PERFORMANCE MODELLING OF AN APPLICATION SERVER USING MATRIX GEOMETRIC METHOD

Vidushi Sharma[*]

Kriti Priya Gupta[**]

Abstract

The paper investigates performance parameters of an application server handling different types of jobs prioritized and non-prioritized. Busy state handles the prioritized jobs which are processed using N-policy. When there are no prioritized jobs the server goes in working vacation state where it handles non prioritized jobs. These jobs can also encounter server breakdown. A two phase repair system is considered for the server when it breaks down in the busy state. The system is modeled as an M/M/1 queue and the matrix geometric method is used for computing the steady state probabilities of various states of the server. Performance measures like throughput, delay, expected number of customers in the different states are computed and illustrated numerically.

Keywords: Application server, Matrix geometric method, Breakdown, Repair, N-policy, Working Vacation.

[*] Associate Professor, School of ICT, Gautam Buddha University

[**] Associate Professor, SCMS-NOIDA, Symbiosis International University

1. Introduction

With the growing demand for information systems and their deployment for mass utilization through internet, intranets and extranets, the requirement of application server has increased. Application servers are dedicated servers to handle various requests for applications. These application servers can be deployed in photonic, wired and wireless network. Researchers like Doshi [1986, 1991] and Takagi [1991] studied the queuing systems in which the server stops during vacation. Server goes on a vacation in case there are no requests in the queue. Servi and Finn [2002] analyzed a queuing system in which the server works at a different rate rather than completely stopping the service during the vacation. Baba [2005] considered a GI/M/1 queue with vacations such that the server works with different rates. They derived the steady-state distributions for the number of customers in the system both at arrival and arbitrary epochs. Wu and Takagi [2006] studied M/G/1 queue with multiple vacations and exhaustive service discipline. The service times in a vacation and in a service period were distributed random variables.

Sometimes the information system server starts providing service to the requests only after a certain number of jobs say N gets accumulate in the queue. Researchers have analyzed this N-policy in different frameworks. Wang et al. [1999] analyzed N-policy for $M/H_2/1$ queueing system with single removable and non-reliable server. They evaluated the minimum expected cost and the optimal operating policy based on numerical values given to the system parameters as well as to the cost elements. Lee et al. [1999] considered non-preemptive priority policies for multiclass queueing systems with N-policy. They obtained Laplace transform for waiting times of each class of customers. Jain and Poonam [2002] gave an N-policy for the state-dependent $M/E_k/1$ queue with server breakdown. Choudhury and Madan [2005] studied two-stage batch arrival queueing system with a modified bernoulli schedule vacation under *N*-policy.

Matrix geometric technique has gained importance and is being utilized to solve various problems. Baily and Neuts [1981] used matrix geometric method to obtain queue length densities at times prior to arrival in a multi server queue model. Gillent and Latouche [1983] gave semi-explicit solutions for M/PH/1-like queuing systems using matrix geometric method. Mirchandany et al. [1990] studied the performance characteristics of simple load sharing algorithms for heterogeneous distributed systems. They used matrix geometric solution technique to solve the model. Chakravarthy and Dudin [2003] analyzed retrial queueing model with markovian arrival process with two types of customers. Pla and Giner [2005] used the matrix geometric technique in the analysis of priority channel assignment scheme in cellular communication system. Ke and Wang [2007] studied the machine repair problem consisting of number of operating machines with two types of spare machines. They considered the vacation policies and used matrix geometric theory to find the steady-state probabilities of the number of failed machines in the system as well as the performance measures.

In this paper, we have analyzed the performance of an application server that can handle two types of requests i) prioritized and ii) non prioritized. Prioritized requests are given preference by serving them at a higher rate as compared to non-prioritized requests. Server is said to be in busy state while serving the prioritized requests. There is also a provision of repair in case of server breakdown in the busy state. If the repair is successful then the server again resumes the busy state and if the repair fails, the server goes into a second repair phase. After the second repair, the busy state is resumed again.

We have used matrix geometric method to compute the steady state probabilities of the application server in various states, i.e., busy and working vacation, breakdown, repair. Performance parameters like delay in the system, throughput and the average number of jobs in the system are evaluated using these steady state probabilities. The paper is organized into various sections. Section 2 describes the model. Section 3 gives the matrix geometric approach to compute the various steady state probabilities of the system. Numerical illustrations are presented in section 3. Finally, conclusion is drawn in section 4.

2. The Model

An application server with working vacation is considered which serves two types of jobs: prioritized jobs and non-prioritized jobs. When there are no prioritized jobs in the system, the server takes vacation form the prioritized jobs and serves the non-prioritized jobs. There is a provision of N-policy in the system according to which, the server comes back from working vacation and starts serving the prioritized jobs, when N number of prioritized jobs gets accumulated in the system. Both types of jobs arrive to the server in a Poisson fashion and are served with exponentially distributed rates. The server is subjected to breakdown which may occur in busy state when it is serving the prioritized jobs and also in working vacation state when it is serving the non-prioritized jobs. A two-phase repair system is considered for the server when it breaks down in the busy state. The breakdown and repair rates are assumed to be exponentially distributed. The system is modeled as an M/M/1 queueing system (fig. 1) with six different states as follows:

- **Working vacation state**: In this state, the server takes vacation from the prioritized jobs and serves the non-prioritized jobs with rate μ_0. The duration of the working vacation is independent and identically distributed random variable having exponential distribution with mean $1/\eta$.
- **Busy state:** In this state, the server renders services to the prioritized jobs with rate $\mu_{2.}$
- **First breakdown state**: This state corresponds to the server breakdown in the working vacation. The server breaks down with rate α_v and is immediately sent for repair which is done with rate β_v.
- **Second breakdown state**: In this state, the server breaks down with a rate α_b and is sent for repair to the first repairman who takes an exponentially distributed set up time θ.
- **First repair state**: In this state, the first repairman tries to repair the server and sends it back to the busy state with a rate $(1-r_1)\beta_{1,}$ if the server is repaired otherwise it sends it the second repairman with a rate $r_1\beta_1$.
- **Second repair state**: In this state, the second repairman repairs the server and sends it back to the busy state with a rate β_2

A Markov process is constructed for the M/M/1 queueing system with the following state space:

$$S = \{(i,j)|\ i=0,1,2,\ldots.;j=0,1,\ldots5\}$$

where i is the number of jobs in the system and

$$j = \begin{cases} 0, \text{if the server is in working vacation state} \\ 1, \text{if the server is in the first breakdown state} \\ 2, \text{if the server is in the busy state} \\ 3, \text{if the server is in the second breakdown state} \\ 4, \text{if the server is in the first repair state} \\ 5, \text{if the server is in the second repair state} \end{cases}$$

Let $P_{i,j}$ be the steady state probability that the there are i number of jobs in the j^{th} state. The arrival rates of the jobs in various states are given by λ_j (j=0,1,…,5).

3. Matrix geometric solution

The matrix-geometric approach developed by Neuts (1981) is employed for obtaining the queue size distribution for the model discussed in the previous section. The generator of the Markov process can be represented in the following partitioned structure

$$Q = \begin{bmatrix} B_{00} & B_{01} & 0 & 0 & 0 & \dots & \dots & \dots \\ B_{10} & A_1 & A_0 & 0 & 0 & \dots & \dots & \dots \\ 0 & A_2 & A_1 & A_0 & 0 & \dots & \dots & \dots \\ 0 & 0 & A_2 & A_1 & A_0 & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \qquad \dots(1)$$

where B_{00}, A_0, A_1 and A_2 are square matrices whereas B_{01} and B_{10}, are rectangular matrices of following orders:

$$o(B_{00}) = 1 + 6(N-1)$$

$$o(B_{01}) = (1 + 6(N-1)) \times 6$$

$$o(B_{10}) = 6 \times (1 + 6(N-1))$$

$$o(A_0) = o(A_1) = o(A_2) = 6 \times 6$$

The solution of the above system can be obtained by solving the following matrix equation

$$A_0 + RA_1 + R^2A_2 = 0 \qquad \dots(2)$$

Here, the matrix R can be computed using the following iterative procedure:

$$R(n+1) = -A_0A_1^{-1} - R_2(n)A_2A_1^{-1}\text{ , n = 01,2,...} \qquad \dots(3)$$

Let **X** be the vector of steady state probabilities P(i,j) associated with Q, such that

$$\mathbf{X}Q = 0 \qquad \dots(4)$$

$$\text{and } \mathbf{X}e = 1 \qquad \dots(5)$$

which is the normalizing condition where e is a column vector of appropriate dimension with all the elements equal to 1.

Let us partition **X** as $\mathbf{X} = [\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2 \dots]$ with the following relationships

$$X_0B_{00} + X_1B_{10} = 0 \qquad \dots(6)$$

$$X_0B_{01} + X_1A_1 + X_2A_2 = 0 \qquad \dots(7)$$

Substituting $X_2 = X_1R$ in (7), the above equations (6) and (7) can be written in the matrix form as

$$\lbrack X_0 \quad X_1 \rbrack \begin{bmatrix} B_{00} & B_{01} \\ B_{10} & A_1 + RA_2 \end{bmatrix} \qquad \ldots(8)$$

This does not give any unique solution. By using the normalizing condition, we obtain

$$X_0e + X_1\sum_{j=1}^{\infty} R^{j-1}e = X_0e + X_1(I-R)^{-1}e = 1 \qquad \ldots(9)$$

where e is a column matrix of suitable dimension having all elements 1. This gives a unique solution for [$\mathbf{X}_0$, $\mathbf{X}_1$]. The rest of the probabilities X_i (i = 2,3,…) can be obtained by using the following relation

$$X_i = X_1R^{i-1} \;, \; i = 2,3,\ldots \qquad \ldots(10)$$

For particular case when N=4, the matrices B_{00}, B_{01}, B_{10}, A_0, A_1 and A_2 can be given as follows:

$\mathbf{B_{00}}$=

$-\lambda_0$	μ_0		μ_2															
λ_0	$-\lambda_0'$	β_v					μ_0											
	α_v	$-\lambda_1'$																
			$-\lambda_3'$		$(1-r_1)\beta_1$	β_2			μ_2									
			α_b	$-\lambda_3'$														
				θ	$-\lambda r_1'$													
					$r_1\beta_1$	$-\lambda_5'$												
	λ_0						$-\lambda_0'$	β_v					μ_0					
		λ_1					α_v	$-\lambda_3'$										
			λ_2						$-\lambda_2'$		$(1-r_1)\beta_1$	β_2			μ_2			
				λ_3					α_b	$-\lambda_3'$								
					λ_4					θ	$-\lambda r_1'$							
						λ_5					$r_1\beta_1$	$-\lambda_5'$						
							λ0						$-\lambda_0'$	β_v				
								λ_1					α_v	$-\lambda_3'$				
									λ_2						$-\lambda_2'$		$(1-r_1)\beta_1$	β_2
										λ_3					α_b	$-\lambda_3'$		
											λ_4					θ	$-\lambda_4'$	
												λ_5					$r_1\beta_1$	$-\lambda_5'$

B_{01}=

	μ_0					
			μ_2			

B_{10}=

													λ_0					
														λ_1				
															λ_2			
																λ_3		
																	λ_4	
																		λ_5

A_0=

λ_0					
	λ_1				
		λ_2			
			λ_3		
				λ_4	
					λ_5

A_1=

$-\lambda_0''$	β_v				
α_v	$-\lambda_1'$				
η		$-\lambda_2'$		$(1-r_1)\beta_1$	β_2
		α_b	$-\lambda_3'$		
			θ	$-\lambda_4'$	
				$r_1\beta_1$	$-\lambda_5'$

A_2=

μ_0					
		μ_2			

where

$\lambda_0' = \lambda_0 + \mu_0 + \alpha_v$
$\lambda_0'' = \lambda_0 + \mu_0 + \alpha_v + \eta$
$\lambda_1' = \lambda_1 + \beta_v$
$\lambda_2' = \lambda_2 + \mu_2 + \alpha_b$
$\lambda_3' = \lambda_3 + \theta$
$\lambda_4' = \lambda_4 + \beta_1$
$\lambda_5' = \lambda_4 + \beta_2$

The cells with no entries indicate zero values.

Performance measures

Below, we provide some measures to characterize the system performance:

- **Average number of jobs in the system:** The average number of any type of jobs in the system is given by

$$E(N) = \sum_{i,j} P(i, j) \qquad ...(11)$$

The average number of prioritized jobs in the system is given by

$$E(B) = \sum_{i} P(i,2) \qquad …(12)$$

The average number of non-prioritized jobs in the system is given by

$$E(V) = \sum_{i} P(i,0) \qquad …(13)$$

- **Throughput:** It is defined as the total number of jobs served by the system. The total system throughput TP is obtained as

$$TP = \sum_{j}\sum_{i} (\mu_0 + \mu_2) P(i, j) \qquad …(14)$$

The throughput of the prioritized jobs is given by

$$TP(B) = \sum_{i} (\mu_2) P(i,2) \qquad …(15)$$

The throughput of the non-prioritized jobs is given by

$$TP(V) = \sum_{i} (\mu_0) P(i,0) \qquad …(16)$$

- **Delay:** This is the average delay in the service of a job, It is given by

$$D = \frac{E(N)}{TP} \qquad \ldots(17)$$

The delay in the service of prioritized jobs is given by

$$D(B) = \frac{E(B)}{TP(B)} \qquad \ldots(18)$$

The delay in the service of non-prioritized jobs is given by

$$D(V) = \frac{E(V)}{TP(V)} \qquad \ldots(19)$$

4. Numerical Illustrations:

We develop computational algorithm using software MATLAB. For illustration purpose, we consider an example and evaluate probability vectors and various performance characteristics by fixing the different system parameters as follows:

We have assumed η=.4, θ=.03 and N=50. Further the arrival rates, service rates, breakdown rates and repair rates are fixed as follows:

Arrival rates:

λ_0=60	λ_2=60
λ_4=36	λ_5=36
λ_1=15	λ_3=15

Service rates:

μ_0=50	μ_2=100

Breakdown rates:

α_v= .5	α_b= .5

Repair rates:

β_v= .2	β_1=1
β_2=2	r_1= .1

Sensitivity analysis of various parameters on system performance measures such as throughput, delay, average number of packets in both busy and working vacation state is done. Fig. 2 (a-d) depicts the effect of λ_2 on the different performance measures in busy state. Fig. 2 (a) reveals that as the arrival rate of prioritized requests increases, throughput also increases but since the service rate remains constant, the throughput drastically decreases after attaining a maximum limit. The graph is made for different values of μ_2 (μ_2=80, 90, 100). This follows that for higher traffic rates maximum peak attained is greater for higher values of service rates. With the help of given model one can compute the optimal service rate to be fixed for expected traffic intensity so that maximum throughput can be achieved and the quality of service can be maintained. Fig. 2 (b) compares the effect of increasing arrival rates on D(B) for different values of μ_2. As the arrival rate increases, D(B) shows a gradual increasing trend but after a particular arrival rate for the given values of service rate, delay of busy packets increases drastically. Besides, this delay is more for lower service rates; hence to minimize delay, service rate needs to be increased. Fig. 2(c) depicts the effect of arrival rates on the average number of jobs in the busy state. For a given value of service rate E(B) increases with increase in λ_2 till it reaches a maximum limit and then E(B) decreases drastically. The maximum limit indicates the maximum number of packets which the system can hold for the given set of μ_2 and λ_2.Similar effect can be seen in E(N) for different values of μ_2 and λ_2 as shown in fig. 2 (d).

Fig. 3 (a-d) shows the effect of λ_0 on different performance measures in working vacation. Fig. 3(a) compares the effect of λ_0 and TP(V) for different values of μ_0. As λ_0 increases TP(V) increases till it attains the maximum limit then decreases for a given value of μ_0. TP(V) is greater for higher service rates. The graph follows a similar pattern as fig. 2 (a). Fig. 3 (b) shows the effect of λ_0 on D(V)for different values of μ_0. The graph follows a similar behaviour as fig. 2 (b), i.e. as λ_0 increases, D(V) increases till a particular limit and then it decreases. Fig 3 (c) is also comparable with 2 (c) where effect of λ_0 is seen on E(V). As λ_0 increases, average number of packets in working vacation increases to a particular limit giving a threshold value beyond which the service degrades for given values of μ_0 and λ_0. Similar effect can be seen for E(N) as shown in fig. 3 (d).

Fig. 4 (a) exhibits the effect of breakdown rate in the busy state on throughput for different values of β_2. As α_b increases, TP(B) decreases. TP(B) is more for higher values of β_2. This illustrates that to increase the throughput of the system, repair rate for a given arrival rate should be increased. Fig 4 (b) shows the effect of α_b on D(B) for different repair rates. Delay of the system increases with increasing α_b and if the repair rate increases the delay decreases. D(B) for β_2=6 is less than D(B) for β_2= 4 and 2.

Fig. 5 (a-b) illustrates the effect of set up time of the repairman θ on E(B) and D(B). It can be noticed that if the repairman takes more set up time, then it causes the accumulation of average number of busy requests in the system and hence increasing the delay. Fig. 6 (a-b) depicts the effect of N-policy on the system's performance in terms of overall throughput and delay. If the value of N increases the overall system throughput decreases and the overall delay increases, thus adversely affecting the system's performance. The value of N specifies the switching of server to the busy state after accumulation of N prioritized requests. Hence to optimize the systems' performance the values of N should be kept low.

5. Conclusion

In this paper we have considered an application server which can handle prioritized and non prioritized jobs. When there are no prioritized requests, server handles non prioritized requests and enters a working vacation state where it serves the non prioritized jobs. It can again attain the busy state after accumulation of N packets of priority requests. Sensitivity analysis of various performance measures is carried out. Numerical illustrations reveal that we can compute the optimal service rate for the expected traffic situation. By increasing the service rate we can increase the throughput and decrease the delay for both the prioritized and non prioritized requests. Capacity of the server to hold the packets can also be increased by increasing the server rate. Breakdown rate also adversely affects the throughput and increases the delay in the system. By increasing the repair rate we can increase the throughput of the system. Further the set up rate for the repairman is another important parameter which should be controlled and kept low. Finally the effect of N-policy is considered which depicts that if the packet accumulation condition by the server to enter the busy state is high the performance of the system degrades, hence the value of N should be kept low. The paper is extremely helpful in developing the application servers to support various applications in a multitasking environment. Here the prioritized requests are served with a faster rate then the non prioritized requests and these types of systems can be applicable for supporting supply chain management, ecommerce systems, video-on-demand systems and other internet based applications.

References

1. Ke, J.C. and Wang, K.H. (2007). Vacation policies for machine repair problem with two type spares, *Appl. Math. Model.*, vol. 31, no. 5, pp. 880-894.
2. Chakravarthy, S.R. and Dudin, A.(2003). Analysis of a retrial queuing model with MAP arrivals and two types of customers, *Mathematical and Computer Modelling*, vol. 37, no. 3-4, pp. 343-363.
3. Mirchandany, R., Toesley, D. and Stankovic, J.A.(1990). Adaptive load sharing in heterogeneous distributed systems, *J. of Parallel and Dist. Comput*., vol. 9, no. 4, pp. 331-346.
4. Baily, D.E. and Neuts, M.F.(1981). Algorithmic methods for multi-server queues with group arrivals and exponential services, *E. J. of Oper. Research*, vol.8, no. 2, pp. 184-196.
5. Gillent, F. and Latouche, G.(1983). Semi-explicit solutions for M/PH/1-like queuing systems, *E. J. of Oper. Research*, vol. 13 no. 2, pp. 151-160.
6. Pla, V. and Giner, V.C.(2005). Analysis of priority channel assignment schemes in mobile cellular communication systems: a spectral theory approach, *Performance Evaluation* , vol. 59, no. 2-3, pp. 199-224.
7. Wu, D. and Takagi, H.(2006). M/G/1 queue with multiple working vacations, *Performance Evaluation*, vol. 63, no. 7, pp. 654-681.
8. Choudhury, G. and Madan, K.C.(2005). A two-stage batch arrival queueing system with a modified bernoulli schedule vacation under *N*-policy, *Mathematical and Computer Modelling*, vol. 42, no. 1-2, pp. 71-85.
9. Baba, Y.(2005). Analysis of a GI/M/1 queue with multiple working vacations, *Operations Research Letters*, vol. 33, no. 2, pp. 201-209.
10. Servi, L. D. and Finn, S. G.(2002). M/M/1 queues with working vacations (M/M/1/WV), *Performance Evaluation*, vol. 50, no. 1, pp. 41-52.
11. Wang, K.H., Chang, K.W. and Sivazlian, B.D.(1999). Optimal control of a removable and non-reliable server in an infinite and a finite M/H_2/1 queueing system, *Appl. Math. Model.*, vol. 23, pp. 651-666.
12. Lee, H.W., Yoon, S.H. and Seo, W.J.(1999). Start-up class models in multiple-class queues with N-policy, *Queueing Systems*, vol. 31, no.1-2, pp. 101-124.
13. Jain, M. and Poonam (2002): Optimal N-policy for the state dependent M/E_k/1 queue with server breakdown, *American Journal of Mathematical and Management Sciences*, USA.

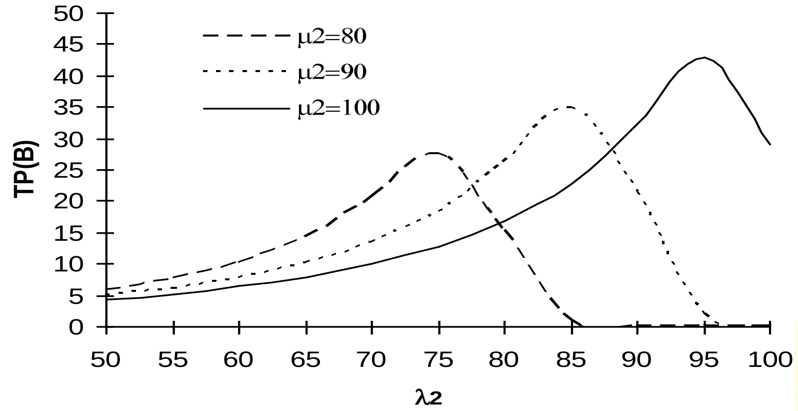

Fig. 2 (a): TP(B) by varying λ_2

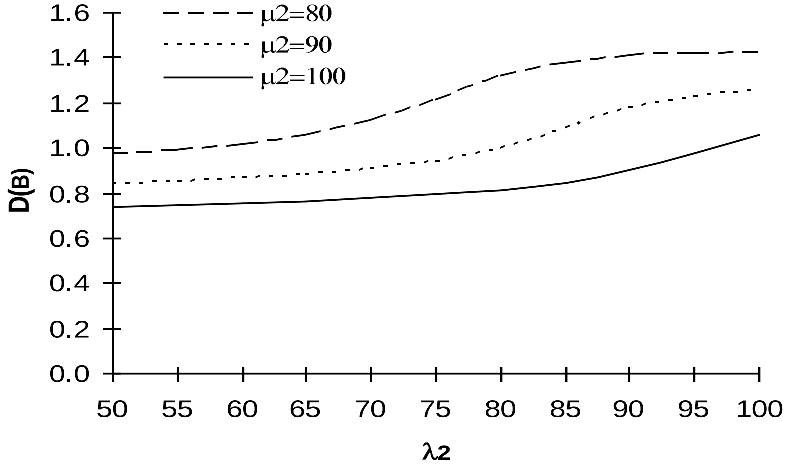

Fig. 2 (b): D(B) by varying λ_2

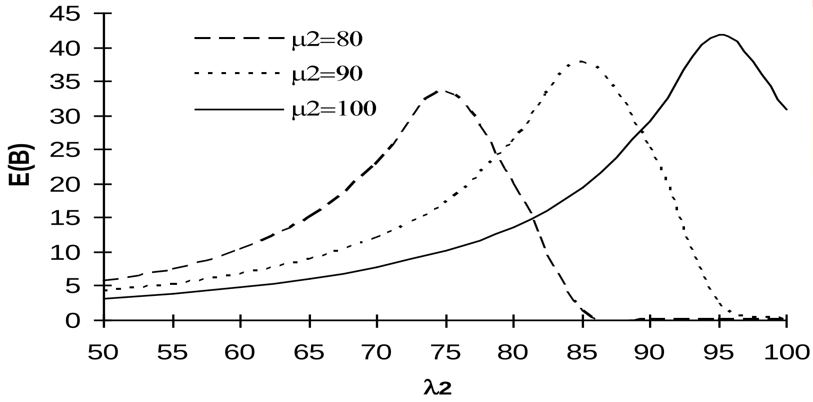

Fig. 2 (c): E(B) by varying λ_2

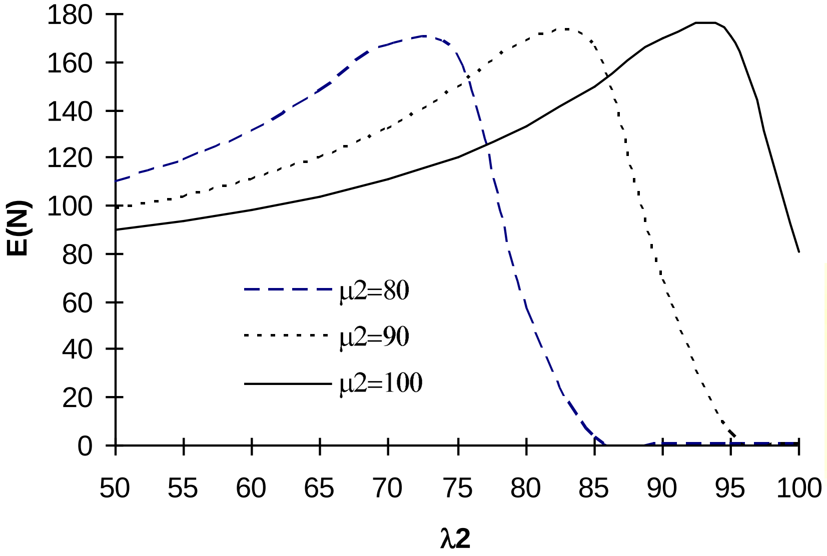

Fig. 2 (d): E(N) by varying λ_2

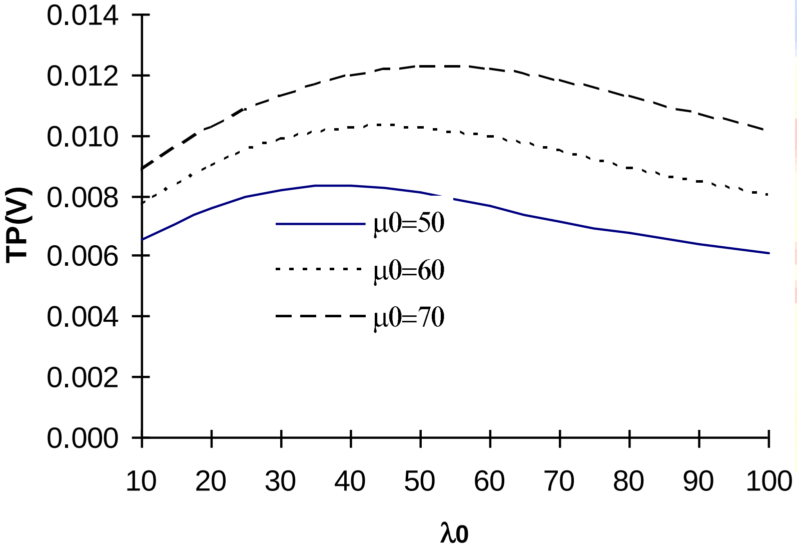

Fig. 3 (a): TP(V) by varying λ_0

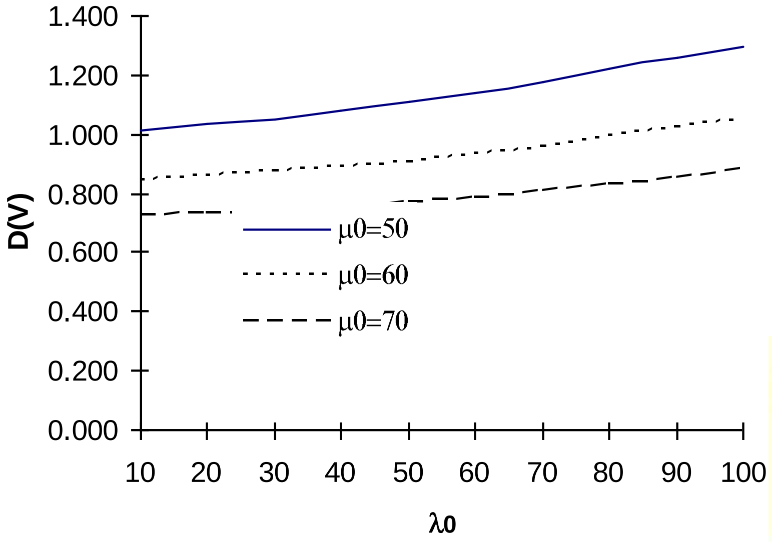

Fig. 3 (b): D(V) by varying λ_0

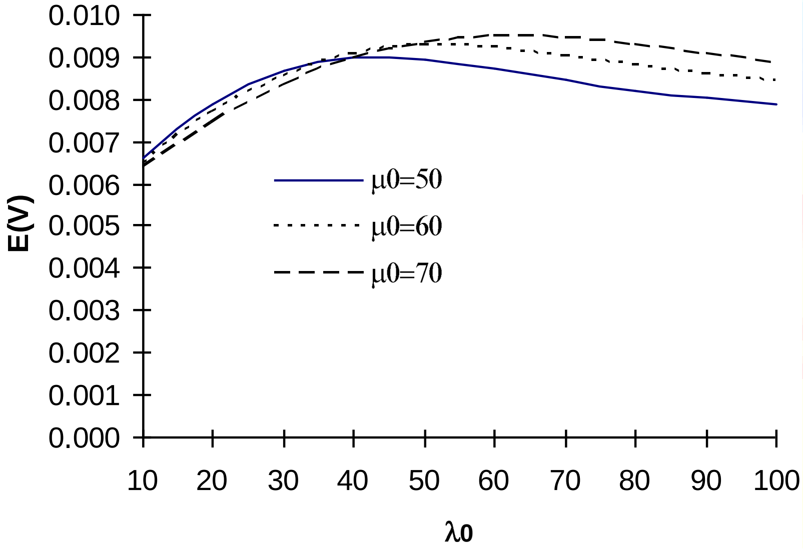

Fig. 3 (c): E(V) by varying λ_0

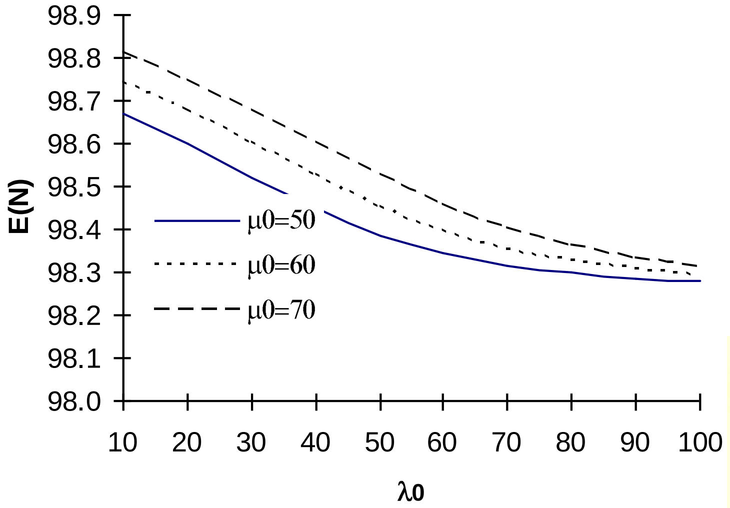

Fig. 3 (d): E(N) by varying λ_0

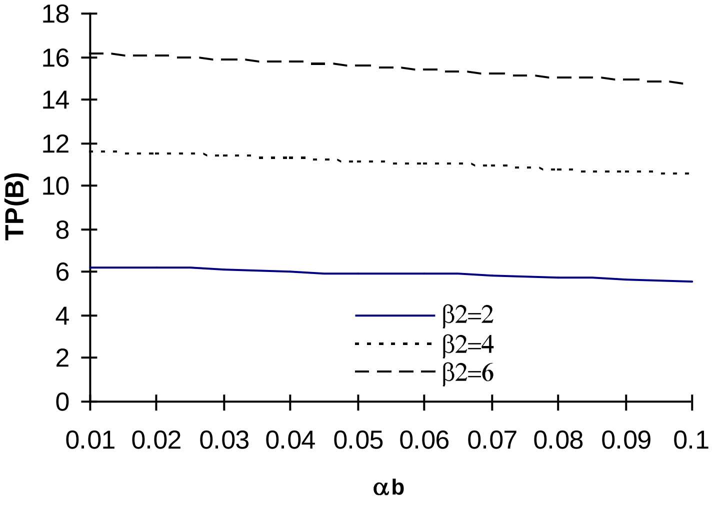

Fig. 4 (a): TP(B) by varying α_b

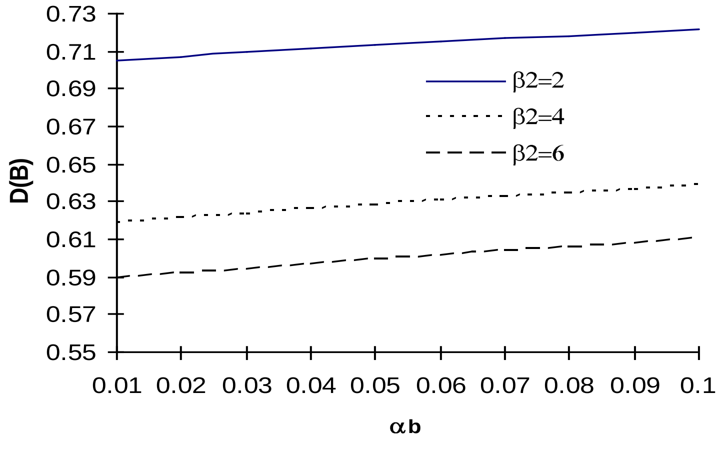

Fig. 4 (b): D(B) by varying α_b

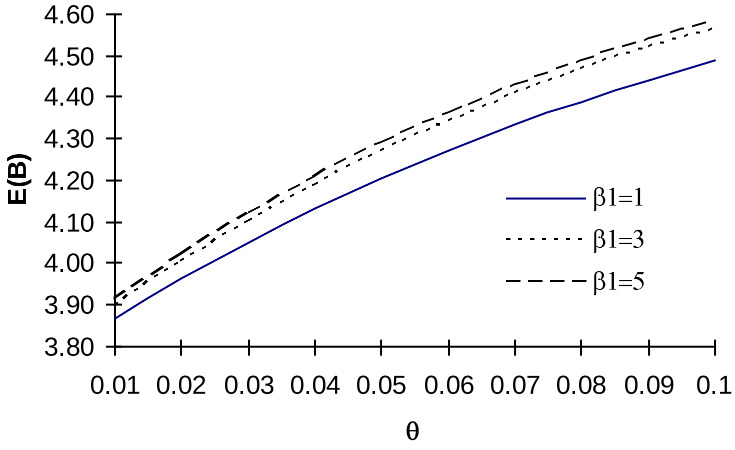

Fig. 5 (a): E(B) by varying θ

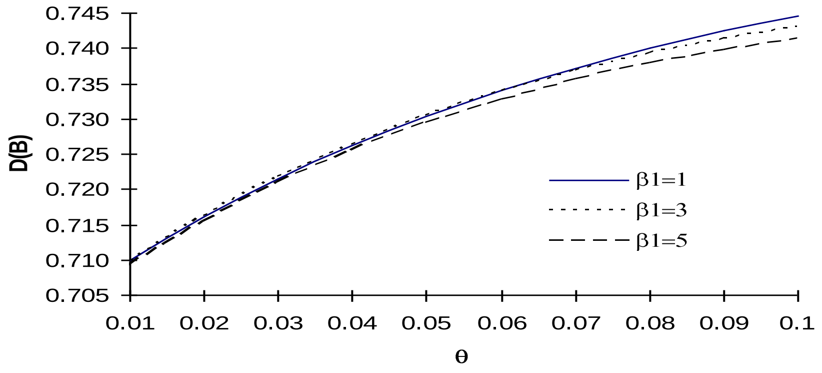

Fig. 5 (b): D(B) by varying θ

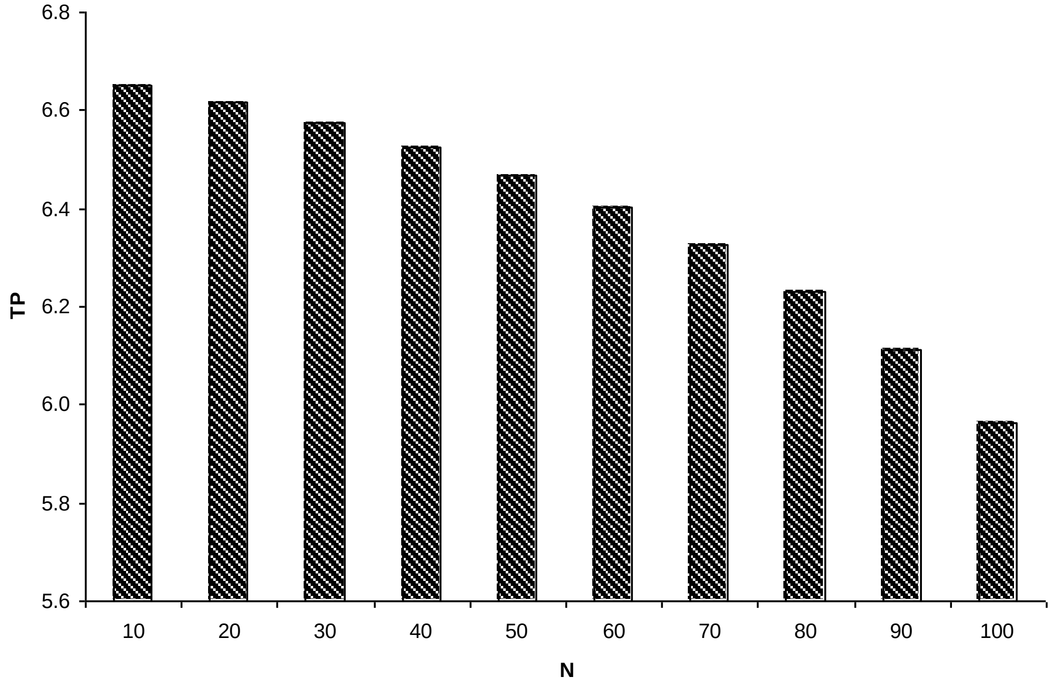

Fig. 6 (a): Effect of N-policy on TP

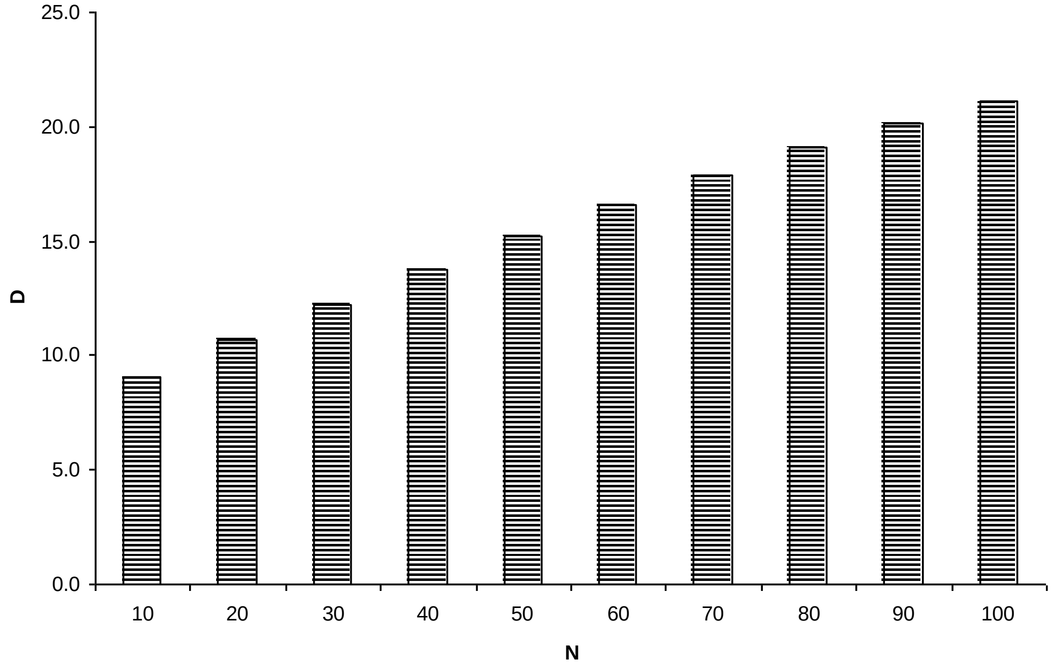

Fig. 6 (b): Effect of N-policy on D