# 逢　甲　大　學

## 資訊工程學系專題報告

# 使用智慧型基因演算法設計隨選視訊系統的最佳影片配置

**Using Intelligent Genetic Algorithms to Design Optimal Video Replication and Placement of Video-on-Demand Systems**


學　　生：　陳彥百（四甲）
　　林凱遠（四甲）
　　黃炯明（四甲）

指導教授：何信瑩


中華民國九十三年十二月

# 摘要

隨著網路、硬碟及多媒體視訊技術的進步，隨選視訊系統的發展日益成形。隨選視訊系統通常需儲存大量的影片視訊，以提供客戶各種不同的需求。過去的研究顯示，影片視訊的複本配置問題不僅決定了儲存成本，同時也密切地影響了隨選視訊系統的效能 。在本專題中，我們將影片配置問題轉換成隨選系統最佳化問題，並分別使用單目標及多目標方式來進行最佳化。在單目標最佳化中，我們將所考慮的最佳化目標：(1)影片複本與(2)負載平衡度，透過適當的權重值，結合成單一最佳化目標 ，並搭配使用智慧型基因演算法尋求出最佳解。在多目標最佳化中 ，我們則同時最佳化以下兩個目標 ：(1)影片視訊的總儲存容量及 (2)系統整體的服務阻隔率 。我們利用智慧型多目標演化式演算法之有效處理具大量參數之多目標問題的能力，尋求出隨選視訊系統之最佳的影片視訊配置，並提供一組 Pareto 解集合，使在建置隨選視訊系統時可有更多選擇方案。從各種實驗顯示，我們所提之方法無論在單目標或是多目標的考量下，均能有效地改善系統效能，並大幅地減少儲存成本。

**關鍵字** – 隨選視訊系統、複本配置、智慧型多目標演化式演算法、Pareto 解集合

# 目 錄

# 圖 表 目 錄

# 第一章 導論

## 1.1 隨選視訊系統介紹

近年來，隨著網路、硬碟及多媒體視訊技術的不斷進步 ，隨選視訊(Video-on-Demand, VOD)系統的發展逐漸成形，其中較明顯的例子就是日前國內電信業者-中華電信-所提供的 MOD 服務。透過隨選視訊系統，使用者可以在任何時間經由網路，點選他們有興趣的多媒體視訊，如電影、電視頻道…等。且更進一步地進行飛梭功能的操作，如暫停、快轉等。使得使用者能完全地掌握播放時間與內容 ，大幅地改變了過去的收看模式，因而提升了許多的彈性及便利性。

概括來說，根據隨選視訊系統所提供的服務 類型，大致可分成下列主要三種 [1]：

(1) Unicast Scheme：每個影片視訊需求 (request)都提供一條服務的串流。由於這樣的視訊串流專門服務單一個客戶 ，因此，互動式的飛梭功能可以容易地被現實 。然而，這樣的系統較不容易被擴充 。因此，同一時間可以提供的服務人數有限。

(2) Multicast Scheme：對於同一部影片的數個要求會被先收集起來，然

後再以一條視訊串流提供服務 。這樣的方式 ，可以有效地減少系統頻寬的負載並改善系統的服務阻隔率[2]。例如：Batching [3]及 Patching [4]。我們將在下一節中介紹批次(batching)隨選視訊系統。

(3) Broadcast Scheme：影片視訊會預先決定好播放排程並藉由一個專門的頻道傳送到客戶端 。此方式的好處在於 ，對於那些很熱門的影片 ，可以提供無上限的服務人數 ；然而，對於較不熱門的影片 ，則會有頻寬浪費的問題。此外，客戶需等待至欲收看之影片視訊所排定的時間達到才能收看。

在伺服器的部分，又可分成兩種不同的架構：(1)集中式(centralized)與(2)分散式(distributed)。集中式的伺服器即為只使用單一台電腦來服務所有的客戶需求，因此，對於電腦的設備要求較高，如硬碟的 I/O 能力、網路頻寬…等。通常會藉由 RAID 系統實作[5]，具有易於建置及管理、低成本等優點。然而，當磁碟個數增多時，這種架構容易發生磁碟存取競爭情形以及錯誤率提升 ，因此限制了這種架構的擴充性（scalability）及可信度（reliability）。而在分散式的架構中，由於各個伺服器內部具有自己的儲存系統，並藉由骨幹網路（backbone network）將這些伺服器連接在一起 ，因此，有較好的擴充性及可信度 。同時，

這種架構也有較大的串流容量（streaming capacity），故可以同時服務較多的用戶。

隨選視訊系統通常儲存大量的影片視訊以因應不同客戶的各種需求。根據被客戶點播的次數，這些影片視訊會有不同的熱門程度(popularity)。如果某部較熱門的影片視訊只有一個複本，則在同一時間中可能會湧進過多需求，並集中於儲存該部影片視訊的伺服器上，這可能會使該視訊伺服器負載過重，並造成整體系統的負載不平衡，進而降低系統效能。因此，在多伺服器（multi-server）的系統中，會將較熱門的影片視訊複製多個複本於若干視訊伺服器中，來分散視訊需求以平衡系統整體的負載。

## 1.2 研究動機與目標

隨選視訊系統通常被要求儲存大量且多樣性的多媒體視訊內容，以因應各種使用者的需求。在一個多伺服器的隨選視訊系統中，如何妥善地儲存這些影片視訊於系統中，即如何決定各影片的複本數及相對的配置位置，密切地影響了系統效能。明顯地，增加影片的複本數可提高其可得性而降低系統服務阻隔率(blocking probability)，但卻增加了儲存成本。然而，決定每部影片的複本數及其相對地配置進而最佳

化整體系統的效能是相當複雜的，研究指出[6]，其複雜度隨著影片及伺服器數量增加而呈指數成長。換言之，我們需要一個有效率的方式來處理影片視訊的複本配置問題。目前將視訊系統配置到伺服器的工作是由具經驗的人員來進行，故很難獲得最好的效能。

因此，我們的研究目標在於，考慮一個多伺服器的隨選視訊系統（分散式架構)，決定其最佳的影片配置方式-決定每部影片的複本數及相對的配置位置，使得在所設定的最佳化目標（objective）下，能獲得最好的效果。在處理最佳化目標時，我們有兩個作法：（1）為所考慮的最佳化目標設定適當的權重然後相加起來，成為一個單一的最佳化目標。再對此目標進行單目標最佳化。為此，我們利用由指導老師何信瑩教授所提之智慧型演算法(intelligent genetic algorithm, IGA) [7]為我們搜尋單目標最佳解。（2）讓各個最佳化目標獨立存在，並同時最佳化這些目標。我們使用也是由何教授所提之智慧型多目標演化式演算法 (intelligent multi-objective genetic algorithm, IMOGA) [8]來處理多目標最佳化問題。由於 IGA 及 IMOGA 均使用了智慧型基因蒐集運算子（intelligent gene collector, IGC） 可合理地規劃實驗並系統化推理出較好的組合，進而有效率地找出最佳解。尤其在解決具大量參數之最

佳化問題上，更有良好的表現。因此，相當適合用來解決隨選視訊系統的影片配置問題。

## 1.3 研究流程

我們的研究流程如圖 1.1 所示。以下將針對各個步驟作出說明：

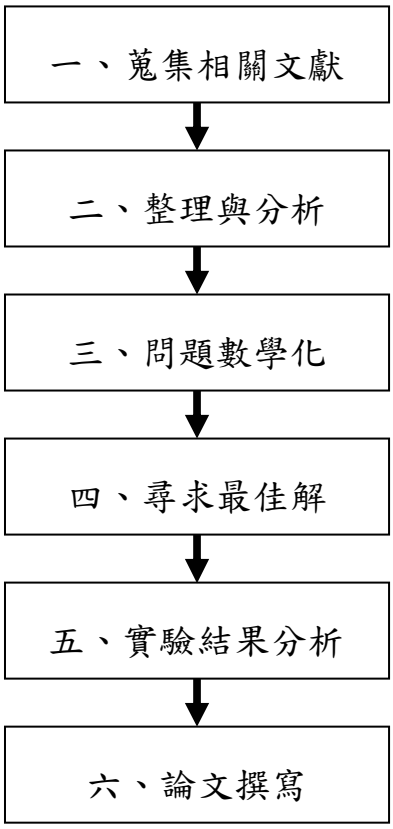



圖 1.1 研究流程圖

### 一、 蒐集相關文獻

一開始我們利用網路上所提供之電子期刊搜尋引擎 ，如 ACM、

IEEEXplore 等，針對隨選視訊系統的議題，搜尋相關的文獻。並詳細研讀各文獻所探討的範疇和所關心的問題 ，以及提出的解決方法 。同時，我們也與指導老師及學長討論研究的發展方向 ，將較不直接相關的議題篩選掉，並進一步地確立研究所探討的領域和範疇。

**二、 整理與分析**

在這個階段，我們將與隨選視訊系統直接相關的文獻作更深入的整理與分析，將他們所專注的問題和 提出的解決方法考慮哪些目標 、有哪些限制及特點作適當而明確地歸類 。並與老師討論後，找出適當的切入點，明確地定義出我們所專注的問題。

**三、 問題數學化**

我們將所專注的問題 ，透過數學的方式描述出來 ，明確定義 在數學模型中使用的參數與變量 。並更進一步地定義出 所研究問題之目標函式(objective function)，以搭配之後的智慧型基因演算法之用。

**四、 尋求最佳解**

在專題的初期 ，我們考慮複本配置問題為一個大量參數的最佳化問題，將所考慮之最佳化目標透 過權重加總的方式結合成為單一目

標，並利用智慧型基因演算化來尋求最佳解 。這個方法在實作後獲得相當良好的效果，我們已將之整理成學術文獻(詳見附錄)。

隨後，我們對複本配置的問題作更深入的研究 ，並考慮其為具大量參數之多目標最佳化問題，並修改所參考的[6]所提出的混合式基因演算法，以智慧型多目標基因演算化來取代原來的傳統基因演算法 ，以有效處理大量的參數(影片視訊)，並提供精確的 Pareto 解集合以提高系統設置(configuration)時的彈性。

## 五、 實驗結果分析

為了驗證所提之方式的可行性及效能 ，我們將設計適當的實驗與原有方法作比較，分析實驗數據所反應的現象並進一步地討論這樣的現象是否合理及是否還有可改進的空間。

## 六、 論文撰寫

最後，我們將研究每個步驟的過程與結果詳細地整理成可用的材料，以供進一步將之撰寫成學術文獻 。在這個階段，我們與老師討論和規畫論文架構，並不斷修正與釐清所研究問題的主軸以求能 簡潔、明確及忠實地呈現出研究成果。

## 1.4 文章架構

報告的剩餘的部分組織如下：在第二章中，我們將介紹以單目標的方式來最佳化隨選視訊系統時，所考慮的系統模型及數學描述。第三章則為隨選視訊系統之多目標最佳化時所考慮的系統模型及相對應的數學描述。第四章將介紹我們所採用的 IGA 及 IMOGA 的原理，包括直交實驗設計(orthogonal experiment design, OED)、GPSIFF (generalized Pareto-based scale-independent fitness function)及智慧型基因採集器(intelligent gene collector, IGC)。第五章則說明在分別考量配置問題為單/多目標的情形下，如何應用 IGA 及 IMOGA 來解決所研究之隨選視訊系統影片複本配置問題。第六章為實驗部分，包括了實驗環境的設定及實驗結果分析，以驗證本專題研究之可行性。第七章為結論部分，主要對專題成果作出總結以及學生參與專題的心得感想。

# 第二章 隨選視訊系統之單目標最佳化

## 2.1 系統模型

我們考慮一個具有 $N$ 台同質伺服器的隨選視訊系統，即每台伺服器具有相同的儲存容量 $C$ 及相同的對外網路頻寬 $B$。且系統中有 $M$ 部不同的影片，但影片的長度皆為 $d$（如 $d=90$）且相同編碼格式。換言之，每部影片所需的儲存空間相同。

如前所述，每部影片根據其被點播的次數，有其對應的熱門程度。在此研究中，我們假設影片的熱門程度在進行配置之前即為已知的資訊。且我們利用 Zipf 分布[9]來描述每部影片的點播機率。對於第 $i$ 部影片而言，其對應的點播機率為 $p_i = i^{-\delta} \Big/ \sum_{i=1}^{M} j^{-\delta}$。其中，$\delta$ 為扭曲參數（skew parameter），用來控制分布的情形。一般而言，$0.271 \le \delta \le 1$。

此外，我們僅考慮在尖峰時段中的隨選視訊系統。在尖峰時段中，負載平衡對於改善整體輸貫量及服務可得性而言是相當重要的。同時，我們考慮對外頻寬為效能主要的瓶頸[10]。因此，我們所考慮的最佳化目標即為提高影片複本個數以增進可得性及 平衡各伺服器的負載以改善系統輸貫量。

## 2.2 數學描述

令$L$代表系統負載不平衡的呈度，$r_i$代表影片$v_i$的複本個數。我們希望利用智慧型基因演算法為我們搜尋出一組最好的配置情形，即最佳解$X$：

$$X = [b_{11},\cdots,b_{1N},b_{21},\cdots,b_{2N},\cdots,b_{MN}]^T \quad (1)$$

其中，$b_{ij}$表示影片$j$是否存在於伺服器$i$上。換句話說，若第$i$台伺服器上存有第$j$部影片複本，則$bij=1$；否則，$b_{ij}=0$。故$X$為一組長度為$M \cdot N$的二元字串。則我們的最佳化目標如下：

$$\max obj(X) = \sum_{i=1}^{M} r_i / M - \alpha L \quad (2)$$

其中，$\alpha$是一個適當的權重。對於負載不平衡的呈度，目前有多種定義[11]，我們選擇其中一種如下：

$$L = \max_{\forall s_i \in s} \left| l_i - \bar{l} \right| \quad (3)$$

其中，$\bar{l} = \sum_{i=1}^{N} l_i / N$。

此最佳化目標遭遇到以下三個限制：（1）伺服器的儲存容量限制，（2）伺服器對外的網路頻寬及（3）每台伺服器不得存放相同影片的

複本。令$\pi(v_i^j)$表示存放影片$v_i$的第$j$個複本的伺服器且$\pi(v_i)=k$表示伺服器$s_k$中儲存了影片$v_i$的一個複本。對於影片$v_i$的每個複本之通訊權重（communication weight）可定義為$w_i = p_i / r_i$。透過靜態輪流的排程策略（round robin），在尖峰時段影片$v_i$的每個複本所提供的服務次數為$w_i \cdot \overline{\lambda} \cdot d$。令$l_k$為伺服器$s_k$的對外通訊負載。則我們所考慮的限制可如下表示：

$$\sum_{\pi(v_i)=k, \forall v_i \in V} b_i \cdot d \le C \qquad (4)$$

且

$$l_k = \sum_{\pi(v_i)=k, \forall v_i \in V} w_i \cdot \overline{\lambda} \cdot d \cdot b_i \le B \qquad (5)$$

為了符合（3）之限制，我們必需將影片$v_i$的$r_i$個複本分別存放在$r_i$個不同伺服器上：

$$\pi(v_i^{j_1}) \neq \pi(v_i^{j_2}) \quad 1 \le j_1, j_2 \le r_i, j_1 \neq j_2 \qquad (6)$$

且

$$1 \le r_i \le N \quad \forall v_i \in V \qquad (7)$$

# 第三章 隨選視訊系統之多目標最佳化

## 3.1 系統模型

在考慮多目標最佳化的情形下，我們把焦點放在批次隨選視訊系統上。以系統資源的使用及控制的觀點，批次隨選視訊系統通常被認為是一個較有效的機制。因此，在我們的專題研究中，我們考慮的隨選視訊系統為影片視訊的需求會先被停駐一段時間，在這一段時間中，系統可以聚集更多對該部影片視訊的需求。然後，系統再以一條視訊串流來服務這些視訊需求。

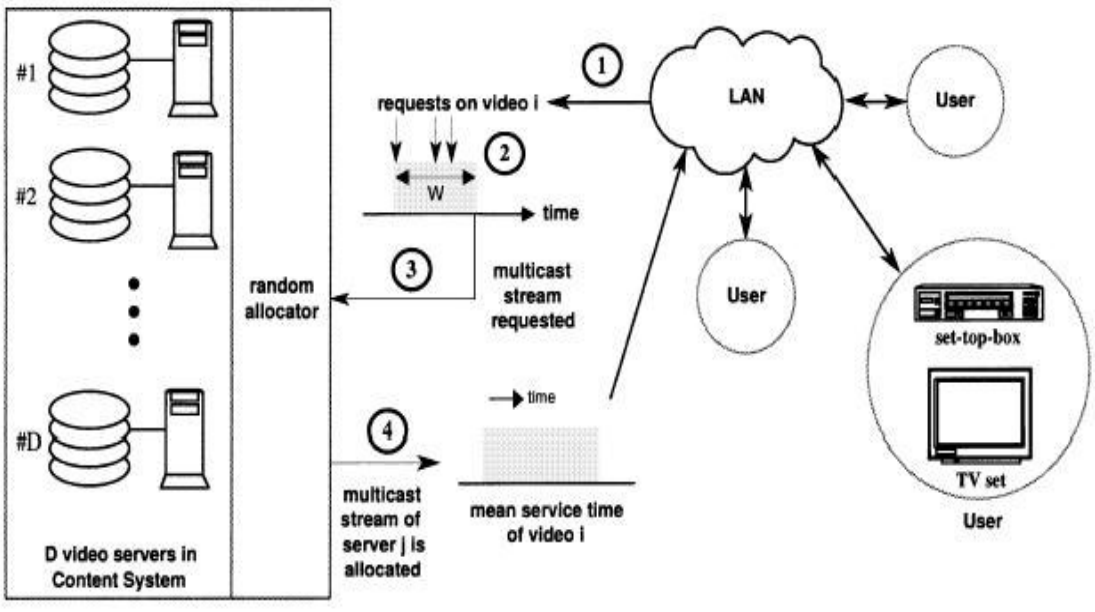



圖 3.1 批次隨選系統架構

我們利用圖三來解說批次隨選視訊系統的運作[6]。圖中左邊的部分為一個由 $D$ 部視訊伺服器所構成的中央儲存系統 (centralized content system)並搭配一個隨機指派器(random allocator)，會隨機指派一部視訊伺服器來提供服務。批次隨選視訊系統的運作方式如下：

1. 當對某部影片的第一個視訊需求到達，一個批次時間(batching interval)為 $W$ 的批次窗(batching window)將被啟動。
2. 在該影片的批次窗內，對於該部影片的需求將會被收集起來。
3. 當批次時間結束，系統會傳送該部影片的串流要求至中央儲存系統。
4. 隨機指派器會從中央儲存系統隨機地指定一台存有該部影片的視訊伺服器，並開啟一條串流，為等候在批次窗中的客戶服務。若指定的伺服器所開啟的串流以達上限，則服務要求將被阻隔。

為了設置隨選視訊系統中的各視訊伺服器 ，我們必需決定每部影片的複本個數以及要這些複本儲存於哪些視訊伺服器上 。然而，在考慮到成本與效能的取捨，我們必需同時最佳化以下兩個目標：(1)最小化整體的儲存容量與(2)最小化整體的服務阻隔率。此外，我們也需考量每個視訊伺服器皆有其儲存上限 ，因此在進行配置時不能超過這些限制。同時，我們也假設所提供之一個視訊串流能服務無限或是足夠

多的客戶。而所有的客戶在批次時間中，都願意等待而不會放棄收看。

## 3.2 數學描述

考慮一個有 $D$ 台視訊伺服器的批次隨選視訊系統並提供 $M$ 部不同的影片視訊[1]。令 $p_i$ 為影片視訊 $i$ 的熱門程度，並假設客戶向系統發出影片需求的過程為帕松過程，並令其到達率為 $\lambda$。則對於影片視訊 $i$ 的有效需求率可計算如下：

$$\lambda_i^B = \frac{\lambda p_i}{1+\lambda p_i W} \tag{8}$$

其中 $W$ 為其批次時間而 $(1+\lambda p_i W)$ 為在批次中的平均要求數。而系統的有效要求率可以如下計算：

$$\lambda_e = \sum_{i=1}^{M} \lambda_i^B = \sum_{i=1}^{M} \frac{\lambda p_i}{1+\lambda p_i W} \tag{9}$$

對於影片視訊 $i$ 而言，被要求提供視訊串流的機率為

$$\alpha_i = \frac{\lambda_i^B}{\lambda_e} \tag{10}$$

而系統的平均服務時間為

$$\frac{1}{\mu_e} = \sum_{i=1}^{M} \frac{\alpha_i}{\mu_i} = \frac{1}{\lambda_e} \sum_{i=1}^{M} \frac{\lambda_i^B}{\mu_i} \tag{11}$$

其中 $1/\mu_i$ 為影片視訊 $i$ 的平均服務時間。則進入隨選視訊系統的有效流量 $A_e$ 可被計算成

$$A_e = \frac{\lambda_e}{\mu_e} = \sum_{i=1}^{M} \frac{\lambda p_i}{(1+\lambda p_i W)\mu_i} \tag{12}$$

此外，我們假設在批次窗中的視訊需求程指數分佈 [1]。且有效流量 $A_e$ 中 $q_j$ 的部分分派由視訊伺服器 $j$ 處理，則視訊伺服器 $j$ 的服務阻隔率可利用 Erlang B Formula [12]求得：

$$B_{q_j} = \frac{(A_e q_j)^{L_j}/L_j!}{\sum_{i=0}^{L_j}(A_e q_j)^i/i!} \tag{13}$$

其中 $L_j$ 為視訊伺服器 $j$ 可提供之最大視訊串流數，且 $q_j \geq 0$。

整體系統之服務阻隔率可由下推導：

$$B = \sum_{j=1}^{D} q_j B_{q_j} = \sum_{j=1}^{D} q_j \frac{(A_e q_j)^{L_j}/L_j!}{\sum_{i=1}^{L_j}(A_e q_j)^i/i!} \tag{14}$$

其中 $\sum_{j=1}^{D} q_j = 1$。整體系統的儲存容量為

$$C = \sum_{i=1}^{M} v_i \cdot n_i \tag{15}$$

其中 $v_i$ 為影片視訊 $i$ 容量大小，而 $n_i$ 為影片視訊 $i$ 的複本個數。

因此，我們所研究之多目標最佳化問題可描述如下：

$$\min F(X)=\begin{cases} f_1(X)=B=\sum_{j=1}^{D} q_j \cdot B_{q_j} \\ f_2(X)=C=\sum_{i=1}^{M} v_i \cdot n_i \end{cases} \qquad (16)$$

其中，$X=[n_1,...,n_M]^T$ 為欲最佳化之參數。

在本專題中，我們利用 Tang [1][6]所提之 HLF 演算法來求得(14)中的系統服務阻隔率 $B$。其中，HLF 演算法請詳見[1][6]。

# 第四章 智慧型演化式演算法

智慧型基因演算法 (IGA)與智慧型多目標演化式演算法 (IMOEA)皆使用智慧型基因蒐集運算子(IGC)來進行基因交配，我們將先在 4.1 節介紹 IGC，4.2 節介紹 IGA，4.3 節介紹 IMOEA。

## 4.1 智慧型基因蒐集運算子 IGC

傳統基因演算法的交配(crossover)運算是由兩個父代染色體，隨機選擇切點然後組合產生兩個新的子代染色體 。IGC 的優點在於把具有系統化推理能力的直交實驗設計 (Orthogonal Experimental Design, OED)[13]合併於交配運算之中 ，使能個別評估染色體的每個基因對於適應函數值的貢獻。並且智慧地選出較好的基因形成子代染色體。

### 4.1.1 直交表與因素分析

直交實驗設計為一種以直交表 (Orthogonal Array, OA)與因素分析(factor analysis)為基礎的品質控管方法。直交表是由 R. A. Fisher 最先提出的。「直交」所代表的意義是平衡(balance)與不混合(not mix)，即所謂統計上的獨立(statistically independence)。應用直交表設計實驗，分析結果可以獨立且均勻的求出每一個因素的主效果（main effect）[14,

15]，並由主效果推論每一個因素對於該實驗結果的影響好壞。因此藉由直交表系統推理化的特性只需進行部份因素實驗就可以推側出最佳的近似解(near optimum)。也就是說使用直交表設計實驗，僅需要進行部份因素實驗(fractional-factorial experiment)，就可以推論全實驗的結果，因此較完全因素實驗 (full-factorial experiment)節省大量執行的時間。

以三個因素而每個因素都有兩個水準值的例子來說 ，若要進行完全因素實驗，應執行 8 (即 $2^3$ ) 個實驗，如圖 4.1 所示。如果現在只能執行 8 個實驗中的 4 個實驗，那麼該如何選擇其中的 4 個實驗呢？傳統的單因素實驗方法 ，在一次只改變一個因素的方式下 ，則會選擇圖 4.1 中實驗編號為 1 、2 、4 與 8 的實驗點進行實驗。但是點 1 、2 、4 與 8 並不均衡，也就是說，在六面體的每一面所選取之實驗點數不盡相同。如果選擇的實驗點是 1 、7 、6 與 4 (或是 2 、3 、5 與 8)，如圖 4.2 所示的黑點，則六面體的每一面都有兩個實驗點 ，而且都是對稱的，此即為直交表均勻取樣的特性 。而且最佳解在其所包圍的立方體之中 ，因此可以藉由均勻取樣的 實驗，來推測全試驗最佳值 。同理可推，128 (即 $2^7$ )個實驗的部份實驗，就是從 128 個實驗中選擇 8 個

實驗來執行。因此，在 $n$ 個因素，每個因素都有兩個水準值的實驗中挑出 $n$+1 個來執行即稱為部份因素設計。

| 實驗編號 | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 1 | -1 | -1 | -1 |
| 2 | +1 | -1 | -1 |
| 3 | -1 | +1 | -1 |
| 4 | +1 | +1 | -1 |
| 5 | -1 | -1 | +1 |
| 6 | +1 | -1 | +1 |
| 7 | -1 | +1 | +1 |
| 8 | +1 | +1 | +1 |

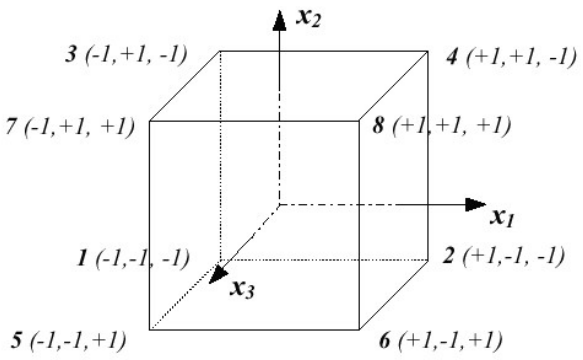



圖 4.1 三因素二水準值完全因素實驗表與空間分佈圖

| 實驗編號 | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| 1 | -1 | -1 | -1 |
| 7 | -1 | +1 | +1 |
| 6 | +1 | -1 | +1 |
| 4 | +1 | +1 | -1 |

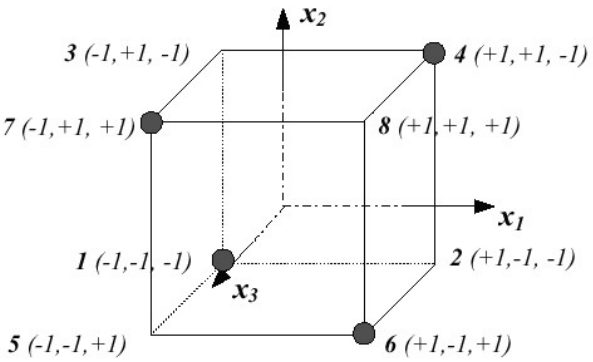



圖 4.2 三因素二水準值部份因素實驗表與空間分佈圖

以上簡要的說明了直交表的原理，接著要以直交表 $L_8(2^7)$ (如表 4.1) 為例說明直交表產生的方法。

表 4.1 直交表 $L_8(2^7)$

| | Factors | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Exp. no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $y_t$ |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $y_1$ |
| 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | $y_2$ |
| 3 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | $y_3$ |
| 4 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | $y_4$ |
| 5 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | $y_5$ |
| 6 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | $y_6$ |
| 7 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | $y_7$ |
| 8 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | $y_8$ |
| $S_{j1}$ | $S_{11}$ | $S_{21}$ | $S_{31}$ | $S_{41}$ | $S_{51}$ | $S_{61}$ | $S_{71}$ | |
| $S_{j2}$ | $S_{21}$ | $S_{22}$ | $S_{32}$ | $S_{42}$ | $S_{52}$ | $S_{62}$ | $S_{72}$ | |

直交表 $L_8(2^7)$的產生程序如下：

步驟 1 ：將 a 因素放入第 1 行位置。其中 a 因素的水準值 1 和水準值 2 連續出現的次數為 $8/2^1=4$ 。即連續出現 4 個水準值 1 ，再連續出現 4 個水準值 2 。

步驟 2 ：將 b 因素放入第 2 行位置，其中 b 因素的水準值 1 與水準值 2 連續出現的次數為 $8/2^2=2$ 。

步驟 3 ：第 3 行的成份為 a × b 。即當 a = b = 1 或 a = b = 2 時 a × b = 1 ，否則 a × b = 2 。

步驟 4 ：將 c 因素放入第 4 行位置，其中 c 因素水準值 1 和水準值 2 連續出現的次數為 $8 / 2^3 = 1$ 。

步驟 5 ：第 5 、6 與 7 行依步驟 3 類推。

由上面所述產生的方式可知直交表考慮了因素之間的交互作用 ，如表 4.1 中的第 3 行即用來表示因素 a 與 b 間的交互作用。但是在多參數最佳化選擇問題中 ，參數間的交互作用很難決定 ，同時直交表實驗如果考慮因素間的交互作用 ，會使直交表可用的參數減少 。例如在表 4.1 中如果考慮因素 a 與 b 間的交互作用，那麼原本可使用來對應 7 個參數的 $L_8(2^7)$直交表，就必須扣除掉第 3 行的對應，而只能解 6 個參數的問題。為了充分運用直交表的每個因素 ，我們所提出的直交表篩選機制在應用上必須使參數間的交互作用較小 ，如此一來才可以利用直交表系統化推理能力 ，分析出參數的效果 ，也稱作主效果評估 。而最終選擇出最佳解的實驗的參數組合 。

什麼是主效果？在表 4.1 中，令 $y_t$ 表示為 $L_8(2^7)$直交表實驗中第

$t$ 次實驗的評估函數值，則第 $j$ 個因素水準值 $k$ 的主效果 $S_{jk}$ 定義為

$$S_{jk} = \sum_{t=1}^{\beta} y_t \cdot F_k \quad , \tag{17}$$

其中 $F_k$ 為一個旗標值，若第 $t$ 次實驗中第 $j$ 個因素選用水準為 $k$，則 $F_k$ 為 1；若否，則 $F_k$ 為 0。若適應函數為望大，則較大的主效果值表示對適應函數具有較佳的貢獻度；反之若適應函數望小，則主效果值小者貢獻度較佳。

主效果可以顯示因素中水準的各別影響。例如主效果 $S_{j1} > S_{j2}$ 則表示在參數最佳化的問題中，第 $j$ 個因素水準值 1 對於整體最佳化函數的貢獻大於水準值 2。如果相反的情形 $S_{j1} < S_{j2}$，則表示水準值 2 較佳，主效果的分析可以用來推測出全實驗可能的最佳解。

直交因素實驗為一種部分因素實驗方式，可以有效減少參數設計時的實驗次數，並同時考量實驗因素之間的交互作用。將直交因素實驗後的數據經過主效果分析，便可以將每個因素對於設計目標的貢獻優劣計算出來，推論出最佳解的實驗的參數。

### 4.1.2 IGC 流程

如何使用 OED 達成 IGC，其詳細步驟如下所示：

步驟一：將染色體切割成 N 個基因區段(gene segment)，以直交表中的一個因素表示為一個基因區段，則直交表的大小即有 N 個因素。直交表的大小即 $Ln(2n^{-1})$，$n = 2^{\lceil \log(N+1) \rceil}$，如此便有 n 次的評估實驗運算。

步驟二：直交表中因素 j 的水準為 1 或 2，表示第 $j_{th}$ 的基因區段，由第一個父代或第二個父代遺傳得來。

步驟三：在此步驟中，單目標與多目標的執行有所不同。單目標只需計算每個實驗的目標函數值並當成其適應值。而多目標需先計算每個染色體對於所有目標函數的值，然後以 GPSIFF 評估函數計算所有實驗的適應值 $y_t$，其中 t = 1, 2, …, n。

步驟四：計算所有因素的主效果，$S_{jk}$，其中 j = 1, 2, …, N，k = 1, 2。

步驟五：根據主效果，來決定基因參數水準的選擇，以決定子代染色體的基因組合。若第 jth 的參數中，主效果 Sj1 > Sj2，則選擇水準 1，即第一個父代的第 jth 的基因區段；若主效果 Sj1 < Sj2，則選擇水準 2，即第二個父代的第 jth 的基因區段。

步驟六：由不同染色體中較佳的部分組合產生最佳基因區段組合的子代染色體。

步驟七：計算不同染色體相同因子間的主效果差(Main Effect Difference, MED)，並依主效果差來將參數作排名。其中最小的最大主效果差擁有最高的排名。

步驟八：產生次佳子代染色體的基因區段組合，將步驟六所產生的最佳子代染色體中 MED 記錄排名最高的基因區段位置，改變成由另一個水準所對應的基因區段來組合，即可產生次佳基因區段組合的子代染色體。

## 4.2 智慧型基因演算法 IGA

### 4.2.1 基因演算法 GA

基因演算法 (Generic Algorithm，GA) 是由美國密西根大學的 J.H.Holland 教授於 1975 年所提出的。基因演算法是取自於大自然的一種演算法，其基本精神在於仿效生物界中物競天擇 ，不適者優勝劣敗的自然進化法則，以尋求出問題的最佳解。根據達爾文所題出的”適者生存”觀念，對環境適應度較高的品種其生存的機會越高，而達成這種演化的關鍵在於基因的複製 (Reproduction)、交配 (Crossover)和突變 (Mutation)三種特徵。基於仿效此種演化的法則，基因演算法透過編碼

技術將所有的可能解都轉換成染色體，再依據求解的條件來設計適應函數(Fitness Function)。因此，大量的基因經由複製、交配和突變等演算過程，不斷地產生出新的基因，且淘汰掉不良不具優勢的基因，經歷數代的篩選淘汰，最後演化出一組讓適應函數達到最佳化的解。

基因演算法在最佳化的過程中扮演最重要的關鍵為，染色體的編碼方式與其對應的適應度評估函數。選擇適當的編碼方式，可以大大地提升計算的效率，而評估函數的目的是對一組經過編碼的可能解加以評分，而評分的依據乃是根據所處理的問題所訂定之。基因演算法的流程如圖一所示，底下則對基因演算法之各種運算做進一步的解說。

1. 複製運算：在此可引用達爾文的”物競天擇說”解釋世上各生物可繁延至今，皆因不斷地演化進步而達成的。在基因演算法中，複製是依據每一物種的適應程度來決定其在下一個子代中應被淘汰或複製的個體多寡的一種運算過程。保留適應度較高的染色體，並在下一代中被大量地複製，而適應度較差的染色體則在下一代中被淘汰掉，其中適應程式的量測是由適應函數來反應的。
2. 交配運算：基因演算法使用大量的可能解在解空間中平行搜

尋，交配運算的主要功能為能夠在平行搜尋的過程中彼此交換資訊。在染色體交配的過程中，父代透過交換彼此的資訊，期望能產生出新一代的染色體的適應度更加優秀 。

3. 突變運算：突變運算在基因演算法中所扮演的角色，為帶領整體的演化跳躍至不同的值域中重新做搜尋，以免陷入區域的最佳解中。因此，加上突變運算後，便可確保基因演算法演化的過程中，空間中的所有可能解都有被搜尋到的機會。在基因演算法的設定參數上 ，突變運算發生的機率通常都設得相當地低，此小小的突變對於單一個體往往會造成傷害，但對於整個族群的演化上，卻是一種非常重要的助益。

### 4.2.2 IGA

智慧型基因演算法(Intelligence Generic Algorithm，IGA)[7]是由何信瑩博士所發展出的 ，其主要的特色融合了基因演算法與直交表實驗這兩種截然不同的最佳化機制 ，擷取了這兩種演算法的優點並補足了對方的缺點。基因演算法能對定義的解空間進行大量的搜尋 ，但在解空間太大及參數過多的情況下 ，將會使得基因演算法過早收斂至一組的最佳解中 。而直交表在實驗前必須由人為決定各因素在不同水準中

所對應的數值，且在大多數的問題中，各因素間有交互作用，使得直交表無法在一次的計算過程中找到全域的最佳解。智慧型基因演算法利用了基因演算法中的演化觀念及編碼方式，補足了直交表實驗無法自動產生各因素對應值的缺點，而直交表實驗的系統化推理，可大大提升基因演算法在交配運算時搜查最佳解的效率。智慧型基因演算法具有強大的搜尋全域的能力，對大量參數最佳化問題提供了最佳解。

### 4.2.2 IGA 流程

傅統基因演算法(Genetic Algorithm, SGA)主要由染色體初始化、演化、染色體挑選、交配、和突變，五項主要運作而組成。智慧型基因演算法基本上除了染色體作智慧型交配外，其他基本運作皆和 SGA 一樣。IGA 詳細步驟如下所示，見圖 4.3：

步驟一：初始化(initialization)。亂數隨機產生群族中所有染色體的基因，群族個數共有 $N_{pop}$ 個。

步驟二：評估運算(evaluation)。計算所有染色體的適應值。

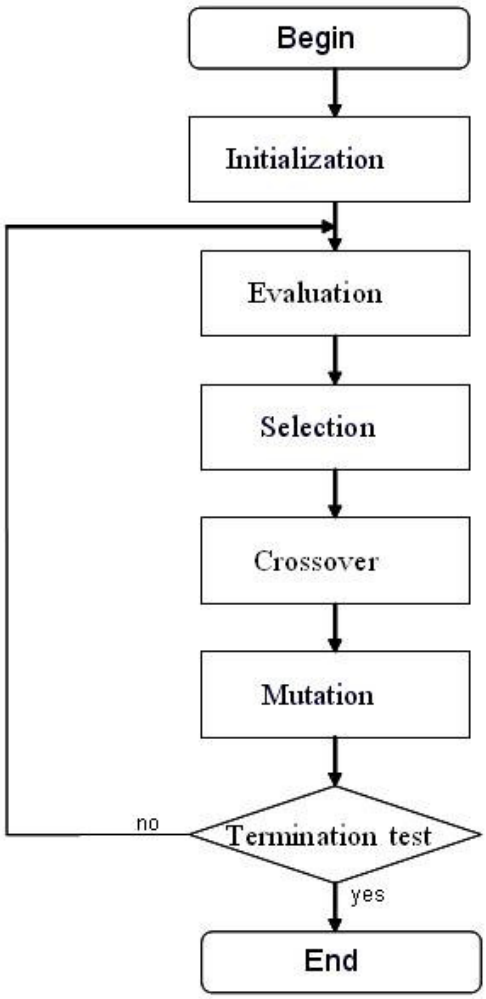



圖 4.3 IGA 流程圖

步驟三：選擇運算(selection)。從目前的族群中選取最佳的 $N_{pop} \cdot P_s$ 個染色體，取代最差的 $N_{pop} \cdot P_s$ 個染色體，組成新的群族。其中 $P_s$ 是選擇率。

步驟四：交配運算(crossover)。首先依據交配率( $P_c$ )，隨機選出 $N_{pop} \cdot P_c$ 個父代染色體進行 IGC。

步驟五：突變運算(mutation)。依突變率($P_m$)，選擇出欲進行突變運算的染色體。並配合題型選擇一種突變運算，對染色體進行突變。本專題使用單點突變(bit mutation)。

步驟六：終止測試(termination test)。判斷是否到達終止條件，否則回到步驟二。

智慧型交配與傳統單點交配運算中以亂數切點產生新 染色體的方法相比較，智慧型交配運算所得的新染色體具有較高適應度的機率 ，明顯的較單點交配運算方式來的高處許多 。因此智慧型交配運算在不需要修改基因演算法的架構之下 ，便能大幅提高基因演算法的搜尋效能，具有收斂速度快與精確度高的優點。

## 4.3 智慧型多目標演化式演算法 IMOEA

### 4.3.1 IMOEA

在搜尋現實問題中的最佳解時 ，有很多最佳化問題無法以一個目標函數(objective function)來決定解的好壞，而是需要同時考量多個目標函數來判斷解的優劣 ，因為這些目標之間彼此往往是互相競爭的 。對於這些問題，我們稱為多目標最佳化問題(Multi-objective Optimiza-

tion Problems)。在多目標最佳化問題中，通常不會只存在單一個最佳解，而是一組可供選擇的解集合。並且在這組解集合中，沒有任何一個解可以在所有的目標函數上皆優於其他的解，我們稱這一組解集合為 Pareto 最佳解集合(Set of Pareto-optimal Solutions)，其中的每一個解又稱為不被支配解(non-dominated solution)。

目前有許多解決多目標最佳化問題的多目標演化式演算法，例如由 Zitzler 提出 SPEA2 [16]，及 VEGA [17]、HLGA [14]、NPGA [18]、NSGA [19]、SPEA [20]、和 NSGA-II [21]等數個多目標演化式演算法。

本研究中，我們採用智慧型多目標演化式演算法 (IMOEA)[8]來解決所研究的多目標最佳化問題。使用智慧型多目標演化式演算法能有效地尋找出 Pareto 最佳解集合。IMOEA 的優點如下：

1. 基於 Pareto 理論，使用通適化且不受尺度因素影響的的評估函數(Generalized Pareto-based Scale-Independent Fitness Function, GPSIFF)給予每個個體(individual)有區別的適應函數值。
2. 採用智慧型基因蒐集運算子(Intelligent Gene Collector, IGC)，以系統化推理的方式有效地搜尋解空間。
3. 使用優生學策略(elitism strategy)來引導運算，即合併目前群族

與優生群族，有效地增強搜尋能力。

IMOEA 的強大搜尋能力，主要是因為其採用一個有效的適應值指派機制 – GPSIFF，並搭配智慧型的組合運算 – IGC。IGC 利用直交實驗設計(OED)來進行。OED 的使用已於 4.1 節中介紹。我們將在 4.3.2 節中介紹 GPSIFF 的概念。最後，在 4.3.3 節中，我們將介紹 IMOGA 的演算法。

### 4.3.2 GPSIFF

計算個體的適應函數值對於多目標最佳化問題是個很重要的議題。為了分辨個體之間的優劣，本文採用了以 Pareto 理論為基礎的評分方式來避免尺度因素的影響，並且對於每個個體給於具有區分能力的適應函數值，以取代傳統有失準確性的排名方式和距離方式，稱之 GPSIFF[8]。

GPSIFF 使用類競爭式 (torunament-like)的評分方式來計算處於欲評分的 Pareto 解集中個體 $x$ 的適應值。GPSIFF 的數學式如下：

$$GPSIFF(x) = p - q + C，\qquad (18)$$

其中 $p$ 表示在目前欲評估的解集中 $x$ 所支配的個體數目，$q$ 表示在目前

欲評估的解集中能夠把 $x$ 支配的個體數目，$C$ 是一個較大的正整數，以保證求出的適應值為一正整數 。通常以目前參與評估運算的所有個體的數目作為正整數 $C$ 的值。

GPSIFF 的優點如下：

1. 不需調整權重值：基於 Pareto 理論來評估解的好壞，沒有權重加總法需要決定權重值的困難，也不會受到人為主觀判斷的影響。
2. 不需考量尺度因素 ：由於各目標函數值的尺度適應值不盡相同，在權重加總法中需要考慮到尺度因素，以免使得權重設定失之準確。
3. 以評分方式有效辨識不同解的優劣程度：取代傳統排名法可能將不同的解給予相同的排名，以及距離法有尺度因素影響的缺點，以精確地評分解的優劣程度。

圖 4.4 展示出在兩個目標的最小化問題中，在 Pareto 解集中，所有個體使用 GPSIFF 所計算出的適應值。其中 $C$=12，○為不被支配解。以解 A 為例，$p$=3，$q$=2，所以適應值為 13。

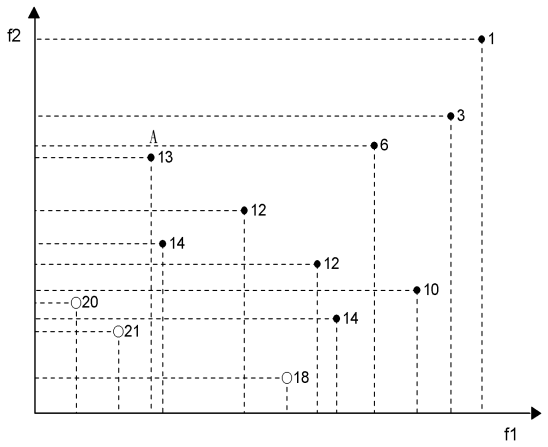



圖 4.4 GPSIFF 評分示意圖

### 4.3.3 IMOEA 流程

結合優生學策略對於多目標演化式演算化的多樣性 (diversity)與效能相當有幫助[22]。IMOEA 使用容量為 $E_{max}$ 的優生集 $E$ 以保持不被支配之解集。IMOEA 的詳細流程如下，見圖 4.5：

步驟一：初始化(initialization)。亂數隨機產生群族中所有染色體的基因，並初始化優生集 E 和暫時優生集 E'為空的狀態。優生集的容量限制為 Emax。

步驟二：評估運算(fitness evaluation)。使用 GPSIFF 計算所有染色體的適應值。

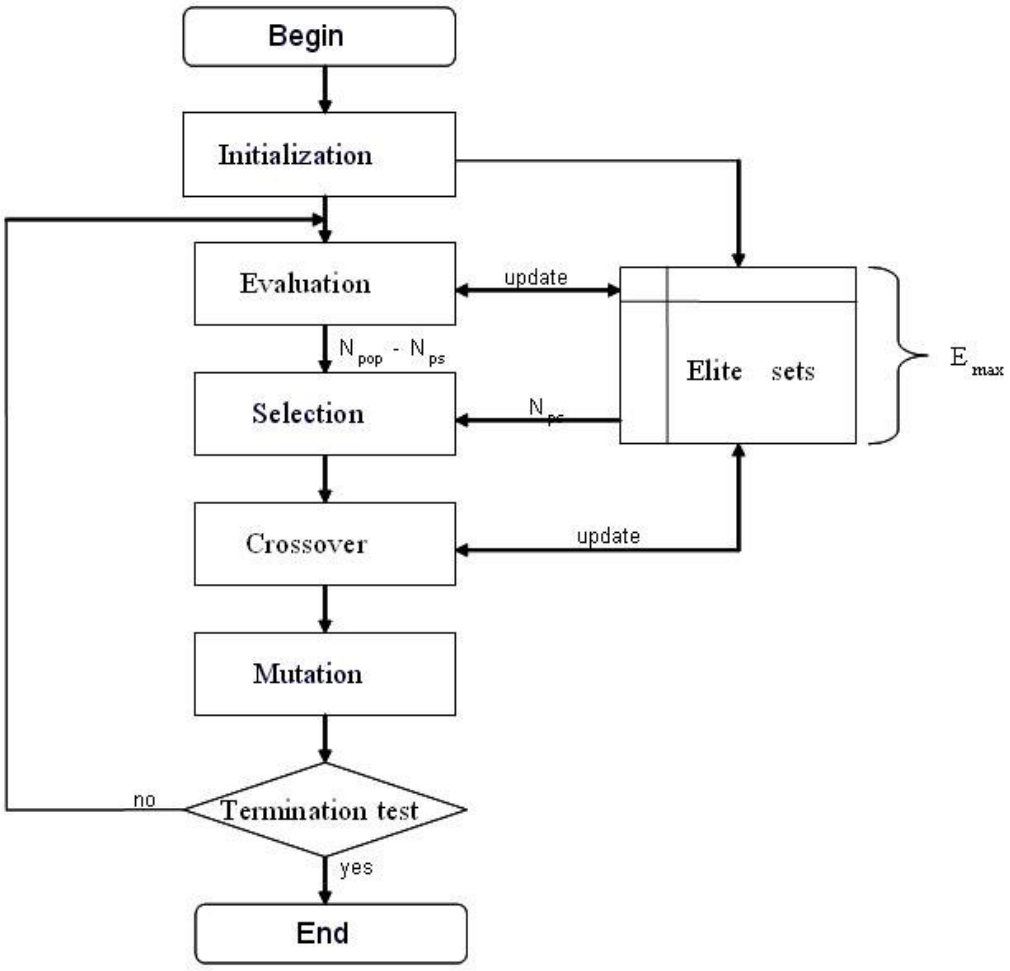



圖 4.5 IMOEA 流程圖

步驟三：更新優生集和暫時優生集(update elite sets)。把群族和 E'的不被支配解集合加入 E，並且把 E'清空。接著考慮在 E 裡面所有的解集，移除被支配的解。假若在 E 裡面所有不被支配的解集數目大於 Emax，則隨機移除超過的解。

步驟四：選擇運算(selection)。從目前的族群中選取 Npop - Nps 個染色體，其中 Nps = Npop × Ps，並從 E 中隨機選取 Nps 個不被支配的染色體，組成新的群族。

步驟五：重組運算(recombination)。首先依據交配率(Pc)，選出 Npop × Pc 父代染色體進行 IGC。對於每次 IGC 運算，會把不被支配的副產品加入 E’。

步驟六：突變運算(mutation)。依突變率(Pm)，選擇出欲進行突變運算的染色體。並配合題型選擇一種突變運算，對染色體進行突變。本專題使用單點突變(bit mutation)。

步驟七：終止測試(termination test)。判斷是否到達終止條件，否則回到步驟二。

# 第五章 單/多目標之影片配置問題最佳化

## 5.1 單目標影片配置問題最佳化

我們使用智慧型基因演算法來處理此單目標最佳化問題 。其系統的目標是最大化平均複本數且平衡系統負載，並遭遇如下限制條件：

(1) 每一伺服器的儲存容量必定不能超過。

(2) 每一伺服器的網路頻寬必定不能超過。

(3) 每個影片的所有複本要分散至不同的伺服器。

初始先隨機產生一組染色體 ，每個染色體可表現出每一部影片的複本數及影片的配置情形，並使用 IGA(詳細步驟請見 4.2 節)來進行演化，得到一個最佳解 ，其解即是最佳化的影片配置情形 ，詳細染色體編碼方式及目標函式之設計見下。

### 5.1.1 染色體編碼方式

單目標染色體編碼方式採用二進位編碼 ，以此編碼的染色體可同時表現出每一部伺服器有哪些影片的複本 ，如圖 5.1。

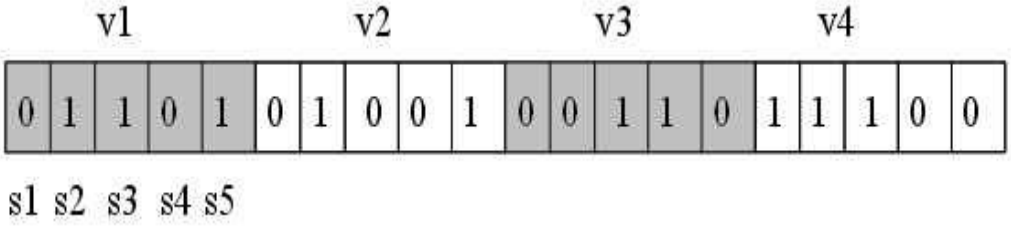



圖 5.1 基因編碼(一)

考慮一個隨選視訊系統有 $D$ 部伺服器以及 $M$ 部不同的影片，則總共有 $D \times M$ 個 bit。假如第 $[(i-1)\times D + j]$ 個 bit 為 1，代表第 $i$ 部影片有複本存在第 $j$ 部伺服器上，為 0 時代表沒有存在此伺服器上。第 $i$ 部影片數為從第 $[(i-1)\times D+1]$ 個至第 $(i \times D)$ 個 bit 的總和。例如設 $D$ 為 5，$M$ 為 4。因為有第二部影片存在第二部伺服器上，所以第 7 個 bit 為 1。圖 5.1 即秀出這個例子，且第二部影片的複本總和為 2。

### 5.1.2 目標函式的設計

影片複本與配置的目標是最大化平均複本數且平衡系統負載。設 $L$ 為系統的負載不平衡值，$r_i$ 為影片 $v_i$ 的複本數，目標函式如下：

$$\max \quad obj = \sum_{i=1}^{M} r_i / M - \alpha L, \qquad (19)$$

$\alpha$ 為權重值，將兩個目標加權起來，成為單一目標。並以此作為單目標基因演算法的目標函式。

## 5.2 多目標影片配置問題最佳化

為了建立此系統，配置系統應該確定每一部影片 的複本數，以及它們相對應的儲存位置。系統目標是：

(1) 所有伺服器的全部儲存容量減到最少。

(2) 服務阻隔率減到最少。

其限制條件是：每一伺服器的儲存容量必定不能超過 ，且每一部影片至少必須有一個複本。

圖 5.2 顯示混合式基因方法之操作流程圖，其演化使用 IMOEA。需要為批次隨選視訊系統給定一個批次時間 (batching interval) $W$。由 4.3 節得知，隨機產生一組染色體，開始進行評估運算。評估運算的方式是先將染色體(詳細編碼方式請見 5.2.1 節)當作參數給 HLF 演算法：一條染色體代表所有影片的複本數，HLF 演算法可為影片進行最佳化的配置，並計算以此配置出的系統之服務阻隔率。在 IMOEA 中，進行第 4.3.3 節的流程，演化出最佳的 Pareto 解集合。

由此，採用基因演算法及 HLF 演算法為給定的$W$找到最佳的配置情形，也得到一個最小的服務阻隔率和最小的儲存容量 。

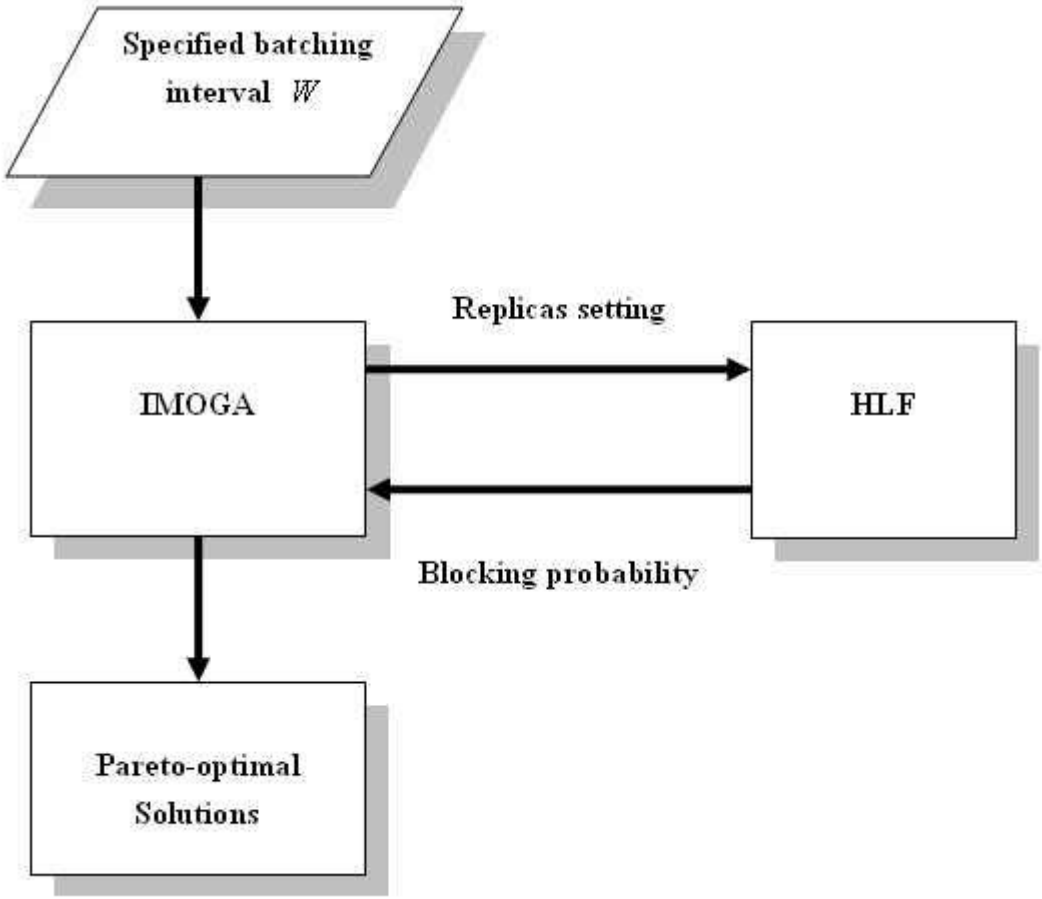



圖 5.2 混合式基因方法之操作流程圖

## 5.2.1 染色體編碼方式

我們考慮一個批次隨選視訊系統有 $D$ 部伺服器以及 $M$ 部不同的影片。但經過深入的研究，我們將染色體編碼成一個整數字串 $I=\{n_i,...,n_M\}$，$n_i$ 表示第 $i$ 部影片的複本數，如圖 5.3，代表第 1 部影片有 3 個複本，第 $M$ 部影片有 6 個複本。如此，這個研究的問題相當於在 $M$ 維度的搜尋空間裡找一個最佳的解 $I=[n_1,...,n_M]^T$，在空間中每一點表示

一組影片的配置。同時它需考慮以下之限制條件：每一部影片至少必須有一個複本，且複本數的上限是 $D$。所以，這個研究的問題之搜尋空間由 $n_i \in [1, D], i = 1,...,M$ 組成。


圖 5.3 基因編碼(二)

## 5.2.2 目標函式的設計與 Pareto 支配條件

我們將此影片配置問題視為一多目標最佳化問題，並採用多目標演化式演算法來尋求最佳解集合。由於這些目標之間彼此是互相競爭的，所採用的多目標演化式演算法可同時考量多個目標函數來判斷解的優劣。以下為本研究所設定的兩個目標函式：

(1) 目標函式 $f1$ 為系統整體服務阻隔率($B$)：

$$f1 = B \times 100\% \tag{20}$$

(2) 目標函式 $f2$ 是所有伺服器的全部儲存容量：

$$f2 = \sum_{i=1}^{M} v_i \cdot n_i \tag{21}$$

$v_i$ 代表第 $i$ 部影片的大小；$n_i$ 代表第 $i$ 部影片的複本數。此外，如

之前所言，我們藉由 HLF 演算法處理影片的配置問題並求得系統整體之服務阻隔率 $B$。

因為我們的系統目標是將所有伺服器的全部儲存容量減到最少，且服務阻隔率減到最少。所以當兩個目標函式值同時比另一染色體的兩個目標值還低時，即被另一染色體支配：

$$f1(I) > f1(I') 且 f2(I) > f2(I') \qquad (22)$$

染色體 $I'$ 之服務阻隔率小於染色體 $I$ 而且染色體 $I'$ 之儲存容量也小於染色體 $I$，此時，染色體 $I'$ 就會支配染色體 $I$。

確定支配條件後，在 Pareto 解集中，所有個體使用 GPSIFF 來計算出其適應值，以此來分辨個體之間的優劣。

# 第六章 實驗結果

## 6.1 單目標最佳化實驗

系統的組態如下所述。為了比較，實驗設計與[11]相同。此隨選視訊系統有 8 部同質的伺服器，每一部伺服器擁有 1.8Gbs 的對外的網路頻寬。系統包含 200 部影片，每一部影片片長皆為 90 分鐘，編碼速率(encoding bit rate)皆固定為 4Mbs。一部影片儲存需求為 2.7GB。每部伺服器最大的儲存空間為 202.5GB。此系統最大儲存空間為 600 個複本，即最大複本等級為 3。

在 90 分鐘的尖峰時刻裡，需求到達是由 Poisson 程序依到達速率(arrival rate)$\lambda$[5]產生。因為系統對外網路頻寬有 3600 條串流，每一串流是 4Mbs，所以$\lambda$最高是每秒 40 個需求到達。影片的熱門程度分布是由$\delta$決定[5,23]。

設定 IGA 的參數 $p_c = 0.8$ ， $p_s = 0.2$ ， $p_m = 0.01$ ， $N_{pop} = 50$ ，終止條件是 100 個代數。另外目標函式的權重值 $\alpha = 1/120$ 。

假如需求的網路頻寬無法得到時 ，這個需求會被拒絕 。我們使用拒絕率當作效能的評估制度。

### 6.1.1 拒絕率和負載不平衡(load imbalance)的表現

為了了解其他演算法在效能上的影響 ，結合了影片複本演算法(Zipf-like distribution based replication[11] and classification based replication[24] )與影片配置演算法(round-robin placement and smallest load first placement[11] )，從[11]搜集報告出這四種演算法的結果。

圖 6.1 與圖 6.2 展現各種演算法的表現 ，如分類複本演算法(classification replication)結合輪詢配置演算法 (round-robin placement)為 CR，分類複本演算法結合最少負載優先配置演算法 (smallest load first placement)為 CS，Zipf-like 分佈複本演算法結合輪詢配置演算法為 ZR，Zipf-like 分佈複本演算法結合最少負載優先配置演算法 (smallest load first placement)為 ZS，和以 IGA 為基礎的影片複本配置演算法(IGA)。

圖 6.1(a)與圖 6.1(b)描述了四種演算法在拒絕率上的表現，其複本等級為 3，$\delta$ 分別是 1.0 與 0.5。結果顯示被提議的方法表現出較低的拒絕率。IGA 的解支配了所有現存方法的解，展現出以 IGA 為基礎的方法之高效能。

圖 6.2 描述了四種演算法在負載不平衡上的表現，其複本等級為

3，$\delta$分別是 1.0。以 IGA 為基礎的影片複本配置演算法表現比其他演算法好，在負載不平衡上獲得令人滿意的表現。

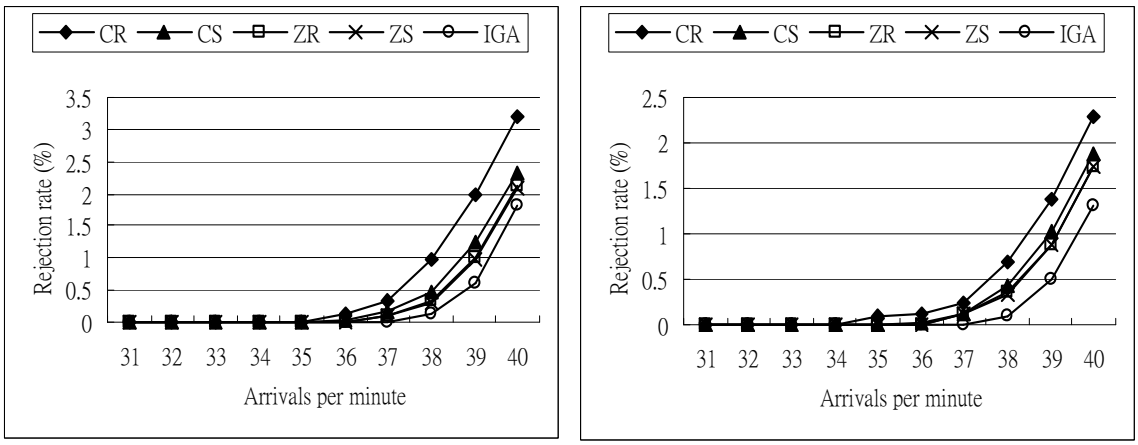



圖 6.1 各種演算法的拒絕率(a) $\delta = 1.0$; (b) $\delta = 0.5$

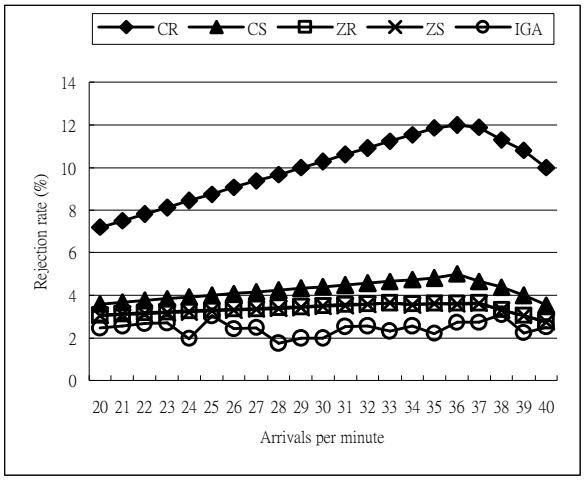



圖 6.2 各種演算法的負載不平衡

## 6.2 多目標最佳化實驗

系統的組態如下所述。設想影片檔案的熱門程度由 Zipf 定律[9]決

定的，計算如下：

$$p_i = \frac{c}{i^{\delta}} \quad i = 1,...,M \qquad (23)$$

其中 $c = \sum_{i=1}^{M} \left(1/i^{\delta}\right)^{-1}$ 且 $\delta$ 是描述檔案熱門程式分布的常數。

在這個系統，設想有 200、500 或 1000 部影片，經由隨機產生成一串列，規格類似一些本地隨選視訊服務。200 部影片的播放時間如圖 6.3。

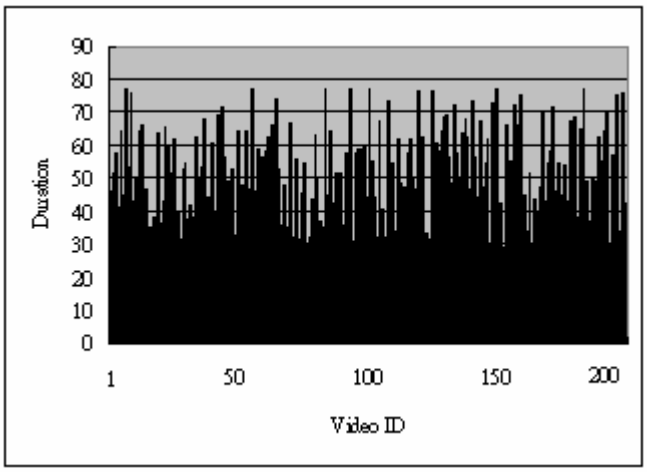



圖 6.3 影片的複本數(依照熱門程度進行排列)

隨選視訊系統由兩種不同的伺服器類型組成 。類型一伺服器可以同時地支援 50 I/O 串流以及 120GB 的儲存容量；類型二伺服器只可以支援 30 I/O 串流以及 70GB 的儲存容量。在系統中，類型一伺服器數量是 15，類型二伺服器數量是 5。

假定到達速率 (arrival rate) $\lambda = 15$ ，問題的準確度 (accuracy of the solution) $Q_j = 0.0001$。設定 IMOEA 的參數 $p_c = 0.8$ ， $p_s = 0.2$ ， $p_m = 0.01$ ， $N_{pop} = 50$ 和 $N_{E\max} = 50$ 。IMOEA 的染色體被切成 $N = 15$ 個基因切段，直交表使用 $L_{16}\left(2^{15}\right)$。對於 IMOEA 和[1]，最後的終止標準是 30,000 個適應函數估算次數。

### 6.2.1 測驗一：多目標最佳化

在這個系統有 200 部影片。影片的熱門程度根據 Zipf 定律 $\delta = 0.271$ 來應用。 IMOEA 的 Pareto fronts 合併了 30 個獨立的實驗結果陳列在圖 6.4。為了與[1]比較，它們的解也顯示在圖 6.4。從圖 6.4 可以發現：IMOEA 大部分的解支配了[1]的解，除了極度小的儲存容量或是極度小的服務阻隔率(blocking probability)。以 IMOEA 為基礎的 Pareto 方法可以提供不被支配解(non-dominated solutions)，它們的兩個目標品質都比[1]的品質好。然而，以 IMOEA 為基礎的 Pareto 方法也提供解的多樣性，這種解擁有不同的儲存容量和服務阻隔率的結合 。隨選視訊系統可以花費較多的儲存容量來降低服務阻隔率 ，或者是花費較少的儲存容量而造成較多的服務阻隔率。以 IMOEA 為基礎的 Pareto 方法在系統的組態上可獲得更好的彈性。

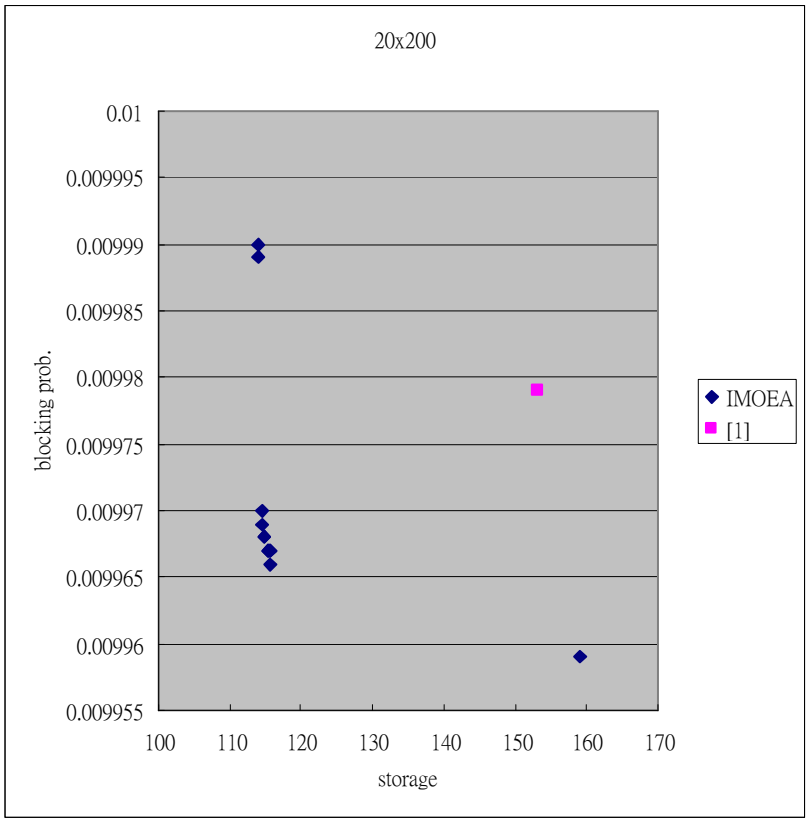



圖 6.4 IMOEA 與[1]的 Pareto fronts，批次時間為 1.97

### 6.2.2 測驗二：熱門程度的分布

影片熱門程度的分布意味著不同顧客的喜愛。當$\delta$較小時，熱門程度是較均勻的；當$\delta$較大時，表示需求集中在某幾部特定的影片上。

設想三個不同的實例(影片數是 200)：

實例一：$\delta = 0$，請看圖 6.5。

實例二：$\delta = 0.271$，請看圖 6.4。(等同於測驗一)

實例三：$\delta = 1$，請看圖 6.6。

每個實例包含 IMOEA 的 Pareto 解與[1]的單一解，合併後的結果展現到相對應的圖上。結果說明不管影片熱門程度的分布有何不同，以 IMOEA 為基礎的 Pareto 方法可以提供較好的解，它們的兩個目標品質都比[1]的品質好。

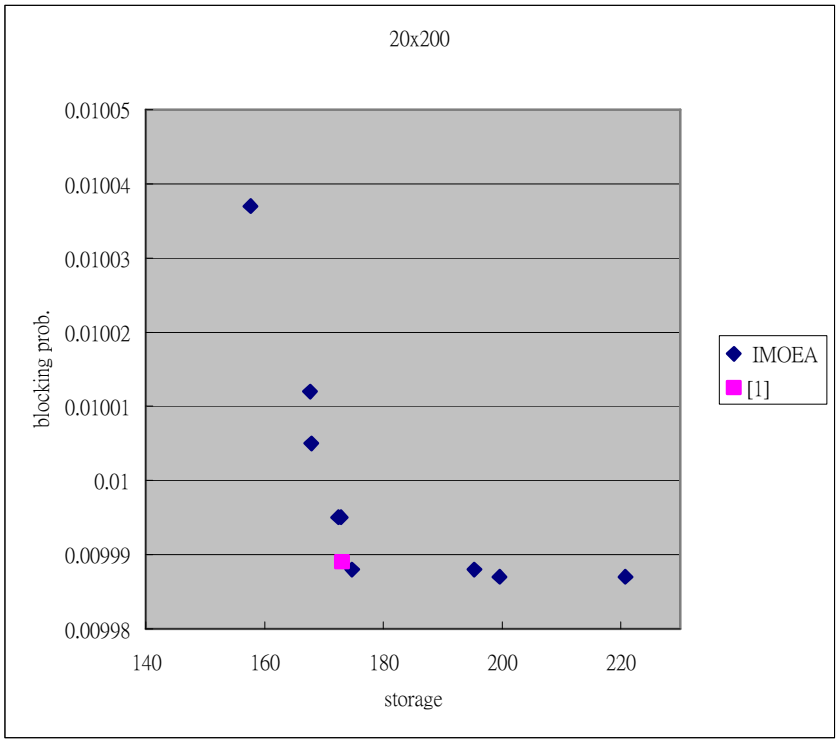



圖 6.5 IMOEA 與[1]的 Pareto fronts，$\delta = 0$

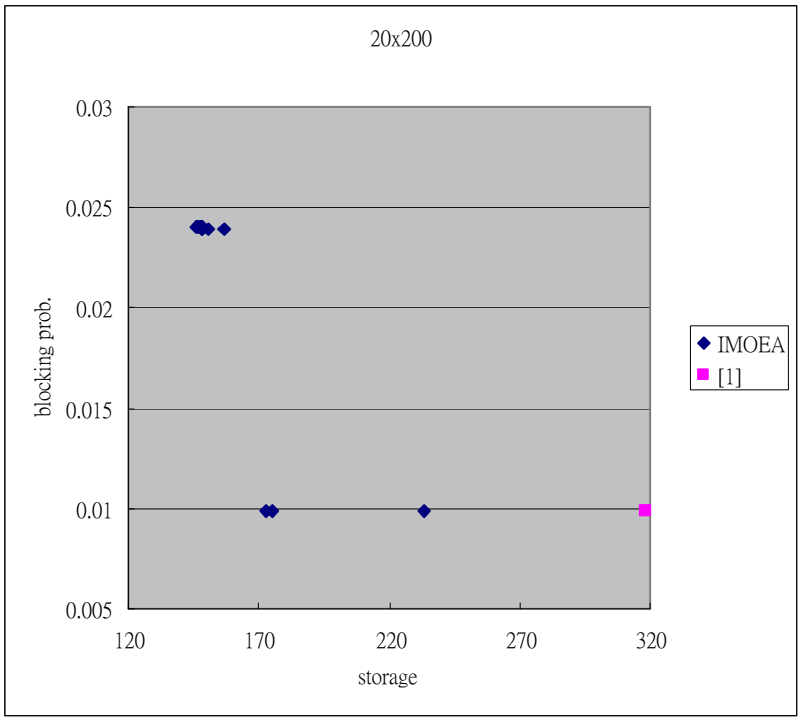



圖 6.6 IMOEA 與[1]的 Pareto fronts，$\delta = 1$

### 6.2.3 測驗三：影片數量

隨選視訊系統通常有很大量的影片數 。隨著影片數的增加 ，將造成搜尋空間變大、更複雜問題的困難度。

設想兩個不同的實例($\delta$ 是 0.271)：

實例一：200 部影片 ，請看圖 6.4。(等同於測驗一)

實例二：500 部影片 ，請看圖 6.7。

實例三：1000 部影片 ，請看圖 6.8。

雖然影片數增加，IMOEA 解的品質仍然比[1]好，甚至更好。以 IMOEA 為基礎的 Pareto 方法可以有效的解決大量的多目標參數之最佳化問題。

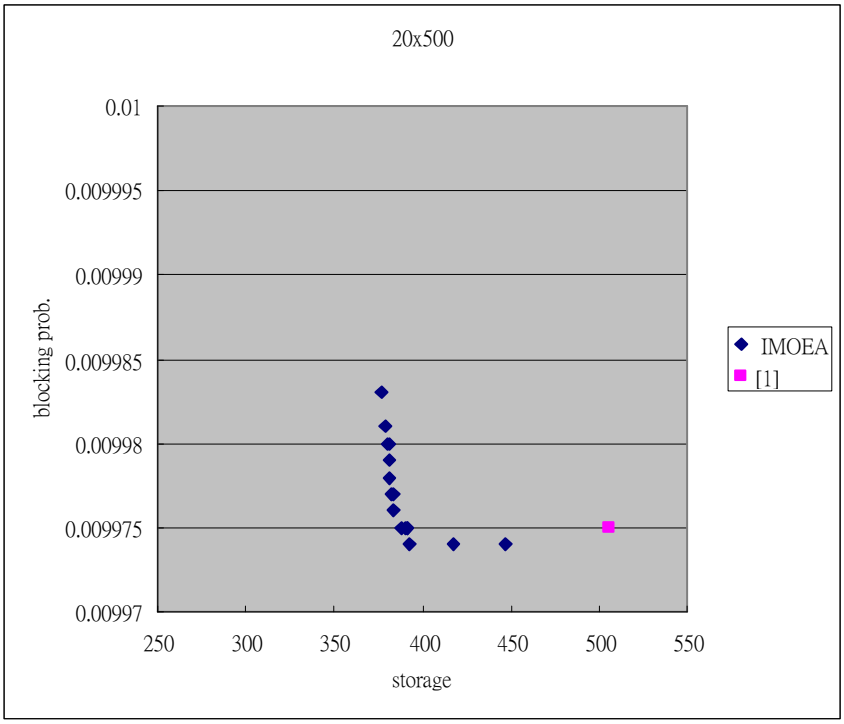



圖 6.7 IMOEA 與[1]的 Pareto fronts，總共有 500 部影片

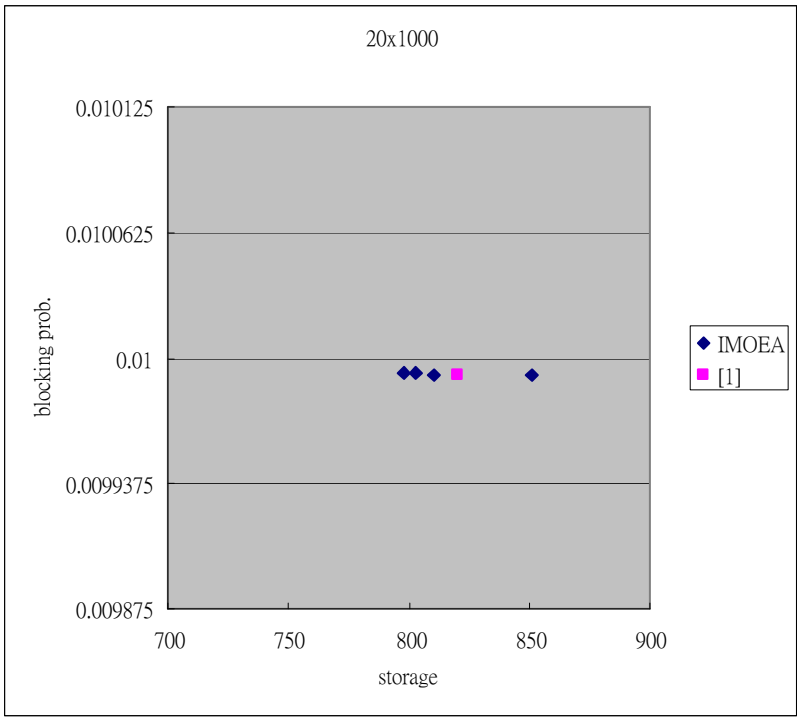



圖 6.8 IMOEA 與[1]的 Pareto fronts，總共有 1000 部影片

# 第七章 結論

## 7.1 專題結論

在本專題所討論的影片配置最佳化問題中，我們分別考量單目標最佳化及多目標最佳化兩種方式。在單目標最佳化所考量的模型中，我們使用了智慧型基因演算法來尋求最佳的配置情形。為了增加服務的可得性，我們一方面儘量提高影片的複本數；另一方面儘量達成系統的負載平衡以降低拒絕率。實驗結果顯示，透過智慧型基因演算法所找到的最佳解，在系統之負載平衡與拒絕率的考量上，相較於實驗中其他非以基因演算法為基礎的方法，均能有較好且明顯的表現。

在多目標最佳化中，我們將配置問題視為一個多目標最佳化問題。在我們所考量的模型中，具有以下兩個望小的最佳化目標：(1)系統總儲存容量及(2)系統整體服務阻隔率。我們使用智慧型多目標基因演算法同時最佳化此兩個目標，並求得一組精確的 Pareto 解集合。使得在建置隨選視訊系統時，系統建置者可以依照成本及效能的考量，來選擇適當的建置方案。因此，大大地提升了建置系統的彈性。透過實驗數據可以觀察到，我們提出的方法在相同的效能下，可有效地節

省儲存成本約 20%~25%。

此外，根據上述之結果，我們可觀察到所使用的智慧型基因演算法與智慧型多目標演化式演算法，在處理具有大量參數的最佳化問題上確實有很好的表現。因此，確實適合用來解決隨選視訊系統上之大量影片配置問題。

## 7.2 參與人員負責工作

**陳彥百**

1. 蒐集專題之相關背景文獻資料。
2. 共同參與專題研究之討論。
3. 學習傳統基因演算法(SGA)。
4. 學習智慧型基因演算法(IGA)。
5. 利用 GA LIB 實作智慧型基因演算法。
6. 撰寫會議論文並發表於『中華民國人工智慧學會』的研討會並參加該研討會。
7. 學習多目標智慧型基因演算法(IMOEA)。
8. 實作 Highest-Load-First(HLF)演算法之程式。

9. 參與整理與分析實驗數據。

10. 整理並撰寫專題研究報告。

11. 撰寫期刊論文。

**林凱遠**

1. 蒐集專題之相關背景文獻資料。

2. 共同參與專題研究之討論。

3. 學習傳統基因演算法(SGA)。

4. 學習智慧型基因演算法(IGA)。

5. 與專題同學一起發表 1 篇會議論文，北上參與『中華民國人工智慧學會』的研討會，並親自上台報告。

6. 學習多目標智慧型基因演算法(IMOEA)。

7. 編寫 SGA、IGA、IMOEA 程式碼。

8. 結合 IMOEA 與 Highest-Load-First(HLF)的程式。

9. 設計並製作實驗。

10. 分析與整理實驗數據。

11. 整理並編寫專題研究報告。

12. 參與指導老師投稿期刊論文之計劃。

13. 參與整理並編寫期刊論文之實驗部分。

**黃炯明**

1. 學習傳統基因演算法(simple genetic algorithm ,SGA)及其應用方式。
2. 學習智慧型基因演算法(intelligent genetic algorithm ,IGA)及其應用方式。
3. 以 C++實作整套 SGA、IGA 的程式庫。
4. 以學長發表過的論文為對象，作為簡報及實驗的練習。
5. 與專題同學一起發表 1 篇會議論文
6. 北上參與『中華民國人工智慧學會』的研討會。
7. 製作研討會的簡報內容。
8. 檢討會議論文的優缺點。
9. 整理 Video-on-Demand(VOD)的傳輸方式。
10. 從傳輸方式裡面尋找適合的論文題目。
11. 分析『串流合併』作為題目的可行性。
12. 與指導教授及專題同學共同決定期刊論文的題目。
13. 學習智慧型多目標基因演算法(intelligent multi objective genetic algorithm ,IMOGA)及其應用方式。

14. 分析 Highest-Load-First(HLF)的流程及意義。

## 7.3 心得感想

### 陳彥百

為期三個學期的專題，其實是相當長的一段時間。現在，終於告一個段落了，心中真是百感交集。回顧整個研究歷程，從一開始對基因演算法的懵懵懂懂，到發表會議論文的初試啼聲，再到遇到平頸時的手足無措，在這一路上，我們遇到各種的拙折，我們也盡我們的力量將之一一克服。最後，到現在走完全程，終於對作研究的概念也算有點心得。其實，作研究的整個流程是一門很高深的學問，每一個環節都要抱持著嚴謹的態度，細細考量。在此，學生很感謝指導老師何信瑩教授的諄諄教誨，讓學生可以學到這麼有用的學問。以及實驗室的學長姐們熱心的提攜與指導，讓我們可以很快地進入狀況。

經過一連串的研究工作，學生體悟到一點：作研究其實是訓練一個人對事物了解、整理、分析的一種過程，並學習釐清問題，把焦點集中在所專注的問題上，最後，清楚明確地提出自己的想法。此一過程無疑地都需要經過再三反覆地訓練才能好好掌握。此外，如果能延

伸此概念應用於其他的範疇，相信不管是處理什麼問題都能獲得很好的結果。

## 林凱遠

我們的專題是三個人為一組來共同研究　，除了增進專業的技術外，也培養我們之間的團隊合作，這是其他課程無法得到的。

在開始規劃專題時，時常會有意見不同的時候，我們靠著密切的討論、同時針對不同的議題進行辯論，經由這種過程，我們時常想出新的作法、解法。有了新的想法之後，將它以投影片的方式報告並講解給老師與學長聽，同時老師也提示很多我們所沒想到的層面。這些都是良性的互動，藉此訓練獨立思考、口才等。專題可說是研究所的開端，藉著專題的過程，我了解到實驗室裡老師與學長的互動，還有作研究是怎麼回事。

三個學期的專題，學習的過程是我以前沒有經歷過的，有難過的事，也有欣慰的事；有專題陷入低潮沒有任何進展，也有上台報告之後的成就感。當我未來有所成就時，會回憶起大學專題所帶來的豐富歷練。

特別感謝何信瑩老師與最佳化系統實驗室的學長姊，他們帶給我

們豐富的知識、研究的態度、待人處事等，還有也感謝我們這一組的成員彥百、炯明，在這專題的過程中帶給我的一切。

**黃炯明**

參與這次的專題，讓我收穫良多，不僅學習到很多專業的知識，及團隊合作的默契，更學習到，很多待人處世的道理。

雖然在專題製作的路上，很辛苦也很忙碌，中間也轉換過跑道，但我覺得這些努力，都是值得的。這個過程，就像是從學會小工具，轉而到製作出一件成果來一樣，從沒有經驗，到不斷的挫折，然後再從低潮向上爬，像是洗三溫暖一樣；如果這僅僅只是學校內的專題，我想以後就業，過程一定是比這更辛苦，而我有了現在的經驗，肯定可以在未來，帶給我比其他人更多的優勢。

這次的專題，我很感謝能夠在指導老師，何信瑩教授的帶領下，走過每一步，不管是專業知識的教導，或者是經驗上的累積，處處都是我學習的目標，同時也很感謝系統最佳化實驗室的學長姐，犧牲自己的時間，提攜後輩，對於我們小組的問題，都是同時兼具耐心與毅力，全心全力的指導。

最後，我一定要感謝同一小組的隊員，我們三個人，彼此都是在

百忙之中，抽空聚會，討論專題內容，大家都是相互扶持和努力，其中最大的收穫，就是讓我學會團隊合作的默契，令我受益良多。

# 參考文獻


[1] K. S. Tang, K. T. Ko, S. Chan, and E. W. M. Wong, "Optimal video placement scheme for batching VOD services," *IEEE Transaction on Broadcasting*, vol. 50, no. 1, pp. 16-25, 2004.

[2] S. H. G. Chan, F. Tobagi, and T. M. Ko, "Providing on-demand video services using request batching," in *IEEE Int. Conf. Communication*, vol. 3, 4998, pp. 1716-1722.

[3] A. Dan, P. Shahabuddin, D. Sitaram, and D. Towsley, "Channel allocation under batching and VCR control in movie-on-demand servers," *J. Parallel Distrib. Comput.*, vol. 30, pp. 168-179, Nov. 1995.

[4] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true vido-on-demand services," in *Proc. ACM Multimedia Conf.*, Bristol, U.K., Sept. 1998.

[5] L. Golubchik, R. R. Muntz, C. Chou, and S. Berson, "Design of fault-tolerant large-scale VoD servers: with emphasis on high-performance and low-cost," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 4, pp. 363-386, 2001.

[6] K. S. Tang, K. T. Ko, S. Chan, and E. W. M. Wong, "Optimal file placement in VOD system using genetic algorithm," *IEEE Transaction on Industrial Electronics*, vol. 48, pp. 891-897, 2001.

[7] S.-Y. Ho, C.-C. Liu, and S. Liu, "Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm," *Pattern Recognition Letters*, vol. 23, no. 13, pp. 1495-1503, 2002.

[8] S.-Y. Ho and L.-S. Shu, and J.-H. Chen, “Intelligent evolutionary algorithms for large parameter optimization problems,” *IEEE Trans. Evolutionary Computation*, in press.

[9] C. Xu and F. Lau. Load Balancing in Parallel Computers: Theory and Practice. Kluwer Academic Publishers, 1997.

[10] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. The maximum factor queue length batching scheme for video-on-demand systems. *IEEE Trans. on Computers*, 50(2):97-110, 2001.

[11] X. Zhou and C. –Z. Xu. "Optimal video replication and placement on a cluster of video-on-demand servers," in *Proc. IEEE ICPP’02* 2002, pp. 547-555.

[12] D. Bertsekas and R. Gallager, Data Networks, 2nd ed: Prentice Hall, 1922, p. 179.

[13] G. Taguchi and S. Konishi, *Orthogonal Arrays and Linear Graphs*. MI: American Supplier Institute, 1987.

[14] P. Hajela and C.-Y. Lin, "Genetic search strategies in multicriterion optimal design," *Structural Optimization*, pp. 99-107, 1992.

[15] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computaton*, vol. 2, pp. 221-248, 1994.

[16] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. EUROGEN 2001-Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95-100.

[17] J. D. Schaffer, "Multi-objective optimization with vector evaluated genetic algorithms," in *Proc. of 1st Int. Conf. Genetic Algorithms*, 1985, pp. 93-100.

[18] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," in *Proc. of 1st IEEE Conference Evolutionary Computation*, 1994, pp. 82-87.

[19] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computaton*, vol. 2, pp. 221-248, 1994.

[20] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strengthen Pareto approach," *IEEE Transaction on Evolutionary Computation*, vol. 3, pp. 257-271, 1999.

[21] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.

[22] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strengthen Pareto approach," *IEEE Transaction on Evolutionary Computation*, vol. 3, pp. 257-271, 1999.

[23] N. Venkatasubramanian and S. Ramanathan, "Load management in distributed video servers," in *Proc. IEEE ICDCS'97*, 1997, pp. 31-39.

[24] X. Zhou and C. –Z. Xu. "Request redirection and data layout for network traffic balancing in cluster-based video-on-demand servers," in *Proc. IEEE PDIVM Workshop*, 2002, pp. 127-134.

# 附錄 – 會議論文

## An Evolutionary Approach to Optimal Video Replication and Placement on a Cluster of Video-on-Demand Servers

Shinn-Ying Ho, Y.-P. Chen, K.-Y. Lin, and C.-M. Huang

Proceedings of Joint Conference on AI, Fuzzy System, and Grey System
Taipei, Taiwan, 2003

# An Evolutionary Approach to Optimal Video Replication and Placement on a Cluster of Video-on-Demand Servers

Shinn-Ying Ho*, Yan-Pai Chen, Kai-Yuan Lin, and Chiung-Ming Huang
(何信瑩) (陳彥百) (林凱遠) (黃炯明)
Department of Information Engineering and Computer Science
Feng Chia University Taichung, Taiwan 407, ROC.
*Tel: (04) 24517250 ext. 3753, Fax: (04) 24516101, e-mail: syho@fcu.edu.tw

## Abstract

*An optimization problem of video replication and placement on a cluster of Video-on- Demand (VoD) servers is investigated in this paper. Under the assumption of single fixed encoding bit rate for all videos, the goals of the optimization problem are to maximize the number of replicas of each video, balance the workload of the servers and enhance the service availability. We propose an efficient method using an intelligent genetic algorithm IGA which is superior to conventional methods in solving large parameter optimization problems to obtain an optimal solution. By the experimental results, it is shown that the IGA-based approach performs better than the existing Zipf-like distribution-based method in solving the optimization problem.*

***Keywords:*** Genetic algorithm, optimization, video placement

## 1. Introduction

With the advances in digital video, disk and network technologies, VoD services have come into practice in recent years. Generally, there are two different architectures for VoD servers: shared storage and distributed storage. A shared storage cluster is usually built on RAID systems [4]. The advantages of such system are easy to build and administrate and low-cost storage. However, they have been limited on scalability and reliability due to disk access contention. As the number of disks increases, so do the controlling overhead and the probability of a failure [3]. On the contrary, in a distributed storage VoD cluster, each server has its own disk storage subsystem [3, 8] and is linked by a backbone network. This type of cluster architecture can offer better scalability in terms of storage, streaming capacity and higher reliability [14].

VoD systems usually require huge storage of videos and have many requirements from clients. Specially, in large-scale systems, the VoD cluster has to store several hundreds of videos and satisfies several thousands of requirements from clients. Therefore, it is critical to system performance that how to determine the replication degree of video and how to place the video replicas on the distributed cluster. It is well known that the increasing replication degree can enhance the flexibility of systems to balance the expected load. Load balancing improves the system throughput in rush-hours and hence reduces the rejection rate of entire system [13].

The video replication and placement problems have been studied extensively in literature [2, 11, 12, 14]. Chervenak et al. argued replication based on Zipf-like distribution access patterns could improve throughput [2], but the authors didn't show how to take the advantage of Zipf-like distributions for replication and placement. N. Venkatasubramanian et al. proposed a family of heuristic algorithms for dynamic load balancing in a distributed VoD server [12]. D. N. Serpanos et al. present MMPacking to achieve load and storage balancing according to weighted scheduling of client request [11]. Zhou and Xu reduce the complexity of the replication algorithm and solve the optimization problem utilizing the information about Zipf-like distribution and the smallest load first placement algorithm [14].

This paper proposes an efficient evolutionary approach to optimal video replication and placement for high availability under resource constraints. In this study, we investigate a key issue in the design of this type of clusters, i.e. initial placement of videos onto the servers and formulate the video replication and placement on a distributed storage VoD cluster for high availability as a combinatorial optimization problem with a large number of parameters. Under the assumption of single fixed encoding bit rate for all videos, we present an effective method capable of solving the large parameter optimization problems using an intelligent genetic algorithm IGA, to obtain an optimal solution. The merit of IGA is the use of orthogonal experimental design (OED) [10] to

achieve an intelligent genetic crossover operation. The chromosomes of children are formed from the best combinations of the better genes of their parents, rather than by random combinations of the parents' genes, as in conventional approaches. The better genes are chosen by a systemic reasoning approach for evaluating the contribution of individual genes based on OED [5]. By the experimental results, it is shown that the IGA-based approach has better performances than the Zipf-like distribution-based method [14] in terms of rejection rate and load imbalance.

The rest of the paper is organized as follows. Section 2 describes the investigated optimization problem. Section 3 briefly introduces the used intelligent genetic algorithm IGA. Section 4 presents the design of optimal video replication and placement using IGA. Section 5 reports the experimental results and Section 6 brings the conclusion.

## 2. The Investigated Optimization Problem

### 2.1 The VoD Model

We make the following considerations: a cluster of $N$ homogeneous servers, $S = (s_1, s_2, \ldots, s_n)$, and a set of $M$ different videos, $V = \{v_1, v_2, \ldots, v_m\}$. Each server has a storage capacity $C$ and an outgoing network bandwidth $B$. All videos in the set V have the same duration $d$, say $d$=90 minutes for typical movies and single fixed bit rate $b$.

The popularity of videos varies with the number of requests of videos. We consider the replication and placement for the peak period of length $d$. Load balancing is critical to improving throughput and service availability during the peak period. We consider the outgoing bandwidth is the major performance bottleneck. Regarding the video relative popularity distributions and the request arrival rates, we make two assumptions as follows:

1. The popularity of the videos, $p_i$, is know before the replication and placement. The relative popularity of videos follows Zipf-like distributions with a skew parameter $\theta$. The probability of choosing the i$^{th}$ video is $p_{i=i^{-\theta}/\sum_{i=1}^{M} j^{-\theta}}$. Typically, $0.271 \le \theta \le 1$ [13].
2. The peak period is same for all videos with various arrival rates. Let $\lambda$ denote the average arrival rate during the peak period. Because of the same peak period assumption, the video replication and placement is conservative as it places videos for the peak period.

### 2.2 Formulation of the Optimization Problem

The objective of the replication and placement is to maximize the replication and balancing the workload of servers. Replication enhances availability and load balancing improves the system throughput. Let $L$ denote the communication load imbalance degree of the cluster and $r_i$ denote the number of replicas of video $v_i$. Specifically, we define the optimization objective as:

$$\max \quad obj = \sum_{i=1}^{M} r_i / M - \alpha L, \qquad (1)$$

where $\alpha$ is an appropriate weight. There are many ways for the definition of load imbalance degree [14]. We choose the following one:

$$\mathrm{L} = \max_{\forall si \in s} \left| l_i - \bar{l} \right|, \ \bar{l} = \sum_{i=1}^{N} li / N. \qquad (2)$$

This objective is subject to the following three constraints: (1) server storage capacity, (2) server outgoing network bandwidth, and (3) distribution of all replicas of an individual video to different servers. All $r_i$ replicas of video $v_i$ have the same encoding bit rate since they are replicated by the same video. Let $\pi(v_i^j)$ be the index of the server on which the $j^{th}$ of replicas of video $v_i$, $v_i^j$, places. And, $\pi(v_i) = k$ means that a replica of video $v_i$ is placed on server $s_k$. The communication weight of each replica of video $v_i$ is defined as $w_i = p_i / r_i$. By the user of a static round robin scheduling policy, the number of requests for video $v_i$ to be serviced by each replica of $v_i$ during the peak period is $w_i \cdot \bar{\lambda} \cdot d$. Let $l_k$ be the outgoing communication load on server $s_k$. Specifically, we consider the resource constraints from the perspective of server $s_k (1 \le k \le N)$ as follows:

$$\sum_{\pi(v_i)=k, \forall v_i \in V} b_i \cdot d \le C, \qquad (3)$$

and

$$l_k = \sum_{\pi(v_i)=k, \forall v_i \in V} w_i \cdot \bar{\lambda} \cdot d \cdot b_i \le B. \qquad (4)$$

We consider that all $r_i$ replicas of video $v_i$ must be placed on $r_i$ servers. Specifically,

$$\pi(v_i^{j_1}) \ne \pi(v_i^{j_2}) \quad 1 \le j_1, j_2 \le r_i, j_1 \ne j_2, \qquad (5)$$

and there is no multiple replicas of a video placed to the same server

$$1 \le r_i \le N \quad \forall v_i \in V. \qquad (6)$$

## 3. Intelligent Genetic Algorithm

The orthogonal experimental design (OED) of intelligent crossover is described in Section 3.1. Section 3.2 presents the main power of IGA, i.e., the intelligent crossover. Section 3.3 gives the algorithm IGA. The various applications using IGA can be referred to the papers [5, 6, 7].

### 3.1 Orthogonal Experimental Design

An efficient way to study the effect of several factors simultaneously is to use OED with both orthogonal array (OA) and factor analysis [10]. Many design experiments use OED for determining which combinations of factor levels (or treatments) to use for each experiment and for analyzing the experimental results. The factors are the variables (parameters), which affect the chosen response variable (fitness function), and a setting (or a discriminative value) of a factor is regarded as a level of the factor. The term "main effect" designates the effect on the response variable that one can trace to a design parameter.

OA is a matrix of numbers arranged in rows and columns where each row represents the levels of factors in each run and each column represents a specific factor that can be changed from each experiment. The array is called orthogonal because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the main effect of another factor.

Factor analysis using the OA's tabulation of experimental results can allow the main effects to be rapidly estimated, without the fear of distortion of results by the effects of other factors. Factor analysis can evaluate the effects of individual factors on the evaluation function, rank the most effective factors, and determine the best level for each factor such that the evaluation is optimized.

OED uses well-planned and controlled experiments in which certain factors are systematically set and modified, and then main effects of factors on the response can be observed. Therefore, OED using OA and factor analysis is regarded as a systematical reasoning method [10]. The merit of intelligent crossover is that the systematic reasoning ability of OED is incorporated to economically identify the good genes of parents and intelligently combine these good genes to generate offspring. The two-level OA used in the intelligent crossover is described below.

Let there be α factors with two levels for each factor. The number of experiments is $2^{\alpha}$ for the popular "one-factor-at-a-time" study. Generally, levels 1 and 2 of a factor represent selected genes from parents 1 and 2, respectively. To use an OA of α factors with two levels, we obtain an integer $\beta = 2^{\lceil \log_2(\alpha+1) \rceil}$, build an orthogonal array $L_{\beta}(2^{\beta-1})$ with β rows and β-1 columns, use the first α columns, and ignore the other β- α-1 columns. For instance, Table 1 shows an OA $L_8(2^7)$. OA can reduce the number of experiments for factor analysis. The number of OA experiments required to analyze a single factor is only β where $\alpha+1 \le \beta \le 2\alpha$. An algorithm of constructing OAs can be found in [9].

After proper tabulation of experimental results, the summarized data are analyzed using factor analysis to determine the relative effects of levels of various factors. Let $y_t$ denote a fitness value to be maximized for experiment $t$, where $t = 1, \ldots, \beta$. Define the main effect of factor $j$ with level $k$ as $S_{jk}$ where $j = 1, \ldots, \alpha$ and $k = 1, 2$:

$$S_{jk} = \sum_{t=1}^{\beta} y_t \cdot F_t , \qquad (7)$$

where $F_t = 1$ if the level of factor $j$ of experiment $t$ is $k$; otherwise, $F_t = 0$. If $S_{j1} > S_{j2}$, the level 1 of factor $j$ makes a better contribution to the fitness function than level 2 of factor $j$ does. Otherwise, level 2 is better. The most effective factor $j$ has the largest main effect difference $| S_{j1} - S_{j2} |$.

Note that the main effect holds only when no interaction exists or when it is weak, and that makes the experiment meaningful. In order to achieve an effective design, experiments should be prepared so as to reduce interaction effects. In addition, to accurately estimate the main effect, all candidate solutions corresponding to the β conducted combinations need to be feasible for constrained problems if possible.

Table 1 Orthogonal array $L_8(2^7)$

| | Factors | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Exp.no. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Function evaluation value |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $y_1$ |
| 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | $y_2$ |
| 3 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | $y_3$ |
| 4 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | $y_4$ |
| 5 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | $y_5$ |
| 6 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | $y_6$ |
| 7 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | $y_7$ |
| 8 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | $y_8$ |

### 3.2 Intelligent Crossover

Each parameter is encoded in a chromosome using binary codes. In IGA, two parents $P1$ and $P2$ produce two children $C1$ and $C2$ in one intelligent crossover operation. The parameters having identical values in two parents do not participate the crossover operation such that the chromosomes can be temporally shorten possibly resulting in using a small OA table. Let the number of all participated parameters be randomly divided into α segments where each segment is treated as a factor. The following steps describe how to use OED to achieve intelligent crossover.

Step 1: Use the first α columns of OA $L_\beta(2^{\beta-1})$ where $\beta = 2^{\lceil \log_2(\alpha+1) \rceil}$.

Step 2: Let levels 1 and 2 of factor $j$ represent the $j$th parameter of a chromosome coming from parents $P1$ and $P2$, respectively.

Step 3: Evaluate the fitness values $y_t$ for experiment $t$ where $t = 2, \ldots, \beta$. The value $y_1$ is the fitness value of $P1$.

Step 4: Compute the main effect $S_{jk}$ where $j = 1, \ldots, \alpha$ and $k = 1, 2$.

Step 5: Determine the better one of two levels of each factor. Select level 1 for the $j$th factor if $S_{j1} > S_{j2}$. Otherwise, select level 2.

Step 6: The chromosome of $C1$ is formed using the combination of the better genes from the derived corresponding parents.

Step 7: The chromosome of $C2$ is formed similarly as $C1$, except that the factor with the smallest main effect difference adopts the other level.

Step 8: The best two individuals among $P1$, $P2$, $C1$, $C2$, and β -1 combinations of OA are used as the final children $C1$ and $C2$ for elitist strategy.

One intelligent crossover operation takes β + 1 fitness evaluations, where $\alpha + 1 \leq \beta \leq 2\alpha$, to explore the search space of $2^\alpha$ combinations. Generally, $C1$ is a potentially good approximation to the best one of $2^\alpha$ combinations. The larger the value of α, the more efficient it is the intelligent crossover if there exists no or weak interaction effect among gene segments. Considering the interaction effect, the smaller the value of α, the more accurate it is the estimated main effects of gene segments. Considering the tradeoff, an efficient criterion is to minimize the interaction effects while maximizing the value of α.

For practical use, the proper value of α depends on the number of encoding parameters and their interaction effects. Generally, there are two approaches to specifying the value of α. One is to adaptively change the value of α during the evolution process [5, 6, 7]. To achieve an efficient coarse-to-fine search, α is gradually increased when the evolution proceeds [7]. The other is to use a constant value of α according to domain knowledge and simulation results.

### 3.3 Intelligent Genetic Algorithm IGA

IGA of the proposed method is given as follows:

Step 1: Initiation: Randomly generate an initial population with $N_{pop}$ individuals.

Step 2: Evaluation: Evaluate fitness values of all individuals.

Step 3: Selection: Use the simple ranking selection that replaces the worst $Ps \cdot N_{pop}$ individuals with the best $Ps \cdot N_{pop}$ individuals to form a new population, where $Ps$ is a selection probability.

Step 4: Crossover: Randomly select $Pc \cdot N_{pop}$ individuals to perform intelligent crossover operations, where $Pc$ is a crossover probability.

Step 5: Mutation: Apply a conventional bit-inverse mutation operator to the population using a mutation probability $Pm$. To prevent the best fitness value from deteriorating, mutation is not applied to the best individual.

Step 6: Termination test: If a prespecified termination condition is satisfied, stop the algorithm. Otherwise, go to Step 2.

## 4. IGA-based Video Replication and Placement

### 4.1. Chromosome Encoding and Fitness Function

The feasible solution S corresponding to the video placement is encoded using a binary string consisting of N*M bits. The [( $i$-1) * N + $j$]-th bit has value 1 when a replica of $i$-th video stored at $j$-th server, and 0 otherwise. The total number of replicas of $i$-th video is the sum of values from [(i-1)*N+1]-th to (i*N)-th bits. For example, let N 5 and M 4. The 7-th bit has value 1 because a replica of the second video stored at the second server. Fig. 1 shows this example and the number of replicas of the second video is 2.

The fitness function F(S) is defined as Eqn. (1). Let $\alpha$ =1/120 in this study.

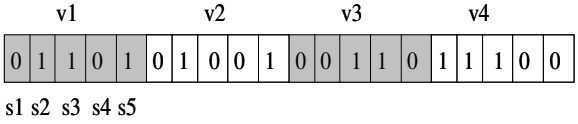



Fig. 1. Chromosome encoding.

### 4.2. The Used IGA

The parameters of IGA are as follows: Npop = 50, Ps = 0.2, Pc = 0.8, and Pm = 0.01. The termination condition is 100 generations. Note that the best individual is retained without being subject to the mutation operation. All the simulation results are the average values of 100 independent runs.

## 5. Performance Comparison

### 5.1 Experiment Design

For comparison, the experiment design is the same with [14], described below. In the simulations, the VoD cluster has 8 homogeneous servers where each has 1.8 Gbs outgoing network bandwidth. The cluster contains 200 videos with duration 90 minutes each. The encoding bit rate for videos is fixed 4 Mbs. The storage requirement of a video is 2.7 GB. The maximum storage capacity of each server is 202.5 GB. There are the maximum storage capacity of the cluster is 600 replicas and the maximum replication degree is 3.0.

Within the peak period of 90 minutes, the request arrivals are generated by a Poisson process with arrival rate $\lambda$ [4]. Since the outgoing network bandwidth of the cluster is 3600 streams of 4 Mbs, the peak rate of $\lambda$ is 40 requests per minute. The video popularity distribution is governed by the parameter $\theta$ [4, 12].

The simulation employed a simple admission control that a request is rejected if required communication bandwidth is unavailable. We use the rejection rate as the performance metric.

### 5.2 Performance of Rejection Rate and Load Imbalance

To better understand the impact of other algorithms on performance, the results of four algorithms combined by replication algorithm (Zipf-like distribution based replication [14] and classification based replication [15]) and placement algorithm (round-robin placement and smallest load first placement [14]) are reported gleaned from [14].

Figs. 2 and 3 show the performances of various algorithms, such as the classification replication with the round-robin placement (CR), the classification replication with the smallest load first placement (CS), the Zipf replication with the round-robin placement (ZR), the Zipf replication with the smallest load first placement (ZS), and the IGA-based video replication and placement algorithm (IGA).

Figs. 3(a) and 3(b) depict the performance of five algorithms on rejection rate where the replication degree is 3.0 and the parameter $\theta$ is 1.0 and 0.5, respectively. The reported results of various algorithms are reported here to demonstrate the low rejection rate of the proposed method. That the solution of IGA dominates all solutions of the existing methods reveals high performance of the IGA-based approach.

Fig. 3 depicts the performance of five algorithms on load imbalance where the replication degree is 3.0 and parameter $\theta$ is 1.0. The IGA-based replication and placement algorithm performs better than the other algorithms and obtains desirable performance of load imbalance.

## 6. Conclusions

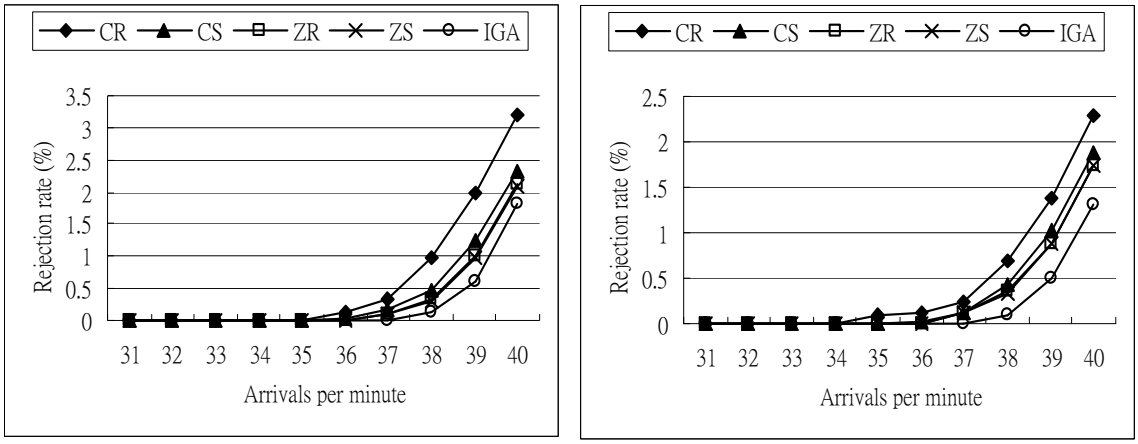
In this paper, we have proposed a method of designing an optimal video replication and placement on a cluster of VoD servers using a novel IGA with an intelligent crossover operation based on orthogonal experimental design. Since the solution space is large and complex, IGA is successfully used to solve the large parameter optimization problem. It has been shown empirically that the IGA-based replication and placement outperforms the existing non-GA- based replication and placement in terms of rejection rate and load balancing. Furthermore, IGA can be easily used without domain knowledge to efficiently design an optimal video replication and placement.

## References

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "The maximum factor queue length batching scheme for video-on-demand systems," *IEEE Trans. Computers*, vol. 50, no. 2, pp. 97-110, 2001.

[2] A. L. Chervenak, D. A. Patterson, and R. H. Katz. "Choosing the best storage system for video service." in *Proc. ACM Multimedia'95*, 1995, pp. 109-119.

[3] J. Gafsi and E. W. Biersack, "Modeling and performance comparison of reliability strategies fro distributed video servers," *IEEE Trans. Parallel and Distributed System*, vol. 11, no. 4, pp. 412-430, 2000.




(a) (b)

Fig. 2. Rejection rates of various replication and placement algorithms. (a) $\theta$ =1.0; (b) $\theta$ =0.5.

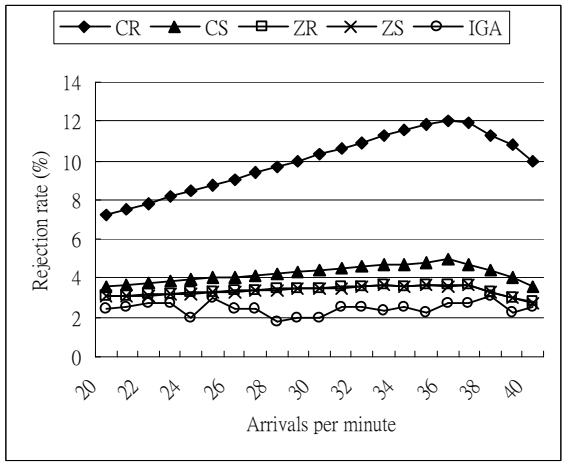



Fig. 3. Load imbalances of various replication and placement algorithms.


[4] L. Golubchik, R. R. Muntz, C. Chou, and S. Berson, "Design of fault-tolerant large-scale VoD servers: with emphasis on high-performance and low-cost," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 4, pp. 363-386, 2001.

[5] S.-Y. Ho and Y.-C. Chen, "An efficient evolutionary algorithm for accurate polygonal approximation," *Pattern Recognition,* vol. 34, pp. 2305-2317, 2001.

[6] S.-Y. Ho, C.-C. Liu, and S. Liu, "Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm," *Pattern Recognition Letters*, vol. 23, no. 13, pp. 1495-1503, 2002.

[7] H.-L. Huang and S.-Y. Ho, "Mesh optimization for surface approximation using an efficient coarse-to-fine evolutionary algorithm," *Pattern Recognition*, vol. 36, no. 5, pp. 1065-1081, 2003.

[8] Y. B. Lee and P. C. Wong, "Performance analysis of a pull-based parallel video server," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 12, pp. 1217-1231, 2000.

[9] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. Evolutionary Computation*, vol. 5, no. 1, pp. 41-53, 2001.

[10] S. H. Park, Robust Design and Analysis for Quality Engineering. Chapman & Hall, 1996.

[11] D. N. Serpanos, L. Georgiadis, and T. Bouloutas, "MMPacking: A load and storage balancing algorithm for distributed multimedia servers." *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8 no. 1 pp.1998.

[12] N. Venkatasubramanian and S. Ramanathan, "Load management in distributed video servers." in *Proc. IEEE ICDCS'97*, 1997, pp. 31-39.

[13] C. Xu and F. Lau. Load Balancing in Parallel Computers: Theory and Practice. Kluwer Academic Publishers, 1997.

[14] X. Zhou and C.-Z. Xu, "Optimal video replication and placement on a cluster of video-on-demand servers," in *Proc. IEEE ICPP'02* 2002, pp. 547-555.

[15] X. Zhou and C.-Z. Xu. "Request redirection and data layout for network traffic balancing in cluster-based video-on-demand servers," in *Proc. IEEE PDIVM Workshop*, 2002, pp. 127-134.