

# K-限制式服務策略輪詢系統之最佳化

洪士程

朝陽科技大學資訊工程系  
e-mail : schong@cyut.edu.tw

陳炯文

朝陽科技大學資訊工程系  
e-mail : s9727632@cyut.edu.tw

## 摘要

在本文中，我們提出以序優化理論為基礎的演算法，在有限的計算時間內，解決 G/G/1/K 輪詢系統與 K-限制式服務策略的最佳化問題；以到達率在短時間內惡化不明顯為前提，我們將提出的方法採用兩個階段方式來進行。在第一階段，採用了典型的基因算法，從龐大的離散解答的空間  $\Omega$  來選擇  $N = 1024$  個粗略好的解決方案，接著使用離線訓練的人工類神經網路做為評估適應函數的替代模型。第二階段包括若干的子階段，由先前獲得的  $N$  個粗略好的解中，挑選估計足夠好的解決辦法，而在最後的子階段獲得的解，即為我們所尋得的足夠好的解。測試結果表明，我們取得的足夠好的解，在解答空間有 0.99 的機率，是在最好的  $3.31 \times 10^{-6}$  % 之內。

**關鍵詞：**輪詢系統，K-限制式服務策略，最佳化，序優化，人工類神經網路。

## Abstract

In this paper, we propose an ordinal optimization theory based algorithm to solve the optimization problem of G/G/1/K polling system with k-limited service discipline for a good enough solution using limited computation time. We assume that the arrival rates do not deteriorate visibly within a very short period. Our approach consists of two stages. In the first stage, we employ a typical genetic algorithm to select  $N=1024$  roughly good solutions from the huge discrete solution space  $\Omega$  using an offline trained artificial neural network as a surrogate model for fitness evaluation. The second stage consists of several substages to select estimated good enough solutions from the previous  $N$ , and the solution obtained in the last substage is the good enough solution that we seek. The results show that the good enough solution that we obtained is among the best  $3.31 \times 10^{-6}$  % in the solution space with probability 0.99.

**Keywords:** Polling systems, k-Limited service disciplines, optimization, Ordinal optimization,

artificial neural network.

## 1 緣由與目的

輪詢系統在各式各樣模型化和分析的應用中扮演著重要的角色，像在計算機網路、通訊、製造系統、運輸系統，以及最近用於模型化其他需求的系統，例如生產和存貨系統。通常來講，這些系統都是極其錯綜複雜，因而輪詢系統的研究已經持續超過半個世紀。目前，大量的文章在探討輪詢模式的排隊分析，以及眾多的參考資料，與透過數值分析所得到各式各樣的結果[1]-[3]。

輪詢系統是根據事先訂定的拜訪計畫和服務策略透過一台伺服器來進行服務的多重佇列系統。通常，伺服器假設為依照循環型態來處理這些佇列。很自然的可以假設，如果伺服器在拜訪一個佇列時已至少有一個客戶在等待，服務將會立即開始。即使當系統是空的時候，伺服器仍繼續循環運行。當伺服器從一個佇列轉換到下一個佇列時將會耗費切換時間，此切換時間通常假設與到達過程及服務過程並無相關。在大多數現有文獻的輪詢模型，每個佇列的排隊原則為先進先出(FIFO)。不過，對於伺服器離開佇列的時機則有幾個服務策略需要考慮。底下有五種服務策略是屬於較為典型的部分。

在一個消耗式(exhaustive)服務策略系統，伺服器持續在佇列中為全部客戶效勞，直到全部清空為止。在一個開門式(gated)服務策略系統，伺服器只連續服務當它到達時在佇列中所發現的客戶數量[4]。在一個限制式(limited)服務策略系統，每一佇列中最多只有一個客戶可被服務。在一個 k-限制式(k-limited)服務策略系統，伺服器持續在佇列中最多可為 k 個客戶效勞，或直到全部清空為止，看哪一個情形先發生。在一個時間限制式(time-limited)服務策略系統，伺服器在一個佇列服務固定的時間，即使它是空的[5]。每個服務策略表示一個決定策略來達成輪詢系統的特定性能，例如平均的等待時間。對於輪詢模型不同的服務策略已發展出許多分析

技術來計算平均等待時間[6]。

由於系統的容納能力有限，因此存在有三種不同的系統。在一個單一緩衝區系統裡，每個佇列最多一次能夠容納一個客戶，而到達時發現緩衝區已被佔用的那些客戶將被遺失。在一個有限緩衝系統內，每個佇列有限制的容量，當佇列已滿時，新到的客戶也會遺失。在一個無限緩衝區系統內，任何客戶在每個佇列上可以無限等待而不會被遺失。基於分析方便，以往在這領域內的研究主要鑽研於無限緩衝區系統，然而在現實中卻不實際。

近年來，很多努力投入於最佳化的研究[7]。他們提供關於一個系統具有波以松(Poisson)外部到達、一般的循環服務和一般的切換時間，分析在系統內的期望客戶數量與期望等待時間等結果。在大多數論文裡關於輪詢模型通常假設為在不同的佇列的到達過程是彼此獨立的波以松過程。不過，在許多輪詢模型的應用中不同的佇列到達過程是相關的。當佇列的數量增加時，由於在系統裡佇列彼此間的相互關係和不同的隨機過程，使得排隊分析將變得更錯綜複雜。於是，非侷限在波以松到達過程與有限緩衝區的佇列，以現有輪詢系統性能上的數學解析結果將無法直接拿來使用。

## 2 問題陳述

### 2.1 輪詢模型與數學模式

這裡所考慮的輪詢模型包含一個伺服器， $J$  個有限長度的佇列編號為  $Q_1, \dots, Q_J$ 。每一個佇列為 G/G/1/K 型式：(i) 客戶依據常態分布的過程到達  $Q_j$ ，(ii) 每一個進來的客戶需要耗費一段隨機的服務時間，(iii) 服務所需的時間是相同且獨立的常態分佈，同時與到達過程彼此獨立，(iv) 每一個佇列的有限長度為  $K$ 。伺服器依照底下的順序循環挑選佇列  $Q_1, Q_2, \dots, Q_J, Q_1, Q_2, \dots$ ，而且依照特定的服務策略(消耗式，開門式，限制式，k-限制式，時間限制式等)來進行服務。當伺服器由  $Q_j$  轉到  $Q_{j+1}$  時，將會耗費一個隨意產生的切換時間，而且就算系統為空蕩蕩時伺服器仍會繼續循環繞行。到達的過程，服務時間與切換時間彼此之間是假設為互相獨立；G/G/1/K 的輪詢模

型如圖 1 所示。

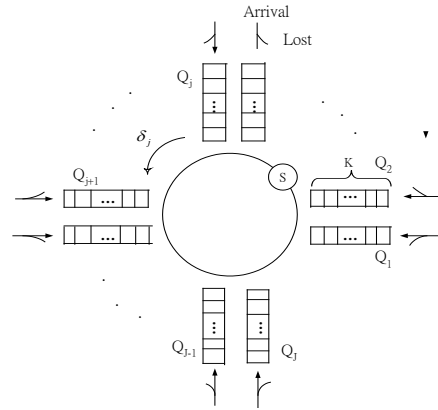


圖 1. G/G/1/K 輪詢模型

G/G/1/K 的模型是較為符合實際情況，但是對於在第一章提及的現有的服務策略來說，分析系統性能將引起很大的困難。既然很難對現有的服務策略得到任何分析的公式來評價系統性能，若能為 G / G / 1 / K 輪詢系統設計一個服務策略來獲得更好的系統性能將會更為實際。

定義隨機模擬程序的符號說明如下：輸入決定向量  $b = [b_1, \dots, b_J]^T$  及到達率  $\lambda = [\lambda_1, \dots, \lambda_J]^T$ 。  $w_{i,j}$ ：在  $Q_j$  中第  $i$  個客戶的等待時間。  $a_{i,j}$ ：在  $Q_j$  中第  $i$  個客戶的到達間隔時間。  $v_{i,j}$ ：在  $Q_j$  中第  $i$  個客戶的服務時間。  $q_j$ ：  $Q_j$  的長度。  $ns_j$ ：在  $Q_j$  中已完成服務的客戶數量。  $nl_j$ ：在  $Q_j$  中被遺失的客戶數量。 EOS：停止模擬所需之已完成服務的客戶的總數量。較長模擬次數之 EOS=10000，較短模擬次數之 EOS=1000。

這裡所採用的服務策略是 k-限制式服務策略，也就是當伺服器開始服務  $Q_j$  佇列時，伺服器持續在佇列中為  $b_j (\leq K)$  個客戶效勞，或直到全部清空為止，看哪一個會先發生，其中的  $b_j \in \{1, 2, \dots, K\}$ ，  $j = 1, 2, \dots, J$ 。當伺服器由  $Q_j$  轉到  $Q_{j+1}$  時，將會耗費一個隨機產生的切換時間。因此 k-限制式服務策略的決定變數空間如下公式(1)所示：

$$\Omega = \{m = (b_1, b_2, \dots, b_J) \mid 1 \leq b_j \leq K, \forall j\} \quad (1)$$

假設當一個客戶完成服務後將會離開系統，而任何客戶在被服務尚未結束前不會被任何因素所中斷(也就是沒有先買權)。當伺服器

由  $Q_j$  轉到  $Q_{j+1}$  時的切換時間，是假設為具有平均值  $\delta_j$  與變異數  $\sigma_j^2$  的常態分布。在此設定隨機變數  $W_j$  為  $Q_j$  在穩定狀態時典型客戶的等待時間，等待時間定義為客戶從到達一直到開始被服務的這段時間。因此， $E[W_j]$  代表  $Q_j$  在穩定狀態時每位客戶的平均等待時間。我們以  $\tau_j$  代表  $Q_j$  的等待成本係數（一位客戶等待一個單位時間所需的成本），於是  $\frac{1}{J} \sum_{j=1}^J \tau_j E[W_j]$  表示 G/G/1/K 輪詢模型的平均等待成本。接著設定隨機變數  $P_j$  為  $Q_j$  在穩定狀態時客戶損失的百分比，也就是  $P_j = (\text{損失的客戶數目} / \text{到達的客戶數目})$ ，且以  $\eta_j$  代表  $Q_j$  的損失成本係數（損失一位客戶所需的成本），於是  $\frac{1}{J} \sum_{j=1}^J \eta_j E[P_j]$  表示 G/G/1/K 輪詢模型的平均損失成本。當系統達到穩定狀態時，操作整個系統的期望成本為  $\frac{1}{J} \sum_{j=1}^J (\tau_j E[W_j] + \eta_j E[P_j])$ 。

接下來便可以將 G/G/1/K 輪詢系統公式化成底下公式(2)的隨機模擬最佳化問題：

$$\min_{b_1, \dots, b_J} \frac{1}{J} \sum_{j=1}^J (\tau_j E[W_j] + \eta_j E[P_j]) \quad (2)$$

換句話說，我們是在找尋一組最佳化的 k-限制式服務策略  $m = (b_1, b_2, \dots, b_J)$ ，來使操作整個 G/G/1/K 輪詢系統的期望成本降至最低。如果  $K=20$ ，那麼(2)式的決定變數空間的大小為  $K^J$ ，若  $J=10$  則  $K^J$  為  $20^{10}$ 。注意  $b_j$  的值是允許超過  $K$ ，因為當伺服器在服務一位客戶時，新的客戶會持續到達，然而  $b_j (> K)$  的設定值並不是一個很好的選擇，除非客戶到達率非常的高。因此，問題(2)是一個具有巨大決定變數空間如同(1)式所描述的隨機最佳化問題，非常適合利用我們提出的序的最佳化方法來且求解一個足夠好的 k-限制式服務策略。

## 2.2 隨機模擬最佳化問題

模擬最佳化問題可以視為一個模擬系統的最佳化問題，其系統的輸出可以透過模擬來評估，不論是透過模擬系統的真实模擬或只是透過電腦的模擬[8]。因此，模擬最佳化的目標

是找出模擬系統決定變數的最佳設定值，使得輸出變數能夠達到最好或最佳的狀況。針對此目的已經發展出很多方法，例如梯度搜尋為基礎的方法[9]，隨機近似方法[10]，反應曲面法[11]，以及啟發式方法。這些方法在[12]中已被完整的探討。其中，啟發式方法包括基因演算法(Genetic Algorithm, GA)[13]，模擬退火法(Simulated Annealing, SA)[14]與禁制搜尋法(Tabu Search, TS) [15]則常常使用於模擬最佳化[16]中。

上述的啟發式方法除了些許成功的應用之外[17]，在其他方面的應用上遭遇了許多技術上的困難與障礙[18]。在這些當中最主要考慮的是速度，因為對於給定的決定變數使用模擬方式來評價輸出性能，當決定變數的空間是巨大時搜尋最好的決定變數確實需要耗費昂貴的計算。而且，模擬經常面對的情況是瞬息變化的特性也屬於整個問題的一部分。因此，隨機的變化特性只要存在著便會使模擬最佳化問題顯的錯綜複雜。這篇研究的主題就是如何有效地來解決這個富有挑戰性的隨機模擬最佳化問題。

在此所考慮的隨機模擬最佳化問題陳述如下公式(3)：

$$\min_{m \in \Omega} E[F(m)] \quad (3)$$

其中的  $\Omega$  為巨大的決定變數空間， $m$  為系統需要的決定變數設定值； $E(\cdot)$  表示  $(\cdot)$  的期望值，而  $F(\cdot)$  表示模擬系統的輸出函數， $E[F(m)]$  表示為目標函數。在這篇研究中考慮到的複雜系統， $F(m)$  的值只可以透過複雜計算的模擬來獲得。為了克服這個問題的計算複雜度，我們將採用序的最佳化(Ordinal Optimization, OO)理論為基礎的目標軟化方法[19]-[20]，可以有效地找出一組具有高可靠性且足夠好的解來取代傳統搜尋最好的解，即使使用的是一個粗略的模型，不過可發現這個粗略的模型仍然保留著決定變數空間設定值的性能順序。從這裡開始，我們將使用設定值來代表決定變數空間的設定值。

序的最佳化理論為基礎的目標軟化方法主要觀念，在於逐次減少搜尋空間大小，這個目標軟化搜尋方法可以描述如下[19]：(i) 均勻的從  $\Omega$  當中挑選出  $N$ ，例如 1000，個設定值，(ii) 對於所考慮的問題使用一個粗略的模型來評估及排序這  $N$  個設定值，然後從這  $N$  個設

定值所構成的選擇子集合(SS)當中挑出最佳的  $S$ ，例如 35，個設定值，可作為估計出來足夠好的子集合(GS)。(iii) 使用完整模型來評估及排序這  $S$  個設定值，然後從這  $S$  個當中挑選出表現最好的設定值。序的最佳化定理已證明，當(1)裡的  $N=1000$  且在(ii)中使用一個粗略且含有適當雜訊的模型，在(iii)裡從  $S \leq 35$  個當中挑選出來的最好的設定值會落在 GS 的機率為 0.95，這裡的 GS 代表從  $N$  當中收集到的最好 5% 真正足夠好的設定值。這意味著從(iii)裡挑選出來在 SS 當中最好的設定值，機率高達 0.95 會落在  $N$  個裡面最好的 5% 當中。

然而，若以  $\Omega$  來代替  $N$  個設定值，我們所尋找之(3)式足夠好的解，要能夠代表一個足夠好的設定值除非是  $\Omega$  跟  $N$  一樣的小[21]。如同由林與何在最近的論文[22]中指出，在一個適當的模型雜訊下，均勻的選取  $N$  個設定值裡最好的 3.5% 將具有高的機率 ( $\geq 0.99$ ) 會落在  $\Omega$  個設定值裡最好的 5% 當中，如果沒有模型誤差的話則最好的設定值將落在  $\Omega$  個裡最好的 3.5% 當中。然而，對於一個  $\Omega$  大小為  $10^{30}$ ，最好的 3.5% 設定值是落在最好的  $3.5 \times 10^{28}$  個裡面。對於全部有  $10^{30}$  個解，而要找尋的解落在  $3.5 \times 10^{28}$  個裡面，在實際應用上並不是一個具有高機率足夠好的解。如此無法直接套用現存目標軟化搜尋方法，因而需要發展一個新的方法從  $\Omega$  當中來挑選  $N$  個粗略足夠好的設定值以取代(i)，以確定最後挑選到的設定值是公式(3)真正具有高機率且足夠好的解。

以啟發式演算方法來得到  $N$  個粗略足夠好的設定值，是需要依靠對於考慮之問題所了解的知識多寡。例如在最佳電力流問題具有離散控制變數，林等[23]提出一個以 OO 理論為基礎的演算法以及工程師的觀點來挑選  $N$  個粗略足夠好的離散控制變數向量。然而，以工程師的觀點只能在特定的系統上運作。因此，在本篇研究中，我們將提出一個以 OO 理論為基礎的系統化方法來從  $\Omega$  當中挑選出  $N$  個粗略足夠好的設定值，並且結合現存目標軟化搜尋方法來尋找公式(3)足夠好的解。

### 3 從決定變數空間挑選 $N$ 個足夠好的設定值

如同在 OO 理論所指出[19]-[20]，設定值順序的性能即使以粗略的模型來評估也會被保

留。因此，要從  $\Omega$  中挑選  $N(=1000)$  個足夠好的設定值且不花費太多的計算時間，必須建構一個粗略的模型。只要給定一組設定值  $m = (b_1, b_2, \dots, b_j)$  便可以方便地估計出公式(2)的目標函數值，接著使用一個有效的方法來挑選出  $N$  個足夠好的設定值。我們建構之粗略的模型為類神經網路 ANN [24]，而挑選機制為基因演算法 GA [13]

#### 3.1 以類神經網路(ANN)為基礎的模型

考慮輸入為設定值  $m \in \Omega$  以及輸出為相對應的目標函數值  $E[F(m)]$ ，可以使用一個類神經網路(ANN)來完成輸入與輸出之間的對應關係[24]。首先，先從  $\Omega$  中均勻挑選出  $M$ ，例如 500，個設定值代表  $\Omega$  的子集合。然後利用一個近似的模型來評估這  $M$  個設定值的  $E[F(m)]$ ，可以透過一個具有適當模擬次數，例如 10000，的隨機模擬來完成，如同在[21]裡指出。這些收集到的  $M$  個輸入-輸出對  $(m, E[F(m)])$  將被用來訓練 ANN 以調整每個類神經的權重。一旦 ANN 訓練完畢，就可以輸入任意的設定值  $m$  到 ANN 裡，由 ANN 的輸出可得到相對應的估計值  $E[F(m)]$ ；在此對於一個設定值  $m$  可以避免需要正確且冗長的隨機模擬來評估  $E[F(m)]$ 。此部分形成了粗略的模型，對於一個給定的設定值  $m$  可以粗略且有效地來評估(3)式的目標函數值。這裡在乎的是相關的順序性能而不是  $F(m)$  的值，因此採用一個簡單的兩層順向 ANN 當作模型，而為了加速收斂速度，則採用比例共軛梯度演算法[27]來訓練 ANN。

#### 3.2 基因演算法(GA)

藉由以上有效的目標函數評估模型，便可以有效地使用啟發式全域搜尋方法從  $\Omega$  中挑選出  $N$  個粗略好的設定值。既然 GA 可以透過上一代至下一代的演化來改進族群的適應函數值，很適合我們的需要。有了上述有效率的目標函數值 (在 GA 術語上稱做適應函數值) 評估方法，可以使用 GA 很有效率的來挑選出  $N$  個出色的設定值。

GA 中有兩個準則可用來判斷是否要停止演化，第一個準則是當前後兩個世代裡面最好的族群的適應函數值已不再改進，另一個準則是已經演化了足夠多的世代。當套用 GA 進行

演化一直到收斂或停止時，將最後一個世代的所有族群  $I$  依照適應函數值由小排列到大，挑選出前面  $N$  個族群，即是我們要找的足夠好的  $N$  個設定值。

### 3.3 由 $N$ 中搜尋足夠好的解

由  $N$  個挑選出來的粗略好的解開始，緊接著將直接利用在 2.2 節描述的現存目標軟化搜尋方法的第(ii)步驟。在這個步驟將使用一個比 ANN 較準確的模型，也就是透過適當數量模擬次數的隨機模擬來評估每一個設定值的目標函數值。我們將依照估計的目標函數值來排序這  $N$  個設定值，然後挑選出最好的  $S$  個設定值當作是選擇子集合(SS)。接著，我們將使用精確的模型來評估這  $S$  個，例如 35，設定值，如同在現存目標軟化搜尋方法的第(iii)步驟。這個精確的模型使用一個足夠大數量模擬次數的隨機模擬，對於給定的  $m$  能夠使得評估值  $E[F(m)]$  變的穩定。在所有  $S$  個設定值當中具有公式(3)最小的目標函數值即是我們要找的足夠好的解。

### 3.4 以 OO 理論為基礎的演算法

接下來，以 OO 理論為基礎的演算法來求解(3)式足夠好的解可以整理如下：

**步驟 1：**從  $\Omega$  中均勻的選取  $M$  個設定值  $m$  當做輸入，使用一個近似的模型以  $L_m$  模擬次數執行圖二中的模擬程序，得到相對應的  $E[F(m)]$  近似值。接著訓練類神經網路，完成  $M$  個輸入與輸出，亦即  $M(m, E[F(m)])$  之間的對映關係。

**步驟 2：**從  $\Omega$  中隨機產生  $I$  個設定值當做基因演算法的初始族群。套用基因演算法到這些族群，採用輪盤選擇法，單點交配方式且  $p_c = 0.7$ ，突變機率  $p_m = 0.02$ ，並使用步驟 1 所訓練出來的類神經網路當做有效且實際的適應函數計算模型。當基因演算法經過  $G=20$  世代後，將所有  $I$  個設定值依照適應函數值由小到大排列，挑選出前面最好的  $N (=1000)$  個設定值。

**步驟 3：**將步驟 2 所得到的  $N$  個設定值進行適當數量模擬次數的隨機模擬，例如  $L_m$  的模擬次數，得到(3)式的各個目標函數值。將這  $N$  個設定值依照目標函數值由小到大排列，挑選出前面最好的  $S (=35)$  個設定值。

**步驟 4：**對所有  $S$  個設定值進行設定值進行足夠大數量模擬次數的隨機模擬，例如  $L_s$  的模擬

次數，得到(3)式的各個目標函數值。找出其中最小的目標函數值所對應的設定值，就是我們要尋找的足夠好的解。

### 3.5 G/G/1/K 輪詢模型的應用

決定變數空間  $\Omega$  的大小是巨大的，例如，對於一個 10 個佇列的系統，每個佇列含有 20 個緩衝容量，那麼整數值  $b_j$  的可能範圍是  $[1,20]$ 。因而(2)式的決定變數空間大小為  $20^{10}$ 。因此，隨機模擬最佳化問題(2)非常適用於 OO 理論為基礎的演算法。所考慮的問題輸入與輸出之間的關係描述於圖 3。

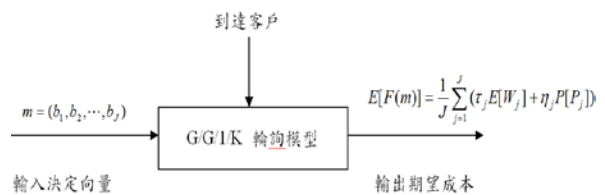


圖 2. G/G/1/K 輪詢模型的輸入與輸出關係

## 4 測試結果與比較

### 4.1 M/M/1/K 輪詢系統的測試結果

針對 M/M/1/K 輪詢系統來測試我們提出的方法，先設定以下的參數值：共有十個有限長度的佇列 ( $J=10$ )；每一個佇列的有限長度為  $K$ ；第  $j$  個佇列的到達程序為波以松分布，而對於  $j=1, \dots, J$  的到達率為  $\lambda_j$  如表 1 第二列所示；服務時間為指數分布且服務率為  $\mu=30$ ；切換時間為常態分布具有平均值  $\delta=1/30$  與標準差  $\sigma=0.01$ ；十個佇列的等待成本係數  $\tau_j$  與遺失成本係數  $\eta_j$ ，如表 1 第三列與第四列所示。

表 1. 十個佇列的到達率與等待成本係數及遺失成本係數。

$j$	1	2	3	4	5	6	7	8	9	10
$\lambda_j$	1	2	3	2	1	1	2	3	2	1
$\tau_j$	200	100	10	100	200	200	100	10	100	200
$\eta_j$	$10^3$	500	100	500	$10^3$	$10^3$	500	100	100	$10^3$

在 OO 理論為基礎的演算法中設定以下參



數: 在步驟 1 設定  $M=500$ , 在步驟 2 設定  $I=1000$  與  $N=1000$ , 在步驟 3 設定  $L_m=1000$  與  $S=35$ , 在步驟 4 設定  $L_s=10000$ 。

由 OO 理論為基礎的演算法所得到之足夠好的決定向量, 以及相對應的操作系統期望成本, 對於兩個不同長度的  $K(=20 \text{ and } 30)$  其結果顯示在表 2。我們方法所耗費的 CPU 時間為 2 分鐘, 如此可以配合即時的應用。同時針對耗盡式、閘道式、限制式與時間限制式(=3 個時間單元)等服務策略, 套用相同的輪詢系統使用步驟 4 精確的模型以同樣數量的客戶來進行模擬。

**表 2.** 不同長度  $K$  所得到  $k$ -限制式服務策略足夠好的決定向量與操作系統期望成本。

長度	決定向量										ECOS <sup>†</sup>
$K=20$	7	5	3	19	8	7	20	2	19	16	380.43
$K=30$	25	17	6	19	18	27	21	7	18	18	497.64

<sup>†</sup>ECOS: 操作系統期望成本

對於兩個不同情形  $K$  所得到的操作系統期望成本顯示如表 3:

**表 3.** 其他現存服務策略之操作系統期望成本。

長度	服務規則	k-limited	exhaustive	gated	limited	time-limited
$K=20$	ECOS <sup>†</sup>	380.43	561.74	850.15	1685.1	1316.4
	ECOS - $\sigma^j$ * .100%	0%	47.65%	123.46%	342.94%	246.02%
$K=30$	ECOS <sup>†</sup>	497.64	854.86	1137.4	2051.3	1540.6
	ECOS - $\sigma^j$ * .100%	0%	71.78%	128.56%	312.21%	209.58%

<sup>†</sup>ECOS: 操作系統期望成本

<sup>‡</sup>\*: 我們的方法得到之 ECOS

在表 3 中, 同時顯示出我們的方法並比較其他現存的服務策略所節省的操作系統期望成本百分比。由以上結果, 發現我們的方法表現超越其他方法, 以及  $k$ -限制式服務策略的優越性。

## 4.2 G/G/1/K 輪詢系統的測試結果

在這一節, 我們將使用三個不同的模式進行模擬,  $\Gamma/\Gamma/1/K$  ( $\Gamma = \text{Gamma}$ ),  $\text{LN}/\text{LN}/1/K$  ( $\text{LN} = \text{lognormal}$ ), 與  $\text{W}/\text{W}/1/K$  ( $\text{W} = \text{Weibull}$ ) 模式。

針對以上三個輪詢系統來測試我們的方法, 先設定以下的參數值: 共有十個有限長度的佇列 ( $J=10$ ); 每一個佇列的有限長度為  $K=20$ ; 切換時間為常態分布具有平均值  $\delta=1/30$  與標準差  $\sigma=0.01$ ; 十個佇列的等待成本係數  $\tau_j$  與遺失成本係數  $\eta_j$ , 如表 4 的第二列與第三列所示。在  $\Gamma/\Gamma/1/K$  輪詢系統, 對於  $j=1, \dots, J$  設定到達的間隔時間分布平均值  $1/\lambda_j$ , 如表 4

的第四列, 服務時間分布的平均值  $1/\mu = 1/30$ ; 到達的間隔時間與服務時間分布的形狀參數為  $\alpha=2$ 。在  $\text{LN}/\text{LN}/1/K$  輪詢系統, 對於  $j=1, \dots, J$  設定到達的間隔時間分布平均值  $1/\lambda_j$  與標準差  $\delta_{A_j}$ , 如表四的第四列與第五列所示, 服務時間分布的平均值  $1/\mu = 1/10$  與標準差  $\delta_s=0.1$ 。在  $\text{W}/\text{W}/1/K$  輪詢系統, 對於  $j=1, \dots, J$  設定到達的間隔時間分布平均值  $1/\lambda_j$  與形狀參數  $\beta_{A_j}$ , 如表 4 的第四列與第六列所示, 服務時間分布的平均值  $1/\mu = 1/20$  與形狀參數  $\beta_s=0.5$ 。

**表 4.** 十個佇列的參數設定與等待成本係數及遺失成本係數。

$j$	1	2	3	4	5	6	7	8	9	10
$\tau_j$	20	10	1	10	20	20	10	1	10	20
$\eta_j$	100	50	10	50	100	100	50	10	50	100
$1/\lambda_j$	1	$1/2$	$1/3$	$1/2$	1	1	$1/2$	$1/3$	$1/2$	1
$\delta_{A_j}$	0.1	0.1	0.01	0.1	0.1	0.1	0.1	0.1	0.1	0.1
$\beta_{A_j}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

$\Gamma/\Gamma/1/K$ ,  $\text{LN}/\text{LN}/1/K$ , 與  $\text{W}/\text{W}/1/K$  三個輪詢模式, 由 OO 理論為基礎的演算法所得到之  $k$ -限制式服務策略足夠好的決定向量與相對應的操作系統期望成本, 顯示在表 5。

**表 5.**  $k$ -限制式服務策略足夠好的決定向量與操作系統期望成本。

輪詢模式	決定向量										ECOS <sup>†</sup>
$\Gamma/\Gamma/1/K$	2	12	16	17	7	7	14	11	19	4	132.01
$\text{LN}/\text{LN}/1/K$	9	12	15	7	2	5	10	19	12	7	312.89
$\text{W}/\text{W}/1/K$	13	9	15	6	8	4	13	15	16	5	378.37

<sup>†</sup>ECOS: 操作系統期望成本

現存的其它服務策略的操作系統期望成本, 顯示在表 6, 同時顯示我們的方法比較其他現存的服務策略所節省的操作系統期望成本百分比。由以上結果, 發現我們的方法表現超越其他方法, 以其  $k$ -限制式服務策略的優越性。

**表 6.** 其他現存服務策略之操作系統期望成本。

輪詢模式	服務規則	k-limited	exhaustive	gated	limited	time-limited
$\Gamma/\Gamma/1/K$	ECOS <sup>†</sup>	132.01	224.92	406.31	938.94	874.77
	ECOS - $\sigma^j$ * .100%	0%	70.38%	207.78%	611.26%	562.65%
$\text{LN}/\text{LN}/1/K$	ECOS <sup>†</sup>	312.89	504.75	716.13	1031.8	1185.6
	ECOS - $\sigma^j$ * .100%	0%	61.32%	128.87%	229.76%	278.91%
$\text{W}/\text{W}/1/K$	ECOS <sup>†</sup>	378.37	615.54	711.49	969.28	1098.2
	ECOS - $\sigma^j$ * .100%	0%	62.68%	88.04%	156.17%	190.24%

<sup>†</sup>ECOS: 操作系統期望成本

<sup>‡</sup>\*: 我們的方法得到之 ECOS

## 5 成果自評與未來發展

本研究完成下列之工作項目及具體成果：

- (1) 形成 G/G/1/K 輪詢系統之隨機最佳化問題。
- (2) 建立模擬的模式。
- (3) 建立以類神經網路尋找設定值解所對應的近似目標函數值。
- (4) 建立基因演算法則以尋求序的最佳化方法第一步所需之  $N$  個設定值解。
- (5) 完成序的最佳化方法並求得所解的隨機最佳化問題的一個不錯的解。
- (6) 與實際測試所得資料做驗證，並做改善情形比較。

為了克服隨機模擬最佳化問題的計算花費大量時間，我們提出了一個序的最佳化方法使用一個合理的計算時間來求解一個足夠好的解。在使操作系統的期望成本最小的考量下，展示了我們的方法與其他現存的服務策略，發現我們的方法表現最好。而在計算效率與得到解的品質上，也展示了我們的方法可以達到即時應用。

### 致謝

本論文承蒙行政院國家科學委員會補助研究經費，計畫編號 NSC 98-2221-E-324-039。特此致謝。

### 參考文獻

- [1] H. Takagi, "Queuing Analysis of polling model," ACM computing surveys, Vol. 20, No.1, pp. 5-28, Mar., 1988.
- [2] H. Levy, and M. Sidi, "Polling systems: applications, modeling, and optimization," IEEE Transactions on Communications, Vol. 38, No. 10, pp.1750-1760, Oct., 1990.
- [3] D. Gupta, and Y. Gunalay, "Recent advances in the analysis of polling systems," in: Advances in Combinatorial Methods and Applications to Probability and Statistics, N. Balakrishnan (editor), pp.339-360, Birkhauser Boston, 1997.
- [4] I. Eliazar, "Gated Polling Systems with Levy Inflow and Inter-Dependent Switchover Times: A Dynamical-Systems Approach," Queueing Systems, Vol. 49, No. 1., pp. 49-72, Jan., 2005.
- [5] K. K. Leung, "Cyclic-service systems with nonpreemptive, time-limited service," IEEE Transactions on Communications, Vol. 42, No. 8, pp. 2521-2524, Aug., 1994.
- [6] H. Takagi, "Analysis and application of polling model," in : Performance Evaluation: Origins and Directions, eds. G. Haring, C. Lindemann and M. Reiser, Lecture Notes in Computer Science, Vol. 1769, pp. 423-442, Springer, Berlin, 2000.
- [7] O.J. Boxma, "Analysis and optimization of polling systems," in: Queueing, Performance and Control in ATM (ITC-13), J. W. Cohen and C. D. Pack (editors), pp.173-183, Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1991.
- [8] F. Azadivar, "Simulation optimization methodologies," Proceedings of the 1999 Winter Simulation Conference, Phoenix, AZ, pp. 93-100, 1999.
- [9] J.R. Swisher, P.D. Hyden, S.H. Jacobson, and L.W. Schruben, "A survey of simulation optimization techniques and procedures," Proceedings of the 2000 Winter Simulation Conference, Orlando, FL, pp. 119-128, 2000.
- [10] S.M. Robinson, "Analysis of sample-path optimisation," Mathematics of Operations Research, Vol. 21, No. 3, pp. 513-528, 1996.
- [11] A.G. Greenwood, L.P. Rees, and F.C. Siochi, "An investigation of the behavior of simulation response surfaces," European Journal of Operational Research, Vol. 110, pp. 282-313, 1998.
- [12] Y. Carson, and A. Maria, "Simulation optimization: methods and applications," Proceedings of the 1997 Winter Simulation Conference, Atlanta, GA, pp. 118-126, 1997.
- [13] R.L. Haupt, and S.E. Haupt, Practical genetic algorithms, 2nd edition, Hoboken, NJ:John Wiley, 2004.
- [14] M. Locatelli, "Simulated annealing algorithms for continuous global optimization: convergence conditions," Journal of Optimization Theory and Applications, Vol. 104, No. 1, pp. 121-133, 2000.
- [15] D.T. Pham, and D. Karaboga, Intelligent optimisation techniques, London, Springer Verlag, 2000.
- [16] O. Hajji, S. Brisset, and P. Brochet, "Comparing stochastic optimization methods used in electrical engineering," 2002 IEEE International Conference on Systems, Man and Cybernetics, Vol. 7, Hammamet, Tunisia, 2002.

- [17] F. Glover, J.P. Kelly, and M. Laguna, "New advances and applications of combining simulation and optimisation," Proceedings of the 1996 Winter Simulation Conference, Coronado, CA, pp. 144-152, 1996.
- [18] H. Pierreval, and J.-L. Paris, "Distributed evolutionary algorithms for simulation optimisation," IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol. 30, No. 1, pp. 15-24, 2000.
- [19] T.W.E. Lau, and Y.C. Ho, "Universal alignment probability and subset selection for ordinal optimisation," Journal of Optimization Theory and Applications, Vol. 39, No. 3, pp. 455-489, 1997.
- [20] Y.C. Ho, "An explanation of ordinal optimization: Soft computing for hard problems," Information Sciences, Vol. 113, No. 3-4, pp. 169-192, 1999.
- [21] C.-H. Chen, S.D. Wu, and L. Dai, "Ordinal comparison of heuristic algorithms using stochastic optimisation," IEEE Transactions on Robotics and Automation, Vol. 15, No. 1, pp. 44-56, 1999.
- [22] S.-Y. Lin, and Y.C. Ho, "Universal alignment probability revisited," Journal of Optimization Theory and Applications, Vol. 113, No. 2, pp. 399-407, 2002.
- [23] S.-Y. Lin, Y.C. Ho, and C.-H. Lin, "An ordinal optimization theory based algorithm for solving the optimal power flow problem with discrete control variables," IEEE Transactions on Power System, Vol. 19, No. 1, pp. 276-286, 2004.
- [24] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, Vol. 2, No. 5, pp. 359-366, 1989.
- [25] M.F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, Vol. 6, No. 4, pp. 525-533, 1993.