# Variables Clustering Method to integrate SSCP and DDMRP

**Emilio Bertolotti**

Fast.square, Milan – Italy, ebertolotti@fastsquare.it

## Abstract

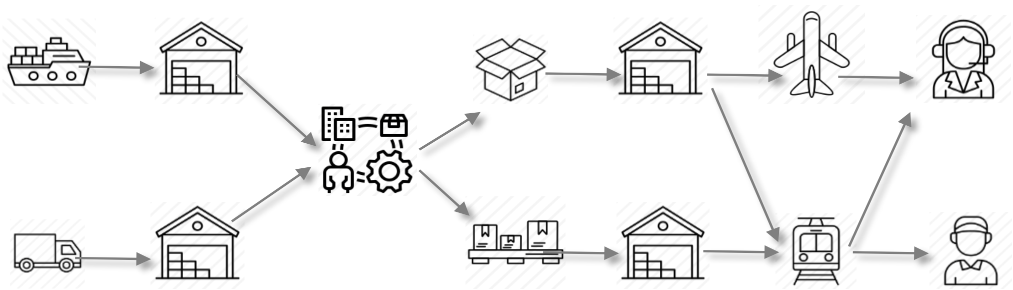
Planning of large-scale Supply Chains remains a genuine challenge because of the huge number of decisions variables to be handled. These planning problems can be formally described as Multi-Objective Combinatorial Optimization (MOCO) models. Two major techniques known as Synchronized Supply Chain Planning (SSCP) based on Mixed Integer Programming (MIP) and Demand Driven Material Requirements Planning (DDMRP) based on ad-hoc heuristics have been developed to tackle extensive supply chain planning problems. In this paper, we describe and analyze a new supply chain planning paradigm, the "Variables Clustering Method" (VCM), that integrates SSCP and DDMRP rather than treating them as alternate independent planning methods. The application of VCM allows to effectively solve larger supply chain planning instances that could be hardly solvable with more traditional approaches.

**Keywords:** SCM, MIP, MOCO, SCP (SSCP), DDMRP

## 1. Introduction

The simple example of Fig. 1 summarizes all the stages, activities, flows that must be planned to ensure end-to-end synchronization of a typical supply chain (*SC*). This synchronization requires the coordinated planning of several operational decisions [VanDerVost 2004] such as Forecast of Multi-channel future Demands (forecasted demand), Management of Customer Demand (customer orders), Plan the Outbound logistic, Planning the Multi-stages Manufacturing facilities, Plan the Purchasing and Inbound logistic of Components and Raw Material

*Fig 1. Supply Chain Synchronization*

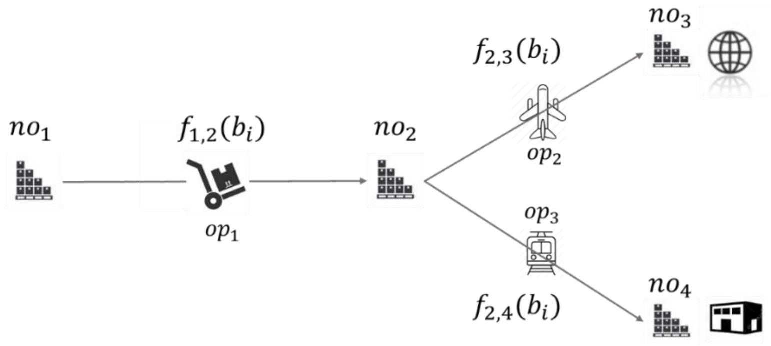


For instance, to satisfy a customer order *"X"* for a finished good *"a"* the following actions must be planned across the entire *SC*: Plan a delivery operation for *"a"*, Select the most appropriate carrier to deliver *"a"* for *"X"*, Plan the manufacturing operations to produce *"a"*, Plan the in-bound arrivals of required components, Plan the purchasing of components observing supplier terms & conditions. All these planning actions are interconnected and must satisfy the physical constraints of the *SC* such as lead times and resources capacities. Moreover, each planning action has normally several possible options that must be evaluated considering all the different conflicting objectives such as "minimize stock and work in progress", "maximize production efficiency", "minimize logistic costs", "maximize demand satisfaction". Within the Supply Chain Management (SCM) theory and practice [Assey & al 20011], several methods and advanced software solutions have been developed to support human SC planners in such a complex task. In this paper, we will focus on two of these most popular methodologies. The first one is known as "SSCP" which stands for Synchronized Supply Chain Planning and is based on a global optimization approach. SSCP is considering all the different interconnected planning decisions altogether to ensure a global synchronization of the SC plan [Assey & al 20011]. The second methodology is called DDMRP which stands for Demand Driven Material Requirements Planning. DDMRP follows a heuristic approach that computes for each location, one by one, the most convenient inventory level ensuring ideal coverage of local demand [Ptak & al 2017]. SSCP and DDMRP descriptions will be summarized in the next chapter (chapter 2) before introducing VCM.

## 2. Introducing Synchronized Planning and DDMRP methods

With the SSCP approach, the physical supply chain is represented as a network of interconnected nodes (*SC* locations) and arcs (material flows) whose flows must be optimized according to predefined optimization objectives [Ahuja & al 1993]. In this way, the planning problem can be formalized as a global Multi-objective Combinatorial Optimization Problem (MOCO) [Ehrgott 2016]. Fig.2 illustrates a simple example of an *SC* network with one "finished-good" $FG_a$ and a small chunk of its entire *SC* (the distribution part, we are ignoring all the manufacturing and supply part).

*Fig 2. A simple network model for one Finished-good, distribution only*



The whole supply chain planning horizon is broken down into a set of disjoints and discrete-time intervals called "time – buckets". For any given “time bucket” $b_i$, a flow $f_{1,2}(b_i) \geq 0$ of the finished-good is generated from node $no_1$ (manufacturing output) through the transfer operation $op_1$. This flow, $f_{1,2}(b_i)$, is directed to node $no_2$ (warehouse location) where the finished good can be stocked: $in_{2,i}$ $represents\ the\ stock\ quantity\ of\ FG_A\ at\ no_2\ in\ bucket\ b_i$

In other words, nodes like node $no_2$ can store material across different time buckets. The stock $in_{2,i}$ can be consumed by the downstream flows $f_{2,3}(b_i)$, $f_{2,4}(b_i)$. The flow $f_{2,3}(b_i)$ transfers the finished good from node $no_2$ (warehouse location) to node $no_3$ (customer site) through the transportation operation $op_2$ using carrier $C_A$ (Cargo Airplane). The flow $f_{2,4}(b_i)$ transfers the finished good from node $no_2$ (warehouse location) to node $no_4$ (customer site) through the transportation operation $op_3$ using carrier $C_T$ (Freight Train). Both carriers $C_A$ and $C_T$ are supply chain “resources” (transfer vehicles) with a limited capacity defined for each bucket $b_i$. These resources capacities are also consumed by flows $f_{2,3}(b_i)$ and $f_{2,4}(b_i)$ and their limits represent hard upper bounds for these flows. Taking a single planning decision corresponds to the computation of the “discrete flow quantities” for a set of flows $f_{i,j}(\ b_i)$ [Ahuja & al 1993]. Despite we are considering only finished-goods flows in the example, with a more extended SC, other types of flows would be modeled: semi-finished goods, components, resources flow such as machines-hours, people-hours, vehicles-space, locations-space. The computations of each flow quantity $f_{i,j}(\ b_k)$ are subject to a set of hard constraints (physical SC constraints) and soft constraints (optimization objectives) [Ehrgott 2016]. In the simple example of Fig. 2:

The set of “Hard Constraints” are:

- Material Flow conservation at each node. For instance, for the node $no_2$:

$f_{1,2}(b_1) + \ in_{2,i-1} = f_{2,3}(b_i) + \ f_{2,4}\ (b_i) + in_{2,i}$

- Capacity Constraints. For instance, warehouse capacity and carrier’s capacity:

$in_{2,i} \leq Capacity\ (n_2, b_i),\ \ f_{2,3}\ (b_i)\ \leq Capacity\ (C_A, b_i), f_{2,4}\ (b_i) \leq Capacity\ (C_T, b_i)$

- Integrity Constraints (most of the flow quantities must be integers: lot size)

$f_{h,k}\,(b_i)\ \ \in\ \ Z$

A set of “Soft Constraints” (Optimization Objectives) could be for instance:

- Minimize Stock: $Min\ (\sum_{bi} in_{2,i}\ )$
- Minimize the number of Late Demands: $Min\ (\sum_{bi} LateD\ (no_3, bi) +\ LateD\ (no_4, bi))$
- Minimize Logistic Costs: $Min\ (\sum_{Cj} \sum_{bi} CarrierCost_{j,bi})$

A complete formalization of all possible SSCP constraints is out of scope for this paper and can be found in [Bertolotti 2021]. Anyway, it is worth mentioning that in a more realistic industrial model, there would be additional hard and soft constraints to be considered.

- Additional SC hard constraints (Bill of Material [Sato & al 2011] that should be formalized: Production and distribution steps (operations $\{op_i\}$), Draw Quantity (how many

components Y are required by product X), Lead times, Multiple / Alternate sources, Lot size constraints
- Additional soft constraints often considered: Work-in-progress Minimization, Maximization of resource utilization (in short term), Overall Profit Optimization

Even for the basic SC example introduced in Fig.2, it's worth introducing at this point what a more formal description of its corresponding SSCP - MOCO model would be. This formalization is useful since we will make extensively use of one of its important definitions within the rest of the paper: "the set of the decision variables $\boldsymbol{X} \coloneqq \{\boldsymbol{x_{i,j,k}}\}^{(\boldsymbol{n\,x\,n\,x\,h})}$"

- SSCP-MOCO variables and parameters:

$B \coloneqq \{b_1, b_2, \dots b_h\}$ $\quad$ $set\ of\ disjoint\ time\ intervals\ (buckets), b_k \in R$

$N \coloneqq \{no_{i,k}\}^{n\,x\,h}$ $\quad$ $set\ of\ nodes, where\ no_{i,k}\ node\ i\ is\ in\ bucket\ k\ (buffers)$

$NM \coloneqq \{nm_{i,j,k}\}^{(n\,x\,n\,x\,h)}$ $\quad$ $nm_{i,j,k} = 1\ if\ n_i\ is\ connected\ to\ n_j\ in\ bucket\ b_k, is\ 0\ otherwise$

$D \coloneqq \{d_{j,k}\}^{n\,x\,h}$ $\quad$ $where\ \ d_{j,k} \coloneqq indep.demand\ for\ n_j\ in\ bucket\ k,\ d_{j,k} \in \mathrm{Z}$

$\boldsymbol{X} \coloneqq \{\boldsymbol{x_{i,j,k}}\}^{(\boldsymbol{n\,x\,n\,x\,h})}$ $\quad$ $\boldsymbol{where\ \ x_{i,j,k} \coloneqq flow\ from\ n_i\ to\ n_j\ in\ bucket\ k,\ x_{i,j,k} \in Z}$

$S \coloneqq \{in_{i,k}\}^{n\,x\,k}$ $\quad$ $where\ in_{i,k} \coloneqq stock\ at\ buffer\ n_i\ in\ bucket\ k,\ \ I_{i,k} \in \mathrm{Z}$

$OP \coloneqq \{op_t\}^p$ $\quad$ $where\ op_t\ represent\ a\ BOM\ defined\ SC\ "operation"$

$g(\bar{x}) \coloneqq \{g_1, g_2, \dots, g_q\}$ $\quad$ $g(\bar{x}): R^n \longrightarrow R^q\ defining\ feasible\ solutions\ \bar{x}\ \in U \subset X$

$ob(\bar{x}) := \{ob_1, ob_2, \dots ob_p\}$ $\quad$ $with\ ob(\bar{x}): R^n \longrightarrow R^p\ \ and\ \ \bar{x}\ \in U \subset X$

- Solving the SSCP-MOCO model means to find:

$$Min\ \{ob(\bar{x}), with\ \bar{x}\ subject\ to\ g(\bar{x}) \leq 0\ and\ \bar{x} \in\ Z^n\} \qquad (1)$$

Solving the SSCP-MOCO model requires assigning a value to all the variables in the set $X \coloneqq \{x_{i,j,k}\}^{(n\,x\,n\,x\,h)}$ while observing all model constraints and objectives. Each variable $x_{i,j,k}$ in the set $X$ represents the quantity that the flow from node $no_i$ to node $no_j$ should have in bucket $b_k$. The cardinality of the set $X \coloneqq \{x_{i,j,k}\}^{(n\,x\,n\,x\,h)}$ is the main driver for the overall solution complexity and problem-solving efficiency. The reduction of this number improves problem-solving efficiency and turns out to be the main objective of the VCM methodology: reduction of $\#x_{i,j,k}$ through variables clustering. SSCP-MOCO models have usually been solved through "MIP-scalarization" techniques [Ehergott 2006]. MIP stands for Mixed Integer Programming [Nemhauser & al 1988] while "MIP-scalarization" is an extension of the classic MIP method used to handle multiple objectives. Rather than solving concurrently, in one shot, the entire set of optimization objectives, MIP-scalarization solves each single objective one at a time in a sequence of prioritized runs. In other words, MIP-scalarization transforms the optimization of the entire "array of objectives" into a sequence of "scalar" optimization steps where each step preserves the quality of the results obtained in previous steps [Ehergott 2006].

MIP-scalarization is known to be time-consuming since it consists of multiple runs on a big number of decision variables. In a typical supply chain planning problem, the number of decision variables depends on $\#Products$, $\#Locations$, $\#Buckets$, $\#Resources$ and the number of optimization runs depends on $\#Objectives$ so it is not unusual to come up with:

$$f(\#Poducts\ ,\#Locations,\#Buckets,\#Resources)\cdot\#Objectives \approx 10^6 \div 10^9 \quad (2)$$

Often, it becomes quite difficult to run an "end-to-end" automated SSCP process observing the response time normally required in real business contexts where the SSCP planning process should run in a few hours since SCM analysts must run several simulations daily [VanDerVost 2004]. Consequently, even if "MIP-scalarization" is the best way to produce "almost exact results" [Ehergott 2006], sometimes is not considered a practical approach, and a more genuine heuristic method is preferred [14]. One of these heuristic approaches is the so-called DDMRP (Demand Driven Material Requirements Planning) that works quite well in supply chains where the computation of each supply flow can be driven by a "local demand" signal rather than depending on global supply chain synchronization [Ptak & al 2017].

The DDMRP approach, transform an SC planning problem into the problem of computing for each location of the entire supply chain the most convenient inventory levels that will likely satisfy the overall SC demand. Contrary to the SSCP approach that generates stock quantities based on a global "end-to-end" supply chain synchronization, the DDMRP is decoupling all these decisions computing the right stock quantity for each location (node) separately. This methodology consists of five steps:

1. *Strategic Inventory Positioning*

   Identify the set of SC nodes $S \coloneqq \{no_i\}$ entitled to keep inventory. Consequently, derive all the nodes that should not carry inventory. The selection of "strategic nodes" entitled to keep inventory is positively influenced by:
   - Variability of local demand (at node $no_i$), local supply (at node $no_i$)
   - Criticality of demand involved (end customer tolerance time)
   - Reduction of "deliver to order" lead time (Product, Customer, Market goals)

   Each node $no_i \in S$ becomes a candidate's locations to carry stock.

2. *Definition of buffers profile and stock levels for each* $no_i$

   For each node $no_i \in S$, a "buffer profile" that captures product-specific characteristics like "local supply lead time" and "local lead time variability" is defined.
   For each node $no_i \in S$, three different levels of stock are computed:
   - RED level: minimum stock level (*Statistical Safety Stock).*
   - YELLOW level: ensures demand coverage *(AverageDailyDemand* $\cdot$ *Leadtime*).
   - GREEN level: max stock (*Minimum Order Quantity, Replenishment Intervals).*

3. *Dynamic Adjustment of Buffer Levels for each* $no_i$

   The buffer profiles and buffer levels are not static values, they can be dynamically arranged according to the variability of endogenous or exogenous SC parameters. Examples of endogenous parameters affecting buffer profiles and buffer levels are seasonality of supplier lead time, planned shut-off of a logistic carrier. Examples of exogenous parameters affecting buffer profiles and buffer levels are weather conditions, expected demand reshaping due to changes in market strategy.

4. *Demand-Driven Generation of replenishment orders*

DDMRP has its own "reorder point" logic [Ptak & al 2017]: the "reorder point" is the upper bound of the YELLOW level while the "actual" stock position for a node $no_i$ is computed as $ActualStock\ (no_i) = OnHand\ (no_i) + Incoming\ (no_i) - QualifiedDemand\ (no_i)$

5. *Execution*

The execution step includes dynamic prioritization of open replenishment orders.

Within the supply chain management theory and practice the two SSCP and DDMRP methodologies just described are often seen as alternative approaches to implement a supply chain planning procedure. On the contrary, the Variable Clustering Method (VCM) introduced in the next chapter (chapter 3.) integrates these two approaches enabling the practical resolution of larger supply chain planning instances that could not be solved with a more traditional approach.

## 3. The Variables Clustering Method (VCM)

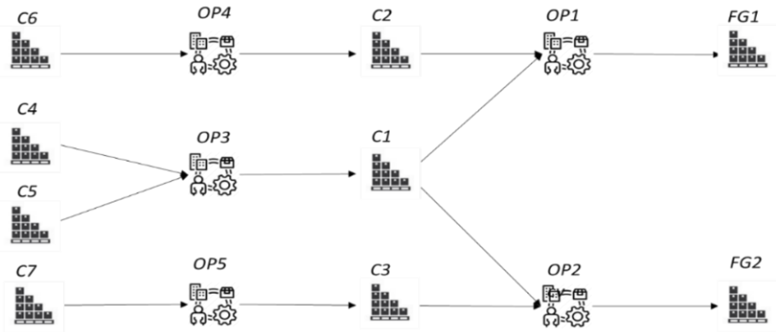
In the definition of VCM, the word "variables" refers to the decision variables that belong to the set $\boldsymbol{X} \coloneqq \{\boldsymbol{x_{i,j,k}}\}^{(\boldsymbol{n\,x\,n\,x\,h})}$. The idea of clustering decision variables within the supply chain planning domain is not a new one. This idea has been successfully applied, for instance, within the automotive industry to solve industry-specific planning problems and also tested for CPG (Consumer Packaged Goods) adopting ad hoc approaches [Bertolotti 2021]. In this paper, a more universal Variables Clustering Method valid for almost all types of industries is proposed. This new VCM approach exploits a combination of the SSCP and the DDMRP paradigms to make it generally applicable.

The simple case illustrated in Fig.3 is a good starting point to explain how VCM is leveraging the synergy between SSCP and DDMRP to reduce the cardinality of $X \coloneqq \{x_{i,j,k}\}^{(n\,x\,n\,x\,h)}$. In Fig.3, the different bottles represent different finished goods that belong to the same product family and share common basic sub-components such as the same "oxy cream" and the same plastic bottle. The proliferation of different finished goods within the same product family is a common trend in many industries that must satisfy markets and consumer's specific needs through products diversification. This "assortment" expansion is even more amplified considering all the different target markets that often require country-specific labels, local packaging formats, particular brands [VanDerVost 2004].

*Fig 3. Example of Finished-goods proliferation within the same products family*


Fig4 represents a simple SC network model capturing this recurrent pattern with two similar finished goods ($FG_1$, $FG_2$) belonging to the same class ($Cl_a$). The two different finished goods ($FG_1$, $FG_2$) are packaged through the same "production line" (same packaging line) and use several common components (could be, for instance, same liquids or same basic food ingredients). In this case, what is specific to each finished good is just the packaging material represented by the components $C_2$ and $C_3$ purchased through operations $OP_4$, $OP_5$. For instance, $FG_1$ and $FG_2$ could be the same core product manufactured for two different countries which require country-specific labeled boxes.

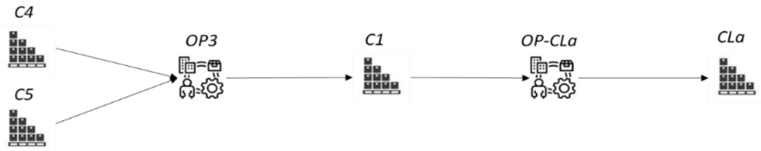
*Fig 4. Finished-goods Commonalities and Peculiarities*




A closer analysis of this example reveals how the entire supply chain model could be split into two different sub-models. The first sub-model representing a 'common' part of the supply chain which is shared between $FG_1$ and $FG_2$ (Fig. 5 Clustered sub-model). This clustered sub-model is representing the aggregated SC for the class $Cl_a$ rather than the two SC of the single finished goods $FG_1$ and $FG_2$. In a real industrial case, this operation could aggregate tens or even hundreds of products.

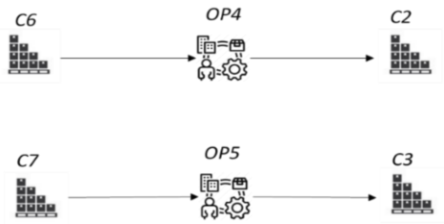
Fig. 5 Clustered sub-model

.




The second sub-model representing the remaining product-specific parts (Fig. 6 Decoupled sub-model).

*Fig. 6 Decoupled sub-model*




The interesting aspect of this model split consists in the possibility of systematically identifying "candidate branches" of the original SC model whose decoupling could effectively generate "clustering opportunities" that reduce the overall number of flows and nodes. More precisely, continuing with the example of Fig.4, the model split logic is looking for "candidate branches" of the model that satisfy the following two conditions:

1. "Candidate branches" that once decoupled can be "planned separately".
   The two branches of Fig.6 ($C_2, C_3$) are there to generate the time-dependent supply/stock quantities for components $C_2$ and $C_3$. To validate the decoupling of these two branches we want to verify that the generation of the supply plan for components $C_2$ and $C_3$ can be done through a simpler stock replenishment logic based on the forecasted demand of $C_2$ or $C_3$ and does not need to be based on the "synchronized" dependent demand generated by $OP_1$ and $OP_2$. This validation can be done by looking at a few predefined planning parameters of these sub-components $C_2$ and $C_3$ (BOM [Sato & al 2011]). These parameters include things like sub-component costs/value and demand/supply historical variability (For instance, a low cost/value of the component enables extra stock carrying to avoid synchronized planning).

2. "Candidate branches" whose decoupling can generate clustering opportunities.
   The decoupling of the two branches of Fig. 6 creates the opportunity to merge the two SC models for $FG_1$ and $FG_2$ into one single model for the class $Cl_a$. The clustered model shown in Fig. 5 represents the new aggregated supply chain model defined for the class "$Cl_a$". The two operations $OP_1$ and $OP_2$ have also been merged into a single operation "$OP - Cla$". The solution of the clustered model requires "synchronized planning" which means we must apply a global optimization approach such as SSCP. Nevertheless, thanks to the aggregation, the size of the model is smaller compared to the original SC model.

Since the computational complexity of a “replenishment” method (heuristics) is inferior to the computational complexity of a synchronized planning procedure (optimization), the complexity of the entire planning solution is mostly depending on the MIP-based method used to solve the clustered sub-model. The “degree of clusterization” is thus heavily affecting the actual reduction of the SC complexity. Even with a simple model like the one in Fig. 4, the VCM approach was able to significantly reduced the number of flows with a consequent reduction of the decision variables that must be handled by the MIP-based optimization algorithm. This complexity reduction can be synthesized in table Tab.1

*Table 1. Reduction of SC decision variables through VCM*

| Number of Decision Variables | Original Model | VCM Model |
|---|---|---|
| $\#x_{i,j}$ | 13 | 5 |

If we consider the total number of time-buckets "$\#b_k$" this reduction would be even more significant: $\#x_{i,j,k} = (\#x_{i,j} \cdot \#b_k)$ in terms of absolute number of decision variables.

To describe more formally the “variables clustering method” introduced with the simple example of Fig. 4, let’s define the following VCM universal procedure that can be applied to any SC model:

$$VCM(SC): \ SC \longrightarrow CSC + DSC \qquad (3)$$

$VCM(SC)$ converts a generic Supply Chain model "$SC$" into two different sub-models: a clustered sub-model "$CSC$" and a decoupled sub-model “DSC”. Given an input model “SC”, the $VCM(SC)$ transformation consists in checking all the “SC” branches to identify the “clustering” and the ”decoupling” opportunities. The procedure is run in a “net change mode” which means that is processing only the BOMs changed since the previous run. The VCM procedure involves three steps (VCM-1, VCM-2, VCM-3):

- *VCM-1: Identify Clustering Opportunities*

This step examines all flows $x^c_{i,j}$ for any given BOM to identify real clustering opportunities. The $x^c_{i,j}$ branches are the $x_{i,j}$ branches marked in the BOM as "clustering candidates" from the product engineers when they must define a new Product BOM. A branch $x^c_{i,j}$ is selected for decoupling if and only if the following two conditions are both satisfied:

*Condition1: The branch $x^c_{i,j}$ can be decoupled from the “planning point of view”*

Check if the generation of the supply plan for the “candidate branch” could be done through a simpler stock replenishment logic based on the forecasted demand rather than strictly requiring the “synchronized” dependent demand generated by downstream SC layers. The checking rules are based on two sets of parameters:

- ✓ Product/location-specific parameters, like associated economic value, market priority, are stored at “any level” of the BOM, SKU data structures [Sato & al 2011]. These parameters are defined by product engineers during the “New Product

Introduction" and massively adapted to business changes happening during the entire product life cycle (product substitutions, new market conditions, …).

- ✓ Statistical parameters extracted from demand/supply historical data to assess the "predictability" of the "local demand/supply" (for instance: demand variance, historical forecast errors, etc.).

As an example of checking rule: LOWCCNC rule: a component with "low carrying cost" and "low demand variance" is a good candidate for carrying stock provided is not "highly critical" and requires synchronized planning (low and high threshold values are configurable by product segments).

*Condition2: The decoupling of the branch $x^c_{i,j}$ creates a clustering opportunity*

The new BOM obtained from the removal of a branch:

- ✓ Can be assigned as a new member of one of the previously generated clustered BOMs (which represent the real clustering).
- ✓ Generates a new clustered BOM (to be hopefully populated with additional BOM members)

All validated "decoupled) and "clustered" BOMs are fed into the second step VCM-2.

- *VCM-2: Construct the two final sets of validated Clustered-BOMs and Decoupled-BOMs*

In this step, all the individual "decoupled-BOMs" and "clustered-BOMs" defined in the previous step are transformed into

- ✓ the final set of "Decoupled-BOMs" that will be used by DDMRP planning
- ✓ the final set of "Clustered-BOMs" that will be used by SSCP planning.

At this point the two output sets $CSC\ and\ DSC$ are completely populated and can be used to run the planning procedures: $CSC := [Clustered - BOMS]$, $DSC := [Decoupled - BOMS]$ The BOMs that could not even be considered for clustering also fall into the CSC set.

- *VCM-3: Run the end-to-end SC solver*

The two solvers (SSCP, DDMRP) are run concurrently. The optimization-based SSCP solver will solve the $CSC := [Clustered - BOMS]$ planning model and the heuristic-based DDMRP solver will solve the $DSC := [Decoupled - BOMS]$ planning model. There are no reasons to run these two steps in sequence, they can be run in parallel. The critical path is on the SSCP solver (optimization algorithm).

- ✓ The run of SSCP on the CSC model will generate a plan for the "classes" $Cl_{i,}$ and not for the finished goods $FG_i$. To find the final plan for finished goods, a post-process populates $FG_i$ quantities breaking down the aggregated values obtained for $CL_i$ (tie-breaking rules are used in case of scarce supply). This breakdown does not compromise the quality of the plan done for the classes since all the constraints that should be observed for the finished goods are exactly the ones that have been observed for the classes. Moreover, this breakdown process does not affect significantly the overall complexity of the end-to-end solution since it is a computation whose complexity is linear in the number of classes.
- ✓ The DDMRP run will generate replenishment orders for the decoupled $x_{i,j,k}$

A final routine will merge the planned orders generated by SSCP (synchronized plan) and the planned orders generated by DDMRP (replenishment plan) to construct the end-to-end output plan that covers the whole original SC model.

## 4. VCM results

All the test cases done so far show that VCM can systematically reduce the complexity of the SC models [Bertolotti 2021]. Before articulating these positive outcomes, it is worth mentioning a few exceptions. These corner cases have been encountered with supply chains that reveal an unusual high proliferation and specialization of product types. Examples of these special cases can be found for instance in the “fashion industry” where even products belonging to the same class/category tend to have very disparate characteristics. For instance, in “Fashion Glasses Manufacturing” it becomes quite challenging to aggregate a reasonable number of products with similar characteristics into a clustered class as evoked by Fig. 7.

*Fig 7 High Product Diversity in Fashion Industries*


Still, for these corner cases, it is always possible to measure and preliminary estimate upfront the effectiveness of VCM. For any SC instance, the following metrics, based on the “clustering factor” (CF), provides an immediate indication of the VCM clustering effectiveness. The clustering factor CF for a given model instance SC can be measured by running the VCM procedure itself. CF(SC) is defined as:

$$CF(SC) = (1 - \frac{\#x_{i,j}\,(CSC)}{\#x_{i,j}\,(SC)}) \qquad (4)$$

*CF* measures the reduction of the decision variables required to solve SSCP for the CSC model. In a few experiments conducted with a set of CPG test models (Consumer Packaged Goods industries like foods and beverages), the CSC model had a significantly lower number of decision variables compared to the corresponding original *SC* model. The test has been conducted on three models of different sizes: SC-CPG1, SC-CPG2, SC-CPG3. The resulting *CF* was in the following range:

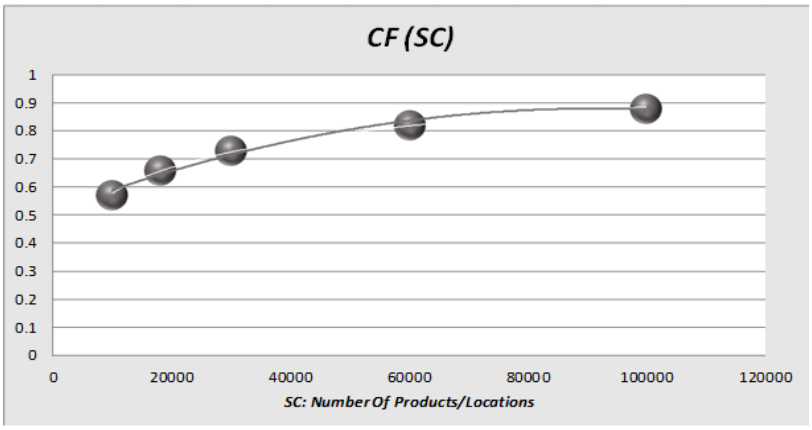
$$CF\ (CPGx) \approx 0.6 \div 0.9 \quad (5)$$

Furthermore, we have seen, for these CPG SC test cases, how $CF(SC)$ tends to improve with the growth of the *SC* model size. This is because the number of clustering opportunities tends to grow with the increasing number of "products/locations" modeled, as illustrated in Tab. 2 and in Fig. 8.

*Table 2 Clustering Factor vs. SC Model Size*

| SC-CPG Test Case | SC-CPG1 | SC-CPG2 | SC-CPG3 |
|---|---|---|---|
| Number of Products/Locations | 10000 | 30000 | 100000 |
| CF(SC) | 0.57 | 0.73 | 0.9 |

*Fig. 8 CF growing trend*




Moreover, the usage of a heuristic approach for the DSC sub-model is not affecting sensibly the overall quality of the plan. This is due to the intelligent detection/evaluation of the candidate branches that could be decoupled and planned through a heuristic approach without affecting too much the overall quality of the SC plan.

Rather than reporting actual planning times, that are subject to the HW platform utilized as well as depending on some specific implementation facet of the pre/post-processing, we provide the following more eloquent relationship (Eq. 8) that gives a good rough indication on the planning time gained when VCM is applied to relatively large *SC* models.

Given:

- $T\ (SC) =\ time\ to\ plan\ a\ given\ SC\ model\ through\ SSCP\ optimization$
- $T\ (CSC) =\ time\ to\ plan\ the\ CSC\ model\ through\ SSCP\ optimization$
- $k1(SC) = time\ taken\ by\ SSCP\ (SC)\ pre/post - processing$
- $k2\ (CSC) = time\ taken\ by\ SSCP\ (CSC)\ pre/post - processing$

Despite an average limited reduction of the $SCS$ model size (70%):

$$\#x_{i,j}(SCS) = \#x_{i,j}\ (SC)\ /\ CF\ (SC)\ =\ \#x_{i,j}\ \ /\ 0.7 \qquad (6)$$

The reduction in terms of planning time becomes quite significant:

$$k2 + T(CSC)\ \ll\ k1 + T(SC) \qquad (7)$$

This result can be explained considering that the MIP is an NP-hard problem [Nemhauser & al 1988], consequently, a decrease by "$\Delta\#x_{i,j}$" in the size of the model corresponds to a decrease in resolution time that is "faster than any polynomial function of $\Delta\#x_{i,j}$"

$$\frac{k2+T(CSC)}{k1+T(SC)} \ll \frac{\#x_{i,j}(CSC)}{\#x_{i,j}\ (SC)} \qquad (8)$$

These outcomes reveal why VCM enables the planning of larger supply chains with a large reduction of the planning time despite a less significant reduction of the CSC number of decision variables. Lastly, we just want to point out that these conclusions are quite conservative since we should also consider that the size of the clustered sub-model ($\#x_{i,j}(CSC)$) is growing less than linearly with the growth of the original model size ($\#x_{i,j}(SC)$) as shown in Fig. 8.

## 5. Conclusions and Implications

VCM is an innovative supply chain planning method that leverages the integration between two consolidated approaches: the MIP-based SSCP and the Heuristics-based DDMRP. Thanks to a smart application of the variables clustering principle VCM can reduce the number of decision variables to be handled within any supply chain planning automated process. This reduction in model size ensures a substantial reduction of the overall planning time. The VCM algorithm is based on an intelligent method that identifies clustering and decoupling opportunities preserving the overall quality of the solution while reducing model complexity. All these aspects enable VCM to practically solve larger supply chain planning instances that would otherwise require unacceptable and unrealistic overall planning time. SSCP and DDMRP have been often seen as alternative methods to approach planning within the Supply Chain Management context. VCM is leveraging the best of the two methods to achieve overall better planning results.

## References

Van der Vorst J.G.A.J. (2004). *Supply Chain Management: theory and practices*, In Theo Camps, Paul Diederen, Gert Jan Hofstede, Bart Vos (eds), The Emerging World of Chains & Networks, Elsevier, Hoofdstuk.

Assey M., Janvier J., (2011), *A New Introduction to Supply Chains and Supply Chain Management: Definitions and Theories Perspective*, Glorious Sun School of Business and Management, Donghua University Shanghai 200051, China.

Ahuja R., Magnanti T., Orlin J., (1993), *Network Flows Theory, Algorithms, and Applications*. Pearson Education, USA.

Ehrgott M, (2016), *Exact Methods for Multi-objective Combinatorial Optimisation*, In Book: Multiple Criteria Decision Analysis pp 817-850, Springer.

Sato R., Matsui Y., Furukado M., (2011), *Concept of Bills of Material for Supply Chain Planning and Its Prototyping with ERP*. Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia.

Ehrgott M, (2006), *A Discussion of Scalarization Techniques for Multi-objective Integer Programming*. Annals of Operations Research 147: 343–360.

Nemhauser G., Wolsey L., (1988), *Integer and Combinatorial Optimization*. John Wiley & Sons.

Ptak C., Smith C., (2017), *Demand Driven Material Requirements Planning*, Industrial Press, DDMRP Institute, USA.

Bertolotti E, (2021), *Variables Clustering Method to enable Planning of Large Supply Chains*. In the book Multi-objective combinatorial optimization: problems and solutions. Elsevier.