

메쉬형 광 네트워크에서 유전자 알고리즘을 이용한 라우팅 전략

박 용규, 위 규범, 홍 만표, 예 홍진
아주대학교 정보통신공학과

{dongas, kbwee, mphone, hjyeh}@madang.ajou.ac.kr

Routing Strategy using Genetic Algorithms In Mesh-Like Optical Networks

Yongkyu Park, Kyubum Wee, Manpyo Hong, Hongjin Yeh
Dept. of Information & Communication, Ajou University

요 약

본 논문에서는 광 네트워크에서 방송(broadcasting)을 위해 유전자 알고리즘을 이용한 라우팅 전략(routing strategy)을 제시하고 있다. 논문에서 제시하는 유전자 알고리즘(genetic algorithms)은 메쉬형(mesh-like) 광 네트워크의 각 링크에 적은 수의 광경로(lightpath)가 통과하도록 하여 네트워크 전체 링크의 사용빈도를 감소시키고 광경로들이 특정 링크로 집중되는 현상을 감소시켜 파장의 수가 고른 노드들을 네트워크에 위치시키므로 네트워크의 효율을 증가시키는 라우팅 전략을 제시할 수 있음을 보이고 있다.

1. 서론¹⁾

광 네트워크(Optical Network)에는 작게는 프로세서부터 크게는 서버나 터미널과 같은 수많은 노드들이 존재한다. 또한 노드들은 정보전달을 위해 파장 분할 다중화(WDM : Wavelength Division Multiplexing) 방식을 사용한다. 파장 분할 다중화란 광케이블로 연결된 하나의 링크(Link) 사이에 서로 다른 파장을 사용하는 다중 통신 채널(Multiple Communication Channel)을 의미한다. 다중 통신 채널이란 1초당 테라(10^{12})비트급의 데이터 전송이 가능한 여러 개의 독립적인 채널을 의미한다.[1][2]. 또한, 송/수신 노드간의 정보는 일정한 크기의 패킷(Packet)으로 나누어진다. 송/수신 노드간에 패킷을 전달하기 위해서는 라우팅을 통해 만들어진 경로에 각 링크에서 사용 가능한 파장이 할당되어야 한다. 그러므로 광경로(Lightpath)란 패킷을 전달하기 위한 경로에 파장을 할당한 경로를 의미한다[3].

파장 분할 다중화 방식을 사용하면 한 링크에는 한 개 이상의 광경로들이 통과할 수 있다[4]. 또한 이

러한 광경로들은 다른 광경로와 충돌하지 않기 위해서 다른 파장을 사용한다. 그러므로 링크에는 라우팅 전략에 따라 다양한 그리고 많은 수의 광경로들이 통과하게 된다. 또한 라우팅 전략에 따라 송/수신 노드간의 거리가 최소가 되도록 하는 광경로들이 만들어 질 수 있다. 하지만 송/수신 노드간의 거리가 최소가 되도록 하는 광경로들은 네트워크의 특정 링크를 공통적으로 통과할 수 있게 되지만 다른 특정 링크는 광경로들이 하나도 통과하지 않을 수도 있으며, 네트워크에는 이러한 링크들이 존재할 수 있다. 링크의 통과는 링크에 존재하는 파장의 사용을 의미하므로 공평하지 않은 링크의 사용은 곧 전체 네트워크 효율의 감소를 가져온다.

광경로를 결정하기 위해서는 먼저 라우팅 전략에 의해 송수신 노드간에 경로(Path)를 결정하고 그 경로에 사용 가능한 파장을 할당하는 작업이 필요한데, 이러한 작업을 라우팅과 파장 할당(Routing and Wavelength Assignment : RWA) 문제라고 한다. Ramamurthy 등[5]의 연구결과에서 패킷 전달의 블로킹 확률(Blocking Probability)이라는 관점에서 볼 때, 효과적인 라우팅 전략이 무엇보다도 중요하다는 사실을 실험적으로 보였다.

* 이 논문은 1999년도 정보통신부 대학기초 연구지원사업에 의해 작성되었음.

따라서, 본 논문에서는 메쉬형 광 네트워크에서 각각의 링크를 통과하는 광경로의 수를 줄이므로 네트워크 전체 링크의 사용빈도를 감소시킬 뿐만 아니라 광경로들이 특정 링크로 집중되는 현상을 감소시켜 전체 네트워크의 효율을 증가시키는 라우팅 전략을 제시하고자 한다. 또한 이러한 알고리즘을 사용하는 라우팅에 의해 만들어진 광경로는 네트워크 전체 링크의 사용빈도도 균등히 분포시키는 효과도 가져오게 된다.

제2장에서 문제에 접근하고 제3장에서 유전자 알고리즘을 간단히 소개하고 있다. 제4장에서는 효과적인 라우팅 전략을 위해 유전자 알고리즘을 어떻게 적용하였는지를 다루었고 실험 결과를 제5장에서 보여주고 있다. 마지막으로 제6장에서 본 논문에서 제시한 알고리즘에 따른 라우팅의 실험결과에 대한 결론을 내리고 있다.

2. 문제

본 논문에서 메쉬형 광 네트워크는, 노드의 집합과 링크의 집합을 각각 $V=\{v_1, v_2, \dots, v_j\}$ 와 $E=\{e_1, e_2, \dots, e_j\}$ 로 정의할 때, $G=(V, E)$ 인 그래프(Graph) G 로 표현한다. 또한 인접한 두 노드 사이의 링크 e_j 는 두 인접노드 a_m 과 a_n 의 순서쌍인 (m, n) 으로 표현한다. 또한 링크는 단방향성을 가진다고 가정한다. 광 네트워크에서 정보의 송수신은 $s, d \in V$ 인 송신 노드 s 와 수신노드 d 사이에서 이루어진다. 정보의 송/수신을 위해서는 송/수신 노드 사이에 경로(Path)가 필요하며, 경로는 라우팅 전략(Routing Strategy)에 의해 구성된다. 경로는 송신노드와 수신노드를 연결하는 한 개 이상의 링크들의 집합인 링크의 순열(Sequence Of Links)을 의미하며 $P=\{(s, d)\}$ 로 표현된다. 네트워크의 모든 노드가 전방송(All-to-All Broadcasting)한다고 가정하면, 광경로는 자기 자신을 제외한 노드 쌍만큼 존재한다. 링크 e_i 를 통과하는 경로의 개수를 N_i 라고 할 때, $Minimize \{ \max_{e \in E} N_i \}$ 이 되도록 하는 것이 문제의 목적이다.

3. 유전자 알고리즘

유전자 알고리즘은 생명체의 적자생존과 진화현상에 기초한 검색 알고리즘으로서 잠정적인 해를 나타내는 개체들을 군집으로 구성하고 이 군집에 교배, 돌연변이 등의 유전자 연산을 반복적으로 적용함으로써 다음 세대에 더 좋은 해를 나타내는 개체군으로 진화시킨다[6]. J. H. Holland에 의해 제안된 유전자 알고리즘 방법은 과학과 공학의 많은 분야에서 적절히 변형되어 사용되고 있다. 예를 들면, 최적화 문제(Optimization Problem), 자동 프로그래밍(Automatic Programming), 생태학(Ecology), 경제학(Economics), 면역체계(Immune Systems), 유전학(Population Genetics), 사회체계(Social Systems) 등의 분야에 응용되고 있다[6][7].

유전자 알고리즘의 기본적인 개념은 적합도가 높은 개체가 가진 좋은 형질의 유전자들이 선택에 의하여 그 수가 불어나면 교배를 통하여 서로 다른 개체에 있었던 좋은 형질의 유전자들이 같은 개체로 모이게 되며, 이에 따라 적합도가 점점 높은 개체군으로 진화해 나간다는 것이다. 그러므로 본 논문에서는 정규화된 광 네트워크에 다음과 같은 세 가지 단계의 유전자 알고리즘을 적용한다.

- (1단계) 경로를 나타내는 개체들이 군집을 구성한다.
- (2단계) 군집에 교배, 돌연변이 등의 유전연산자를 반복하여 적용한다.
- (3단계) 링크의 사용빈도를 균등히 하는 군집으로 진화한다.

4. 유전자 알고리즘을 이용한 라우팅

4.1 인코딩(Encoding)

메쉬형 광 네트워크에서 각각의 노드는 집합 $V=\{v_1, v_2, \dots, v_j\}$ 로 표현되며, 인접 노드 n_a, n_b 사이의 링크는 순서쌍 (a, b) 로 나타낸다. 또한 경로에 동(E)/서(W)/남(S)/북(N)의 네 가지 방향벡터(Direction Vector)를 두어 링크를 통과할 때의 경로의 방향을 타나내도록 한다. 경로에 방향벡터를 포함하는 이유는 경로가 자신이 거쳐온 노드에 대해서는 두 번 다시 반복하여 통과하지 않도록 하여 송신노드와 수신노드사이의 광경로가 항상 최단거리를 유지하도록 할 수 있기 때문이다. 경로에 파장을 할당하지 않은 이유는 각 링크를 지나는 경로의 개수를 최소화하는 라우팅 전략을 찾는 것이 목적이라고 서론에서

가정하였기 때문이다. 만약 메쉬형 네트워크의 크기가 $m \times n$ 이라면 송신 노드에서 수신 노드로의 가능한 경로는 이러한 링크의 순열과 그에 대한 방향 벡터로 표현되며, 하나의 경로가 하나의 크로모솜이 된다. 그러므로 크로모솜들이 모여 $mn(mn-1)$ 크기의 군집 (Population)을 구성하게 된다. 다음은 이러한 크로모솜을 구성하기 위한 알고리즘이다. 알고리즘은 2차원 메쉬에서의 인코딩과 토러스나 하이퍼큐브와 같은 3차원에서의 인코딩으로 나뉜다. [알고리즘 1]에서 X_{com}, Y_{com} 은 3차원형의 네트워크에서 송수신 노드 사이에 최단거리를 갖도록 하는 방향을 선택하기 위한 기준이며 $MakePath()$ 는 두 점의 x 값과 y 값을 이용하여 링크를 생성하는 함수이다. 이때, 경로는 송신노드 s 에서 수신노드 d 까지 이동할 때, 동(East)쪽이나 서(West)쪽으로 먼저 이동한 후 남(South)쪽이나 북(North)쪽으로 이동하며 인코딩 된다.

```

Algorithms : 인코딩
Procedure : encoding()
 $X_{offset} = X_{dest} - X_{current}$  ;  $Y_{offset} = Y_{dest} - Y_{current}$ 
 $X_{temp} = X_{current}$  ;  $Y_{temp} = Y_{current}$ 
 $X_{com} = \lfloor \frac{ROW}{2} \rfloor$  ;  $Y_{com} = \lfloor \frac{COL}{2} \rfloor$ 
if  $|X_{offset}| \leq X_{com}$  and  $|Y_{offset}| \leq Y_{com}$  then
    PathInMesh()
elseif  $|X_{offset}| > X_{com}$  or  $|Y_{offset}| > Y_{com}$  then
    if  $Y_{offset} < 0$  then
         $Y_{current} = (Y_{current} + 1) \bmod COL$ 
        Path.Linkcurrent = MakePath( $X_{temp}, Y_{temp}$ ,  $X_{current}, Y_{current}$ )
        Path.Directcurrent = East
    endif
    if  $Y_{offset} > 0$  then
         $Y_{current} = Y_{current} - 1$ 
        if  $Y_{current} < 0$  then  $Y_{current} = Y_{current} + COL$  endif
        Path.Linkcurrent = MakePath( $X_{temp}, Y_{temp}$ ,  $X_{current}, Y_{current}$ )
        Path.Directcurrent = West
    endif
    if  $X_{offset} < 0$  and  $Y_{offset} = 0$  then
         $X_{current} = (X_{current} + 1) \bmod ROW$ 
        Path.Linkcurrent = MakePath( $X_{temp}, Y_{temp}$ ,  $X_{current}, Y_{current}$ )
        Path.Directcurrent = South
    endif
    if  $X_{offset} > 0$  and  $Y_{offset} = 0$  then
         $X_{current} = X_{current} - 1$ 
        if  $X_{current} < 0$  then  $X_{current} = X_{current} + ROW$  endif
        Path.Linkcurrent = MakePath( $X_{temp}, Y_{temp}$ ,  $X_{current}, Y_{current}$ )
        Path.Directcurrent = North
    endif
endif
EndProcedure

```

[알고리즘 1] : 경로의 인코딩

4.2 적합도 함수(Fitness Function)

적합도 함수란 현 세대의 개체들보다 진화된 좋은 자손들이 다음 세대에 더 많이 나타나도록 하기 위해서 좋은 형질을 가진 부모 개체의 선택이 확률적으로 높아지도록 하는 함수이다. 본 논문의 목적은 임의의 링크를 통과하는 경로의 개수를 최소화할 뿐만 아니라 전체 링크의 사용빈도도 균등히 하는 라우팅 정책을 찾는 것이다. 그러므로 목적에 알맞은 적합도 함수는 임의의 경로가 사용빈도가 높은 링크를 많이 포함한 경로를 확률적으로 더 많이 선택되도록 유도하는 함수이어야 한다.

따라서 본 논문의 유전자 알고리즘에서 사용될 적합도 함수 F 는 전체노드의 집합을 N 으로, 전체 링크의 집합을 L 로, 전체 경로의 집합을 P 로 정의할 때, 임의의 $p_i \in P$ 에 대하여 다음과 같이 정의한다.

$$F(p_i) = \frac{\max_{l_j \in p_i} use(l_j)}{\max_{l_k \in L} use(l_k)}$$

여기서 $use(l_j)$ 는 링크 l_j 의 사용빈도를 나타내며, 적합도 값은 $0 < F(p_i) \leq 1$ 의 범위를 갖는다.

4.3 선택(Selections)

선택이란 적합도 값에 알맞은 크로모솜을 택하는 것이다. 만약 적합도 값이 좋은 개체만을 선택한다면 군집의 다양성을 저해하여 지역적인 최적치에 빠지게 되므로 최상의 값을 얻어내지 못할 수 있다. 반대로 적합도 값에 준하지 않는 경우는 세대의 진화의 속도를 늦추게 되어, 긴 진화의 시간을 가지게 되므로 선택 방법은 교배와 돌연변이의 변화와 균형적으로 이루어져야 한다. 본 논문에서 사용하는 크로모솜의 선택을 위해 룰렛 휠(Roulette Wheel) 선택 방식을 사용한다. 룰렛 휠 선택 방식을 사용하는 이유는 유전자 알고리즘에서 가장 많이 사용하는 방법이며 구성 또한 간단하기 때문이다. 만약, 임의의 경로에 사용빈도가 높은 링크가 많이 포함되면, 그 경로의 적합도 함수 값이 커지며 그 경로는 룰렛 휠 선택 방식으로 선택될 확률이 높다.

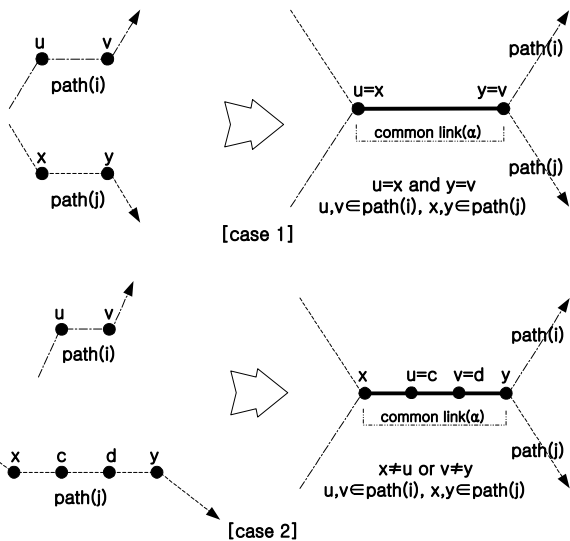
4.4 유전 연산자(Genetic Operator)

4.4.1 교배(Crossover) 연산

교배연산은 선택된 두 개의 개체에서 임의로 선택

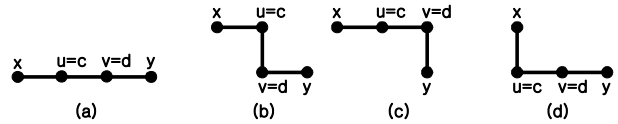
한 교배지점을 기준으로 두 개체의 유전 정보를 서로 교환하여 새로운 개체를 구성하는 연산[7]이다. 본 논문에 적용한 교배연산은, 먼저 적합도 함수 값이 높은 두 개의 경로를 선택한다. 두 번째 단계로 선택한 경로들 중 임의의 한 경로 내의 사용빈도가 가장 높은 링크를 선택하여 사용빈도가 적은 다른 링크로 교체하는 상대적인 교배연산이다. 이러한 교배연산을 수행하기 위해서는 먼저 경로의 공통 구간(Common Link)과 교배 영향구간을 정의해야 한다.

공통 구간이란 선택된 두 개의 경로에서 서로 공통으로 사용하는 연속된 링크들 중에서 사용빈도 즉, $use()$ 값이 가장 높은 구간으로 실제 교배연산이 수행되어질 구간이며 [그림 1]은 $i \neq j$ 이고 링크 (u, v) 를 포함하는 $path(i)$ 와 링크 (x, y) 를 포함하는 $path(j)$ 의 공통 구간(common link : a)을 나타내고 있다. [case 1]은 $path(i)$ 의 사용빈도가 가장 높은 단일링크 (u, v) 와 $path(j)$ 의 사용빈도가 가장 높은 단일링크 (x, y) 가 만나는 경우이며, 공통링크를 이루는 경우를 보여준다. [case 2]는 $path(i)$ 의 사용빈도가 가장 높은 단일링크 (u, v) 와 $path(j)$ 의 사용빈도가 가장 높은 복합링크 (x, y) 가 만나는 경우이다. 즉, 임의의 링크 (m, n) 의 사용회수를 $|m, n|$ 라고 할 때, 경로의 사용회수가 $|u, v| = |x, c| = |c, d| = |d, y|$ 이라 할지라도 두 경로의 공통 링크는 (u, v) 이 아니라 (x, y) 가 되는 경우를 보여준다.



[그림 1] 공통 구간(Common Link)의 예

[case 2]인 경우는 다음 [그림 2]와 같이 방향의 변화가 없는 (a)형과 방향의 변화를 갖는 (b), (c), (d)형으로 나눌 수 있다.



[그림 2] [case 2]의 다른 형태

교배영향구간은 교배구간을 우회하기 위하여 함께 대체되어야 할 교배구간 이전 또는 이후의 최소한의 연속된 링크들을 의미한다. 그러므로 공통링크 앞뒤의 링크들을 검사하여 방향벡터의 변화가 생기는 곳까지를 교배영향구간으로 설정하게 되며 [알고리즘 2]에 의해 교배영향구간(β)을 구하게 된다.

```

Algorithm : 교배영향구간
Procedure : FindCOEffect()
...
case 1 :
    if ( length(a)=length(path(i))
        || path(i)의 방향벡터의 종류=1 )
         $\beta = 0$ 
    else
         $\beta = path(i)$ 의 앞/뒤 중
        방향벡터가 변화되는
        곳까지의 길이중 작은 구간
case 2 :
    if (path(i)의 방향벡터의 종류=1)
         $\beta = case 1$ 과 같이 계산
    if (path(i)의 방향벡터의 종류 $\geq 2$ )
         $\beta = 0$ 
...
EndProcedure
    
```

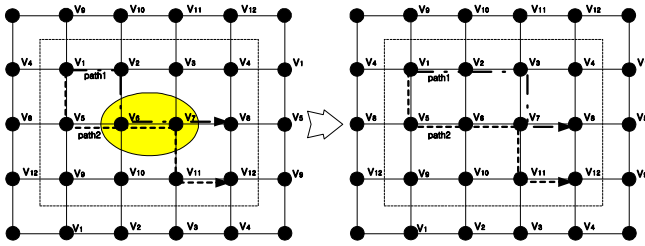
[알고리즘 2] : 교배영향구간

그러므로 이러한 공통구간과 교배영향구간을 합친 구간을 교배 구간($L = \alpha + \beta$)로 정의하며 이 구간을 다른 경로로 대체하는 연산을 교배연산으로 정의한다. 교배연산은 [알고리즘 3]에 의해 수행되며 [그림 3]는 3×4 메쉬형 네트워크에서 교배 연산을 수행한 결과를 보여준다.

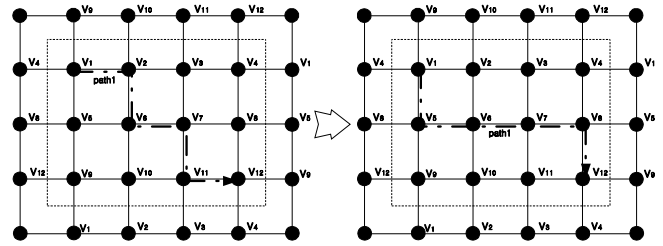
```

Algorithm : 교배 연산
Procedure : Crossover()
...
case 1 :
    if ( length(a)=length(path(i))
        || path(i)의 방향벡터의 종류=1 )
        NOP
    else
        Permutation( I )
case 2 :
    if (path(i)의 방향벡터의 종류=1)
        case 1의 연산 수행
    if (path(i)의 방향벡터의 종류=2)
        ReversePath( I )
    if (path(i)의 방향벡터의 종류 $\geq 3$ )
        Permutation( I )
...
EndProcedure
    
```

[알고리즘 3] : 교배 연산



[그림 3] 교배 연산의 예



[그림 4] 돌연변이 연산의 예

4.4.2 돌연변이(Mutation) 연산

돌연변이 연산이란 하나의 개체에 새로운 유전적 정보를 가하는 연산으로 군집이 국부적인 최적화 값에 수렴하지 않도록 하는 역할을 가지는 연산[7]이다. 돌연변이 연산은 임의의 경로를 선택하여 특정 구간의 방향벡터의 순서를 치환(Permutation)하는 연산으로 정의하였다. 교배 연산과 마찬가지로 돌연변이 연산에서도 돌연변이 구간을 정의해야 한다. 돌연변이 구간이란 어떤 경로에서 임의로 선택된 연속된 링크들 중에서 방향벡터의 순서가 치환될 구간이다. 이러한 돌연변이 구간은 다음과 같은 [알고리즘 4]에 의해 수행되며 [그림 4]는 3×4 메쉬형 네트워크에서 돌연변이 연산을 수행한 결과를 보여준다.

```

Algorithms : 돌연변이 연산
Procedure : Permutation()
...
DirectionRandomSort(TempPath, MInt)
...
for i=1 to MInt do
...
if TempPath.DirectIntStart = East then
YIntStart = (YIntStart + 1) mod COL
TempPath.LinkIntStart = MakePath(Xtemp, Ytemp, XIntStart, YIntStart)
if TempPath.DirectIntStart = West then
YIntStart = YIntStart - 1
if YIntStart < 0 then YIntStart = YIntStart + COL endif
TempPath.LinkIntStart = MakePath(Xtemp, Ytemp, XIntStart, YIntStart)
if TempPath.DirectIntStart = South then
XIntStart = (XIntStart + 1) mod ROW
TempPath.LinkIntStart = MakePath(Xtemp, Ytemp, XIntStart, YIntStart)
if TempPath.DirectIntStart = North then
XIntStart = XIntStart - 1
if XIntStart < 0 then XIntStart = XIntStart + ROW
TempPath.LinkIntStart = MakePath(Xtemp, Ytemp, XIntStart, YIntStart)
EndProcedure
  
```

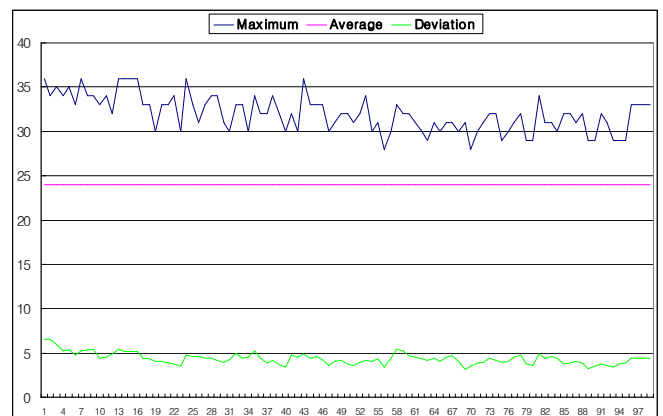
[알고리즘 4] : 돌연변이 연산

5. 실험결과

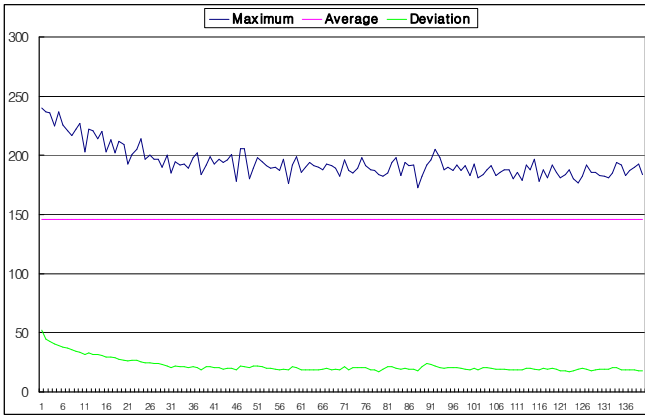
실험에 필요한 환경변수로 군집(Population)의 크기는 메쉬형(mesh-like) 네트워크의 크기인 $m \times n$ 에 의하여 $mn(mn-1)$ 로 고정하였으며 세대의 수는 100으로 설정하였다. 또한 교배연산 비율은 0.8로, 돌연변이 연산의 비율은 0.2로 각각 설정하였다.

이때, 각 링크를 지나는 경로의 개수가 감소되는지를 판단하기 위해 평균과 함께 링크 사용회수의 최대값을 나타냈으며, 네트워크의 링크들의 사용빈도가 균등하게 분포되는지를 분석하기 위해서 표준편차를 구하였다.

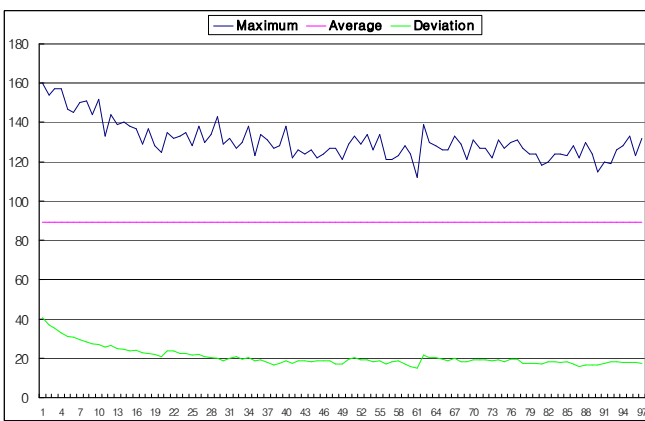
[그림 5], [그림 6], [그림 7]와 같이 실험 결과를 살펴보면, 최대값이 세대를 거듭할수록 점차 평균값에 가까워짐을 확인할 수 있다. 이것은 링크를 지나는 경로의 개수가 감소함을 나타낸다. 또한 표준편차 역시 세대를 거듭할수록 점차 감소함을 볼 때, 전체 링크의 사용빈도도 점차 균등해짐을 확인하였다.



[그림 5] 3×5 mesh-like에서의 평균, 표준편차, 최대값



[그림 6] 4×9 mesh-like에서의
평균, 표준편차, 최대값



[그림 7] 6×9 mesh-like에서의
평균, 표준편차, 최대값

6. 결론

본 논문에서는 메쉬(mesh), 토러스(torus), 하이퍼 큐브(hyper-cube)와 같은 메쉬형(mesh-like) 광 네트워크에서 각각의 링크를 통과하는 경로의 개수를 최소화하여 네트워크 전체 링크의 사용회수를 감소시킬 뿐만 아니라 경로들이 특정 링크로 집중되는 현상을 감소시켜 파장의 수가 고른 노드들을 네트워크에 위치시키므로 네트워크의 효율을 증가시키는 라우팅 전략을 유전자 알고리즘을 이용하여 찾을 수 있음을 제안하였다.

실험 결과 제안된 유전자 알고리즘을 이용한 라우팅 전략은 경로들로 하여금 최단 거리를 유지하면서 특정 링크를 집중적으로 통과하는 현상을 감소시킴을 확인할 수 있었다.

향후 연구과제로는 제안된 라우팅 알고리즘을 더욱 보완하여 불규칙적인 형태를 갖는 임의의 네트워크에서 발생하는 트래픽(Traffic)에 동적(Dynamic)으

로 대응하는 일반화된 알고리즘을 개발하는 것이다. 또한 파장 할당 문제를 추가적으로 고려하여 RWA(Routing and Wavelength Assignment) 문제를 해결하는 것이다.

7. 참고문헌

- [1] C. A. Brackett, "Dense Wavelength Division Multiplexing Networks: Principles and Applications," IEEE J. on Selected Areas in Communications. Vol. 8, No. 6, pp. 948-964, Aug. 1990.
- [2] R. J. Vetter and D. H. C. Du, "Distributed Computing with High-Speed Optical Networks," IEEE Computer, pp. 8-18, Feb. 1993.
- [3] K.-C. Lee and V. O. K. Li, "A wavelength convertible optical network", IEEE/OSA Journal of Lightwave Technology, vol. 11, pp. 962-970, May/June 1993.
- [4] Biswanath Mukherjee, "Optical Communication Networks", McGraw-Hill, 1997.
- [5] S. Ramamurthy. Biswanath Mukherjee. "Fixed-Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks", IEEE GLOBECOM, pp. 2295-2302, Nov. 1998.
- [6] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.
- [7] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1996.