



Three-level inventory deployment for a luxury watch company facing various perturbations

Jean Respen¹, Nicolas Zufferey^{1*} and Philippe Wieser²

¹Geneva School of Economics and Management, GSEM, University of Geneva, Uni-Mail, 1211 Geneva 4, Switzerland; and ²Logistics, Economy and Management Chair, EPFL, Lausanne, Switzerland

A well-known Swiss watch brand, active in the top-end luxury market, is facing a complex inventory deployment problem where watches of different models (more than 100 different models) must be dispatched first to wholesalers to finally reach the shops where consumers come in. Along the way, different perturbations are expected at three levels (production plan, demand, and dispatching), and accurate reactions must be taken to fit to these uncertainties. Solution methods are proposed to solve realistic instances. Results show the relevance of the methods and the robustness of the solutions.

Journal of the Operational Research Society (2017) **68(10)**, 1195–1210. doi:10.1057/s41274-016-0136-9; published online 16 December 2016

Keywords: inventory deployment; perturbed dispatching; perturbed production; heuristics

1. Introduction

In this paper, a well-known Swiss watch brand (denoted SWB), which is active in the luxury industry, is facing a multi-level inventory deployment problem called (P), where watches (called *items*) of different models (called *products*) are produced, and must be sent to the final shops while satisfying various constraints. Inventory deployment is often regarded as a very complex problem by companies, as different constraints and perturbations often raise along the different levels (i.e. from the production to the final consumers), and the dispatching system must react accordingly. The supply chain is composed of three different levels, namely a single factory, the wholesalers (WS), and the final destinations, which are the shops where the goods are sold to the consumers. The factory is the only place where the goods are produced. For each considered geographical region, there exists at least one wholesaler owned by SWB which then delivers to its associated set of shops. Each shop can only be delivery by its single associated wholesaler. As far as SWB is concerned, the shops are the final points of interest, and the final consumer, who actually buys an item, is not integrated in the system as the shops are usually not owned by SWB. Thus, in the following, the *customers* of SWB are the WS, and the *consumers* are the persons actually buying the items at the shops' level. Unfortunately, in (P), three different perturbations are encountered along the supply chain: first at the production plan level, then at the dispatching level (WS), and

finally on the demand (i.e. the actual demand is different from the forecasted demand). These perturbations are, respectively, due to the suppliers, the behaviours of the WS (on which SWB has not a total control), and the behaviours of the consumers.

The considered market involves a relatively small number of watches but a very high price per watch. Even if we tackle a three-echelon inventory deployment problem which would involve three types of decision-maker (i.e. plant, WS, shops), the only true and crucial decision-maker is located at the plant level, which has to decide on the deployment of the produced items (and thus also on the assortment planning), following an imposed push policy (from the plant to the shops, transiting through the WS). In other words, no inventory is kept at the production and WS levels (for safety, strategic, and confidential reasons). As soon as an item is produced, it has to be shipped to a shop (through a wholesaler), and the planned shop is decided at the production level. We, however, account for the behaviour's of the decision-makers located at the WS' level. Indeed, we propose to model the perturbations on their dispatching decisions (which do not always match the decisions made at the plant level). Therefore, the shops (resp. WS) do not order the watches to the WS (resp. plant), but they only provide them with information on their on-hand inventories. In other words, the WS and the shops have to agree on getting what the plant level has planned to send to them. Therefore, given this restrictive and imposed push approach (indeed, SWB only wants to improve its current practices), no inventory management policy is used along the supply chain (e.g. reorder point s with a fixed order quantity Q , reorder point s with order-up-to-level S).

As the watches produced by SWB are expensive, the suppliers have a key role because the components used to

*Correspondence: Nicolas Zufferey, Geneva School of Economics and Management, GSEM, University of Geneva, Uni-Mail, 1211 Geneva 4, Switzerland.

E-mail: n.zufferey@unige.ch

produce the luxury items are expensive and rare. Thus, there is no long list of suppliers which can provide the components. The number of products and the number of items yearly produced are both increasing with the demand, which makes the plant reach its production capacity limit. From a SWB perspective, the consumer wants his/her watch when he/she has decided to buy it. There are thus inventories in the stores, from which the final consumer is immediately delivered as soon as he/she has decided to buy a watch. As a consequence, a perturbation of the production plan due to suppliers has a strong impact on the availability of the items at the shops' level and is hard to manage, as every item that is not produced due to unreliable suppliers must be produced later and therefore disturbs the production plan of the following weeks.

The planning horizon is a year, and the time unit is a week. (P) is a three-level inventory deployment problem, where the main decision to make is the number of items of each product to send (from the plant and through the WS) to each shop on a per week basis. A *solution* is an aggregation of these numbers for a whole year. Even if the final clients of the plant are the WS, SWB must integrate the shops in their supply chain, as they are the only actors dealing with the consumers' demand. The objective function involves three different components: shortages, rarity, and inventory. The rarity of a solution is defined as the opposite of the diversity. For a given time, the diversity indicator of a shop is the number of different available products it contains. Then, the diversity of a solution is the summation of the diversity indicators over all the shops and time periods. In an ideal solution, at least one item of each product is represented in each shop for each time period. Thus, for a given time period, if a product is not represented in the assortment of a shop, then a rarity penalty is applied. One can already observe that rarity and inventory are conflicting components of the objective function. Indeed, a low rarity value leads to high inventory costs.

Most of the literature involves only two levels. There is no literature dealing with such a complex problem (P) with unreliable behaviours of suppliers, WS, and consumers. Moreover, the perturbation on the production plan can result in an infeasible solution, and thus, a specific repair procedure is requested. This paper aims at proposing dedicated solution methods and a simulator to improve the quality of the inventory deployment of SWB. The simulator is used to evaluate the quality of a solution according to the above described random events and is helpful to conduct a sensitivity analysis, allowing to identify the key perturbation parameters (i.e. the ones which have a significant impact on the objective function). One of the proposed solution methods shows that the use of relevant distance functions can be very powerful for tackling (P), where exact methods are not appropriate on realistic instances. Considering the computation time limit imposed by SWB, the best method is a matheuristic, which combines the accuracy of an exact method with the speed of a heuristic. If larger time limits are allowed, the use of local search approaches is beneficial.

The paper is organized as follows. In Section 2, an exhaustive literature review is provided. In Section 3, the three different perturbations faced by SWB are first exposed, and then, a formal description of (P) is proposed. Section 4 describes the different solution methods used to provide a solution *S*, and the simulator is designed to evaluate the quality of a solution *S*. The experiments are presented in Section 5. Section 6 concludes the paper and identifies possible future works.

2. Literature review

As explained in Glasmeier (1991), the Swiss industry lost control on the watch industry in the 1990s, due to the different actors gaining power in the technological field. But, as of today, Swiss luxury watches are selling in an increasing fashion each year (FSWIFH, 2014), which shows how the Swiss luxury industry was able to overcome its past drawbacks to return to expansion. Switzerland is not the only country facing luxury expansion. In Brun *et al* (2008), the authors state that in Italy, the luxury segment was worth 26 billion Euros per year in 2006 and is growing. The paper aims at investigating 12 luxury Italian firms of different luxury segments, to define the behaviours of each segment in the supply chain strategies. Extending the previous paper, the authors of Caniato *et al* (2011) propose to derive different clusters of companies and identify which supply chain strategy is applied in each cluster. In Catry (2003), a study is proposed on how luxury companies are adept of pretending to sell rare and exclusive products, even if the quantity of each product is so important that it does not belong to rarity anymore.

Usually, the actors of the Swiss watch industry started with a small family-sized company and gained world-class profits, focusing on the products and having only basic knowledge of supply chain management. This trend tends now to be reversed and many companies are trying to optimize their supply chain systems. In Hnaien *et al* (2010), a two-level assembly problem is tackled using a multi-objective approach based on genetic algorithms. Uncertainties appear in lead-times, and the main objective is to maximize the service level from the client point of view. In Silver and Zufferey (2005), a known demand and uncertain lead-times are expected, where the perturbations of the lead-times depend on the season. In such a situation, heuristics are proposed and their performance are analysed. Extending the previous paper, perturbations on the lead-times are again expected in Silver and Zufferey (2011), but the company faces a planned shutdown period. Simulation is used to assess the quality of the proposed solution methods.

As also discussed in the above papers, to correctly assess the quality of the models and methods, simulation has to be used. With regard to simulation, different interesting surveys can be found in Banks (1998), Law (1991), Terzi and Cavalieri (2004). A reverse logistics problem is tackled in Kara *et al* (2007), where defective products must be brought back to the

factory from the client level. The simulator is used to assess the robustness of the presented model. A simulator is used in Daniel and Rajendran (2005) to tackle an inventory deployment problem using a genetic algorithm. The robustness of the algorithm is tested with different supply chain settings. In Miranda and Garrido (2004), an inventory control problem is considered with stochastic demand, using either a nonlinear mixed integer model and a heuristic based on Lagrangian relaxation. The exact model achieves impressive results on this problem, whereas the heuristic is used for larger instances only. In Ben-Daya and Hariga (2004), a problem with stochastic demand and dynamic lead-times is investigated and an exhaustive sensitivity analysis is performed for the important parameters (e.g. lot sizes, reorder points). In Schmitt *et al.* (2010), a model is proposed for a problem with complete supplier disruptions. Different levels of uncertainty are modelled (e.g. demand, disruptions) and a closed-form approximation method is proposed. In Anupindi and Bassok (1999), the problem of supply contracts under quantity commitment and stochastic demand is studied. Simulation-based optimization approaches are proposed in Otamendi and Doncel (2012) for the entire transmission problem (i.e. arrival of ships, regasification, transportation, and distribution) faced by gas utilities companies and in Tsai and Zheng (2013) for a two-echelon inventory system with service-level constraints.

In Madadi *et al.* (2010), a multi-level inventory management problem is tackled, where suppliers, warehouses, and retailers are considered, and transportation costs are encountered. Centralized and decentralized ordering models are compared. A two-level inventory problem with one warehouse and many retailers is proposed in Forsberg (1995), where the retailers face different compound Poisson demand processes, and the facilities apply order-up-to- S replenishment policies. A very similar problem is proposed in Axsäter (1993), where simple recursive procedures are proposed to evaluate the shortage costs. In Axsäter and Juntti (1996), distribution systems with stochastic demands are analysed using simulation. Advantages of echelon stock (Clark and Scarf, 1960) and installation stock (Lee and Moinzadeh, 1987; Moinzadeh and Lee, 1986) are assessed. In Segerstedt (1996), a mathematical formulation is presented for a capacity-constrained multi-stage inventory and production control problem. Numerical experiments for a trivial version of the problem are presented, and links among other problems are proposed [e.g. Kanban (Bard and Golany, 1991), cover-time planning (Segerstedt, 1991)]. In Yao *et al.* (2009), three different inventory strategies are designed for a two-level inventory deployment problem, and managerial insights are provided. In Svoronos and Zipkin (1988), the authors propose methods to assess the performance of different multi-level inventory strategies. In Kunnumkala and Topaloglu (2011), the authors consider an inventory–distribution system consisting of one warehouse and multiple retailers. The retailers face random demand and are supplied by the warehouse. The warehouse replenishes its stock from an external supplier, and the objective is to minimize the total

expected replenishment, holding and backlogging cost over a finite planning horizon. Linear-programming-based decomposition methods are proposed to tackle this problem. In Schildbach and Morari (2016), a scenario-based model is designed for multi-echelon supply chain management. The authors showed that except for a few special cases, optimal solutions are computationally intractable for systems of realistic size. Their approach can handle supply chains with stochastic planning uncertainties from various sources (e.g. demands, lead-times, prices).

In Melo *et al.* (2009), an interesting survey is proposed on facility location and supply chain management, in addition to recent advances in optimization techniques for these problems and for inventory deployment. In line with the current trend but in contrast with the practice of SWB, an integrated inventory management model is depicted in Shafieezadeh and Sadegheih (2014) for multi-item multi-echelon supply chains. The benefit of managing jointly different decision levels will be highlighted in the sensitivity analysis performed in Section 5.3.

3. Presentation of problem (P)

In this section, the different perturbations are explained, followed by a formal problem description of (P).

3.1. Perturbations

As explained earlier, the complex process of producing watches suffers multiple uncertainties. Three different perturbations are exposed, namely perturbations on the demand, perturbations on the production plan, and finally perturbations on the dispatching of the items at the WS' level. The intervals in which the perturbations parameters belong to are set thanks to the SWB knowledge of their supply chain.

3.1.1. Perturbations on the demand There is obviously a gap between the forecasted and the actual demands encountered at the shops' level. For SWB, the consumers are wealthy persons, who wants the item at the time they decide to buy it. From forecasted demands (given by SWB) and following the historical observations of SWB, we generate perturbations based on a normal distribution. Thus, each shop has its own mean and standard deviation for each product, used to generate the perturbation on the demand for each week. The means and standard deviations are set with the help of SWB to accurately reflect its reality.

3.1.2. Perturbations on the production plan For various reasons (suppliers being unreliable, problems in the deliveries of the components, etc.), two different types of perturbation on the production plan can occur every week. The first one sees some products being not produced at all

(because the associated components were not delivered), whereas the second sees some items of some products being not produced (as the production was slower than expected). In other words, the production plan is decided (i.e. fixed), but it can suffer from perturbations. The challenge is thus to develop an inventory deployment method which can react appropriately to the uncertainty associated with the production plan.

Let \hat{p}_t^i be the planned number of items of product i to produce at week t , and p_t^i be the corresponding actual number (i.e. with the perturbation). Each week, a maximum of δ_1 per cent of models are not produced, due to non-deliveries from suppliers (i.e. $p_t^i = 0$ for these models, instead of $p_t^i = \hat{p}_t^i$). The perturbation affects 2 weeks, and the corresponding production must be performed later. If a component is not delivered to the plant at period t , SWB assumes that it will be necessarily delivered by period $t + 2$. During period $t + 2$, SWB is then always able to adjust its production capacity in order to recover from this previous components' shortage. Thus, $p_t^i = p_{t+1}^i = 0$, and from a practical standpoint, we can always set $p_{t+2}^i = \hat{p}_{t+2}^i + \hat{p}_t^i + \hat{p}_{t+1}^i$. SWB proposes that δ_1 belongs to interval $[5, 15]$.

On the remaining $1 - \delta_1$ per cent of the models (i.e. the ones which are produced), each model has δ_2 per cent of risk to be slowed down. Slowed down means that at most δ_3 per cent of \hat{p}_t^i is not produced at week t but at week $t + 1$. At the end of week t , the total number of items not produced due to slowed down performances is $Q_t = \hat{P}_t - P_t$, where $\hat{P}_t = \sum_i \hat{p}_t^i$ and $P_t = \sum_i p_t^i$. On the total number \hat{P}_t of items expected to be produced at week t , a given threshold of ε per cent of items are allowed to be slowed down, meaning that if $Q_t \leq \varepsilon \cdot \hat{P}_t$, then the factory does not take any action. But if $Q_t > \varepsilon \cdot \hat{P}_t$, then $C_t = Q_t - \varepsilon \cdot \hat{P}_t$ watches must be compensated (meaning that items of different models are produced instead of planned ones) with the following rule: models which are not slowed down at week t and planned at week $t + 1$ are eligible for compensation, and the models which were not expected to be produced at week t (i.e. $\hat{p}_t^i = 0$), but are expected to be produced at week $t + 1$, can be eligible for compensation as well. For the eligible models, some watches (randomly selected, one at a time) have to be produced at week t , until quantity C_t is fulfilled, as explained further in Section 4.2. SWB proposes the following intervals for the above discussed parameters: $\delta_2 \in [45, 55]$, $\delta_3 \in [5, 15]$, $\varepsilon \in [5, 15]$. Note that, for each product, SWB has a fixed annual quantity to produce, which has to be met in any case. The current practices of SWB allow to assume that SWB is always able to adjust its production capacity if a compensation action is triggered.

3.1.3. Perturbations on the dispatching from the wholesalers As shown in Figure 1, the supply chain network linking the factory, the different WS, and the shops is a three-level inventory model. The WS belong to SWB and are the only

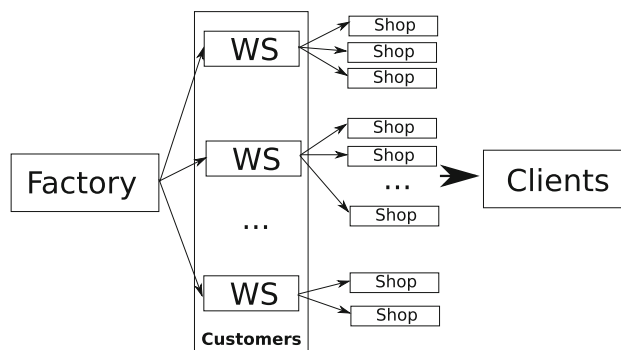


Figure 1 Inventory deployment network.

customers regarding SWB. It is expected that the different WS manage the received inventory and deliver items to their corresponding shops in a reliable manner, but unfortunately perturbations occur. Indeed, each wholesaler has its “confidential” list of preferred shops, and it can decide to ship an item to a preferred shop instead of the one initially planned by SWB at the plant level. The perturbation which can happen is the following: SWB estimates a target inventory for each final shop and sends the corresponding items to the associated wholesaler. Each shop j has a priority w_j and let w^{\max} be the largest possible priority (i.e. the priority of the most important shops). Any wholesaler can perturb the SWB estimations by sending the items dedicated to a given shop j to a different shop j' . SWB assumes that δ_4 per cent of the items does not go to the expected shop. On these δ_4 , δ_5 per cent would go to a shop with a higher priority. SWB proposes to use $\delta_4 \in [30, 50]$ and $\delta_5 \in [60, 80]$.

3.2. Formal description

To deeply understand (P), we propose a formal description. N models (i.e. products) are produced in the factory. The number of weeks in a year is W . Each week t , p_t^i items of model i arrive to the different WS around the world. At the end of each week, because of safety/strategic/confidential reasons, all the produced items are sent to the shops (through their associated WS), and thus, none are kept in inventory at the plant and WS levels. Such a push policy is imposed by SWB which wants to improve its current practices. As a consequence, no inventory management policy is used by the shops and the WS. J shops must be supplied by the different WS, and each shop j is associated with exactly one wholesaler m_j (thus, we can have $j \neq j'$ but $m_j = m_{j'}$). The total number of WS is M . Let I_t^{i,m_j} be the on-hand inventory of wholesaler m_j regarding model i at week t . Each wholesaler delivers to its shops and tries to empty its inventory. Indeed, the WS are owned by SWB, which naturally prefers to put the inventory costs at the shops' level. Let $\hat{x}_t^{i,j}$ be a decision variable (managed by the only key decision-maker, located at the plant level) describing the number of items of model i supposed to arrive at shop j at week t from the corresponding wholesaler m_j (it thus

corresponds to a planned shipment from the plant). One can observe that the $\hat{x}_t^{i,j}$'s and the \hat{p}_t^i 's are related. Indeed, the former variable is a part of the latter, but two different times are involved as the shipment from the plant to the shop requires a lead-time. As explained in Section 3.1.3, each variable $\hat{x}_t^{i,j}$ suffers a perturbation, and let $\tilde{x}_t^{i,j}$ be the resulting variable involving the perturbation. In other words, $\tilde{x}_t^{i,j}$ (resp. $\hat{x}_t^{i,j}$) is an expected (resp. actual or observed) value. Moreover, a lead-time of L_{m_j} weeks occurs from the plant to the wholesaler m_j . Another lead-time of L_j weeks is then required to reach the shop j from the corresponding wholesaler m_j . For each week t , a demand $\hat{d}_t^{i,j}$ is forecasted for model i at shop j and is subject to the perturbation depicted in Section 3.1.1. Thus, let $d_t^{i,j}$ be the corresponding actual demand. As soon as the items $\hat{x}_t^{i,j}$ reach the shop j , the corresponding on-hand inventory of shop j is updated: $I_t^{i,j} = \max(0, I_{t-1}^{i,j} - d_{t-1}^{i,j} + \hat{x}_t^{i,j})$ [we also have the expected updating: $\tilde{I}_t^{i,j} = \max(0, \tilde{I}_{t-1}^{i,j} - \hat{d}_{t-1}^{i,j} + \hat{x}_t^{i,j})$]. Thus, the inventory at period t is set as the inventory at period $t - 1$ augmented by the number of delivered items and decreased by the demand at period $t - 1$. A solution \hat{S} can be denoted $\hat{S} = (\hat{S}_1, \hat{S}_2, \dots, \hat{S}_t, \dots, \hat{S}_W)$, which is a per week list of \hat{S}_t 's, where $\hat{S}_t = (\hat{S}_t^1, \hat{S}_t^2, \dots, \hat{S}_t^i, \dots, \hat{S}_t^N)$ is the corresponding solution for week t for each product i . For each week t and each product i , $\hat{S}_t^i = (\hat{x}_t^{i,1}, \hat{x}_t^{i,2}, \dots, \hat{x}_t^{i,j}, \dots, \hat{x}_t^{i,J})$ is a vector of decision variables for each shop j .

The overall objective function f to minimize involves three different components f_1, f_2 and f_3 to minimize in a lexicographic order (i.e. no deterioration on f_i can be compensated by improvements on f_{i+1}), as motivated in SWB (2014). More precisely, one can define f as $\min f = \alpha \cdot f_1 + \beta \cdot f_2 + \gamma \cdot f_3$, where $\alpha > \beta > \gamma$ are chosen to respect the lexicographic order (which is ensured for example with $\alpha = 1,000,000,000$, $\beta = 1,000,000,000$ and $\gamma = 1$). In preliminary experiments, we have encountered several solutions with equivalent values for f_i while satisfying an upper bound on f_{i-1} (for $i \in \{2, 3\}$). This actually means that all the components of f are relevant and none can be ignored. In other words, even f_3 can be the discriminating component. This observation is in line with the findings of Respen *et al* (2016), where lexicographic optimization (involving three components as well) was applied in an industrial context arising in the car industry.

Considering (P), f_1 is the expected shortage penalty for each item i for each week t in each shop j , which occurs if $\tilde{I}_t^{i,j} < \hat{d}_t^{i,j}$. Let $\hat{B}_t^{i,j} = \max(0, \hat{d}_t^{i,j} - \tilde{I}_t^{i,j})$ be the expected shortage quantity of model i at week t in shop j , and $\hat{v}_t^{i,j}$ be a binary value which is 0 if $\hat{x}_t^{i,j} > 0$, and 1 otherwise. Then, f_1 can be described as in Equation (1). It is the sum of the shortage penalties, where a shortage which has started a long time ago

costs more than a recent shortage. SWB assumes that if a watch of model i , which is assumed to be in shortage, arrives in a shop j at week t , then the considered shortage penalty is reset to 0 from that time period, even if the demand is above the resulting on-hand inventory. In other words, a consumer wants his/her watch when he/she decided to buy it; otherwise, the sale is lost. As long as a product i is not represented in a shop j , a shortage penalty is applied by SWB. Then, as soon as an item of product i arrives at the considered shop j , the penalty disappears even if the demand at that time is above the number of delivered items of product i . The removal of the shortage penalty is part of the strategy used by SWB, which focusses on the presence or absence of product i in shop j .

$$f_1 = \sum_{j=1}^J w_j \sum_{t=1}^W \sum_{t'=1}^{t-1} (t-t') \sum_{i=1}^N \hat{B}_t^{i,j} \cdot \hat{v}_t^{i,j} \tag{1}$$

Note that t' is used to sum the shortage penalty from the time it begins to the current time, in order to penalize the shop until the considered product is present in its assortment. Let $\hat{y}_t^{i,j} = 1$ if $\tilde{I}_t^{i,j} = 0$, and $\hat{y}_t^{i,j} = 0$ otherwise. Objective f_2 is described in Equation (2), which is a measure of the rarity of each model i in each shop j for each week t . As defined in Section 1, the larger is f_2 , the poorer is the assortment planning of the solution.

$$f_2 = \frac{1}{J \cdot W \cdot N} \cdot \sum_{t=1}^W \sum_{j=1}^J \sum_{i=1}^N \hat{y}_t^{i,j} \tag{2}$$

The last objective f_3 is described in Equation (3), which is a weighted inventory penalty.

$$f_3 = \sum_{j=1}^J [(w^{\max} + 1) - w_j] \left[\sum_{t=1}^W \sum_{i=1}^N \tilde{I}_t^{i,j} \right] \tag{3}$$

Observe that the inventory penalty $(w^{\max} + 1) - w_j$ depends on the priority of the involved shop. Indeed, as SWB makes more margins with the shops with higher priorities, SWB prefers to allow more stock in such important shops. For this reason, the inventory penalty is lower for the shops with a higher priority.

To ensure a better understanding of the components involved in the objective function, we now propose an example. Consider an instance with $J = 1$ shop (with $w_1 = 1$ and thus $w^{\max} = 1$), $N = 1$ product, $W = 4$ weeks, and the initial inventory at the shop is empty (i.e. $\hat{I}_0^{1,1} = 0$). Furthermore, consider solution \hat{S} given by $\hat{x}^{1,1} = (1, 2, 0, 3)$. Assume that the forecasted demand is $\hat{d}^{1,1} = (1, 3, 0, 1)$. Since only $\hat{x}_3^{1,1}$ has a zero value, we have only $\hat{v}_3^{1,1}$ positive (equal to 1). Therefore, $f_1 = \sum_{t'=1}^2 (3-t') \cdot \hat{B}_t^{1,1} = 2 \cdot \hat{B}_1^{1,1} + \hat{B}_2^{1,1}$. Next,

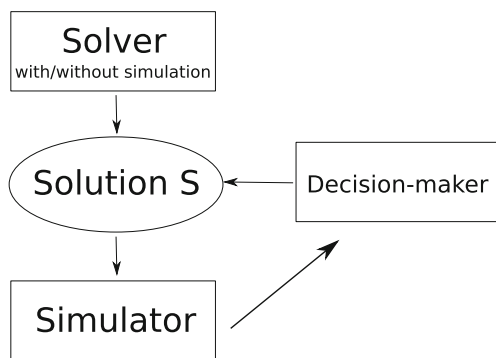


Figure 2 Sketch of the decision analysis tool.

$\hat{B}_1^{1,1} = 0$ because $\hat{I}_1^{1,1} = \hat{d}_1^{1,1} = 1$. Thus, we finally get $f_1 = 1$. We can easily compute that $f_2 = \left(\frac{1}{1.14}\right) \cdot (1 + 1 + 1 + 0) = \frac{3}{4}$ and $f_3 = (2 - 1) \cdot (0 + 0 + 0 + 2) = 2$.

4. Solution methods and simulator

To evaluate the actual value of a solution, it is mandatory to develop a simulator. Its input data are a solution, the production plan, the initial inventories (for each shop and each wholesaler, for each product), and the forecasted demand. The overall approach is presented in Figure 2. This decision process is in line with the decision-making literature, for which the reader is referred to Malakooti (2012) for having a general reference. A solver is first used to generate a solution S , which is then evaluated with the simulator. Such information (i.e. the solution S and its simulation along with the various objective function components) is then provided to the decision-maker, which can then decide to manually modify the solution (based on his/her expertise and the non-modelled elements) and simulated it again. The solver is made of a solution method, which can involve or not simulation. In the former case, the solver accounts for uncertainty (it will be for example the case for the proposed methods denoted BSS and BSSC).

Two computation time limits are considered in this study: $T = 30$ min and $T = 10$ h. A strict time limit of $T = 30$ min (including simulation) is imposed by SWB. This forbids the use of time-consuming methods such as metaheuristics. This restrictive time limit is imposed because of managerial reasons. For instance, after the use of the solution method, the involved decision-maker has to take the provided solution as an input and rework on it (relying on non-modelled features) based on the process depicted in Figure 2. Such an optimization process involving the decision-maker was also confirmed in other

studies [e.g. Syntetos *et al* (2011)]. We will see in Section 4.3 that simulating one solution requires roughly 3 min on the used computer. For these reasons, constructive methods are appropriate to tackle (P) if $T = 30$ min. However, in order to further evaluate the quality of the proposed constructive algorithms, a time limit of $T = 10$ h is also considered, allowing the use of more advanced metaheuristics, namely a descent local search DLS and a tabu search TS.

In the remaining part of this section, three constructive solution methods are proposed in Section 4.1, the simulator is presented in Section 4.2, important computing time aspects are discussed in Section 4.3, and metaheuristics are proposed in Section 4.4 (for the case $T = 10$ h).

4.1. Constructive solution methods

We first design three solution methods, which rely on a population of pop (parameter) solutions and use a constructive heuristic called *SmartPriorityGreedy* (denoted SPG) as an underlying method. SPG is depicted in Algorithm 1, where $A_t^{i,w}$ is a set that contains all the shops with priority w that have a strictly positive forecasted demand for week t and model i , and A contains all the shops with the highest priority w^{\max} (regardless of the demand). Note that for each shop and each wholesaler, the initial associated inventory level (i.e. the available quantity) of each model is a given as input. Next, for each week t and each model i , the available items are the sum of the produced items in the factory and the items available at the WS' level. At each step, the algorithm attributes one item to a shop (through its associated single wholesaler). After intensive tests, the most efficient method (according to speed and results' quality) consists in randomly selecting the shop to which an item is assigned in the considered set $A_t^{i,w}$. Using more advanced methods (using f_1 , f_2 and f_3 for example) dramatically slows down the overall process, without improving the quality of the solutions. As an example, on an instance with $N = 143$, $W = 47$, $J = 192$ and $M = 9$, a method which computes f_1 , f_2 and f_3 at each step to attribute the best item to the best shop requires more than 15 h to return a solution! Here, the efficiency of SPG relies on the design and efficient use of the $A_t^{i,w}$'s. f_1 and f_3 are very influenced by the weights of the shops, and creating the $A_t^{i,w}$'s allows to very quickly and efficiently improve these objective components. If all the expected demands of model i of all the shops have been fulfilled for the considered week t , the last phase of SPG is triggered [see the last part of step (3c) in Algorithm 1]. This phase consists in shipping the remaining items (i.e. for which there is no demand) to the shops with priority w^{\max} .

Algorithm 1 SmartPriorityGreedy (SPG)

Input: production plan, forecasted demand, inventory levels (at both shops' and wholesalers' levels), shops' priorities, set A

For each week t and each model i , **do**

1. Compute the sets $A_t^{i,w}$ (for each w).
2. Select the set $A_t^{i,w}$ with the highest priority w as the current set \mathcal{C} .
3. **For** each available item, **do**
 - (a) Assign the item to a randomly chosen member of \mathcal{C} .
 - (b) If a member of \mathcal{C} has its expected demand fulfilled, remove it from \mathcal{C} .
 - (c) If $\mathcal{C} = \emptyset$, go to step (2), except if all the $A_t^{i,w}$'s have been emptied, set $\mathcal{C} = A$.

Output: inventory deployment solution S .

The first solution method, called *BestSolution* (denoted BS), generates a set of pop solutions using SPG and returns the best one (according to f). Following this method, *BestSolutionSimulated* (denoted BSS) generates a set of pop solutions with SPG, simulates the k (parameter) best ones (according to f), and finally returns the solution with the best simulated value. Finally, *BestSolutionSimulatedClique* (denoted BSSC) is a matheuristic (Boschetti *et al.*, 2009).

To properly understand BSSC, let us first introduce a distance function $\text{dist}(\hat{S}_1, \hat{S}_2)$ which returns the distance between two solutions \hat{S}_1 and \hat{S}_2 . We propose to set $\text{dist}(\hat{S}_1, \hat{S}_2) = \sum_{i,j} w_j \cdot \lambda_t^{i,j}$, assuming that $\hat{x}_t^{i,j}(\hat{S})$ is the value of $\hat{x}_t^{i,j}$ in solution \hat{S} , and $\lambda_t^{i,j}$ is set as in Equation (4). Therefore, this distance function assumes that two solution components are distant if one is zero and the other is strictly positive (regardless of the amplitude of the positiveness), which is relevant in the context of SWB, which wants at least one item of each product available in each shop.

$$\lambda_t^{i,j} = \begin{cases} 1 & \text{if } [(\hat{x}_t^{i,j}(\hat{S}_1) = 0) \text{ and } (\hat{x}_t^{i,j}(\hat{S}_2) > 0)] \\ & \text{or if } [(\hat{x}_t^{i,j}(\hat{S}_2) = 0) \text{ and } (\hat{x}_t^{i,j}(\hat{S}_1) > 0)] \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Let $G = (V, E)$ be a graph with vertex set V and edge set E . A clique C of G is a subset of V such that its vertices are pairwise adjacent. In addition, let $w(v, v')$ be the weight associated with edge $(v, v') \in E$. We define here the weight $w(C)$ of a clique C by $\frac{1}{2} \cdot \sum_{v,v' \in C} w(v, v')$. The coefficient $\frac{1}{2}$ ensures that the weights are not counted twice. Further, the k -clique with maximum weight problem consists in searching the clique C of size k in G which maximizes $w(C)$. This problem is also known in the literature as the *maximum diversity problem*, for which extensive computational experiments are performed in Marti *et al.* (2013) to compare 10 heuristics and 20 metaheuristics. Surveys on graph theory and cliques can be found in Brandstädt and Spinrad (1999), Pardalos and Xue (1994), Schaeffer (2007). This problem can be exactly solved with an integer linear program (ILP) using CPLEX. Let $x_v = 1$

if vertex $v \in C$, and $x_v = 0$ otherwise. Let $y_{v,v'} = 1$ if $x_v = x_{v'} = 1$, and $y_{v,v'} = 0$ otherwise. The ILP model is given in Equations (5–10). Constraints (6) and (7) are used to compute correctly the value of $y_{v,v'}$, and Constraints (8) ensure that the clique size is k .

$$\max \sum_{(v,v') \in E} w(v, v') \cdot y_{v,v'} \quad (5)$$

$$x_v + x_{v'} - 1 \leq y_{v,v'} \quad \forall v, v' \quad (6)$$

$$2 \cdot y_{v,v'} \leq x_v + x_{v'} \quad \forall v, v' \quad (7)$$

$$\sum_v x_v = k \quad (8)$$

$$x_v \in \{0, 1\} \quad \forall v \quad (9)$$

$$y_{v,v'} \in \{0, 1\} \quad \forall v, v' \quad (10)$$

BSSC starts by generating a set of pop solutions with SPG, then builds a set B with the best b (parameter) generated solutions, and on this set computes distance between each pair of solutions. The above ILP clique problem is then solved to optimality with CPLEX [a vertex v represents a solution $S_v \in B$ and $w(v, v') = \text{dist}(S_v, S_{v'})$]. BSSC finally simulates only the k solutions associated with the provided clique. At the end, BSSC returns the solution with the best simulated value. The clique approach used in BSSC allows to select the solutions that differ the most (among the most promising ones), with respect to their structures. In addition to being efficient, this technique allows the decision-makers to select a solution among highly different solutions.

4.2. Simulator

The simulator returns a value $f(S)$ of a solution S after having added the different perturbations to the corresponding solution

without perturbation \hat{S} (as explained in Section 3.1) in that order: generate the demand perturbations, perturb the production plan, and finally dispatch. Algorithm 2 proposes a sketch of the different simulation steps. After the second step, which perturbs the production plan, the solution \hat{S} could possibly not be feasible anymore, as the \hat{p}_i^j 's could be different from the p_i^j 's (i.e. some items are unfortunately not produced, and others are produced in counterpart). Thus, a repairing step is performed, which ensures that the resulting solution remains feasible. Let f_k^j be the contribution of shop j to the objective function component f_k . To repair a solution, $f^j = \alpha \cdot f_1^j + \beta \cdot f_2^j + \gamma \cdot f_3^j$ is computed for each shop j , and the resulting f^j 's are sorted in an increasing fashion. If for a given week, a $\hat{x}_t^{i,j}$ has to be reduced (because of the production plan perturbations, some watches do not exist anymore), then they are one by one removed from the shops with the smallest f^j value. On the opposite, if some items must be added (because of the compensation production), then they are added one by one to the shops with the largest f^j value. A pseudo-code of the repair procedure is given in Algorithm 3. Note that even after the perturbations on the dispatching, solution S remains feasible because the watches are managed.

At the very end, the simulator returns a value $f(S)$ for the input solution \hat{S} . To properly capture the operating mode of the simulator, let us propose a small example. Assume that an instance involves one product, 2 weeks, one wholesaler, and

two shops (with $w_1 = 2$, $w_2 = 1$, and thus, $w^{\max} = 2$). All initial inventories are empty, and all the lead-times are zero. The production plan is $\hat{p}^1 = (2, 3)$ and the forecasted demand $\hat{d}^{1,1} = (1, 3)$ and $\hat{d}^{1,2} = (1, 1)$. The solver, based on the expected production plan and demand, proposes a solution \hat{S} with $\hat{x}^{1,1} = (1, 3)$ and $\hat{x}^{1,2} = (1, 0)$, as it fulfils in priority the demand for the first shop. Then, the actual demand is generated as $d^{1,1} = (1, 2)$ and $d^{1,2} = (2, 1)$, whereas the actual production plan is generated as $p^1 = (3, 3)$ (we produce more than expected, due to a perturbation which occurred before the current events). The solution then needs to be repaired, and the system returns a modified solution $\hat{x}^{1,1} = (1, 2)$ and $\hat{x}^{1,2} = (2, 1)$, which is optimal. But the behaviour of the wholesaler perturbs the solution such that $x^{1,1} = (1, 3)$ and $x^{1,2} = (2, 0)$. Finally the objective function is computed as $f(S) = 1 + \frac{1}{4} + 1 = \frac{9}{4}$ (using $\alpha = \beta = \gamma = 1$).

In order to be robust, sim (parameter) runs of the above described simulator are performed with solution \hat{S} as input. The value $f(S)$ is then the average value over the sim runs. Let $f_{\text{sim}}(S)$ be the average value of $f(S)$ after sim runs of the simulator. Parameter sim was tuned such that $f_{\text{sim}}(S)$ and $f_{\text{sim}-1}(S)$ differ by <1 . After intensive tests on various instances, we have set $\text{sim} = 40$. More generally, if $\text{sim} < 40$, there exists minor variations of the average value of $f(S)$ over the sim runs, whereas if $\text{sim} \geq 40$, only very negligible variations of $f_{\text{sim}}(S)$ are observed.

Algorithm 2 Sketch of the simulator

Input: a solution \hat{S} , the forecasted demands and the associated normal distributions for their perturbations, the inventories of the wholesalers and shops, the production plan, the shops' priorities.

1. **Perturb the demand.** For each week t , shop j and model i , compute the actual demand by adding a perturbation value (generated based on a normal distribution) to the forecasted demand.
2. **Perturb the production plan** (based on Section 3.1.2 and parameters $\delta_1, \delta_2, \delta_3, \varepsilon$). More precisely, for each week t , do:
 - (a) For each model i , do
 - i. Use δ_1 to select if the production of i is cancelled. If yes, postpone the production of i from weeks t and $t + 1$ to week $t + 2$.
 - ii. If model i is not cancelled, use δ_2 to select if its production is slowed down. If yes, use δ_3 to select the percentage of slowed down items, and postpone their productions to week $t + 1$.
 - (b) Use the threshold parameter ε to perform a compensation or not.
3. **Possibly repair the solution.** If the resulting solution is not feasible, perform the repair procedure (see Algorithm 3).
4. **Perturb the dispatching.** For each week t , shop j and model i , use δ_4 to select if the involved wholesaler will ship the items to the planned shop or not. If not, use δ_5 to decide if a shop with a higher/lower priority will get the items.
5. **Evaluate the resulting solution S :** compute $f(S)$.

Output: $f(S)$.

Algorithm 3 Repair procedure

Input: a non-feasible solution \hat{S} , the perturbed production plan.

For each week t and each product i **do**

1. Get the number of watches $E_t^i = p_t^i - \hat{p}_t^i$ that does not fit the perturbed production plan.
2. If $E_t^i < 0$, remove watches (one by one) from the shops with the lowest f^j value (i.e., reduce some $\hat{x}_t^{i,j}$ values to generate the corresponding $x_t^{i,j}$).
3. If $E_t^i > 0$, add watches (one by one) to the shops with the highest f^j value (i.e., augment some $\hat{x}_t^{i,j}$ values to generate the corresponding $x_t^{i,j}$).

Output: a repaired solution S (feasible).

4.3. Computing time considerations

Remind that the time limit T allowed by SWB is 30 min. A typical instance of SWB contains 47 weeks, from 7 to 9 WS, from 160 to 250 shops, and from 100 to 200 models (these intervals are voluntarily consequent, to respect a SWB non-disclosure agreement). SPG requires half a second to get a solution. A complete simulation (i.e. sim = 40 runs of the simulator on one solution) requires around 3 min. Based on such constraining time considerations, preliminary experiments lead to the following parameter setting: pop = 100, $b = 25$, and $k = 5$. With such values, CPLEX requires about 5 min to optimally solve the ILP clique problem [see Equations (5)–(10)]. BSS and BSSC need between 20 and 30 min to complete, whereas BS requires only 15 s. If one would reduce the computing time of BSSC, an option is to solve the ILP clique problem with a (meta)heuristic [as proposed in Marti *et al.* (2013), but without any guarantee on optimality] instead of with CPLEX. However, as BSSC meets the 30-min time limit imposed by SWB, we propose to keep the use of CPLEX.

4.4. Local search methods

Assume that an objective function f has to be minimized within a given time limit T . A *local search* algorithm starts from an initial solution (usually provided by a constructive heuristic) and tries to improve it iteratively. In each iteration, a move (i.e. a slight modification) is performed in order to transform the current solution into a neighbour solution. In a *descent local search* (DLS), the best move (according to the reduction of f) is performed in each iteration, and the process is restarted from another initial solution as soon as a local minimum is found (the best encountered solution is provided at the end). In contrast, *tabu search* (TS) continues performing moves until the time limit T is reached, but in order to prevent cycling (i.e. coming back to an already visited solution, like the last encountered local minimum), the reverse of the most recent moves are forbidden (they are called *tabu moves*) for tab (parameter) iterations. It results that in each iteration of a basic TS, the best non-tabu move is performed. The reader is

referred to Gendreau and Potvin (2010) for a book on metaheuristics, whereas Zufferey (2012a) proposes general guidelines on metaheuristics to efficiently design them depending on the encountered problem.

Considering (P), remember that each decision variable $\hat{x}_t^{i,j}$ is the number of items of model i supposed to arrive at shop j at week t from the corresponding wholesaler m_j (it corresponds to a planned shipment from the plant). For fixed values of t and i , a straightforward move (both for DLS and TS) consists in shipping an item of model i to another shop j' instead of the currently planned shop j , while possibly involving another week $t' \neq t$ because of lead-time considerations. In other words, $\hat{x}_t^{i,j}$ (resp. $\hat{x}_{t'}^{i,j'}$) is reduced (resp. augmented) by one unit. As discussed in Section 3.2, as long as a model i is not represented in a shop j , a shortage penalty is applied by SWB. In order to focus on the most promising moves, the following two conditions have to be satisfied: $\hat{x}_t^{i,j} \geq 2$ and $\hat{x}_{t'}^{i,j'} \leq 1$. In other words, for model i , an interesting move should not lead to a shortage penalty in shop j , it should not augment too much the inventory level in shop j' , it should have a chance to improve the assortment planning of shop j' , and it should have a chance to remove a shortage in shop j' . When a move is performed in TS, it is then forbidden to augment (resp. reduce) again $\hat{x}_t^{i,j}$ (resp. $\hat{x}_{t'}^{i,j'}$) for tab iterations, where tab is an integer randomly generated in interval [5, 15] (thus, each tabu feature has its own tabu duration). Preliminary experiments have confirmed that such a way of managing the tabu status is more efficient than having a single value of tab for the whole search process.

We have two evaluation measures: (1) the expected cost $f(\hat{S})$ of the expected solution \hat{S} ; (2) the simulated cost $f^{(\text{sim})}(S)$ of the actual solution S associated with \hat{S} , where sim is the number of replications (tuned to 40 for having reliable and robust results, as discussed in Section 4.2). As the simulation duration is significant (3 min/solution), not all the neighbour solutions can be accurately simulated in each iteration (otherwise <5 iterations would be performed in 10 h). For each iteration, a sensitive issue is thus the way to build a small but promising set of neighbour solutions to evaluate with $f^{(\text{sim}=40)}$. The following process is proposed: (1) generate a

Table 1 Aggregated results for each method

δ_1	δ_2	δ_4	BS f_1 (%)	BS f_2 (%)	BS f_3 (%)	BSS f_1 (%)	BSS f_2 (%)	BSS f_3 (%)	BSSC f_1 (%)	BSSC f_2 (%)	BSSC f_3 (%)
5	40	30	0.30	0.07	0.27	0.21	0.05	0.42	0.21	0.06	0.36
5	40	40	0.66	0.05	0.34	0.31	0.04	0.40	0.05	0.09	0.27
5	50	30	0.37	0.11	0.19	0.24	0.01	0.37	0.31	0.13	0.23
5	50	40	0.24	0.09	0.34	0.28	0.05	0.43	0.07	0.15	0.13
10	40	30	0.34	0.07	0.45	0.10	0.05	0.36	0.12	0.05	0.20
10	40	40	0.85	0.10	0.67	0.21	0.06	0.32	0.52	0.11	0.44
10	50	30	0.62	0.08	0.37	0.20	0.11	0.43	0.30	0.10	0.62
10	50	40	0.75	0.11	0.47	0.35	0.04	0.44	0.17	0.09	0.18

random set of N_1 (parameter) neighbour solutions of the current solution \hat{S} ; (2) evaluate the best (according to f and using incremental computation) $N_2 \leq N_1$ solutions with $f^{(\text{sim}=1)}$; (3) evaluate the best [according to $f^{(\text{sim}=1)}$] $N_3 \leq N_2$ solutions with $f^{(\text{sim}=40)}$; (4) the selected neighbour solution \hat{S}' of \hat{S} is finally the best [according to $f^{(\text{sim}=40)}$] of these N_3 solutions. In other words, the best expected solutions are first partially (but quickly) simulated, and only the resulting most promising solutions are then accurately evaluated with a full simulation. After a preliminary set of experiments, parameters (N_1, N_2, N_3) were, respectively, tuned to (20, 10, 3). With such a parameter tuning, a single iteration requires roughly 10 min of computation.

In order to generate an initial solution, both DLS and TS use BSSC. However, anytime DLS is restarted, the new initial solution is provided by BS, otherwise BSSC (which requires about 30 min) will use too much of the time budget T allocated to DLS.

5. Results

Tests were performed on an Intel Quad-core i7 @ 3.4 GHz with 8 GB DDR3 of RAM memory. Section 5.1 describes the instances used to evaluate the quality of the different solution methods. Section 5.2 shows the experiments on the realistic instances (considering the imposed time limit of $T = 30$ min). Section 5.3 conducts a sensitivity analysis of the perturbation parameters. Finally, Section 5.4 presents the results obtained with DLS and TS when the time limit is extended to $T = 10$ h.

5.1. Generation of the instances

Based on the various data provided by SWB, we have developed an instance generator, which is able to quickly generate some realistic instances. An instance is composed of the following parameters: the number N of models, the number W of weeks in a year, the number J of shops, the number M of WS, the perturbation parameters ($\delta_1, \delta_2, \delta_3, \delta_4, \delta_5$ and ε), the production plan, the forecasted demand, the lead-times (from the factory to the WS, and from the WS to the shops), the shop priorities, the links WS–shops, the inventory levels at the WS

and at the shops. The data parameters were reasonably fixed after discussions with SWB. In the *reference* instance, we use $N = 143$, $W = 47$, $J = 192$, $M = 9$, $w^{\max} = 3$, $\delta_1 = 10$, $\delta_2 = 50$, $\delta_3 = 10$, $\delta_4 = 40$, $\delta_5 = 70$, $\varepsilon = 10$. We have generated a total of 64 instances around the *reference* instance by varying $\delta_1, \delta_2, \delta_3, \delta_4, \delta_5$ and ε .

5.2. Results on realistic instances

We propose below a summary of the results on the realistic instances, whereas the detailed results (and their analysis) can be found in the [Appendix](#).

Aggregated results are presented in Table 1, where for each triplet $(\delta_1, \delta_2, \delta_4)$, we have generated eight instances by considering different values for the parameters $(\delta_3, \delta_5, \varepsilon)$. More precisely, $\delta_3 \in \{5, 10\}$, $\delta_5 \in \{60, 70\}$ and $\varepsilon \in \{5, 10\}$. For each instance, let f^* be the best value ever found for a solution. For each method and each instance group (i.e. for each line of Table 1), average gaps with respect to the average value of f^* are provided. We decided to focus on the parameters $(\delta_1, \delta_2, \delta_4)$ because they are the roots of the perturbations. More precisely, δ_1 is the key perturbation parameter for the non-production perturbation, δ_2 for the slowed down perturbation, and δ_4 for the dispatching perturbation. We can roughly observe that both BSS and BSSC outperform BS on most of the instance groups, whereas BSSC gets the best results on average, which highlights that the proposed combination of a constructive heuristic with an exact model (based on an appropriate distance function) seems to be relevant for (P). Interestingly, when BSSC obtains the smallest gap on f_1 , it usually reaches the lowest gap on f_3 . On the opposite, when BSS gets the smallest gap on f_1 , it generally reaches the lowest gap on f_2 but not on f_3 . This tends to prove that BSSC is better at optimizing both f_1 and f_3 . Note that all the gaps for f_2 are relatively small (as also confirmed in the [Appendix](#)). Regarding the number of times that BSS or BSSC reach the lowest gaps, both methods are comparable. But when BSSC gets the best results on f_1 and f_3 , the corresponding gaps are significant when compared to the other two methods.

Table 2 summarizes (over all the 64 instances) the performance of the different methods on the three objectives. The

second row (labelled with “Number of zero gaps”) gives the number of times that the corresponding algorithm finds the best value (over all the tests with all the methods). The last row shows the average percentage gap of each method with respect to the average value of f^* . These results show that BSSC is the most efficient method on f_1 and f_3 , but it can be outperformed on f_2 , which is not surprising because of the lexicographic approach involving conflicting functions.

5.3. Sensitivity analysis

In this subsection, a sensitivity analysis of the most sensitive perturbation parameters is conducted. Figure 3 shows a wider variation of parameters δ_1 , δ_4 and δ_5 (respectively). For each part of the figure, the other involved parameters are set to the *reference* values. The sensitivity analysis of the other parameters (i.e. $\delta_2, \delta_3, \epsilon$) was not very conclusive and therefore not presented here.

Figure 3 (left part) shows that δ_1 has an impact on both f_1 and f_2 (as f_1 decreases by almost 1% if δ_1 decreases by 4%). Finding more reliable suppliers could lead to such a δ_1 reduction. Figure 3 (middle and right parts) shows that both δ_4 and δ_5 are the most sensitive parameters, as they significantly reduce f_1 and f_3 (even if in the middle part of the figure, it leads to a small augmentation of f_2). This is due to the WS, which are likely to sacrifice the considered shop and favour higher priority shops. More precisely, as the shops with the lowest priority are the most present, the watches will rather reach shops of priority 2 or 3 and therefore decrease f_1 and f_3 (as shortages are less penalized if more watches arrive, and shops of higher priorities are less penalized on inventories). Therefore, f_2 increases for the shops of lower priorities, which are more present. These conclusions, regarding the WS, were acknowledged by SWB. This analysis is in line with the recent

literature and trends, where in Yu *et al.* (2012), a consignment policy is proposed for inventory management in supply chains, based on strong interaction and reliable collaboration between the vendor and buyer levels.

To measure the impact of the perturbation parameters on a solution \hat{S} , we propose to use $\text{Dist}(\hat{S}, S) = \sum_{i,j} \lambda_t^{i,j}$, with $\lambda_t^{i,j}$ defined as in Section 4.1. Function Dist measures the structural difference between the expected solution \hat{S} and the actual solution S (after simulation). To better capture the impact of the perturbations, we propose to compute such a distance value for each priority level. More precisely, at the end of each single run of the simulation, the distance difference (in %) between \hat{S} and the returned actual solution is computed for each priority level (i.e. each gap is divided by $J^w \cdot W \cdot N$, where J^w is the number of shops with priority w). These gaps are then averaged over the sim runs. Such a test was performed 100 times on two different instances, and the results are averaged for each priority level. The used instances are $(\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \epsilon) \in \{(5, 40, 5, 30, 60, 5), (10, 50, 10, 40, 70, 10)\}$ (i.e. the instances with the smallest and largest perturbations, respectively). The resulting gaps are presented with the notation (w^1, w^2, w^3) , where w^i is the gap corresponding to the shops with priority i . The obtained gaps are (2.92, 3.11, 3.69%) for the least perturbed instance and (3.15, 3.50, 4.49%) for the *reference* instance. As an example, the shops with priority 3 have a gap of 4.49% on the *reference* instance. The results show that the larger gaps concern the most important shops, which is consistent as such shops are the least represented in the distribution network.

5.4. Results with $T = 10 h$

Three methods are compared in Table 3, namely BSSC, DLS, and TS. As shown in Table 1, aggregated but representative

Table 2 Summary of performances of the methods

	BS f_1	BS f_2	BS f_3	BSS f_1	BSS f_2	BSS f_3	BSSC f_1	BSSC f_2	BSSC f_3
Number of zero gaps	11	25	21	23	29	17	30	15	27
Averages	5.23	0.84	3.92	2.45	0.53	4.02	2.29	1.03	3.01

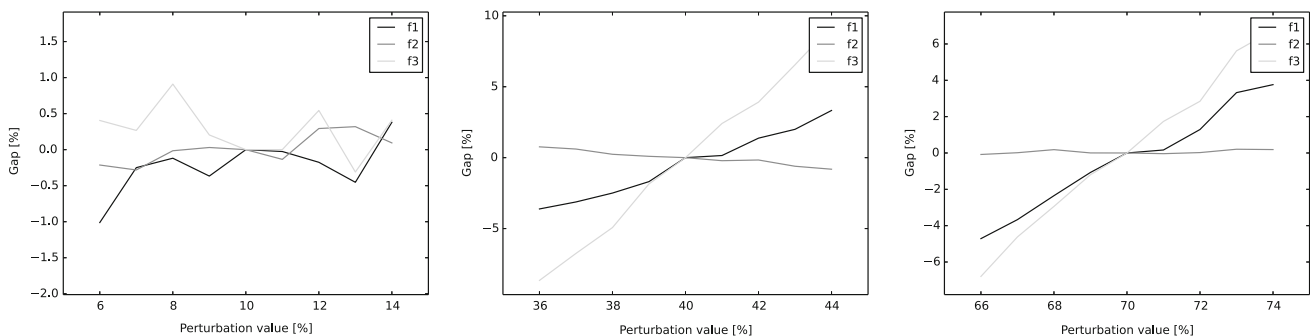


Figure 3 Sensitivity analysis on δ_1 (left) and δ_4 (middle) and δ_5 (right).

Table 3 Aggregated results for DLS and TS (improvements on BSSC)

Instance group			% Improvement with DLS			% Improvement with TS		
δ_1	δ_2	δ_4	f_1	f_2	f_3	f_1	f_2	f_3
5	40	30	0.99	0.39	1.55	1.75	0.86	2.80
5	40	40	1.02	0.72	1.75	3.19	0.45	2.46
5	50	30	1.71	0.54	1.75	2.28	0.71	1.52
5	50	40	1.33	0.34	1.48	3.07	0.99	1.76
10	40	30	0.87	0.47	1.90	1.71	0.61	2.85
10	40	40	1.99	0.55	1.89	2.15	0.99	3.31
10	50	30	1.52	0.43	0.96	2.91	0.85	2.31
10	50	40	1.65	0.59	1.33	1.86	0.89	0.95
Averages			1.38	0.50	1.58	2.36	0.79	2.25

results are showed (indeed, the detailed results for the 64 instances do not provide additional interesting information). For each instance group (i.e. each line of the table) and each objective function component f_i (for $i \in \{1, 2, 3\}$) of the overall objective function, the percentage improvement in DLS and TS with respect to BSSC is indicated (see the six last columns of the table). On the one hand, we can observe that unsurprisingly, both DLS and TS are able to improve the results provided by BSSC (the larger improvements are associated with f_1 and f_3). The use of an efficient local search framework (indeed, only the most promising moves are investigated) is therefore beneficial, as above 2% can be saved on some f_i 's. Interestingly, it is worth mentioning that both DLS and TS have improved each f_i value (associated with BSSC) for all the 64 instances. On the other hand, as TS outperforms DLS (it is the case for each instance group), it seems more appropriate to employ tabu status rather than restarts in order to better control and diversify the search process.

6. Conclusion and future works

In this paper, we investigate a relevant and complex problem (P) faced by a well-known Swiss watch brand. We propose powerful solution methods to tackle (P) and showed the superiority of the proposed metaheuristic compared with standard heuristics when considering the strict computation time limit imposed by the company. Moreover, we highlight the most sensitive parameters on which luxury companies should focus on. If larger computation time limits are allowed, it was showed that local search approaches (e.g. tabu search) can improve the results by above 2% on some objectives. Future works include developing other metaheuristics for the problem [e.g. among the different existing ant algorithms (Zufferey, 2012b)] or modelling a more complete problem that integrates more types of perturbations. One can also work on improving the speed of the proposed algorithms by jointly using lower bounds and solution space reduction techniques [as performed in Hertz *et al* (2005)]. Moreover, SWB could

evaluate and optimize an integrated supply chain where suppliers, production, and dispatching would be centralized in a common system. This would allow SWB to reduce the perturbation parameters by gaining control over the complete supply chain.

References

- Anupindi R and Bassok Y (1999). Supply contracts with quantity commitments and stochastic demand. *Pages 197–232 of: Quantitative Models for Supply Chain Management*. Springer: Berlin.
- Axsäter S (1993). Optimization of order-up-to-S policies in two-echelon inventory systems with periodic review. *Naval Research Logistics* **40**(2): 245–253.
- Axsäter S and Juntti L (1996). Comparison of echelon stock and installation stock policies for two-level inventory systems. *International Journal of Production Economics* **45**(1): 303–310.
- Banks J (1998). *Handbook of Simulation*. Wiley: New York.
- Bard JF and Golany B (1991). Determining the number of kanbans in a multiproduct, multistage production system. *International Journal of Production Research* **29**(5): 881–895.
- Ben-Daya M and Hariga M (2004). Integrated single vendor single buyer model with stochastic demand and variable lead time. *International Journal of Production Economics* **92**(1): 75–80.
- Boschetti MA, Maniezzo V, Roffilli M and Röhrler AB (2009). *Matheuristics: Optimization, simulation and control. Pages 171–177 of Hybrid Metaheuristics*. Springer: New York.
- Brandstädt A and Spinrad JP (1999). *Graph Classes: A Survey*. Vol. 3. SIAM.
- Brun A, Caniato F, Caridi M, Castelli C, Miragliotta G, Ronchi S, Sianesi A and Spina G (2008). Logistics and supply chain management in luxury fashion retail: Empirical investigation of Italian firms. *International Journal of Production Economics*, **114**(2): 554–570.
- Caniato F, Caridi M, Castelli C and Golini R (2011). Supply chain management in the luxury industry: a first classification of companies and their strategies. *International Journal of Production Economics*, **133**(2): 622–633.
- Catry B (2003). The great pretenders: the magic of luxury goods. *Business Strategy Review* **14**(3): 10–17.
- Clark AJ and Scarf H (1960). Optimal policies for a multi-echelon inventory problem. *Management Science* **6**(4): 475–490.
- Daniel J and Rajendran C (2005). A simulation-based genetic algorithm for inventory optimization in a serial supply chain.

- International Transactions in Operational Research* **12**(1): 101–127.
- Forsberg R (1995). Optimization of order-up-to-S policies for two-level inventory systems with compound Poisson demand. *European Journal of Operational Research* **81**(1): 143–153.
- FSWIFH (2014). Swiss watch export: World. *Federation of the Swiss Watch Industry FH 2014 Report*.
- Gendreau M and Potvin J-Y (2010). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer: New York.
- Glasmeier A (1991). Technological discontinuities and flexible production networks: The case of Switzerland and the world watch industry. *Research Policy* **20**(5): 469–485.
- Hertz A, Schindl D and Zufferey N (2005). Lower bounding and tabu search procedures for the frequency assignment problem with polarization constraints. *4OR* **3**(2): 139–161.
- Hnaïen F, Delorme X and Dolgui A (2010). Multi-objective optimization for inventory control in two-level assembly systems under uncertainty of lead times. *Computers & Operations Research* **37**(11): 1835–1843.
- Kara S, Rugrungruang F and Kaebnick H (2007). Simulation modelling of reverse logistics networks. *International Journal of Production Economics* **106**(1): 61–69.
- Kunnunkala S and Topaloglu H (2011). Linear programming based decomposition methods for inventory distribution systems. *European Journal of Operational Research* **211**(2): 282–297.
- Law AM and Kelton WD (1991). *Simulation Modeling and Analysis*. Vol. 2. McGraw-Hill: New York.
- Lee HL and Moïnzadeh K (1987). Two-parameter approximations for multi-echelon repairable inventory models with batch ordering policy. *IIE Transactions* **19**(2): 140–149.
- Madadi A, Kurz ME and Ashayeri J (2010). Multi-level inventory management decisions with transportation cost consideration. *Transportation Research Part E: Logistics and Transportation Review* **46**(5): 719–734.
- Malakooti B (2012). Decision making process: Typology, intelligence, and optimization. *Journal of Intelligent Manufacturing* **23**(3): 733–746.
- Marti R, Gallego M, Duarte A and Pardo EG (2013). Heuristics and metaheuristics for the maximum diversity problem. *Journal of Heuristics* **19**(4): 591–615.
- Melo MT, Nickel S and Saldanha-da Gama F (2009). Facility location and supply chain management—A review. *European Journal of Operational Research* **196**(2): 401–412.
- Miranda PA and Garrido RA (2004). Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. *Transportation Research Part E: Logistics and Transportation Review* **40**(3): 183–207.
- Moïnzadeh K and Lee HL (1986) Batch size and stocking levels in multi-echelon repairable systems. *Management Science* **32**(12): 1567–1581.
- Otamendi FJ and Doncel LM (2012). Towards an auction-driven gas supply: A simulation-based optimization framework for utilities. *Journal of the Operational Research Society* **63**(9): 1189–1198.
- Pardalos PM and Xue J (1994). The maximum clique problem. *Journal of Global Optimization* **4**(3): 301–328.
- Respen J, Zufferey N and Amaldi E (2016). Metaheuristics for a job scheduling problem with smoothing costs relevant for the car industry. *Networks* **67**(3): 246–261.
- Schaeffer SE (2007). Graph clustering. *Computer Science Review* **1**(1): 27–64.
- Schildbach G and Morari M (2016). Scenario-based model predictive control for multi-echelon supply chain management. *European Journal of Operational Research* **252**(2): 540–549.
- Schmitt AJ, Snyder LV and Shen Z-JM (2010). Inventory systems with stochastic demand and supply: Properties and approximations. *European Journal of Operational Research* **206**(2): 313–328.
- Segerstedt A (1991). *Cover-Time Planning: An Alternative to MRP*. Ph.D. thesis, Linköping University.
- Segerstedt A (1996). A capacity-constrained multi-level inventory and production control problem. *International Journal of Production Economics* **45**(1): 449–461.
- Shafieezadeh M and Sadegheih A (2014). Developing an integrated inventory management model for multi-item multi-echelon supply chain. *International Journal of Advanced Manufacturing Technology* **72**(5): 1099–1119.
- Silver EA and Zufferey N (2005). Inventory control of raw materials under stochastic and seasonal lead times. *International Journal of Production Research* **43**(24): 5161–5179.
- Silver EA and Zufferey N (2011). Inventory control of an item with a probabilistic replenishment lead time and a known supplier shutdown period. *International Journal of Production Research* **49**(4): 923–947.
- Svoronos A and Zipkin P (1988). Estimating the performance of multi-level inventory systems. *Operations Research* **36**(1): 57–72.
- SWB (2014). Private communication. Switzerland.
- Syntetos AA, Georgantzias NC, Boylan JE and Dangerfield BC (2011). Judgement and supply chain dynamics. *Journal of the Operational Research Society* **62**(6): 1138–1158.
- Terzi S and Cavalieri S (2004). Simulation in the supply chain context: A survey. *Computers in Industry* **53**(1): 3–16.
- Tsai SC and Zheng Y-X (2013). A simulation optimization approach for a two-echelon inventory system with service level constraints. *European Journal of Operational Research* **229**(2): 364–374.
- Yao D-Q, Yue X, Mukhopadhyay SK and Wang Z (2009). Strategic inventory deployment for retail and e-tail stores. *Omega* **37**(3): 646–658.
- Yu J, Sarker BR, Duan Q and Wu B (2012). Single-manufacturer, multi-retailer consignment policy for retailers' generalized demand distributions. *Journal of the Operational Research Society* **63**(12): 1708–1719.
- Zufferey, N. 2012a. Metaheuristics: Some principles for an efficient design. *Computer Technology and Applications* **3**(6): 446–462.
- Zufferey N (2012b). Optimization by ant algorithms: Possible roles for an individual ant. *Optimization Letters*, **6**(5): 963–973.

Appendix: Results on the realistic instances

Table A1 shows the results for the three solution methods when δ_1 , δ_2 , δ_3 , δ_4 , δ_5 and ε vary. We have set $\sigma = 2$ (which is the standard deviation of the demand perturbation), as larger values are totally not realistic for SWB. Two different values are presented for each parameter, the first is lower than the reference value, and the second is the reference value. The goal is to compare the performances of the algorithms and the impact of lower perturbations on the objectives. The first six columns give the values of each parameter. The seventh column, called f_1^* , is the best f_1 value returned by one of the algorithms on the considered instance. The same applies for the next two columns regarding f_2 and f_3 . Starting at column ten, the next three columns show the gaps between the f_i^* 's

Table A1 Results (percentage gaps) on realistic instances

δ_1	δ_2	δ_3	δ_4	δ_5	ϵ	f_1^*	f_2^*	f_3^*	B Sf_1	B Sf_2	B Sf_3	B SSf_1	B SSf_2	B SSf_3	B $SSCf_1$	B $SSCf_2$	B $SSCf_3$
5	40	5	30	60	5	93856809.67	103.84	6426820.70	0.42%	0.12%	0.04%	0.00%	0.00%	0.53%	0.11%	0.04%	0.00%
5	40	5	30	60	10	93611706.42	104.04	6393880.08	0.00%	0.13%	0.00%	0.66%	0.00%	0.33%	0.24%	0.00%	0.70%
5	40	5	30	70	5	104748487.28	103.00	8210743.70	0.00%	0.00%	0.18%	0.08%	0.05%	0.07%	0.11%	0.02%	0.00%
5	40	5	30	70	10	104160132.28	103.05	8188112.60	0.26%	0.00%	0.00%	0.00%	0.04%	0.53%	0.66%	0.15%	0.88%
5	40	5	40	60	5	104584290.88	100.48	9127778.05	1.29%	0.14%	0.00%	0.25%	0.00%	0.20%	0.00%	0.11%	0.29%
5	40	5	40	60	10	104521898.40	100.56	9044377.57	1.00%	0.00%	1.10%	0.49%	0.02%	0.00%	0.00%	0.03%	0.74%
5	40	5	40	70	5	118030025.88	100.49	11122331.10	0.47%	0.09%	0.07%	0.03%	0.04%	0.39%	0.00%	0.00%	0.00%
5	40	5	40	70	10	117299231.35	100.42	11082133.78	1.24%	0.00%	1.12%	0.82%	0.06%	1.03%	0.00%	0.31%	0.00%
5	40	10	30	60	5	94229919.08	103.87	6400873.60	0.12%	0.18%	1.46%	0.00%	0.00%	0.92%	0.20%	0.15%	0.00%
5	40	10	30	60	10	93813698.90	103.90	6434238.35	0.44%	0.10%	0.00%	0.00%	0.00%	0.28%	0.32%	0.02%	0.84%
5	40	10	30	70	5	104678949.62	102.99	8241697.45	0.33%	0.00%	0.00%	0.32%	0.07%	0.46%	0.00%	0.05%	0.49%
5	40	10	30	70	10	104376624.28	102.87	8229631.92	0.82%	0.00%	0.48%	0.61%	0.24%	0.26%	0.00%	0.06%	0.00%
5	40	10	40	60	5	104522811.25	100.53	9098167.30	0.58%	0.00%	0.46%	0.42%	0.11%	0.29%	0.00%	0.14%	0.00%
5	40	10	40	60	10	104718207.03	100.62	9051360.93	0.32%	0.00%	0.00%	0.00%	0.06%	0.72%	0.10%	0.02%	0.90%
5	40	10	40	70	5	117686732.45	100.35	11145245.72	0.36%	0.17%	0.00%	0.21%	0.00%	0.38%	0.00%	0.13%	0.00%
5	40	10	40	70	10	117642362.15	100.46	11148556.55	0.00%	0.00%	0.00%	0.27%	0.02%	0.16%	0.29%	0.01%	0.21%
5	50	5	30	60	5	93974135.67	103.95	6444403.80	0.03%	0.07%	0.21%	0.00%	0.00%	0.05%	0.17%	0.04%	0.00%
5	50	5	30	60	10	94002498.80	103.86	6443839.08	0.35%	0.14%	0.00%	0.00%	0.00%	0.38%	0.45%	0.11%	0.57%
5	50	5	30	70	5	104492704.28	103.09	8241887.65	0.00%	0.08%	0.49%	0.25%	0.03%	0.00%	0.32%	0.00%	0.09%
5	50	5	30	70	10	103862333.53	102.99	8162655.60	0.00%	0.24%	0.00%	0.53%	0.00%	1.16%	1.05%	0.24%	1.01%
5	50	5	40	60	5	105574759.25	100.44	9125329.62	0.10%	0.00%	0.68%	0.00%	0.10%	0.49%	0.08%	0.18%	0.00%
5	50	5	40	60	10	104700360.65	100.56	9070600.65	0.04%	0.00%	0.00%	0.79%	0.05%	0.55%	0.00%	0.09%	0.35%
5	50	5	40	70	5	117369288.78	100.43	11145628.20	0.58%	0.07%	0.00%	0.40%	0.00%	0.24%	0.00%	0.02%	0.45%
5	50	5	40	70	10	116791994.05	100.37	11100587.30	0.51%	0.24%	0.63%	0.71%	0.00%	0.44%	0.00%	0.24%	0.00%
5	50	10	30	60	5	94075385.40	103.88	6457223.20	0.68%	0.11%	0.00%	0.41%	0.00%	0.90%	0.00%	0.27%	0.13%
5	50	10	30	60	10	93606039.33	103.84	6455417.00	0.58%	0.14%	0.02%	0.00%	0.00%	0.04%	0.52%	0.11%	0.00%
5	50	10	30	70	5	104844657.33	102.96	8250414.83	0.14%	0.07%	0.32%	0.21%	0.00%	0.24%	0.00%	0.17%	0.00%
5	50	10	30	70	10	104136960.10	102.93	8248901.70	1.19%	0.00%	0.50%	0.50%	0.08%	0.21%	0.00%	0.13%	0.00%
5	50	10	40	60	5	104825043.78	100.46	9103612.25	0.39%	0.00%	0.63%	0.30%	0.13%	0.16%	0.07%	0.19%	0.00%
5	50	10	40	60	10	10559244.03	100.42	9136612.82	0.06%	0.18%	0.09%	0.30%	0.00%	0.48%	0.00%	0.16%	0.00%
5	50	10	40	70	5	118241676.47	100.28	11146366.88	0.00%	0.26%	0.65%	0.01%	0.00%	0.88%	0.15%	0.18%	0.00%
5	50	10	40	70	10	117588781.15	100.39	11113507.18	0.25%	0.00%	0.00%	0.00%	0.13%	0.18%	0.22%	0.11%	0.21%
10	40	5	30	60	5	94207031.50	104.34	6529899.92	0.47%	0.00%	1.08%	0.00%	0.00%	0.00%	0.37%	0.03%	0.25%
10	40	5	30	60	10	94340721.88	104.24	6557394.03	0.31%	0.11%	0.31%	0.00%	0.00%	0.00%	0.15%	0.21%	0.29%
10	40	5	30	70	5	105345990.90	103.23	8249574.90	0.10%	0.00%	0.99%	0.05%	0.13%	1.41%	0.00%	0.11%	0.00%
10	40	5	30	70	10	104650005.45	103.35	8269697.12	0.53%	0.08%	0.00%	0.50%	0.13%	0.32%	0.00%	0.00%	0.16%
10	40	5	40	60	5	105322713.28	100.74	9104621.30	0.00%	0.11%	0.00%	0.23%	0.27%	0.24%	0.79%	0.00%	0.23%
10	40	5	40	60	10	105729258.72	100.90	9052801.07	0.00%	0.00%	0.00%	0.05%	0.00%	1.59%	0.24%	0.02%	0.81%
10	40	5	40	70	5	117661305.95	100.75	11080102.28	0.49%	0.15%	0.72%	0.00%	0.00%	0.00%	0.62%	0.02%	0.21%
10	40	5	40	70	10	117698666.75	100.61	11088766.53	0.97%	0.33%	0.55%	0.72%	0.00%	0.05%	0.00%	0.28%	0.00%
10	40	10	30	60	5	94554839.78	104.21	6566525.05	0.00%	0.09%	0.16%	0.06%	0.09%	0.74%	0.04%	0.00%	0.00%
10	40	10	30	60	10	94712670.17	104.24	6595042.62	0.02%	0.00%	0.39%	0.14%	0.00%	0.44%	0.00%	0.03%	0.00%
10	40	10	30	70	5	104941038.42	103.18	8315693.42	1.02%	0.23%	0.00%	0.14%	0.00%	0.00%	0.00%	0.00%	0.73%
10	40	10	30	70	10	104981703.83	103.33	8312080.92	0.28%	0.03%	0.45%	0.00%	0.06%	0.00%	0.39%	0.00%	0.18%
10	40	10	40	60	5	104064078.90	100.93	9057680.45	1.37%	0.01%	0.98%	0.00%	0.00%	0.00%	1.43%	0.00%	0.55%

Table A1 continued

δ_1	δ_2	δ_3	δ_4	δ_5	ε	f_1^*	f_2^*	f_3^*	BSf_1	BSf_2	BSf_3	$BSSf_1$	$BSSf_2$	$BSSf_3$	$BSSCf_1$	$BSSCf_2$	$BSSCf_3$
10	40	10	40	60	10	104990602.78	100.83	9092775.85	1.30%	0.01%	0.84%	0.00%	0.05%	0.00%	0.68%	0.00%	1.06%
10	40	10	40	70	5	117748578.03	100.65	11059492.68	1.26%	0.20%	0.91%	0.70%	0.00%	0.71%	0.00%	0.18%	0.00%
10	40	10	40	70	10	117612171.58	100.50	11066311.35	1.38%	0.00%	1.36%	0.00%	0.16%	0.00%	0.39%	0.34%	0.65%
10	50	5	30	60	5	94552063.95	104.09	6500546.62	0.09%	0.17%	0.73%	0.07%	0.36%	0.00%	0.00%	0.00%	1.51%
10	50	5	30	60	10	94624449.00	104.19	6559273.08	0.00%	0.00%	0.00%	0.06%	0.07%	0.26%	0.13%	0.10%	0.23%
10	50	5	30	70	5	104841454.97	103.34	8332678.75	1.23%	0.00%	0.36%	0.00%	0.05%	0.00%	0.68%	0.03%	0.42%
10	50	5	30	70	10	104644649.20	103.26	8311685.03	0.92%	0.09%	0.12%	0.00%	0.12%	0.00%	0.35%	0.00%	0.53%
10	50	5	40	60	5	105256370.90	100.74	9124679.62	0.70%	0.22%	0.60%	0.64%	0.00%	0.47%	0.00%	0.28%	0.00%
10	50	5	40	60	10	104892459.38	100.72	9148605.15	1.41%	0.08%	0.00%	0.59%	0.03%	0.34%	0.00%	0.00%	0.25%
10	50	5	40	70	5	118134879.38	100.80	11101943.82	0.25%	0.14%	0.70%	0.21%	0.00%	0.91%	0.00%	0.03%	0.00%
10	50	5	40	70	10	118136339.90	100.75	11154865.12	0.89%	0.07%	0.07%	0.31%	0.00%	0.00%	0.00%	0.01%	0.67%
10	50	10	30	60	5	94337018.88	104.08	6558153.50	1.37%	0.00%	0.56%	0.65%	0.01%	1.04%	0.00%	0.44%	0.00%
10	50	10	30	60	10	94388692.80	104.12	6560253.45	0.66%	0.00%	0.36%	0.28%	0.08%	1.16%	0.00%	0.19%	0.00%
10	50	10	30	70	5	105347982.90	103.16	8353232.15	0.00%	0.17%	0.00%	0.56%	0.00%	1.01%	0.63%	0.07%	0.70%
10	50	10	30	70	10	104673098.12	103.26	8288574.45	0.72%	0.20%	0.84%	0.00%	0.18%	0.00%	0.64%	0.00%	1.53%
10	50	10	40	60	5	105867424.58	100.85	9153754.32	0.34%	0.00%	0.94%	0.47%	0.05%	0.34%	0.00%	0.22%	0.00%
10	50	10	40	60	10	105854191.80	100.83	9166580.05	0.27%	0.12%	0.00%	0.61%	0.03%	1.45%	0.00%	0.00%	0.03%
10	50	10	40	70	5	117234709.35	100.67	11094372.15	1.45%	0.21%	0.50%	0.00%	0.00%	0.00%	0.84%	0.17%	0.39%
10	50	10	40	70	10	117928718.10	100.79	11107442.60	0.66%	0.00%	0.96%	0.00%	0.17%	0.00%	0.54%	0.01%	0.12%
Number of zero gaps						11			11	25	21	23	29	17	30	15	27
Averages						5.23			5.23	0.84	3.92	2.45	0.53	4.02	2.29	1.03	3.01

and the f_i 's for method BS. The same information is then given for BSS and BSSC. The last two rows show the number of times that the corresponding algorithm found a gap of zero, followed by the average values of the different gaps. Note that the first row is the instance with the lowest perturbations, and the last row shows the instance with the largest perturbations (which is the *reference* instance).

The results shows that BSSC outperforms BSS on both f_1 and f_3 , whereas BS is significantly overpowered by both BSS and BSSC. Regarding f_2 , BSS finds zero gaps 29 times, versus 15 times for BSSC, but the gaps for BSSC are so small when BSS finds zero gaps that no conclusion can come out. Moreover, as f_1 has the priority over f_2 , it is not surprising that if BSSC outperforms BSS on f_1 , then BSS can sometimes outperform BSSC on f_2 . When comparing the last row, the averages over the complete set of instances clearly show that BSSC is the most efficient method, as it shows the lower percentages for f_1 and f_3 , and very low gaps for f_2 .

Regarding the impact of a single parameter variation (for example $\delta_1 \in \{5, 10\}$), and leaving the other parameters unchanged, we can see that the impacts on f_1 and f_3 are significant, but not on f_2 . The reasons to this are detailed in Section 5.3. The parameters that induce the greater reductions on f_1 and f_3 are obviously δ_4 and δ_5 . Both parameters are used when encountering the perturbations involved by the WS' behaviours. The improvements are such that it would be very important for SWB to reduce the perturbations by either finding more reliable suppliers (for $\delta_1, \delta_2, \delta_3$) or by having a better control on the WS (for δ_4, δ_5). To properly assess the impact of such reductions, Section 5.3 proposes a more detailed sensitivity analysis.

*Received 27 July 2015;
accepted 4 October 2016*