The Rental Problem

René Caldentey Susana Mondschein

March 14, 2000

1 Introduction

In this report we study the optimal rental policy for a house (apartement or cabin) that is offered for renting during a certain planning horizon H. The planning horizon is divided into consecutive time-units that correspond to the smallest length of time in which the property can be rented (*minimum rental time-periods*). Let $N = \{1, 2, \ldots, n\}$ be the set of all indivisible time-periods during the planning horizon H. For simplicity and without loss of generality, in what follows we consider that the minimum rental time-periods correspond to days. Thus, N is the set of days that define H and $i \in N$ is the i^{th} day contained in H according to a chronological order.

Let $F(N) \subseteq 2^N$ be the set of feasible periods of time to be rented. In the most general case we can have that $F(N) = 2^N$. However, there might exist cases where some combinations of days are not offered (or not demanded) and $F(N) \subset 2^N$. For the moment, we do not impose any particular constraint on $F(N)$. However, and as we will see later, important simplifications on the resolution of the rental problem can be obtained by restricting in a *natural way* the set $F(N)$.

For each set $f \in F(N)$, we define $C(f) = \{\bar{f} \in F(N) : f \cap \bar{f} \neq \emptyset\}$. In words, the set $C(f)$ contains all the periods that can not be rented if period f is rented.

We denote by T the time-period available for customers to make reservations. This reservation planning horizon (T) is partitioned in a sequence of disjoint intervals that we call stages. At each stage, at most one customer makes a reservation for a period $f \in F(N)$. We denote by $q(f)$ the probability that at any stage there is a customer requesting period f. For each period $f \in F(N)$, we associate a rental price $p(f)$.

Finally, let $B_k(S, f)$ be the maximum expected profit from the beginning of stage k onwards, if the set of available rental periods is equal to $S \subseteq F(N)$ and there is a request for period $f \in F(N)$. Similarly, $B_k(S)$ represents the maximum expected profit if at the beginning of stage k a set $S \subseteq F(N)$ of rental periods is available.

Given the discrete characterization of the demand process, the rental problem can be formulated as an stochastic dynamic programming problem that is given by:

Stochastic Rental Problem: **(SRP)**

$$B_k(S) = \sum_{f \in F(N)} [B_k(S, f)] \cdot q(f) \tag{1}$$

$$B_k(S,f) = \begin{cases} B_{k+1}(S) & f \notin S \\ max\{B_{k+1}(S);\ p(f) + B_{k+1}(S - C(f))\} & f \in S \end{cases}$$

The border conditions are $B_k(\emptyset, f) = 0,\ \forall f \in F(N)$, and $B_{K+1}(S,f) = 0,\ \forall S \subseteq F(N),\ \forall f \in F(N)$, where K is the index of the last stage when a period can be rented.

Let notice that the optimization problem under **SRP** is related to the definition of $B(S,f)$ above. The control for this DP formulation is the binary decision of renting or not a period f at stage k when it is available.

The function $B_k(S)$ satisfied some basic monotonicity properties that are summarized in the following proposition:

Proposition 1 *The function $B_k(S)$ is:*

1. *A non-decreasing function of the set of available rental periods S, i.e., let R and S be two sets of available rental periods, such that $R \subseteq S$, then $B_k(R) \leq B_k(S), \quad \forall k = 1, 2, \ldots, K$.*

2. *A non-decreasing function of time, i.e., let S be the set of available rental periods, then $B_{k-1}(S) \geq B_k(S), \quad \forall k = 2, 3, \ldots, K$.*

PROOF:
For the first part, the result can be proved using backward induction over k. From the border conditions, it is clear that $B_{K+1}(R) \leq B_{K+1}(S),\ \forall R, S$ with $R \subseteq S$. Let suppose that the result holds for $k+1, \ldots, K$. Therefore if $R \subseteq S$, from the definition of $B_k(\cdot, f)$ we have $B_k(R,f) \leq B_k(S,f)$ (since $\forall f \in R,\ f \in S$). Finally, combining this result an (1) the results holds for k.

For the second part, let notice that in period T_{k-1} the alternative of rejecting any customer (not to rent) is feasible and leads to $B_{k-1}(S) = B_k(S)$. Therefore, choosing the optimal solution, instead of a feasible one, must lead to $B_{k-1}(S) \geq B_k(S)$. ∎

Even though the DP formulation for **SRP** is simple and does not requires too much computation at each state (S, f), the state space can be big enough to make this approch unfeasible. In fact, in the worst case we have $|F(N)| = 2^n$ and therefore solving the DP formulation when the number of stages is K, requires to store $K \cdot 2^n$ values of the function $B_k(\cdot),\ k = 1, \ldots, K$. For this reason, we decide to approach the problem by first analyzing the deterministic version of it. Our goal behind this is to develop a solution method for the deterministic case that can be used as a building block for solving the stochastic version.

2 Deterministic Model

In this section we describe the deterministic version of the model and present a graph representation and a greedy algorithm to solve it in polynomial time. We consider the case where the demand for the whole period H is known at the beginning of the rental period T. Therefore, the decision maker knows exactly which periods in the set $F(N)$ are requested and the corresponding prices.

Let $\tilde{F} = \{f_i\}_1^m$ be the set of requested rental periods, and $\{p(f_i)\}_1^m$ the corresponding rental price. The deterministic problem consists of finding the optimal rental policy, i.e., which periods to rent among the set of all rental requests in $\tilde{F}$ in order to maximize the total profit.

In what follows we describe a graph representation of the problem. We define the graph $G = (\tilde{F}, A)$, where the set of nodes $\tilde{F}$ corresponds to the requested rental periods. The set of arcs A is defined by $A = \{(i,j) : f_i \cap f_j \neq \emptyset\}$, i.e., if the requested rental periods f_i and f_j have at least one day in common, then the arc (i,j) is in the graph. We define the decision variable x_i as follows:

$$x_i = \begin{cases} 1 & \text{rent period} \quad f_i \\ 0 & \text{otherwise} \end{cases}$$

The only set of constraints correspond to the condition that the decision maker cannot rent two periods that overlap. This condition is represented by the constraint that two nodes (requested rental periods) connected by an arc cannot belong simultaneously to a feasible solution. This problem is known as the classic *"node packing"* problem (see, for example, Nemhauser and Wolsey (1988)). The straightforward formulation is given by:

$$\begin{aligned} Z(\tilde{F}, p) = \max_{\{x_i, \forall i=1,\ldots,m\}} \; & \textstyle\sum_i p(f_i) \cdot x_i \\ & \text{s.t.} \\ & x_i + x_j \leq 1 \qquad \forall (i,j) \in A \\ & x_i \in \{0,1\} \quad \forall i = 1, \ldots, m. \end{aligned}$$

An equivalent formulation, based on the "fractional node-packing polytope" of graph G is given by:

Deterministic Rental Problem: (**DRP**)

$$\begin{aligned} Z(\tilde{F}, p) = \max_{\{x_i, \forall i=1,\ldots,m\}} \; & \textstyle\sum_i p(f_i) \cdot x_i \\ & \text{s.t.} \\ & \mathcal{K}x \leq 1 \\ & x_i \in \{0,1\}, \qquad \forall i = 1, \ldots, m, \end{aligned}$$

where $\mathcal{K}$ is the (0,1) incidence matrix whose rows correspond to all of the maximal cliques of G and whose columns correspond to the nodes of G, known as the *clique matrix* of G.

The structure of the problem is the one of a *stable set problem* [1]. The stable set problem is an *NP*-complete problem (see for example Garey and Johnson (1979)). However, the problem **DPR** does not represent a stable set problem per se. In fact, the decision problem associated to the rental problem can be stated as:

The Rental Problem:
INSTANCE: A set $N = \{1, 2, \ldots, n\}$, a collection $C = \{C_1, C_2, \ldots, C_m\}$ of subsets of N, a weight function $w : C \to Q^+$, and a positive integer B.
QUESTION: Is there a subcollection $\tilde{C} \subseteq C$ such that any two different set C_1, $C_2 \in \tilde{C}$ satisfy $C_1 \cap C_2 = \emptyset$, and $\sum_{c \in \tilde{C}} w_c \geq B$.

[1] The stable set problem is define, in general, as follows: Given a graph $G = (V, A)$ and a weight function $w : V \to Q^+$, find the subset of vertices $\tilde{V} \subseteq V$ such that *(i)* any two different vertices in $\tilde{V}$ are not adjacent in G, and *(ii)* $\tilde{V}$ maximizes $\sum_{v \in \tilde{V}} w_v$.

Proposition 2 *The rental Problem is an NP-complete problem.*

PROOF: See the Appendix. ∎

Given the complexity of the problem, we introduce an additional restriction to the demand process (or equivalently to $F(N)$).If the graph G is *chordal*, then it is possible to define a greedy algorithm to solve the problem in polynomial time (a graph is called *chordal* (or *triangular*) if it does not contain any k-holes, for $k \geq 4$, see Nemhauser and Wolsey (1988)). In our problem, for an arbitrary set of feasible rental periods, the resulting graph is not necessarily chordal. For example, if we consider 4 consecutive days and feasible periods (1,2), (2,3), (3,4), and (1,4), then they form a 4–hole (a cycle of four nodes without a chord). However, under certain regularity conditions for the set of feasible rental periods, the graph G is chordal.

Definition 1 *A rental period f_i is* regular *if it consists of consecutive days. Thus, if days k and l belong to f_i, with $k \leq l$, then j is in f_i, $\forall k \leq j \leq l$.*

A regular rental period is completely identified by the first and last days of it. In what follows we use the notation r_{kl} to identify a regular rental period that begins at day k and ends at day l ($k \leq l$).

Proposition 3 *If the rental periods are regular, the graph $G = (\tilde{F}, A)$ associated to the problem DRP is chordal.*

PROOF: See the Appendix. ∎

In fact, under the assumption of regularity for the rental periods, the underlying graph is exactly an *interval graph*[2] (see for example Lekkerkerker and Boland (1962)).

The important property of a chordal graph is that it contains a *perfect elimination scheme*, (PES)[3] which can be constructed in polynomial time (see Nemhauser and Wolsey (1988)). In what follows we derive a PES for the problem DRP, which is used in the greedy algorithm to find the optimal rental solution.

Definition 2 *Let $f_i = r_{kl}$ be a regular rental period of G. The cardinality of f_i is defined as the number of days in f_i and denoted by $|f_i|$.*

ELIMINATION PROCEDURE

The elimination procedure, described by the following two rules, induces a PES for graph $G = (\tilde{F}, A)$.

Rule 1: a node $f_i = r_{kl}$ can be eliminated if and only if all nodes $f_j = r_{st}$ with $t < l$ have been already eliminated.

[2] Interval graphs are defined in the following way: Given a collection C of closed intervals of the real line, the nodes of the graph correspond to each of the intervals in C, and two nodes are adjacent only if the corresponding interval have nonempty intersection.

[3] An ordering of the nodes of G, $\sigma = [\nu_1, \nu_2, \ldots, \nu_m]$ is a PES if, for $i = 1, \ldots, m-1, \nu_i$ is a simplicial node of the subgraph induced by $\{\nu_i, \ldots, \nu_m\}$. A node is simplicial if it and its neighbors form a clique.

Rule 2: a node $f_i = r_{kl}$, with $|f_i| = q$, can be eliminated if and only if all nodes $f_j = r_{st}$ with cardinality less than q have been already eliminated.

Once a node is eliminated from the graph, all arcs incident to it are also eliminated.

Proposition 4 *The elimination procedure described above induces a perfect elimination scheme of the nodes of $G(\tilde{F}, A)$ defined in problem DRP, when the rental periods are regular. The PES corresponds to the sequence in which the nodes are eliminated in the elimination procedure.*

PROOF:
In what follows we show that each node is a simplicial node at the step when it is eliminated from the remaining graph, i.e., it forms a clique with its neighbors.

Let us assume that, applying rules 1 and 2, node $f_i = r_{kl}$ is the candidate to be eliminated in the next iteration of the elimination procedure. We note that f_i is the only candidate to be eliminated. If we assume that there are two different candidates to be eliminated ($f_i = r_{kl}$ and $f_j = r_{st}$), then by rule 1, we have $l = t$, and by rule 2 we have that $k = s$ (they have the same cardinality and are regular rental periods). Therefore, both nodes correspond to the same rental period ($f_i = f_j$), which is unique.

Let us assume that $f_j = r_{st}$ is a neighbor of node f_i. Because f_i is the candidate to be eliminated, then $t \geq l$. Additionally, because f_j is an adjacent node of f_i, then we have that $s \leq l$ (both nodes have some days in common). Then, $s \leq l \leq t$, and day l is in node f_j. Thus, every adjacent node of $f_i = r_{kl}$ contains day l, and therefore, all pair of adjacent nodes of f_i are connected by an arc and f_i is a simplicial node. ∎

The importance of having a chordal graph and a corresponding PES is that problem **DPR** can be solved in polynomial time using a greedy algorithm. The details of the greedy algorithm, as well as a proof of its correctness, can be found in Nemhauser and Wolsey (1988). Here, for completeness, we present the main steps of the algorithm.

The Greedy Algorithm:
Let reorder the elements of $F(N)$ such that $[f_1, f_2, \ldots, f_m]$ is a PES. Let $\tilde{K}$ be an $m \times m$ matrix such that the i^{th} row of $\tilde{K}$ ($\tilde{k}^i$) is the characteristic vector of the clique containing node i in the subgraph induced by nodes $\{f_i, \ldots, f_m\}$.
The first step of the algorithm solves the dual problem of **DRP**.

Dual Step:

1. *Initialization:* $i = 1$, $c^i = p(f_i)$, $J_0 = \emptyset$, and $[f_1, f_2, \ldots, f_m]$ is a PES.

2. *Iteration:* $u_i = max(o, c^i_i)$.
   If $u_i > 0$, let $J_i = J_{i-1} \cup \{i\}$, and $c^{i+1} = c^i - u_i \tilde{k}^i$.
   If $u_i = 0$, then $J_i = J_{i-1}$, $c^{i+1} = c^i$.
   If $i = m$, stop; $(u_1, \ldots, u_m)$ is an optimal solution, Otherwise, $i \leftarrow i + 1$.

From the solution of the Dual step, the greedy algorithm computes the optimal solution for the primal problem **DRP**.

Primal Step:

1. *Initialization:* $J = J_m$ (from Dual step), $x_j = 0$ for $j \notin J$.

2. *Iteration:* Let l be the last element of J. Set $x_l = 1$ and $J \leftarrow J - \{i : \tilde{k}_{il} = 1\}$.
   If $J = \emptyset$, stop $(x_1, \ldots, x_m)$ is an optimal solution. Otherwise repeat.

The previous algorithm is very efficient and represents the main tool that we will use to approximate a solution to the original stochastic problem **SRP**.

3 Stochastic-Demand Case

In the previous section, we solved the deterministic case by exploting the well-behave structure of the problem under the assumption of regularity. In this section, we approach the more realistic, but also more complex, version of the problem that includes the uncertainty that the seller has about the demand.

We approach the solution of the stochastic problem by computing bounds on the values of the profit function $B_k(S)$. These bounds are based on the solution of some modified deterministc problems. Thus, since the deterministic problem is easy to solve under the assumption of regularity, we will assume throughout this section that this condition is satisfied.

We define D_k as the random variable describing the demand at stage k. D_k has a multinomial distribution with parameter q. At each stage k, the seller observe a realization of the demand $d_k \in F(N) \cup \emptyset$ [4] of D_k and decides whether or not to rent period d_k based on the value of S_k. Then given a state variable S_k and a demand d_k at stage k, let $x_k(S_k, d_k)$ be the control variable associated to the decision of renting, ie,

$$x_k(S_k, d_k) = \begin{cases} 1 & \textit{Period } d_k \textit{ is rented at stage } k \textit{ given } S_k \\ 0 & \textit{Otherwise} \end{cases}$$

From section (1), $B_k(S)$ is represented through the recursion:

$$B_k(S_k) = \sum_{f \in S_k} [max\{p_k(f) + B_{k+1}(S_k - C(f)), B_{k+1}(S_k)\} \cdot q(f)] + B_{k+1}(S_k) \sum_{f \notin S_k} q(f)$$

The border conditions are $S_1 = F(N)$, $B_{K+1}(S_{K+1}) = 0$, $\forall S_{K+1} \subseteq F(N)$, and $B_k(\emptyset) = 0$, $\forall 1 \leq k \leq K$.

As we notice before, the DP formulation of the problem is simple and the computation of $B_k(S_k)$ at each stage does not involve complicate operations, however the size of the state space can make the DP formulation inappropriate, even under the assumption of regularity. In fact under this assumption, it can be easily shown that for stage $k \geq \frac{n}{2}$ the size of the sate space (i.e., the number of different subsets S_k) is equal to 2^n. For example, if we are interested to find the optimal renting policy for an horizon of one month (n=30), then the DP formulation requires the storage of more

[4] We use the notation $D_k = \emptyset$ to describe the event when there is no demand in stage k

than a billion of different values of $B_k(\cdot)$ at each stage k ($k \geq 15$). This rapid growth of the state space makes the DP formulation untractable, despite of its simplicity.

In order to solve the problem, a different approach is required. From the seller's perspective, the objective is to maximize the value of $B_1(F(N))$. Since the way that this maximization is achieved is through the selection of an optimal set of control variables $\{x_k(S_k, d_k) : \forall S_k,\ \forall d_k\}$, the optimization problem can be understood as finding an optimal set of control variables.

Now, from the definition of $B_k(\cdot)$, we have the following equivalence

$$\{x_k(S_k, d_k) = 1\} \Longleftrightarrow \left\{ \underbrace{(d_k \in S_k)}_{(a)} \wedge \underbrace{(p_k(d_k) \geq B_{k+1}(S_k) - B_{k+1}(S_k \setminus C(d_k)))}_{(b)} \right\} \tag{2}$$

Condition (a) can be easily checked, however condition (b) requires the knowledge of the values of $B_{k+1}(\cdot)$ that as we mentioned above is not doable in a reasonable amount of time. However, we can partially solve the decision problem if we are able to find good bounds for the expression $B_{k+1}(S_k) - B_{k+1}(S_k \setminus C(d_k))$. In fact, suppose that we can easily find lower and upper bounds $LB_k(S_k, d_k)$ and $UB_k(S_k, d_k)$ respectively, then if $p_k(d_k) \geq UB_k(S_k, d_k)$ it is optimal to set $x_k(S_k, d_k) = 1$, and if $p_k(d_k) \leq LB_k(S_k, d_k)$ then it is optimal to set $x_k(S_k, d_k) = 0$.

3.1 Lower Bounds

3.1.1 Open-Loop Solution

As we mentioned above, solving the original stochastic problem is equivalent to find the values of the binary control variables $\{x_k(S_k, d_k)\}$. In order to do this, we first compute a lower bound for the value of $B_k(S_k)$, $\forall k$, $\forall S_k$.

Suppose that the system is at stage k and the state variable is $S_k \subseteq F(N)$. A lower bound on the value of $B_k(S_k)$ can be found by solving the Open-Loop control problem from stage k onwards, starting at state S_k [5]. For this particular problem, the Open-Loop formulation assumes that the planner selects at the beginning of stage k the set $S \subseteq S_k$ of rental periods that she is willing to rent. The set S satisfies the stable set requierement, i.e., if f_1, $f_2 \in S$ then $f_1 \cap f_2 = \emptyset$. Let $OL_k(S_k)$ be the optimal expected value for the Open-Loop formulation, then we have

$$OL_k(S_k) = \max_{S \subseteq S_k} \left\{ \sum_{f \in S} \left[p(f) \cdot Pr(f \in \bigcup_{i=k}^{K} D_i) \right] \right\}$$
$$s.t. \quad f_i \cap f_j = \emptyset \quad \forall f_i, f_j \in S$$

The above optimization problem has the same structure as the deterministic problem **DRP** presented in section (2). In fact, let $\bar{p}(f) = p(f) \cdot Pr(f \in \bigcup_{i=k}^{K} D_i)$ be the expected profit associated to period f if the decision maker decides to include it in the set S. Then the above optimization problem is equvalent to:

$$OL_k(S_k) = \max_{\{x_i:\, f_i \in S_k\}} \sum_i \bar{p}(f_i) \cdot x_i$$

[5]See for example Bertsekas (1995) for a general description of the Open-Loop control problem associated to a DP formulation.

s.t.
$$x_i + x_j \leq 1 \quad \forall f_i, f_j \in S_k \text{ such that } f_i \cap f_j \neq \emptyset$$
$$x_i \in \{0,1\} \quad \forall i = 1, \ldots, m.$$

This is exactly the same formulation of **DRP**. Therefore under the assumption of regularity, we can find $OL_k(S_k)$ using the greedy algorithm. It must be clear at this point that $OL_k(S_k)$ is a lower bound for $B_k(S_k)$ since the Open-Loop formulation solves a single stage optimization problem and does not take into account the dynamic behavior of the demand process. We formalize this result in the following property that says something more about the assymptotic behavior of $OL_k(S_k)$.

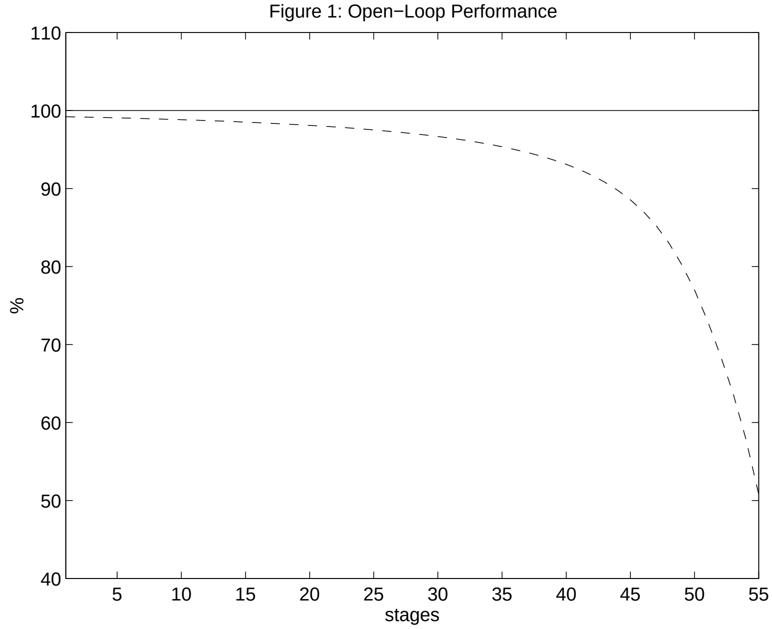
Proposition 5 *For any $k = 1, \ldots, K$ and any $S_k \subseteq F(N)$, $OL_k(S_k)$ is a lower bound for $B_k(S_k)$, i.e., $OL_k(S_k) \leq B_k(S_k)$. In addition, for any fixed k*

$$\lim_{K \to \infty} (B_k(S_k) - OL_k(S_k)) = 0$$

PROOF: See the Appendix. ∎

Given the assymptotic behavior of $OL_k(S_k)$, we expect that the difference $B_k(S_k) - B_k(S_k - C(f)) \approx OL_k(S_k) - OL_k(S_k - C(f))$ for K large enough. Thus, using $OL_k(\cdot)$ as a proxi for $B_k(\cdot)$ will allow to select optimally the control variable $x_k(\cdot,\cdot)$ when the reservation horizon $K - k$ is large enough. This result is presented in the following figure:

Figure 1: Open−Loop Performance



The figure presents the ratio $\frac{OL_k(S)}{B_k(S)}$ for $k = 1, \ldots, 55$, and $S = F(\{1, 2, \ldots, 7\})$ (dotted line). The demand and price parameters were randomly selected. In figure 1, we can see the assymptotic behavior of the Open-Loop solution, however as the lenght of the planning horizon decreases $OL_k(S)$ behaves poorly.

In order to avoid in part the undesirable behavior of $OL_k(\cdot)$ when the reservation horizon is short, we introduce a simple modification to $OL_k(\cdot)$ that leads to a better lower bound.

3.1.2 Modified Open-Loop Solution

The main advantage of the Open-Loop formulation presented above is that requieres to solve a single optimization problem. Moreover, under the regurality assumption this task can be done easily using the greedy algorithm. The main problem, however, is that this single optimization is not able to capture the possibility of making decisions according to the sequence of realizations of the demand process. This is, of course, the main attribute of the DP formulation.

An intermedia solution is to allow the central planner to observe the first realization of the demand process before selecting which are the periods that is willing to rent. This modified version of the Open-Loop formulation is given by:

$$MOL_k(S_k) = \sum_{f \in S_k} [max\{p(f) + OL_{k+1}(S_k - C(f)); OL_{k+1}(S_k)\} \cdot q(f)] + OL_{k+1}(S_k) \sum_{f \notin S_k} q(f)$$

The following property is the analog of proposition (5) for $OL_k(\cdot)$:

Proposition 6 *For any $k = 1, \dots, K$ and any $S_k \subseteq F(N)$, $MOL_k(S_k)$ is a lower bound for $B_k(S_k)$, i.e., $MOL_k(S_k) \leq B_k(S_k)$. In addition, for any fixed k*

$$\lim_{K \to \infty} (B_k(S_k) - MOL_k(S_k)) = 0$$

The proof of this proposition is a straightforward application of the result of proposition (5) and the definition of $MOL_k(\cdot)$. ∎

Figure 2: Modified Open−Loop Performance

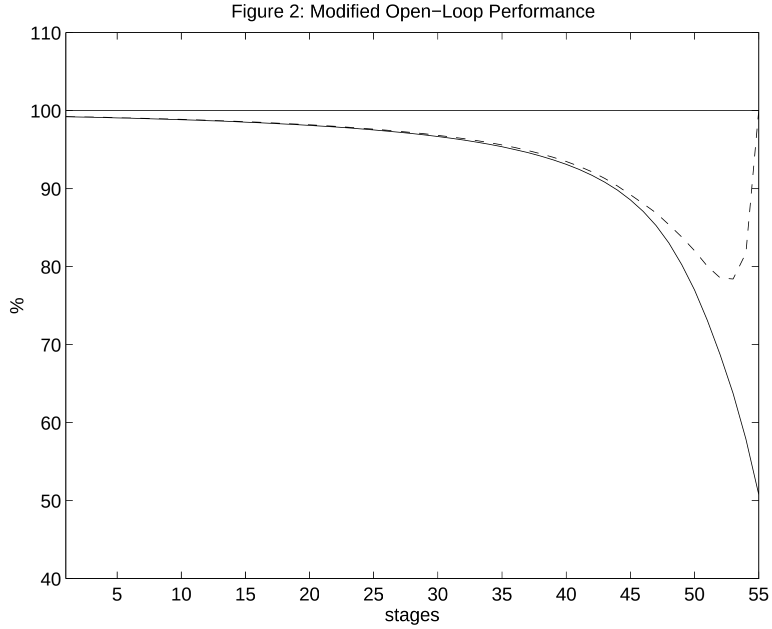


Figure 2 was drawn using the same set of parameters than figure 1. The dotted line corresponds to $MOL_k(\cdot)$ and the straight line is $OL_k(\cdot)$. As we can see, the modified Open-Loop solution has the same type of assymptotic behavior than the original Open-Loop solution. However, when the reservation horizon is short, $MOL_k(\cdot)$ performs better than $OL_k(\cdot)$. In fact, it can be easily seen that at the last stage $B_K(\cdot)$ is always equal to $MOL_K(\cdot)$.

3.2 Upper Bound

Finding an upper bound for $B_k(\cdot)$ is not as simple as finding a lower bound. As we will see in this section, the solution that we propose requires more computational effort than the computation of the lower bounds. However, the computational requirements for this upper bound are still much lower than those required in the DP formulation.

The idea behind this upper bound is still the use of the simple solution that we have for the deterministic case. The following procedure explains how to compute the upper bound $UB_k(\cdot)$ for $B_k(\cdot)$, which is basically a MonteCarlo simulation.

Upper Bound Computation:
Given a set S_k at stage k and an integer J big enough:

1. Generate randonmly a sequence of demand realization $\{D^j\}_{j=1}^{J}$. Each D^j is a $(K-k)$-dimensional vector whose components are the realization of the demand process during the $K-k$ stages that remain in the reservation horizon and that consider only those rental periods f that belong to S_k.
2. For each D^j solve the deterministic version of the problem considering only those rental periods that were demanded in D^j. Let $R(j)$ be the solution obtained using the demand vector D^j.
3. Set $UB_k(S_k) = \frac{\sum_{j=1}^{J} R(j)}{J}$.

Proposition 7 *For any $k = 1, \ldots, K$ and any $S_k \subseteq F(N)$, $UB_k(S_k)$ is an upper bound for $B_k(S_k)$, i.e., $MOL_k(S_k) \geq B_k(S_k)$.*
In addition, for any fixed k

$$\lim_{K \to \infty} (UB_k(S_k) - B_k(S_k)) = 0$$

PROOF: See the Appendix. ∎

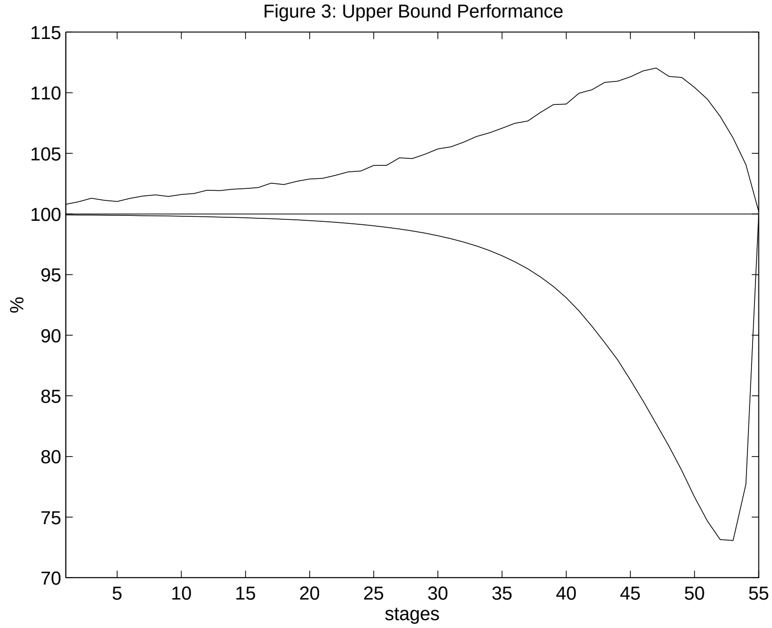
We have again the desirable property of a bound being assymptotically optimal, however the performance of this upper bound deteriorates as the reservation horizon $K-k$ decreases. The following figure presents the relative performance of this upper bound:

Figure 3 plots the ratios $\frac{UB_k(S)}{B_k(S)}$ and $\frac{MOL_k(S)}{B_k(S)}$ for $S = F(\{1, 2, \ldots, 7\})$, randonmly selected parameters for the demand and the rental prices, and $J = 5000$.

4 Numerical Study

As we mentioned at the beginning of section (3), solving the problem from a practical perspective requires the specification of the control variables $\{X_k(S_k, d_k)\}$ rather than the values of the function $B_k(S_k)$.

Figure 3: Upper Bound Performance



From (2), we know that

$$\{x_k(S_k, d_k) = 1\} \iff \{(d_k \in S_k) \wedge (p(d_k) \geq B_{k+1}(S_k) - B_{k+1}(S_k \setminus C(d_k)))\}$$

Then if $d_k \in S_k$ and $p(d_k) \geq UB_{k+1}(S_k) - MOL_{k+1}(S_k \setminus C(d_k))$, then we can set $x_k(S_k, d_k) = 1$. On the other hand, if $d_k \notin S_k$ or $p(d_k) \leq MOL_{k+1}(S_k) - UB_{k+1}(S_k \setminus C(d_k))$, then we can set $x_k(S_k, d_k) = 0$. However, for those cases where both situations do not hold, the bounds can not be used in a clear way to define the value of the control variables. For this reason, in this section we analyze numerically three heuristics than are derived from the values of the bounds.

The first heuristic (HEUR1) sets the value of the control in the following way:

$$\{x_k(S_k, d_k) = 1\} \iff \{(d_k \in S_k) \wedge (p(d_k) \geq MOL_{k+1}(S_k) - MOL_{k+1}(S_k \setminus C(d_k)))\}$$

In other words, HEUR1 replaces the optimal values of $B_k(\cdot)$ by those of $MOL_K(\cdot)$ to decide the values of the control variables.

The second heuristic (HEUR2) operates in the same way that HEUR1 but using the values of $UB_k(\cdot)$ instead of $MOL_k(\cdot)$. Finally, the third heuristic (HEUR3) is defined in same terms than the two previous but considering the arithmetic average of $UB_k(\cdot)$ and $MOL_k(\cdot)$.

Table 1 presents the average relative error of these three heuristics. Each element of the table was computed as the average of 10.000 simulations of the demand process.

As we can see on the table, HEUR1 is significantly better than the other two. The maximum error during those simulation for HEUR1 is less than 3%.

Table 1

n	HEUR1	HEUR2	HEUR3
2	98.30%	85.43%	85.43%
3	100%	53.85%	100%
4	97.75%	69.51%	77.21%
5	99.36%	48.63%	83.75%
6	98.95%	66.33%	94.74%
7	100.11%	50.18%	98.36%
8	100.33%	44.80%	99.09%
9	99.48%	63.23%	NA
10	100.01%	50.92%	87.81%
11	99.63 %	NA	NA
12	98.50%	NA	NA

(The values of the table correspond to the initial condition $S_1 = F(\{1, \ldots, n\})$.)

References

[1] BERTSEKAS D.(1995), Dynamic Programming and Optimal Control (Vol I), Athena Scientific, Belmont, Massachusetts.

[2] GAREY M. AND JOHNSON D. (1979), Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness, Freeman, San Francisco.

[3] LEKKERKERKER C. AND BOLAND J. (1962), Representation of a Finite Graph by a Set of Intervals on the Real Line, Fundamenta Mathematicae 51, 45-64.

[4] NEMHAUSER G. AND L. WOLSEY (1988), Integer and Combinatorial Optimization, Wiley–Interscience Series in Discrete Mathematics and Optimization.

Appendix

PROOF OF PROPOSITION 2

To prove the $\mathcal{NP}$-completeness of the Rental Problem, we exhibit a polynomial-time transformation of the Stable Set problem into the Rental Problem.

Consider a graph $G = (V, A)$ with $|V| = n$ and $|A| = m$. Let also consider a weight function $w : V \to Q^+$ and a positive integer B. The Stable Set decision problem is to decide whether or not there is a stable set $\tilde{V} \subseteq V$ such that $\sum_{v \in \tilde{V}} w_v \geq B$.

The first step of the transformation is to define the ground set E for the rental problem. For each arc $a \in A$, we associate an element $e_a \in E$ (thus, $|E| = |A| = m$). The next step is the definition of the collection C of subsets of E. For each node $v \in V$, we define a set $C_v \in C$ such that $e_a \in C_v$ if and only if a is an incident edge to node v in G. We set $C = \bigcup_{v \in V} C_v$, and we notice that $|C| = |V| = n$. Finally, we introduce the weight function $\bar{w} : C \to Q^+$ such that $\bar{w}(C_v) = w_v$.

The definition of E requires to check once all the arcs in A. The definition of C requires to check once all the nodes in V and for each nodes the set of incident edges. Finally, the definition of $\bar{w}$ requires the assignment of $|V|$ values. Thus, the three steps of the transformation above can be carried out in polynomial time.

Suppose that we are able to solve the Rental problem in polinomial time. That is, find a subcollection $\tilde{C} \subseteq C$ such that any two different elements $C_{v_1}, C_{v_2} \in \tilde{C}$ have empty intersection, $C_{v_1} \cap C_{v_2} = \emptyset$, and $\sum_{c \in \tilde{C}} \bar{w}(c) \geq B$, or assert that there is not such a subcollection. By the definition of C we know that there is a biyection between the elements of C and the element of V. We then define a stable set $\tilde{V}$ in the following way: $v \in \tilde{V} \iff C_v \in \tilde{C}$. Given this definition of $\tilde{V}$, we have that $\sum_{v \in \tilde{V}} w_v = \sum_{C_v \in \tilde{C}} \bar{w}(c) \geq B$. Thus, it only remains to show that $\tilde{V}$ is in fact a stable set. Suppose that this is not the case, i.e., exist $v_1, v_2 \in \tilde{V}$ adjacent nodes in G. Then, there exists an arc $a \in A$ that connects v_1 and v_2. Since a is incident to v_i, by construction we have $e_a \in C_{v_i}$ $(i = 1, 2)$. Then $C_{v_1} \cap C_{v_2} \neq \emptyset$, but this result violates the definition of $\tilde{C}$, since C_{v_1} and C_{v_2} belong to $\tilde{C}$. From this contradiction we conclude that $\tilde{V}$ is a stable set. ∎

PROOF OF PROPOSITION 3

Let us consider a cycle of nodes $\{f_1, f_2, ..., f_s\}$, with $s \geq 4$ (we represent node f_i by $r_{k_i l_i}$). Given the regularity assumption, and the fact that if they are connected by an arc then they have at least one day in common, two consecutive nodes in the cycle, $r_{k_i l_i}$ and $r_{k_{i+1} l_{i+1}}$, must satisfy one of the following two conditions:

1. One rental period is a subset of the other.
2. The rental periods overlap only in their extreme days; for example, one period could be $\{3, 4, 5, 6\}$ and the other $\{5, 6, 7, 8, 9\}$.

Case 1: Two consecutive days $f_i = r_{k_i l_i}$ and $f_{i+1} = r_{k_{i+1} l_{i+1}}$ satisfy condition 1. Without loss of generality, we assume that $k_i \leq k_{i+1}$ and $l_i \geq l_{i+1}$, i.e. $f_{i+1} \subseteq f_i$. Additionally, we know that nodes f_{i+1} and f_{i+2} have at least one day in common (they are consecutive nodes in the cycle), which must also belong to f_i ($f_{i+1} \subseteq f_i$). Thus, nodes f_i and f_{i+2} have also at least one day in common and there is an arc in $G(\tilde{F}, A)$ between them. Therefore, this arc is a chord in the cycle.

Case 2: All nodes in the cycle satisfy condition 2 (no pair of nodes satisfies condition 1). Let us denote by A_i the arc that connects nodes f_i and f_{i+1} (A_s is the arc connecting nodes f_s and f_1). We define A_i to be a positive arc, if the upper extreme of f_i overlaps with the lower extreme of f_{i+1}, and a negative arc if the lower extreme of f_i overlaps with the upper extreme of f_{i+1}.

Case 2.1: The cycle contains positive and negative arcs. Therefore, there must exist two consecutive arcs A_i and A_{i+1} such that one is negative and the other positive. Without loss of generality, we assume that A_i is positive and A_{i+1} is negative. Given that A_i is positive, f_i and f_{i+1} have at least the first day of f_{i+1} in common (day k_{i+1}, which is the lower extreme of f_{i+1}). Additionally, because A_{i+1} is negative, then f_{i+1} and f_{i+2} overlap at least in the lower extreme of f_{i+1}, i.e. day k_{i+1}. Thus, f_i and f_{i+2} overlap at least in day k_{i+1}. Therefore, there exists a chord in the graph that connects nodes f_i with f_{i+2}.

Case 2.2: All arcs in the cycle are of the same sign (all arcs are positive or negative). Without loss of generality, we assume that all arcs are positive. Then, we have that $l_i < l_{i+1}, \forall i = 1, .., s$, where $l_{s+1} = l_1$. However, this condition leads to the condition $l_1 < l_1$, which is clearly a contradiction. Therefore, there are no cycles with all arcs of the same sign. ∎

PROOF OF PROPOSITION 5

Let first show that $OL_k(S_k) \leq B_k(S_k)$, $\forall k$, $\forall S_k$. In fact, since $B_k(S_k)$ is computed using a DP formulation, let introduce some additional notation that will help us to formulate the optimization problem associated to $B_k(S_k)$ in simple terms:

- The dynamic of the system $S_{k+1} = f_k(S_k, x_k, d_k)$ are given by

$$f_k(S_k, x_k, d_k) = \begin{cases} S_k & if\, x_k = 0,\, or\, d_k \notin S_k \\ S_k - C(d_k) & Otherwise \end{cases}$$

  where, d_k is the realization of the demand D_k at stage k, and x_k is the control variable that decides whether or not d_k is rented at stage k given the the state variable S_k.

- For each k, $\mu_k(S_k, d_k) \in \{0, 1\}$ represents the control policy that decides whether or not to rent period d_k at stage k given a state variable S_k.

Given this definitions, $B_k(S_k)$ can be found solving

$$\begin{aligned} B_k(S_k) = \max_{\mu_k,\ldots,\mu_K} & \left\{ E_{\{d_j:\, j=k,\ldots,K\}} \left[\sum_{j=k}^{K} p(d_j) \cdot \mu_j(S_j, d_j) \right] \right\} \\ s.t. \quad & S_{j+1} = f_j(S_j, \mu_j(S_j, d_j), d_j) \end{aligned}$$

On the other hand, $OL_k(S_k)$ is computed assuming that the decision maker selects at the beginning of stage k which are the periods that she will be willing to rent, we have that

$$\begin{aligned} OL_k(S_k) = \max_{\mu_k,\ldots,\mu_K} & \left\{ E_{\{d_j:\, j=k,\ldots,K\}} \left[\sum_{j=k}^{K} p(d_j) \cdot \mu_j(S_j, d_j) \right] \right\} \\ s.t. \quad & S_{j+1} = f_j(S_j, \mu_j(S_j, d_j), d_j) \\ & \mu_j(S_j, d_j) = \mu_k(S_k, d_j), \;\; if\; d_j \in S_j \end{aligned}$$

As we can see, $OL_k(S_k)$ is obtained optimizing over a set of policies $\{\mu_k, \ldots, \mu_K\}$ that is more restricted than the set of policies used for computing $B_k(S_k)$. Therefore, $OL_k(S_k) \leq B_k(S_k)$.

The proof of the assymptotic behavior of $OL_k(S_k)$ is carried out in two steps. First we found and upper bound for $B_k(S_k)$. In the second step, we show that the value of $OL_k(S_k)$ converges to the value of the upper bound as $K \to \infty$.

The upper bound on the value of $B_k(S_k)$ can be found by considering the situation where all the rental periods in S_k with $q(f) > 0$ will be demanded with probability one during the reservation horizon $\{k, \ldots, K\}$. Clearly, under this scenario, the decision maker does not have any uncertainty about the behavior of the demand process. Thus, she solves the problem of selecting those periods in S_k that maximizes her profits. This solution is, of course, an upper bound on $B_k(S_k)$ since in reality the decision maker faces uncertainty on the demand process. The value of the upper bound is given by

$$\begin{aligned} U_k(S_k) = \max_{\{x_i:\, f_i \in S_k,\, q(f)>0\}} \ & \textstyle\sum_i p(f_i) \cdot x_i \\ & \text{s.t.} \\ & x_i + x_j \leq 1 \quad \forall f_i, f_j \in S_k \text{ such that } f_i \cap f_j \neq \emptyset \\ & x_i \in \{0,1\} \quad \forall i = 1, \ldots, m. \end{aligned}$$

and we have $OL_k(S_k) \leq B_k(S_k) \leq U_k(S_k)$. Now, in order to show that $lim_{K\to\infty} U_k(S_k) - OL_k(S_k) = 0$, we make use of the representation of $OL_k(S_k)$ presented in section (3):

$$\begin{aligned} OL_k(S_k) = \max_{\{x_i:\, f_i \in S_k\}} \ & \textstyle\sum_i \bar{p}(f_i) \cdot x_i \\ & \text{s.t.} \\ & x_i + x_j \leq 1 \quad \forall f_i, f_j \in S_k \text{ such that } f_i \cap f_j \neq \emptyset \\ & x_i \in \{0,1\} \quad \forall i = 1, \ldots, m. \end{aligned}$$

where $\bar{p}(f) = p(f) \cdot Pr(f \in \bigcup_{i=k}^{K} D_i)$. Given the multinomial distribution of D_j we have that $Pr(f \in \bigcup_{j=k}^{K} D_j) = 1 - (1 - q(f))^{K-k}$. Thus, for any f such that $q(f) > 0$, $\lim_{K\to\infty} Pr(f \in \bigcup_{j=k}^{K} D_j) = 1$ and $\bar{p}(f) = p(f)$. Comparing the definition of $U_k(S_k)$ and $OL_k(S_k)$, this observation implies $\lim_{K\to\infty} U_k(S_k) - OL_k(S_k) = 0$.

Finally, from the relation $OL_k(S_k) \leq B_k(S_k) \leq U_k(S_k)$, we conclude $\lim_{K\to\infty} B_k(S_k) - OL_k(S_k) = 0$. ∎