Generation of predictive configurations for production planning

Tilak Raj Singh[1] and Narayan Rangaraj [2]

[1]Production Tools (IT), Mercedes-Benz R& D India, Bangalore

[2]IEOR, Indian Institute of Technology, Bombay, Mumbai

Configuration workshop 2013, Vienna

August 29-30, 2013

Outline

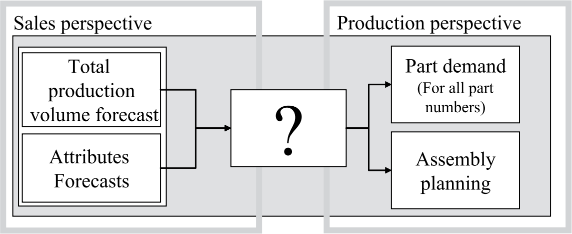
Outline

Motivation

- In mass customized product (e.g. Automotive), customer can make choice over large number of customizable attributes (options, accessories,..)
- Variety generated through assembly of multiple attributes
  - Attributes taking on different values
  - Not all attribute combinations feasible
  - Millions of feasible configurations
- For order fulfilment (e.g. ATO), demand planning of large number of components and parts need to be done much before the actual customer order.

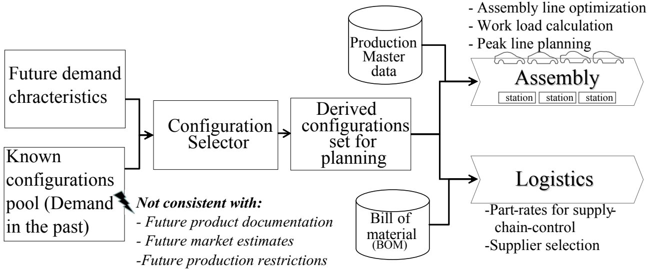
Motivation

- Some aggregate planning estimates from sales can be used
  - Total production volume (e.g. 3000 Type A car, in 04/2014 at Plant B)
  - Key attribute selection rate (Navi=50%, Sunroof 30% etc.)
- Starting for sales estimates, How to derive detailed part / component level demand? (e.g. Bumper, Wire-harness, Seat)



Current situation: Estimating future demand through extrapolation of past customer demand

- One way to get configurations for future production, is through extrapolation of configurations produced in the past
  - How to account engineering and market changes?
  - Need methods for New Product Projects (e.g. Hybrid, Electric etc.)



Outline

1. Motivation
2. The planning Problem
3. Input data and its characteristics
4. Solution Approach
5. Conclusion & Future work

Some problems in variety management

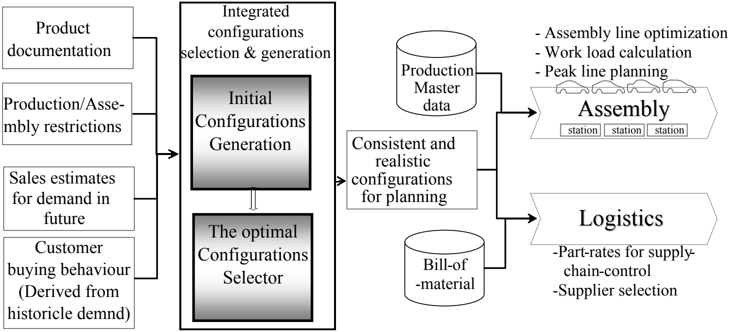
- How do we plan configurations that we will build, with such huge variety?
  - Need configuration level forecasts for various purposes
- How do we account for product changes in fast changing technology?
  - Product documentation and rules governing configurations
- How do we make use of past demand data?
  - What level of aggregation?

The problem

Given : (1) product documentation (2) market estimates (3) customer behaviour and (4) assembly restriction:
Generate valid configurations then select optimal ones in order to propose a production plan

Proposed Framework : Calculation of consistent order sets as a foundation for efficient production planning

- Consider data from various sources (e.g. Development, Sales, Production) and try to produce configurations which reflects target input characteristics in best possible way



Outline

Attributes

Configurations can be represented as 0-1 vector over product attributes

- Country of sale
- Engine- diesel, petrol, turbo, etc.
- Features like sun roof
- Production related (no data from sales, but needed for planning)
  - Plant where production takes place
  - Regulatory laws
- Typical numbers
  - 100-200 attributes from sales (known target selection rates)
  - 500-1000 attributes overall, all of which need to be planned for, eventually

Product Documentation

To accommodate variety and use of product data for different planning, product can be documented at the level of feature/attribute list and part list (Flat BOM)

Example: attribute list from product documentation

Attribute	Name	Relation	Rule	Description
1	climate control	$\rightarrow$	$(2)\wedge(3\vee 4)$	attribute 1 only with attribute 2 and at least 3 or 4

Example: part list

Sub-mod.	**POS**	**PV**	**Part**	**Name**	**Relation**	**Rule**
1000	100	50	$part_1$	Radiator	$\leftarrow$	$1 \wedge 2 \wedge 3$

Transforming rules to a set of constraints

Attribute	Name	Relation	Selection Rule
1	Rear-view camera	$\rightarrow$	$\neg(4 \vee 5) \wedge (\neg 6)$
2	Parking Assistant	$\rightarrow$	$(1) \wedge (\neg(4 \vee 5))$
3	Air Bag	$\leftarrow$	$(1 \vee (4 \wedge 5))$

$$\begin{bmatrix} 3 & 0 & 0 & 1 & 1 & 1 & 0 \\ -1 & 3 & 0 & 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} \leq \begin{bmatrix} 3 \\ 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Constraints from product documentation (Rules) as linear inequalities

$$\mathbf{B} \times [y] \leq \mathbf{b} \tag{1}$$

Where y_i is 1 if attribute i is selected in configuration else 0

Input data and characteristics

Sales and marketing estimates

- Total production volume (e.g. 3000 Type A car, in 04/2014 at Plant B)
- Single attribute selection rates (e.g. Navigation=50%, Sunroof=30 %)

Customer demand characteristics from past orders

- Joint selection rate of attributes (e.g. P(1,2)=25%)
- List of attribute combinations whose selection rate is time invariant (e.g. Expensive interior package with high end music system)

Production restriction

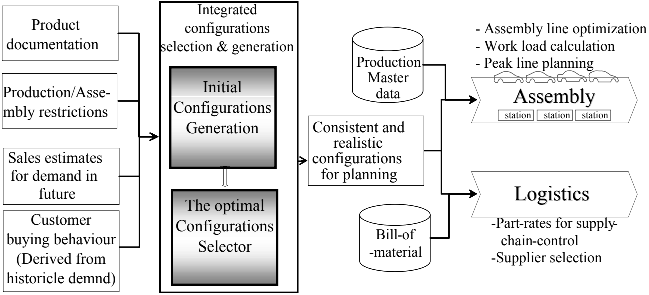
- Capacity limitations on parts (e.g. diamond grill less than 30%)

Outline

The solution Framework

The problem

Given : (1) product documentation (2) market estimates (3) customer behaviour and (4) assembly restriction:
Generate valid configurations then select optimal ones in order to propose a production plan



Structure of the decision

- $i \in I$, set of attributes in product - say 1000 in number, but about 100 may be specified
- $j \in J$, set of configurations to be selected - say K = 3000 in number for a typical plan for a model
- Does attribute i belong to the configuration j finally selected?
- 0-1 variables y_{ij} : 3000000 of them (large number)
- We do know something about the proportion of i's in the final configuration set (d_i)

Nature of the problem

- Minimize ($\sum_i |\sum_j y_{ij} - d_i|$) Subject to each y vector being a feasible configuration set (a set of linear inequalities/equalities defined from product documentation)
- General structure combinatorial optimization problem (Integer programme), with a very large number of variables that is quite difficult to solve

Another possibility

- List all possible configurations with the given number of attributes
  - This runs into the millions! Much larger than the previous formulation
- Define variable $X_j = 0$ or 1 depending on whether it is selected or not
  - Huge number of integer variables
- Surprisingly, this way of thinking is still useful!

New Formulation

New Formulation

$$\text{Minimize } \sum_j C_i | \sum_j |A_{ij}X_j - d_i|) \tag{2}$$

$$\text{s.t. } \sum_j X_j = K \tag{3}$$

$$X_j \text{ is } 0, 1 \tag{4}$$

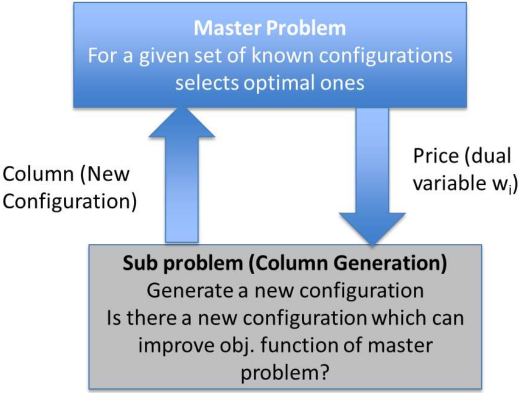
- Here, A_{ij} is 1 if attribute i is present in configuration j, and 0 otherwise.

$$A_{ij} = \begin{array}{c c} & \begin{array}{ccc} c_1 & .. & c_j \end{array} \\ \begin{array}{c} 1 \\ .. \\ i \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

- C_i is mismatch cost associated with attribute i
- Problem has a simpler structure, but the number of variables is in the millions

CG Algorithm Structure

- Number of variables (columns of A) huge
- Can generate column A_j by some "oracle" that can answer the question, "Does there exist a column with some property?" If so, the oracle returns one



Lagrangean approach using column generation

The master problem

$$\text{Minimize}\left(\sum_j C_i * Z_i\right) + \lambda\left(\sum_j X_j - K\right) \tag{5}$$

$$Z_i \geq \sum_j A_{ij} X_j - d_i \ldots \forall i \tag{6}$$

$$Z_i \geq d_i - \sum_j A_{ij} X_j \ldots \forall i \tag{7}$$

$$x_j \in 0, 1 \tag{8}$$

- Approach:
  - Start with a possible set of X_j variables (may be more than K)
  - Solve the LP relaxation of the problem above - decide which of those X_j's are 1
  - How do we know if the current selection of configurations is good?

The sub Problem: configuration generation problem

The subproblem

$$\text{Maximize } \sum_i w_i * y_i \tag{9}$$

$$\text{s.t. } B[y] \leq b ... \forall i \tag{10}$$

$$y_i \in 0, 1 \tag{11}$$

- This generates a possible new configuration j
- If this configuration j satisfies

$$\sum_i A_{ij} + \lambda - \sum_i w_i * y_i < 0 \tag{12}$$

then configuration j enters the pool.

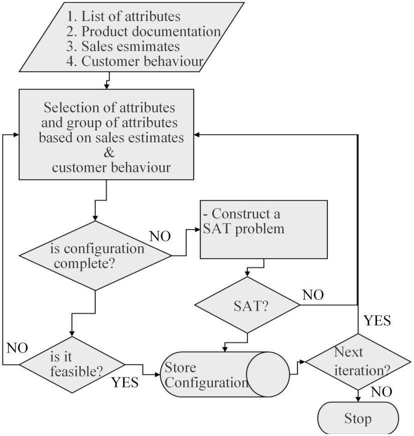
- Dual costs are re-computed and the process terminates when no more configurations are found to be worth taking in

Procedure, Extensions

- The master and dual problem may have to be solved multiple times
- Sub problem generate one configuration in each iteration
- The method works successfully for the size of the problem under considerations
- For some problem (with no initial configurations/solution) CG solution time is grater than 10 hr
  - Can we support/start CG with good starting solution?
  - In current setting configuration generation is mainly task of optimization model (the sub problem)

Hueristics for configuration generation: initial set of configurations for CG

- Can we build configurations by selecting some attributes in a guided way?
- Some attribute can exhibit multinomial choice (e.g. Engine, Steering, Country, Exterior colour etc.)
- Configuration is build as a guided search procedure which select some attributes as per customer Information and others are completed by rules
- Partial configuration is checked for satisfiability
- Procedure selects feasible configurations with some probability



First evaluation results

- Generate 3000 configuration using 1000, attributes
- For 130 attributes, target selection rate is known
- Column generation is able match demand rate precisely

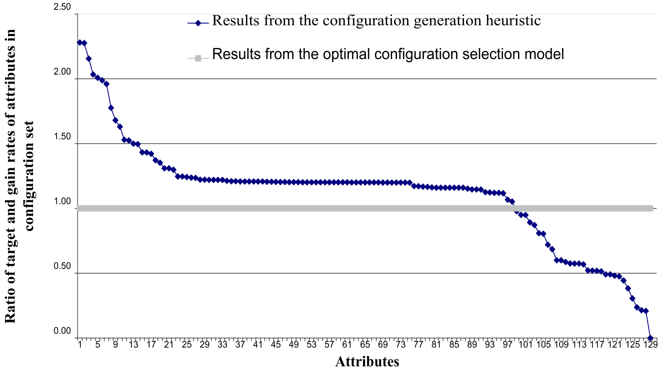


Figure: Attribute frequency match between target demand and gain rate in generated configuration set

Outline

1. Motivation
2. The planning Problem
3. Input data and its characteristics
4. Solution Approach
5. Conclusion & Future work

Conclusion & Future work

- An automated framework is discussed to generate predictive configurations for production planning
- Current model is able to consider different planning data and use them during configuration generation
- Improvement in configuration generation heuristics to speed up column generation model with better starting solution
- Further enhancement of heuristics using STA based methods. Some more product specific structure can also be exploited while generation of configuration (e.g. Tabu-list)
- Current model will be enhanced by including part and assembly level restrictions
- Benchmarking the generated results with real customer demands (e.g. component level similarity)