

彈性光網路中預先預留傳送模式的虛擬網路嵌入問題

專題學生 曾天裕/蔡孟宏/江政宏/林信博/林子亨

指導老師 丁德榮

國立彰化師範大學資訊工程學系

摘要

彈性光網路(elastic optical network, EON)為未來骨幹網路的發展趨勢，因為雲端計算之普及虛擬網路嵌入(virtual network embedding, VNE)問題近年受到重視。本專題研究在彈性光網路上，預先預留(advance reservation, AR)服務虛擬網路嵌入的問題，本專題設計出多種不同的嵌入優先權的啟發式演算法(heuristic algorithm)來解決此問題，並實作驗證所提出方法的效能。透過實驗數據分析，得出路由調變等級與頻譜配置方式會比傳統的路由頻譜配置來得有效率。

1. 導論

1.1 彈性光網路(elastic optical network, EON)

彈性光網路[1]，使用光學正交多工(optical-orthogonal frequency division multiplexing, O-OFDM)技術，用於實現頻譜調變的彈性。在網路波長頻譜配置方面，傳統的分波多工(wavelength division multiplexing, WDM)網路，僅針對流量需求變化而增減所需要的波長數，而無法對分割波長或頻譜的固定網格做變動，其分割的基準固定為 100 或 150 GHz。所以對於流量較小的需求，會造成頻寬的浪費。然而在 EON 網路上，每段頻譜能夠分成較精密的寬度，最常見為 12.5 GHz；除此之

外，可以依照請求的流量來配置不同的數量的頻譜，而同個請求也必須遵守頻譜連續性；請求與請求之間，必須加上間隔頻段(guard band, GB)，才不會互相干擾。因此彈性光網路已成為未來骨幹網路的發展趨勢。

1.2 虛擬光網路(virtual network, VN)

虛擬網路是由虛擬節點(virtual nodes, VNs)和虛擬鏈路(virtual links, VLs)組成。虛擬節點是種需要計算資源(computation requirement)的節點，而虛擬鏈路則是一種兩端點間(end-to-end)需要頻譜資源的連結。一般常見的應用為公司、學校的內部網路傳輸需求。

1.3 虛擬網路嵌入(virtual network embedding, VNE)

隨著全球網路流量的急速成長，為既有的網路結構帶來巨大挑戰。為了提升現有網路的靈活性，出現了網路虛擬化的技術(network virtualization)，這也被視為下一代 Internet 架構的重要特徵。網路虛擬化技術是一種常用且有效的資源分配與共享方式。透過將實體資源(包括 CPU 計算能力、記憶體、儲存空間、網路頻寬等)虛擬化，進而達成小至單一電腦資源分割，大至跨網路、跨資訊中心的資源共享與分配的功能。基於上述網路虛擬化概念，近幾年整合行動網路、有線網路、多媒體網路等各式的網路的虛擬化需求受到重視[2,5]，相關產品也陸續推出。其中虛擬網路嵌入(virtual network embedding, VNE)[2,5]問題的輸入包含兩個部分，一個是實體網路(physical network or substrate network)，另一個則是虛擬網路(VN)。如圖 1 所示，下層為實體網路，上層為虛擬網路。在實體網路運算資源足夠的前提下，實體節點或實體鏈路可以被不同的虛擬網路(VN_1~VN_n)所分享。

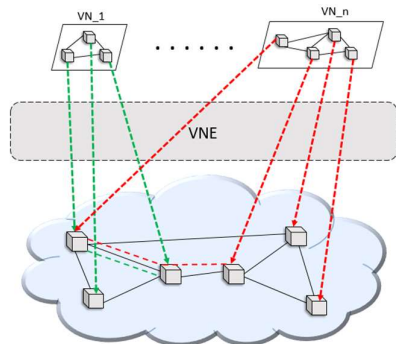


圖 1. 虛擬網路嵌入實體網路示意圖[5]

虛擬網路嵌入問題主要考慮將上層的虛擬網路需求對應到實體網路資源。更精確而言，對於每一個虛擬網路的建立需求，VNE 負責將虛擬節點對應到實體網路節點，並將虛擬鏈路對應到實體網路上的一或多條傳輸路徑並配置頻寬。然而，在一般網路上的 VNE 問題已被證明為 NP-hard 問題[2]。許多的文獻研究 VNE 問題的不同變形與衍伸的問題，相關文獻可以參考論文[2,5]。

1.4 預先預留(advance reservation, AR)

相較於傳統的立即預留(immediate reservation, IR)服務相比，預先預留(advance reservation, AR)服務[3]要求用戶為其請求提供一些與時間相關的參數，即期望的到達/離開時間，持續時間(holding time)等。從概念上講，IR 和 AR 服務之間的主要差異是請求到達和服務開始之間的時間(如圖 2 與 圖 3)，稱為預訂時間。在這方面，IR 可以被歸類為“零預訂”AR 服務。此外，IR 請求的保留時間通常不會提前知道或假設為無窮。當需求到達後立即提供服務。相比之下，大多數 AR 服務保持時間是預先知道的。這允許運營商實現更高效的資源配置。一般而言，靈活的請求計時可以為大型文件傳輸提供較佳的支持達到更好的網路資源使用，但相對的也提高問題的解決難度，因為必須要同時考量時間與頻譜資源的多為空間管理與分配問題。

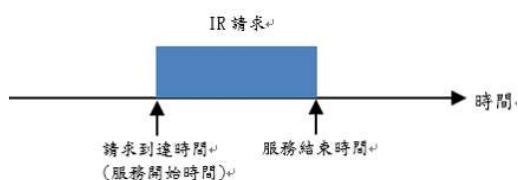


圖 2. IR 請求示意圖

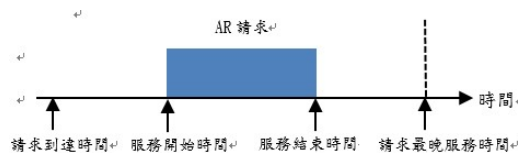


圖 3. AR 請求示意圖

在 AR 的需求中，需求到達時並不需要立即提供服務，通常 AR 會描述期望被服務的時間、服務的持續時間與最後被服務結束的時間，

只要能夠符合需求的限制，管理者可以依據網路資源做出最佳的規劃。

1.5 路由調變等級與頻譜配置

EON 網路上主要的問題為路由與頻譜配置(routing and spectrum assignment, RSA)問題，對於從來源節點 s 到目的節點 d 所需要頻寬為 BW (Gbps)的傳輸需求，記為 (s, d, BW) ，必需能夠找到來源節點與目的節點之間的傳輸路徑與可用的頻寬。於配置網路資源時，RSA 問題需遵守頻譜連續性限制(spectral continuity)和頻譜相鄰性(spectrum contiguity)限制[1]。但在 RSA 問題中，並未考慮傳輸路徑的距離。而較複雜的且實際的路由調變等級與頻譜配置(routing, modulation level and spectrum assignment, RMLSA)問題加入考慮傳輸距離與調變等級(modulation level)[4]；而調變等級的使用與距離有關，距離越大，使用的調變等級越低，反之亦然。常見的調變格式為 BPSK (binary phase shift keying)，QPSK (quadrature phase shift keying)，8QAM (8 quadrature amplitude modulation)，其最大傳送距離限制與每一個頻率槽可以傳送的資料率大約為(4000 km, 12.5 Gb/s)，(2000 km, 25 Gb/s)，(1000 km, 37.5 Gb/s)。若一路徑其傳輸距離為 1800km，故可以使用 QPSK 的調變，每個頻率槽可以傳送 25Gb/s，相較於另一光徑 2500km 時只能使用 BPSK 格式調變，所需要的頻譜槽數相差近一倍。相較於 RSA 的問題，EON 網路的 RMLSA 問題更加彈性且複雜，也已被證明為 NP-hard 問題。

1.6 研究動機

在先前的研究之中，有不少論文考慮彈性光網路上的虛擬網路嵌入問題當作主題[2,5]，其中有些會考慮到調變格式與距離限制，有些會考慮預先預留(AR)的問題。除此之外，文獻[2]中的作者討論 transparent VNE 與 opaque VNE 的差異[2]。其中 **Transparent VNE**：要求同一個虛擬網路上所有的虛擬鏈路所對應到的

實體鏈路的頻譜區段為連續且相同，並假設每一個虛擬鏈路所需要的頻寬均相同。而 **Opaque VNE**：對於單一虛擬鏈路所對應到的實體鏈路的頻譜區段需要為連續，虛擬鏈路所需要的頻寬可以不同。作者[2]針對這兩種問題均提出整數線性規劃 (integer linear programming, ILP) 公式來定義問題，並設計啟發式演算法。於 Opaque VNE 問題中，作者考慮調變格式與最大傳輸距離的限制 (RMLSA 模式)，並假設路徑若確定，則選擇滿足最大傳送距離限制且需要頻譜槽數最小的最佳的調變格式來傳送。所設計的演算法分成兩個階段，先處理節點對應再處理鏈路對應。

在論文[5]中，作者將傳送器的調變格式納入考慮，考慮 Opaque VNE 問題，目標在使得對應所有虛擬網路需求所需要的對應成本最少，作者利用 ILP 公式定義問題，並設計三個啟發式演算法來解決問題。在論文[8]中，考慮 Opaque VNE 問題，對於靜態多個 VNE 需求，作者以 ILP 模式定義，並提出碎片感知的 (fragmentation-aware) 對應演算法。在論文[9]中，針對 Transparent VNE 模式，作者同時考慮能源消耗與頻譜資源配置時的最佳化問題。

依據本組間成員的文獻分析，目前並沒有整合考慮彈性光網路上預先預留的虛擬網路嵌入問題，因此我們以 opaque VNE 為基底，加入了調變等級以及 AR 作為本研究的研究主題。本研究預計研究在彈性光網路上，預先預留服務虛擬網路嵌入的問題，預計設計出多種不同的嵌入優先權的啟發式演算法來解決此問題，並實驗驗證所提出方法的效能。

2. 研究問題與定義

2.1 研究問題

給定實體網路與一些虛擬網路嵌入需求，假設每個虛擬網路需求都有各自要被服務的時限區間 (time period)。考量實體網路中不同時間區間的運算能力及可用頻譜資源，來進行虛擬網路的嵌入，並透過不同的優先權來決定嵌入順序，以期能夠減少需求的阻塞率。最後利用整個系統的阻塞率、連續破碎率、執行時間來

評估不同演算法的效能。

2.2 假設

研究的問題假設如下：對於彈性光網路上，每個鏈路都有一條連接端節點的光纖，而信號能雙向傳輸，所有節點都配備了帶寬可變波長交叉連接器 (bandwidth-variable wavelength cross-connects, BVWXC)[5]。為了研究需要，每條鏈路所提供的頻譜槽 (frequency slots, FSs) 數量皆相同。在節點之間虛擬連線的頻寬需求可以透過使用相同或不同的路由及數個頻譜槽的光路徑來傳輸。

2.3 限制

在彈性光網路中，對於給定的光路徑所分配到的每條鏈路間，其所使用的頻譜槽位置必須相同，這稱為頻譜連續性限制 (spectrum continuity constraint)[1]。而傳送同一資料的子載波必須沿頻域連續，稱為子載波連續性限制 (subcarrier consecutiveness constraint)，且用於路徑分配的頻譜槽，彼此間需要以保護頻寬 (guard band) 隔開，來防止訊號干擾。另外，同一鏈路中所使用的頻譜槽必須為連續的，這稱為頻譜相鄰性限制 (spectral contiguity constraint)[1]。

2.4 研究目的

本專題的研究目的在彈性光網路的實體網路上處理具有預先預留服務的虛擬網路嵌入 (VNE) 問題。在嵌入過程中包含了虛擬節點對應 (node mapping) 與虛擬鏈路對應 (link mapping) 兩階段。對應時需要考慮到虛擬節點的計算容量需求與各實體節點所能提供的運算處理能力；鏈路對應時，需要考慮到每條實體線路剩餘的可用頻譜與碎片率 (fragmentation ratio) 來排定虛擬網路的嵌入順序，並透過分析最終系統的阻塞率 (blocking ratio)、連續破碎率，以及執行時間來比較不同演算法的優劣。

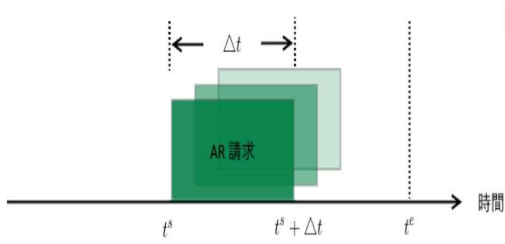


圖 4. AR 請求模擬圖

2.5 符號定義

問題的符號定義如下表 1，在 AR 請求模擬中的符號使用請參考圖 4。

表 1. 問題的符號定義

$G^s(V^s, E^s)$	實體網路
G^{temp}	實體網路複製圖
$G^r(V^r, E^r)$	虛擬網路拓模
$v^s \in V^s$	實體節點
$v^r \in V^r$	虛擬節點
$r(G^r, t^a, t^s, \Delta t, t^e)$	虛擬網路請求
t^a	請求到達時間
t^s	請求可開始執行時間
Δt	請求執行所需時間
t^e	請求被執行底線時間
$bw_e^r, e \in E^r$	虛擬鏈路 e 所需的頻寬
$C_i^r, i \in V^r$	虛擬節點 i 所需容量
$C_i^s, i \in V^s$	實體節點 i 的容量
g	實際使用的頻譜槽數量
FS	鏈路中總頻譜槽數
M	路徑使用的調變等級
$rem_v(t)$	在時間點 t 節點的剩餘容量
C^{min}	實體節點在時間區段中的最小容量
GB	保護頻帶所使用的頻率槽數量(通常為 1)
$\alpha, (0 \leq \alpha \leq 1)$	調整用參數

2.6 公式

2.6.1 計算的調變等級

將利用 k-shortest path (KSP)演算法找出的所有可能節點對之間的候選路徑，路徑依其距離由小到大排序，並假設 s 到 d 第 k 條的路徑 p_{sd}^k 長度為 $D(p_{sd}^k)$ 。將該路徑的調變等級 M 設定為 $M = mlvl(D(p_{sd}^k))$ ，其中 $mlvl()$ 回傳傳輸距離可以支援的最高調變等級。

2.6.2 計算實際使用的頻譜數量

計算出調變等級 M 之後[6]，即可算出每一個虛擬鏈路的頻寬需求 bw_e^r (Gbps) 實際所需要配置的頻譜數量：

$$g = \lceil \frac{bw_e^r}{12.5M} \rceil + GB$$

2.6.3 阻塞率(blocking ratio, BR)

當需求的工作路徑或備用路徑在分配頻率槽時，若因頻譜連續性、頻譜相鄰性、保護頻寬等限制而無法配置的連線將被阻塞。定義 **阻塞率 = 阻塞數 / 總需求數**。

3. 研究方法

因為本專題探討的主題為 NP-complete 問題，所以我們提出啟發式演算法來解決這個問題，在演算法中依所採用的優先順序，可以有種不同的變化。以下介紹所提出的演算法。

3.1 區域資源容量(local resource capacity, LRC)

節點對應可以藉由區域資源容量(local resource capacity, LRC)[5]的幫助來執行，LRC 分成虛擬節點以及實體節點。

- (1) 虛擬節點 LRC：虛擬節點 $a \in V^r$ ($L^v(a)$ 表示為虛擬節點 a 的 LRC) 需要計算容量乘以所有與該節點連結虛擬鏈路所需頻寬(bw_e^r)的總和。

$$L^v(a) = c_a^r \times \sum_{(a,b) \in E^r} bw_{(a,b)}^r$$

假設有一個五個節點的虛擬網路，如圖 5。為了要計算節點 a 的 LRC，我們將節點 a 的容量乘以所有和節點 a 連結的虛擬鏈路的頻寬總和。

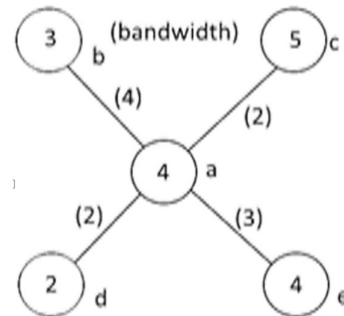


圖 5. 虛擬節點 LRC 示意圖

$$L^v(a) = 4 \times (4 + 2 + 3 + 2) = 44$$

- (2) 實體節點 LRC：實體節點 $a \in V^s$ ($L^s(a)$ 表示為實體節點 a 的 LRC) 的計算容量乘以節點分支度數 (計成 $\text{degree}(a)$)。

$$L^s(a) = c_a^s \times \text{degree}(a)$$

假設有一個五個節點的實體網路，如圖 6。為了要計算節點 a 的 LRC，我們將節點 a 的容量乘上節點 a 的度數，即

$$L^s(a) = 4 \times 4 = 16$$

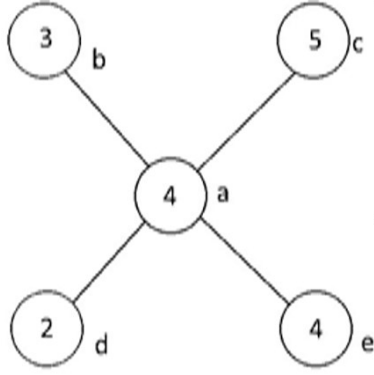


圖 6. 實體節點 LRC 示意圖

3.2 請求處理優先權公式

當有多筆請求待處理時，需使用下列

公式判定處理的優先順序。

$$P(x) = \alpha L(x) - (1 - \alpha) T(x)$$

權重值越大的請求優先處理，其中

$L(x)$ 為請求 x 在考量 LRC 條件下的權重而

$T(x)$ 為請求 x 在考量 t^e 條件下的權重

$L(x)$ 與 $T(x)$ 之值均經過正規化介於 0 到 1 之間，並透過調整 α 值來形成不同的演算法組合。

說明： α 為 0 時表示 $P(x)$ 只受到 $T(x)$ 影響， α 為 0.5 時表示 $P(x)$ 均勻的受到 $L(x)$ 和 $T(x)$ 的影響， α 為 1 則表示 $P(x)$ 只受到 $L(x)$ 影響。

3.3 研究步驟

我們將嵌入過程分成節點對應 (node mapping)

和鏈路對應 (link mapping) 兩個階段，分述如下：

3.2.1 Node Mapping

- (1) 先分別以 LRC 和需求容量決定虛擬節點嵌入的順序，LRC 或需求容量越高優先對應。

- (2) 將實體網路圖複製，並把複製圖中在該請求要求的時間區段裡，無法提供足夠容量的實體節點刪除。
- (3) 以實體節點的 LRC 來當作挑選被嵌入節點的順序，若節點已經被使用則跳過。
- (4) 如果有任一虛擬節點配置失敗，則將請求開始與結束時間往後延一個單位時間，再重複以上的對應，結束時間最多延至請求給的時間底線 t^e 。若節點對應成功則開始鏈路對應階段，否則該需求阻斷。

3.2.2 Link Mapping

- (1) 以頻寬需求的多寡來決定虛擬鏈路嵌入的順序，頻寬越高優先對應。
- (2) 選定虛擬鏈路後，使用 KSP 演算法產生此虛擬鏈路可能的候選實體路徑。
- (3) 依序選定路徑，根據路徑長度決定可以使用的最高調變等級，以及計算利用該路徑傳送時所需要實際使用的頻寬數量。
- (4) 因為頻譜連續性限制，該路徑不同鏈路中的頻譜位置必須相同，若現有彈性光網路上可以找到符合條件的路徑則該路虛擬路徑嵌入成功。
- (5) 若所有 k 條候選路徑皆無法配置，則先釋放所有配置的節點與鏈路對應資源，則將請求開始與結束時間往後延一個單位時間，再重複以上的對應，重新執行節點對應在執行鏈路對應，最多延至請求給的時間底線。若所有虛擬鏈路均可以對應成功，則配置資源，否則該虛擬網路需求阻斷。

3.4 演算法

以下將說明本論文提出演算法各階段之虛擬碼。

演算法 1: 頻譜優先演算法 (FS_First)

(流程圖參見附錄)

輸入：實體網路 $G(V^s, E^s)$ ，多筆虛擬請求 $r(G^r, t^s, t^e, \Delta t, t^e)$

輸出：VNE 嵌入結果

根據虛擬優先權排序選定一虛擬請求 $r(G^r, t^s, t^e, \Delta t, t^e)$

```

計算每個虛擬節點  $v^r$  的 LRC;
for (在開始和結束時間底線[  $t^s$ ,  $t^e$  ]中的所有時間區間  $\Delta t$ )
{
    X = True;
    for (根據虛擬節點  $v^r$  的 LRC 或需求容量排序後依序嵌入)
    {
        建立實體網路複製圖  $G^{temp}$ ;
        移除複製圖中在時間區間內容量不足的節點;
        計算每個實體節點  $v^s$  的 LRC;
        將未使用的實體節點  $v^s$  以 LRC 排序;
        將虛擬節點  $v^r$  嵌入擁有最大 LRC 的實體節點  $v^s$ ;
        標記實體節點  $v^s$  被使用過;
        if (嵌入失敗)
        {
            if( $t^s + \Delta t = t^e$ )
                return 請求  $r$  嵌入失敗;
            else
                X = False;
                break ;
            }
        }
        if(X = False)
            continue;
        for (根據每一條虛擬鏈路的頻寬大小依序嵌入)
        {
            for (i = 1 到 K) /* k shortest path 演算法找出 K 條路徑*/
            {
                計算第 i 條路徑可以使用的最高調變等級  $M = mlvl(D(p_{sd}^k))$ ;
                計算實際使用的頻譜  $g = \lfloor \frac{bw_r^T}{12.5M} \rfloor + G$ 
                Y = False;
                for (實體鏈路中所有的頻譜位置)
                {
                    if (在路徑上所有的實體鏈路在時間區間內頻譜槽從 i 到 i+g-1 都未使用)
                        標記這個 link
                    mapping 成功;
                    Y = True;
                    break ;
                }
                if (Y = True)
                    break ;
            }
            if (第 K 條路徑失敗)
            {
                if( $t^s + \Delta t = t^e$ )
                    return 請求  $r$  嵌入失敗;
                else
                    X = False;
                    break ;
                }
            }
        }
        if (X = False)
            continue;
        else
            return 請求  $r$  嵌入成功;
    }
}

```

演算法 2: 路徑優先演算法(Path_First)

(流程圖參見附錄)

輸入 : 實體網路 $G^s(V^s, E^s)$, 多筆虛擬請求 $r(G^r, t^s, t^e, \Delta t, t^e)$

輸出 : VNE 嵌入結果

根據虛擬優先權排序選定一虛擬請求 $r(G^r, t^s, t^e, \Delta t, t^e)$

計算每個虛擬節點 v^r 的 LRC;

for (在開始和結束時間底線[t^s , t^e]中的所有時間區間 Δt)

```

{
    X = True;
    for (根據虛擬節點  $v^r$  的 LRC 或需求容量排序後依序嵌入)
    {
        建立實體網路複製圖  $G^{temp}$ ;
        移除複製圖中在時間區間內容量不足的節點;

```

```

計算每個實體節點  $v^s$  的 LRC;
將未使用的實體節點  $v^s$  以 LRC 排序;
將虛擬節點  $v^r$  嵌入擁有最大 LRC 的實體節點  $v^s$ ;
標記實體節點  $v^s$  被使用過;
if (嵌入失敗)
{
    if( $t^s + \Delta t = t^e$ )
        return 請求  $r$  嵌入失敗;
    else
        X = False;
        break ;
    }
}
if(X = False)
    continue;
for (根據每一條虛擬鏈路的頻寬大小依序嵌入)
{
    for (實體鏈路中所有的頻譜位置)
    {
        for (i = 1 到 K) /* k shortest path 演算法找出 K 條路徑*/
        {
            計算第 i 條路徑可以使用的最高調變等級  $M = mlvl(D(p_{sd}^k))$ ;
            計算實際使用的頻譜  $g = \lfloor \frac{bw_r^T}{12.5M} \rfloor + G$ 
            Y = False;
            if (在路徑上所有的實體鏈路在時間區間內頻譜槽從 i 到 i+g-1 都未使用)
                標記這個 link
            mapping 成功;
            Y = True;
            break ;
        }
        if (Y = True)
            break ;
    }
    if (Y = False)
    {
        if( $t^s + \Delta t = t^e$ )
            return 請求  $r$  嵌入失敗;
        else
            X = False;
            break ;
        }
    }
    if (X = False)
        continue;
    else
        return 請求  $r$  嵌入成功;
}

```

4. 實驗結果

4.1 實驗規劃

我們使用 C++ 程式語言來開發所提出來的演算法，包含模擬實體網路和連線需求。並應用 Boost graph library (BGL) 函式庫[7]。BGL 為 C++ Boost 函式庫的成員之一。BGL 廣泛了運用泛型程式的概念，使其用任意資料結構所實現的圖容器能夠應

用在正確實作其使用者介面的任意演算法上。BGL 並提供了一個快速且具有高執行效率的處理圖論問題的解決方案。實驗環境敘述如下：Intel core i7-7700hQ CPU 2.80 GHz、8G RAM、Windows 10 pro 64bit 平台的桌上型電腦下實驗模擬，網路採用 NSFNET topology (14 nodes)之網路(圖 7)。

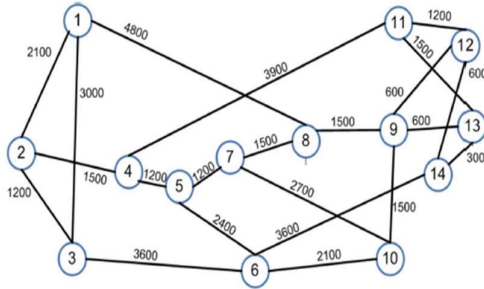


圖 7. 實驗用網路拓模

實驗資料的產生是透過自行設計的測資產生器隨機產生數筆需求；每筆需求中的節點計算能力上限有 50、100、150 三大類，而頻寬需求的上限值為 250Gbps，皆為產生器產生之亂數。

4.2 評估標準

評估效能部分包含

- (1)阻塞率(Blocking ratio, BR)：當需求的工作路徑或備用路徑在分配頻率槽時，若因頻譜連續性、頻譜相鄰性、保護頻帶等限制而無法成功配置時，這條連線將被阻塞。
定義阻塞率 = 阻塞數/總需求數。
- (2)執行所耗費時間。
- (3)阻斷原因為計算能力不足或頻譜不足。
- (4)考慮調變等級與否的效能差異，也就是 RMLSA 與 RSA 傳輸模式實驗結果的比較。

4.3 實驗結果

首先我們定義點嵌入採用虛擬節點的計算容量做判定加上路徑優先的演算法稱為 CAP_PATH，而若是節點計算容量搭配頻譜優先的演算法則稱為 CAP_FR。點嵌入採用虛擬節點的 LRC 做判定加上路徑優先的演算法稱

為 LRC_PATH，而 LRC 搭配頻譜優先則是稱為 LRC_FR。首先我們先進行前置實驗找出各組演算法的最佳 α 值(圖 8)，獲得表 2 之結果。

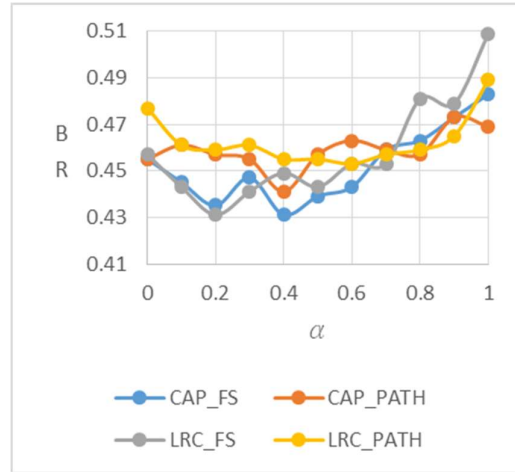


圖 8. 不同 α 值之阻塞率比較

表 2. 各組演算法其最佳 α 值

演算法	α
CAP PATH	0.43
CAP FR	0.24
LRC PATH	0.48
LRC FR	0.36

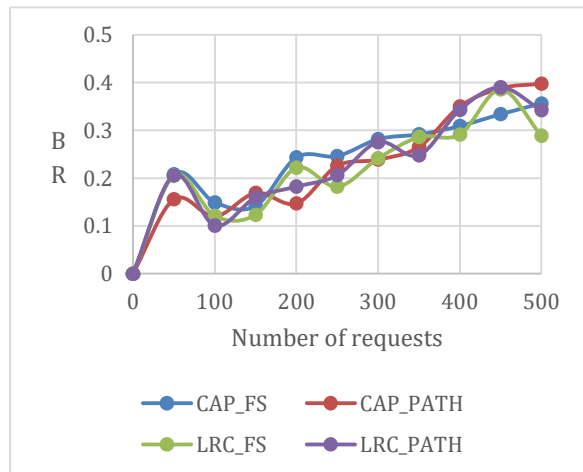


圖 9. RSA 模式(50)

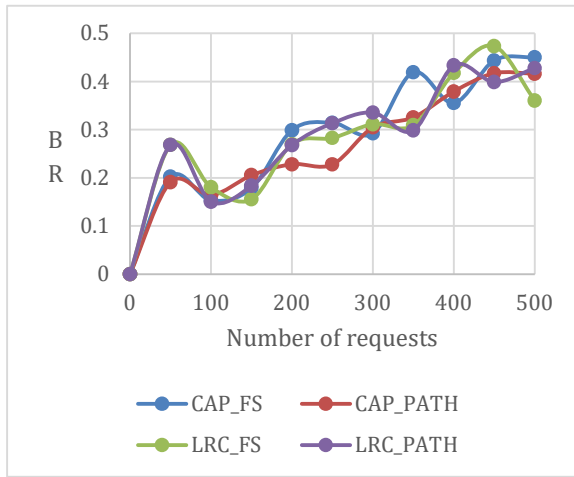


圖 10. RSA 模式(100)

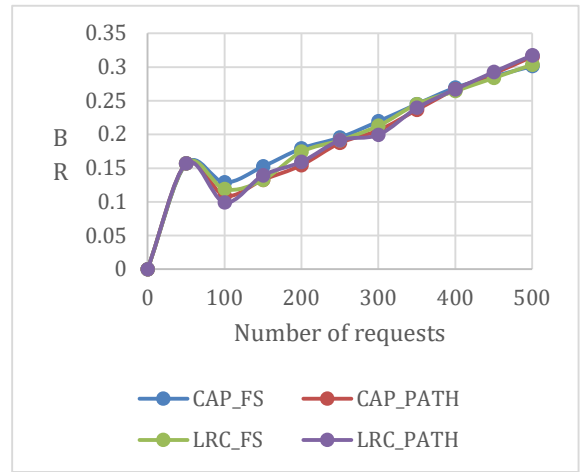


圖 13. RMLSA 模式(100)

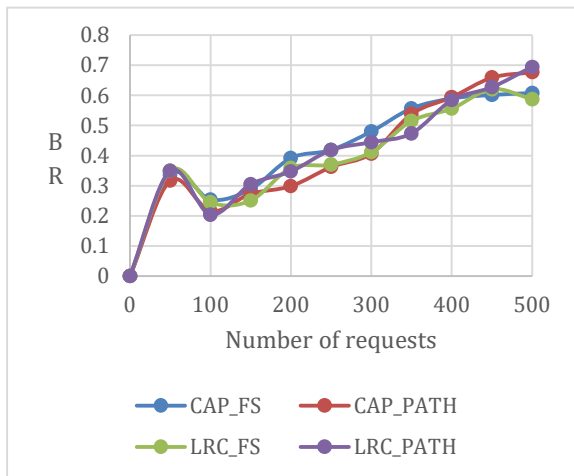


圖 11. RSA 模式(150)

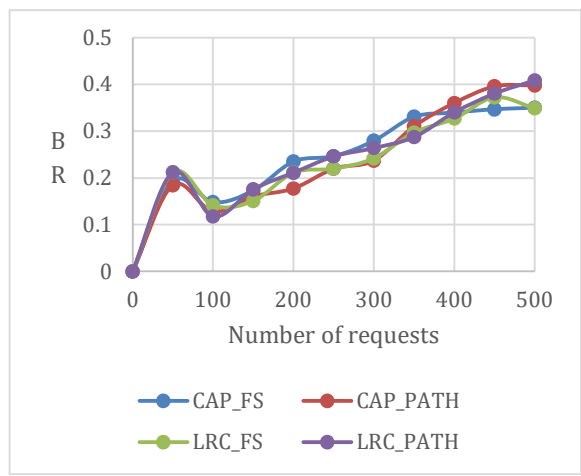


圖 14. RMLSA 模式(150)

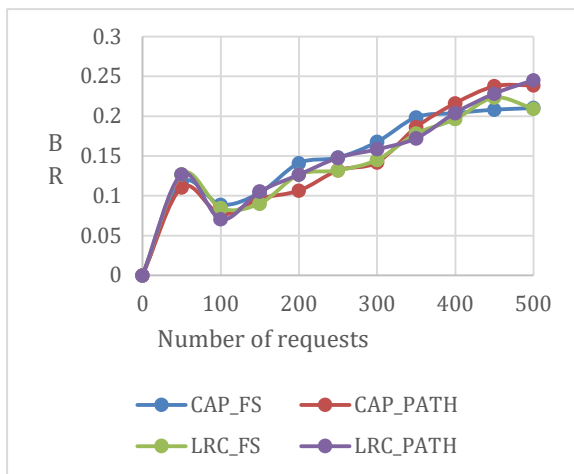


圖 12. RMLSA 模式(50)

圖 9 至圖 14 為在不同需求的虛擬節點計算容量上限分別為 50、100 及 150 的環境下評估阻斷率的表現，圖 9 到圖 11 為不考慮調變等級(RSA 模式，即所有傳送需求均以最小之調變等級傳送，以期能傳送較遠之距離，但所需要的頻譜數較高)下的實驗環境，圖 12 到圖 14 為考慮調變等級(RMLSA)下的實驗模式。可以發現在 RSA 或 RMLSA 模式下，LRC-FR 演算法皆有著較低的阻斷率。且能夠明顯看出考慮調變等級的 RMLSA 有著比 RSA 還低的阻塞率。

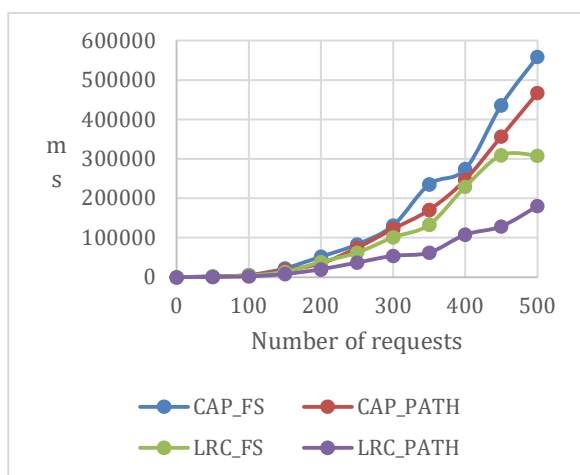


圖 15. RSA 模式(50)

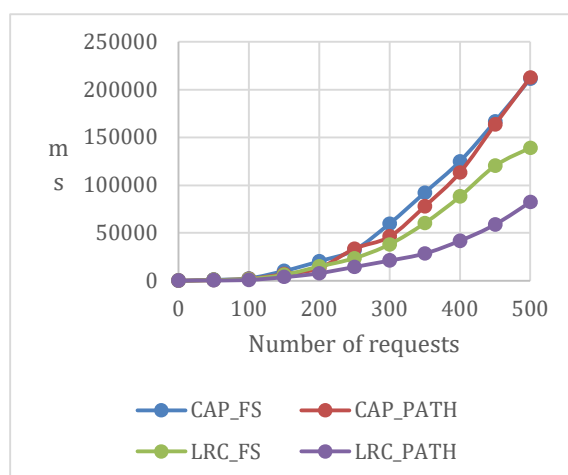


圖 18. RMLSA 模式(50)

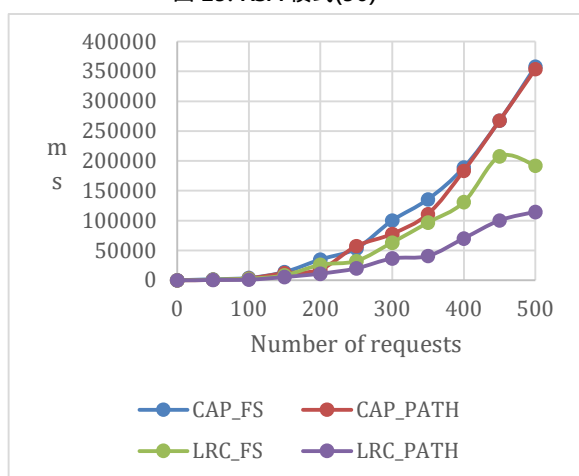


圖 16. RSA 模式(100)

圖 19. RMLSA 模式(100)

圖 20. RMLSA 模式(150)

圖 15 至圖 21 為執行所需時間的數據圖表，一樣在圖 16 到圖 18 是 RSA 模式的狀況，圖 19 至圖 21 是 RMLSA 的模式。我們可以發現 LRC-PATH 演算法在考慮調變等級與否的狀態下都有著最低的執行時間，且明顯 RMLSA 模式跟 RSA 模式比起來，前者所需要的執行時間來得更少。

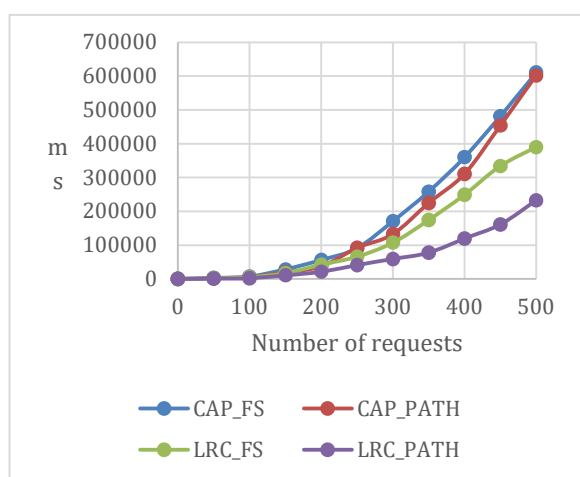


圖 17. RSA 模式(150)

圖 22. RSA 模式(50)

圖 23. RSA 模式(100)

圖 24. RSA 模式(150)

圖 25. RMLSA 模式(50)

圖 26. RMLSA 模式(100)

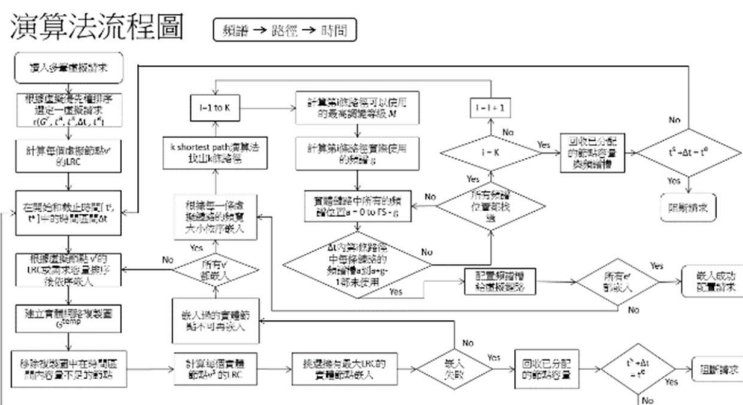
圖 27. RMLSA 模式(150)

圖 22 至圖 27 為阻斷原因的分析統計圖，圖 22 到圖 24 為 RSA 模式的情形，圖 25 至圖 27 是 RMLSA 的狀況。我們可以從圖表

embedding with adaptive modulation in flexi-grid networks,” *Journal of Lightwave Technology*, 36(17): 3551-3563.

[6] Zhe Ding, Zhanqi Xu & Xiaodong Zeng. (2018) “Aggregation defragmentation algorithm based on channel density and minimum gap in elastic optical networks,” *Photonic Network Communications*, 35(1): 1-10.

[7] Jeremy Siek, Lie-Quan Lee & Andrew Lumsdaine. (2001). The boost Graph Library-User Guide and Reference Manual [Online]. Available from: <https://markqiu.files.wordpress.com/2009/12/b>



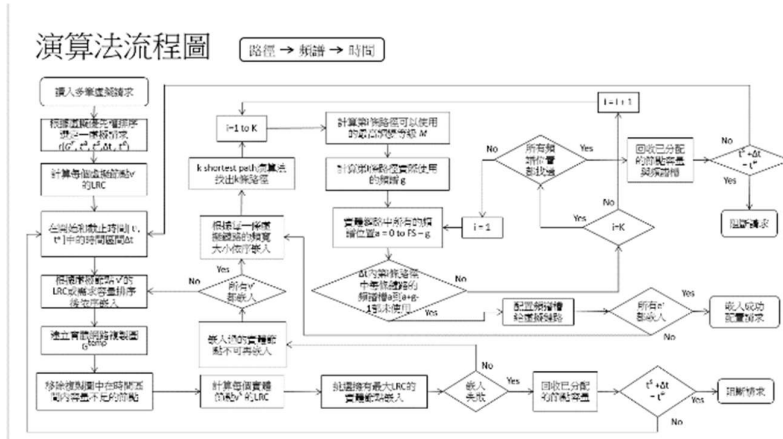
[oost-graph-library.pdf](#) [Accessed 6 March 2019]

[8] M. Zhu, S. Zhang, Q. Sun, G. Li, B. Chen, and J. Gu, (2018) “Fragmentation-Aware VONE in elastic optical networks,” *J. Opt. Commun. Netw.*, 10(9): 809-822.

- [9] H. Liu, C. Xiong, Y. Chen, C. Li, and D. Chen, (2018) “An optimization method of VON mapping for energy efficiency and routing in elastic optical networks,” *Opt. Fiber Technol.*, 41: 173–181.

7. 附錄

7.1 流程圖



7.2 正規化

求得需求 x 在 LRC 條件下的權重 $L(x)$

(X 為當前需求的總 LRC 值)

令 $a = \min\{\text{all requests 的總 LRC}\}$

令 $b = \max\{\text{all requests 的總 LRC}\}$

$L(x) = (x-a) / (b-a)$; $0 \leq L(x) \leq 1$

求得需求 x 在 t^s 條件下的權重 $T(x)$

(t^s 是當前需求 x 的初始開始時間)

令 $a = \min\{\text{all requests 的 } t^s\}$

令 $b = \max\{\text{all requests 的 } t^s\}$

$T(x) = (t^s-a) / (b-a)$; $0 \leq T(x) \leq 1$