

Date: 09/2008

Version: 1.0

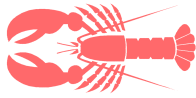
Id:

ROSTUDEL Professional Services

*PROCUREMENT WITH INVENTORY,  
PURCHASE AND TRANSPORTATION  
COSTS*

\* \* \*

GESTION D'APPROVISIONNEMENT  
AVEC COUT DE STOCKAGE,  
D'ACHAT ET DE TRANSPORT



## 1. *CONTEXT/CONTEXTE*

---

*A major hard-discount European leader wants to develop a decision-making tool allowing to decide when and what quantity to purchase for a set of products and from the same supplier.*

*The deliverable consists in:*

- *Validating with ILOG ODMS mathematic models to address the business requirements. ODMS is a bundle of OPL modelling platform, on top of Cplex or CP-optimizer engines, while ODM provides a scenario-analysis platform.*
- *Integrating these models in JAVA, with fast answers when calling the engine.*
- *The time decomposition of the horizon allows with no model difference to handle items with a fast rotation (in days), medium one (weeks), or slow one (months).*
- *The number of items varies from 100 to 600 according to different scenarios. Time scope varies from 6 to 18 periods.*

Un acteur majeur du hard-discount européen souhaite intégrer à son système logistique un outil d'aide à la décision sur les quantités et les dates d'achat d'un ensemble d'articles fournis par un même fournisseur.

L'étude demandée consiste à valider via ILOG ODMS des modèles mathématiques couvrant les 3 types principaux de fournisseurs et fournir une API JAVA s'intégrant au système logistique, le module de calcul propre à un fournisseur s'exécutant en quelques secondes.

ODMS est constitué du modelleur OPL qui appelle les moteurs d'optimisation Cplex ou CP-Optimizer, tandis qu'ODM propose un gestionnaire de scénarios.

La granularité temporelle permet indifféremment d'adresser ce problème pour des classes d'article à rotation rapide (jour), moyenne (semaine), ou lente (mois). En ce qui concerne la volumétrie, on envisage dans le cas nominal une centaine de produits et 6 périodes dans l'horizon, la volumétrie maximale atteignant 500 produits et 18 périodes



---

## 2. METHODOLOGY / METHODOLOGIE

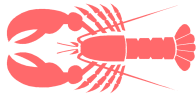
---

*ROSTUDEL methodology consists in quantifying the customer's requirements in an analytic way. We illustrate this here by a use-case "timetabling in retail world"*

- *What are the data of the business case? The set of shop's employees with their characteristics (skills, holidays already taken...), the different activities (cashier, grocery...), the daily workload (how many employees are required on a given activity at every hour), teams definition, the list of business rules (what is a day pattern, what is the maximum hours to be worked in one day for each employee...)?*
- *What are the decision variables? For instance, one would be interested to decide for each employee and for each day if the employee should work or should take a day off. At a lower granularity, on which activity at every hour of the working day?*
- *What are the drivers and objectives? In a given scenario, one could be interested to minimize the labour cost while covering workload. On another one, one would prefer to be as close as possible to the workload with respecting labour rules (min-max daily, skills, min-max weekly). Some other scenario would add skills to a certain set of people or simply remove some of the employees out of the scope of the schedule.*

La méthode ROSTUDEL consiste à traduire les besoins métiers de manière qualitative et quantitative par une approche analytique. Prenons l'exemple de la génération d'emplois du temps dans la distribution:

- Quelles sont les données métiers ? On s'intéressera à la liste des employés du magasin avec leurs caractéristiques (compétences, compteur de congés payés...), les activités (caisse, réassort, boucherie...), la charge horaire pour ces activités (combien de personnes requises et avec quel profil de compétence), les groupes de personnes, les règles métiers (quel « pattern » de travail pour une journée, les limites horaires journalières ou hebdomadaires...)
- Quelles sont les variables de décision ? Une première approche pourrait simplement décider chaque jour des personnes qui travaillent puis à une granularité plus fine, décider du choix de l'activité toutes les heures.
- Quels sont les scénarii et les objectifs ? On peut s'intéresser à réduire le coût salarial tout en couvrant la charge de travail, ou bien couvrir au mieux la charge tout en respectant les contraintes individuelles. On peut vouloir jouer sur les compétences des personnes ou en exclure certaines du périmètre d'un scénario. Ces scénarii conditionnent le choix de l'objectif quantitatif que l'on souhaite minimiser.



### 3. BUSINESS REQUIREMENTS AND DRIVERS / MODELE METIER ET SES VARIANTES

*By deriving the methodology presented above, we first describe the business model concerning items to be purchased, which is common to the different drivers:*

*Data for each item:*

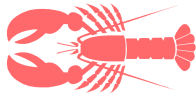
1. *Its demand over the time horizon, with known prices possibly dynamic (they may change at any period to reflect inflation or promotion).*
2. *Inventory costs and shortage costs (shortage is allowed only for some of the items)*
3. *Minimum level for each item purchase order*
4. *Items follow packaging rules: there are up to 3 conditioning units (see figure 1). For instance, milk bottles will be purchased at least as a multiple of 18 bottles, which is the "lot" unit. Upper units are the layer (72 bottles or 4 lots) and then the pallet (360 bottles or 5 layers or 20 lots). Purchasing prices are usually decreasing with the conditioning unit.*
5. *Items have lead-time : the item will be available at a given time if the order has been placed some time before (this time may differ between items)*
6. *Items once purchased may not be ordered before a given fixed period from the same supplier*
7. *Business cases vary from 100 products/6 periods to 500 products/18 periods*

**Problème 1 :** Pour chaque article, étant donné des demandes connues sur l'horizon, des prix d'achat connus et variables dans le temps (promotions), des coût de possession (stockage) connus, trouver la politique d'achat optimale minimisant les coûts d'approvisionnement (prix d'achat \* quantité commandée) + coûts de stockage (coût de possession \* quantité stockée) sous les contraintes :

1. Demande dynamique (i.e déterministe mais variable) sur l'horizon
2. Coûts de possession (stock) ou de rupture (pour certains produits)
3. Prendre en compte des unités minimales de commande (chaque commande est multiple d'une quantité minimale)
4. Prendre en compte des prix d'achat dégressifs par "packaging" : le prix unitaire si on commande par palette est plus petit que le prix si on commande par couche ou par lot (notion de seuil sur les prix d'achats, cf figure 1)
5. Inclure des délais de livraison (multiples de période)
6. Pour certains articles, inclure la couverture ou cadencement (écart minimum entre deux commandes)

Item	LOT	LAYER	PALLET
MILK	18	72	360
CORNFLAKES	6	60	300

*Table 1: conditioning units for two items*



**Decision variables:** we need to calculate for each item and each period of time the purchased quantity, or equivalently speaking, the purchased quantity of each decomposition unit (lot, layer, and pallet).

Finally, we describe the different business drivers or scenarios that we would like to model and test:

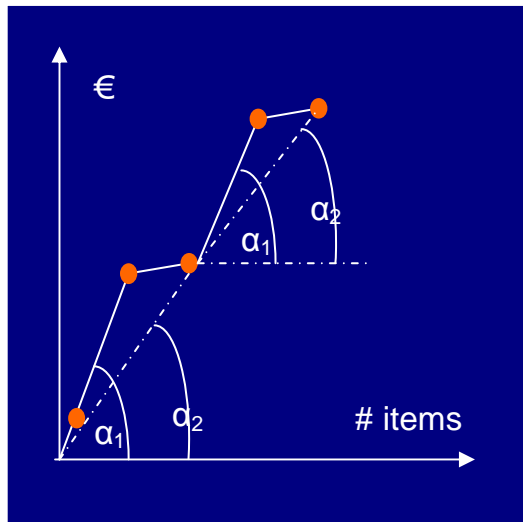


Figure 1: Cost function of the quantity of items purchased

**Driver 1:** Find an optimal procurement policy for each item in order to minimize the sum of purchasing, inventory and shortage costs. The policy must respect each item specific rule (conditioning, lead-time, lag-time). For some family of products, there is no coupling constraint between the items. Therefore, one can see this problem with  $N$  items as solving  $N$  independent sub-problems.

**Driver 2:** Find the optimal policy with previous item rules and a new transportation constraint: items must be grouped in order to fill a full truck. A truck has a given capacity in pallets.

**Driver 3:** Similar to case 2, but this time, the truck constraint is slightly different: it is allowed to fill partially the truck but there is a tax when it is not fully loaded. The tax applies to the stock value (that is the purchasing cost of all items in the truck) and the tax decreases stepwise according to the number of pallets in the truck.

Ce problème en soi n'a aucun couplage entre les articles et est rencontré tel quel pour certains fournisseurs. Les quantités suggérées par une approche manuelle ou par un système d'aide à la décision respectant les contraintes peuvent être envoyées telles que au fournisseur. Pour d'autres en revanche, les quantités commandées par période ne peuvent être agrégées telles quelles car elles ne satisferaient pas des contraintes de transport. C'est alors l'objet du problème 2 de coupler ces modèles de stock avec une contrainte dite contrainte de camion.

**Problème 2 :** "une fois connues" les quantités à acheter, remplir les camions du fournisseur. Si celui-ci n'a pas de contrainte particulière sur son remplissage, il n'y a rien à faire. Sinon, on a suivant le fournisseur un choix (exclusif et connu) entre deux remplissages :

1. "Camion complet" : la capacité du camion est un multiple de nombre de palettes (typiquement 33 palettes ou 66) et il faut agréger les commandes de telle façon qu'elles tombent pile sur un multiple de cette capacité. Il n'y a pas là de coût de transport mais il faut impérativement respecter cette contrainte dans la constitution des commandes.
2. Barèmes quantitatifs : le coût est fonction du remplissage du camion (exemple : coût unitaire 10 de 1 à 15 palettes, 9 de 16 à 32 palettes, 8 pour 33 palettes). Voir figure 2.

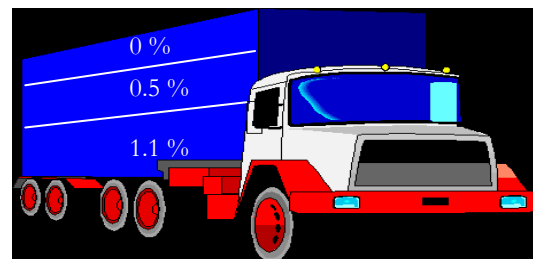
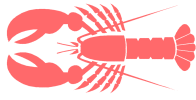


Figure 2: Transportation cost as a stepwise tax of the value inside the truck. Steps are given by number of pallets thresholds



#### 4. MODEL AND ANALYSIS

Once the business requirements have been clearly defined between ROSTUDEL and the customer comes the time of modeling and solving, one of the crucial times when addressing academic and practical combinatorial problems [2]. Thanks to years of experience in both theoretic and practical combinatorial problems, coming with a mathematic model that fully addresses business requirements is ROSTUDEL specialty.

##### Single-item network structure

The single-item model is quite similar to single-item lot sizing model described in [1]. In particular, there is an underlying network structure since at each period the input flow summing the previous inventory and the purchased item available equals the output flow summing the demand and the next inventory level:

Formally speaking, we have for each period of time:  $s_{i,t-1} + x_{i,t-\delta} = d_{i,t} + s_{i,t}$  (1), where  $s$  is for the inventory while  $x$  is the quantity of purchased item and  $d$  the corresponding demand. Note that in the formula we introduce the lead time  $\delta$  which is 1 period in figure 3.

The minimum number of periods between consecutive deliveries is handled by additional binary variables and a “big  $M$ ”:

$$x_{i,t} \leq M \cdot y_{i,t}$$

$$\forall t, \sum_{\delta=0}^{c_i-1} y_{i,t-\delta} \leq 1 \quad (2)$$

The first constraint sets the binary variable to 1 any time we purchase any quantity of the item while the second one forbids two occurrences of  $y$  in any sequence of  $c$  consecutive periods.

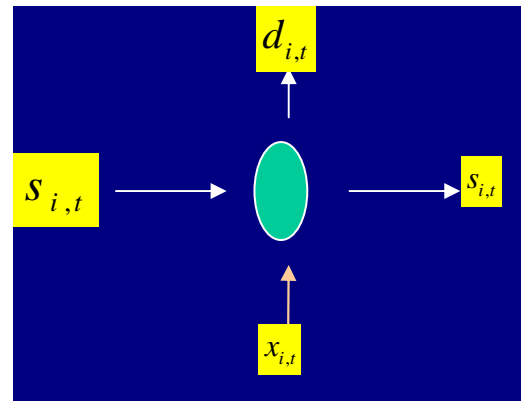


Figure 3: network structure of single-item procurement model

Une fois le modèle métier capturé, reste à l’écrire dans un formalisme mathématique. Une expertise ancienne et une connaissance pointue des plateformes de modélisation du marché permet à ROSTUDEL d’illustrer son savoir-faire dans cette phase cruciale tant dans la qualité et la concision du code associé que dans la rapidité de livraison du modèle. La plupart des spécialistes académiques et praticiens insistent sur la nécessité d’aboutir de manière itérative à un « bon » modèle avant tout développement ultérieur [2]

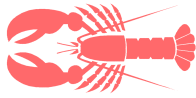
##### Structure de flot pour le problème mono-article

On retrouve la structure de production manufacturière par lot décrite dans [1] dans l’équation de « conservation du flot » à chaque (cf équation 1 en regard et figure 3):

$$s_{i,t-1} + x_{i,t-\delta} = d_{i,t} + s_{i,t} \quad (1)$$

, où  $s$  désigne le niveau de stock,  $d$  la demande et  $x$  la quantité commandée en  $t - \delta$ . Voir la figure 3.

L’écart minimal entre deux périodes consécutives de livraison se linéarise avec un indicateur booléen et un « big  $M$  » (cf équation 2 en regard). La première contrainte force à 1 le booléen dès que la quantité achetée est non nulle tandis que la seconde interdit plus d’une occurrence de l’indicateur  $y$  sur toute période glissante de  $c$  périodes.



Decomposing the purchased quantity of item into its different conditioning units is based on pure integer arithmetic division. Let's assume the item has the conditioning 6 items per lot, 4 lots per layer and 5 layers per pallets. Then introducing integer variables that count the number of each conditioning unit purchased, we link our initial decision variable to these new variables by:

$$x_{i,t} = 6.u_{i,t,lot} + 24.u_{i,t,layer} + 120.u_{i,t,pallet} \quad (3)$$
$$u_{i,t,lot} < 4, u_{i,t,layer} < 5$$

Bounding the new decision variables is necessary: if we would purchase 5 lots, it would not match with the fact that any set of 4 lots is automatically grouped in a layer. Note that this decomposition has to be taken into account in the model because of the price structure that decreases according to the conditioning units. If it would not be the case, it would be much cleverer to postpone the calculation of conditioning units after the optimization engine would have solved the problem.

Here, the two first new variables are bound by 4 and 5 according to the decomposition scheme itself. The last variable counting the number of purchased pallets can also be bounded by the sum of the demand from period  $t$  to the end of the horizon. Therefore, we are able to give upper bounds to our entire new integer decision variables (see table 2). Once again, this is common knowledge in combinatorial optimization that providing best bounds from the beginning ("presolve" phase) to decision variables is critical for the good behaviour of the resolution phase [3].

	1	2	3	4	5	6
1	4	4	4	4	4	4
2	5	5	5	5	5	5
3	5	4	3	2	1	1

Table 2: upper bounds for conditioning unit decision variables

Les contraintes de conditionnement des produits en unités logistiques font appel à la division entière. Supposons qu'un produit soit vendu au minimum par lots de 6 unités, que 4 lots forment une couche et que 5 couches forment une palette. On peut ainsi relier par division entière le nombre d'unités logistiques à la quantité commandée par l'équation (3) en regard.

Borner les variables entières introduites pour représenter le nombre d'unités logistiques commandées (lot/couche/palette) est nécessaire. En effet, on ne peut commander 5 lots car cela correspond en fait à une couche plus un lot. De plus, la structure de coût d'achat est elle aussi dépendante du conditionnement.

Sans cette dernière contrainte, une approche heuristique visant à reconstruire le nombre de lots/couches/palettes en post-optimisation (heuristique d'arrondi) aurait pu être envisagée pour simplifier le modèle.

Ici, les deux premières variables (lot et couche) sont majorées respectivement par 4 et 5. Il est possible également d'inférer une borne supérieure pour le nombre de palettes en considérant qu'au maximum, on pourra commander à une période donnée au plus la quantité totale équivalent palettes sur le reste de l'horizon. La table 2 montre ainsi pour un produit les bornes supérieures des nouvelles variables sur chacune des périodes.

Rappelons qu'il est toujours utile de borner de manière aussi serrée que possible les bornes des variables par des considérations d'inférence dès le départ (notion de « presolve ») et éventuellement au cours de la recherche arborescente (voir par exemple [3]).



---

## 5. PERFORMANCE ANALYSIS / ANALYSE DES PERFORMANCES

---

We derived from the original drivers three classes of problems:

1. The uncapacited multi-items procurement model
2. The full-truck capacited procurement model
3. The stepwise capacited procurement model (the trucks may be partially filled)

These three model have been captured in a single generic OPL model. The OPL syntax allows one to map variables to sparse data straightforward. For instance, one item with only two conditioning unit will have just two decision variables for each period of time.

Not surprisingly, the two latter ones are the most difficult to solve. The full-truck one is difficult due to precision of the integrity constraint that matches the sum of all items orders with an exact multiple of a fixed number of pallets that is the truck capacity.

Even building a nice model for the stepwise problem has been also an issue. We first read a survey [4] that addresses similar models with a stepwise cost structure but in this study, the steps depend more on the number of trucks that on the thresholds within the trucking of each truck. The additional difficulty comes from the fact that the tax of partially filling the truck does not apply simply to the number of pallets in the truck but to the purchasing value of these pallets. One then have to not only decompose any order into conditioning units but also decide for each unit how it decomposes into different areas of any of the available trucks. Typically, if you order 5 lots of milk bottles, 2 of them may be placed on the bottom of one truck and the remaining on the top of a second truck...

Optimality gaps vary from 0% for driver 1 to 50% for drivers 2 and 3. No particular work has been done according to literature [1] to add lot-sizing cuts. However, the results are operational while not (or not proven) optimal.

Les scénarios envisagés initialement ont engendré 3 classes de modèles :

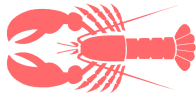
1. Le modèle d'approvisionnement multi-produit sans contrainte de capacité (camion)
2. Le modèle d'approvisionnement à camion complet
3. Le modèle d'approvisionnement « par seuils » (de remplissage partiel des camions)

Ces 3 modèles ont été capturés dans un seul fichier générique OPL. La syntaxe OPL permet de manière immédiate l'indexation de variables sur des données creuses. Ainsi, un article conditionné uniquement en couche et palettes (et non lots), aura deux variables de décision par période et non trois.

Les deux derniers modèles sont sans surprise les plus difficiles à résoudre. La contrainte couplante de remplissage de camion complet pose en effet des problèmes de précision numérique en tentant d'égaliser la somme des ordres d'achat exprimé en palettes avec un multiple de la capacité fixe du camion.

La structure par paliers du problème 3 a été approchée dans une étude intéressante [4] dans laquelle les seuils correspondent au nombre de camions engagés et non aux niveaux de remplissage des camions. Ce qui complique encore la modélisation dans notre cas est que la taxe s'applique par paliers de remplissage mais à la « valeur d'achat du palier ». Il faut donc pour chaque ordre d'achat non seulement décomposer en unités logistiques mais aussi savoir pour chacune de ces unités sa décomposition dans les différents paliers des camions afin de reconstituer la valeur du palier. Ainsi, 5 lots de bouteilles de lait pourront être dans ce modèle en partie dans le fonds d'un premier camion et la quantité restante dans le haut d'un second camion...

Les gaps d'optimalité varient de 0% pour le modèle 1 à 50% pour les modèles 2 et 3. Nous n'avons pas inclus de modèle de génération de coupes type lot-sizing[1]. Quoiqu'il en soit, les premiers résultats sont directement opérationnels même s'ils ne sont pas optimaux (ou non prouvés optimaux).



## 6. MOVING FORWARD / PERSPECTIVES

Using state of the art optimization platforms such as ILOG ODMS, OPTIMJ, COMET, COIN-OR, CHOCO and a good experience of modelling prove to be very successful to address our customer's requirements. We've been able to deliver in two weeks only a full coordinate procurement model for realistic set of products to be purchased by a hard-discount European leader.

You see on figure 4 below a snapshot of ODM what-if analysis tool that allows the user to easily create and duplicate scenarios. Here, we compare the inventory level of a given item subject to nominal purchasing prices (lower curve) with an "inflation scenario" where the prices increase from period 5. We clearly notice the order anticipation at period 4 that reflects a "purchase-to-stock" policy rather than buying at higher prices.

Last but not least, we also integrated our model in our client full JAVA application straightforward due to powerful APIs in OPL. Note that this is actually integration, not at all code rewriting! Therefore, we keep our model portable thanks to a clear separation between the model and the data, it will be very easy to test new models and enhance existing ones without entering in tedious object oriented code maintenance. We are looking forward to enlarging the scope of such application to different areas of supply chain management.

Avoir une bonne expérience de la modélisation trouve toute sa valeur lorsqu'on la marie aux plateformes logicielles matures type ILOG ODMS, OPTIMJ, COMET, COIN-OR, CHOCO. Nous avons ici délivré en deux semaines seulement une application gérant l'approvisionnement coordonné pour un leader européen du hard-discount.

La figure 4 ci-dessous présente une vue d'écran de l'application dans le gestionnaire de scénarii ODM qui permet la création et la duplication de scénarii « what\_if ». Ainsi, nous comparons ici le niveau de stock pour un produit dont les prix augmenteraient de manière importante à partir de la période 5. La courbe supérieure indique dans ce cas un niveau de stock supérieur pour anticiper cette inflation.

Nous avons dans le même temps fourni l'intégration dans un contexte pur JAVA grâce aux APIs d'OPL. A noter qu'il s'agit bel et bien d'une intégration et pas du tout d'une réécriture du code. Nous préservons ainsi la nette séparation entre modèle et données, ce qui rend l'écriture de nouveaux modèles ou l'amélioration des existants beaucoup plus aisés que ne serait la réécriture fastidieuse de code objet par exemple. Nous envisageons avec beaucoup d'enthousiasme d'appliquer cette approche gagnante dans d'autres secteurs de la gestion de la chaîne logistique.

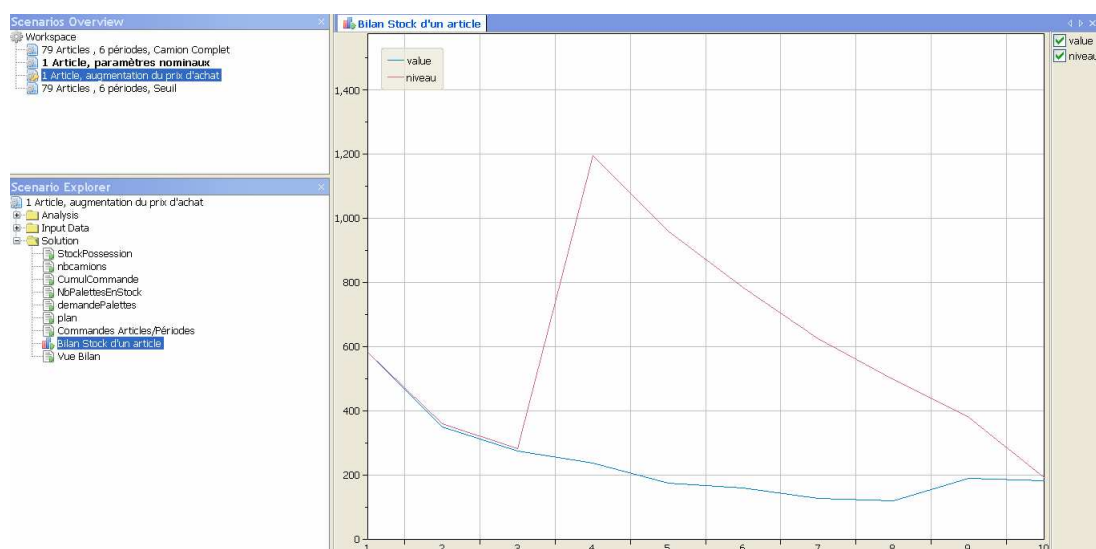
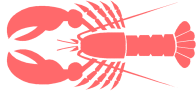


Figure 4: snapshot of ODM environment for scenario and what-if analysis



---

## 7. BIBLIOGRAPHY / BIBLIOGRAPHIE

---

- [1] Pochet – Wolsey : Production Planning by Mixed Integer Programming . Springer
- [2] HP Williams : Model Building in Mathematical Programming –4th Edition WILEY
- [3] John N. Hooker : Integrated Methods for Optimization – Springer 2007
- [4] MéliSSa Levasseur . Approvisionnement coordonné de plusieurs articles avec demande dynamique et coût commun de commande. Rapport de MBA université Laval (Québec – Canada)