When to Renew Software Licences at HPC Centres?
A Mathematical Analysis

Ge Baolai\textsuperscript{1,1} and Allan B. MacIsaac\textsuperscript{2}
\textsuperscript{1} SHARCNET, The University of Western Ontario, London, Ontario, Canada N6A 5B7
\textsuperscript{2} Department of Applied Mathematics, The University of Western Ontario, London, Ontario, Canada N6A 5B7
E-mail: bge@sharcnet.ca

Abstract. In this paper we study a common problem faced by many high performance computing (HPC) centres: When and how to renew commercial software licences. Software vendors often sell perpetual licences along with forward update and support contracts at an additional, annual cost. Every year or so, software support personnel and the budget units of HPC centres are required to make the decision of whether or not to renew such support, and usually such decisions are made intuitively. The total cost for a continuing support contract can, however, be costly. One might therefore want a rational answer to the question of whether the option for a renewal should be exercised and when. In an attempt to study this problem within a market framework, we present the mathematical problem derived for the day to day operation of a hypothetical HPC centre that charges for the use of software packages. In the mathematical model, we assume that the uncertainty comes from the demand, number of users using the packages, as well as the price. Further we assume the availability of up to date software versions may also affect the demand. We develop a renewal strategy that aims to maximize the expected profit from the use the software under consideration. The derived problem involves a decision tree, which constitutes a numerical procedure that can be processed in parallel.

1. Introduction
In this paper we study a common problem faced by the personnel at many high performance computing (HPC) centres: When and how to renew commercial software licences. Software vendors often sell perpetual licences along with forward update and support contracts at an additional, annual cost, which cover not only technical support, but also the entitlement for future upgrades. The HPC providers have the option to renew or not to renew the update contract for a particular software package, depending on the demand for the software and budget. They may also opt to renew the support contract for multiple years at a discounted price. By paying a fee for the update contract, an HPC provider can protect itself against any incompatibilities that might occur due to the upgrade of their software environment on the deployed systems, such as the operating system, system libraries and compilers. The total cost for the continuing support contract can, however, be costly. One might therefore want a rational answer to the question of whether the option for a renewal should be exercised and when.

In an attempt to study this problem within a market framework, aiming at the self-adjustment of resource supply and demand, we present the analysis of a mathematical problem derived from
the day to day operation of a hypothetical HPC centre that charges fees for using resources and software. In the mathematical analysis, we assume that the uncertainty comes from the demand (number of users using a package) as well as the price. Further we assume the availability of the up to date software versions may affect the demand, hence the usage of the resources as well. We develop a software support renewal strategy that aims to maximize the expected profit from the usage of resources.

The derived mathematical problem involves a decision tree, which constitutes a numerical optimization problem that is best solved with parallel processing.

2. Related Work
The hardware and software upgrade problem has been studied from a number of different perspectives. In the literature, related, representative work includes the work of Pfening and Telek [1], which aimed to find an optimal maintenance policy for a slowly degrading system, Zhao et al [2] which considered the problem of the optimal rejuvenation schedule for an application server, Hartman and Rogers [3] which presented two dynamic programming models for equipment replacement problems and Mehra and Seidmann [4] which considered the problem of finding an optimal timing for software upgrade releases for a monopoly software maker. The nature of the problem under consideration here belongs to a boarder class of problems - revenue management, which has been extensively studied (See for example the classic works [5] and [6]).

A problem that is closely related to what we are studying is the application of the real options approach (See for example [7]) to refurbishment of hotels and shopping malls (See for example, [8], [9] and the recent work [10]). In hotel renovation problems, the quality upgrade schedule is to be determined over the life span of a hotel. In our case we use the same methodology applied to a new problem. To our knowledge, the problem we are considering is unique with respect to the objectives we attempt to achieve. We consider an HPC resource provider, in which a charging model is used: for every user, the usage of resources is billed in credits or internal currency. We present a mathematical model based on stochastic analysis and devise a renewal strategy that aims at maximizing the profitability of the HPC provider in terms of either real money or internal currency.

3. The Problem Statement
We devise our models based on a number of assumptions; Resources (i.e. time on CPU cores) have values, the resource provider makes a profit by offering available resources to users for the return of currency or credits, and the price of a resource is determined by the demand and may exhibit stochastic motion.

To begin, we introduce the following notation:

- $t$ – Time, with the notation $t + k$ denoting the $k$th renewal time in years;
- $q_k$ – A short notation for time dependent function $q(t)$ at $t = k$;
- $T$ – The end life span, e.g. 5 years.
- $u(t)$ – Demand (number of users);
- $p(t)$ – Price per use;
- $C(t)$ – Cost of renewal;
- $\omega$ – Renewal “path”, i.e. a sequence of decisions on whether to renew or not to renew at a number of specified times in year;
- $w_k$ – For a particular renewal path at the $k$th year, an integer that indicates the number of years to renew for.
- $f(t, u, p; \omega)$ – The “cash flow” rate at time $t$, determined by the demand $u(t)$, price $p(t)$, and renewal path $\omega$. 
- $V$ – Expected profit.

By making a sequence of decisions at renewal times $t_1, t_2, \ldots$, we attempt to solve the problem

\[
\max_{\omega} \left\{ \int_0^T f(t, p, u; \omega) dt - \sum_{k=0}^T w_k C_k \right\},
\]

over all possible renewal paths $\omega$ subject to certain constraints in $p$ and $u$ and

\[
\sum_k w_k \leq T.
\]

We shall apply Bellman’s principle of optimality [11] to solve this optimization problem backward in time.

In the following sections we consider two models: a simplified, deterministic model that provides an intuitive look at the problem and a stochastic model, for which we explore the solutions within the framework of a Monte Carlo approximation.

4. Deterministic Model - A Simplified Case

We first consider a simplified model. To illustrate the method, suppose the time frame to be considered is divided into five intervals which might represent a typical life span of five years. During this time frame, we face a sequence of decisions to make. This can be illustrated by an ensemble of possible upgrade paths, as shown in Figure 1. An example of five years of coverage: $w = \{1, 2, 0, 0\}$ is represented by the line in bold. In this example at the end of year one the contract is renewed for one year. At the end of the year two the contract is renewed for two years. No renewal is needed at the end of year three and at the end of year four the renewal for year five is declined. The first year is assumed to be covered by the initial contract.

![Figure 1. Upgrade “paths” in years of coverage. An example of coverage: $w = \{1, 2, 0, 0\}$ is represented by the line in bold. The first year is covered by the initial contract.](image)

An intuitive, tree representation is given in Figure 2, again for a five year period. Each circled number indicates the number of years that would have been covered by a support contract. Each of the triangles on the right resembles the branch immediately left to it at the same level. Our algorithm, which recursively determines the renewal paths, is given below

**Algorithm 1** Computing renewal paths. Given the number of years $n$–the life span– of renewal opportunities, the following procedure determines all the possible renewal paths, stored in two dimensional array $\omega(:, :)$. 

Assume we know the pattern of demand $u(t)$ and that demand drops following a known path when an annual renewal is skipped and hence the upgrades are frozen. When the support contract is renewed, the demand will increase to or remain at the level at the end of previous support contract.

We derive our mathematical formulation by carrying out our analysis backward in time.
starting at \( t = T = 5 \) as follows. Let \( U_k^t(u, p) \) be the revenue from the use of the software of \( k \) years old and \( \bar{u}_k^t \) be the average use over interval \([t, t+1]\). Assume, for simplicity, over the entire time period, the price \( p \) for use remains constant, as does the cost of renewal, denoted by \( c \).

Now at \( t = 4 \), we denote the expected profit by \( V_k^4 \), for \( k \) year old software, with and without a renewal, by the end of the interval \([4, 5]\), when looking forward at \( t \). Note that, it is a common practise that a vendor will only renew the software to the current release and therefore will charge backward the missing upgrades, due to the cost of software development. That is, if the cost for annual renewal is \( c \), the cost to renew a \( k \) years old software will be \( kc \). Therefore we write, denoting \( E(\cdot) \) the expectation operator

\[
V_k^4 = \max \{ E(U_k^4(u, p)), E(T_4^0(u, p)) - kc \}
= \max \{ \bar{u}_k^4 p, \bar{u}_0^4 p - kc \}.
\]

At \( t = 3 \), we have these options, not to renew, renew for one year or renew for two years. Therefore, we have\(^2\)

\[
V_k^3 = \max \{ \bar{u}_3^k p + V_4^k + \bar{u}_0^3 p - kc + V_4^0, 2\bar{u}_3^0 p - (k + 1)c \}.
\]

We iterate this analysis until \( t = 1 \), at which we have

\[
V_1^0 = \max \{ \bar{u}_1^1 p + V_2^1, \bar{u}_1^0 p - c + V_2^0, 2\bar{u}_0^0 p - 2c + V_3^0, 3\bar{u}_1^0 p - 3c + V_4^0, 4\bar{u}_1^0 p - 4c \}.
\]

In summary, knowing \( V_k^t, t = 4, \ldots, 1, k = 0, 1, \ldots, t - 1 \), we can find backward in time the path that leads to the maximum of expected profit, subject to the assumptions we have made on the demand and its response to the upgrade decisions.

5. Stochastic Model

We now assume in Eq. (1) the demand \( u = u(t) \) follows a stochastic process, as does the price \( p(t) \). We use the expectation operator \( E_t \) given the information available at \( t \), and rewrite Eq. (1) as

\[
\max_{\omega} \left\{ E_0 \left[ \int_0^T f(\tau, p, u; \omega) d\tau - \sum_{k=0}^T w_k C_k \right] \right\},
\]

where \( C_k \) is the cost for renewal at the \( k \)th year and \( w_k \) are control parameters

\[
w_k = \begin{cases} m & \text{if renewed for } m \text{ years,} \\ 0 & \text{otherwise;} \end{cases}
\]

subject to the demand equation

\[
du = a(t, u; \omega) dt + b(t, u; \omega) dX,
\]

where \( a \) and \( b \) are known, deterministic functions and \( X \) is a stochastic process. To simplify the problem further, we assume the price is constant and consider only the demand \( u \).

\(^2\) Note that for a renewal that covers two years, for which the software is kept up to date, we assume the same average uses \( \bar{u}_3^0 \) and \( \bar{u}_0^1 \). The same applies to the expression of \( V_1^0 \) below as well.
Using the notation for renewal times as in the previous section, from (3), define

\[
V(t, u(t)) = \max_\omega \left\{ E_t \left[ \int_t^T f(\tau, u; \omega) d\tau - \sum_{k=t}^T w_k C_k \right] \right\}
\]

\[
= \max_\omega \left\{ E_t \left[ \int_t^{t+1} f(\tau, u; \omega) d\tau - w_t C_t + \int_{t+1}^T f(\tau, u; \omega) d\tau - \sum_{k=t+1}^T w_k C_k \right] \right\}.
\]

Denoting

\[
\phi(t, u(t)) = \int_t^{t+1} f(\tau, u; \omega) d\tau,
\]

\[
\phi(t, u(t)) \text{ now represents the realization of the cash flow for the period of } [t, t+1] \text{ when the uncertainty is revealed at } t.
\]

We arrive at the following recursive equation

\[
V(t, u(t)) = \phi(t, u(t)) + \max_\omega E_t \left[ V(t+1, u(t+1)) - w_tC_t \right].
\]  

(7)

The Bellman’s principle is interpreted as follows: Upon knowing the demand \( u \), which is a function of some uncertainty \( X \) revealed at \( t \), one is to make strategic decisions to maximize the expected value of \( V \) by \( t = T \).

To obtain the solution to (7), which involves a series of sub-problems represented by the recursive relation, we use a procedure backward in time. If we know \( E_t [V(t+1, u(t+1))] \), then we can get \( V(t, u) \) and subsequently, we can move backwards in time.

Let

\[
\psi(t, u(t)) = E_t [V(t+1, u(t+1))],
\]

thus

\[
V(t, u(t)) = \phi(t, u(t)) + \max \{ \psi(t, u(t)) - w_tC_t \}.
\]  

(9)

In the following section, we will elaborate on a numerical procedure to solve (9) based upon the least squares Monte Carlo (LSM) approach proposed by Longstaff and Schwartz [12].

6. Numerical Procedure

In practice, an analytic form of \( \psi \) is nearly impossible to derive, hence we seek a numerical solution instead. The idea is to approximate the expected value \( \psi(t, u(t)) \) when the uncertainty is revealed at \( t \). To achieve this goal, we generate \( N \) samples of simulated demands \( U^i(t) \) and \( U^i(t-1), i = 1, 2, \ldots, N \), (also denoted for short \( U^i_t \) and \( U^i_{t-1} \), respectively) using the demand model \( u(t) \) at each \( t \). For each \( U^i_t \), we have a corresponding realization

\[
\phi^i_t = \phi(t, U^i(t))
\]

and

\[
\psi^i_t = \psi(t, U^i(t)) = E_t \left[ V(t+1, U^i(t+1)) \right].
\]  

(10)

Given \( \psi^i_t \), we obtain from (9) \( V(t, U^i_t) \), denoted by \( V^i_t \), where

\[
V^i = \phi^i_t + \max \{ \psi^i_t - w_tC_t \}.
\]  

(11)

We then approximate the expected value \( \psi(t, u(t)) \) by regressing \( V^i \) on \( U^i \) at \( t - 1 \). We can then obtain \( V(t-1, U^i_{t-1}) \) and so on in the same manner. The maximization of \( V^i \) for all simulated sample paths determines the optimal renewal process at time \( t \).

We perform this procedure for each renewal path from \( t = T \). Note that from (8) \( \psi(T, u(T)) = 0 \). The following algorithm summarizes the numerical procedure.
Algorithm 2 Approximating $\psi(t - 1, u(t - 1))$ given $\psi(t, u(t))$.

\[
\psi(T, u(T)) \leftarrow 0; \\
t \leftarrow T; \\
\text{while } t - 1 > 0, \text{ do} \\
\quad \text{for } i = 1, \ldots, N, \text{ do} \\
\quad \quad \text{Generate } U^i_{t-1}; \\
\quad \quad V^i \leftarrow \phi(t, U^i_t) + \max\{\psi(t, U^i_t) - mC_t\}; \\
\quad \quad \text{Regress } V^i \text{ on } U^i_{t-1} \text{ to obtain approximated } \psi(t - 1, u(t - 1)); \\
\quad \text{end} \\
\quad t \leftarrow t - 1; \\
\text{end}
\]

To find the approximate solution to the expected value $\psi(t - 1, u(t - 1))$, we assume a functional form of $\psi(t - 1, u(t - 1))$ spanned by a linear combination of $d$ selected basis functions on demand $U$

\[
\psi = \sum_{k=0}^{d} a_k L_k(U),
\]

where $L_k()$ are the basis functions and $a_k$ are coefficients to be determined. With generated samples $U^i_{t-1}$ at $t - 1$, and computed values $V^i$, we obtain the following overdetermined linear system in $a_k$

\[
V^i = \sum_{k=0}^{d} a_k L_k(U^i(t - 1)), \quad i = 1, \ldots, N.
\]

The coefficients $a_k$ are solved using a standard least squares fitting method.

7. Numerical Results

To highlight the utility of this analysis we present some representative sample calculations in the following section. In the first case, we derive the optimal renewal strategy using the deterministic model of Section 4. In the second example, we present two cases with results obtained from numerical simulation using the stochastic method of Section 5.

7.1. Deterministic Example

To implement our deterministic model, we first generate yearly demand samples $U_1, U_2, \ldots, U_T$, based on the mean of daily demand of 30 observed (from real logged data). We assign the values $U_1, U_2, \ldots, U_T$ at $t = 1, 2, \ldots, T$, respectively. During the interval between the start of years $t$ and $t + 1$, the demand is approximated as a linear function; this interval will be written as $[t, t + 1]$. We assume a renewal decision at time $t$ will affect the subsequent demand during the interval $[t, t + 1]$. A drop in the demand (in this example by 10%) is assumed at $t$ when a renewal is skipped. By contrast, at $t$, if a renewal is made, the demand will be raised to the same level as that in the previous year. In addition we assume there is no penalty for skipping renewals.\(^3\)

The optimal renewal decision is then relatively straightforward to determine as we calculate the net income along all possible paths. The optimal renewal policy is thus the path of renewals that results in the maximal net income. When the number of years to consider increases, however, the number of renewal options grows exponentially. The work to compute the net

\(^3\) This might not be realistic for some commercial software packages, for which certain penalties often apply if the renewal is not made continuously.
Table 1. Deterministic case. Renewal paths and expected revenue (net income) for the price \( p = $0.5 \) per use, mean of daily demand 30 and the cost to renew $1000 with annual increase of 10% for a period of 5 years. The optimal renewal policy is found to be \( \{0 \ 1 \ 0 \ 0 \ 0\} \).

| No. | Path (years to cover at renewal times) | Expected value of net income ($) |
|-----|--------------------------------------|----------------------------------|
| 1   | 4 0 0 0 0                             | 48175.25                         |
| 2   | 3 0 0 1 0                             | 49102.85                         |
| 3   | 3 0 0 0 0                             | 49175.25                         |
| 4   | 2 0 2 0 0                             | 49077.85                         |
| 5   | 2 0 1 1 0                             | 49589.65                         |
| 6   | 2 0 1 0 0                             | 50177.85                         |
| 7   | 2 0 0 1 0                             | 50102.85                         |
| 8   | 2 0 0 0 0                             | 50175.25                         |
| 9   | 1 3 0 0 0                             | 48641.80                         |
| 10  | 1 2 0 1 0                             | 49619.40                         |
| 11  | 1 2 0 0 0                             | 49691.80                         |
| 12  | 1 1 2 0 0                             | 48968.10                         |
| 13  | 1 1 1 1 0                             | 49479.90                         |
| 14  | 1 1 1 0 0                             | 50068.10                         |
| 15  | 1 1 0 1 0                             | 50669.40                         |
| 16  | 1 1 0 0 0                             | 50741.80                         |
| 17  | 1 0 2 0 0                             | 50077.85                         |
| 18  | 1 0 1 1 0                             | 50589.65                         |
| 19  | 1 0 1 0 0                             | 51177.85                         |
| 20  | 1 0 0 1 0                             | 51102.85                         |
| 21  | 1 0 0 0 0                             | 51175.25                         |
| 22  | 0 3 0 0 0                             | 49537.35                         |
| 23  | 0 2 0 1 0                             | 50514.95                         |
| 24  | 0 2 0 0 0                             | 50857.35                         |
| 25  | 0 1 2 0 0                             | 49863.65                         |
| 26  | 0 1 1 1 0                             | 50375.45                         |
| 27  | 0 1 1 0 0                             | 50963.65                         |
| 28  | 0 1 0 1 0                             | 51564.95                         |
| 29  | 0 1 0 0 0                             | 51637.35                         |
| 30  | 0 0 2 0 0                             | 50501.85                         |
| 31  | 0 0 1 1 0                             | 51013.65                         |
| 32  | 0 0 1 0 0                             | 51601.85                         |
| 33  | 0 0 0 1 0                             | 51526.85                         |
| 34  | 0 0 0 0 0                             | 51599.25                         |

income becomes tedious. However one can easily create computer code to generate the complete set of renewal paths by implementing Algorithm 1, and calculating the corresponding net incomes.

Table 1 shows the results for the case where the price per use is $1, the mean of daily demand is 30 and the cost to renew is $1000 with annual increase of 10%. The optimal renewal path is found by computing the maximum net income from the revenue generated less the total cost of renewals.
Table 2. Results for the non deterministic mode. Renewal paths and expected revenue (net income) for the price \( p = \$0.5 \) per use, mean of daily demand 30 and the cost to renew \$1000 with annual increase of 10% for a period of 5 years. The optimal renewal policy is found to be \( \{1 3 0 0 0 \} \).

| No. | Path (years to cover at renewal times) | Expected value of net income ($) |
|-----|--------------------------------------|---------------------------------|
| 1   | 4 0 0 0 0                             | 18039.50                        |
| 2   | 3 0 0 1 0                             | 18366.50                        |
| 3   | 3 0 0 0 0                             | 13555.00                        |
| 4   | 2 0 2 0 0                             | 18889.50                        |
| 5   | 2 0 1 1 0                             | 18805.27                        |
| 6   | 2 0 1 0 0                             | 13960.50                        |
| 7   | 2 0 0 1 0                             | 13910.50                        |
| 8   | 2 0 0 0 0                             | 9123.50                         |
| 9   | 1 3 0 0 0                             | **19466.00**                    |
| 10  | 1 2 0 1 0                             | 19325.79                        |
| 11  | 1 2 0 0 0                             | 14553.00                        |
| 12  | 1 1 2 0 0                             | 19263.14                        |
| 13  | 1 1 1 1 0                             | 19226.70                        |
| 14  | 1 1 1 0 0                             | 14414.00                        |
| 15  | 1 1 0 1 0                             | 14364.00                        |
| 16  | 1 1 0 0 0                             | 9581.00                         |
| 17  | 1 0 2 0 0                             | 14453.00                        |
| 18  | 1 0 1 1 0                             | 14314.00                        |
| 19  | 1 0 1 0 0                             | 9531.00                         |
| 20  | 1 0 0 1 0                             | 9481.00                         |
| 21  | 1 0 0 0 0                             | 4601.00                         |
| 22  | 0 3 0 0 0                             | 15035.50                        |
| 23  | 0 2 0 1 0                             | 14929.00                        |
| 24  | 0 2 0 0 0                             | 10098.00                        |
| 25  | 0 1 2 0 0                             | 14945.00                        |
| 26  | 0 1 1 1 0                             | 14841.50                        |
| 27  | 0 1 1 0 0                             | 10014.00                        |
| 28  | 0 1 0 1 0                             | 9964.00                         |
| 29  | 0 1 0 0 0                             | 5082.50                         |
| 30  | 0 0 2 0 0                             | 9998.00                         |
| 31  | 0 0 1 1 0                             | 9914.00                         |
| 32  | 0 0 1 0 0                             | 5032.50                         |
| 33  | 0 0 0 1 0                             | 4982.50                         |
| 34  | 0 0 0 0 0                             | 4982.50                         |

7.2. Monte-Carlo Example

To illustrate the Monte Carlo method of section 5 we simulate the case of a software package with a daily mean usage of 30 units for a period of five years. Once again we assume there is an annual increase in the cost of renewal of 10%. As in the previous example, we assume there is no penalty when the renewals are skipped. We also assume, exactly as in the previous example, that the demand is affected by the renewals or non-renewals. For each renewal path, at each time of renewal, 10,000 samples of demand are used.

For the least squares Monte Carlo simulation, we have compared the results using a number of
Table 3. Results for the non-deterministic mode. Renewal paths and expected revenue (net income) for the price $p = \$1$ per use, daily mean of demand 30 and the cost to renew $\$1000$ with annual increase of 10% for a period of 5 years. The optimal renewal policy is found to be \{13000\}.

| No. | Path (years to cover at renewal times) | Expected value of net income ($) |
|-----|--------------------------------------|----------------------------------|
| 1   | 40000                                | 40079.00                         |
| 2   | 30100                                | 40883.00                         |
| 3   | 30000                                | 30110.00                         |
| 4   | 20200                                | 41979.00                         |
| 5   | 20110                                | 41861.00                         |
| 6   | 20100                                | 31021.00                         |
| 7   | 20010                                | 30971.00                         |
| 8   | 20000                                | 20247.00                         |
| 9   | 13000                                | 43082.00                         |
| 10  | 12010                                | 42902.00                         |
| 11  | 12000                                | 32206.00                         |
| 12  | 11200                                | 42776.00                         |
| 13  | 11110                                | 42753.00                         |
| 14  | 11100                                | 31978.00                         |
| 15  | 11010                                | 31928.00                         |
| 16  | 11000                                | 21212.00                         |
| 17  | 10200                                | 32106.00                         |
| 18  | 10110                                | 31878.00                         |
| 19  | 10100                                | 21162.00                         |
| 20  | 10010                                | 21112.00                         |
| 21  | 10000                                | 10202.00                         |
| 22  | 03000                                | 33221.00                         |
| 23  | 02010                                | 33108.00                         |
| 24  | 02000                                | 22296.00                         |
| 25  | 01200                                | 33140.00                         |
| 26  | 01110                                | 32983.00                         |
| 27  | 01100                                | 22178.00                         |
| 28  | 01010                                | 22128.00                         |
| 29  | 01000                                | 11215.00                         |
| 30  | 00200                                | 22196.00                         |
| 31  | 00110                                | 22078.00                         |
| 32  | 00100                                | 11165.00                         |
| 33  | 00010                                | 11115.00                         |
| 34  | 00000                                | 11115.00                         |

different basis functions, including generic polynomials of various degrees\(^4\), Chebyshev, Hermite, Legendre and Laguerre. Our experiments have shown that each of these basis functions gives the same desired accuracy.

The GNU scientific library is used for generating the samples and solving the least squares problem via singular value decomposition.

Table 2 and 3 show the results for two cases: (a) when the price per use is $0.5$ and (b) when the price per user is $1$. The mean of daily demand is set to 30. The renewal strategy shown in

\(^4\) Using more than five terms does not lead to any difference in the results
boldface in each table is the optimal strategy as it maximizes the net income.

8. Concluding Remarks
In this paper we have analysed the complex issue of software support renewal policies. This is a problem that requires decision making at multiple stages. The problem is modelled using a mathematical procedure that aims at maximizing the expected value of net income from the revenue generated by the use of the software package minus the total cost of renewals. The mathematical model results in a Bellman equation which involves a series of sub optimization problems and can be solved backward in time. A numerical procedure based on Monte Carlo simulation can be used to obtain the optimal renewal strategy. The optimal renewal strategy is determined by maximizing the expected value of the net income over all possible renewal options.

The computational work required by the Monte Carlo method is generally quite large. One needs to exhaust all the possible paths of renewals and for each path, at each renewal time, to simulate $N$ realizations of the revenue and perform a least squares Monte Carlo simulation by solving an $N \times d$ linear system, with $d$ being the degree of selected basis functions of polynomials, preferably solved by singular-value-decomposition. In the examples shown in this study $N$ was taken at 10,000. Fortunately the nature of the Monte Carlo approach makes the numerical procedure readily parallelisable. The realizations of revenues for each sample path can be computed independently of each other, and therefore, is in fact perfectly parallelisable.

The purpose of this study is to offer a rational, mathematics-based analysis that aims to provide some insight to software renewal policies. However, the utility and the relevance of the analysis relies on how closely the simulation model can characterize the upgrade schedule and the use patterns of the software under consideration, both of which can be quite diverse and are out of the scope of this study. As well, it is implied in the solution procedure, to which the Bellman’s “principle of optimality” is applied, that the uncertainty, that is the demand for the software, is revealed at the time of each renewal decision. This is an ideal assumption, which is rarely true in the real world.

Acknowledgement
This work was inspired by Friday afternoon discussions with the late Professor Emeritus, Henning Rasmussen of The Department of Applied Mathematics at The University of Western Ontario, who passed away in the summer of 2009. The first author is indebted to Henning for his thoughts, guidance and encouragement to complete this work.

References
[1] Pfening A and Telek M 1996 Optimal Renewal Policy for Slowly Degrading Systems Working Paper
[2] Zhao T, Qi Y, Shen J, Hou D and Zheng X 2006 Journal of Communication and Computer 3 76–81
[3] Hartman J and Rogers J 2006 IMA Journal of Management Mathematics 17 143–158
[4] Mehra A and Seidmann A 2008 Optimal Timing of Upgrades over a Software Products Life Cycle Working Paper
[5] Schwartz E S and Trigeorgis L 2001 Real Option and Investment under Uncertainty (The MIT Press)
[6] Dixit A K and Pindyck R S 1994 Investment under Uncertainty (Princeton University Press)
[7] Anderson C K, Davison M and Rasmussen H 2004 Naval Research Logistics 51 686–703
[8] Wong K C 1993 The Journal of Real Estate Research 9 33–47
[9] Wong K C 1999 Property Management 18 16–24
[10] Rasmussen H 2007 Refurbishment as A Compound Real Option Problem Working Paper
[11] Bellman R 1957 Dynamic Programming (Princeton University Press)
[12] Longstaff F A and Schwartz E S 2001 The Review of Financial Studies 14 113–147