메쉬형 광통신 네트워크에서의 광경로를 위한 유전자 알고리즘 기반의 라우팅

박용규[1] 위규범 홍만표 예홍진
아 주 대 학 교 정 보 통 신 전 문 대 학 원
{dongas, kbwee, mphong, hjyeh}@madang.ajou.ac.kr

Genetic Algorithms Based Routing for LightPath In Mesh-Like Optical Communication Networks

YongKyu Park[1] Kyubum Wee Manpyo Hong Hongjin Yeh
Professional Graduate School of Information and Communication Technology

요 약

본 논문은 광네트워크(optical network)에서 정보의 송/수신에 필요한 광경로를 설정하기 위한 방법으로 유전자 알고리즘을 이용한 라우팅 전략을 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 네트워크의 효율을 증가시키는 라우팅 전략을 제시할 수 있음을 보이고 있다.

1. 서론[1]

파장 분할 다중화 방식에 의하면 한 링크에는 한 개 이상의 광경로들이 통과할 수 있으며[1] 각 경로는 서로 다른 파장을 사용하여 다른 광경로와 충돌하지 않도록 한다. 라우팅에 따라 링크에는 많은 수의 광경로(lightpath)들이 통과할 수 있다. 또한 송/수신 노드간의 거리가 최소가 되도록 하는 광경로들이 만들어지도록 라우팅할 수 있다. 하지만 송/수신 노드간의 거리가 최소가 되도록 하는 광경로들은 네트워크의 특정 링크를 공통적으로 통과할 수는 있지만 다른 특정 링크는 하나도 통과하지 않을 수도 있는 경우의 수가 발생할 수 있다. 링크의 통과는 링크에 존재하는 파장의 사용을 의미하므로 공평하지 않은 링크의 사용은 곧 전체 네트워크 효율의 감소를 가져온다.

광경로는 먼저 라우팅에 의해 송/수신 노드간에 경로(path)를 결정하고 그 경로에 사용 가능한 파장을 할당하는 작업을 통해 만들어지는데, 이러한 작업을 라우팅과 파장 할당(Routing and Wavelength Assignment : RWA) 문제라고 한다. Ramamurthy 등[2]의 연구결과에서 패킷 전달의 블로킹 확률(Blocking Probability)이라는 관점에서 볼 때, 효과적인 라우팅 전략이 무엇보다도 중요하다는 사실을 실험적으로 보였다. [3]은 메쉬 구조의 광네트워크에서의 라우팅를 위한 유전자 알고리즘을 제안하였다.

본 논문에서는 선형(linear), 링(ring), 메쉬(mesh), 토러스(torus)와 같은 메쉬형(mesh-like) 광통신 네트워크에서 각 링크를 통과하는 광경로의 수를 최소화하여 네트워크 전체 링크의 파장의 수를 감소시키는 알고리즘을 제안하였다. 제2장에서는 문제의 접근방법을 제시하며 제3장에서는 유전자 알고리즘을 라우팅에 어떻게 적용하였는지를 소개하였다. 제4장에서 실험결과를 보여준 후 제5장에서 결론을 내리고 있다.

※ 이 논문은 1999년도 정보통신부 대학기초 연구지원사업에 의해 작성되었음.

2. 문제

논문에서 메쉬형(mesh-like) 광통신 네트워크는, 노드와 링크를 각각 $V=\{v_1, v_2, \cdots, v_i\}$ 와 $E=\{e_1, e_2, \cdots, e_j\}$의 집합으로 정의할 때, $G=(V,E)$로 표현한다. 또한 인접한 두 노드 사이의 링크 e_j 는 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. 송/수신 노드사이에는 한 개 이상의 링크들의 집합인 링크의 순열(Sequence Of Links)에 의해 경로가 설정되며 $P=\{(s,d)\}$로 표현한다. [3]에서와 같이 네트워크의 모든 노드가 전방송 한다고 가정하면, 광경로는 자기 자신을 제외한 노드 쌍만큼 존재한다. 즉, 링크 e_i 를 통과하는 경로의 개수를 N_i 라고 할 때, $Minimize\{\max_{e_i \in E} N_i\}$ 이 되도록 하는 것이 문제의 목적이다.

3. 유전자 알고리즘을 이용한 라우팅

3.1 인코딩(Encoding)

각각의 노드는 집합 $V=\{v_1, v_2, \cdots, v_i\}$로 표현되며, 인접 노드 v_a, v_b 사이의 링크는 순서쌍 (a,b)로 나타낸다. [그림 1]과 같이 메쉬형 네트워크에서, 송/수신 노드 사이의 경로는 다음과 같은 약속으로 생성된다. 송신노드를 v_i, 수신노드를 v_j라 하면 v_i의 x 좌표와 y 좌표는 i_x와 i_y로, v_j의 x 좌표와 y 좌표는 j_x와 j_y로 나타낼 수 있으며, x 좌표간의 거리와 y 좌표간의 거리도 각각 $x_{offset}=|i_x-j_x|$와 $y_{offset}=|i_y-j_y|$로 계산할 수 있다. 또한, x축의 길이를 $L_x=\left\lfloor \frac{\text{행의 개수}}{2} \right\rfloor$, y축의 길이를 $L_y=\left\lfloor \frac{\text{열의 개수}}{2} \right\rfloor$라 하자. 이때, 두 노드간의 경로는 $x_{offset}<L_x$와 $y_{offset}<L_y$이 되는 방향으로 경로의 방향을 설정한 다음 [알고리즘 1]을 이용하여 송/수신 노드간의 링크를 인코딩한다. x

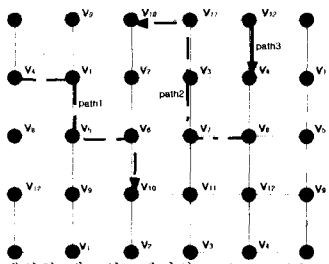
축과 y축의 길이를 각각 행의 개수와 열의 개수의 반으로 설정한 이유는 경로를 송/수신 노드간의 최소거리 방향으로 진행시키기 위해서이다.

```
Function : GetLink() {
for( i=0 ; i<송신노드의 개수 ; i++ )
  for( j=0 ; j<수신노드의 개수 ; j++ )
    현재노드=i, PathIndex=0;
    while( 현재노드 != j )
      CurX=현재노드의 X좌표,CurY=현재노드의 Y좌표;
      현재노드=MakePath(CurX, CurY, j의 X값, j의 Y값, PathIndex);
      PathIndex++ ;
End Function

Function : MakePath( i, j, k, l )
/* Row:행의 수, COLUMN:열의 수 */
if( j==l && i<k ) temp=i+l*(ROW-1);
else if( j==l && i>k ) temp=i+l*(ROW-1);
if( i==k && j<l )
  temp=(ROW-1)*COLUMN+j+i*(COLUMN-1);
else if( i==k && j>l )
  temp=(ROW-1)*COLUMN+j+i*(COLUMN-1);
return temp ;
End Function
```

[알고리즘 1] 경로의 인코딩

위의 방법으로 구성된 경로에는 파장을 할당하지 않는데, 그 이유는 각 링크를 지나는 경로의 개수를 최소화하는 라우팅 전략을 찾는 것이 목적이라고 서론에서 가정하였기 때문이다. [그림 1]을 보면, (a)에서는 $v_4 \rightarrow v_{10}$; $v_8 \rightarrow v_{10}$; v_{12}, v_4로의 세 가지 경로를 보여주며 (b)에서는 각 경로가 인코딩된 모습을 보여준다.



(a) 메쉬형 네트워크에서의 path1, path2, path3

path1	(4,1)	(1,5)	(5,6)	(6,10)
path1	(8,7)	(7,3)	(3,11)	(11,10)
path2	(12,4)			

(b) path1, path2, path3의 인코딩의 예

[그림 1] 3×4 메쉬형 네트워트에서의 경로

3.2 적합도 함수(Fitness Function)

적합도 함수란 현 세대의 개체들보다 진화된 좋은 자손들이 다음 세대에 더 많이 나타나도록 하기 위한 함수이다[3][4]. 본 논문의 목적은 임의의 링크를 통과하는 경로의 개수를 최소화할 뿐만 아니라 전체 링크의 사용빈도도 균등히 하는 라우팅을 찾는 것이다. 따라서 본 논문의 유전자 알고리즘에서 사용될 적합도 함수는 [4]에서 소개한 적합도 함수를 사용한다.

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

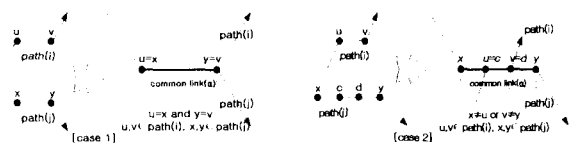
3.3 선택(Selections)

선택이란 적합도 값에 알맞은 크로모좀을 택하는 것이다. 본 논문에서는 [3][4]에서 소개한 유전자 알고리즘에서 가장 많이 사용하고 구성 또한 간단한 루울렛 휠(Roulette Wheel) 선택 방식을 사용하도록 한다.

3.4 유전 연산자(Genetic Operator)

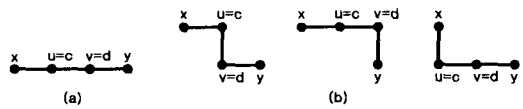
■ **교배(Crossover) 연산**

교배연산은 선택된 두 개의 개체에서 임의로 선택한 교배지점을 기준으로 두 개체의 유전 정보를 서로 교환하여 새로운 개체를 구성하는 연산[5]으로 본 논문에 적용한 교배연산은 다음과 같다. 먼저, 적합도 함수 값이 높은 두 개의 경로를 선택한다. 두 번째는, 선택한 경로 중 임의의 한 경로 내의 사용빈도가 가장 높은 링크를 선택하여 사용빈도가 적은 다른 링크로 교체한다[3][4]. 교배연산을 수행하기 위해서는 먼저 경로의 공통 구간과 교배 영향구간을 정의해야 한다. 공통 구간이란 선택된 두 개의 경로에서 서로 공통으로 사용하는 연속된 링크들 중에서 사용빈도가 가장 높은 구간으로 실제 교배연산이 수행되어질 구간이며 [그림 2]는 임의의 노드 u, v, x, y, c, d에 대해 $i \neq j$이고 링크 (u, v)를 포함하는 $path(i)$와 링크 (x, y)를 포함하는 $path(j)$의 공통 구간(common link: α)을 보여준다. [case 1]은 $path(i)$의 사용빈도가 가장 높은 단일링크 (u, v)와 $path(j)$의 사용빈도가 가장 높은 단일링크 (x, y)가 만나는 경우로 오직 하나의 링크가 공통링크를 이루는 경우를 보여준다. [case 2]는 $path(i)$의 사용빈도가 가장 높은 단일링크 (u, v)와 $path(j)$의 사용빈도가 가장 높은 두 개 이상의 링크로 구성된 복합링크 (x, y)가 만나는 경우이다. 만약 $path(j)$가 선택되면 단일링크 (u, v)만을 공통링크로 선택하는 것이 아니라 복합링크 (x, y)를 공통링크로 설정한다. 즉, (m, n)의 사용회수를 $|(m, n)|$라고 나타낸다면, 경로의 사용회수가 $|(u, v)| = |(x, c)| = |(c, d)| = |(d, y)|$라 할지라도 두 경로의 공통 링크는 (u, v)이 아니라 (x, y)가 된다.



[그림 2] 공통 구간(Common Link)의 예

[case 1]은 반드시 사용빈도가 가장 높은 링크 하나만을 구간으로 설정하지만, [case 2]는 사용빈도가 높은 링크 전체를 구간으로 설정하는 경우이다. 또한 [case 2]는 [그림 3]과 같이 경로의 변화가 없는 (a)와 경로의 변화를 갖는 (b)로 나뉠 수 있다.

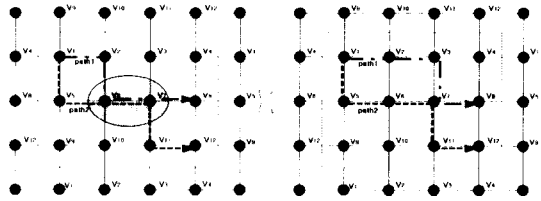


[그림 3] [case 2]의 여러 가지 형태

교배영향구간(β)은 교배구간을 우회하기 위하여 함께 대체되어야 할 교배구간 이전 또는 이후의 최소한의 연속된 링크들을 의미한다[3][4]. 그러므로 이러한 공통 구간과 교배 영향구간을 합친 구간을 교배 구간($I = \alpha + \beta$)로 정의하며 이 구간을 다른 경로로 대체하는 교배연산을 수행한다. 그러므로 교배연

산은 [그림 2]의 두 가지 경우에 대해 교배 연산을 수행한다. [case 1]의 경우에는 아무 연산을 수행하지 않는다. 왜냐하면 기존의 선형경로를 대체하면 우회경로가 만들어져 최단 경로를 벗어날 수 있기 때문이다. [case 2]의 경우에는 링크의 순서를 불규칙적으로 재배열하며, 다양한 형태가 만들어질 수 있다. [그림 3]은 교배 연산의 예를 보여준다.

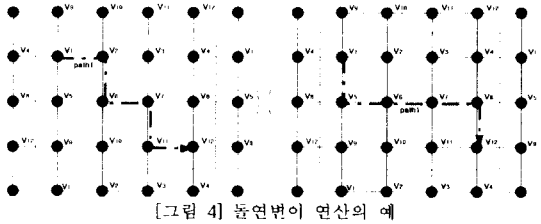
[알고리즘 2] 교배 연산



[그림 3] 교배 연산의 예

■ **돌연변이(Mutation) 연산**

돌연변이 연산이란 하나의 개체에 새로운 유전적 정보를 감하는 연산으로 군집이 국부적인 최적화 값에 수렴하지 않도록 하는 역할을 가지는 연산[5]이다. 돌연변이 연산은 임의의 경로를 선택하여 특정 구간의 방향벡터의 순서를 치환(Permutation)하는 연산으로 정의하였다[3][4]. 교배 연산과 마찬가지로 돌연변이 연산에서도 돌연변이 구간을 정의해야 한다. 돌연변이 구간이란 어떤 경로에서 임의로 선택된 연속된 링크들 중에서 방향벡터의 순서가 치환될 구간이다. [그림 4]는 돌연변이 연산의 예를 보여준다.



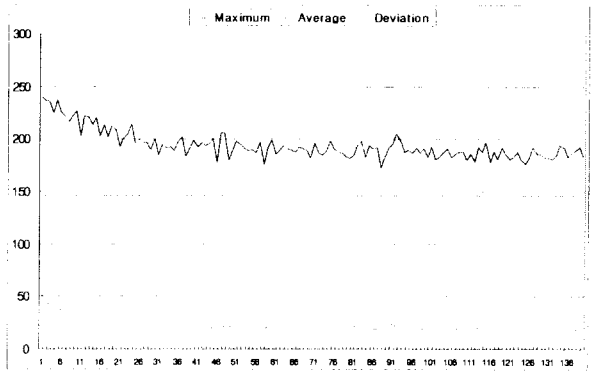
[그림 4] 돌연변이 연산의 예

4. 실험결과

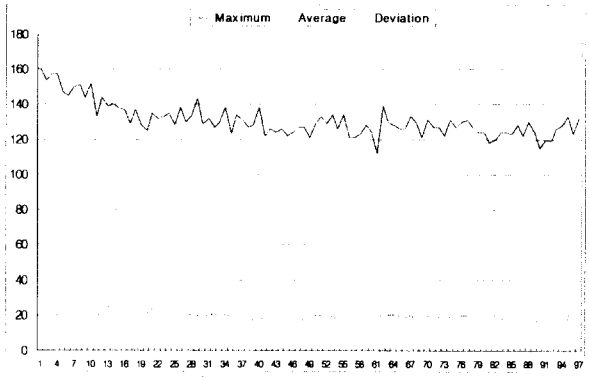
각 링크를 지나는 경로의 개수가 감소되는지를 판단하기 위해 평균과 함께 링크 사용회수의 최대값을 나타냈으며, 네트워크의 링크들의 사용빈도가 균등하게 분포되는지를 분석하기 위해서 표준편차를 보였다. [그림 5], [그림 6]]과 같이 실험 결과를 살펴보면, 최대값이 세대를 거듭할수록 점차 평균값에 가까워짐을 확인할 수 있다. 이것은 링크를 지나는 경로의 개수가 감소함을 나타낸다. 또한 표준편차 역시 세대를 거듭할수록 점차 감소함을 볼 때, 전체 링크의 사용빈도도 점차 균등해짐을 확인하였다.

5. 결론

본 논문에서는 선형(linear), 링(Ring), 메쉬(mesh), 토러스(torus)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 전체 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



[그림 5] 4×9 mesh-like에서의 평균, 표준편차, 최대값



[그림 6] 6×9 mesh-like에서의 평균, 표준편차, 최대값

집중적으로 통과하는 현상을 감소시킴을 확인할 수 있었다.

향후 연구과제로는 제안된 라우팅 알고리즘을 더욱 보완하여 불규칙적인 형태를 갖는 임의의 네트워크에서 발생되는 트래픽(traffic)에 동적(dynamic)으로 대응하는 일반화된 알고리즘을 개발하는 것이다.

6. 참고문헌

[1] Biswanath Mukherjee, "Optical Communi -cation Networks", McGraw-Hill, 1997.

[2] S. Ramamurthy. Biswanath Mukherjee. "Fixed-Alternate Routing and Wavelength Conversion in Wavelength-Routed Optical Networks", IEEE GLOBECOM, pp. 2295 - 2302, Nov. 1998.

[3] 박 용규, 위 규범, 예 홍진. "메쉬형 광 네트워크의 라우팅 문제를 위한 유전자 알고리즘", 정보과학회 가을학술발표논문집 vol III, pp768-770. 1999년 10월(제26권 2호).

[4] 박 용규, 위 규범, 홍만표, 예 홍진. "메쉬형 광 네트워크에서 유전자 알고리즘을 이용한 라우팅 전략". 2000년 정보처리학회 춘계학술발표대회 발표예정.

[5] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1996.