On the optimal control of manufacturing and remanufacturing activities with a single shared server

Simme Douwe Flapper, Jean-Philippe Gayon, Lâm Laurent Lim

On the optimal control of manufacturing and remanufacturing activities with a single shared server

Simme Douwe Flapper, Jean-Philippe Gayon, Lâm Laurent Lim

October 21, 2013

Abstract

We consider a single-product make-to-stock manufacturing-remanufacturing system. Returned products require remanufacturing before they can be sold. The manufacturing and remanufacturing operations are executed by the same single server, where switching from one activity to another does not involve time or cost and can be done at an arbitrary moment in time. Customer demand can be fulfilled by either newly manufactured or remanufactured products. The times for manufacturing and remanufacturing a product are exponentially distributed. Demand and used products arrive via mutually independent Poisson processes. Disposal of products is not allowed and all used products that are returned have to be accepted. Using Markov decision processes, we investigate the optimal manufacture-remanufacture policy that minimizes holding, backorder, manufacturing and remanufacturing costs per unit of time over an infinite horizon. For a subset of system parameter values we are able to completely characterize the optimal continuous-review dynamic preemptive policy. We provide an efficient algorithm based on quasi-birth-death processes to compute the optimal policy parameter values. For other sets of system parameter values, we present some structural properties and insights related to the optimal policy and the performance of some simple threshold policies.

Keywords

Manufacturing, remanufacturing, shared server, inventory control, stochastic dynamic programming

1 Introduction

The reuse of products and materials is not a recent discovery. However, it is only during recent decades that product recovery and reverse logistics have gained increasing importance as a profitable and sustainable strategy for companies around the world (see e.g. Dekker et al., 2004; Srivastava, 2007). Reusing may be motivated by economical, legislative or environmental reasons. As stated in Thierry et al. (1995), there are several options to recover a product. Via remanufacturing, a product is completely recovered and its quality after remanufacturing is as good as that of a new product. Remanufacturing complicates inventory control. Integrating this reverse flow of used products affects both the materials planning and the inventory control of the supply chain. Indeed, managers have to take into account the

uncertain flow of used products and they have to coordinate the remanufacturing stage with the regular mode of procurement (Inderfurth and van der Laan, 2001).

Coordination between manufacturing and remanufacturing operations is a central issue (Van der Laan et al., 1999). The company has to decide who takes care of each operation, when and how much to manufacture or to remanufacture. In our model, we consider a single resource that can perform manufacturing and remanufacturing operations. The server can switch at any time between these two operations. As stated in Ferrer and Whybark (2000), shared resources increase the flexibility of the system but also the complexity of the coordination. The study of Tang and Teunter (2006) is an example of a concrete application of a hybrid manufacturing and remanufacturing system with a shared resource. Their research was motivated by a company which manufactures and remanufactures car parts. New and remanufactured parts are processed in the same facilities by the same workers. In their study, the remanufactured products represent approximately 30% of the annual sales. Trebilcock (2002) discusses a catalog retailer that has to prepare and ship around 65,000 products weekly. Around 15% of the products come back to the retailer. In this case, managers can ask some workers for a few days to inspect, clean and repack returned products. This type of situation (where workers switch between manufacturing and remanufacturing to satisfy customer orders) may be modeled by a hybrid system as described in our study.

In this paper, we consider a hybrid manufacturing-remanufacturing system. Manufacturing and remanufacturing operations are executed by a single shared ressource. This situation can be found in small companies or in dedicated specialized units within companies dealing with complex products (like large medical systems and dedicated copiers), when both manufacturing and remanufacturing require deep understanding of the product. Remanufactured and new products can equally well be used to satisfy customer demand. The time to manufacturing and remanufacturing a product is exponentially distributed. Demand and used products arrive via mutually independent Poisson processes. We consider the situation where products are manufactured or remanufactured one by one. The objective is to minimize discounted or average costs (holding, manufacturing, remanufacturing, backorder) over an infinite horizon. We do not include neither setup times and costs nor disposal option. These limits of our model our discussed in Section 8.

More generally, our problem can be seen as a production-inventory control with two supply channels. The first supply channel is completely controlled while the second supply channel is autonomous and random. For example, car companies like Chrysler can get very easily low quality (often less expensive) engines that are supplied by dealers in the context of take back programs or internet offers. However, it is much more difficult and uncertain to get high quality (often more expensive) engines.

To the best of our knowledge, the problem described in this paper has not been studied before. In particular it is the first paper to investigate an hybrid system with a shared resource in a stochastic environment. For the problem described above, we characterize the optimal manufacture-remanufacture policy for certain sets of system parameter values and provide for these sets an efficient method to calculate the optimal values of the policy parameters. For other sets of system parameter values, we provide insights into the optimal policy structure. We also present a sensitivity analysis for several system parameters.

In Section 2, we review the literature and our contributions. In Section 3, a detailed description and mathematical formulation of the problem is given. In Section 4, we derive several characteristics of the optimal policy structure for different sets of parameter values. In Section 5, we provide an efficient

method to calculate the optimal values of the policy parameters. In Section 6, we present some insights into the optimal behavior of the system for sets of parameter values for which it is not possible to derive the optimal policy structure. Based on these insights, we introduce in Section 7 some simple heuristics that we compare with the optimal policy. In Section 8 we conclude the paper and indicate directions for further research.

2 Literature review

There is a vast literature on inventory control with product returns (see e.g. Fleischmann et al., 1997, 2002; Pokharel and Mutha, 2009; Ilgin and Gupta, 2010). Hybrid manufacturing and remanufacturing systems have received growing attention in recent years (see e.g. Van der Laan et al., 1999; Teunter et al., 2004).

In a deterministic environment, only a few authors consider a shared capacity for manufacturing and remanufacturing. The first deterministic model involving product returns is introduced by Schrady (1967). He considers an EOQ setting with constant demand and returns rates, infinite manufacturing and remanufacturing rates, different holding costs for recoverable and serviceable items and different fixed costs. For the class of $(1, R)$ policies that alternates a manufacturing lot and a fixed number R of remanufacturing lots, he is able to derive EOQ formulae. Several papers consider variants of Schrady's model. Nahmias and Rivera (1979) considers a variant with a finite recovery rate. Teunter (2001) includes a disposal option for returned products. Teunter (2004) investigates $(1, R)$ policies and $(P, 1)$ policies where the server alternates between manufacturing P lots and remanufacturing 1 lot. He obtains optimal lot-sizes for both manufacturing and recovery operations. These formulae are valid for both finite and infinite manufacturing and remanufacturing rates. Tang and Teunter (2006) study the multi-product economic lot scheduling problem with returns. Like previous papers, manufacturing and remanufacturing operations are performed on the same shared production line. Due to the complexity of this multi-product problem, the authors restrict their study to common cycling policies with one manufacturing lot and one remanufacturing lot in each cycle. The problem is formulated as a mixed-integer problem.

In a periodic review setting, several authors present models with stochastic demand and returns but they do not consider a shared capacity for manufacturing and remanufacturing. Simpson (1978) considers a system where returns are held in a separate buffer until they are remanufactured or disposed of. He characterizes the optimal policy for the case with zero manufacturing and remanufacturing lead times. Inderfurth (1997) consider positive and identical manufacturing and remanufacturing leadtimes. He also shows that the optimal policy may have a very complex structure when lead times are different. Li et al. (2010) include fixed manufacturing costs and fixed disposal costs. Zhou et al. (2011) investigates a situation where returns can have different quality levels. DeCroix (2006) extends the results of Inderfurth (1997) to a multi-stage series system where products are remanufactured at the upstream stage. Fleischmann and Kuik (2003) study the optimal policy structure for a single stock point with a stochastic demand that is either positive or negative in each period. They show the average-cost optimality of an (s, S) policy. Teunter and Vlachos (2002) study the necessity of a disposal option for a hybrid manufacturing and remanufacturing system with constant lead times over a finite time horizon. By using simulation, their results show that a disposal option leads to an important cost reduction only if the demand rate is low, the recovery rate is high and remanufacturing is almost as expensive

as manufacturing. The authors also notice that it is more difficult to find the optimal policy when considering a disposal option for product returns.

As far as we know, Heyman (1977) is the first who investigates a continuous-time inventory control model with product returns. He assumes zero lead times, linear costs and a disposal option. Heyman (1977) shows that the optimal disposal policy is a threshold policy and derives an explicit formula for this threshold. The author also indicates that introducing a remanufacturing lead time requires a new variable (the remanufacturable inventory) resulting in a very complex model. Gayon (2006) considers a variant of Heyman (1977) where the manufacturing leadtime is exponentially distributed and there is no disposal option. He proves the optimality of a base-stock policy to control the serviceable inventory and derives an exact formula for the optimal base-stock level. Muckstadt and Isaac (1981) introduce a hybrid system with non-zero lead times for both procurement and remanufacturing. The authors investigate a simple (s, Q)-rule for the procurement policy and returns are remanufactured as long as returned products are available. Since an exact analysis is difficult, an approximation is given for the steady-states distribution of the system. Fleischmann et al. (2002) consider an inventory system with Poisson demand and returns, fixed procurement leadtime and zero remanufacturing processing time. They show that the optimal policy to minimize the total average cost is an (s, Q)-policy. Van der Laan et al. (1999) investigate an inventory system where manufacturing and remanufacturing occur simultaneously. They consider constant leadtimes, stochastic demands and returns, fixed set-up costs per batch and linear holding costs for the remanufacturable and serviceable inventories. They compare systems without remanufacturing with push and pull controlled hybrid systems with remanufacturing. Ching et al. (2007) suggest a new approach to analyze a hybrid system with remanufacturing under continuous-review with Markovian assumptions. The serviceable inventory is controlled by an (s, S)-policy and they use quasi-birth-death processes to model the remanufacturable and serviceable inventories. The authors provide a quasi-optimization method to compute the optimal replenishment level. Aras et al. (2006) propose a new model for hybrid systems with a constant manufacturing lead time and a remanufacturing process that depends on the quality of returns. In this model, the remanufacturing lead time and cost are uniformly distributed between minimum and maximum values. Two policies are compared: priority-to-manufacturing (remanufacturing station is only used when the manufacturing facility is not sufficient to satisfy all demands) and priority-to-remanufacturing (replenishment orders are placed first with the remanufacturing facility). The authors show the benefits of coordinating manufacturing and remanufacturing.

To summarize, we note that there is a lot of literature on hybrid manufacturing-remanufacturing systems but only few papers that deal with a shared capacity for manufacturing and remanufacturing. All of these papers deal with deterministic demand and returns. To the best of our knowledge, the research presented in this paper is the first one that studies a stochastic model with shared capacity for manufacturing and remanufacturing. Our model is close to the one of Teunter (2004). Like in his research, we consider a shared resource for both manufacturing and remanufacturing with finite manufacturing and remanufacturing capacities. While Teunter (2004) considers a deterministic setting, we assume random (exponential) processing times for manufacturing and remanufacturing and random (Poisson) processes for demand and returns. Furthermore, contrary to Teunter (2004) who investigates the optimal lot-sizes for $(P, 1)$ and $(1, R)$-policies, we do not restrict our research to certain classes of policies but we investigate the optimal policy to control the system. However, unlike Teunter, we do not include setup times and costs in our problem.

3 Notations and assumptions

We consider a single-product system with product returns (Figure 1). Returned products require remanufacturing operations (always successful) before they can be sold. Manufacturing and remanufacturing operations are executed by a single shared resource. The single server can perform only one operation at the same time: remanufacturing or manufacturing, or be idle. New products are manufactured one by one, and returns are remanufactured one by one too. Switching times and costs are neglected. Raw materials for manufacturing are assumed to be always available. We consider a situation with preemption: the server can stop a job at any time and she or he can resume the job immediately or after a while.

All returns have to be accepted. Each return can be remanufactured into a serviceable product and disposal is not allowed. Customer demands can be fulfilled by either newly manufactured or remanufactured products. All demand has to be fulfilled. If a customer order is not immediately satisfied, it is backlogged. Two inventories are distinguished: the remanufacturable inventory of product returns, and the serviceable inventory of new and remanufactured products. There are no capacity limitations for the inventories.

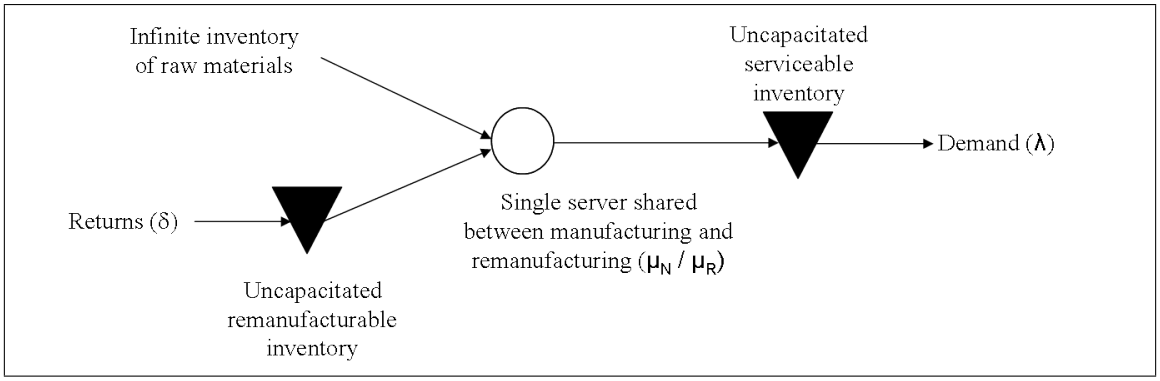


Figure 1: Schematic representation of a hybrid manufacturing-remanufacturing system with one shared server

We use a continuous-time infinite horizon model with Markovian assumptions: demands and returns follow independent Poisson processes while manufacturing and remanufacturing times are exponentially distributed. We further assume linear holding and backlogging costs.

We use the following notation:

- b: backlog cost (per product per unit of time).
- c_N: manufacturing cost (per product).
- c_R: remanufacturing cost (per product).
- h_R: remanufacturable holding cost (per product per unit of time).
- h_S: serviceable holding cost (per product per unit of time).
- $X(t)$: remanufacturable on-hand inventory level at time t.
- $\overline{X}$: average remanufacturable on-hand inventory level.

- $Y(t)$: serviceable net inventory level at time t.
- $\overline{Y}$: average serviceable net inventory level.
- $\overline{Y^+}$: average serviceable on-hand inventory level.
- $\overline{Y^-}$: average number of backlogs.
- α: discount factor.
- δ: average return rate (returned products per unit of time).
- λ: average demand rate (products per unit of time).
- μ_N: average manufacturing rate (products per unit of time).
- μ_R: average remanufacturing rate (products per unit of time).

Moreover, we introduce the following additional notation:

- $\rho_N = \frac{\lambda-\delta}{\mu_N}$: Manufacturing utilization (percentage of time spent on average by the server on manufacturing).
- $\rho_R = \frac{\delta}{\mu_R}$: Remanufacturing utilization (percentage of time spent on average by the server on remanufacturing).
- $\rho_{server} = \frac{\delta}{\mu_R} + \frac{\lambda-\delta}{\mu_N} = \rho_R + \rho_N$: Server utilization (percentage of time that the server is busy).
- $\tau = \frac{\delta}{\lambda}$: Percentage of demand satisfied by remanufactured returns.

To ensure the stability of the system, we have to assume that

$$\tau < 1 \tag{1}$$

and

$$\rho_N + \rho_R = \rho_{server} < 1\,. \tag{2}$$

Equation (1) ensures that the serviceable inventory will not increase to infinity and (2) ensures that all demand can be satisfied by the single server.

To control the system, we have to decide when the server should manufacture new products, when the server should remanufacture returns and when the server should be idle. Let $AC(\pi)$ be the undiscounted average costs per unit of time under policy π. Our objective is to find the optimal policy that minimizes the average cost. We have

$$AC(\pi) = c_N(\lambda - \delta) + c_R\delta + h_R\overline{X} + h_S\overline{Y^+} + b\overline{Y^-}\,. \tag{3}$$

Observe that quantities $\overline{X}$, $\overline{Y^+}$ and $\overline{Y^-}$ depend on the chosen policy π, though it is not explicitly written in order not to overload notations. The quantity $c_N(\lambda - \delta)$ represents the manufacturing cost for the demand that cannot be satisfied by remanufactured products (we have to satisfy all demands) and does not depend on the policy used. The quantity $c_R\delta$ represents the remanufacturing cost (we have to remanufacture all returns, there is no disposal) and does not depend on the policy used. The quantities

$h_R\overline{X}$, $h_S\overline{Y^+}$, $b\overline{Y^-}$ are respectively the average holding cost related to the inventory of product returns, the average holding cost related to the serviceable inventory of new and remanufactured products and the average backlog cost, which depend on the policy used. The optimal policy that minimizes the total undiscounted average cost per unit of time does not depend on c_N and c_R. Therefore, we can set $c_N = c_R = 0$ without loss of generality.

4 Structure of the optimal policy

4.1 Markov decision process formulation

We model the system by a continuous-time Markov decision process (MDP) (see e.g. Puterman, 1994). In state $(X(t), Y(t))$, the system incurs a cost rate $C(X, Y) = h_R X + h_S Y^+ + bY^-$. Policy π specifies for the server when to manufacture, when to remanufacture and when to be idle. The discounted expected cost over an infinite horizon of policy π, with initial state (x, y) and discount rate $\alpha > 0$, is given by

$$v_\alpha^\pi(x,y) = E\left[\int_0^{+\infty} e^{-\alpha t} C(X(t),Y(t))dt \,|(X(0),Y(0)) = (x,y), \pi\right]. \tag{4}$$

The objective is to find the optimal policy π^* that minimizes the expected discounted-cost $v_\alpha^\pi(x, y)$ over an infinite horizon. We denote by $v_\alpha^*(x, y)$ the optimal value function:

$$v_\alpha^*(x,y) = \min_\pi v_\alpha^\pi(x,y)\,. \tag{5}$$

The optimal policy for the average-cost problem minimizes

$$AC^* = \min_\pi \lim_{T\to+\infty} \frac{E\left[\int_0^T C(X(t),Y(t))dt \,|(X(0),Y(0)) = (x,y), \pi\right]}{T}. \tag{6}$$

The undiscounted average-cost optimal policy has the same structure as the optimal policy for the discounted-cost problem and can be obtained as the limit of the discounted-cost optimal policy by letting α go to 0 (Weber and Stidham, 1987).

After uniformizing the MDP using rate $u = \lambda + \delta + \mu_R + \mu_N$ and normalizing $u + \alpha = 1$, we transform the continuous-time MDP into a discrete MDP (Lippman, 1975; Puterman, 1994). The optimal value function v_α^* satisfies the following optimality equation:

$$v_\alpha^* = Tv_\alpha^* \tag{7}$$

with the operator T defined, for any value function v, by

$$Tv(x,y) = C(x,y) + \lambda v(x,y-1) + \delta v(x+1,y) + (\mu_R+\mu_N)v(x,y) + T_{server}v(x,y) \tag{8}$$

and

$$
\begin{aligned}
T_{server}v\,(x,y) &= \min\left\{0;\ \mu_N\Delta_y v(x,y);\ \mathbf{1}_{x>0}.\mu_R\Delta_R v(x,y)\right\}\,, \qquad (9)\\
\Delta_y v(x,y) &= v(x,y+1) - v(x,y)\,,\\
\Delta_R v(x,y) &= v(x-1,y+1) - v(x,y) = -\Delta_{xy}v(x-1,y)\,,\\
\mathbf{1}_{x>0}&, \text{the indicator function for } x>0\,.
\end{aligned}
$$

In the following subsections, several structural properties of the optimal policy are detailed for some specific sets of system parameters values.

4.2 Structural results for $h_R \geq h_S$ and $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$

In this section, we show that the optimal policy is a 3PR base-stock policy (see definition 1 below) when $h_R \geq h_S$ and $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$. At first sight, the condition $h_R \geq h_N$ may not seem to be often fulfilled in practice. However, when remanufacturing implies a disassembly operation, products that are returned can take far more space than the individual components that we want to reuse, incurring large storage costs. In practice, components are often not disassembled before remanufacturing starts, in order to avoid (further) damage. This may be the case, for instance, with car wrecks or photocopy machines. The second condition $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$ is weaker than the condition $\mu_R \geq \mu_N$ which simply states that it is faster to remanufacture a return (in average) than to manufacture a new product.

Definition 1. *A Push Preemptive Priority on Returns (3PR) base-stock policy is such that :*

- *When there are returns ($x > 0$), remanufacturing has always priority over the production of new products. If a return occurs while the server is manufacturing a new product, then the server stops immediately manufacturing and switches to remanufacturing returns, until the remanufacturable inventory is empty ($x = 0$). When the server is idle at the moment a return arrives, the server immediately starts remanufacturing this return.*
- *When there are no returns left ($x = 0$), the server manufactures if and only if the serviceable inventory y is below a certain base-stock level S.*

The following matrix displays one example of the optimal policy structure, where each element in the matrix corresponds to the optimal decision for the server. Letters I, R and M refer respectively to "idle", "remanufacturing" and "manufacturing". Because we cannot display huge matrices, we only provide a representative extract of the total decision matrix. This matrix was computed with a value iteration algorithm (see e.g. Puterman, 1994).

To show that the optimal policy is a 3PR base-stock policy, we need to define the following set of value functions $\mathcal{V}$.

Definition 2. *A value function v belongs to $\mathcal{V}$ if for all (x,y)*

(P1) $\forall x > 0,\ \Delta_R v(x,y) \leq 0$.

(P2) $\forall x > 0,\ \mu_R\Delta_R v(x,y) \leq \mu_N\Delta_y v(x,y)$.

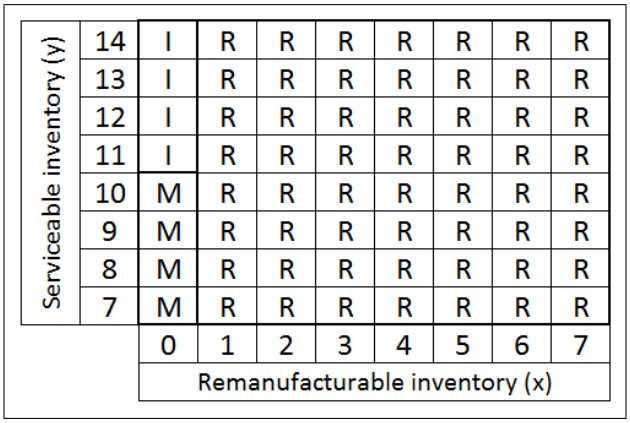


Figure 2: Optimal policy when $h_R \geq h_S$ and $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$ (based on the instance with parameter values $\lambda = 1, \delta = 0.4, \mu_R = 2, \mu_N = 1, h_R = 2, h_S = 1, b = 10$)

(P3) $\Delta_y v(x,y) \leq \Delta_y v(x,y+1)$.

In the appendix, we show that the optimal value function v_α^* belongs to $\mathcal{V}$. The proof is based on the property that any sequence of value functions (v_n) defined as $v_{n+1} = Tv_n$ will converge to the optimal value function v_α^*, the unique solution of the optimality equation $v_\alpha^* = Tv_\alpha^*$ (Puterman, 1994). Hence, to prove that $v_\alpha^* \in \mathcal{V}$, it suffices to prove that if $v \in \mathcal{V}$, then $Tv \in \mathcal{V}$.

The properties of the optimal value function results in properties of the optimal policy. Property (P1) ensures that, for $x > 0$, it is better to remanufacture than doing nothing. Property (P2) ensures that, for $x > 0$, it is better to remanufacture than to manufacture. Property (P3) ensures that, when $x = 0$, it is better to manufacture below a certain threshold (base-stock policy).

Theorem 1. *If $h_R \geq h_S$ and $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$, then the optimal value function v_α^* belongs to $\mathcal{V}$. Moreover, the optimal discounted-cost (or average-cost) policy is a 3PR base-stock policy.*

The proof of Theorem 1 is given in the appendix. We can explain Theorem 1 as follows. If $h_R \geq h_S$, then it is logical to remanufacture returns to save inventory costs. Giving priority to remanufacturing is due to $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$. To understand this, we rewrite the condition $\mu_R\left(1+\frac{h_R}{b}\right) \geq \mu_N$ as $\frac{b+h_R}{\mu_N} \geq \frac{b}{\mu_R}$. We can interpret this inequality as follows: if there is a shortage and if we remanufacture, it takes on average $\frac{1}{\mu_R}$ time units to satisfy the shortage and it costs b (due to the shortage) during this period. If we manufacture, it takes on average $\frac{1}{\mu_N}$ time units to satisfy the shortage and it costs $h_R + b$ during this period (because by manufacturing a new product, one returned product is still kept in the remanufacturable inventory). Since $\frac{b+h_R}{\mu_N} \geq \frac{b}{\mu_R}$, it is better to remanufacture.

4.3 Structural results for $\mu_R = \mu_N$

In this section, we focus on the properties of the optimal policy when the remanufacturing rate is equal to the manufacturing rate. In practice, $\mu_R = \mu_N$ holds for instance if the remanufacturing operation is the same as the manufacturing operation. For example assembling two new components or two used components requires about the same amount of time.

When $\mu_R = \mu_N$ and $h_S \leq h_R$, we have already shown that the optimal policy is a 3PR base-stock policy in Theorem 1. When $\mu_R = \mu_N$ and $h_S > h_R$, the structure of the optimal policy is more complex and is illustrated in Figure 3.

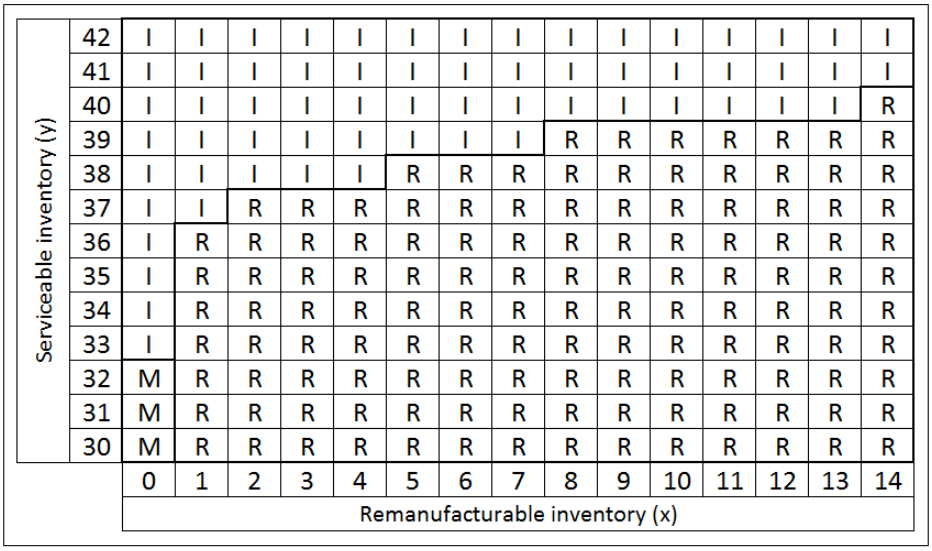


Figure 3: Optimal policy for $\mu_R = \mu_N$ and $h_S > h_R$ (based on the instance: $\lambda = 1; \delta = 0.7; \mu_R = 1.1; \mu_N = 1.1; h_R = 2; h_S = 3; b = 100$)

When $\mu_R = \mu_N = \mu$, we can rewrite the operator T as follows

$$Tv(x,y) = C(x,y) + \lambda v(x,y-1) + \delta v(x+1,y) + \mu T_s v(x,y) \tag{10}$$

$$\text{with } T_s v(x,y) = \begin{cases} \min\{v(x,y), v(x,y+1)\} & \text{if } x = 0 \\ \min\{v(x,y), v(x-1,y+1), v(x,y+1)\} & \text{if } x > 0 \end{cases} \tag{11}$$

The following notations will be useful to derive the optimal policy: $\Delta_x v(x,y) = v(x+1,y) - v(x,y)$, $\Delta_y v(x,y) = v(x,y+1) - v(x,y)$ and $\Delta_{xy} v(x,y) = v(x+1,y) - v(x,y+1)$. We will also combine these notations. For instance, we have $\Delta_y \Delta_y v(x,y) = \Delta_y v(x,y+1) - \Delta_y v(x,y)$ and $\Delta_x \Delta_{xy} v(x,y) = \Delta_{xy} v(x+1,y) - \Delta_{xy} v(x,y)$. In order to prove that the optimal policy follows the pattern of Figure 3, we introduce a second set of value functions $\mathcal{W}$.

Definition 3. *A value function v belongs to $\mathcal{W}$ if for all (x,y)*

(P2) $\Delta_x v(x,y) \geq 0$ *(equivalent to Property (P2) in $\mathcal{V}$ when $\mu_R = \mu_N$)*

(P3) $\Delta_y \Delta_y v(x,y) \geq 0$

(P4) $\Delta_x \Delta_y v(x,y) \geq 0$

(P5) $\Delta_y \Delta_{xy} \leq 0$

(P6) $\Delta_x \Delta_{xy} \geq 0$

In Koole (1998), Property (P3) is called convexity, Property (P4) is called supermodularity and properties (P5) and (P6) are called superconvexity. Properties (P4) and (P6) together imply Property (P3) (Koole, 1998). In the appendix, we show that the optimal value function v_α^* belongs to $\mathcal{W}$. This result is not straightforward since operator T_{server} has not been studied in the literature so far. The closest operator is probably the one considered by Ha (1997) who considers the dynamic scheduling of a make-to-stock system.

The properties satisfied by the optimal value function yield properties of the optimal policy. Property (P2) ensures that, for $x > 0$, it is better to remanufacture than to manufacture. When $x = 0$, Property (P3) ensures the existence of a threshold level $S(0) = \min\{y : \Delta_y v^*_\alpha(0, y) \geq 0\}$ such that it is optimal to manufacture when $y < S(0)$ and to idle otherwise. When $x > 0$, Property (P5) ensures the existence of a threshold level $S(x) = \min\{y : -\Delta_{xy} v^*_\alpha(x-1, y) \geq 0\}$ such that it is optimal to remanufacture when $y < S(x)$ and to idle otherwise. Property (P6) implies that $S(x+1) \geq S(x)$ when $x > 0$ and Property (P2) implies that $S(1) \geq S(0)$.

Theorem 2. *If $\mu_R = \mu_N$, then the optimal value function v^*_α belongs to $\mathcal{W}$. Moreover, the optimal discounted-cost (or average-cost) policy have the following structural properties. There exists a switching curve $S(x)$ such that :*

1. *When $x = 0$, it is optimal to manufacture new products when $y < S(0)$ and to idle otherwise.*
2. *When $x > 0$, it is optimal to remanufacture when $y < S(x)$ and to idle otherwise.*
3. *For all $x \geq 0$, $S(x) \leq S(x+1)$.*

The proof of Theorem 2 is given in the appendix.

5 Evaluation and optimization of 3PR base-stock policies

In the previous section we showed that the optimal policy belongs to the class of 3PR base-stock policies when $h_R \geq h_S$ and $\mu_R \left(1 + \frac{h_R}{b}\right) \geq \mu_N$ (see Theorem 1). In this section, we first explain how to compute efficiently the steady-state probabilities under 3PR base-stock policies. Then we provide a more efficient method than stochastic dynamic programming to compute the optimal policy.

For a given 3PR base-stock policy, the two-dimensional stochastic process $(X(t), Y(t))$ can be modelled by a continuous-time two-dimensional Markov chain, more precisely a quasi-birth-death process (QBD). We refer to Neuts (1981, chapter 3) and Latouche and Ramaswami (1993) for a formal definition of QBD. In Table 1, we show the system states and output transition rates for each area of the Markov chain. For $x = 0$ and $y \geq S$, the server is idle. For $x = 0$ and $y < S$, the server is manufacturing new products. For $x > 0$, the server is remanufacturing product returns.

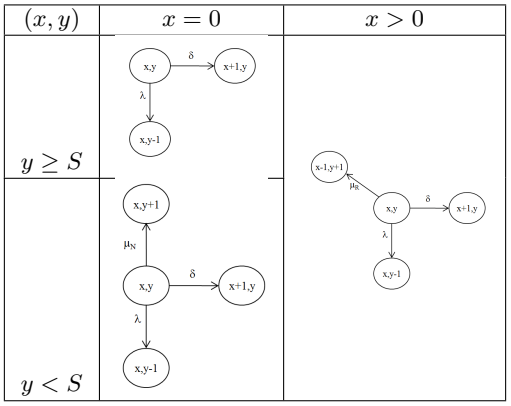


Table 1: Output rates for each area of the Markov chain

Because of the repeating structure of the Markov chain, if we have the steady-state probabilities for any particular base-stock S, then we can deduce the steady-state probabilities for all base-stock levels. Indeed, the two-dimensional stochastic process $(X(t), Y(t) - S)$ is independent of S. Therefore, for a given instance, we need to compute probabilities only for this stochastic process and then we can derive easily probabilities for any base-stock level. This property will be very useful in the optimization procedure.

For all $i \in \mathbb{N}$ and $j \in \mathbb{Z}$, we denote the steady-state probability of the state (i, j) by $\pi_{i,j}$. For a given base stock level S, the steady-state equations related to the Markov chain are

$$\begin{cases} (\lambda + \delta)\,\pi_{0,j} = \lambda\pi_{0,j+1} + \mu_R\pi_{1,j-1} & \text{for } j > S \\ (\lambda + \delta)\,\pi_{0,S} = \lambda\pi_{0,S+1} + \mu_R\pi_{1,S-1} + \mu_N\pi_{0,S-1} & \text{for } j = S \\ (\lambda + \delta + \mu_N)\,\pi_{0,j} = \lambda\pi_{0,j+1} + \mu_R\pi_{1,j-1} + \mu_N\pi_{0,j-1} & \text{for } j < S \\ (\lambda + \delta + \mu_R)\,\pi_{i,j} = \lambda\pi_{i,j+1} + \mu_R\pi_{i+1,j-1} + \delta\pi_{i-1,j} & \text{for } i \geq 1 \end{cases} \tag{12}$$

To compute the steady-state probabilities, we need to truncate the state space. To do so, we increase the state space until the steady-state probabilities are no longer sensitive to further increasing the state space. We denote by X_{max}, Y_{min} and Y_{max} respectively the maximum remanufacturable inventory level, the minimum serviceable inventory level and the maximum serviceable inventory level in the truncated state space. In the following section, we detail the method used to compute them.

5.1 Algorithm to compute the steady-state probabilities

There are many methods to solve a QBD. We refer to Latouche and Ramaswami (1993) and van Leeuwaarden and Winands (2006) for an overview. Here we adopt the matrix-geometric approach (see chapter 3 in Neuts, 1981), which is usually used in problems that deal with our type of QBD (see e.g. Song et al., 1999; Chang and Lu, 2008, 2010).

Firstly, we note that after truncation, there are $(X_{max} + 1) \times (Y_{max} - Y_{min} + 1)$ different states. For $k = 0 \dots X_{max}$, we denote by p_k the line-vector $(\pi_{k,Y_{min}}, \dots, \pi_{k,Y_{max}})$.

To compute the steady-state probabilities, we derive the generator matrix Q related to the Markov chain. Based on (12), we know that matrix Q is a block-tridiagonal matrix of size $(X_{max} + 1)^2 \times (Y_{max} - Y_{min} + 1)^2$. There are 4 types of blocks: B_1, A_0, A_1 and A_2, which are detailed in the appendix.

$$Q = \begin{pmatrix} B_1 & A_0 & 0 & \dots & 0 & 0 & 0 \\ A_2 & A_1 & \ddots & \ddots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & A_1 & A_0 \\ 0 & 0 & 0 & \dots & 0 & A_2 & A_1 + A_0 \end{pmatrix}$$

The steady-state probabilities satisfy the following global balance and normalization equations

$$\begin{cases} \pi Q = \mathbf{0} & \text{(13a)} \\ \sum_{i,j} \pi_{i,j} = 1 & \text{(13b)} \end{cases}$$

with π the line-vector of steady-state probabilities and $\mathbf{0}$ the line-vector null.

Equation (13a) is equivalent to the following set of equations

$$\begin{cases} p_0 B_1 + p_1 A_2 = \mathbf{0} & \text{(14a)} \\ p_{k-1} A_0 + p_k A_1 + p_{k+1} A_2 = \mathbf{0} \qquad \text{if } 1 \leq k < X_{max} & \text{(14b)} \\ p_{X_{max}-1} A_0 + p_{X_{max}} (A_0 + A_1) = \mathbf{0}, & \text{(14c)} \end{cases}$$

Now, we define recursively the following matrices R_k

$$\begin{aligned} R_{X_{max}} &= -A_0 (A_0 + A_1)^{-1} , \\ R_k &= -A_0 (A_1 + R_{k+1} A_2)^{-1} \quad \text{for } 1 \leq k < X_{max} , \\ R_0 &= Id . \end{aligned} \tag{15}$$

We can show that (see e.g. chapter 3 in Neuts, 1981)

$$p_k = p_0 \prod_{i=0}^{k} R_i \quad \text{for } 0 \leq k \leq X_{max} . \tag{16}$$

In this way, we obtain the following set of equations (equivalent to (13a) and (13b))

$$\begin{cases} p_0 (B_1 + R_1 A_2) = \mathbf{0} & \text{(17a)} \\ p_0 \sum_{k=0}^{X_{max}} \left(\prod_{i=0}^{k} R_i \right) \mathbf{1}' = 1 & \text{(17b)} \end{cases}$$

with $\mathbf{1}$ the line-vector which contains only 1, and $\mathbf{1}'$ the transpose of this vector.

Now, we only need to solve (17) to determine the steady-state probabilities. Solving these equations requires to invert matrices of size $(Y_{max} - Y_{min} + 1)^2$. This is much faster than solving (13) that requires to invert matrices of size $(X_{max} + 1)^2 \times (Y_{max} - Y_{min} + 1)^2$.

Once we have the steady-state probabilities, it is straightforward to compute the total average cost AC of the system as follows:

$$\begin{aligned} AC &= h_S \overline{Y^+} + b \overline{Y^-} + h_R \overline{X} \\ &= \sum_{x=0}^{X_{max}} \sum_{y=1}^{Y_{max}} y \, \pi_{x,y} \, h_S + \sum_{x=0}^{X_{max}} \sum_{y=Y_{min}}^{-1} (-y) \, \pi_{x,y} \, b + h_R \frac{\rho_R^2}{1 - \rho_R} . \end{aligned} \tag{18}$$

Because of the 3PR base-stock policy, the remanufacturing stage behaves like an $M/M/1$ queue with an utilization rate of ρ_R and we know that the average number of jobs in the queue is $\frac{\rho_R^2}{1-\rho_R}$.

5.2 Optimization procedure

The algorithm described above allows us to compute efficiently the steady-state probabilities and the average cost for a given base-stock level. The following theorem shows how to compute the optimal base-stock level S^*, with very few little extra computational efforts.

Theorem 3. *The total average cost function $AC(\cdot)$ is convex in the base stock level S. If we denote by $F(\cdot)$ the cumulative distribution function of the random variable $S-Y$, then the optimal base-stock level is given by*

$$S^* = \min\left\{S : F(S) > \frac{b}{h_S + b}\right\}. \tag{19}$$

Proof. This proof is adapted from equations (5) and (6) in Veatch and Wein (1996). S is the base-stock level. We consider the random variable $S-Y$. Functions $f(.)$ and $F(.)$ are respectively its probability density function and its cumulative distribution function. We have

$$\begin{aligned} AC(S) &= b\sum_{k=-\infty}^{-1}(-k)\,P\,(Y=k) + h_S\sum_{k=1}^{+\infty}kP\,(Y=k) \\ &= b\sum_{k=S+1}^{+\infty}(k-S)\,f\,(k) + h_S\sum_{k=-\infty}^{S-1}(S-k)f\,(k)\,. \end{aligned}$$

Then

$$\begin{aligned} AC(S+1) - AC(S) &= b\sum_{k=S+2}^{+\infty}(k-S-1)\,f\,(k) + h_S\sum_{k=-\infty}^{S}(S+1-k)f\,(k) \\ &\quad - b\sum_{k=S+1}^{+\infty}(k-S)\,f\,(k) - h_S\sum_{k=-\infty}^{S-1}(S-k)f\,(k) \\ &= -b + (h_S+b)\,F\,(S)\,. \end{aligned}$$

Since the cumulative distribution function $F(.)$ is non-decreasing, the function $AC(S+1) - AC(S)$ is non-decreasing in S. Therefore, $AC(.)$ is convex in S. Moreover, the optimal base-stock level is given by

$$S^* = \min\left\{S : F(S) > \frac{b}{h_S + b}\right\}.$$

□

Once steady-state probabilities have been computed for a given base-stock level with the QBD method (e.g. $S=0$), we can use a dichotomic algorithm to compute the optimal base-stock level with (19). The QBD method with dichotomy search has been implemented in *MATLAB* (version 7.9). Using a double-core 2 Ghz processor, it took between less than one second and about one minute to find the optimal base-stock level for a given instance, with an average time of one second based on 1024 instances.

6 Insights into the structure of the optimal policy

In this section we want to show how the structure of the optimal policy depends on the values of the system parameters. To compute the optimal policy for a given instance, we used stochastic dynamic programming (Puterman, 1994). Four different situations are investigated:

1. Situation I : $h_R \geq h_S$ and $\mu_R \left(1 + \frac{h_R}{b}\right) \geq \mu_N$.
2. Situation II : $h_R \geq h_S$ and $\mu_R \left(1 + \frac{h_R}{b}\right) < \mu_N$.
3. Situation III : $h_R < h_S$ and $\mu_R \left(1 + \frac{h_R}{b}\right) < \mu_N$.
4. Situation IV : $h_R < h_S$ and $\mu_R \left(1 + \frac{h_R}{b}\right) \geq \mu_N$.

Figure 4 illustrates for which values of μ_R and h_S the four situations hold. When $\rho_{server} \geq 1$, the system is unstable and the number of backlogged demands goes to infinity. For this figure, we assume that the unstable region is not too big in order to have the four situations in the stable region. For this, we need that $\frac{\delta}{1-\rho_N} < \frac{\mu_N}{1+h_S/b}$. In the numerical study that follows, we assume that this condition is always satisfied.

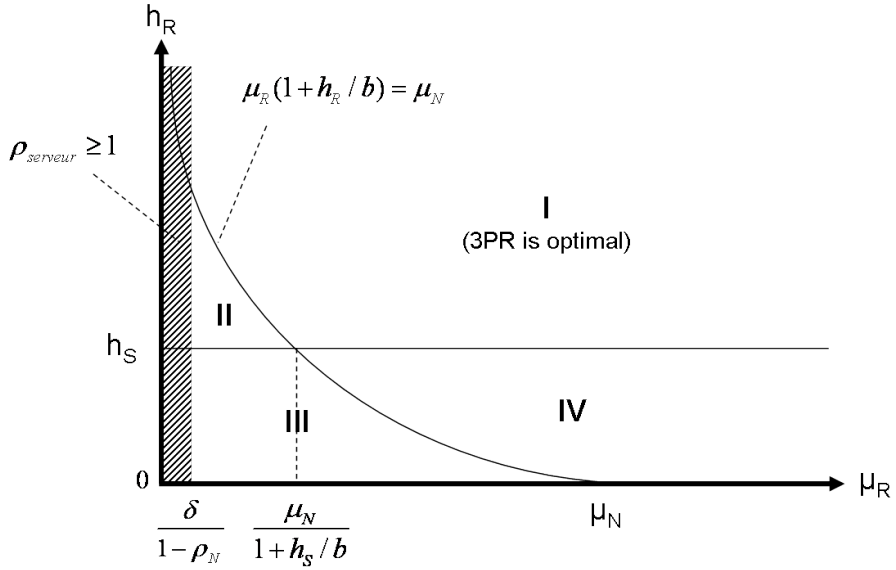


Figure 4: The four situations in (μ_R, h_R) graph

For each of the four different situations we plot an example of the optimal policy (see Figure 5). For situation I, we have proved in Theorem 1 that the optimal policy is a 3PR base-stock policy.

For situation II, the shape of the optimal policy can be explained as follows. It is always better to remanufacture rather than idle the server, since it reduces both the backorder costs and the holding costs (because $h_R \geq h_S$). As a consequence, when the remanufacturable inventory is not zero, the server is never idle and either manufactures or remanufactures. As the return flow is not sufficient to satisfy all the demand, we know that there is a region where the server should manufacture. As the manufacturing rate is higher than the remanufacturing rate ($\mu_N > \mu_R$), it is better to manufacture when the serviceable inventory is low in order to diminish backlog costs. We also observe that the switching

Serviceable inventory (y)								
14	I	R	R	R	R	R	R	R
13	I	R	R	R	R	R	R	R
12	I	R	R	R	R	R	R	R
11	I	R	R	R	R	R	R	R
10	M	R	R	R	R	R	R	R
9	M	R	R	R	R	R	R	R
8	M	R	R	R	R	R	R	R
7	M	R	R	R	R	R	R	R
	0	1	2	3	4	5	6	7
	Remanufacturable inventory (x)							

(a) Situation I ($\lambda = 1, \delta = 0.4, \mu_R = 2, \mu_N = 1, h_R = 2, h_S = 1, b = 10$)

Serviceable inventory (y)								
5	I	R	R	R	R	R	R	R
4	I	R	R	R	R	R	R	R
3	M	R	R	R	R	R	R	R
2	M	R	R	R	R	R	R	R
1	M	R	R	R	R	R	R	R
0	M	M	M	M	M	R	R	R
-1	M	M	M	M	M	M	M	M
-2	M	M	M	M	M	M	M	M
	0	1	2	3	4	5	6	7
	Remanufacturable inventory (x)							

(b) Situation II ($\lambda = 1; \delta = 0.4; \mu_R = 1; \mu_N = 2; h_R = 2; h_S = 1; b = 10$)

Serviceable inventory (y)								
6	I	I	I	R	R	R	R	R
5	I	I	R	R	R	R	R	R
4	I	R	R	R	R	R	R	R
3	I	R	R	R	R	R	R	R
2	I	R	R	R	R	R	R	R
1	M	R	R	R	R	R	R	R
0	M	M	M	M	R	R	R	R
-1	M	M	M	M	M	M	M	M
	0	1	2	3	4	5	6	7
	Remanufacturable inventory (x)							

(c) Situation III ($\lambda = 1; \delta = 0.4; \mu_R = 1; \mu_N = 2; h_R = 1; h_S = 2; b = 10$)

Serviceable inventory (y)								
9	I	I	I	I	I	I	I	I
8	M	I	I	I	I	I	I	I
7	M	M	M	I	I	I	I	I
6	M	M	M	M	I	I	I	I
5	M	M	M	M	M	M	I	I
4	M	M	M	M	M	M	R	R
3	M	R	R	R	R	R	R	R
2	M	R	R	R	R	R	R	R
	0	1	2	3	4	5	6	7
	Remanufacturable inventory (x)							

(d) Situation IV ($\lambda = 1; \delta = 0.4; \mu_R = 2; \mu_N = 1; h_R = 1; h_S = 2; b = 10$)

Figure 5: Structure of the optimal policy for the four situations

curve, which delimitates when to manufacture and when to remanufacture, is slightly decreasing in the remanufacturable inventory in order to reduce remanufacturable inventory holding costs.

In situation III, we still have $\mu_N > \mu_r$ but now $h_R < h_S$. The structure of the optimal policy is very close to the one for situation II except that the server is idle when the serviceable inventory is high enough in order to limit serviceable inventory holding costs. We also observe that the switching curve, which delimitates when to remanufacture and when to be idle, is increasing in the remanufacturable inventory. This entails reduction of remanufacturable inventory holding costs.

In situation IV (Figure 5d), it is the reverse of situation III since it is better to remanufacture when the serviceable inventory is low enough. This is due to smaller remanufacturing leadtimes ($\mu_R > \mu_N$). The switching curve, which delimitates when to remanufacture and idle, is decreasing in the remanufacturable inventory since the higher x is, the lower the remanufacturing replenishment leadtimes will be.

Finally, we note that in all situations if there are backorders ($y < 0$), the server is never idle in order to diminish backorder costs as well as holding costs.

7 Some simple heuristic threshold policies

For situation I, we have proved in Theorem 1 that the optimal policy belongs to the class of 3PR base-stock policies. In this section we introduce new simple heuristic policies for the three other situations and compare their performance with the optimal policy.

These heuristic policies, called RM, IRM and IR, are presented in Table 2 and have been designed for respectively situations II, III and IV. We have on purpose chosen very simple policies defined by at most two thresholds. One could imagine more complex heuristic policies with more policy parameters. However the interest would be limited. On one hand, it would reproduce somehow the complexity of the optimal policy and would be relatively difficult to implement in practice. On the other hand, the time to find the optimal policy parameters is exponential within the number of policy parameters. In Table 2, we also remind the shape of 3PR which is a special case of RM and IR.

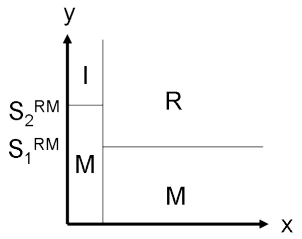
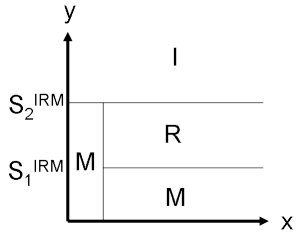
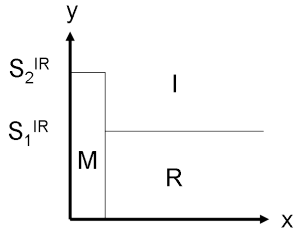
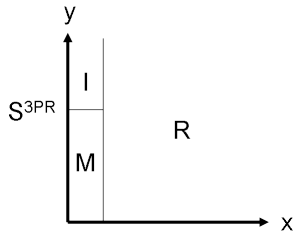
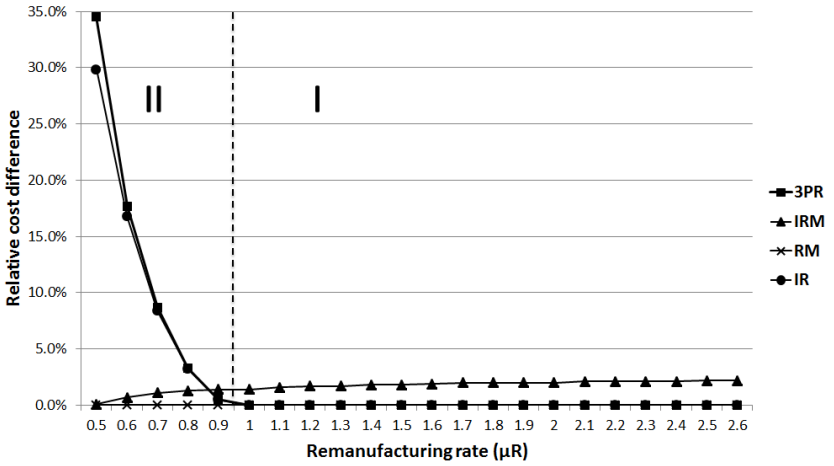
Heuristic names	Illustration	Details
RM (designed for situation II)		If $x = 0$, manufacture when $y < S_2^{RM}$ and idle the server otherwise. If $x > 0$, remanufacture when $y \geq S_1^{RM}$ and manufacture otherwise.
IRM (designed for situation III)		If $x = 0$, manufacture when $y < S_2^{IRM}$ and idle the server otherwise. If $x > 0$, manufacture when $y < S_1^{IRM}$, remanufacture when $S_1^{IRM} \leq y < S_2^{IRM}$ and idle the server when $y \geq S_2^{IRM}$.
IR (designed for situation IV)		If $x = 0$, manufacture when $y < S_2^{IR}$ and idle the server otherwise. If $x > 0$, remanufacture when $y < S_1^{IR}$, and idle the server otherwise.
3PR (optimal for situation I)		3PR is a special case of RM and IR if we set respectively $S_1^{RM} = -\infty$, $S_2^{RM} = S^{3PR}$ and $S_1^{IR} = +\infty$, $S_2^{IR} = S^{3PR}$

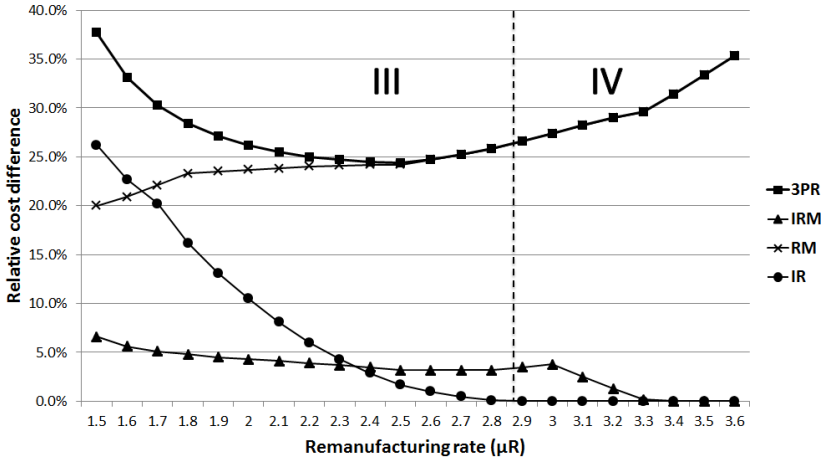
Table 2: Heuristic policies

Hereafter we compare these heuristic policies with the optimal policy. For each heuristic policy, we compute the optimal values of the policy parameters and the corresponding optimal average cost. In the rest of the paper, the policy parameters are optimized and simply denoted by S_1^{RM} etc. The relative cost difference is defined by the relative average cost increase (AC(heuristic policy)-AC(optimal policy))/AC(optimal policy) that results from using a heuristic policy (RM, IRM, IR and R) instead of the optimal policy.

Figures 6 and 7 show respectively the effect of μ_R and h_R on the cost performances of RM, IRM, IR and 3PR. The dashed vertical lines indicate when the system goes from one situation to another.



(a) $\lambda = 1, \delta = 0.1, \mu_N = 1.3, h_R = 8, h_S = 1, b = 20$



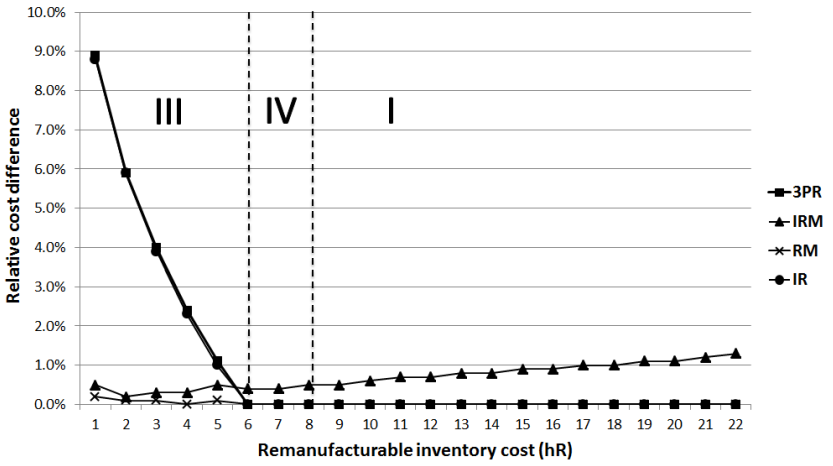
(b) $\lambda = 1, \delta = 0.5, \mu_N = 3, h_R = 1, h_S = 10, b = 20$

Figure 6: Performance of the heuristic policies as a function of μ_R

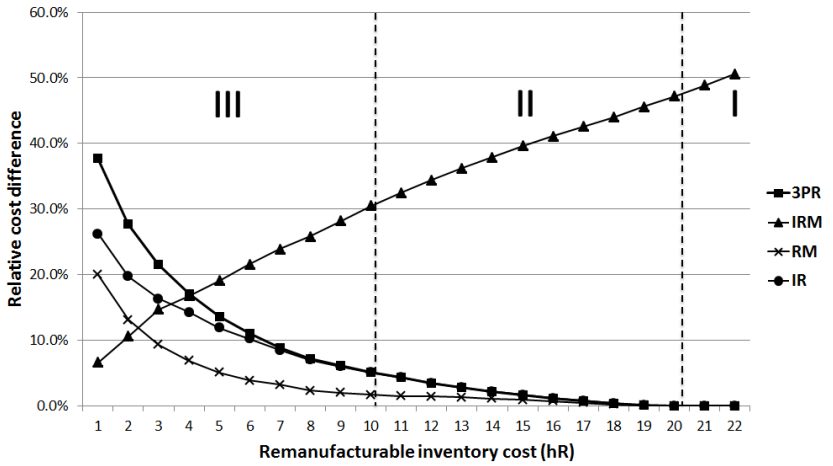
In situation I, the figures show that 3PR is optimal, in accordance with Theorem 1, and that IR and RM are also optimal, as they generalize 3PR. For the three other situations, we observe that the 3PR policy is no longer optimal and can result in a very poor performance. This is expected because the 3PR policy suggests to remanufacturing whenever possible. In situations with low remanufacturing rates (II and III), it is better to reduce the backlog by quickly manufacturing new products. When the remanufacturing holding cost is small (III and IV), it is preferable to keep returned products in the relatively cheap remanufacturable inventory when the serviceable inventory is sufficiently high.

In situation II, figures 6 and 7 show that RM is the best heuristic policy and has a very good performance with a maximum relative cost increase smaller than 2 %. This is due to the very close structure of the RM policy to the optimal one in situation II. Though we have not been able to prove it, it is intuitively clear that the optimal policy never idles the server when $x > 0$, in agreement with the RM policy. It is also reasonable that the optimal policy uses the fastest server mode when the serviceable inventory is low, as does the RM policy.

In situation IV, IR works very well with a relative cost increase smaller than 0.1 % on the tested instances. This is again due to the structural closeness of the IR policy with the optimal one in situation



(a) $\lambda = 1, \delta = 0.1, \mu_R = 1, \mu_N = 1.3, h_S = 8, b = 20$



(b) $\lambda = 1, \delta = 0.5, \mu_R = 1.5, \mu_N = 3, h_S = 10, b = 20$

Figure 7: Performance of the heuristic policies as a function of h_R

IV. The IR policy never manufactures when the remanufacturable inventory is not zero, although the optimal policy may do so. However we believe that has little effect on the costs because the manufacturing rate is lower than the remanufacturing rate.

It is much more difficult to design simple efficient heuristic policies for situation III, due to the complex structure of the optimal policy which can not be imitated with a simple threshold policy. None of the heuristic policies work well systematically, and the relative cost increase can be larger than 25 % on our test bed. Finally, we observe that there exist instances for which all heuristic policies have poor performances (see Figure 7b: when $h_R \simeq 2$, the relative cost increase is greater than 10 %).

8 Conclusion and further research

We have studied a make-to-stock queue with product returns and a remanufacturing stage where manufacturing and remanufacturing are performed by one single server. To the best of our knowledge, this problem has never been studied before in a stochastic environment. For a specific set of system parameters values, we were able to characterize completely the optimal policy which is a 3PR base-stock

policy. In the class of 3PR base-stock policies, we have provided an efficient method to compute the steady-state probabilities and the optimal policy. We also partially characterized the optimal policy when the remanufacturing rate is equal to the manufacturing rate. Additionally, we have given some insights into the optimal policy structure in the general case. Based on these insights, we have introduced three simple threshold policies (RM, IRM and IR) and compared them to the 3PR base-stock policy and to the optimal policy. The performances of these heuristic policies are very good in some situations. However none of them has a good performance for all situations and there exist situations where none of the heuristic policies performs well. Therefore implementing the optimal policy seems desirable.

There are several ways to extend our model. Firstly, we can consider more general distributions for manufacturing and remanufacturing times. Secondly, we could include to possibility to dispose products. It could mean to reject a return at the moment it is submitted or to dispose a remanufacturable product or even to dispose a serviceable product. The description of the optimal policy would require three additional switching curves to include these three options. Thirdly, it would be interesting to study the problem with setup costs. This requires to add additional state variables to represent the state of the server. Computing the optimal policy remains numerically tractable. However, it is unlikely that structural results could be derived for the optimal policy, because even in the special case without returns, the (s, S) structure of the optimal policy is conjectured but has not been proven properly in a Markov decision process framework. Finally, our model assumes that preemption is allowed. The optimality of the 3PR base-stock policy, for a specific set of system parameters values, is no more guaranteed when preemption is not allowed. Indeed preempting the manufacturing of new products is interesting when the remanufacturing rate is higher than the manufacturing rate. The approach to forbid preemption is similar to the one to include setup costs and requires to add a variable representing the state of the server. Again the structure of the optimal policy is expected to be significantly more complex and it is unlikely to characterize the optimal policy theoretically. However, it would be interesting to test numerically if the 3PR policy is a good approximation for the optimal policy.

References

N. Aras, V. Verter, and T. Boyaci. Coordination and priority decisions in hybrid manufacturing/remanufacturing systems. *Production and Operations Management*, 15:528–543, 2006.

K.H. Chang and Y.S. Lu. Queueing analysis on a mto production system consisting of mts pre-production. *Proceedings of the 4^{th} International Conference on Networked Computing and Advanced Information Management*, 2008.

K.H. Chang and Y.S. Lu. Queueing analysis on a single-station make-to-stock/make-to-order inventory-production system. *Applied Mathematical Modelling*, 34:978–991, 2010.

W.K. Ching, T. Li, and J.G. Xue. On hybrid re-manufacturing systems: A matrix geometric approach. *Proceedings of the International Conference on Industrial Engineering and Systems Management (available online at: http:// hkumath.hku.hk/ imr/IMRPreprintSeries/ 2007/IMR2007-16.pdf)*, 2007.

G.A. DeCroix. Optimal policy for a multiechelon inventory system with remanufacturing. *Operations Research*, 54:532–543, 2006.

R. Dekker, M. Fleischmann, K. Inderfurth, and L.N. van Wassenhove. *Reverse logistics: Quantitative models for closed-loop supply chains*. Berlin: Springer, 2004.

G. Ferrer and D.C. Whybark. From garbage to goods: Successful remanufacturing systems and skills. *Business Horizons*, 43:55–64, 2000.

M. Fleischmann and R. Kuik. On optimal inventory control with independent stochastic item returns. *European Journal of Operational Research*, 151:25–37, 2003.

M. Fleischmann, J.M. Bloemhof-Ruwaard, R. Dekker, E. van der Laan, J.A.E.E van Nunen, and L.N. van Wassenhove. Quantitative models for reverse logistics: A review. *European Journal of Operational Research*, 103:1–17, 1997.

M. Fleischmann, R. Kuik, and R. Dekker. Controlling inventories with stochastic item returns: A basic model. *European Journal of Operational Research*, 138:63–75, 2002.

J.P. Gayon. A make-to-stock queue with product return. *Proceedings of the 12th IFAC International Symposium*, 2006.

A.Y. Ha. Optimal dynamic scheduling policy for a make-to-stock production system. *Operations Research*, 45(1):42–53, 1997.

D.P. Heyman. Optimal disposal policies for a single-item inventory system with returns. *Naval Research Logistics Quarterly*, 24:385–405, 1977.

M.A. Ilgin and S.M. Gupta. Environmentally conscious manufacturing and product recovery (ECMPRO): A review of the state of the art. *Journal of Environmental Management*, 91:563–591, 2010.

K. Inderfurth. Simple optimal replenishment and disposal policies for a product recovery system with leadtimes. *OR Spektrum*, 19:111–122, 1997.

K. Inderfurth and E. van der Laan. Leadtime effects and policy improvement for stochastic inventory control with remanufacturing. *International Journal of Production Economics*, 71:381–390, 2001.

G. Koole. Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*, 30:323–339, 1998.

G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-death proceses. *Journal of Applied Probability*, 30:323–339, 1993.

Y. Li, J. Zhang, J. Chen, and X. Cai. Optimal solution structure for multi-period production planning with returned products remanufacturing. *Asia-Pacific Journal of Operational Research*, 27(05):629–648, October 2010.

S. Lippman. Applying a new device in the optimization of exponential queueing systems. *Operations Research*, 23:687–710, 1975.

J.A. Muckstadt and M.H. Isaac. An analysis of single item inventory systems with returns. *Naval Research Logistics Quarterly*, 28:237–254, 1981.

S. Nahmias and H. Rivera. A deterministic model for a repairable item inventory system with a finite repair rate. *International Journal of Production Research*, 17:215–221, 1979.

M. Neuts. *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, MD, 1981.

S. Pokharel and A. Mutha. Perspectives in reverse logistics: A review. *Resources, Conservation and Recycling*, 53:175–182, 2009.

M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons: Hoboken, New Jersey, 1994.

D.A. Schrady. A deterministic inventory model for repairable items. *Naval Research Logistics Quarterly*, 14:391–398, 1967.

V.P. Simpson. Optimum solution structure for a repairable inventory problem. *Operations Research*, 26: 270–281, 1978.

J.S. Song, S. Xu, and B. Liu. Order-fulfillment performance measures in an assemble-to-order system with stochastic leadtimes. *Operations Research*, 47:131–149, 1999.

S.K. Srivastava. Green supply-chain management: A state-of-the-art literature review. *International Journal of Management Reviews*, 9:53–80, 2007.

O. Tang and R. Teunter. Economic lot scheduling problem with returns. *Production and Operations Management*, 15:488–497, 2006.

R. Teunter. Economic ordering quantities for recoverable item inventory systems. *Naval Research Logistics*, 48:484–495, 2001.

R. Teunter. Lot-sizing for inventory systems with product recovery. *Computers & Industrial Engineering*, 46:431–441, 2004.

R. Teunter and D. Vlachos. On the necessity of a disposal option for returned items that can be remanufactured. *International Journal of Production Economics*, 75:257–266, 2002.

R. Teunter, E. van der Laan, and D. Vlachos. Inventory strategies for systems with fast remanufacturing. *The Journal of Operational Research Society*, 55:475–484, 2004.

M. Thierry, M. Salomon, J. van Nunen, and L.N. van Wassenhove. Strategic issues in product recovery management. *California Management Review*, 37:114–135, 1995.

B. Trebilcock. Return to sender. *Warehousing Management (May)*, pages 25–27, 2002.

E. Van der Laan, M. Salomon, R. Dekker, and L.N. van Wassenhove. Inventory control in hybrid systems with remanufacturing. *Management Science*, 45:733–747, 1999.

J.S.H. van Leeuwaarden and E.M.M. Winands. Quasi-birth-and-death processes with an explicit rate matrix. *Stochastic Models*, 22:77–98, 2006.

M.H. Veatch and L.M. Wein. Scheduling a make-to-stock queue: index policies and hedging points. *Operations Research*, 44:634–647, 1996.

R.R. Weber and S. Stidham. Optimal control of service rates in networks of queues. *Advances in Applied Probability*, 19:202–218, 1987.

S.X. Zhou, Z. Tao, and X. Chao. Optimal control of inventory systems with multiple types of remaunfacturable products. *Manufacturing & Service Operations Management*, 13(1):20–34, 2011.

A Appendix

A.1 Proof of Theorem 1

In all this section, we assume that $h_R \geq h_S$ and $\mu_R \left(1 + \frac{h_R}{b}\right) \geq \mu_N$ and $v \in \mathcal{V}$. Moreover, we assume that $v \in \mathcal{V}$. As v satisfies property (P1) and (P2), we can rewrite operator T as

$$Tv(x,y) = C\left(x,y\right) + \lambda v\left(x,y-1\right) + \delta v\left(x+1,y\right) + \mu_R T_R v\left(x,y\right) + \mu_N T_N v\left(x,y\right) \tag{20}$$

with

$$T_R v(x,y) = \begin{cases} v(x,y) & \text{if } x = 0 \\ v(x-1,y+1) & \text{if } x > 0 \end{cases} \tag{21}$$

$$T_N v(x,y) = \begin{cases} \min\left\{v(x,y);\, v(x,y+1)\right\} & \text{if } x = 0 \\ v(x,y) & \text{if } x > 0 \end{cases} \tag{22}$$

In the following proof we will either use this formulation or the one given in (8).

Property (P1)

Hereafter, we assume that $x > 0$. We have

$$\begin{aligned} \Delta_R Tv(x,y) &= Tv(x-1,y+1) - Tv(x,y) \\ &= \Delta_R C(x,y) + \lambda \Delta_R v(x,y-1) + \delta \Delta_R v(x+1,y) \\ &+ (\mu_R + \mu_N)\Delta_R v(x,y) + T_{server} v(x-1,y+1) - T_{server} v(x,y) \end{aligned}$$

Since v satisfies properties (P1) and (P2), we have $T_{server} v(x,y) = \mu_R \Delta_R v(x,y)$ and then

$$\begin{aligned} \Delta_R Tv(x,y) &= Tv(x-1,y+1) - Tv(x,y) \\ &= \Delta_R C(x,y) + \lambda \Delta_R v(x,y-1) + \delta \Delta_R v(x+1,y) \\ &+ \mu_N \Delta_R v(x,y) + T_{server} v(x-1,y+1) \end{aligned}$$

The first term equals

$$\Delta_R C(x,y) = h_S\left((y+1)^+ - y^+\right) + b\left((y+1)^- - y^-\right) + h_R\left(x - 1 - x\right) \tag{23}$$

$$= \begin{cases} h_S - h_R & \text{if } y \geq 0 \\ -b - h_R & \text{if } y < 0 \end{cases} \tag{24}$$

and is always nonpositive when $h_R \geq h_S$. The second, third and fourth third term are nonpositive since v satisfies (P1). The fifth term, $T_{server} v(x-1,y+1)$, is nonpositive by definition of T_{server} (see (9)). We conclude that $\Delta_R Tv(x,y) \leq 0$, for any $x > 0$, and Tv satisfies property (P1).

Property (P2)

We will show that the following quantity is nonpositive.

$$\begin{aligned}
&\mu_R\Delta_R Tv(x,y) - \mu_N\Delta_y Tv(x,y)\\
&\quad = \big(\mu_R\Delta_R C(x,y) - \mu_N\Delta_y C(x,y)\big) && (25)\\
&\quad + \lambda\big(\mu_R\Delta_R v(x,y-1) - \mu_N\Delta_y v(x,y-1)\big) + \delta\big(\mu_R\Delta_R v(x+1,y) - \mu_N\Delta_y v(x+1,y)\big) && (26)\\
&\quad + \mu_R\big(\mu_R\Delta_R T_R v(x,y) - \mu_N\Delta_y T_R v(x,y)\big) + \mu_N\big(\mu_R\Delta_R T_N v(x,y) - \mu_N\Delta_y T_N v(x,y)\big) && (27)
\end{aligned}$$

We have

$$\mu_R\Delta_R C(x,y) - \mu_N\Delta_y C(x,y) = \begin{cases} \mu_R(h_S - h_R) - \mu_N h_S & \text{if } y \geq 0\\ \mu_R(-b - h_R) + \mu_N b & \text{if } y < 0 \end{cases}$$

that is nonpositive since $h_R \geq h_S$ and $\mu_R\left(1 + \frac{h_R}{b}\right) \geq \mu_N$. Hence, (25) is nonpositive. As v satisfies property (P2), (26) is also nonpositive. (27) will be denoted A and is equal to

$$\begin{aligned}
A = \mu_R\Big(&\mu_R\big(T_R v(x-1,y+1) - T_R v(x,y)\big) - \mu_N\big(T_R v(x,y+1) - T_R v(x,y)\big)\Big)\\
+ \mu_N\Big(&\mu_R\big(T_N v(x-1,y+1) - T_N v(x,y)\big) - \mu_N\big(T_N v(x,y+1) - T_N v(x,y)\big)\Big)
\end{aligned}$$

To show that A is nonpositive, we distinguish two cases: $x > 1$ and $x = 1$.

First case: Assume that $x > 1$. From equations (21) and (22), we have that $T_R v(x-1,y+1) = v(x-2,y+2)$, $T_R v(x,y) = v(x-1,y+1)$, $T_N v(x-1,y+1) = v(x-1,y+1)$ and $T_N v(x,y) = v(x,y)$. Hence

$$\begin{aligned}
A = \mu_R\Big(&\mu_R\big(T_R v(x-1,y+1) - T_R v(x,y)\big) - \mu_N\big(T_R v(x,y+1) - T_R v(x,y)\big)\Big)\\
+ \mu_N\Big(&\mu_R\big(T_N v(x-1,y+1) - T_N v(x,y)\big) - \mu_N\big(T_N v(x,y+1) - T_N v(x,y)\big)\Big)\\
= \mu_R\Big(&\mu_R\Delta_R v(x-1,y+1) - \mu_N\Delta_y v(x-1,y+1)\Big)\\
+ \mu_N\Big(&\mu_R\Delta_R v(x,y) - \mu_N\Delta_y v(x,y)\Big)
\end{aligned}$$

As v satisfies (P2), A is the sum of two nonpositive terms and is also nonpositive.

Second case: Assume that $x = 1$. Then $T_R v(x-1,y+1) = v(x-1,y+1)$, $T_R v(x,y) = v(x-1,y+1)$, $T_N v(x-1,y+1) = \min\{v(x-1,y+1);\, v(x-1,y+2)\}$ and $T_N v(x,y) = v(x,y)$. Hence

$$\begin{aligned}
A = \mu_R\Big(&\mu_R\big(v(x-1,y+1) - v(x-1,y+1)\big) - \mu_N\big(v(x-1,y+2) - v(x-1,y+1)\big)\Big)\\
+ \mu_N\Big(&\mu_R\big(\min\{v(x-1,y+1);\, v(x-1,y+2)\} - v(x,y)\big) - \mu_N\big(v(x,y+1) - v(x,y)\big)\Big)\\
= -\mu_R\mu_N&\big(v(x-1,y+2) - v(x-1,y+1)\big)\\
+ \mu_N\Big(&\mu_R\big(\min\{v(x-1,y+1);\, v(x-1,y+2)\} - v(x,y)\big) - \mu_N\Delta_y v(x,y)\Big)
\end{aligned}$$

As v satisfies property (P2), we have $-\mu_N \Delta_y v(x,y) \leq -\mu_R \Delta_R v(x,y)$ and

$$\begin{aligned}
A &\leq -\mu_R\mu_N\big(v(x-1,y+2) - v(x-1,y+1)\big) \\
&\quad + \mu_N\Big(\mu_R\big(v(x-1,y+2) - v(x,y)\big) - \mu_R\Delta_R v(x,y)\Big) \\
&= \mu_R\mu_N\Big(- v(x-1,y+2) + v(x-1,y+1) + v(x-1,y+2) - v(x,y) - \Delta_R v(x,y)\Big) \\
&= \mu_R\mu_N\Big(v(x-1,y+1) - v(x,y) - v(x-1,y+1) + v(x,y)\Big) \\
&= 0
\end{aligned}$$

We conclude that, for any $x > 0$, $A \leq 0$. Finally, $\mu_R \Delta_R Tv(x,y) \leq \mu_N \Delta_y Tv(x,y)$ for all $x > 0$ and Tv satisfies property (P2).

Property (P3)

From (20), we have

$$\begin{aligned}
\Delta_y Tv(x,y+1) - \Delta_y Tv(x,y) = \lambda(\Delta_y v(x,y) - \Delta_y v(x,y-1)) + \delta(\Delta_y v(x+1,y+1) - \Delta_y v(x+1,y)) \\
+ \mu_R(\Delta_y T_R v(x,y+1) - \Delta_y T_R v(x,y)) \\
+ \mu_N(\Delta_y T_N v(x,y+1) - \Delta_y T_N v(x,y))
\end{aligned}$$

As v satisfies (P3), the first line is nonpositive. By definition of operator T_R (see (21)), we have

$$\Delta_y T_R v(x,y+1) - \Delta_y T_R v(x,y) = \begin{cases} \Delta_y v(x,y+1) - \Delta_y v(x,y) & \text{if } x = 0 \\ \Delta_y v(x-1,y+2) - \Delta_y v(x-1,y+1) & \text{if } x > 0 \end{cases}$$

which is nonpositive since v satisfies (P3). Remains to show that $T_N v$ also satisfies property (P3). When $x > 0$, we have immediately from the definition of T_N (see (22))

$$\Delta_y T_N v(x,y+1) - \Delta_y T_N v(x,y) = \Delta_y v(x,y+1) - \Delta_y v(x,y)$$

which is again nonpositive since v satisfies (P3). When $x = 0$, operator T_N can be seen as the acceptance operator $T_{AC(y)}$ in Koole (1998) which propagates convexity in y.

We conclude that $(\Delta_y Tv(x,y+1) - \Delta_y Tv(x,y) \leq 0)$ and Tv satisfies (P3).

A.2 Proof of Theorem 2

Let $v \in \mathcal{W}$ and let assume that $\mu_R = \mu_N = \mu$. We will show that $Tv \in \mathcal{W}$. Since v satisfies (P2), we have when $x > 0$

$$T_s v(x,y) = \min\{v(x,y), v(x-1,y+1), v(x,y+1)\} = \min\{v(x,y), v(x-1,y+1)\} \tag{28}$$

In the following proof, we will use one or the other expression of $T_s v(x,y)$.

Property (P2)

We would like to show that $\Delta_x Tv(x,y) \geq 0$ for all (x,y). We have

$$\begin{aligned}\Delta_x Tv(x,y) = \Delta_x C(x,y) + \lambda \Delta_x v(x,y-1) + \delta \Delta_x v(x+1,y) \\ + \mu \Delta_x T_s v(x,y)\end{aligned}$$

The first line is nonpositive since $\Delta_x C(x,y) = h_R \geq 0$. By definition of T_s (see (28)), we have when $x = 0$

$$\Delta_x T_s v(x,y) = \min\{v(x+1,y), v(x,y+1)\} - \min\{v(x,y), v(x,y+1)\}$$

If $\min\{v(x+1,y), v(x,y+1)\} = v(x+1,y)$, then $\Delta_x T_1 v(x,y) \geq v(x+1,y) - v(x,y) = \Delta_x v(x,y) \geq 0$. Else $\Delta_x T_1 v(x,y) \geq v(x,y+1) - v(x,y+1) = 0$. We conclude that $\Delta_x T_s v(x,y) \geq 0$ when $x = 0$. Similarly, we can show that $\Delta_x T_s v(x,y) \leq 0$ when $x > 0$. Finally, $\Delta_x Tv(x,y) \geq 0$ for all (x,y) and Tv satisfies (P2).

Property (P4)

We want to show that the following quantity is nonnegative:

$$\begin{aligned}\Delta_y \Delta_x Tv(x,y) = \Delta_y \Delta_x \{C(x,y) + \lambda v(x,y-1) + \delta v(x+1,y)\} \\ + \mu \Delta_y \Delta_x T_s v(x,y).\end{aligned}$$

It is straightforward to check that the first term is nonnegative. To show that $\Delta_y \Delta_x T_s v(x,y) \geq 0$, we distinguish two cases: $x = 0$ and $x > 0$.

First case: Assume that $x = 0$. Unfortunately, we can not reuse the results of the literature and we will have to distinguish four subcases.

$$\begin{aligned}\Delta_y \Delta_x T_s v(x,y) =& T_s v(x+1,y+1) - T_s v(x,y+1) - T_s v(x+1,y) + T_s v(x,y) \\ =& \min[v(x+1,y+1), v(x,y+2)] - \min[v(x,y+1), v(x,y+2)] \\ & - \min[v(x+1,y), v(x,y+1)] + \min[v(x,y), v(x,y+1)]\end{aligned}$$

If $\min[v(x+1,y+1), v(x,y+2)] = v(x+1,y+1)$ and $\min[v(x,y), v(x,y+1)] = v(x,y)$, then

$$\begin{aligned}&\Delta_y \Delta_x T_s v(x,y) \\ &= v(x+1,y+1) - \min[v(x,y+1), v(x,y+2)] - \min[v(x+1,y), v(x,y+1)] + v(x,y) \\ &\geq v(x+1,y+1) - v(x,y+1) - v(x+1,y) + v(x,y) = \Delta_y \Delta_x v(x,y) \geq 0\end{aligned}$$

If $\min[v(x+1,y+1), v(x,y+2)] = v(x,y+2)$ and $\min[v(x,y), v(x,y+1)] = v(x,y+1)$, then

$$\begin{aligned}&\Delta_y \Delta_x T_s v(x,y) \\ &= v(x,y+2) - \min[v(x,y+1), v(x,y+2)] - \min[v(x+1,y), v(x,y+1)] + v(x,y+1) \\ &\geq v(x,y+2) - v(x,y+2) - v(x,y+1) + v(x,y+1) = 0\end{aligned}$$

If $\min[v(x+1,y+1),v(x,y+2)] = v(x+1,y+1)$ and $\min[v(x,y),v(x,y+1)] = v(x,y+1)$, then

$$\begin{aligned}
&\Delta_y\Delta_x T_s v(x,y)\\
&= v(x+1,y+1) - \min[v(x,y+1),v(x,y+2)] - \min[v(x+1,y),v(x,y+1)] + v(x,y+1)\\
&\geq v(x+1,y+1) - v(x,y+1) - v(x,y+1) + v(x,y+1) = -\Delta_x v(x,y+1) \geq 0
\end{aligned}$$

If $\min[v(x+1,y+1),v(x,y+2)] = v(x,y+2)$ and $\min[v(x,y),v(x,y+1)] = v(x,y)$, then

$$\begin{aligned}
&\Delta_y\Delta_x T_s v(x,y)\\
&= v(x,y+2) - \min[v(x,y+1),v(x,y+2)] - \min[v(x+1,y),v(x,y+1)] + v(x,y)\\
&\geq v(x,y+2) - v(x,y+1) - v(x,y+1) + v(x,y+1) = \Delta_y\Delta_y v(x,y) \geq 0
\end{aligned}$$

Second case: Assume that $x > 0$. Fortunately, we can reuse the results of Koole (1998). He shows that the jockeying operator $T_s v(x,y) = \min[v(x,y), v(x-1,y+1)]$ propagates properties (P4), (P5) and (P6) altogether. Hence, when $x > 0$, $\Delta_y\Delta_x T_s v\,(x,y) \geq 0$.

Property (P5)

The proof is similar to the one of Property (P5). The only issue is to show that the following quantity is nonpositive when $x = 0$.

$$\begin{aligned}
\Delta_y\Delta_{xy} T_s v(x,y) =& T_s v(x+1,y+1) - T_s v(x,y+2) - T_s v(x+1,y) + T_s v(x,y+1)\\
=& \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)] - \min[v(x,y+2), v(x,y+3)]\\
&- \min[v(x+1,y), v(x,y+1)] + \min[v(x,y+1), v(x,y+2)]
\end{aligned}$$

If $\min[v(x,y+2),v(x,y+3)] = v(x,y+2)$ and $\min[v(x+1,y),v(x,y+1)] = v(x,y+1)$,

$$\begin{aligned}
\Delta_y\Delta_{xy} T_s v(x,y) &= \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)] - v(x,y+2)\\
&\quad - v(x,y+1) + \min[v(x,y+1), v(x,y+2)]\\
&\leq v(x,y+2) - v(x,y+2) - v(x,y+1) + v(x,y+1) = 0
\end{aligned}$$

If $\min[v(x,y+2),v(x,y+3)] = v(x,y+2)$ and $\min[v(x+1,y),v(x,y+1)] = v(x+1,y)$,

$$\begin{aligned}
\Delta_y\Delta_{xy} T_s v(x,y) &= \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)] - v(x,y+2)\\
&\quad - v(x+1,y) + \min[v(x,y+1), v(x,y+2)]\\
&\leq v(x+1,y+1) - v(x,y+2) - v(x+1,y) + v(x,y+1)\\
&= \Delta_y\Delta_{xy} v(x,y) \leq 0
\end{aligned}$$

If $\min[v(x,y+2), v(x,y+3)] = v(x,y+3)$ and $\min[v(x+1,y), v(x,y+1)] = v(x,y+1)$,

$$\begin{aligned}\Delta_y\Delta_{xy}T_s v(x,y) &= \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)] - v(x,y+3)\\ &\quad - v(x,y+1) + \min[v(x,y+1), v(x,y+2)]\\ &\leq v(x,y+2) - v(x,y+3) - v(x,y+1) + v(x,y+2)\\ &= -\Delta_y\Delta_y v(x,y) \leq 0\end{aligned}$$

If $\min[v(x,y+2), v(x,y+3)] = v(x,y+3)$ and $\min[v(x+1,y), v(x,y+1)] = v(x+1,y)$,

$$\begin{aligned}\Delta_y\Delta_{xy}T_s v(x,y) &= \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)] - v(x,y+3)\\ &\quad - v(x+1,y) + \min[v(x,y+1), v(x,y+2)]\\ &\leq v(x+1,y+2) - v(x,y+3) - v(x+1,y) + v(x,y+1)\\ &= \Delta_{xy}v(x,y+2) - \Delta_{xy}v(x,y) \leq 0\end{aligned}$$

Property (P6)

The proof is similar to the one of properties (P4) and (P5). The only issue is to show that the following quantity is nonnegative when $x = 0$.

$$\begin{aligned}\Delta_x\Delta_{xy}T_s v(x,y) =& T_s v(x+2,y) - T_s v(x+1,y+1) - T_s v(x+1,y) + T_s v(x,y+1)\\ =& \min[v(x+2,y), v(x+1,y+1)] - \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)]\\ &- \min[v(x+1,y), v(x,y+1), v(x+1,y+1)] + \min[v(x,y+1), v(x,y+2)]\end{aligned}$$

If $\min[v(x+2,y), v(x+1,y+1)] = v(x+2,y)$ and $\min[v(x,y+1), v(x,y+2)] = v(x,y+1)$,

$$\begin{aligned}\Delta_x\Delta_{xy}T_s v(x,y) &= v(x+2,y) - \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)]\\ &\quad - \min[v(x+1,y), v(x,y+1), v(x+1,y+1)] + v(x,y+1)\\ &\geq v(x+2,y) - v(x+1,y+1) - v(x+1,y) + v(x,y+1)\\ &= \Delta_x\Delta_{xy}v(x,y) \geq 0\end{aligned}$$

If $\min[v(x+2,y), v(x+1,y+1)] = v(x+1,y+1)$ and $\min[v(x,y+1), v(x,y+2)] = v(x,y+1)$,

$$\begin{aligned}\Delta_x\Delta_{xy}T_s v(x,y) &= v(x+1,y+1) - \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)]\\ &\quad - \min[v(x+1,y), v(x,y+1), v(x+1,y+1)] + v(x,y+1)\\ &\geq v(x+1,y+1) - v(x+1,y+1) - v(x,y+1) + v(x,y+1) = 0\end{aligned}$$

If $\min[v(x+2,y), v(x+1,y+1)] = v(x+2,y)$ and $\min[v(x,y+1), v(x,y+2)] = v(x,y+2)$,

$$\begin{aligned}\Delta_x\Delta_{xy}T_s v(x,y) &= v(x+2,y) - \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)]\\ &\quad - \min[v(x+1,y), v(x,y+1), v(x+1,y+1)] + v(x,y+2)\\ &\geq v(x+2,y) - v(x+1,y+1) - v(x+1,y+1) + v(x,y+2)\\ &= \Delta_{xy}v(x+1,y) - \Delta_{xy}v(x,y+1)\\ &\geq \Delta_{xy}v(x,y) - \Delta_{xy}v(x,y+1) \geq 0\end{aligned}$$

If $\min[v(x+2,y), v(x+1,y+1)] = v(x+1,y+1)$ and $\min[v(x,y+1), v(x,y+2)] = v(x,y+2)$,

$$\begin{aligned}\Delta_x\Delta_{xy}T_s v(x,y) &= v(x+1,y+1) - \min[v(x+1,y+1), v(x,y+2), v(x+1,y+2)] \\ &\quad - \min[v(x+1,y), v(x,y+1), v(x+1,y+1)] + v(x,y+2) \\ &\geq v(x+1,y+1) - v(x,y+2) - v(x+1,y+1) + v(x,y+2) = 0\end{aligned}$$

A.3 Blocks of matrix Q

We use the notation $N = Y_{max} - Y_{min}$. Each block is a $(N+1)^2$ square-matrix.

Block B_1 is defined by

$$B_1 = \begin{pmatrix} \Delta_0 & u_0 & 0 & \dots & 0 & 0 & 0 \\ \lambda & \Delta_1 & u_1 & \ddots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & u_{N-2} & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \Delta_{N-1} & u_{N-1} \\ 0 & 0 & 0 & \dots & 0 & \lambda & \Delta_y \end{pmatrix}$$

with

$$u_m = \begin{cases} \mu_N & \text{for} \quad 0 \leq m \leq S-1 \\ 0 & \text{for} \quad S \leq m \leq N \end{cases}$$

$$\Delta_m = \begin{cases} -\mu_N - \delta & \text{for} \quad m = 0 \\ -\lambda - \mu_N - \delta & \text{for} \quad 1 \leq m \leq S-1 \\ -\lambda - \delta & \text{for} \quad S \leq m \leq N \end{cases}$$

Block A_0 is defined by $A_0 = \delta Id$ with Id is the identity matrix.

Block A_1 is defined by

$$A_1 = \begin{pmatrix} a & 0 & 0 & \dots & 0 & 0 & 0 \\ \lambda & b & \ddots & \ddots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & b & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda & c \end{pmatrix} \qquad \text{with} \qquad \begin{aligned} a &= -\delta - \mu_R\,, \\ b &= -\lambda - \delta - \mu_R\,, \\ c &= -\lambda - \delta\,. \end{aligned}$$

Block A_2 is defined by

$$A_2 = \begin{pmatrix} 0 & \mu_R & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \mu_R & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 & \mu_R & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & \mu_R \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$