,500	160	12	22,000,000	274,000	39,000	x	-	x	-	-	-	-	5 h	Expert consulting supported because of numerical challenges.

SCENARIO: CONSUMER PRODUCTS MANUFACTURER

In these scenarios the focus was on a very efficient production. The demands could be propagated to the plants via the supply heuristic. Then the optimizer could compute the optimal production plan. The complete supply chain was divided in many planning runs, each one for a single plant. The main challenge was the small planning windows which allow at most 3h planning per plant. In parallel the customer wants to respect the lot sizes for as many periods as possible. The product decomposition was used to compute these lot sizes with a high accuracy. The overall problem was divided in 180 sub-problems. To keep the balance between all these product-groups a pre-allocation is used before decomposition.

The different plants showed different kinds of complexity. Beside the challenging lot sizing and grouping of the products an unusual feature was the extensive use of min utilization on the production resources. Not all instances could be solved within the goal of 3 h runtime so far. Current benchmarks indicate that optimization as a service (planned for IBP 2108) will bring the required performance to keep the runs below 3h. The table shows the most challenging plants. Example Ia and IB are from the same plant on different days.
The setup of the product decomposition was done together with SAP Expert Consulting.

	Business Model					**Math Model**			**Features**					**Runtimes**			**Remarks**
Name	Locations	Customer Loc’ s	Nb of Products	Periods	Lot size periods	Continuous Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Consumer Products manufacturer I a	20	36	347	401	210	9,600,000	21,000	42,000	x	-		-	-			4:50 h	176 of 180 subproblems solved to optimality 4:15h for pre-allocation with lot sizes
Consumer Products manufacturer I b	20	36	347	401	210	8,600,000	20,000	40,000	x					2:30 min		3:30h	All 181 subproblems solved to optimality
Consumer Products manufacturer II	18	11	1657	401	210	8,200,000	97,000	195,000	x					3 min		3:55h	333 of 348 subproblems solved to optimality 3:25 for pre-allocations with lot sizes

SCENARIO: CONSUMER GOODS

For this consumer good producer, the optimizer has to solve a combined multi-level supply chain and distribution scenario. Many components and semi-finish products were used for very different finish products. Because of that a split into sub-networks could not be found. Instead the internal product decomposition was used, and the common components were re-computed in every sub-problem, in which they occur.
The scenario was too large, to run a pre-allocation with lot-sizes before the prod-decomposition. So the current setting is running without pre-allocation. As soon as the feature will be available in IBP it might be a good idea to use the linear pre-allocations without lot-sizes and to solve the lot-sizes only during product decomposition
SAP Expert consulting was involved to check if a split in sub-networks is possible and in setting up the product decomposition.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc’ s	Nb of Products	Periods	Lot size periods	Continuous Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Consumer Goods	324	26	21,700	37	36	20,900,000	-	812,000	x			-	-	-	-	4:10min	187 of 248 sub-problems solved to optimality

COLLECTION: FOOD INDUSTRY

The following tables shows figures from planning runs in the food industry done by several customers. The numbers show, how the runtime is increasing as soon as you are considering lot sizes. The different rows show also how powerful an approach is, which is dividing the complete scenario into several small steps. So you can relax the lot sizes for a lot of planning steps and finally you consider the lot sizes for some important sub-networks with a short horizon, like shown in scenario 6.
If you consider lot-sizes it is often a good approach to leverage the product-decomposition, the telescopic time-aggregation or both, to find good solutions during an acceptable run-time.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Food 1	536	6	7,600	109		7,400,000	-	-		x				3 min		3 min	
Food 2	90	70	16,300	75	12	22,200,000	9,800	9,500	x							8:50 h	Using heuristic result as stat solution
Food 3	68	60	8,500	75	12	1,600,000	270	2,300	x	x						4 h	
Food 4	162	1800	5,300	104	0	30,300,000	-	-						22:20 min		22:20 min	
Food 5	135	0	4500	96	0	11,800,00	-	-						7 min		7min	
Food 6	26	28	2,800	18	13	240,000	11,000	-								2:35 h	

SCENARIO: DEPLOYMENT FOR FOOD INDUSTRIES

The customer started to use the optimizer for their deployment planning as a first step, before production planning will follow. To guarantee fast and fresh deliveries the network has a complex five-tier structure. Because of the fluctuations in demand and in supply, a robust deployment automation is required to balance shortages and excesses, which can arise at the same time inside the different parts of the deployment network.

For the success of the deployment optimization it was important to separate it completely for the production. To archive that several BOM’S had to be restructured, so that parts the production can be handled fixed, even if final products are also used for some production steps as components.
For the deployment an automatic sot setting was set up. Based on the number of active lanes inbound and outbound at each location, to ascertain the role of each location in 3-5 tier network w/ multi-sourcing, the cost rates for non-delivery or inventory target violations were adapted. This allowed an automatic adaption of the cost to the changed or added lanes. This approach avoids an error free run, without the risk of missing or wrong cost-rates.
One corner stone of the implementation was a comprehensive test-strategy: Two data sets with 25% and 100% of the planning volume were always available in the tests system and allowed the agile adaption of the data model, the cost generation and the customizing of the optimization.
The complete planning run over the complete network needed at the beginning 5-10h runtime, caused by the many integer variables for rounding values. By introducing the product decomposition this could be reduced to 20 minutes. Parts of the network can be solved separately with a few minutes’ runtime. Over the day planner do approximately 80 runs for the sub-networks including also customer locations and a reduced horizon. Fair share is enabled with 20% accuracy on the demand fulfilment and 10% on the inventory targets.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc’ s	Nb of Products	Periods	Lot size periods	Continuous Variables Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Complete Network	270	-	10,400	60	14	8,100,000	20	440,000	x	-	-	-	-			17:30 min	63,000 lanes
Subnetwork ex. 1	74	27	8,000	22	14	1,540,000	33	58,000	x	-	-	-	-			1:45 min	6,000 lanes
Subnetwork ex ,2	91	37	507	22		563,000	0	20,500	x	-	-	-	-			3:26 min	3,900 lanes

COLLECTION: LIFE SCIENCES

The following figures are from some health care companies. The figures show two interesting aspects

- Dividing your supply chain in separate planning steps allow you fast and flexible way to re-plan certain parts quite fast
- If you really plan to optimality without a limit for the runtime, the resulting runtime can vary, dependent on the concrete constraints. Similar input sizes can lead to different run times. As alternative you could add a limit on the runtime or gap. Usually the solution is not improving much while proving the optimality of the solution.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Life Sciences 1	770	23	1200	105	26	2,700,000	2,500	6,5000								1 min	solved to optimiality
Life Sciences 2	770	27	2400	105	26	20,000,000	7,200	18,000								16 min	solved to optimiality
Life Sciences 3	780	24	1000	105	26	5,800,000	52	150								63 min	solved to optimiality
Life Sciences 4	770	20	1800	105	26	600,000	368	775								20 sec	solved to optimiality
Life Sciences 5	15	9	16	93	80	730,000	1604		x							70 min	
Life Sciences 6	15	105	172	93	80	18,100,000	20233		x							3 h	
Life Sciences 7	14	79	6	93	80	740,000	868		x							3 min	
Life Sciences 8	73	36000	7500	102	102	8,200,000	40,000		x							23 min	

SCENARIO : MANUFACTURING

This manufacturer is using the optimizer to organize it purchasing. The planning is split into two steps which are also modelled as separate optimization runs

- Tactical purchase planning
  In a rough planning run the requirements on certain components is computed. These results are used to select the suppliers and to negotiate the right contingents. Planning horizon is one year in technical weeks

- Operative purchase planning
  The optimizer is used to decide in detail which products are purchased when and how much. Again, the planning horizon is one year in technical weeks, but now with a focus on the first six month. Now details of the target inventory and the max inventory are driving the purchasing.

The explanation of Time Series was used, consuming 15 minutes of the runtime. During the project numerical issues occurred and required to set the numerical focus and scaling. SAP Expert consulting supported this process. As the both runs are limited by the same runtime the figures look quite identical.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc’ s	Nb of Products	Periods	Lot size periods	Continuous Variables Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Manufacturing tactical	80	-	4,100	52		5,000,000	510,000		-	-	x	-	-			3:05 h	
Manufacturing operative	80	-	4,100	52		5,000,000	510,000		-	-	x	-	-			3:01 h	

SCENARIO: SEMICONDUCTOR

In this scenario a semiconductor company is planning their business for 2 years in weekly periods. The optimizer is used for two different horizons, the mid-term planning (close to 1 year) and the long-term planning (over 2 years). Because each finished good is sold in different variations for each customer (resulting in a lot of different customer products), the resulting supply chain model is very complex. Each customer product / finished goods combination has its own Supply chain network attached to its background. Another issue is the sharing of resources for different subnetworks, which can be solved by pre-allocating the capacity per subnetwork and modelling the resources as separate instances. The main challenge was to model all customers, which lead to an optimization model of a lot of variables.
As no lot sizes were required, barrier could be used to solve the linear program. To handle the extreme number of the customers the planning was divided into 11 independent planning runs. The following table shows the KPI's of some of these runs.

	Math Model					Features											
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Semi Conductor I	612	72,000	24,000	117	0	166,000,000	-	-	-	-	-	-	-	11:00 h	11:00 h	11:00 h	Barrier
Semi Conductor II	394	7,100	13,000	117	0	13,470,000	-	-	-	-	-	-	-	12:55 min	12:55 min	12:55 min	Barrier

To reduce the long planning time of some scenarios, the following steps were done to reduce the number of variables.

- The maximal lateness was reduced from 50 to 14 weeks
- The overall run was split into a propagation of demands at the customer locations to the DC's, the essential planning run and a final assignment to the demands. With this approach the supply planning run needs only to care about customer locations with a multi sourcing. As mentioned above, the finished goods are split up into different customer products, which is not happening with the aggregation on the DC anymore.
- Parallelization with 4 Threads was used (which is only available for the Barrier method)

The following table shows the results for the lager former planning run Semi Conductor 1, containing 50% of the original products. The propagating of the customer locations leads to such a great reduction of variables, that it is planned to provide it as a new feature of the optimization engine to all users of optimization.

	Business Model					Math Model			Features					Runtimes			Remarks
Name	Locations	Customer Loc' s	Nb of Products	Periods	Lot size periods	Continuous Variables Variables	Binary Variables	Integer Variables	Product Decomposition	Time Aggregation & Telescopic Plg.	Numeric	Quotations	Fair-Share	Without lot sizes	Fst. Solution	Final	
Semi Conductor I*	616	32700	12370	117	-	33,800,000	-	-	-	-		-	-	18:20 min	18:20 min	18:20 min	Parallel Barrier, Customer Propagated

This scenario is a very good example, how adaptions of the model, like the maximal lateness or modelling of customers, can lead to much less variables and much shorter run-times. Combined with the decision not to consider lot-sizes, it shows also how you can keep the basic relations in your scenario and to reduce the runtimes by ignoring the minor details of the model.