

第五章

效能評估 (Performance Evaluation)

本研究採用幾組模擬來評估我們所提出的路徑規劃解決方案之效能，並藉由觀察系統的獲利指標和獲利密度來瞭解演算法的效能。我們會以模擬實際網路的運作情形，比較本研究所提出的演算法和傳統的 OSPF 之差異，來說明 BBQ 架構的優點及其效能的改善。

5.1 效能評估指標 (Performance Evaluation Metrics)

以下是模擬實驗所觀察的效能指標。

1. 獲利指標：網路所允入的每一個訊務都帶有一定的獲利，藉由模擬一個參考時段所產生的訊務需求，觀察本路徑規劃演算法滿足訊務需求的能力，然後可由這些訊務所產生的利潤，來評量本演算法之好壞。
2. 鏈結使用率之標準差：藉由觀察鏈結使用率之標準差，可得知在拓撲中，每一個鏈結使用情形的差異。
3. 獲利密度：獲利除以平均的鏈結使用率即為獲利密度。由獲利密度我們可以得知，每單位鏈結使用率的獲利情形。

5.1.1 獲利指標 (Profit)

在路徑規劃演算法中，我們以訊務所要求的頻寬和服務品質作為評量訊務價值的方法，而一個訊務所要求的頻寬除以 Quality Entropy 即為此訊務的價值，在我們的路徑規劃方法中，可以容許更多的訊務獲得服務即表示所得到的獲益越大，因此我們在

改變不同的實驗變因時也觀察此一獲益指標來評估演算的效能。而在彈性的資源保留路徑規劃方案中，我們提出的以每一個鏈結的被使用率之標準差做為我們最佳化的目標，因此此一標準差指標也是我們在變化實驗變因時的觀察對象。

5.1.2 鏈結使用率之標準差 (Link Utilization S.D.)

鏈結的使用率，是由鏈結上的負載除以鏈結的能力所得，鏈結的使用率標準差代表各個鏈結使用情形的離散度，當鏈結使用率標準差的數值越大，代表各個鏈結的使用率差異越大。在同樣訊務負擔的情形下，一個路徑演算法使得鏈結使用率之標準差越大，可能的情形即為，以其演算法封包所走的路徑，將集中於幾個鏈結之上，而標準差值越小，則代表各鏈結的資源使用較平均。

5.1.3 獲利密度(Profit Density)

網路中的資源，即為每個鏈結上的頻寬，我們將總獲利除以鏈結使用率作一個觀察的指標，稱為獲利密度，鏈結使用率計算如下：將計算單一個鏈結上的被使用的頻寬除此鏈結的總頻寬，得到一個單一鏈結上的資源使用率，然後對模擬網路中所得鏈結使用率求平均，此為此組路徑所使用的鏈結使用率。因為若經過路徑規劃的結果，我們可以在越少的頻寬裡，就能獲得較高的使用率，即每單位的鏈結頻寬獲益較高，則表示其方法較佳。

5.2 實驗設計 (Design of Experiment)

由於本研究主要是在 BBQ 的整體架構中，利用預先規劃路徑的方法來模擬整體的實驗架構，因此本研究所進行的實驗中除了觀察演算法的效能外，也利用一個簡單的模擬環境來比較經過路徑規劃和傳統沒有經過路徑規劃之網路效能。

5.2.1 實驗工具

本實驗以 FreeBSD 為作業平台，在其上以 GNU C++ 3.2 實作一個本文所提出之路

徑規劃演算法和一個 Simplify BBQ Computational Environment。前者用來產生利用 PPA 元件所規劃的路徑，後者為一個簡化的 BBQ 架構 Computational Simulator，內含有簡單的 BBQ 功能元件，用來模擬 BBQ 架構的運作流程，但是為了簡化實驗流程，我們目前只有實作模擬系統運作時，訊務進入 Ingress 時所進行之允入控制和路徑指定的程序，以觀察我們路徑規劃的效能。

5.2.1.1 BBQ Computational Simulator

BBQ Computational Simulator 是本研究以 C++ 所實作的一套模擬程式，主要的功能在於模擬網路系統的運作流程和簡單的 BBQ 功能，此一模擬系統以兩種繞徑模式進行模擬，一為傳統的 OSPF 演算法，在此種模式之中，進來的訊務會依照 OSPF 的演算法來進行繞徑，另一種則會依照 BBQ 的運作程序，將欲進入系統內的訊務，經由此 Simulator 的允入控制功能，從 Ingress 的路徑資料庫中尋找一條符合訊務需求的路徑並指定給此一訊務，此一 Simulator 會詳細記錄每一個訊務所獲得的獲利及頻寬的使用，以分析整理。

5.2.2 實驗測試組產生方式

實驗組的產生包含網路拓樸和訊務兩部分。在網路型態方面，其設定的參數包括，節點數量、節點的連接率、鏈結上的頻寬、鏈結的傳輸延遲設定。而對於訊務的設定參數包括訊務進入點、離開點、需求的頻寬和所需要的服務品質參數。各參數的設定數值及範圍如表 5.1 所示。

5.2.2.1 網路拓樸之產生與設定

網路節點的數目範圍從 10 到 50，而節點的組成分為 Edge Node 和 Core Node，訊務會從某個 Edge Node 出發，並從其他 Edge 離開，而 Core Node 只有轉送封包的功能，Edge 和 Core 的比例為一比一。而一個節點連接率的定義如下，假設一個拓樸內有 N 個節點，若一個節點與其他 $N-1$ 個節點皆有鏈結連接，則此點的連接率定義為 100%，

因此一個節點的連結率為 $L/N-1$ 。

而實驗變因所指之連結率，則是將實驗網路拓樸中的每一個節點其連接率求平均值的結果，我們設定的連接率從 20% 至 100%。

5.2.2.2 訊務之產生與設定

對於路徑規劃所需的歷史訊務資料，我們假設其是以一個集合的方式呈現，每一個節點可能都有一組訊務流從此節點至其他網路內的任何一個節點。

為求模擬資料的真實性，我們採用 Network Simulator 2(NS2)，來產生 Run-time 的 Request Data，NS-2 (Network Simulator Version 2) [4,5] 是一個用來提供網路相關研究的物件導向模擬器，可提供完整、趨於真實的網路模擬環境，是個具有擴充性、可程式化事件驅動模擬引擎、支援各種網路通訊協定與路由排程演算法的網路模擬器。發展至今，已經是一個很成熟的網路模擬環境。

本實驗中所產生的每一個 request 所要求的頻寬則是呈現 normal 分佈，而每一組 request 所要求的總頻寬約為一個固定值。

表 5.1：實驗組參數設定及其範圍

參數	數值範圍
節點數目	10, 20, ..., 50
Edge 節點和 Core 節點之比	1:1
節點連接率	20%, 40%, 60%, 80%, 100%
鏈結頻寬	1G bps
鏈結延遲	5 ~ 25 ms
訊務所需頻寬	0 ~ 100 Mbps
訊務之傳送延遲限制	20 ~ 100 ms

5.2.3 實驗流程

首先將此一個 Reference Time Period 的一組歷史訊務統計資料，及一組實驗的網路拓樸，輸入路徑規劃的演算法中，經過計算得到一組規劃路徑，然後利用 NS2 產生 Run-time 訊務流。然後，將此上述的三個資料輸入 BBQ Real-Time Simulator 之中，模擬實際的 BBQ 的運作流程，藉以觀察此演算法之效能。

本實驗的對照組是為未經訊務預測和路徑規劃的 OSPF 演算法，藉以比較經過路徑規劃程序和無路徑規劃方法的優劣。

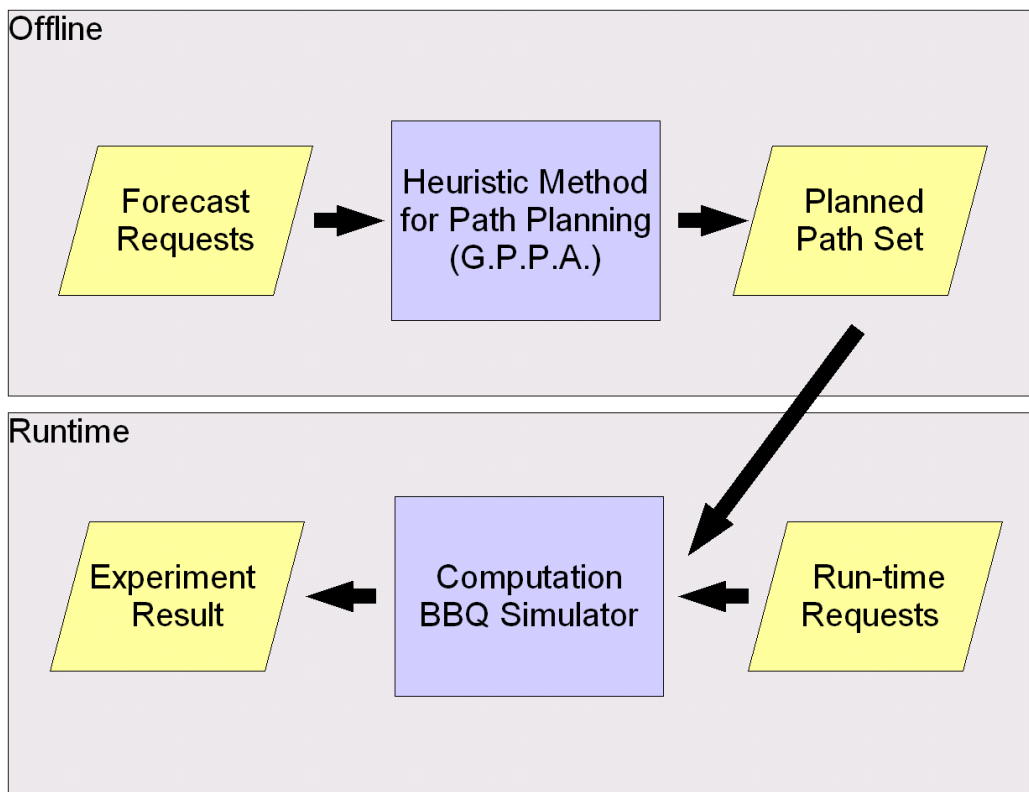


圖 5.1：實驗流程圖

5.2.4 實驗組

以下是本研究所進行之實驗：

- 實驗一的目的在於關於本演算法的行為，我們以一個小型的網路為環境，觀察本演算法與最佳解之差距。

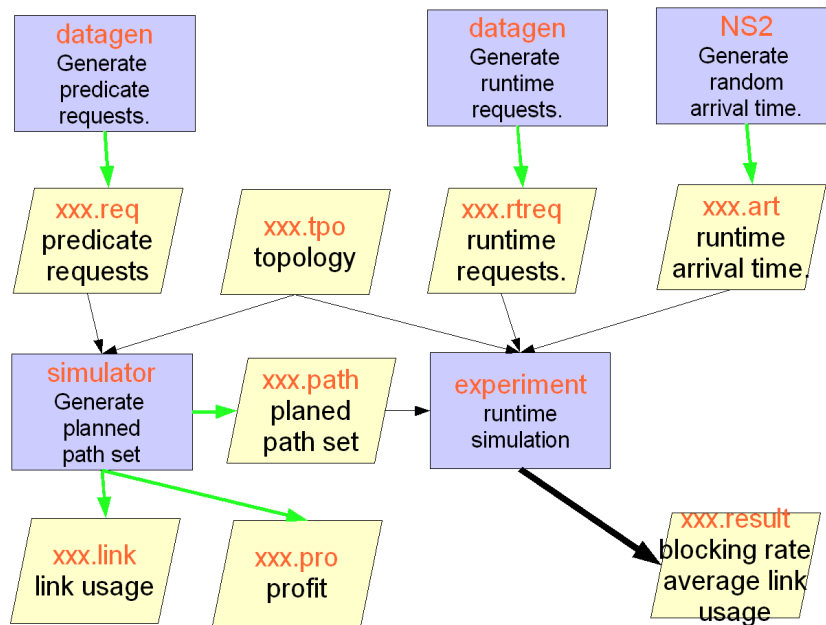


圖 5.2：實驗步驟及資料流向

- 實驗二的目的在了解網路的節點數對於本演算法效率的影響。
- 實驗三的目的在了解網路的鏈結率對於本演算法效率的影響。
- 實驗四的目的在了解預測誤差對於本演算法效率的影響。

表 5.2：效能評估實驗

實驗	目標
實驗一：演算法行為測試	<ol style="list-style-type: none"> 1. 觀察本演算法之行為 2. 衡量本演算法之效能與最佳化結果之差距
實驗二：網路節點多寡影響測試	以二個指標來觀察本演算法之效能
實驗三：網路連接率影響測試	<ol style="list-style-type: none"> 1. 獲利 (Profit)
實驗四：預測誤差影響測試	<ol style="list-style-type: none"> 2. 鏈結使用率標準差(Link Utilization S.D.) 3. 獲利與鏈結使用率之比 (P.U. Ratio)

5.3 模擬過程及結果分析

5.3.1 實驗一：演算法的行為測試

在實驗一中，我們以路徑規劃時所獲得的利潤和利潤密度來觀察演算法的行為。

5.3.1.1 獲利趨勢

本實驗以拓樸中的節點數量為實驗變因，探討當固定網路連接率時，拓樸中的節點數量對獲利之影響。同時，測試當不相同的網路連接率，觀察是否有相同的趨勢。

圖 5.3 中，橫軸為節點的數量，直軸為獲利，由圖中可看出，當節點增加時，因為

可以選擇的路徑增加，使得獲利也有提升。而當連接率的增加時，代表鏈結數目增加，同時也會增加可供選擇的繞徑因此也會提高獲利。而在網路連接率提升至在 80%至 100%時，可以發現，兩條線的趨勢幾乎是一樣的，這是因為進入的訊務數量並未跟著提昇，所以獲利並未再增加。

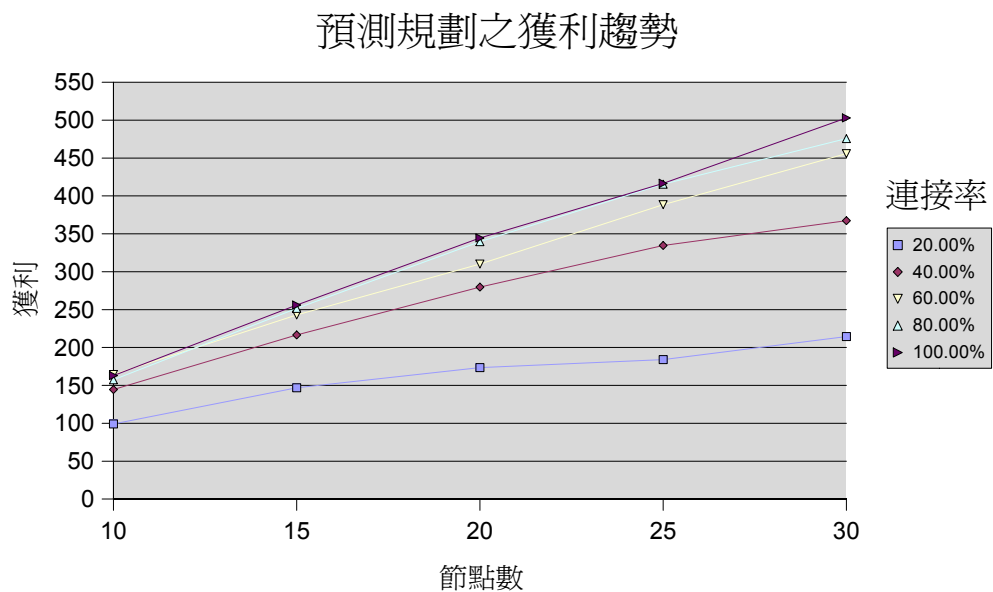


圖 5.3：預測規劃的獲利趨勢

5.3.1.2 獲利密度趨勢

如同上，本實驗以拓樸中的節點數量為實驗變因，探討當固定網路連接率時之影響，不同的是，我們將觀察拓樸中的節點數量對獲利密度之影響。同時，測試當不相同的網路連接率，觀察是否有相同的趨勢。

在圖 5.4中，在我們所觀察的是與預測規劃之利潤密度趨勢，由圖中可以看出，當節點數增加時，獲利會隨之上昇。而當連接率增加時，獲利也會隨之上昇。

預測規劃之利潤密度趨勢

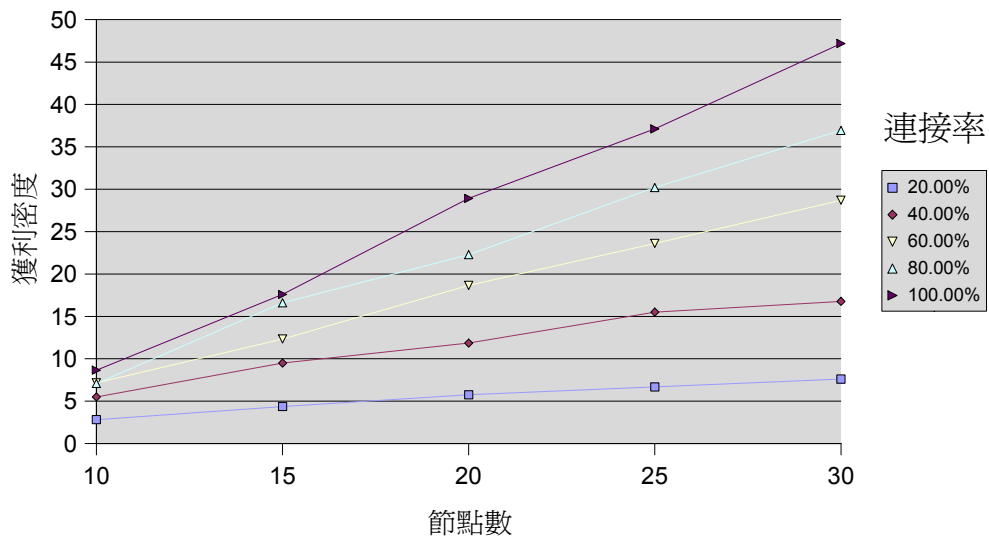


圖 5.4：預測規劃與利潤密度趨勢

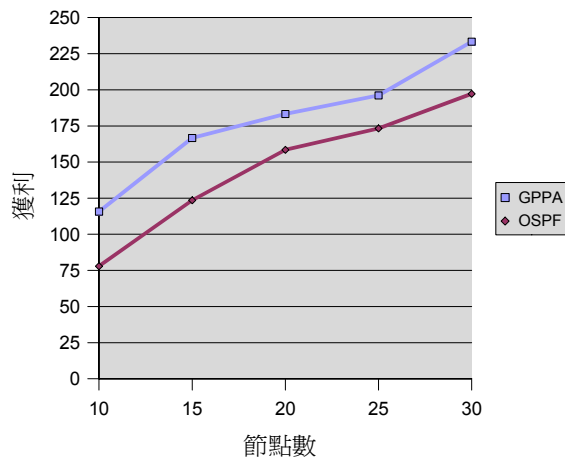
5.3.2 實驗二：節點多寡之影響 (Number of Nodes)

在此實驗中我們將觀察一個網路拓樸中節點的多寡對於演算法效能的影響。在 BBQ 所假設的網路底層架構中，是以 DiffServ 為底層的網路架構，而在 DiffServ 中，網路中的節點分為邊緣節點(Edge Router)和中間轉送的節點(Core Router)，邊緣節點是訊務進入或離開網路的節點，所以一個訊務在描述訊務時所使用的進入點和離開點，皆為網路中的邊緣節點。在本實驗中所設定的網路拓樸其 Core 和 Edge 節點的比例為一比一。

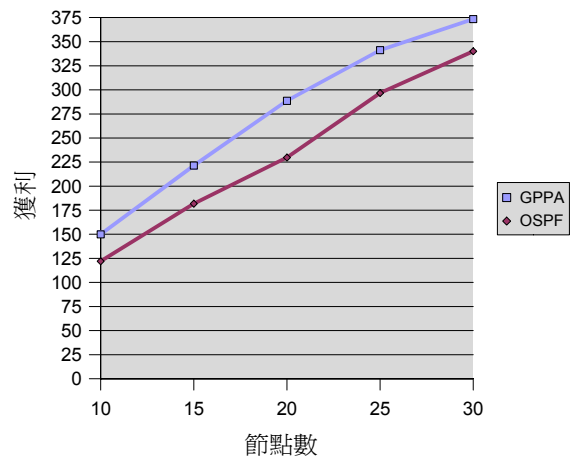
5.3.2.1 拓樸中節點多寡對於獲利指標的影響

本實驗以網路拓樸節點的多寡為實驗變因，探討當網路拓樸節點的多寡變化時對獲利之影響，本實驗設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的连接率時，觀察是否有相同的趨勢。

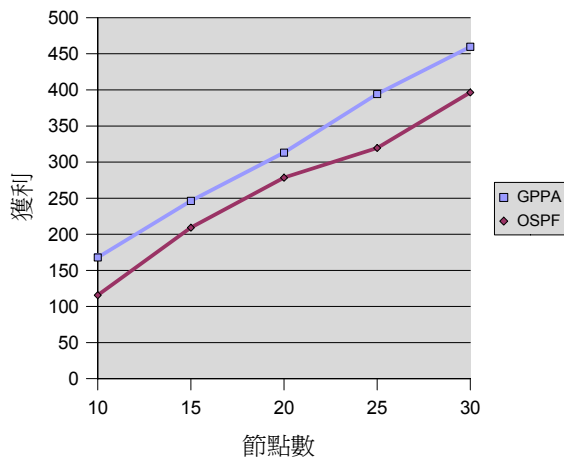
在圖 5.5(A)和圖 5.5(B)，其橫軸為節點數目的變化，直軸為獲利的變化，在圖 5.5 (A)為連接率固定為 20%時節點數目變化時，獲利情形的變化，圖 5.5(B)為連接率固定為 40%時的變化情形。我們可以從所獲得的利率來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，可以發現在節點數增加後，所獲得的獲利比 OSPF 高出許多。



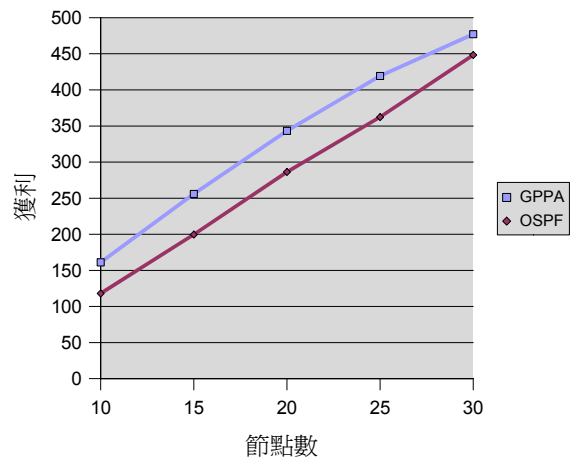
(A) 連接率為 20%



(B) 連接率為 40%



(C) 連接率為 60%



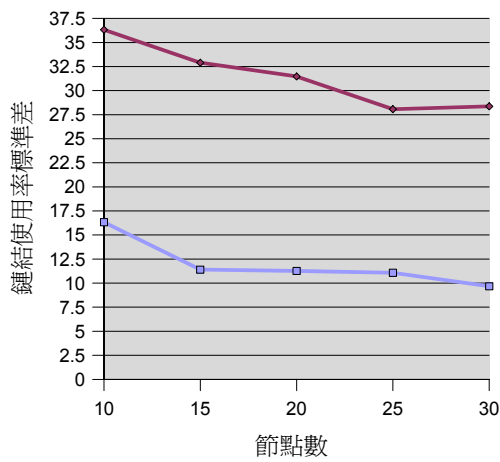
(D) 連接率為 80%

圖 5.5：節點多寡對於獲利指標的影響

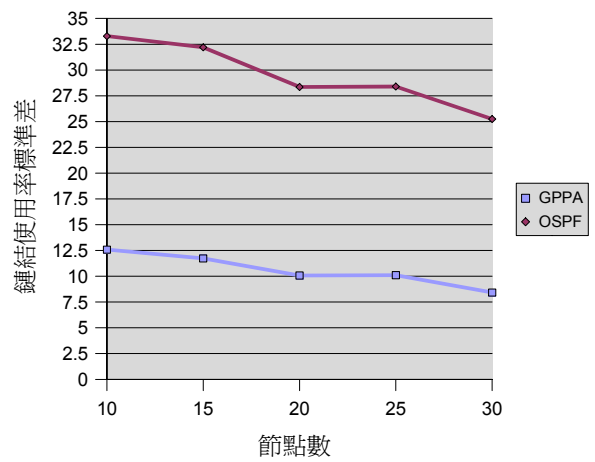
5.3.2.2 拓樸中節點多寡對於鏈結使用率標準差的影響

本實驗依然以網路拓樸節點的多寡為實驗變因，不過觀察的是當網路拓樸節點的多寡變化時對鏈結使用率標準差之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的連接率時，觀察是否有相同的趨勢。

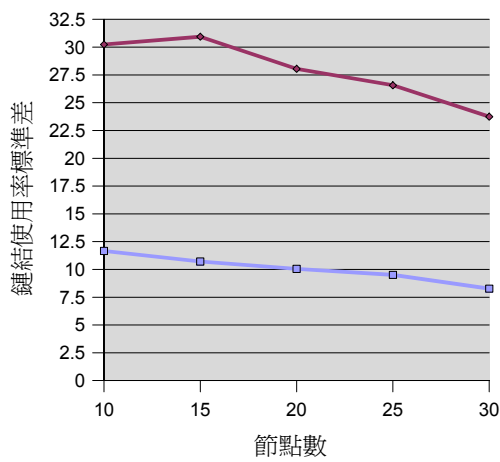
在圖 5.6(A)和圖 5.6(B)，其橫軸為節點數目的變化，直軸為鏈結使用率標準差的變化，在圖 5.6(A)為連接率固定為 20%時節點數目變化時，鏈結使用率標準差的變化，圖 5.6(B)為連接率固定為 40%時的變化情形，圖 5.6(C)和圖 5.6(D)則分別為連接率 60%和 80%的情形。我們可以從所獲得的利率來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，可以發現在節點數增加後，所獲得的獲利比 OSPF 高出許多。



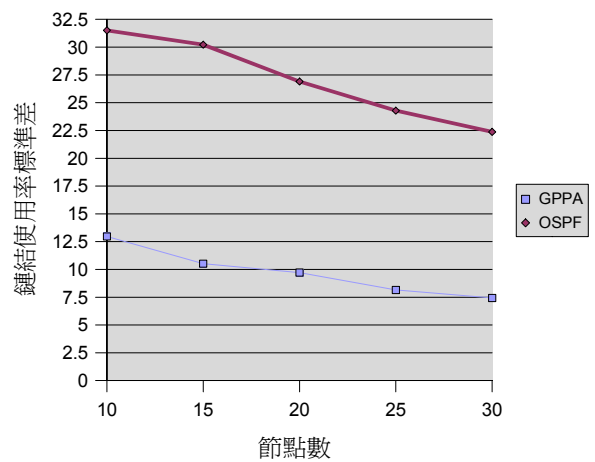
(A) 連接率為 20%



(B) 連接率為 40%



(C) 連接率為 60%



(D) 連接率為 80%

圖 5.6：節點多寡對於鏈結使用率標準差的影響

5.3.2.3 拓樸中節點多寡對於獲利密度的影響

在圖 5.7，我們觀察的是節點數目增加時，對於獲利密度的影響，當網路節點增加時，代表我們繞徑演算時所可以使用的節點越多，則可以規劃出的路徑也會相對的較多。由圖可以看出，經過路徑規劃後的獲利密度將比 OSPF 為佳，即表示本路徑規劃方

法之獲利和網路使用率之比較高，本方法可以較低的網路使用率來獲得與 OSPF 相同的獲利。

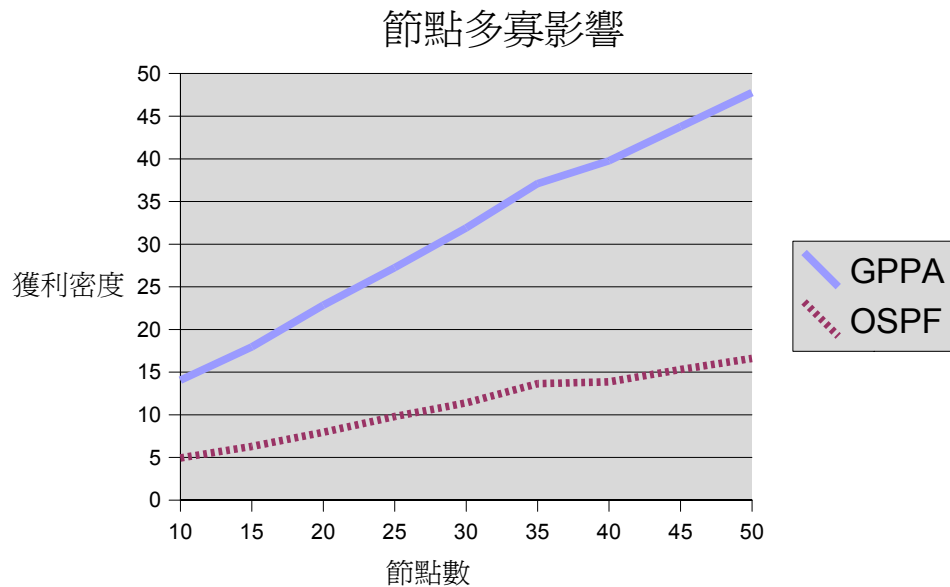


圖 5.7：節點多寡對於獲利密度的影響

5.3.3 實驗三：連接率之影響 (Connectivity)

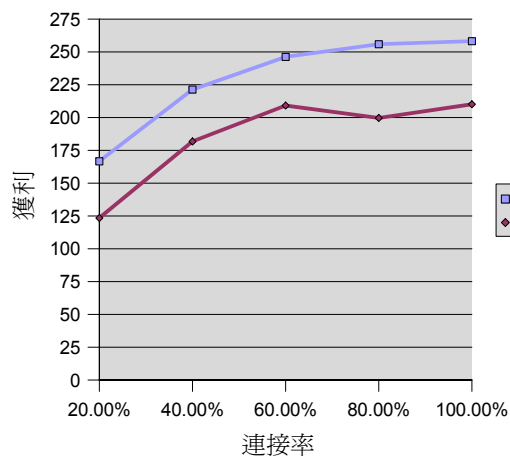
在此實驗中我們將觀察一個網路連接率大小對於演算法效能的影響。

5.3.3.1 連接率對於獲利指標的影響

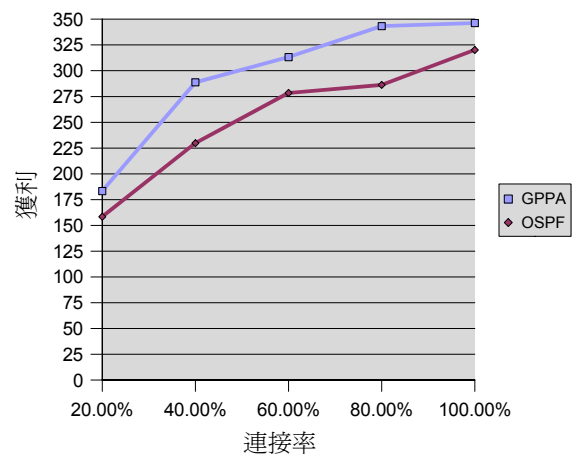
本實驗以網路連結率之大小為實驗變因，探討當網路連結率之大小變化時對於獲利之影響，本實驗設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的節點數時，觀察是否有相同的趨勢。

在圖 5.8(A)和圖 5.8(B)，其橫軸為網路連結率的變化，直軸為獲利大小，在圖 5.8 (A)為網路節點數目固定為 20 時網路連結率改變時，獲利情形之變化，圖 5.8(B)為網路節點數目固定為 25 時的變化情形。連接率的增加表示鏈結數目增加，而路徑計算時中

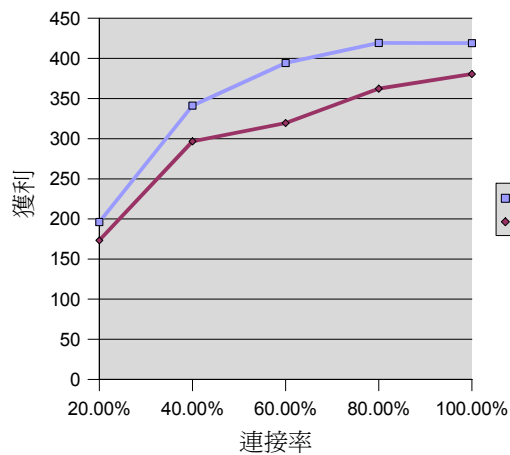
可行的路徑也是相對增加。我們可以從獲利的情形來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，在圖 5.8 中，我們可以看出，在較低的連接率時，OSPF 與本演算法的效能相差不多，但在連接率增加時，本方法獲利成長的速度是遠超過 OSPF。



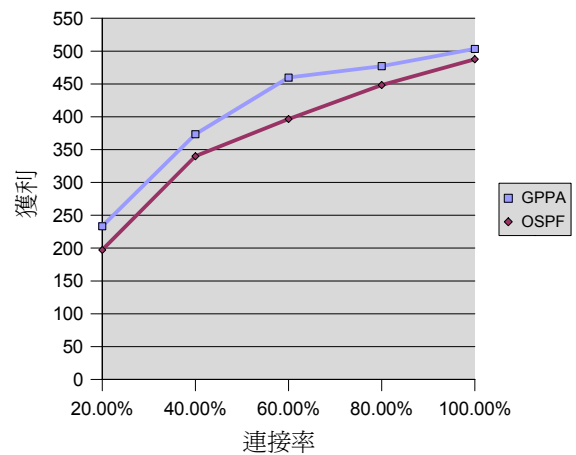
(A) 網路節點數目為 15



(B) 網路節點數目為 20



(C) 網路節點數目為 25



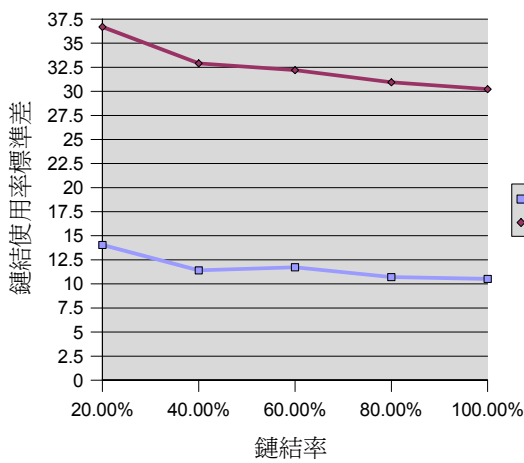
(D) 網路節點數目為 30

圖 5.8：連接率對於獲利指標的影響

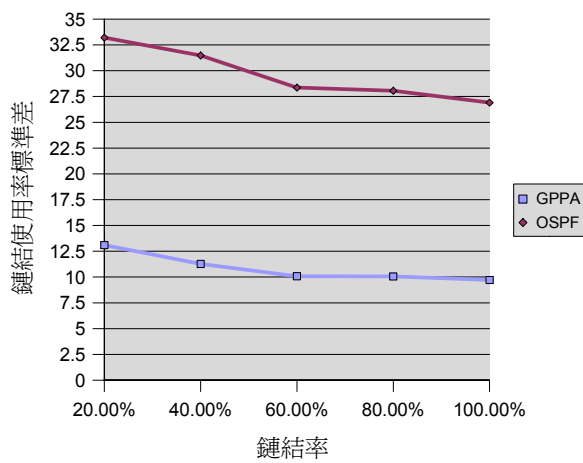
5.3.3.2 連接率對於鏈結使用率標準差的影響

本實驗仍以網路連結率之大小為實驗變因，探討當網路連結率之大小變化時對於鏈結使用率標準差之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的節點數時，觀察是否有相同的趨勢。

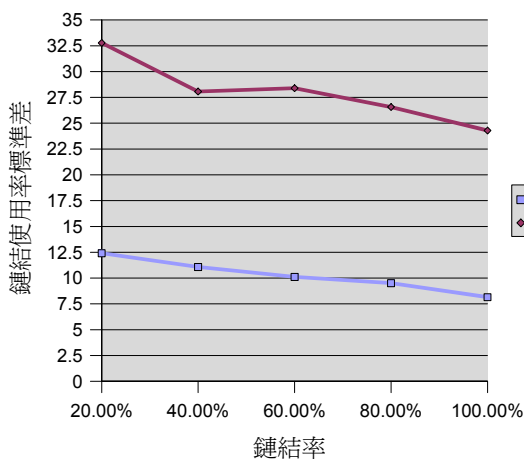
在圖 5.9 中其橫軸為網路連結率之大小，直軸為鏈結使用率標準差的變化，其中圖 5.9(A) 為網路節點數目固定為 15 時網路連結率改變時，鏈結使用率標準差之變化，圖 5.9(B) 為網路節點數目固定為 20 時的變化情形，而圖 5.9(C) 和圖 5.9(D) 則分別為節點數目為 25 和 30 時的情形。連接率的增加表示鏈結數目增加，而路徑計算時中可行的路徑也是相對增加。我們可以從鏈結使用率標準差來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，在圖 5.9 中，我們可以看出，在連接率增加時，GPPA 演算法和 OSPF 的標準差都會下降，因為可以使用的連結增加了，不過 GPPA 的標準差都與 OSPF 有一段差距，顯示 GPPA 在對於相同數量的訊務輸入時，對於網路拓樸中每個鏈結的使用率較為平均。



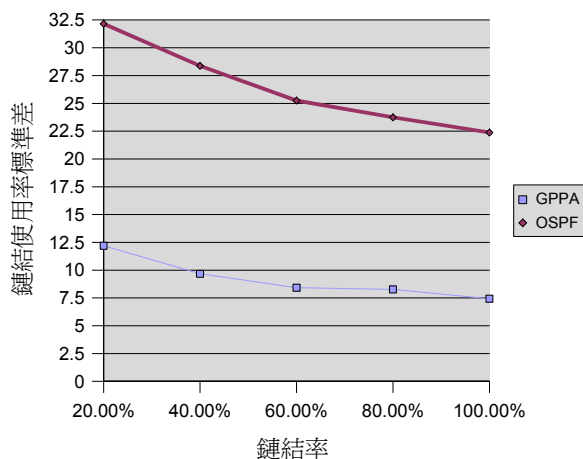
(A) 網路節點數目為 15



(B) 網路節點數目為 20



(C) 網路節點數目為 25



(D) 網路節點數目為 30

圖 5.9：鏈結率對於鏈結使用率標準差的影響

5.3.3.3 連接率對於獲利密度的影響

在圖 5.10 中其橫軸為連接率的變化，直軸為獲利密度的變化，從圖中可以觀察到在獲利密度上，本方法在連接率增加後，其比值增加的幅度是遠超的 OSPF，也代表路徑規劃演算法在鏈結率增加後，有較佳的網路利用率。

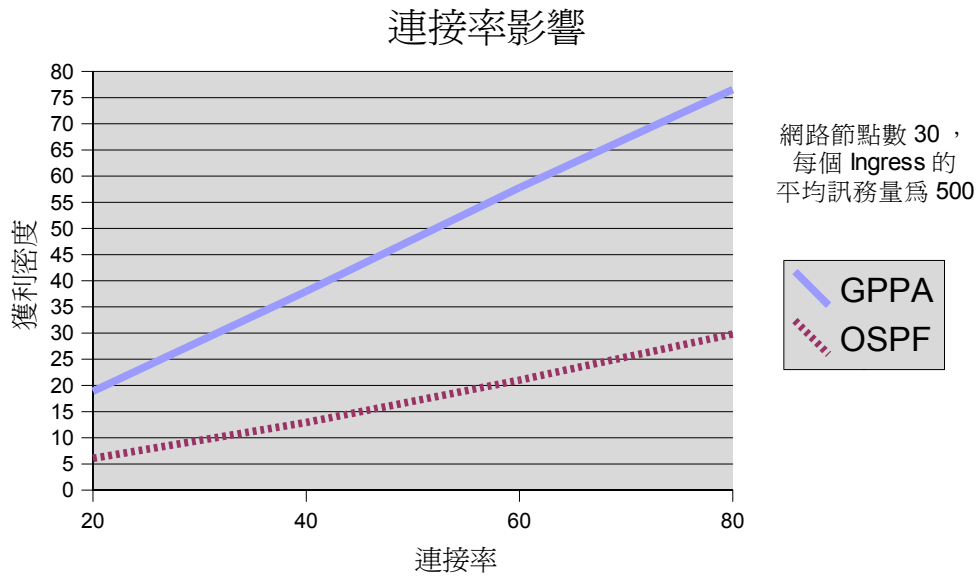


圖 5.10：連接率對於獲利密度的影響

5.3.4 實驗四：預測誤差之影響 (Forecast Error)

5.3.4.1 預測誤差之定義

由於 BBQ 中的路徑規劃方法，是以過去的歷史資料為基礎，對於未來可能出現的訊務作預先的路徑規劃，但是在系統運行的時候，會出現的訊務需求將會有很大的不確定性，因此必然會產生一定的預測誤差出現，實驗三的目的在測試當預測誤差出現時，對於本方法效能之影響。

我們以如下的方法來定義預測誤差的產生，若過去的歷史資料和系統運行時所進入網路中的訊務完全一樣的話，則預測誤差為零，若在系統運行時，欲進入網域之總訊務數量中的 10% 統計資料，和歷史的資料不合，則我們定義預測誤差為 10%，而這些欲進入網路中的訊務所要求的頻寬總和則會維持在一定範圍。

5.3.4.2 預測誤差對於獲利指標的影響

本實驗以預測誤差之大小為實驗變因，探討當預測誤差之大小變化時對於獲利之

影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。

在圖 5.11，橫軸為需求預測誤差，直軸為獲利大小。如圖所示在此實驗中，我們可以看出，當預測誤差在較小的時候，本方法仍較 OSPF 為佳，但當誤差漸漸拉大時，其影響開始出現，本方法所獲得的利益已被 OSPF 趕過，但是這情形是發生在預測誤差已在 80%、90% 以上時所產生的情形。

推測其可能的成因為，當預測誤差增加時，我們所事先規劃的路徑皆和系統運行時的訊務需求不合，因此，造成無法滿足訊務的實際需求，且會浪費的那先已經預先配置的頻寬資源，而 OSPF 採用即時的路徑規劃，因此其獲利會大概的維持一定的水準，但是在欲進入網路中的訊務個數不變時獲利並無法提升，但是本演算法的獲利卻會逐漸下降，因此造成兩者的獲利逐漸接近，甚至出現 GPPA 演算法被超過的情形，因此我們需要有預測誤差的彌補方案來減低所造成的影響。

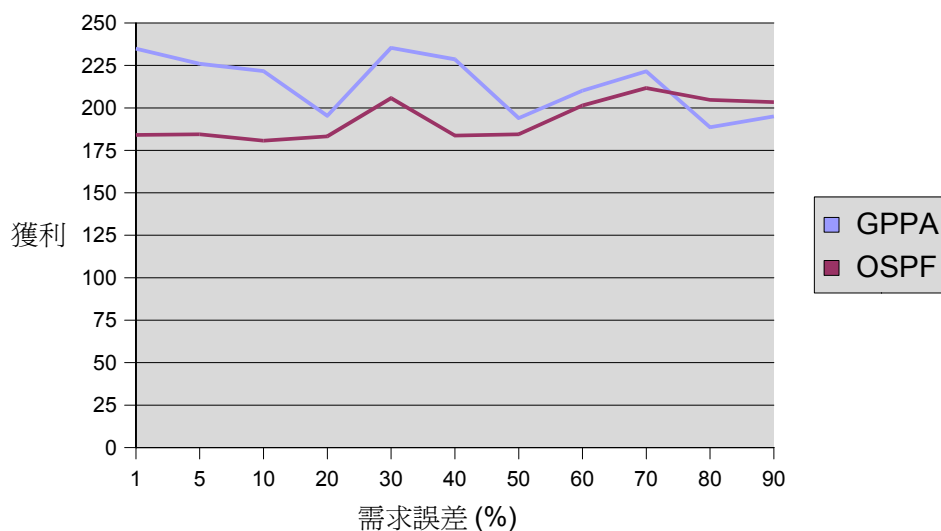


圖 5.11：預測誤差對於獲利指標的影響

5.3.4.3 預測誤差對於獲利密度的影響

本實驗仍以預測誤差之大小為實驗變因，探討當預測誤差之大小變化時對於獲利密度之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。

在圖 5.12 中，橫軸為需求預測誤差，直軸為獲利密度大小。如圖所示，預測誤差雖然使得獲得的利益可能漸漸不如 OSPF，可是就網路的使用率來說，本方法仍比 OSPF 為佳。推測其原因可能有二，其一為本演算法的將會較為分散訊務的路徑，使得鏈結的平均使用率較低，而在獲利相差不多的情形下，其獲利密度自然較高。而另一個原因為在預測誤差增加時，本演算法所能允入的訊務數較少，導致獲利下降，但是其鏈結的使用率也跟者下降，因此對於獲利密度而言並未早成太大的變化。

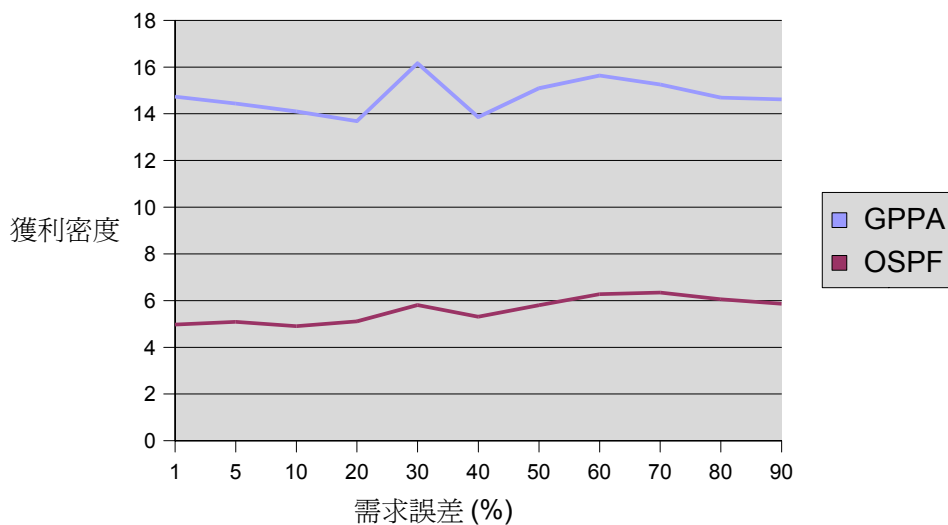


圖 5.12：預測誤差對於獲利密度的影響

5.4 結論

在此章中，我們以一個 BBQ Computational Simulator 來對我們的演算法作效能測試，並和傳統的 OSPF 作比較，我們以系統獲利、鏈結使用率之標準差及獲利密度作為效能評量的指標，並觀察節點多寡，連接率和預測誤差三種變因對於本方法之影響。

在實驗一中，我們觀察 GPPA 路徑規劃演算法在預先規劃時，拓樸中的節點數目多寡及拓樸的連接率對於系統獲利的影響，結果顯示，當節點數目及連接率增加時，皆會增加其獲利，其原因是因為當拓樸中可以使用的節點數及鏈結增加時，演算法所提供的路徑也增加，可以滿足更多的訊務需求，因此獲利增加。

在實驗二中，我們變化拓樸中的節點數目，來觀察節點數目對於本演算法效能的影響並和 OSPF 作比較，結果顯示在獲利方面，GPPA 演算法在節點數目增加時的獲利比 OSPF 高出許多，且在鏈結使用率標準差上，可以發現 GPPA 演算法會較分散訊務的對於鏈結的使用量，使得此值較 OSPF 為低。而在獲利密度上，也顯示 GPPA 在每單位的鏈結使用量的獲利會比 OSPF 為佳。

在實驗三中，我們變化拓樸中的連接率的大小，來觀察節點數目對於本演算法效能的影響並和 OSPF 作比較，結果顯示和實驗二中的結果大致相同，原因應該跟節點數目變化時相同，因此連接率的大小變化，跟節點數目的多寡對於本演算的影響是大致相同的。

在實驗四中，我們探討預測誤差對於本演算法的影響，其結果顯示，當預測誤差增加時，本演算法的弱點將顯現出來，就獲利看來，本演算法會因預測誤差的逐漸增加而導致獲利逐漸下降，甚至低於 OSPF。

從以上四個實驗中，我們可以歸納出，GPPA 的演算法的確要 OSPF 為佳，可以增加系統的獲利，但是在預測出現時，則會讓本演算法的效能退步，因此對於若要使用本演算法則要對可能出現的預測誤差有所因應。