Archives of Current Research International
4(1): 1-15, 2016, Article no.ACRI.26021
ISSN: 2454-7077

SCIENCEDOMAIN ***international***
www.sciencedomain.org

Design and Full Automation of Excel Solution Templates for a Time-perspective Class of Machine Replacement Problems with Pertinent Dynamic Data

Ukwu Chukwunenye[1*]

[1]*Department of Mathematics, University of Jos, P.M.B. 2084, Jos, Plateau State, Nigeria.*

Author's contribution

The sole author designed, analyzed and interpreted and prepared the manuscript.

Article Information

DOI:10.9734/ACRI/2016/26021
Editor(s):
(1) Gongxian Xu, Associate Professor, Department of Mathematics, Bohai University, Jinzhou, China.
Reviewers:
(1) Hugo Cruz-Suárez, Benemérita Universidad Autónoma de Puebla, Mexico.
(2) Stephen Akandwanaho, University of KwaZulu-Natal, South Africa.
Complete Peer review History: http://sciencedomain.org/review-history/14423

Original Research Article

Received 29th March 2016
Accepted 18th April 2016
Published 3rd May 2016

ABSTRACT

Aim: The aim of this paper is to design and automate optimal policy prescriptions and returns for a time perspective class of machine replacement problems with pertinent dynamic data.
Methodology: The aim was achieved by the exploitation of the structure of the states in time-perspective dynamic programming recursions and the coding functionality in Microsoft Excel.
Results: The paper designed and fully automated the solution templates for the determination of the optimal time replacement policies in machine replacement problems, with pertinent data given as dynamic functions of new machine purchase year and machine age.
Conclusion: The automation of these templates obviates the need for manual inputs of the states and stage numbering as well as the inherent tedious and prohibitive manual computations associated with dynamic programming formulations and may be optimally appropriated for sensitivity analyses on such model, ensuring that problems that could take days to solve are solved in a matter of just a few minutes.

Keywords: Dynamic programming recursions; excel solution templates; full automation; machine replacement; pertinent dynamic data; time optimal policy.

**Corresponding author: Email: cukwu@hotmail.com;*

1. INTRODUCTION

Consider the problem of researching an optimal Machine Replacement policy over an *n*-period planning horizon. At the start of each year a decision is made whether to keep the machine in service an extra year or to replace it with a new one at some salvage value. As remarked by Taha [1], "the determination of the feasible values for the age of the machine at each stage is somewhat tricky". The latter went on to obtain the optimal replacement ages using network diagrammatic approach, with machine ages on the vertical axis and decision years on the horizontal axis. In an alternative time perspective approach, Winston [2] initiated the determination process for the optimal replacement time with network diagrams consisting of upper half-circles on the horizontal axis, initiating from each feasible time of the planning horizon and terminating at feasible times, with the length of successive transition times at most, the maximum operational age of the machine. Sequel to this, Winston [2] formulated dynamic recursions as functions of the decision times, the corresponding feasible transition times, the problem data and the cash-flow profile. Unfortunately network diagrams are unwieldy, cumbersome and prone to errors, especially for large problem instances; consequently the integrity of the desired optimal policies may be compromised. In what followed, Ukwu [3] deftly deployed change of variables technique, maximum machine operational age constraint and appropriate set addition definition to obtain the structure of the sets of feasible machine ages corresponding to various decision periods, in machine replacement problems, thereby obviating the need for network diagrams for such determination and followed it up by designing solution implementation templates for the corresponding dynamic programming recursions, for problems with stationary pertinent data, thereby circumventing the inherent tedious and prohibitive manual computations associated with dynamic programming formulations*.* These results were extended to problems with dynamic pertinent data, by Ukwu [4], with their associated complexity. Finally, Ukwu [5] used the state concept to obtain the structure of the sets of feasible replacement times corresponding to various decision times, in machine replacement problems and then designed solution implementation templates for the corresponding dynamic programming recursions for problems with stationary pertinent data. Other related works include Verma [6], Gupta and Hira [7], in which the average annual cost criteria was used to determine alternative optimal policies and the corresponding optimal rewards in a non-dynamic programming fashion; the use of life curves for the determination of an optimal maintenance policy was discussed by Anders and Vaccaro [8]. Furthermore, Bagui et al. [9] presented a methodology for determining the economic life of pavement-based replacement decision. In their contribution, Gress et al. [10] modeled the machine replacement problem using Markov decision process, in which the instance was optimized via linear programming. Their goal was to analyze the sensitivity and robustness of the optimal solution across the perturbation of the optimal basis. This was the only discussion on sensitivity analysis. Cruz-Suárez & Ilhuicatzi-Roldan [11] considered a random-horizon problem, involving a system consisting of *n* independently operating deteriorating machines, with an associated cost function. The system was assumed to be observed at discrete times and the objective function was the total expected cost. The work provided an optimal replacement policy that minimized the operating cost of the system, as well as the solution implementation of a problem instance on MATLAB platform. Ukwu [12] examined the effects of different planning horizons, with machine replacement age fixed, in the Excel automated solutions to a class machine replacement problems with stationary pertinent data. The investigation revealed that if the replacement age is fixed ,and n_1 and n_2 are any two horizon lengths with $n_1 < n_2$, and p^*_j, $g(j)$ are stage j optimal decision and reward from the template with horizon length n_2, for $j \in \{n_2+1-n_1, \cdots, n_2\}$, then $p^*_j + n_1 - n_2$ and $g\left(j+n_1-n_2\right)$ are the corresponding optimal decision and reward in stage $j+n_1-n_2$ for the template with the horizon length n_1. Moreover the corresponding optimal rewards are equal. The findings in Ukwu [12] are of great research and practical interests, as they clearly unveil tremendous savings in cost, time and energy in the search for and evolution of optimal replacement strategies with respect to problems with the same or unspecified replacement ages but different horizon lengths. They are especially appealing due to the invariance of the remaining pertinent data and hence the preservation of the columns of the solution templates. This article sets out to nontrivially extend the results in Ukwu [5] to a class of machine replacement problems with dynamic pertinent data. By appropriating the structure of the states in time-perspective dynamic programming recursions the article makes positive and significant contribution to knowledge by designing and fully automating the solution templates for the determination of the optimal time replacement policies in machine replacement problems, with pertinent data given as dynamic functions of new machine purchase year and machine age. The article goes further to give an exposition on the automation process and computational details.

2. METHODOLOGY

In this section, the problem data, working definitions, elements of the DP model and the dynamic programming (DP) recursions are laid out as follows:

2.1 Pertinent Dynamic Data

Equipment Replacement age $= m$

Problem horizon length $= n$

$m_{\lambda j} =$ Cost of maintaining an , in its λ^{th} year of operation, given that the equipment was purchased in year $j;\ \lambda \in \{1,\cdots,m\},\ j \in \{1,2,\cdots,n\}$

$r_{\lambda j} =$ Revenue from an equipment in its λ^{th} year of operation, given that the equipment was purchased in year $j;\ \lambda \in \{1,\cdots,m\},\ j \in \{1,2,\cdots,n\}$

$s_{\lambda j} =$ Salvage value of a λ year-old equipment,given that the equipment was purchased in year j; $\lambda \in \{1,\cdots,m\},\ j \in \{1,2,\cdots,n\}$

$I_j =$ Cost of acquiring a new equipment in year j.

The elements of the DP are the following:

1. Stage i, represented by time $i, i \in \{1,2,\ldots,n-1\}$
2. The alternatives at stage (time) i. These call for keeping or replacing the machine at one of the times $i+1,\cdots,\min\{m+i,n\}$.
3. The state at stage (time) i, represented by the time to advance to from time i.

Let S_i be the set of feasible equipment transition times (states) from decision time $i,\ i \in \{0,2,\ldots,n-1\}$.

Let $g(i)$ be the minimum net cost incurred in operating machine during the periods spanning times $i, i+1,\ldots,n-1, n$, including purchase cost, revenue and salvage value for the newly purchased machine given that the machine has been purchased at time i.

Let c_{ip} be the net cost (including purchasing, salvage value and earned revenue) of purchasing machine i and operating it until time p.

Note: The definition of $g(i)$ starting from time i to time n implies that backward recursion will be used. Forward recursion would start from time 1 to time i.

The ensuing theorem is relevant to the main result of this article.

2.2 Theorem on Structure of States, Optimal Policy Prescriptions and Rewards, Ukwu [5]

For $i \in \{0,1,\cdots,n-1\}$,

(a) $S_i = \{\min\{i+1,n\},\cdots,\min\{m+i,n\}\}$

with terminal state specification $S_n = \{n\}$

(b) $p \in S_i \Rightarrow c_{ip} = I + \sum_{j=1}^{p-i} m_j - s_{p-i} = k_{p-i}$

(c) $g(i) = \min_{p \in S_i} \{c_{ip} + g(p)\},\ g(n) = 0$

Cf. Ukwu [5] for the proof.

(a) translates to

$$S_i = \begin{cases} \{i+1,\cdots,\ m+i\},\ \text{if}\ \ i \in \{0,\cdots,n-m\} \\ \{i+1,\cdots,\ n\},\ \text{if}\ \ i \in \{n-m,\cdots,n-1\} \end{cases}$$

The assumption $n \geq m$ is implicit, since machine cannot be operated beyond the planning horizon.

The terminal state specification makes sense since the process terminates at time n**.** Thus n can be regarded as an absorbing state. Cf. Ukwu [5] for the proof of the theorem.

The next section provides an instance of the class of machine replacement problems for which Excel solution templates will be designed and automated.

2.3 Archetypal Application Problem for Solution Template Design and Automation

A company reviews the status of its heavy machine at the end of each year, and a decision is made to either to keep the machine an extra year or to replace it. However, machine that has been in service for 4 years must be replaced. The company wishes to develop a replacement policy for its fleet over the next ten years. The following table provides the pertinent data. All monetary values are in hundreds of dollars. The machine is new at the start of year 1.

Table 1. Pertinent data for optimal policy and reward determination for given purchase prices, costs, ages, revenues and salvage values

Purchase		Maintenance cost ($) for given age (yr.)				Revenue ($) for given age (yr.)				Salvage value ($) for given age (yr.)			
Yr.	**Price**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**	**1**	**2**	**3**	**4**
1	100	2.0	5.0	6.0	6.5	20	19	18	17	90	70	50	45
2	120	2.5	5.5	6.2	6.8	21	20	19	18	110	95	80	75
3	130	2.8	5.6	6.5	7.0	22	21	20	19	120	110	100	95
4	135	3.2	6.0	6.7	7.3	21	20	19	18	120	115	110	115
5	138	3.5	6.3	7.0	7.5	21	20	19	17	120	118	112	115
6	142	3.9	6.4	7.2	7.7	22	21	20	19	125	120	112	117
7	148	4.1	6.6	7.3	8.0	22	21	20	19	135	129	119	120
8	152	4.3	6.7	7.5	8.2	22	21	20	19	140	132	120	121
9	155	4.5	7.0	7.8	8.5	22	21	20	18	150	145	138	140
10	160	5.0	7.1	8.0	8.8	23	22	21	20	158	150	145	147

Design Automated Solution Templates and consequently Alternate Optimal Replacement Policies and rewards for the Machine Fleet.

2.3.1 Solution to problem 2.3

The given data must be restructured in three – dimension format, to depend only on the age of the machine for a given machine purchase year, starting from the terminal year, as follows:

Table 2. Pertinent data for decision year 10, showing equipment ages, revenues maintenance costs and salvage values

Year 10: Purchase price = $16,000			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2300	500	-
1	2200	710	15800
2	2100	800	15000
3	2000	880	14500
4			14700

Table 3. Pertinent data for decision year 9 showing equipment ages, revenues maintenance costs and salvage values

Year 9: Purchase price = $15,500			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2200	450	-
1	2100	700	15000
2	2000	780	14500
3	1800	850	13800
4			14000

Table 4. Pertinent data for decision year 8 showing equipment ages, revenues maintenance costs and salvage values

Year 8: Purchase price = $15,200			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2200	430	-
1	2100	670	14000
2	2000	750	13200
3	1900	820	12000
4			12100

Table 5. Pertinent data for decision year 7 showing equipment ages, revenues maintenance costs and salvage values

Year 7: Purchase price = $14,800			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2200	410	-
1	2100	660	13500
2	2000	730	12900
3	1900	800	11900
4			12000

Table 6. Pertinent data for decision year 6 showing equipment ages, revenues maintenance costs and salvage values

Year 6: Purchase price = $14,200			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2200	390	-
1	2100	640	12500
2	2000	720	12000
3	1900	770	11200
4			11700

Table 7. Pertinent data for decision year 5 showing equipment ages, revenues maintenance costs and salvage values

Year 5: Purchase price = $13,800			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2100	350	-
1	2000	630	12000
2	1900	700	11800
3	1700	750	11200
4			11500

Table 8. Pertinent data for decision year 4 showing equipment ages, revenues maintenance costs and salvage values

Year 4: Purchase price = $13,500			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2100	320	-
1	2000	600	12000
2	1900	670	11500
3	1800	730	11000
4			11500

Table 9. Pertinent data for decision year 3 showing equipment ages, revenues maintenance costs and salvage values

Year 3: Purchase price = $13,000			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2200	280	-
1	2100	560	12000
2	2000	650	11000
3	1900	700	10000
4			9500

Table 10. Pertinent data for decision year 2 showing equipment ages, revenues maintenance costs and salvage values

Year 2: Purchase price = $12,000			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2100	250	-
1	2000	550	11000
2	1990	620	9500
3	1800	680	8000
4			7500

Table 11. Pertinent data for decision year 1 showing equipment ages, revenues maintenance costs and salvage values

Year 1: Purchase price = $10,000			
Age: *t* yrs.	**Revenue: *r*(*t*) ($)**	**Maintenance cost: *c*(*t*) ($)**	**Salvage value: *s*(*t*) ($)**
0	2000	200	-
1	1900	500	9000
2	1800	600	7000
3	1700	650	5000
4			4500

Equipment Replacement Model	***I***	**Stage**		***m***	***n***
Automated Solution Template	16,000	10		4	10
t = p-i		1	2	3	4
$m_{\lambda j}$		500	710	800	880
$s_{\lambda j}$		15800	15000	14500	14700
$-r_{\lambda j}$		-2300	-2200	-2100	-2000
k_{ti}		-1600	-2290	-3090	-4410
	I	**Stage**			
	15,500	9			
$m_{\lambda j}$		450	700	780	850
$s_{\lambda j}$		15000	14500	13800	14000
$-r_{\lambda j}$		-2200	-2100	-2000	-1800
k_{ti}		-1250	-2150	-2670	-3820
	I	**Stage**			
	15,200	8			
$m_{\lambda j}$		430	670	750	820
$s_{\lambda j}$		14000	13200	12000	12100
$-r_{\lambda j}$		-2200	-2100	-2000	-1900
k_{ti}		-570	-1200	-1250	-2430
	I	**Stage**			
	14,800	7			
$m_{\lambda j}$		410	660	730	800
$s_{\lambda j}$		13500	12900	11900	12000
$-r_{\lambda j}$		-2200	-2100	-2000	-1900
k_{ti}		-490	-1330	-1600	-2800
	I	**Stage**			
	14,200	6			
$m_{\lambda j}$		390	640	720	770
$s_{\lambda j}$		12500	12000	11200	11700
$-r_{\lambda j}$		-2200	-2100	-2000	-1900
k_{ti}		-110	-1070	-1550	-3180

Fig. 1. Preliminary computations for stages 10 to 6

Equipment Replacement Model	***I***	**Stage**		***m***	***n***
Automated Solution Template	13,800	5		3	10
$m_{\lambda j}$		350	630	700	750
$s_{\lambda j}$		12000	11800	11200	11500
$-r_{\lambda j}$		-2100	-2000	-1900	-1700
k_{ti}		50	-1120	-1720	-2970
	I	**Stage**			
	13,500	4			
$m_{\lambda j}$		320	600	670	730
$s_{\lambda j}$		12000	11500	11000	11500
$-r_{\lambda j}$		-2100	-2000	-1900	-1800
k_{ti}		-280	-1180	-1910	-3480
	I	**Stage**			
	13,000	3			
$m_{\lambda j}$		280	560	650	700
$s_{\lambda j}$		12000	11000	10000	9500
$-r_{\lambda j}$		-2200	-2100	-2000	-1900
k_{ti}		-920	-1460	-1810	-2510
	I	**Stage**			
	12,000	2			
$m_{\lambda j}$		250	550	620	680
$s_{\lambda j}$		11000	9500	8000	7500
$-r_{\lambda j}$		-2100	-2000	-1900	-1800
k_{ti}		-850	-800	-580	-1200
	I	**Stage**			
	10,000	1			
$m_{\lambda j}$		200	500	600	650
$s_{\lambda j}$		9000	7000	5000	4500
$-r_{\lambda j}$		-2000	-1900	-1800	-1700
k_{ti}		-800	-200	600	50

Fig. 2. Preliminary computations for stages 5 to 1

Stage *j* Computations: *i = j - 1*					
Stage	**10**				
i	9				
q	9	10			
k_{ti}		-1600			
g(q)	-1600	0			
f(i, p)		-1600			
g(i)	**-1,600**				
p*		10			
Stage	**9**				
i	**8**				
q	**8**	9	10		
k_{ti}		-1250	-2150		
g(q)	(2,850)	-1600	0		
f(i, p)		-2850	-2150		
g(i)	**-2,850**				
p*		9			
Stage	**8**				
i	**7**				
q	**7**	8	9	10	
k_{ti}		-570	-1200	-1250	
g(q)	-3420	-2850	-1600	0	
f(i, p)		-3420	-2800	-1250	
g(i)	**-3,420**				
p*		8			
Stage	**7**				
i	**6**				
q	**6**	7	8	9	10
k_{ti}		-490	-1330	-1600	9200
g(q)	-4180	-3420	-2850	-1600	0
f(i, p)		-3910	-4180	-3200	9200
g(i)	**-4,180**				
p*			8		
Stage	**6**				
i	**5**				
q	**5**	6	7	8	9
k_{ti}		-110	-1070	-1550	-3180
g(q)	-4780	-4180	-3420	-2850	-1600
f(i, p)		-4290	-4490	-4400	-4780
g(i)	**-4,780**				
p*					9

Fig. 3. Main Computations for the optimal prescription policy and rewards for stages 10 to 6

Stage	**5**				
i	**4**				
q	**4**	5	6	7	8
k_{ti}		50	-1120	-1720	-2970
g(q)	-5820	-4780	-4180	-3420	-2850
f(i, p)		-4730	-5300	-5140	-5820
g(i)	**-5,820**				
p*					8
Stage	**4**				
i	**3**				
q	**3**	4	5	6	7
k_{ti}		-280	-1180	-1910	-3480
g(q)	-6900	-5820	-4780	-4180	-3420
f(i, p)		-6100	-5960	-6090	-6900
g(i)	**-6,900**				
p*					7
Stage	**3**				
i	**2**				
q	**2**	3	4	5	6
k_{ti}		-920	-1460	-1810	-2510
g(q)	-7820	-6900	-5820	-4780	-4180
f(i, p)		-7820	-7280	-6590	-6690
g(i)	**-7,820**				
p*		3			
Stage	**2**				
i	**1**				
q	**1**	2	3	4	5
k_{ti}		-850	-800	-580	-1200
g(q)	-8670	-7820	-6900	-5820	-4780
f(i, p)		-8670	-7700	-6400	-5980
g(i)	**-8,670**				
p*		2			
Stage	**1**				
i	**0**				
q	**0**	1	2	3	4
k_{ti}		-800	-200	600	50
g(q)	-9470	-8670	-7820	-6900	-5820
f(i, p)		-9470	-8020	-6300	-5770
g(i)	**-9,470**				
p*		1			

Fig. 4. Main Computations for the optimal prescription policy and rewards for stages 5 to 1

To determine S_i for $i \in \{1, 2, \cdots, 10\}$, invoke the following result from Ukwu [5]:

2.4 Corollary to (a) of Theorem 2.2, Ukwu [5]

If $t_1 < m$, then for $i \in \{2, \ldots, n\}$,

$$S_i = \begin{cases} \left\{ \min\limits_{2 \le j \le i} \{ j-1, m \} \right\} \cup \{ i - 1 + t_1 \}, \text{ if } i \le m + 1 - t_1 \\ \left\{ \min\limits_{2 \le j \le i} \{ j-1, m \} \right\}, \text{ if } i > m + 1 - t_1 \end{cases}$$

Cf. Ukwu [5] for the proof.

In the given problem,

$$t_1 = 0,\ m = 4,\ n = 10,\ i \in \{1, 2, \cdots, 10\}, S_1 = \{0\} \Rightarrow m + 1 - t_1 = 4$$
$$\Rightarrow S_2 = \{1\}, S_3 = \{1, 2\}, S_i = \{1, 2, 3\}, i \in \{4, 5, \cdots, 10\}.$$

Since the revenue profile is furnished the template for the implementation of $g(i)$ in theorem 1 may be exploited to obtain the optimal solution to the problem noting that the optimal objective value of the given problem is $-g(i)$: the maximum net profit derived from operating each machine in the fleet.

The interface and Excel solution templates and implementations of the optimal policy prescriptions and rewards are given below. These are followed by interpretations of the optimal reward and policy prescription and a general exposition on prototypical solution templates for the given class of problems. The optimal strategies and returns and overall outputs for the given problem are seen to be consistent with the general exposition.

3. RESULTS

3.1 Optimal Reward for Problem 2.3

The maximum net profit is given by $-g(0)$ = \$9,470.00

The optimal time replacement policy is given schematically by:

$$0 \to 1 \to 2 \to 3 \to 7 \to 8 \to 9 \to 10$$

3.2 Interpretation of the Optimal Time Replacement Schema

Starting with a new machine at time 0, machine replacements should be effected at times 1, 2, and 3; then the replacement machine at time 3 should be kept until time 7 when it is 4 years old and mandatorily replaced with a new machine. Subsequently, machine replacements should be effected at times 8 and 9. The replacement machine at time 9 should be deployed for one year until time 10 when the process terminates

4. DISCUSSION

4.1 An Exposition on the Solution Templates

The exposition on the solution template encompasses notations and features of the design template, the preliminary computations for k_{ti}, the core row-spacing imperatives among consecutive stages and main computations for optimal policies and corresponding rewards.

4.1.1 Notations and features of the design template

1. Identifiers are written in bold typeface while numeric values are not bolded; num_val is the numeric value assigned to **num** and stored at the cell location.
2. Formulas are preceded by ' = '
3. Following the execution of a formula by the keyboard operation **'< Enter > '**, the act of clicking back on a specified cell, positioning the cursor at the right edge of the cell until a crosshair appears and dragging the cross-hair horizontally or

vertically to some cell will be referred to as clerical routine/duty.

4. Cross-hair horizontal dragging routines terminate in column N, except for the rare case $m \geq 13$, as in the Aviation Industry, Power sector or in developing economies, where machine may be in use for 13 years or more due to poor economic conditions.

4.1.2 Preliminary computations for k_{ti} and stage separation imperatives

1. In the preliminary computations for k_{ti}, three rows separate the stages. There is an additional row in stage n of the process – the row for the operational periods identifier $\boldsymbol{t = p - i}$.
2. There are 6 rows in each of stages $n-1$ to 1, starting from the stage identifier and terminating in the optimal solution identifier p^*. In stage n, the relevant nonblank rows are 1 through 7.
3. Stage j data and k_{ti} are located in rows: $2+9(n-j)$ to $7+9(n-j)$, for $j \in \{n-1, n-2, \cdots, 1\}$, where $i = j-1$.

4.1.3 Main computations for optimal policies, rewards and stage separation imperatives

1. In the main computations for optimal policies, one row separates the stages. There is an additional row in stage n of the process – the row for the identifier. Altogether there are nine rows here, namely row $9n$ for the identifier and rows $9n+1$ to $9n+8$ for the computations
2. **Stage j Computations:** $\boldsymbol{i = j} - 1$.
3. There are 8 rows in each of stages $n-1$ to 1, starting from the stage identifier and terminating in the optimal solution identifier p^*.
4. Stage j main computations are located in rows: $18n-9j+1$ to $18n-9j+8$, for $j \in \{n-1, n-2, \cdots, 1\}$, starting from the stage number implementation to the optimal solution p^*.

Step 1: Storage of Parameters, Automation of Stage Numbers and Operational Periods

(a) Type the values of I_n, m and n in the cell locations B2, E2 and F2 respectively, where I_n is the purchase price of the machine in year (stage) n. Automate the stage number n by typing

' = \$B\$2', in C2, followed by **<Enter>.** Type the value of I_{n-1} in B12 and those of I_j in cells $\mathrm{B}\big(3+9(n-j)\big)$, for $j \in \{n-2, n-3, \cdots, 1\}$.

Type ' = \$F\$2 – 1 ' in C12, **<Enter>;** type ' = \$C12 – 1 ' in C21, **<Enter>.** Then copy the formula in C21 and paste it successively into the cell locations:

$\mathrm{C}\big(3+9(n-j)\big)$, for $j \in \{n-3, n-4, \cdots, 1\}$, to automate the stage numbering.

(b) Type in 1 in C3. Type ' **=IF(C3="","",IF(C3>=MIN(\$E\$2,\$F\$2),"",1+C3))** ' in D3, **<Enter>** Then perform the horizontal clerical routine to secure $t = p - i \in \{1, 2, \cdots, m\}$.

Step 2: Computations of the k_{ti}s in stage n

Type the following code segment in C7:

= IF(C$3="","",$B2+SUM($C4:C4,($C6:C6))-C5) <Enter>

Then perform the horizontal clerical routine to secure the k_{ti}s in stage n.

Step 3: Computations of the k_{ti}s in stages $n-1$ to 1

Type the following code segment in C16:

=IF(C$3="","",$B12+SUM($C13:C13,($C15:C15))-C14) <Enter>

Then perform the horizontal clerical routine to secure the k_{ti}s in stage $n-1$.

Finally, copy the contiguous region of the above formula and paste it successively on the corresponding contiguous regions in Excel rows $7+9(n-j), j\in\{n-2,n-3,\cdots,1\}$, starting from column C, to secure the k_{ti}s in stage $n-2$ to 1.

4.1.4 Main computations for optimal policies and rewards

Step 1: Automation of Stage Numbers and Operational Periods

(a) Type ' =$F$2', in cell B91, **<Enter>** ; Type ' =$B91 – 1, in cell B100, **<Enter>**
(b) Copy the formula in B100 and paste it successively in the cell locations

$\mathrm{B}\left(100+9(n-1-j)\right),$ for $j\in\{n-2,n-3,\cdots,1\}$, to secure the stage numbers.

Step 2: Automation of values of i and the first value of q in stages $n, n-1$ and $n-2$

(a) Type ' = $B91-1' in B92 . **<Enter>,** to obtain i in stage n
(b) Type ' = $B92' in B93. **<Enter>,** to obtain the first q in stage n
(c) Type ' = $B92 – 1 ' in B101. **<Enter>,** to obtain value of i in stage $n-1$
(d) Type ' = $B93 – 1 ' in B102. **<Enter>,** to obtain the first q in stage $n-1$
(e) Copy the contiguous cells B101:B102 of stage $n-1$ into the contiguous region B110:B111 of stage $n-2$, to secure the value of i and the first value of q, in stage $n-2$

Step 3: Automation of the values of $q, k_{ti}, g(q), f(i,p), g(i)$ and p^* in stages $n, n-1$ and $n-2$

(a) Type ' **=IF(B93>=MIN($B92+$E$2,$F$2),"",1+B93)**' in C93. **<Enter>.**

Then perform the horizontal clerical routine to secure the remaining values of q in stage n.

Copy the above formula for q, in stage n, starting from column C to the corresponding contiguous regions in stages $n-1$ and $n-2$, in rows 102 and 111 respectively, starting from column C, to secure the values of q in those stages.

(b) Type **'=IF (C93="","",** C7)' in C94. **<Enter>.**

Then perform the horizontal clerical routine to secure the values of k_{ti} in stage n.

Copy the above formula for k_{ti}, in stage n, starting from column C to the corresponding contiguous regions in stages $n-1$ and $n-2$, in rows 103 and 112 respectively, starting from column C, to secure the values of k_{ti} in those stages.

(c) Type in 0 in C95, in stage n

Type '=IF(C102="","",IF(C102 = $F$2,0, B95))' in C104. **<Enter>.**

Then perform the horizontal clerical routine to secure the values of $g(q)$ in stage $n-1$.

Copy the above formula for $g(q)$, in stage $n-1$, starting from column C to the corresponding contiguous region in stage $n-2$, in row 113, starting from column C, to secure the values of $g(q)$ in stage $n-2$.

(d) Type '= IF(C93 = "", "",C94+C95)' in C96, in stage n. **<Enter>.**

Then perform the horizontal clerical routine to secure the values of $f(i,p)$ in stage n.

Copy the above formula for $f(i,p)$, in stage n, starting from column C to the corresponding contiguous regions in stages $n-1$ and $n-2$, in rows 105 and 114 respectively, starting from column C, to secure the values of $f(i,p)$ in those stages.

(e) Type '=MIN ($C96:$N96)' in B95. **<Enter>.** Then copy this formula for $g(q), \text{ for } q=i=n-1$ to B104 and B113, in stages $n-1$ and $n-2$ respectively.

(f) Type '=$B95' in B97. **<Enter>.** Then **c**opy this formula to B106 and B115, in stages $n-1$ and $n-2$ respectively.

(g) Type ' = IF(C96=$B97,C93,"")' in C98. **<Enter>.** Then perform the horizontal clerical routine to secure the value(s) of p^* in stage n. Then copy this formula to the corresponding regions in stages $n-1$ and $n-2$, starting from C107 and C116, in respectively.

Step 4: Automation of the computations in stages $n-3$ to 1

Copy the contiguous region of stage $n-2$, in rows 109 to 116, starting from column A onto the contiguous regions in stages $n-3$ to 1, starting from row 118, with a blank row between consecutive stages. This single Copy and $n-3$ Paste operations secure the desired values in stages $n-3$ to 1, and hence the optimal policy prescriptions and corresponding rewards.

5. CONCLUSION

This paper exploited the structure of the transition time states in dynamic programming recursions, Microsoft Excel functionality and coding possibilities to design and fully automate the solution templates for the determination of machine time-optimal replacement policies in a certain class of machine replacement problems, with pertinent data given as dynamic functions of new machine purchase year and machine age. The automation of these templates was stringent on the imperative of prefixed spacing of three rows between consecutive stages in the preliminary computations and one row between consecutive stages in the main computations. The templates with dynamic data were obtained through extensive modifications of the base templates in Ukwu [5] and demonstrated consistency with the latter, as verifiable by the use of the same data set in all the stages of the process. The automation of these templates is a novelty that has introduced/created implementation paradigm shift, paving the way for speedy and optimal solutions to large-scale problems in the given class with almost effortless resolutions of associated issues of sensitivity analyses.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Taha HA. Operations research: An introduction. Seventh Edition. Prentice-Hall of India, New Delhi; 2006.
2. Winston LW. Operations research: Applications and algorithms, Duxbury Press, Boston; 1994.
3. Ukwu C. (2015d). Novel state results on machine replacement problems and excel solution implementation templates. Transactions of the Nigerian Association of Mathematical Physics. 2015;1:237-254.
4. Ukwu C. (2015e). Excel solution implementation templates for machine replacement problems with age and decision period variable data. Transactions of the Nigerian Association of Mathematical Physics. 2015;1:255-264.
5. Ukwu C. Novel formulations and prototypical solution template for time-based recursions of machine replacement problems with stationary pertinent data. Journal of Basic and Applied Research International. 2016;16(2):138-154.
6. Verma AP. Operations Research. S.K. Kataria & Sons. New Delhi; 2010.
7. Gupta PK, Hira DS. Problems in Operations Research (Principles and Solutions). S. Chand & Company Ltd. New Delhi; 2012.
8. Anders G, Vaccaro A. (eds.). Innovations in power systems reliability, springer series in reliability engineering. Springer-Verlag London Limited. 2011;13-58.
DOI: 10.1007/978-0-85729-088-5_2
9. Bagui SK, Chakraborti AK, Bhadra SM. Application of replacement theory in determination of pavement design life. Jordan Journal of Civil Engineering. 2012;6(1):147-160.
10. Gress ESH, Lechuga GP, Gress NH, Sensitivity analysis of the replacement problem. Intelligent Control and Automation. 2014;5:46-59.
Available:http://dx.doi.org/10.4236/ica.2014.52006
11. Cruz-Suárez H, Ilhuicatzi-Roldan R. Optimal replacement in a system of n-Machines with random horizon. Proyecciones Journal of Mathematics. 2012;31(1):219-232.
12. Ukwu C. Sensitivity analysis of time horizon for a class machine replacement problems with stationary pertinent data. Journal of Basic and Applied Research International. 2016;16(2):80-87.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/14423