



**HAL**  
open science

## Dynamic resource allocation in a multi-product make-to-stock production system

D. J. Hodge, K. D. Glazebrook

► **To cite this version:**

D. J. Hodge, K. D. Glazebrook. Dynamic resource allocation in a multi-product make-to-stock production system. *Queueing Systems*, 2011, pp.333-364. 10.1007/s11134-011-9217-2 . hal-00803937

**HAL Id: hal-00803937**

**<https://hal.science/hal-00803937>**

Submitted on 24 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

## Dynamic resource allocation in a multi-product make-to-stock production system

D. J. Hodge · K. D. Glazebrook

Received: date / Accepted: date

**Abstract** We consider optimal policies for a production facility in which several ( $K$ ) products are made to stock in order to satisfy exogenous demand for each. The *single machine* version of this problem in which the facility manufactures at most one product at a time to minimise inventory costs has been much studied. We achieve a major generalisation by formulating the production problem as one involving dynamic allocation of a key resource which drives the manufacture of all products under an assumption that each additional unit of resource allocated to a product achieves a diminishing return of increased production rate. A Lagrangian relaxation of the production problem induces a decomposition into  $K$  single product problems in which the production rate may be varied but is subject to charge. These reduced problems are of interest in their own right. Under mild conditions of full indexability the Lagrangian relaxation is solved by a production policy with simple index-like structure. This in turn suggests a natural index heuristic for the original production problem which performs strongly in a numerical study. The paper discusses the importance of full indexability and makes proposals for the construction of production policies involving resource idling when it fails.

**Keywords** backordering · dynamic programming · dynamic resource allocation · index heuristic · Lagrangian relaxation · make-to-stock policy · queueing control

---

D. J. Hodge  
Department of Mathematics and Statistics, Lancaster University, LA1 4YF, UK  
Tel.: +44-1524-593780  
E-mail: d.hodge@lancaster.ac.uk

K. D. Glazebrook  
Department of Management Science, Lancaster University, LA1 4YX, UK  
Department of Mathematics and Statistics, Lancaster University, LA1 4YF, UK  
E-mail: k.glazebrook@lancaster.ac.uk

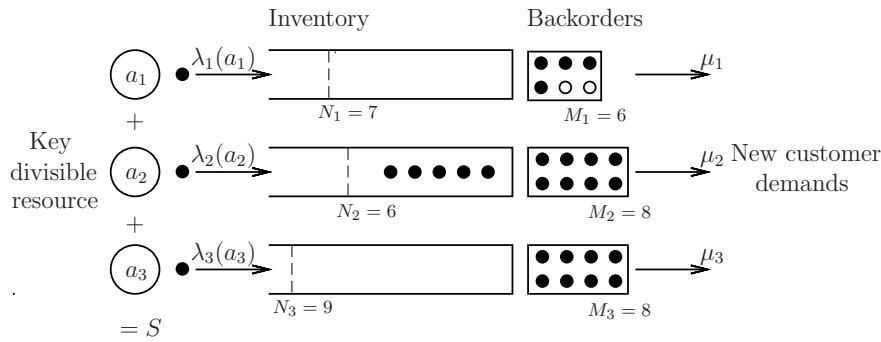


Fig. 1: An illustration of a three-product ( $K = 3$ ) queueing system currently in state  $(-2, 5, 0)$ . White circles can be seen as customers waiting for backorders to be fulfilled.

## 1 Introduction

The paper discusses the development of resource allocation policies for a facility which manufactures several ( $K$ ) products. Each product has an associated inventory and stored items are available to meet exogenous demand which is assumed to occur according to  $K$  independent Poisson streams. Limited storage space is available for each product, while orders arriving for a product which is out of stock may be backordered. A maximum number of backorders is allowed for each product. Any order arriving at a point in time at which the corresponding maximum backorder level is achieved results in a lost sale. Inventory-based costs are incurred for holding items in stock, for maintaining backorders and for lost sales.

A single key resource (manpower, equipment, money) is available to drive the production process. At each point in time, each item's inventory level (which will be negative in the case of backorders) is available to a central controller to inform decisions concerning how the key resource should be distributed among the  $K$  products to secure a production schedule to minimise the time average rate at which inventory costs are incurred overall. This may be viewed as a problem concerning the optimal control of arrivals to a multi-class queueing system. See Figure 1. To produce an account which is natural to the application, we shall use the language of inventory/production systems rather than that of queueing systems throughout.

The case in which the key resource is indivisible (the so-called *single machine problem*), namely where the facility manufactures at most one product at a time, is well studied. In our model, the allocation of each additional unit of resource to a product achieves a diminishing return in terms of production rate. Equivalently, to achieve successive equal increases of production rate requires escalating amounts of additional resource. An equivalent assumption of service costs which are convex in a service rate is made by Ata and Shnoerson

---

[1] in their analysis of an M/M/1 service system. This means that in our model it will rarely be optimal to allocate all of the key resource to a single product.

In the paper, we first introduce the full multi-product scenario. We then consider a relaxation which we can proceed to solve via a resulting product decomposition. The decomposed single product problem is that in which the resource may be applied to production at a variety of levels but is subject to a per unit charge. Typically, we expect that higher production levels are mandated when the inventory level and/or the resource charge are low. We seek to establish structural properties of such single product problems in order to inform the construction of strongly performing heuristics for the multi-product case of primary interest.

A range of contributions have been made to single machine versions of (variants of) our make-to-stock model. Unsurprisingly, the strongest results regarding the structure of optimal policies relate to the simplest models. Ha [6] followed by de Vericourt et al. [2] discuss the status of switching curve policies with associated hedging points for the single machine, two product case ( $K = 2$ ) with Markovian dynamics. Sobel [12] and Gavish and Graves [3] analyse a single machine, single product problem where the time to produce a single item is deterministic and where a start-up cost is incurred whenever the machine is switched on. Zheng and Zipkin [19] compare centralised and decentralised control policies for a single machine setup with  $K = 2$  and where the two products have identical cost and production characteristics. A corresponding multi-product study is given by Zipkin [20]. Perez and Zipkin [10] propose a myopic heuristic for a single machine model in which the products have identical cost characteristics and where the time to produce a single item has a general distribution. Wein [17] uses a heavy traffic approach for a single machine, multi-product problem with general stochastic structure for demands and production times. Veatch and Wein [15] compare a number of heuristic policies for single machine, multi-product problems, including the myopic policy proposed by Perez and Zipkin [10], an index heuristic based on Whittle's proposal [18] for restless bandits and one based on a heavy traffic analysis.

Menich and Serfozo [7] consider a very special multi-server, multi-product model in the symmetric case in which product characteristics are identical. They study a queueing problem with both customer routing available and flexible service placement. They show that it is optimal to concentrate service resources on the longest queues and to route new arrivals to the shortest queues. Also of relevance is the work of Sobel [13] who, in the context of a general queueing model with many possible service rates, shows that an assumption that the resulting return is convex in the effort expended leads inevitably to the conclusion that service policies which always choose maximal service rates will dominate.

The model for our multi-product problem is presented in Section 2. In order to construct strongly performing heuristics, we follow Whittle [18] and Glazebrook et al. [4] in developing a Lagrangian relaxation of the original problem in which a charge is levied for the deployment of the key resource. In this relaxation the Lagrange multiplier ( $W$ ) has an economic interpretation as

---

a charge for one unit of resource applied to production for one unit of time. As is demonstrated in Section 2, the Lagrangian relaxation yields a product-wise decomposition which consequentially focusses attention on single product problems with variable production rates and a charge for the resources used. The single product problems will be said to be *fully indexable* if it is the case that optimal production rates are decreasing in the resource charge for any inventory level. Section 2 discusses the importance of full indexability and the consequences of its failure. *Inter alia*, full indexability enables the construction of simply structured index heuristics for the multi-product problem. The above single product problems are studied in detail in Section 3, where a range of sufficient conditions for full indexability are elucidated. The paper concludes in Section 4 with an extensive numerical study.

In our numerical study our proposed index heuristic is applied to a range of problems which are fully indexable. The heuristic comfortably outperforms a range of competitors. The study illuminates the importance of the development of policies which respond (i.e. dynamically) to changes in inventory level and which are capable of dividing the resource between products in a way which balances their individual needs. We also address the issue of resource idling. In the bulk of our work we concern ourselves, not unreasonably, with models for which we do not have large surpluses of resource which we desire not to use for fear of over-stocking. Our indexability approach does, however, give rise to fair charges which can be used to derive heuristic policies which perform strongly in non-indexable cases with high holding costs which call for significant amounts of resource idling. This discussion of heuristic design in the wake of the failure of indexability is wholly novel to the index literature and could well have future benefits in other areas of application. Details of a numerical study are given in Subsection 4.2.

The class of heuristic policies developed in the paper can be viewed as a generalisation of the restless bandit proposal of Veatch and Wein [15] to our divisible resource model, by the deployment and development of the generalised index ideas of Glazebrook et al. [4]. The work additionally develops that of Veatch and Wein by allowing for backordering of demand and by the illumination of the connection between resource idling and the failure of indexability in our more general context. Further, the work of Niño-Mora [8], [9] on the development of marginal productivity indices, argues the importance of indexability, albeit in the much simpler context of restless bandit-type models with simple active-or-passive actions. Niño-Mora [9] follows Veatch and Wein [15] in proposing restless bandit models, *inter alia*, for single machine make-to-stock queues.

## 2 The Model

Demands for  $K$  products arise in  $K$  independent streams, with  $\mu_k$  the demand rate for product  $k$ ,  $1 \leq k \leq K$ . In order to meet these demands all products are made to stock. A finite resource is available to be divided between the

products for this purpose. Such resource allocations are made dynamically in response to changes in the stock levels of the products. Choice of a resource level  $a_k$  for product  $k$  from the set  $\{0, 1, \dots, S\}$  yields a production rate for product  $k$  of  $\lambda_k(a_k)$ . Each function  $\lambda_k$  is strictly positive, strictly increasing and strictly concave (i.e. the marginal rate of production is decreasing). The positivity assumption ensures the system is ergodic under all policies. The reader may think of the resource level  $a_k$  as the size of the workforce dedicated to production of product  $k$ .

Inventory costs are determined by the product specific parameters  $h_k$ ,  $b_k$  and  $D_k$  while the set of allowable inventory levels for product  $k$  is finite and is given by  $\Omega_k \equiv \{-M_k, -M_k + 1, \dots, N_k\}$ . A negative inventory corresponds to a number of backordered customers. These product  $k$  parameters have the following interpretations:

- $N_k$ : maximum inventory level;
- $M_k$ : maximum number of backordered customers;
- $h_k$ : holding cost rate per unit of time and per unit of inventory held;
- $b_k$ : cost rate per unit of time for each backordered customer;
- $D_k$ : penalty cost for each unit of demand which arises while the system has  $M_k$  backordered customers for product  $k$ .

We write

$$f_k(i) = h_k i^+ + b_k i^- + D_k \mu_k I(i = -M_k), \quad -M_k \leq i \leq N_k, \quad 1 \leq k \leq K, \quad (1)$$

for the instantaneous rate of inventory costs incurred by product  $k$  when in state  $i$ , where  $i^+ = \max(i, 0)$ ,  $i^- = \max(-i, 0)$  and  $I$  is an indicator. The system state  $\mathbf{i} \in \Omega_1 \times \Omega_2 \times \dots \times \Omega_K \equiv \Omega$  specifies every product's inventory level. The system state is observed at all times.

The inventory level of product  $k$  evolves under choice of action (resource level)  $a_k$  as follows: If the inventory level  $i \neq -M_k, N_k$  then under action  $a_k$  the sojourn time in state  $i$  is exponentially distributed with rate  $\mu_k + \lambda_k(a_k)$ . The next system state will be  $i + 1$  with probability  $\lambda_k(a_k) \{\mu_k + \lambda_k(a_k)\}^{-1}$  and  $i - 1$  otherwise. If  $i = N_k$ , the system spends an  $\exp(\mu_k)$  amount of time there before transitioning to  $N_k - 1$  while if  $i = -M_k$ , the system spends an  $\exp(\lambda_k(a_k))$  amount of time there before transitioning to  $-M_k + 1$ . For any given vector of actions, the levels of distinct products evolve independently.

We suppose that there is a limited resource  $S$  available at all times for the  $K$  products. An admissible action  $\mathbf{a} = \{a_1, a_2, \dots, a_K\}$  will specify a resource level for each product from the *admissible set*

$$\mathcal{A} = \left\{ \mathbf{a} : a_k \in \mathbb{N}, 1 \leq k \leq K \text{ and } \sum_k a_k \leq S \right\}. \quad (2)$$

A stationary admissible policy  $\boldsymbol{\pi}$  identifies an admissible action for each system state and is identified with a function from  $\Omega$  to  $\mathcal{A}$ . We wish to identify

production policies which minimise the average rate at which inventory costs, aggregated over all products, are incurred. We express the optimisation as

$$V^{\text{opt}} = \min_{\boldsymbol{\pi}} \sum_{k=1}^K E_{\boldsymbol{\pi}} [f_k(X_k)]. \quad (3)$$

In (3) the minimisation is over the set of stationary admissible policies while the  $k^{\text{th}}$  summand on the right hand side represents the average inventory cost per unit of time incurred by project  $k$  under policy  $\boldsymbol{\pi}$ . We use  $X_k$  for the project  $k$  inventory level with its stationary distribution under  $\boldsymbol{\pi}$ . Results for stochastic dynamic programming (DP) guarantee the existence of an optimal policy,  $\boldsymbol{\pi}^*$  say, which is stationary and which satisfies the DP optimality equations for the system (see, for example, Puterman [11]). In light of this, our restriction to policies from the stationary class is without loss of generality. We write  $\boldsymbol{\pi}(\mathbf{j})$  for the action taken by (stationary) policy  $\boldsymbol{\pi}$  in state  $\mathbf{j}$ .

*Remark 1* A special case of our general make-to-stock model is the *lost sales model* where  $b_k = M_k = 0$  and the inventory cost rate for product  $k$  in state  $i$  is given by

$$f_k(i) = h_k i^+ + D_k \mu_k I(i=0), \quad 0 \leq i \leq N_k.$$

In this model there is no backordering and demand arising when there is no stock results in lost sales, with a penalty cost  $D_k$  incurred for each.

*Remark 2* It is worthy of note here that the scope of the model we shall solve is not restricted in any practical sense by the limitations to integral resource levels. We can subdivide the resources into any desired number of pieces and this is only at the expense of computation time in our later algorithms. We also note that the assumption of strict concavity for the production rate functions is made to simplify and clarify the account. The ‘strictness’ may be dropped at little cost. Furthermore, it is, at least intuitively, clear that even for non-concave  $\lambda$  the Markovian dynamics of the problem imply that the concave hull of  $\lambda(a)$ ,  $a = 0, \dots, S$  can be used.

## 2.1 Indexability and index heuristics

While in principle it is true that (3) could be solved by direct application of the methods of stochastic dynamic programming (DP), in practice this is computationally infeasible other than for very small problems (crucially, small  $K$ ). Hence we seek to develop strongly performing *heuristic policies* which are close to cost minimising. To do this we develop a Lagrangian relaxation of (3) which (a) extends the policy class to functions from  $\boldsymbol{\Omega}$  to  $\{0, 1, \dots, S\}^K$  and (b) penalises departures from the resource constraint in (2). We write

$$V(W) = \min_{\boldsymbol{\pi}} \left\{ \sum_{k=1}^K E_{\boldsymbol{\pi}} [f_k(X_k) + W \pi_k(X_k)] - WS \right\}. \quad (4)$$

In (4) the minimisation is now over policies which can choose resource at any level in the range  $\{0, 1, \dots, S\}$  for each product. The quantity  $E_{\pi} [\pi_k(X_k)]$  is the average resource applied to product  $k$  per unit of time under policy  $\pi$ . The parameter  $W$  is a Lagrange multiplier which has an interpretation as a charge levied for each unit of resource per unit of time. This relaxation is due to Whittle [18]. It is the Lagrangian arising from a relaxation of (3) in which the *average* resource applied to the  $K$  products per unit of time is constrained to be no greater than  $S$ . Since in the Lagrangian relaxation in (4) we have eliminated the resource constraint in (2), and since the objective is additive across products, it is clear that we have available a product-wise decomposition of the optimisation problem. As a result of this decomposition an optimal policy for (4) will have the form  $\pi(W) = \{\pi_1(W), \pi_2(W), \dots, \pi_K(W)\}$  where  $\pi_k(W)$  is an optimal policy for a single product problem in which a variable production rate is applied to product  $k$  and where the instantaneous cost rate is given by

$$f_k(i) + Wa_k \tag{5}$$

when  $i$  is the inventory level of product  $k$  and  $a_k$  is the resource level applied to it. Such single product problems are discussed at greater length in Section 3.

*Remark 3* While our prime focus is on problem (3), the unconstrained problem (4) in which production capability can be secured for a per unit charge at rate  $W$  is of independent interest.

After this brief introduction of the Lagrange multiplier  $W$  and its natural *charge* interpretation we now sidestep its actual consideration via a simple variable transformation. We multiply each  $f_k$  by a constant  $c = W^{-1}$  and consider minimising the aggregated costs  $cf_k(i) + a_k$  across the  $K$  products. We write

$$\bar{V}(c) = \min_{\pi} \left\{ \sum_{k=1}^K E_{\pi} [cf_k(X_k) + \pi_k(X_k)] - S \right\}. \tag{6}$$

The problems of determining optimal policies in (3) and (6) are plainly equivalent, with the problem values being directly related by the equation  $\bar{V}(c) = cV(W/c)$ .

*Remark 4* We have found that our analyses are more easily accomplished via optimisation problem (6) than via direct use of the Lagrangian relaxation (4). The use of  $c$  in place of  $W$  is well suited for proofs and calculations but, as we shall see, the equivalent problems formulated with variable  $W$  lend themselves much more directly to economic interpretation and heuristic understanding.

Write  $\bar{P}_k(c)$  for the single product problem in which resource levels from the range  $\{0, 1, \dots, S\}$  are chosen for *product  $k$  alone* to minimise a combination of inventory costs ( $cf_k$ ) and production costs, the latter being levied at rate 1. Problem  $\bar{P}_k(c)$  may be expressed as

$$\bar{V}_k(c) = \min_{\pi_k} \{E_{\pi_k} [cf_k(X_k) + \pi_k(X_k)]\}. \tag{7}$$



We now describe what needs to be true of solutions to the  $\bar{P}_k(c)$ ,  $c \in \mathbb{R}^+$ ,  $1 \leq k \leq K$ , to enable us to develop effective solutions to the resource constrained production problem in (3).

**Definition 1 (Full Indexability)** *Product  $k$  is fully indexable if there exists a family of stationary policies  $\bar{\pi}_k(c) : \Omega_k \rightarrow \{0, 1, \dots, S\}$ ,  $c \in \mathbb{R}^+$ , such that*

(a)  $\bar{\pi}_k(c)$  is optimal for  $\bar{P}_k(c) \forall c \in \mathbb{R}^+$ ;

and

(b)  $\bar{\pi}_k(c)$  is increasing componentwise in  $c$ .

An appropriate form of index for this formulation is as follows:

**Definition 2 (Product Indices)** *If product  $k$  is fully indexable with associated optimal policies  $\bar{\pi}_k(c)$ ,  $c \in \mathbb{R}^+$ , its  $c$ -index function*

$$c_k : \{0, 1, \dots, S - 1\} \times \Omega_k \rightarrow \mathbb{R}^+ \cup \{\infty\}$$

is given by

$$c_k(a, i) = \inf \{c : \bar{\pi}_k(c, i) \geq a + 1\}, \text{ or equivalently} \quad (8)$$

$$= \sup \{c : \bar{\pi}_k(c, i) \leq a\}. \quad (9)$$

Non-indexability occurs if and only if these inf and sup definitions differ. We discuss how this difference is manifested in Subsections 3.2 and 4.2. In the former we show that, under given conditions, preventing a discrepancy for  $i = N$  suffices to ensure full indexability.

The following result is a straightforward consequence of the above discussion. Its proof is omitted.

**Lemma 1** *Suppose that all  $K$  products are fully indexable with  $c$ -index functions  $c_k$ ,  $1 \leq k \leq K$ . The policy  $\bar{\pi}(c)$  for optimisation problem (6) such that*

$$\bar{\pi}(c, \mathbf{i}) = \mathbf{a} \iff c_k(a_k - 1, i_k) < c \leq c_k(a_k, i_k), \quad 1 \leq k \leq K, \quad \mathbf{i} \in \Omega \quad (10)$$

achieves  $\bar{V}(c)$ ,  $\forall c \in \mathbb{R}^+$ .

Further, under full indexability we define equivalent  $W$ -indices by  $W_k(a, i) = c_k(a, i)^{-1}$ . This  $W_k(a, i)$  has the interpretation as the maximal value of the resource charge  $W$  at which using action  $a + 1$  in state  $i$  is optimal. It has a natural interpretation as a *fair charge* for raising the resource level available to product  $k$  above level  $a$  when its inventory level is  $i$ .

For intuition and for the discussion in the remainder of this section we return to  $W$  ( $= c^{-1}$ ). The  $W$  equivalent of (10) tells us that under full indexability an optimal solution to the Lagrangian relaxation (4) may be characterised as follows: in each state  $\mathbf{i}$ , increase the amount of resource available to each product  $k$  until the point is reached when the fair charge for adding further resource (the  $W$ -index  $W_k(a_k, i_k)$ ) falls below the actual (or prevailing) charge  $W$ . This construction is strongly suggestive of a natural greedy index heuristic  $\pi_W$ , say, for (3) when all products are fully indexable.

The *greedy index heuristic*,  $\pi_W$ , allocates resources in system state  $\mathbf{i}$  as follows:

**Step 1** Start with initial allocation  $\mathbf{0} = \{0, 0, \dots, 0\}$ . Suppose the current allocation is  $\mathbf{a} = \{a_1, a_2, \dots, a_K\}$  with  $\sum_k a_k < S$ .

**Step 2** Choose any  $k$  satisfying

$$W_k(a_k, i_k) = \max_{1 \leq j \leq K} W_j(a_j, i_j). \quad (11)$$

**Step 3** Increment current allocation  $\mathbf{a}$  by one in the  $k$ -th component. So  $a_k^{\text{new}} = a_k + 1$  and  $a_j^{\text{new}} = a_j$  for all  $j \neq k$ .

**Step 4** If  $\sum_k a_k^{\text{new}} < S$  return to Step 1. Otherwise to to Step 5.

**Step 5** We declare  $\mathbf{a}$  to be the action dictated by  $\pi_W$  now that all  $S$  units of resource have been allocated.

In words, in each state  $\mathbf{i}$  the greedy index heuristic  $\pi_W$  increases the amount of resource available to the  $K$  products in decreasing order of their  $W$ -indices  $W_k(\cdot, i_k)$  until the total resource allocated reaches  $S$ .

## 2.2 The greedy index heuristic and the importance of full indexability

If we write

$$V^* = \max_{W \geq 0} V(W) \quad (12)$$

then  $V^*$  is an easily computed lower bound on  $V^{\text{opt}}$ . Write  $W^*$  for the maximising  $W$  in (12). In a context much simpler than the current one, Glazebrook et al. [5] discuss how the nature of solutions to (4) and (12) can shed light on the performance of a greedy index heuristic. Should it be the case that, under full indexability, the policy  $\pi(W^*)$  which achieves  $V^*$  be such that, the system states which require that  $\pi(W^*)$  take an inadmissible action have small probability, in equilibrium, then  $V^*$  will be close to  $V^{\text{opt}}$  and  $\pi_W$  will be close to optimal. However, as Glazebrook et al. [5] indicate, this approach is conservative and  $\pi_W$  can perform strongly even when this is not the case.

Asymptotic optimality of greedy index policies for restless bandits is known to hold under given conditions. See Weber and Weiss [16]. Their result applies to fully indexable versions of our model with each  $\lambda_k(a)$  constant for  $a \geq 1$ . The limit concerned allows the number of products  $K$  and the maximal resource  $S$  to diverge to infinity such that  $S/K \rightarrow \beta < 1$ .

We now give an example which illustrates the importance of full indexability.

*Example 1* Consider a two product problem ( $K = 2$ ). Both products have  $M = N = 10$ ,  $h = 0.025$ ,  $b = 1.5$ , and  $D = 200$  and the total resource available is  $S = 25$ . Product 1 has demand rate  $\mu_1 = 1.576$ , a high production rate given by  $\lambda_1(a) = 4.5a(a + 5.971)^{-1}$  and is *not* fully indexable. For full

indexability we would need *inter alia* that  $\pi_1(W, 9)$  is monotone decreasing in  $W$ . To see that this is not so, we observe that for  $W > 200$ ,  $\pi_1(W)$  is the zero policy and yet when  $W = 100$  we have  $\pi_1(W, 9) = 1$ . Further, we also have  $\pi_1(W, 9) = 0$ ,  $W \leq 0.056$ . Hence indices are not defined, but there are two natural values for a fair charge  $\widetilde{W}_1(0, 9)$ , namely a  $W$ -value at which optimal policy  $\pi_1(W)$  is indifferent between  $a = 0$  and  $a = 1$  in state 9. One fair charge is close to 0.056 while the other is around 140.

Product 2 has demand rate  $\mu_2 = 1.046$  and production rate given by  $\lambda_2(a) = 1.5a(a + 5.971)^{-1}$ . The latter is considerably lower than for product 1 but is more appropriate to meet the product 2 demand. Product 2 is fully indexable and its indices are well defined. Now consider the following heuristics for the two product problem: Take as Heuristic 1 (H1) a version of the greedy index heuristic  $\pi_W$  in which we always take the index for product 1 to be the largest available fair charge and as H2 a version in which the index for product 1 is always taken to be the smallest available fair charge. Direct application of DP to this example yields  $V^{\text{opt}} = 4.333$  while the cost rates associated with H1 and H2 are respectively 4.370 (0.86% suboptimal) and 4.926 (13.7% suboptimal) and hence H1 clearly outperforms H2. If we now consider a modified version of product 2 in which  $\mu_2$  is significantly reduced then the position is reversed and H2 outperforms H1 (see later Table 4).

One way of understanding this is via the value of  $W^*$ , the maximiser in (12). When  $\mu_2$  is large, so is  $W^*$  and the high fair charge heuristic H1 may be regarded as an appropriate admissible approximation to inadmissible  $\pi(W^*)$  with strong performance. When  $\mu_2$  is small, so is  $W^*$  and now the low fair charge heuristic H2 is closer to  $\pi(W^*)$ . Since when full indexability fails, which fair charge to use in the construction of effective heuristic policies can depend upon the other products, it is clear that failures of full indexability are a serious issue.

In addition to the issues addressed by Example 1, full indexability also yields indices which are monotone in the resource level. This means that greedy heuristic  $\pi_W$  is trivial to implement and has a strong intuitive grounding. The greedy heuristic provides a marked reduction in complexity compared to an optimal brute force DP solution approach. A straightforward analysis of our algorithm presented later, which calculates  $\pi_W$ , yields that the number of calculations used in obtaining this policy is  $O(KS(N+M)^4)$ . Importantly, it scales linearly in both the number of products and the number of actions in each state. It is the indexable decomposition of products which is key to linear scaling in products. The natural LP formulation of (3) results in a formulation which grows with the state-space size,  $N^K$ , with  $(S+1)N^K$  variables and  $O(N^K)$  constraints. Even in comparison with other very simple heuristics we find the index approach superior. A brute force approach to obtaining the best myopic allocation (for an exact description see Section 4) makes at least  $\binom{S-1}{K}$  checks and hence its requirements are  $O(K\binom{S-1}{K})$ . Finding the best static allocation needs to check amongst  $O(\binom{S-1}{K})$  policies and also to evaluate their performances.

### 3 The single product problem $\bar{P}(c)$ and sufficient conditions for full indexability

Following the approach sketched out in Section 2, we now proceed to study the single product problems  $\bar{P}_k(c)$ ,  $c \in \mathbb{R}^+$ ,  $1 \leq k \leq K$ . Since our focus now shifts to individual products, we can drop the product identifier  $k$  from the notation for the remainder of the section.

#### 3.1 DP formulation of $\bar{P}(c)$

It is straightforward to show that the DP optimality equations for the single-product make-to-stock model  $\bar{P}(c)$  take the form given in (13) below. In (13),  $\gamma(c)$  is the *optimal average cost rate* per unit of time for  $\bar{P}(c)$  and  $\delta(c, i) : \{-M, -M+1, \dots, N-1, N\} \rightarrow \mathbb{R}$  is the associated *bias function* which measures the transient effect of starting the system (assumed to be evolving under an optimal policy) in state  $i$ . We have

$$\mu \{ \delta(c, i) - \delta(c, i-1) \} I(i \neq -M) = -\gamma(c) + cf(i) + \min_a [a + \lambda(a) \{ \delta(c, i+1) - \delta(c, i) \} I(i \neq N)], \quad -M \leq i \leq N. \quad (13)$$

The minimisation in (13) is over  $a \in \{0, 1, \dots, S\}$ . If we denote by  $V(c, i, t)$  the minimal  $t$ -horizon cost for  $\bar{P}(c)$  with initial state  $i$  then  $V(c, i, t) \sim \gamma(c)t + \delta(c, i)$ . As is plain from (13), successive differences in the function  $\delta(c, i)$  play a crucial role. We write

$$\Delta(c, i) = \delta(c, i) - \delta(c, i-1),$$

The ergodicity of our inventory level process under all policies guarantees that  $\delta(c, i)$  and  $\Delta(c, i)$  are well-defined and finite (see Remark 5). We re-express (13) as

$$\mu \Delta(c, i) I(i \neq -M) = -\gamma(c) + cf(i) + \min_a \{ a + \lambda(a) \Delta(c, i+1) I(i \neq N) \}, \quad -M \leq i \leq N. \quad (14)$$

For any given policy  $\pi$ , we have an associated cost rate  $\gamma^\pi(c)$ , a bias function  $\delta^\pi(c, i)$  and a differenced bias  $\Delta^\pi(c, i)$ , defined in the natural analogous way.

Our system's ergodicity guarantees that we can develop tractable expressions for the key quantities  $\Delta(c, i)$  and  $\Delta^\pi(c, i)$ . Write  $\mathbf{p}^c \equiv \{p^c(j), -M \leq j \leq N\}$  and  $\mathbf{p}^\pi \equiv \{p^\pi(j), -M \leq j \leq N\}$  for the inventory level's stationary distribution under, respectively, some optimal policy  $\bar{\pi}(c)$  for  $\bar{P}(c)$  and some stationary policy  $\pi$ . We note in passing that, in the event of multiple optimal policies for  $\bar{P}(c)$ , no part of our analysis is dependent on which one we choose. We further write  $\mathbf{p}_i^c \equiv \{p_i^c(j), -M \leq j \leq N\}$  and  $\mathbf{p}_i^\pi \equiv \{p_i^\pi(j), -M \leq j \leq N\}$  for the modified stationary distributions obtained for the inventory level process starting in state  $i$  when an instantly reflecting barrier is placed just below  $i$ , blocking transitions to state  $i-1$ . These are equivalent to the stationary

distributions conditioned upon the product state being  $i$  or greater. The birth-death nature of our models means that closed forms are easily available for the above distributions. It is trivial that if  $i > -M$ , we have

$$p_i^c(j) > p^c(j) \text{ and } p_i^\pi(j) > p^\pi(j) \text{ for } j \geq i, \text{ and } p_i^c(j) = p_i^\pi(j) = 0 \text{ for } j < i. \quad (15)$$

We also write  $\tau^c(i)$  for the expected time taken, under optimal policy  $\bar{\pi}(c)$ , between entry into state  $i$  and the first subsequent entry into state  $i - 1$ . We use  $\tau^\pi(i)$  for the corresponding quantity when the system evolves under policy  $\pi$ . Standard theory (see, for example, Tijms [14]) yields the expressions

$$\Delta(c, i) = \tau^c(i) \sum_{j=-M}^N \{p_i^c(j) - p^c(j)\} \{cf(j) + \pi^*(j)\}, \quad -M + 1 \leq i \leq N, \quad (16)$$

and

$$\Delta^\pi(c, i) = \tau^\pi(i) \sum_{j=-M}^N \{p_i^\pi(j) - p^\pi(j)\} \{cf(j) + \pi(j)\}, \quad -M + 1 \leq i \leq N. \quad (17)$$

To understand (17) we observe that

$$\delta^\pi(c, i) = C_i^\pi(c) - \gamma^\pi(c) \sum_{j=1}^i \tau^\pi(j)$$

where  $C_i^\pi(c)$  is the expected cost incurred by  $\bar{P}(c)$  when policy  $\pi$  is applied from initial state  $i$  up to the state's first entry into state 0. It then follows that

$$\Delta^\pi(c, i) = C_i^\pi(c) - C_{i-1}^\pi(c) - \gamma^\pi(c) \tau^\pi(i) \quad (18)$$

and the r.h.s. of (17) re-expresses the r.h.s. of (18) in terms of the distributions  $\mathbf{p}^\pi$  and  $\mathbf{p}_i^\pi$ . The development which yields (16) is similar, with optimal policy  $\bar{\pi}(c)$  replacing  $\pi$  throughout. Some intuition for these expressions can be gained by seeing each sum as the difference between the costs incurred going from state  $i$  to  $i-1$  under the policy considered, and the costs incurred in equilibrium over an identical time period.

Plainly,  $\Delta(c, i) = \Delta^{\bar{\pi}(c)}(c, i)$ ,  $-M + 1 \leq i \leq N$ . It is also plain from the DP equations (14) that if  $\bar{\pi}_1(c)$  and  $\bar{\pi}_2(c)$  are distinct optimal policies for  $\bar{P}(c)$  then  $\Delta(c, i) = \Delta^{\bar{\pi}_1(c)}(c, i) = \Delta^{\bar{\pi}_2(c)}(c, i)$ ,  $-M + 1 \leq i \leq N$ .

The following results are straightforward to establish from the above definitions and the relations (15). They concern the behaviour of the key quantities

as specified parameter values vary, while holding all non-specified parameters fixed. A sketch proof is provided. Full details are available from the authors.

**Proposition 1**

- (a) Minimised cost rate  $\gamma(c)$  is
  - (i) continuous and piecewise linear in each of  $h, b, D$  and  $c$ ;
  - (ii) increasing and concave in  $c$ ;
- (b) For all  $-M \leq i \leq N$  and all stationary policies  $\pi$ ,  $\Delta^\pi(c, i)$  is
  - (i) affine in  $h, b, D$  and  $c$ ;
  - (ii) increasing in  $h$ ;
  - (iii) decreasing in  $b$  and  $D$ ;
- (c) For all  $-M \leq i \leq N$ ,  $\Delta(c, i)$  is continuous and piecewise linear in  $h, b, D$ , and  $c$ .

*Sketch Proof:* For part (a)(i),  $\gamma(c)$  can be seen as the minimum over a finite collection of stationary policies of  $F^\pi + A^\pi$ , the sum of the average inventory costs and resource costs under  $\pi$ . For any fixed policy  $\pi$ ,  $F^\pi$  and  $A^\pi$  are both continuous and indeed affine in  $h, b, D$  and  $c$ . Part (a)(ii) follows from the fact that  $\gamma(c)$  is a minimum of functions, each of which is increasing and affine in  $c$ . Parts (b) and (c) follow from these arguments using (14) to define  $\Delta(c, i)$  inductively for  $i$  decreasing from initial case  $i = N$ . We observe that  $\Delta(c, N)$  trivially has the required properties.

We conclude this section with the following characterisations of optimal policies which are direct consequences of the DP optimality equations (14). Before proceeding, please note that the policy  $\pi$  which chooses minimizing actions in each state from (14) is an optimal policy. Naturally this characterisation is only helpful once the values of the  $\Delta(c, i)$  are known.

**Proposition 2** *A stationary policy  $\pi$  is optimal for  $\bar{P}(c)$  if and only if  $\pi(N) = 0$  and*

$$\Delta^\pi(c, i+1) \{ \lambda(\pi(i)) - \lambda(\pi(i) + 1) \} \leq 1 \leq \Delta^\pi(c, i+1) \{ \lambda(\pi(i) - 1) - \lambda(\pi(i)) \},$$

$$-M \leq i \leq N - 1, \quad (19)$$

where  $\lambda(S + 1) \equiv \lambda(S)$  and  $\lambda(-1) \equiv -\infty$  in (19).

*Sketch Proof:* This follows since firstly  $\Delta^\pi$  as defined above satisfies the DP equations (14) with the  $\pi(i)$  replacing the choices in the minimisations. Then secondly, the concavity of  $\lambda$  ensures that satisfying (19) means that  $\pi(i)$  will indeed be the arg min in (14).

*Remark 5* The assumption we shall make that all production rates are positive, and hence that  $\lambda(0) > 0$  is worthy of comment at this point. It may be that  $\lambda(0)$  represents a minimal production rate which is permanently maintained and that the application of additional resource raises production rate

above this minimal level. That said, our primary motivation for the assumption is presentational. When  $\lambda(0) > 0$  the inventory level process is ergodic under all stationary production policies. Most importantly the process becomes irreducible on  $[-M, N]$ , so that extra notation to deal with non-uniqueness of  $\Delta(i)$  on states never visited does not complicate matters. This fact significantly simplifies the discussion at key points. However, problems with  $\lambda(0) = 0$  pose no serious difficulties. If we consider an approximating production rate function  $\lambda_\epsilon(a)$ ,  $a \in \{0, 1, \dots, S\}$  such that  $\lambda_\epsilon(0) = \epsilon < 2\lambda(1) - \lambda(2)$  and  $\lambda_\epsilon(a) = \lambda(a)$ ,  $a \neq 0$ , then  $\lambda_\epsilon$  will be strictly increasing and strictly concave whenever  $\lambda$  is. Since, moreover, for the problems we consider, the inventory process behaves well under the limit  $\epsilon \rightarrow 0$  it is easy to conclude that our main results hold when  $\lambda(0) = 0$ .

### 3.2 Sufficient conditions for full indexability

We now explore the issue of full indexability which is crucial for consideration of the multi-product resource allocation problem. Recall from Definition 1 that full indexability is the requirement that there exist optimal policies for  $\bar{P}(c)$  which are monotone increasing in the inventory cost multiplier  $c$ . We begin our discussion with two examples.

*Example 2* Consider a product with  $N = M = 10$ ,  $S = 25$ ,  $h = 0.05$ ,  $b = 1.5$ ,  $D = 50$ ,  $\mu = 0.65$ . The production rate model is given by

$$\lambda(a) = \lambda a(a + m)^{-1} + \epsilon,$$

where  $\lambda = 1$ ,  $m = 0.8$  and  $\epsilon = 0.05$ . The unique optimal stationary policy for  $\bar{P}(c)$  is computed at four values of  $c$  as follows:

$j$	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	
$\pi^*(j)$	25	25	25	25	25	25	23	20	18	16	12	10	8	6	5	4	3	2	1	1	0	$c = 26.4644$
$\pi^*(j)$	25	25	25	25	25	25	23	21	18	16	12	10	8	6	5	4	3	2	1	1	0	$c = 26.6114$
$\pi^*(j)$	25	25	25	25	25	25	23	21	18	16	12	10	8	6	5	4	3	2	1	0	0	$c = 26.7760$
$\pi^*(j)$	25	25	25	25	25	25	23	21	18	16	13	10	8	6	5	4	3	2	1	0	0	$c = 27.0362$

Please note that the action taken in state 9 decreases from 1 to 0 when  $c$  increases from 26.6114 to 26.7760, thus the product is non-indexable. From computations it is possible to verify that the right gradient  $\partial_c^+ \Delta(c, 9)$  is positive when  $c = 26.7760$ . As we shall see, the requirement that all the  $\Delta(c, i)$  are decreasing in  $c$  is a sufficient condition for full indexability.

It could be conjectured from the above example that the product's non-indexability is driven by the truncated state space. In response to that, we give a further example of non-indexability for a product with  $M = N = \infty$ .

*Example 3* Consider a product with  $M = N = \infty$ , with inventory costs given by  $f(i) = 10i^- + 0.05i^+$  and other parameters given by  $\lambda(0) = 0$ ,  $\lambda(1) = 1$ ,  $\lambda(2) = 1.75$ ,  $\mu = 0.6$ . We make the natural demand that we only consider stable policies.

It is straightforward that all policies of the form  $\pi_m(i) = I(i \leq m)$  minimise the average rate at which resource is used in the class of stationary stable policies. It will simplify our argument from this point if we revert to  $W = c^{-1}$  and the instantaneous cost rate  $f(i) + Wa$ , as in (5). We then have that the average cost incurred under  $\pi_m$  takes the form  $A_m + 0.6W$  for some constant  $A_m$ . We have shown numerically that  $\min_m A_m = A_9 = 0.5162$  and hence it must follow that  $\pi_9$  is optimal for the single product problem  $P(W)$  for  $W$  large enough.

We now consider the performance of policies defined by  $\pi_{m,2} = 2I(i \leq m)$ . Calculations yield that the average cost rate incurred under  $\pi_{3,2}$  is  $0.2464 + 0.6857W$  and hence that the optimal cost rate is bounded above by 0.3 when  $W < 0.07$ . It must then be true that when  $W < 0.07$  it can never be optimal to reach inventory level 7 where the inventory cost rate is 0.35. It must then follow that, for example,  $\pi(0.05, 6) = 0$ , namely that it is optimal to allocate zero resource in state 6 when  $W = 0.05$ . However, we have already seen that, for large enough  $W$ ,  $\pi(W, 6) = \pi_9(6) = 1$ . Hence the product fails to be fully indexable.

In the next result we present a sufficient condition for full indexability. This condition fails to hold in Example 2. Note that the constructive nature of the proof of Theorem 1 will be exploited in the development of the algorithms presented in Section 3.3.

**Theorem 1** *If  $\Delta(c, i)$  is strictly decreasing in  $c$  for all  $i$  in the range  $-M \leq i \leq N$  then there exists a finite collection of intervals  $[C_n, C_{n+1})$ ,  $0 \leq n \leq L$ , say, with  $C_0 = 0$  and  $C_{L+1} = \infty$  and stationary policies  $\pi_n$ ,  $0 \leq n \leq L$ , such that*

- (a)  $\pi_n$  is optimal for  $\bar{P}(c)$  over the range  $c \in [C_n, C_{n+1})$ ,  $0 \leq n \leq L$ ,
- and
- (b)  $\pi_{n+1}(i) \geq \pi_n(i)$ , for all  $-M \leq i \leq N$ ,  $0 \leq n \leq L$ .

*It follows that under these conditions the product is fully indexable.*

*Proof* Note from Proposition 1 that all of the  $\Delta(c, i)$  are continuous and piecewise linear in  $c$ . Hence the hypothesis of the theorem implies that the right gradients  $\partial_c^+ \Delta(c, i)$  must be (strictly) negative for all choices of  $c$  and  $i$ . Recall the optimality conditions of (19), as these are pivotal in what follows.

We use  $\Pi(c)$  to denote the set of stationary policies which are optimal at  $c$ . We have  $\Delta(c, i) = \Delta^\pi(c, i)$ ,  $-M \leq i \leq N$ ,  $\pi \in \Pi(c)$ . Fix  $c = c_1$  and define

$$\theta(c_1) = \min_{\pi \notin \Pi(c_1)} \max_{-M \leq i < N} \left( \left| 1 - \Delta^\pi(c_1, i+1) \{ \lambda(\pi(i)) - \lambda(\pi(i)+1) \} \right| \right. \\ \left. \wedge \left| 1 - \Delta^\pi(c_1, i+1) \{ \lambda(\pi(i)) - 1 - \lambda(\pi(i)) \} \right| \right) > 0$$

and

$$\xi = \max_{c, i, \pi} \left\{ \left| \partial_c \Delta^\pi(c, i) \right| \right\}. \quad (20)$$



Note that, since from Proposition 1, for each  $i$ ,  $\pi$  the quantity  $\Delta^\pi(c, i)$  is affine in  $c$ , the maximisation in (20) is over a finite number of elements. Hence  $\xi < \infty$ . If we now write  $\epsilon(c_1) = \theta(c_1) [2\xi \{\lambda(1) - \lambda(0)\}]^{-1}$  then by construction it is straightforward that no  $\pi \notin \Pi(c_1)$  can satisfy (19) within the range  $c_1 \leq c \leq c_1 + \epsilon(c_1)$ . Hence for  $c \in [c_1, c_1 + \epsilon(c_1))$  optimal stationary policies for  $\bar{P}(c)$  must be members of  $\Pi(c_1)$ .

Now suppose that  $|\Pi(c_1)| > 1$ . The strictly decreasing nature of  $\Delta(c, i+1)$  in  $c$  allows us to deduce that any policy  $\pi \in \Pi(c_1)$  for which the left hand side of (19) is satisfied with equality for some  $i$  at  $c = c_1$ , namely

$$\Delta^\pi(c_1, i+1) \{\lambda(\pi(i)) - \lambda(\pi(i) + 1)\} = \Delta(c_1, i+1) \{\lambda(\pi(i)) - \lambda(\pi(i) + 1)\} = 1, \quad (21)$$

cannot itself be optimal for any  $c > c_1$ . It must follow that any policy  $\pi \in \Pi(c_1)$  which is optimal for some  $c \in (c_1, c_1 + \epsilon(c_1))$  must satisfy, for each  $-M \leq i \leq N-1$ ,

$$\Delta^\pi(c_1, i+1) \{\lambda(\pi(i)) - \lambda(\pi(i) + 1)\} < 1 \leq \Delta^\pi(c_1, i+1) \{\lambda(\pi(i) - 1) - \lambda(\pi(i))\}. \quad (22)$$

However, the strict concavity of  $\lambda(\cdot)$  means that only one member of  $\Pi(c_1)$ ,  $\pi_1$  say, can satisfy (22). Moreover, it must be that

$$\pi_1(i) \geq \pi(i), \text{ for all } \pi \in \Pi(c_1), \text{ for all } -M \leq i \leq N, \quad (23)$$

and hence  $\pi_1$  is maximal in  $\Pi(c_1)$ .

We now define

$$c_2 = \inf \{c > c_1 : \pi_1 \notin \Pi(c)\}.$$

We infer from the continuity of the  $\Delta(c, i)$  (given in Proposition 1(c)) and from (19) that  $\pi_1 \in \Pi(c_2)$  but that

$$\begin{aligned} \Delta^{\pi_1}(c_2, i+1) \{\lambda(\pi_1(i)) - \lambda(\pi_1(i) + 1)\} &= \Delta(c_2, i+1) \{\lambda(\pi_1(i)) - \lambda(\pi_1(i) + 1)\} \\ &= 1 \end{aligned} \quad (24)$$

for some  $i$ .

We now repeat the above argument, replacing  $c_1$  with  $c_2$ . There will be some interval  $[c_2, c_3)$  for which the maximal member of  $\Pi(c_2)$  will be optimal. Call this policy  $\pi_2$ . Since  $\pi_1 \in \Pi(c_2)$  it must be that  $\pi_1 \neq \pi_2$  and

$$\pi_2(i) \geq \pi_1(i), \quad -M \leq i \leq N. \quad (25)$$

To summarise, we note that  $\pi_1$  is optimal for  $c \in [c_1, c_2)$  while  $\pi_2$  is optimal for  $c \in [c_2, c_3)$ . Further, the two policies satisfy (25).

The proof is completed by a continuation of this argument together with the observations (i) that  $c_1$  was chosen arbitrarily, and (ii) that the number of distinct stationary policies is finite.  $\square$

The following result is an immediate consequence.

**Corollary 1** *For fixed values of the remaining system parameters  $\exists \bar{h}$  such that  $h \leq \bar{h}$  implies that the product is fully indexable.*

*Proof* Fix  $h = 0$ . From (17) we infer that, for any choice of  $i$  and  $\pi$ ,

$$\{\tau^\pi(i)\}^{-1} \partial_c \Delta^\pi(c, i) = \sum_{j=-M}^N \{p_i^\pi(j) - p^\pi(j)\} \{bj^- + D\mu I(j=-M)\}. \quad (26)$$

But the quantity  $bj^- + D\mu I(j=-M)$  is decreasing over the range  $-M \leq j \leq N$  and hence it is straightforward to conclude from (15) that  $\partial_c \Delta^\pi(c, i) < 0$ . It must then follow that, for any  $i$  in the range  $-M \leq i \leq N$ ,  $\Delta(c, i)$  has a negative gradient everywhere and so is strictly decreasing in  $c$ . Full indexability when  $h = 0$  follows from Theorem 1. It is straightforward to establish that full indexability extends to a positive range for  $h$ . This concludes the proof.  $\square$

Given that full indexability is *not* universal it is natural to develop a measure of the *extent* to which we have it. In light of Theorem 1, one such measure could be the first  $c$ -value above zero at which some  $\Delta(c, i)$  fails to be strictly decreasing. Hence we write

$$\bar{c} = \inf \{c \geq 0 : \partial_c^+ \Delta(c, i) \geq 0 \text{ for some } -M \leq i \leq N\}. \quad (27)$$

Note that  $\Delta(c, -M) = -cD$  and hence is always decreasing for  $D > 0$ . It will be useful to register the dependence of  $\bar{c}$  on system parameters in the notation. Hence the conclusion of the proof of Corollary 1 may be expressed as  $\bar{c}(h) = \infty, 0 \leq h \leq \bar{h}$ .

One critically important fact is that  $\bar{c}$  is bounded away from zero for reasonable models. Proposition 4 which follows is one expression of this fact and gives a positive lower bound for  $\bar{c}$  which applies for  $D$  large enough. Before proceeding to the proof of Proposition 4, we need to state a result which is of independent interest concerning the optimality of state-monotone policies for  $\bar{P}(c)$ . Its proof may be found in Appendix A.

**Proposition 3** *If  $\lambda(S) > \mu$  then for fixed values of the remaining system parameters,  $\exists \hat{D}$  such that  $D \geq \hat{D}$  implies the existence of an optimal policy for  $\bar{P}(c)$  which is monotone decreasing in the state.*

Please note that state monotonicity of optimal policies for  $\bar{P}(c)$  for fully indexable products will feed through into indices  $c(a, i)$  which are themselves monotone increasing in  $i$  for each fixed  $a$ . By abuse of notation we now use  $S$  for the policy for  $\bar{P}(c)$  which uses action  $S$  (maximal resource) in all states up to  $N - 1$  and  $p^S(j)$  for the stationary probability of state  $j$  under such a policy.

**Proposition 4** *If  $\lambda(S) > \mu$  then for fixed values of the remaining system parameters,  $\exists D^*$  such that if  $D \geq D^*$  then  $\bar{c}(D) \geq p^S(-M)(D\mu + bM)^{-1}$ .*

*Proof* From Proposition 3 we know that, if  $\lambda(S) > \mu$  then for  $D$  large enough, the search for optimal policies can be restricted to those stationary policies which are monotone decreasing in the state. Within this set of policies, cost rates for those policies which are not identically zero are trivially bounded below by  $p^S(-M)$ . Further, if  $D$  is such that  $hN \leq D\mu + bM$  then it is easy to show that the cost rate for the zero policy is bounded above by  $c(D\mu + bM)$ . Hence, for  $D$  large enough, if  $c(D\mu + bM) < p^S(-M)$  then the zero policy (which we call  $\emptyset$ ) is optimal. However, when the zero policy is optimal  $\partial_c^+ \Delta(c, i)$  ( $i > -M$ ) has the same sign as

$$\sum_{j=-M}^N \{p_i^\emptyset(j) - p^\emptyset(j)\} f(j) \leq \max\{b(M-1), hN\} - (D\mu + bM) p^\emptyset(-M). \quad (28)$$

The right hand side of (28) is negative for  $D$  large enough. The result now follows.  $\square$

Before proceeding to develop a simple sufficient condition for full indexability in the form of a lower bound for the  $\bar{h}$  in Corollary 1, we elucidate an important simplification to (27) which is available under reasonable assumptions regarding system parameters. It states that, in the search for  $\bar{c}$ , the smallest value of  $c$  at which a positive right gradient in one of the  $\Delta(c, i)$  is encountered, the  $i$ -value concerned must be  $N$ .

**Proposition 5** *If  $bM + D\mu \geq hN$  then*

$$\bar{c} = \inf \{c \geq 0 : \partial_c^+ \Delta(c, N) \geq 0\}.$$

*Proof* Recall from (17) that, for all choices of  $i$  and  $\pi$ ,

$$\{\tau^\pi(i)\}^{-1} \partial_c^+ \Delta^\pi(c, i) = \mathcal{X}_i^\pi - \sum_{j=-M}^N p^\pi(j) f(j), \quad (29)$$

where

$$\mathcal{X}_i^\pi = \sum_{j=-M}^N p_i^\pi(j) (hj^+ + bj^-). \quad (30)$$

It is then trivial from (30) that over the range  $0 \leq i \leq N$ ,  $\mathcal{X}_i^\pi$  must be increasing in  $i$ . Hence from (29) we have that, for all  $\pi$ ,

$$\partial_c^+ \Delta^\pi(c, i) > 0 \implies \partial_c^+ \Delta^\pi(c, i+1) > 0, \quad 0 \leq i < N. \quad (31)$$

We now introduce the complementary subsets of the set  $\{-M+1, -M+2, \dots, -1\}$  defined by

$$\begin{aligned} S_1 &= \{i : -M < i < 0 \text{ and } b|i| < hN\}, \\ S_2 &= \{i : -M < i < 0 \text{ and } b|i| \geq hN\}. \end{aligned}$$

It is trivial that, for all  $\pi$ ,

$$\mathcal{X}_i^\pi < hN = \mathcal{X}_N^\pi, \quad i \in S_1$$

and hence, using (29) that

$$\partial_c^+ \Delta^\pi(c, i) \geq 0 \implies \partial_c^+ \Delta^\pi(c, N) > 0, \quad i \in S_1. \quad (32)$$

It is also easy to establish that  $i \in S_2 \Rightarrow b|i| \geq hN \Rightarrow b|i-1| \geq hN \Rightarrow i-1 \in S_2$ . Further,  $\mathcal{X}_i^\pi < \mathcal{X}_{i-1}^\pi$  for any  $\pi$  and any  $i \in S_2$ . By induction we then conclude that

$$\mathcal{X}_i^\pi < \mathcal{X}_{i-1}^\pi < \dots < \mathcal{X}_{1-M}^\pi, \quad i \in S_2. \quad (33)$$

However, if  $bM + D\mu \geq hN$  then, by a direct calculation based on (29) we easily infer that, for all  $\pi$ ,

$$\partial_c^+ \Delta^\pi(c, 1-M) < 0. \quad (34)$$

Combining (33) and (34) we deduce that, for all  $\pi$ ,

$$\partial_c^+ \Delta^\pi(c, i) < 0, \quad i \in S_2. \quad (35)$$

It now follows from (31), (32), and (35) that, under the hypothesis of the Proposition,

$$\begin{aligned} \partial_c^+ \Delta(c, i) &\geq 0 \text{ for some } -M < i \leq N \\ &\implies \partial_c^+ \Delta(c, N) \geq 0. \end{aligned}$$

This completes the proof.  $\square$

We can now use Proposition 5 to strengthen Corollary 1.

**Corollary 2** *For fixed values of the remaining system parameters, if*

$$0 \leq h < N^{-1} \left[ \sum_{j=-M}^{-1} p^S(j) \{bj^- + D\mu I(j=-M)\} \right] \equiv \hat{h} \quad (36)$$

*then  $\bar{c}(h) = \infty$  and the product is fully indexable.*

*Proof* If  $h \leq \hat{h}$  then  $hN \leq bM + D\mu$  and hence from Proposition 5 we infer that

$$\bar{c} = \inf \{c \geq 0 : \partial_c^+ \Delta(c, N) \geq 0\}.$$

However, for any policy  $\pi$ , we infer from (17) that

$$\{\tau^\pi(N)\}^{-1} \partial_c \Delta^\pi(c, N) = hN - \sum_{j=-M}^N p^\pi(j) f(j), \quad (37)$$

$$\begin{aligned} &\leq hN - \sum_{j=-M}^{-1} p^S(j) f(j), \\ &= hN - \sum_{j=-M}^{-1} p^S(j) \{bj^- + D\mu I(j=-M)\}. \quad (38) \end{aligned}$$

Plainly from (38) it follows that for all stationary policies  $\pi$

$$\begin{aligned} h < \hat{h} &\implies \partial_c \Delta^\pi(c, N) < 0 \\ &\implies \partial_c^+ \Delta(c, N) < 0, \text{ for all } c. \end{aligned}$$

The result follows.  $\square$

*Remark 6* In practice, Corollary 2 guarantees full indexability for most reasonable values of the holding cost parameter  $h$ . By reasonable we mean those which conform with the conventional stock management interpretation of  $h$  as a single time-step cost for product depreciation and storage costs. This will be dwarfed by the backorder cost rate  $b$  and the penalty  $D$  for lost sales. For example, taking  $M = N = 10$ ,  $b = 1.5$ ,  $D = 200$  and  $\lambda(S) = 1$  then  $\hat{h}$  takes values 0.006, 0.07 and 0.32 when  $\mu$  equals 0.667, 0.8 and 0.9 respectively. Reducing  $D$  to 50, say, only marginally reduces these  $\hat{h}$ -values. Such bounds on  $h$  are actually relatively high. However, should it prove necessary, a stronger but more complicated form of Corollary 2 is available. The  $h$ -dependent optimal policy which minimises the summation in equation (37),  $\pi^*(h)$ , say, takes the place of the full service policy  $S$ . Hence if we now write

$$\psi(h) = \min \left[ \frac{bM + D\mu}{N}, \frac{\sum_{j=-M}^{-1} p^{\pi^*(h)}(j) \{bj^- + D\mu I(j=-M)\}}{N - \sum_{j=1}^N j p^{\pi^*(h)}(j)} \right] \quad (39)$$

and

$$\tilde{h} = \inf \{h; h > 0 \text{ and } h > \psi(h)\} \quad (40)$$

then the conclusion that  $\bar{c}(h) = \infty$ ,  $0 \leq h \leq \tilde{h}$ , and hence that the product is fully indexable over this  $h$ -range follows from a modest modification of the proof of Corollary 2.

Example 3 illustrates the possibility of non-indexability even when the finite state space limits are removed. However, the proof of Corollary 1 makes it clear that, in the absence of holding costs ( $h = 0$ ), we do have indexability for arbitrarily large  $M$  and  $N$ . Proposition 6 states that this continues to hold in the limit which dispenses with the bound on the number of available backorders ( $M = \infty$ ).

**Proposition 6** *If  $M = \infty$ ,  $h = 0$  and  $N < \infty$  then the product is fully indexable provided stable finite cost policies exist.*

*Proof* This follows by differentiating equation (17). First we need to observe that  $\Delta(c, N)$  is well-defined and continuous in  $c$  since it is a simple function of the optimal minimised costs. The values of all  $\Delta(c, i)$  are then continuous and unique via the DP equation. Then differentiating (17) yields the fact that all  $\Delta(c, i)$  are decreasing in  $c$  as required.  $\square$

Resolving the position when  $h = 0$  and the bound on the inventory level is dispensed with ( $N = \infty$ ) has proved challenging and is deferred. However, we pause to point out that if  $M < \infty$ ,  $N = \infty$  and  $\lambda(0) = 0$  then  $h > 0$  can plainly lead to non-indexability since states  $i$  for which  $hi > bM + D\mu$  can never be used under an optimal policy and so there will always be an effective reduction to the finite state space case.

We conclude this subsection by reference to an important sub-case namely the *lost sales model*. See Remark 1 in Section 2. Analyses along the lines of which gave Proposition 3 and Corollary 1 yield the following:

- Proposition 7 (Lost sales model)** (i) *For fixed values of the remaining system parameters,  $\exists \hat{D}$  such that  $D \geq \hat{D}$  implies the existence of an optimal policy for  $\bar{P}(c)$  which is monotone decreasing in the state;*  
(ii) *For fixed values of the remaining system parameters,  $\exists \hat{h}$  such that  $h \leq \hat{h}$  implies that the product is fully indexable.*

### 3.3 Algorithmic checks for full indexability and index computation

We conclude this section by remarking that, should the question of full indexability *not* be resolved by reference to the preceding theoretical results, then it is a straightforward matter to check it numerically in any particular case. Algorithm 1 below both checks for full indexability and produces a set of optimal policies for  $\bar{P}(c)$ , namely  $\{\pi_n, 0 \leq n \leq L\}$  and the  $c$ -intervals  $\{[C_n, C_{n+1}), 0 \leq n \leq L\}$  over which they are optimal. This is as in the statement of Theorem 1, and the reason for the overly constructive nature of the proof of that theorem. In order to develop the algorithm we define, for any stationary policy  $\pi$ , two complementary subsets of the state space as follows:

$$J_-(\pi) = \{i : -M \leq i < N \text{ and } \partial_c \Delta^\pi(c, i+1) \leq 0\} \quad (41)$$

and

$$J_+(\pi) = \{-M, -M+1, \dots, N-1\} \setminus J_-(\pi). \quad (42)$$

In what follows we use  $e_i$  for an  $M+N+1$ -vector with a 1 in position  $M+1+i$  and zeroes elsewhere.

---

#### Algorithm 1

*Computing optimal policies for  $\bar{P}(c)$  and testing for full indexability*

---

**Step 0:** Set  $c = 0$  and initial optimal policy  $\pi \equiv \{\pi(-M), \pi(-M+1), \dots, \pi(N)\} = \{0, 0, \dots, 0\}$ .

**Step 1:** Use (17) to calculate  $\Delta^\pi(c, i)$  and  $\partial_c \Delta^\pi(c, i)$  for the currently optimal policy  $\pi$  and for  $-M < i \leq N$ .

**Step 2:** If  $J_-(\pi) \neq \emptyset$ , define

$$C^- = \inf \{c' > c : \Delta^\pi(c', i+1) \{\lambda(\pi(i)) - \lambda(\pi(i)+1)\} = 1, \text{ some } i \in J_-(\pi)\}. \quad (43)$$

Else let  $C^- = \infty$ . If  $C^- < \infty$  let  $i^-$  be the attaining  $i \in J_-(\pi)$  in (43).

**Step 3:** If  $J_+(\pi) \neq \emptyset$ , define

$$C^+ = \inf \{c' > c : \Delta^\pi(c', i+1) \{\lambda(\pi(i)-1) - \lambda(\pi(i))\} = 1, \text{ some } i \in J_+(\pi)\}. \quad (44)$$

Else let  $C^+ = \infty$ . If  $C^+ < \infty$  let  $i^+$  be the attaining  $i \in J_+(\pi)$  in (44).

Let  $C = \min(C^-, C^+)$ .

**Step 4:** If  $C^+ < C^-$  then declare the problem NOT FULLY INDEXABLE.

**Step 5:** If  $C = \infty$  then store  $\pi$  as optimal on  $[c, \infty)$  and STOP.

**Step 6:** If  $C < \infty$  then store  $\pi$  as optimal on  $[c, C)$ .

**Step 7:** If  $C = C^-$  then define  $\pi_{new} = \pi + e_{i^-}$ .

**Step 8:** If  $C = C^+$  then define  $\pi_{new} = \pi - e_{i^+}$ .

**Step 9:** Update  $\pi_{new} \mapsto \pi$ ,  $C \mapsto c$  and return to Step 1.

Please note that, should we be dealing with a problem which we know *a priori* to be fully indexable, then at every stage in the application of Algorithm 1 we will have  $C^- < C^+$  and hence  $C = C^-$ . In such cases we need never consider steps 3, 4 and 8. Hence we then have the reduced version, namely Algorithm 2, which only computes optimal policies.

---

### Algorithm 2

*Computing optimal policies for  $\bar{P}(c)$  in known fully indexable cases*

**Step 0:** Set  $c = 0$  and initial optimal policy  $\pi \equiv \{\pi(-M), \pi(-M+1), \dots, \pi(N)\} = \{0, 0, \dots, 0\}$ .

**Step 1:** Use (17) to calculate  $\Delta^\pi(c, i)$  and  $\partial_c \Delta^\pi(c, i)$  for the currently optimal policy  $\pi$  and for  $-M < i \leq N$ .

**Step 2:** Let

$$C = \inf \{c' > c : \Delta^\pi(c', i+1) \{\lambda(\pi(i)) - \lambda(\pi(i)+1)\} = 1, \text{ some } i \in J_-(\pi)\}. \quad (45)$$

Else let  $C = \infty$ . If  $C < \infty$  let  $i^-$  be the attaining  $i \in J_-(\pi)$  in (45).

**Step 3:** If  $C = \infty$  then store  $\pi$  as optimal on  $[c, \infty)$  and STOP.

**Step 4:** If  $C < \infty$  then store  $\pi$  as optimal on  $[c, C)$ .

**Step 5:** If  $C < \infty$  then define  $\pi_{new} = \pi + e_{i^-}$ .

**Step 6:** Update  $\pi_{new} \mapsto \pi$ ,  $C \mapsto c$  and return to Step 1.

---

That both Algorithms 1 and 2 do indeed produce optimal policies for  $\bar{P}(c)$  uses Propositions 1 and 2. In particular the continuity in  $c$  of the  $\Delta(c, i)$ , asserted in Proposition 1(c), ensures that the algorithms work effectively at  $c$ -values at which the optimal policy changes. Please also note that, once we have a full specification of optimal policies for  $\bar{P}(c)$ ,  $c \geq 0$ , then in the fully indexable cases it is trivial to infer index values. See Definition 2.

## 4 Numerical Study

In the principal part of our numerical study, the performance of the greedy heuristic  $\pi_W$  proposed in Section 2.1 is compared with four competitor heuristics. In most problems reported here there are two products ( $K = 2$ ) with

twenty-five units of resource available to drive production ( $S = 25$ ). As observed in Remark 2, the choice of such a moderately large  $S$  suffices to model the divisibility of our resource into small pieces. The exact value of moderately large  $S$  is immaterial. The important feature is the range of production rates available as determined by the ranges of the functions  $\lambda_k(a)$ . The restriction to two product problems is made so that direct application of stochastic DP is possible (though expensive) to solve the problems to optimality. In all problems considered and for all heuristics the percentage cost rate suboptimality  $100(V^{\text{HEURISTIC}} - V^{\text{OPT}})(V^{\text{OPT}})^{-1}$  was computed by means of DP value iteration. At the end of the next subsection we do consider some *four* product problems ( $K = 4$ ), where finding optimal policies becomes prohibitively time consuming, but the comparison of heuristics is still possible.

#### 4.1 Comparative heuristic performance in fully indexable cases

As we saw in Section 3, products with realistically low  $h$  are generally fully indexable. Thus the results presented in this subsection are for models which exhibit full indexability. An important observation regarding our numerical study here is that in studying fully indexable products we are consequently restricting our analysis to models where the primary concern for the system administrator is the optimal dynamic balance of resource sharing without idling. We shall illustrate, by comparing  $\pi_W$  to *block policies* which dynamically use exclusively either  $\mathbf{a} = (S, 0)$  or  $\mathbf{a} = (0, S)$ , the value of the sharing of resource under a concave production rate and, by comparing to *static policies*, the value of being able to act dynamically depending on the system state. In the interests of providing a competitive heuristic which is *both* dynamic and also concerned with resource balancing we further implement a one-step policy improvement of the best static policy. We shall see that the greedy index heuristic,  $\pi_W$ , comfortably outperforms all of these competitor heuristics.

With further regard to the model parameters in our study, we shall usually study problems with a maximum of ten backorders ( $M = 10$ ) and maximally ten items of inventory ( $N = 10$ ), for each product. For many of the system costs it is only relative values that concern us, so we shall maintain the backorder costs at  $b = 1.5$  throughout. The lost sales penalty  $D$  will be considered at two levels,  $D = 50$  and  $D = 200$ , and the depreciation costs  $h$  initially will be held at  $h = 0.0002$  though we later entertain larger  $h$  values when considering idling. In Table 2 we will apply our heuristics to models with larger  $N$  values. However, keeping  $M + N$  relatively low is important in keeping the computation times of optimal policies manageable. Regarding computational effort, it is worthy of note that computing indices can be done offline for each product independently, and so while recalculating an optimal policy for a multi-product system after a small change to one product necessitates a full recalculation, the new  $\pi_W$  requires re-evaluation only of that product's indices. Our primary goal in the selection of  $\mu_k$  and  $\lambda_k(a)$  is to obtain problems where dynamic sharing of the resource is of interest. Hence the parameter sets



selected try to avoid cases where one product has such high relative demand (and consequently, costs) that the problem is very close to one involving only a single product. At the end of this section we present some results from some four product problems, where optimal policies are out of reach in computationally feasible timescales, but where our index heuristic is seen to comfortably outperform even our relatively strong policy improvement heuristic.

In our main study with  $K = 2$  we consider 6300 randomly generated problems over a range of system parameters. These problems are constructed from seven smaller studies of 900 problems over three distinct families of production rate functions  $\lambda_k(a)$  with varying degrees of concavity. We consider one family of *reciprocal form*:

$$\lambda_k(a) = \frac{a\lambda_k}{a + \beta_k}, \quad 0 \leq a \leq 25, \quad k = 1, 2;$$

a second with a *power form*:

$$\lambda_k(a) = 0.2\sqrt{a\lambda_k}, \quad 0 \leq a \leq 25, \quad k = 1, 2;$$

and finally a *logarithmic* family:

$$\lambda_k(a) = \lambda_k \log \left( \frac{a\beta_k + 25}{25} \right), \quad 0 \leq a \leq 25, \quad k = 1, 2,$$

where  $\lambda_k$  and  $\beta_k$  are positive constants. Natural calibration of the choices of  $\lambda_k$ ,  $\beta_k$  and  $\mu_k$  to develop interesting and diverse problems means that we consider four primary parameter sets A, B, C and D. Details of these sets can be found in Appendix B, but it is sufficient here to observe that A, B and C respectively represent models with decreasing demand rates  $\mu$ , while dataset D was designed for use with the *logarithmic* production rate and draws parameter values randomly from a subset of the range of set A. Within each of the datasets the values of  $\mu_k$  are drawn randomly from a uniform distribution and a variety of fixed  $\lambda_k$  and  $\beta_k$  combinations are cycled through to model problems with (i) identical production rates, (ii) products with rates in a two-to-one speed-up ratio, (iii) a three-to-one ratio; as well as different  $\beta_k$  which has the added effect of varying the degree of concavity of the production rate.

We now describe the heuristics for resource allocation considered in the numerical study.

INDEX : the greedy index heuristic,  $\pi_W$ , proposed in Section 2.1. All problems generated in this part of the study are fully indexable;

STATIC : the optimal policy in the *static class* which chooses a fixed resource allocation to drive production for all time. Development of this policy is easily accomplished and results from the minimisation

$$\min_{\mathbf{a} \in \mathcal{A}} \sum_{k=1}^2 E_{a_k} [f_k(X_k)]$$

where  $\mathcal{A} = \{\mathbf{a} : a_1, a_2 \in \mathbb{N}, a_1 + a_2 \leq 25\}$ ;

BLOCK : the policy which treats all 25 resource units as a single block, but allocates them dynamically according to an appropriately designed index policy. Thus at each stage it allocates either (25, 0) or (0, 25) according to INDEX above applied to a model with  $S = 1$  and  $\hat{\lambda}_k(1) = \lambda_k(25)$ ;

MYOPIC : the policy which in all states  $\mathbf{x}$  chooses an action to maximise the rate of reduction of the instantaneous inventory cost rate, as determined by the minimisation

$$\min_{\mathbf{a} \in \mathcal{A}} \sum_{k=1}^2 \lambda_k(a_k) \{f_k(x_k + 1) - f_k(x_k)\} I(x_k < N_k).$$

OSPI : the policy resulting from taking STATIC and applying one step of the policy improvement algorithm.

Table 1 demonstrates the exceptional performance of  $\pi_W$  in a wide variety of problems. The first observation to make from the table is that BLOCK and MYOPIC heuristics perform very badly. MYOPIC performs so badly because when  $h > 0$  it refuses to stockpile any inventory. Thus in the problems of interest to us, where  $h$  is small and optimal policies do not idle below state  $N$ , MYOPIC performs very poorly. This emphasises the importance of considering longer-term costs in making good policy decisions. The very poor performance of BLOCK also serves to illustrate the critical importance of balancing resource usage under the concavity assumptions on  $\lambda_k(a)$ . Restricting to either  $\mathbf{a} = (S, 0)$  or  $\mathbf{a} = (0, S)$  at all times, albeit allowing dynamical reallocation, leads to serious losses in potential productive capacity when  $\lambda_k(a)$  is not close to linear.

While STATIC outperforms MYOPIC and BLOCK, its evident weakness does make clear that the inability to act dynamically to adjust allocations can lead to significant losses. In part, because of the poor performance of MYOPIC, BLOCK and STATIC we have also implemented OSPI, a one-step policy improvement on STATIC, in the interests of comparing INDEX directly with a well designed resource balancing, dynamic policy. We know that for dynamic programming problems in many application contexts the policy improvement updates to sensible policies do often very quickly return excellent policies. Hence OSPI should provide a strong benchmark against which to test our INDEX heuristic.

By way of explanation of the results in Table 1, it is worth noting that the optimal minimised costs decrease as we use datasets A, B and C in turn. For the *reciprocal form* problems dataset A consists of problems with a median optimal cost rate of 128.28 and maximal cost rate of 421.57, while datasets B and C have medians of 14.31 and 0.59 and maxima of 77.34 and 11.92 respectively. Thus while maximal larger percentage suboptimality are observed for dataset C these correspond to very small absolute errors as they are inflated by very low optimal costs. It is clear, however, that across this wide range of problems  $\pi_W$  performs outstandingly well. INDEX quite clearly outperforms even the OSPI heuristic across all rows of the table. In Table 2 we briefly present a

PROBLEM SET	INDEX		STATIC		POLICIES		BLOCK		MYOPIC	
	MED	MAX	MED	MAX	MED	MAX	MED	MAX	MED	MAX
Recip A	0.004	0.859	0.870	42.955	0.047	3.943	131.496	1470.290	72.292	558.735
Recip B	0.239	4.055	25.513	122.785	2.311	11.536	1094.912	>10000	347.324	>2000
Recip C	1.262	11.327	94.945	284.436	7.376	22.091	>10000	>20000	>2000	>20000
Power A	0	0.025	0.013	1.528	0	0.458	21.282	115.359	14.897	43.987
Power B	0.001	0.163	0.149	14.712	0.029	4.892	39.350	299.305	25.192	103.310
Power C	0.049	7.603	3.141	347.921	0.999	48.271	129.329	>9000	50.768	>2000
Log D	0.230	5.458	8.776	164.421	2.902	43.489	76.113	1671.783	75.289	1181.374

Table 1: Median and maximum percentage suboptimality of five heuristic policies for three forms of production rate. There are 900 problems summarised in each row of the table. Details in text.

smaller random selection of problems using datasets A and B but with  $M = 5$ ,  $N = 20$  and  $D = 50$ . The results are consistent with those in Table 1.

PROBLEM SET	POLICIES					
	INDEX		OSPI		STATIC	
	MED	MAX	MED	MAX	MED	MAX
Recip A	0.003	1.452	0.038	8.130	0.617	66.894
Recip B	0.560	10.033	4.131	22.545	37.051	240.303

Table 2: Median and maximum percentage suboptimality of heuristic policies for problems with the *reciprocal form* of production rate. Other details in text.

To complete our analysis of the greedy index heuristic we finally consider much larger problems. We present in Table 3 the relative performance of INDEX, OSPI, and STATIC for 900 four station problems ( $K = 4$ ). The state space of these problems is already large enough (around 200000 states) to make finding optimal policies prohibitive. Dataset G used for these problems is described in Appendix B, though essentially is an extension of dataset A to four products. One result of considering more products is that even with similarly constrained parameter values a very wide range of resultant achievable performances arise. The fact that INDEX outperforms OSPI across the entire range of problems should stand as an illustration of its effectiveness in a wide variety of scenarios. The results indicate that  $\pi_W$  improves its performance relative to its best competitor, OSPI, when more products are considered. Over the 900 problems the median percentage cost increase from INDEX to OSPI is 19%. The lowest losses from using OSPI compared to INDEX arise in problems with one very costly queue where overall costs are high and thus percentage differences inherently low. In problems with four relatively similar products the cost of OSPI is often around 50% above that of INDEX. As would be expected, the benefits of dynamic policies grow with the number of products, resulting in median losses of STATIC (relative to INDEX) of 114%. In conclusion, these results point toward even stronger performance of INDEX in product scenarios with large  $K$  than in the two-product problems earlier considered for tractability.

#### 4.2 The use of fair charges in heuristics with idling for non-indexable cases

Finally, we address the issues of idling (i.e. not using the maximal resource  $S$  in all system states) and non-indexability, which often will arise concurrently. The issue of idling arises exactly when  $h$  is large, that is, when holding costs in states visited under the optimal policy are comparable with the backorder penalties incurred. For such problems Algorithm 1 remains a valid approach to finding optimal policies. In the non-indexable cases we discover that the optimal policies exhibit a very particular form of non-indexability. We find

% LOSS	POLICIES	
	OSPI	STATIC
Min	0.68	2.34
LQ	9.37	39.41
Median	18.91	113.87
UQ	30.72	246.41
Max	87.47	1059.30

Table 3: Four product problems. Percentage losses relative to INDEX from using OSPI and STATIC over the 900 problems from dataset G.

that, for a fixed state  $i$ , the optimal action values for  $\bar{P}(c)$  as  $c$  increases form either (i) a sequence which is monotonically increasing to  $S$ , or (ii) a sequence which is monotonically increasing to some  $s^*$  and which then decreases to 0. Thus in case (ii) non-indexability is manifest by there being exactly *two* distinct fair charges at which indifference between allocation levels  $a - 1$  and  $a$  occurs for some states at some action levels and also some states where the optimal action level is below  $S$  for all  $c$ . The intuition of these values as *fair charges* suggests the creation of the fair charge heuristics  $H_1$  and  $H_2$  described below.

$H_1$  : A greedy ‘fair charge’ policy performs as  $\pi_W$ , treating unique fair charges as indices, and always using the **higher** available fair charge as the index for any state-action pair  $(i, a)$  in the case of two available fair charges.

$H_2$  : A greedy ‘fair charge’ policy performs as  $\pi_W$ , treating unique fair charges as indices, and always using the **lower** available fair charge as the index for any state-action pair  $(i, a)$  in the case of two available fair charges.

With direct reference to our definition of the greedy heuristic  $\pi_W$  in Subsection 2.1 and index definitions (8) and (9) we observe that idling is not optimal in state  $\mathbf{i}$  of our relaxed problem when  $\bar{\pi}_k(c, i_k) \uparrow S$  as  $c \rightarrow \infty$  for every product  $k$  (barring maximal-state products where idling is naturally mandated) – this corresponds with case (i) above. Yet in case (ii), (8) and (9) will disagree on an index value if  $c_k^{\text{inf}}(a, i) \neq \infty$ , where  $c_k^{\text{inf}}(a, i)$  denotes the  $c$ -value emerging from (8). In this case we resort to the fair charge intuition, the two fair charges being  $W^{\text{high}} = (c_k^{\text{inf}}(a, i))^{-1}$  from (8) and  $W^{\text{low}} = (c_k^*(a, i))^{-1}$  where  $c_k^*(a, i) = \inf\{c > c^{\text{inf}} : \bar{\pi}_k(c, i) \leq a\}$ . Any resource state pairs  $(a, i)$  such that  $c_k^{\text{inf}}(a, i) = \infty$  are states where the relaxed solution never allocates more than resource  $a$ . It is clear that in this case the appropriate unique fair charge is  $c_k(a, i) = \infty$ , and hence  $W_k(a, i) = 0$ . Implementation of  $\pi_W$ , as described in Subsection 2.1, for  $H_1$  and  $H_2$  then uses these new  $W$ -indices with a minor modification to equation (11) in ‘Step 2’ of the algorithm. Namely, that allocation terminates without using all resource (idling) when  $\max_{1 \leq j \leq K} W_k(a_j, i_j) = 0$ . Idling is performed when at an allocation in which there is surplus resource no project has a positive fair charge for raising its resource level.

When discussing the importance of indexability in Subsection 2.2 we observed that when a problem is non-indexable the best choice of fair charge

for use in a greedy fair charge policy can depend on the system as a whole. The intuition for a policy like  $H_2$  performing well is as follows. Assume that  $\pi(W^*)$ , which achieves  $V^*$  in (12), is a strongly performing policy for the relaxed problem with no strict  $S$  limit for the resource deployed. We conclude that the preferred fair charges for use in the design of heuristic policies are those in the region of  $W^*$ . In our models here we take  $h = 0.25$  to create problems with idling. We also choose parameter sets such that under optimal policies the time spent in idling states is significant enough to have a marked effect upon optimal costs. In light of this, we created datasets E and F, which are in fact just datasets A and C but with demand rates  $\mu$  drawn from uniform distributions whose limits are halved. These models have lower  $W^*$  values than those considered earlier since  $W^*$  is a measure of the underlying value placed on resource by the system. Thus if the lower fair charges arising from the use of Algorithm 1 are comparable with  $W^*$  then we would like our heuristic for these problems to use these lower fair charges. Hence in these cases heuristic  $H_2$  is the more compelling proposal. We shall, in fact, implement  $H'_2(T)$  which modifies  $H_2$  by ignoring lower fair charges if they exceed some threshold  $T$ . This reflects the fact that in some non-indexable cases, it can be that *both* fair charges are well in excess of  $W^*$ .

In Table 4 we present the relative performances of  $H_1$ ,  $H'_2(1/70)$ , STATIC, and OSPI for 1800 two product problems ( $K = 2$ ) with parameters drawn from datasets E and F and where  $h = 0.25$ . We do not suggest that  $H_1$ ,  $H_2$  and  $H'_2(T)$  are the *only* worthwhile proposals for the deployment of fair charges in non-indexable contexts to develop heuristics for problems with idling. It is, however, clear that while  $H_1$  quite obviously breaks down as a sensible heuristic in a non-indexable environment,  $H'_2(T)$  is a very serious proposal for the use of the fair charge methodology. Median suboptimality of 1.155% (dataset E) and 0.470% (dataset F) for  $H'_2(1/70)$  compare with 5.737% and 27.429% respectively for OSPI and suggest strongly that carefully designed policies based on fair charges can offer compelling solutions to these challenging problems.

% SUBOPT.		DATASET E				DATASET F			
		H2'	OSPI	H1	STATIC	H2'	OSPI	H1	STATIC
Min		0.034	0.108	6.425	26.491	0.001	0.016	123.708	269.711
LQ		0.557	2.583	45.938	103.908	0.057	7.276	155.009	333.037
Median		1.155	5.737	61.183	135.974	0.470	27.429	172.520	359.905
UQ		2.149	11.560	76.132	168.242	1.259	54.766	189.450	387.342
Max		8.887	54.747	134.928	279.051	4.672	257.294	282.338	496.718

Table 4: Percentage suboptimality of  $H'_2(1/70)$ , OSPI,  $H_1$ , and STATIC, in a high  $h$  scenario. There are 900 problems for each dataset.

---

*Acknowledgement*

The authors acknowledge the support for this work by the Engineering and Physical Sciences Research Council (EPSRC) through grant EP/E049265/1. They are also grateful to two referees and a co-ordinating editor for valuable suggestions which have strengthened the paper.

**References**

1. Ata, B. and Shnoerson, S. (2006). Dynamic control of an M/M/1 service system with adjustable arrival and service rates. *Management Science*, 52:1778–1793.
2. de Vericourt, F., Karaesman, F., and Dallery, Y. (2000). Dynamic scheduling in a make-to-stock system: a partial characterization of optimal policies. *Operations Research*, 48(5):811–819.
3. Gavish, B. and Graves, S. C. (1980). A one-product production/inventory problem under continuous review policy. *Operations Research*, 28(5):1228–1236.
4. Glazebrook, K. D., Hodge, D. J., and Kirkbride, C. (2011). General notions of indexability for queueing control and asset management. *The Annals of Applied Probability*, (to appear).
5. Glazebrook, K. D., Kirkbride, C., and Ouenniche, J. (2009). Index policies for the admission control and routing of impatient customers to heterogeneous service stations. *Operations Research*, 57:975–989.
6. Ha, A. Y. (1997). Optimal dynamic scheduling policy for a make-to-stock production system. *Operations Research*, 45(1):42–53.
7. Menich, R. and Serfozo, R. F. (1991). Optimality of routing and servicing in dependent parallel processing systems. *Queueing Systems*, 9(4):403–418.
8. Niño-Mora, J. (2001). Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, 33:76–98.
9. Niño-Mora, J. (2006). Restless bandit marginal productivity indices, diminishing returns, and optimal control of make-to-order/make-to-stock M/G/1 queues. *Mathematics of Operations Research*, 31(1):50–84.
10. Perez, A. P. and Zipkin, P. (1997). Dynamic scheduling rules for a multi-product make-to-stock queue. *Operations Research*, 45(6):919–930.
11. Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY.
12. Sobel, M. (1969). Optimal average cost policy for a queue with start-up and shut-down costs. *Operations Research*, 17:145–162.
13. Sobel, M. (1982). The optimality of full-service policies. *Operations Research*, 30:636–649.
14. Tijms, H. (1994). *Stochastic Models: An Algorithmic Approach*. John Wiley and Sons, New York, NY.
15. Veatch, M. H. and Wein, L. M. (1996). Scheduling a make-to-stock queue: index policies and hedging points. *Operations Research*, 44(4):634–647.

16. Weber, R. R. and Weiss, G. (1990). On an index policy for restless bandits. *Journal of Applied Probability*, 27:637–648.
17. Wein, L. M. (1992). Dynamic scheduling of a multi-class make-to-stock queue. *Operations Research*, 40(4):724–735.
18. Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298.
19. Zheng, Y.-S. and Zipkin, P. (1990). A queueing model to analyze the value of centralized inventory information. *Operations Research*, 38(2):296–307.
20. Zipkin, P. (1995). Performance analysis of a multi-item production-inventory system under alternative policies. *Management Science*, 41(4):690–703.

## Appendix A On state monotonicity of optimal policies

For the proof of Proposition 3, we need the following preparatory lemma.

**Lemma A.1** *For fixed values of the remaining system parameters,  $\exists \bar{D}$  such that  $D \geq \bar{D}$  implies that all optimal policies for  $\bar{P}(c)$  choose maximal resource level  $S$  in state  $-M$ .*

*Proof* We have that

$$\gamma(c) = \sum_{j=-M}^N p^{\bar{\pi}(c)}(j) \{chj^+ + cbj^- + cD\mu I(j=-M) + \bar{\pi}(c, j)\}, \quad (\text{A.1})$$

where as usual  $\bar{\pi}(c)$  is any optimal policy for  $\bar{P}(c)$ . We have a similar expression for  $\gamma^\pi(c)$  for any stationary policy  $\pi$ . These expressions are used in the subsequent argument.

Let  $A$  be the set of stationary policies  $\pi$  for which  $\pi(-M) < S$ . Within  $A$ , it is easy to show that the smallest value of  $p^\pi(-M)$  is achieved by the policy,  $\pi^A$ , which uses action  $S-1$  in state  $-M$  and action  $S$  in all other states. Therefore we deduce  $p^\pi(-M) \geq p^{\pi^A}(-M) \equiv p_1$  for all  $\pi \in A$  for some constant  $p_1$ . Plainly, however, we also have

$$\gamma^\pi(c) \geq p_1 c(bM + D\mu), \quad \pi \in A. \quad (\text{A.2})$$

Consider now the policy which uses action  $S$  in all states and has associated cost rate  $\gamma^S(c)$ . We find, via a simple calculation, that  $p^S(-M) < p_1$  and yet

$$\gamma^S(c) < p^S(-M)c(bM + D\mu) + chN + cb(M-1) + S. \quad (\text{A.3})$$

From (A.2) and (A.3),

$$\gamma^\pi(c) - \gamma^S(c) > (p_1 - p^S(-M))c(bM + D\mu) - chN - cb(M-1) - S, \quad \pi \in A. \quad (\text{A.4})$$

The expression on the r.h.s of (A.4) is positive for sufficiently large  $D$ . Hence for fixed values of the remaining system parameters, there exists  $\bar{D}$  such that  $D \geq \bar{D}$  implies that no member of  $A$  can be optimal. This concludes the proof.  $\square$

State monotonicity of optimal policies for sufficiently large  $D$  under given conditions now follows.

**Proposition 3** *If  $\lambda(S) > \mu$  then for fixed values of the remaining system parameters,  $\exists \hat{D}$  such that  $D \geq \hat{D}$  implies that*

- (a)  $\Delta(c, i)$  is increasing in  $i$  on  $-M \leq i \leq N$ ;



(b) there exists a stationary optimal policy  $\bar{\pi}(c)$  for  $\bar{P}(c)$  which is monotone decreasing in the state.

*Proof* It is a straightforward consequence of the DP optimality equations (14) that, over the range  $1 \leq i \leq N - 2$ , the following hold:

- (i)  $\Delta(c, i + 1) \leq \Delta(c, i + 2) \Rightarrow \Delta(c, i) \leq \Delta(c, i + 1) - ch/\mu$ .
- (ii)  $\Delta(c, i) \geq \Delta(c, i + 1) \Rightarrow \Delta(c, i + 1) \geq \Delta(c, i + 2) + ch/\lambda(S)$ .

Taken together, (i) and (ii) imply that over the range  $1 \leq i \leq N$  either  $\Delta(c, i)$  is increasing in  $i$  or it is increasing up to some  $i^*$  and is then decreasing. To eliminate the second possibility we need to demonstrate that  $\Delta(c, N - 1) < \Delta(c, N)$  for large enough  $D$ . However, from (14) we have

$$\mu\Delta(c, N) = -\gamma(c) + chN, \quad (\text{A.5})$$

and

$$\mu\Delta(c, N - 1) = -\gamma(c) + ch(N - 1) + \min\{a + \lambda(a)\Delta(c, N)\}, \quad (\text{A.6})$$

with the minimisation in (A.6) taken over  $a \in \{0, 1, \dots, S\}$ . Now, from the proof of Lemma A.1,

$$\gamma(c) \geq cD\mu p^S(-M),$$

which in turn yields from (A.5) that

$$D > \frac{hN}{\mu p^S(-M)} \Rightarrow \mu\Delta(c, N) \leq -cD\mu p^S(-M) + chN < 0.$$

However, from (A.6) we see that upon choosing  $a = 0$  in the minimisation,

$$\Delta(c, N) < 0 \Rightarrow \Delta(c, N - 1) < \Delta(c, N).$$

From the above we conclude that

$$D > \frac{hN}{\mu p^S(-M)} \Rightarrow \Delta(c, N - 1) < \Delta(c, N), \quad (\text{A.7})$$

and hence that  $\Delta(c, i)$  is increasing in  $i$  over the range  $1 \leq i \leq N$ .

We now consider the range  $-M \leq i \leq 1$ . We first note that the  $I(i \neq -M)$  on the l.h.s. of (14) can be dropped if we replace  $f(i)$  by  $\bar{f}(i) = bi^- + hi^+$ . It follows that a modified form of (14) holds at  $i = -M$  without the  $I(i \neq -M)$  term if we define  $\Delta(c, -M) = -cD$ . The expressions for  $\Delta(c, i)$  and  $\Delta(c, i - 1)$  from this modified (14) are

$$\begin{aligned} \mu\Delta(c, i) &= -\gamma(c) + c\bar{f}(i) + \min_a \{a + \lambda(a)\Delta(c, i + 1)\}, \quad -M \leq i \leq 0 \\ \mu\Delta(c, i - 1) &= -\gamma(c) + c\bar{f}(i - 1) + \min_a \{a + \lambda(a)\Delta(c, i)\}, \quad -M < i \leq 1 \end{aligned}$$

and we can deduce that for  $-M < i \leq 0$ ,

$$\Delta(c, i) \geq \Delta(c, i - 1) \Rightarrow \Delta(c, i + 1) \geq \Delta(c, i) + cb/\lambda(S).$$

In light of this, and of the above, to prove (a) it is now sufficient to show that  $\Delta(c, 1 - M) > \Delta(c, -M)$  for large enough  $D$ .

From Lemma A.1 we know that  $\exists \bar{D}$  such that  $D \geq \bar{D}$  implies that the optimising action in state  $-M$  must be  $S$  and hence that

$$\Delta(c, -M) = -cD = -\gamma(c) + c\bar{f}(-M) + S + \lambda(S)\Delta(c, 1 - M). \quad (\text{A.8})$$

Rearranging, we conclude from (A.8) that

$$cD\{\lambda(S) - \mu\} + \gamma(c) - cbM = S + \lambda(S)\{\Delta(c, 1 - M) - \Delta(c, M)\}. \quad (\text{A.9})$$

However, from Proposition 1,  $\gamma(c)$  is continuous in  $D$  with positive right gradient

$$\partial_D^+ \gamma(c) = c\mu p^c(-M) \geq c\mu p^S(-M) > 0. \tag{A.10}$$

Hence it is plain that  $\gamma(c)$  is increasing everywhere in  $D$ . We conclude that if  $\lambda(S) > \mu$ , the l.h.s. of (A.9) is increasing strictly in  $D$  with strictly positive right gradient everywhere. Hence for all large enough  $D$ , it must follow that  $\Delta(c, 1 - M) > \Delta(c, M)$  as required. This concludes the proof of (a).

We conclude the proof of Proposition 3 by noting that (a) together with the observation made before Proposition 2 on DP equation solutions straightforwardly yield (b).  $\square$

## Appendix B Numerical parameter sets

Table 5 describes the ranges of parameter values used in each dataset in the studies presented in Section 4 of the paper. The key difference in construction between datasets A, B, C, D, E and F are the selections of the pair  $(x, y)$ . Dataset A uses  $(x, y) = (1, 1.5)$ , Dataset B uses

PARAMETERS					
$\lambda_1$	$\lambda_2$	$\mu_1$	$\mu_2$	$\beta_1$	$\beta_2$
1.5	1.5	$(x, y)$	$(x, y)$	(4,6)	(4,6)
1.5	3.0	$(x, y)$	$(2x, 2y)$	(4,6)	(4,6)
1.5	4.5	$(x, y)$	$(3x, 3y)$	(4,6)	(4,6)
1.2	1.2	$(x, y)$	$(x, y)$	(2,3)	(2,3)
1.2	2.4	$(x, y)$	$(2x, 2y)$	(2,3)	(2,3)
1.2	3.6	$(x, y)$	$(3x, 3y)$	(2,3)	(2,3)
1.5	1.2	$(x, y)$	$(x, y)$	(4,6)	(2,3)
1.5	2.4	$(x, y)$	$(2x, 2y)$	(4,6)	(2,3)
3.0	1.2	$(2x, 2y)$	$(x, y)$	(4,6)	(2,3)

Table 5: Parameter set descriptions. Open set brackets  $(a, b)$  represent independent draws from a continuous uniform distribution on  $[a, b]$ . For each row 100 samples are used to generate problems.

$(x, y) = (0.9, 1.05)$  and Dataset C uses  $(x, y) = (0.7, 0.9)$ . Datasets E and F use  $(0.5, 0.75)$  and  $(0.45, 0.575)$  respectively. Dataset D, devised for the *logarithmic form*  $\lambda(a)$  functions, uses  $(x, y) = (1.1, 1.3)$  but also modifies  $\beta_1$  and  $\beta_2$  to be drawn from  $U[2, 2.5]$  as opposed to  $U[4, 6]$  or  $U[2, 3]$  used for the other datasets. In all tables based on these data 100 problems were generated randomly selecting  $\mu_1, \mu_2, \beta_1$  and  $\beta_2$  uniformly from their respective distributions resulting in a total of 900 problems for each dataset.

Dataset G used for the 900 four product models uses the same structure as in Table 5 with dataset A. To cater for the increased demand from the extra product queues we increment all  $\lambda_k$  values above by one unit. In practice this has led to a very wide range of achieved performance values, but none too extremely low or high to make the problems concerned of negligible interest. The new third and fourth product queues are modelled upon the basic queue 1, and thus (except in row 9 where the basic non-spiced up product 1 is used) share  $\lambda_k$  value with  $\lambda_1$ . In keeping with product 1,  $\mu_3$  and  $\mu_4$  are drawn from independent continuous uniform distributions on  $(x, y)$ .