

케이블 비용을 고려한 링메쉬 구조의 광통신망 설계

한정희^{1*}

¹강원대학교 경영대학 경영학과

Design of Hierarchical Ring-Mesh Optical Networks Considering Cabling Cost

Junghee Han^{1*}

¹Department of Business Administration, Kangwon National University

요약 본 논문에서는 링과 메쉬 구조를 이용한 경제적인 계층형 광통신망 설계 문제를 다룬다. 계층형(hierarchical) 광통신망 설계 문제에서 고려하는 비용 요소는 링 구조의 광통신망 내부(intra-ring) 트래픽 처리를 담당하는 OADM(Optical Add-Drop Multiplexer), 메쉬 구조의 링간(inter-ring) 트래픽 처리를 담당하는 OXC(Optical Cross-Connect) 및 OADM간 케이블 비용 및 OXC간 케이블 비용이다. 이 논문에서 제시하는 계층형 광통신망 설계 문제에서는 OADM 및 OXC의 트래픽 처리 용량과 각 링의 OADM 개수 제약을 고려한다. 이 논문에서는 계층형 광통신망 설계 문제를 정수계획법(integer programming) 모형으로 모델링하며, 링의 대칭성(symmetry)을 부분적으로 제거하는 절단평면(cutting planes)을 개발한다. 또한, 우수한 품질의 해를 적절한 시간동안 찾기 위해 휴리스틱 알고리즘을 개발한다. 모의 실험을 통해 절단평면과 휴리스틱 알고리즘의 성능을 평가한 결과 크기가 작은 문제에 대해서는 이 논문에서 개발한 절단평면을 정수계획법 모형에 추가함으로써 최적해를 구하는 시간을 크게 단축시키며, 크기가 큰 문제에 대해서는 이 논문에서 개발한 휴리스틱 알고리즘이 상용 소프트웨어인 CPLEX가 찾는 휴리스틱 해보다 우수한 해를 발견함을 확인하였다.

Abstract In this paper, we deal with a hierarchical ring-mesh optical network design problem. The objective is to minimize the total cost of optical add-drop multiplexers (OADMs) handling intra-ring traffic, optical cross-connects (OXCs) handling inter-ring traffic, and cabling cost among OADMs and among OXCs, while satisfying intra-ring and inter-ring capacities. We develop an integer programming (IP) formulation for the problem and devise some cutting planes that partially break the symmetry of rings. Dealing with the inherent computational complexity of the problem, we devise an effective heuristic procedure that finds a good quality feasible solution within reasonable computing times. Computational results demonstrate the efficacy of the proposed solution procedure; the developed symmetry breaking inequalities significantly reduce the computing time to find an optimal solution for small size problems, and the heuristic procedure finds a better feasible solution than that CPLEX, a commercial optimization software, finds for large size problems.

Key Words : Ring-Mesh Optical Network, Integer Programming, Valid Inequality, Heuristic.

1. 서론

1.1 연구 배경

인터넷의 광범위한 보급과 고용량/고품질의 유무선 통신 서비스 개발로 유선 통신망에서 처리할 트래픽의 양

이 크게 증가함에 따라 통신 회사들은 원활한 유무선 통신 서비스 제공을 위해 광통신망(optical network) 구축 및 고도화를 추진하고 있다. 이 논문에서는 광통신망 구축에 널리 사용되는 SONET(Synchronous Optical Network)과 WDM(Wavelength Division Multiplexing) 기술을 적용한 계층형 링메쉬 (HRM: Hierarchical

*교신저자 : 한정희(jhhan@kangwon.ac.kr)

접수일 09년 01월 11일

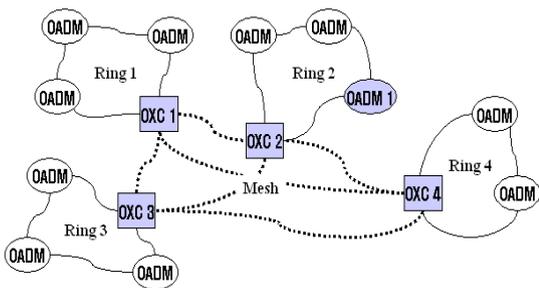
수정일 (1차 10년 04월 12일, 2차 10년 04월 28일)

게재확정일 10년 05월 13일

Ring-Mesh) 네트워크 설계 문제를 다룬다.

SONET은 링 구조를 갖는 광통신망에서 트래픽의 라우팅 및 전송 기능을 수행하며, 광케이블 단절과 같은 사고 발생 시에는 즉각적으로 트래픽을 반대방향으로 우회 시킴으로써 서비스를 정상화하는 자가복구 능력이 뛰어나다. 고용량의 트래픽을 처리하는 광통신망에서 빠르고 신뢰할만한 자가복구 능력은 매우 중요하다. 최근 IPTV (Internet Protocol TV)와 같은 고용량 통신 서비스의 등장으로 트래픽의 양이 크게 증가함에 따라 SONET망의 용량 증설 및 추가 구축이 진행되고 있다. 그러나, 링 형태의 구조 때문에 SONET망은 용량 증설 및 구조 변경이 까다로운 단점을 갖고 있어 기존의 SONET망을 활용하면서 저렴한 비용으로 용량을 증설하기 위해 확장성이 뛰어난 링메쉬(ring-mesh) 구조를 도입하고 WDM 기술을 적용하는 연구가 진행되고 있다[8,10,11]. 메쉬 구조의 상위 계층 광통신망이 링 구조를 갖는 하위 계층의 SONET 망들을 연결하는 링메쉬 구조를 도입하면, 필요에 따라 링 구조의 SONET망을 추가하거나 삭제할 수 있다.

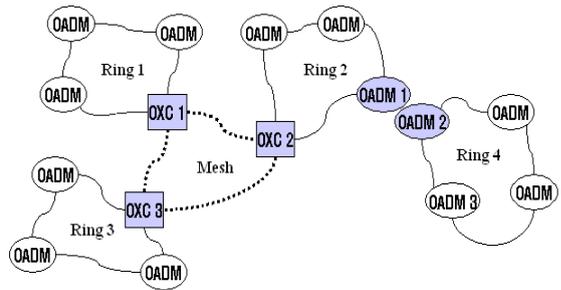
한편, WDM은 다수의 광신호를 하나의 광섬유를 이용하여 동시에 전송하는 기술로서, 각 노드에 OADM (Optical Add Drop Multiplexer) 및 OXC (Optical Cross-Connect)를 설치하면 기존에 매설된 광케이블의 용량을 수십배까지 늘릴 수 있으므로 SONET 기반의 기존 광통신망의 용량을 적은 비용으로 증가시킬 수 있다. OADM은 링 구조의 SONET-WDM망에서 각 파장(wavelength)에 트래픽을 추가하거나 종료시키는 기능을 하며, OXC는 링과 링 사이의 트래픽을 라우팅하는 기능을 수행한다.



[그림 1] OXC를 이용한 링메쉬 광통신망 구조 예시

그림 1에는 모든 링에 OXC를 설치하는 링메쉬 구조를 나타낸다. 그림 1에서 OXC 4는 나머지 OXC들과 멀리 떨어져 있어 4개의 OXC를 모두 연결할 경우 메쉬망 구축에 많은 비용이 소요될 수 있다. 특히, 메쉬망에서의 전송지연은 주로 홉(hop) 수에 비례하므로 홉수를 줄이기

위해서는 메쉬망의 연결 밀도를 높일 필요가 있는데, 이 경우 OXC가 서로 멀리 떨어져 있으면 메쉬망 구축 비용이 크게 증가하므로 OXC를 서로 근거리에서 위치시키는 것이 바람직하다. 그림 2에는 동일한 노드에 OADM을 중복 설치하여 각각의 OADM이 서로 다른 링에서 동작하도록 함으로써 네트워크의 연결성을 보장하는 구조를 나타낸다. 즉, Ring 2에서 OADM 1을 설치한 노드에 두 번째 OADM 2를 설치하여 OADM 2는 Ring 4에서 동작하도록 하는 것이다. 이와 같은 구조는 Ring 4에 속한 노드와 다른 링에 위치한 노드간 트래픽이 존재하지 않을 때 고려할 수 있다. 그림 1과 비교할 때 그림 2에서는 OADM 1 위치에 Ring 4를 위한 OADM 2를 추가 설치하고 OXC 4 위치에 OXC보다 저렴한 OADM 3을 설치하고 있다. 또한, 그림 2에서는 다른 OXC와의 거리가 가장 먼 OXC 4를 사용하지 않음으로써 메쉬망 구축 비용을 줄일 수 있다. 그러나, 그림 2의 Ring 4에는 OADM 2가 추가되었으므로 Ring 4 구축시 OADM 비용이 증가할 수 있다.

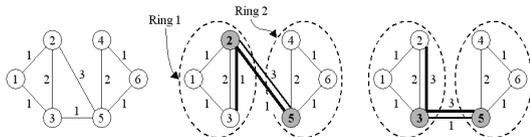


[그림 2] OXC와 OADM을 이용한 링메쉬 광통신망 구조 예시

1.2 연구 내용

이 논문에서 다루는 계층형 링메쉬 광통신망 설계 문제에서는 네트워크의 연결성을 보장하기 위해서 그림 2의 Ring 1 ~ Ring 3과 같이 각 링에 OXC를 설치하거나 Ring 4와 같이 OADM을 중복 설치하는 방법을 모두 허용하며, 노드쌍(node pair)간에 정의된 통신 수요를 처리하기 위해 각 링마다 OXC 설치 여부를 결정한다. 또한, OXC를 사용하는 링에서는 메쉬망 구축비용을 최소화하는 OXC 위치를 결정한다. 이 때, 목적함수는 OADM과 OXC 비용, 각 링에 설치되는 OADM간 거리비용 및 OXC간 거리비용의 합을 최소화하는 것이며, OADM 및 OXC의 용량 제약과 각 링에 설치하는 OADM의 개수 제약을 고려한다. OXC는 각 링마다 1개만 설치할 수 있다고 가정한다.

OXC가 설치되는 위치에 따라 메쉬망 구축비용 뿐만 아니라 각 링의 내부 트래픽(intra-ring traffic)의 양이 달라질 수 있다. 그림 3을 참고로 OXC 위치와 내부 트래픽(즉, OADM 트래픽)의 관계를 설명한다. 그림 3 (a)에서 링크위의 숫자는 노드쌍간의 통신 수요를 나타내며, 그림 3 (b)와 (c)에는 주어진 노드쌍간 통신 수요를 링에 할당하는 방법이 정해졌을 때, OXC 위치에 따라 Ring 1의 내부 트래픽이 변하는 2가지 예를 나타낸다. 노드쌍 (2, 5)와 (3, 5)의 통신 수요는 링간(inter-ring) 트래픽으로 처리하고 있으므로 2개의 링에는 모두 OXC를 설치한다. 그림 3 (b)와 같이 Ring 1의 노드 2와 Ring 2의 노드 5에 OXC를 설치하면, 노드쌍 (2, 5)의 수요는 Ring 1과 Ring 2의 내부 트래픽을 증가시키지 않지만, 노드쌍 (3, 5)의 수요는 Ring 1의 내부 트래픽을 증가시켜 Ring 1의 OADM 트래픽이 5가 된다. 반면, 그림 3 (c)와 같이 Ring 1의 OXC 위치를 노드 3으로 바꾸면, 링간 트래픽의 양은 4로 변함없지만 Ring 1의 내부 트래픽이 7로 증가한다. OADM의 용량이 7보다 적다면 그림 3 (c)의 할당 방법은 고려할 수 없으며, OADM 용량이 7이상이면 그림 3 (b)와 (c)의 할당 방법이 모두 가능하므로, 이때에는 OXC간 거리비용이 적은 방법을 선택하는 것이 바람직하다.



(a) 수요 그래프, (b) 노드 2와 노드 5가 허브 노드인 경우, (c) 노드 3과 노드 5가 허브 노드인 경우

[그림 3] OXC 위치에 따라 OADM 트래픽 양이 변화하는 예

1.3 논문 구성

이 논문의 구성은 다음과 같다. 제2장에서는 이 논문에서 제시하는 계층형 광통신망 문제와 관련된 기존연구를 조사하고, 제3장에서는 HRM 문제의 정수계획법 모형을 정의하고, 정수계획법 모형의 특징을 분석하여 branch-and-bound 기법의 해 탐색 속도를 증가시키는 절단평면(cutting planes)을 개발한다. 제4장에서는 사이즈가 큰 문제의 우수한 해를 효과적으로 찾는 휴리스틱을 개발한다. 제5장에서는 실험을 통해 휴리스틱 및 절단평면의 성능을 평가하고, 제6장에서 결론을 맺는다.

2. 기존연구 조사

지난 십수년간 다양한 형태의 광통신망 설계 문제가 다루어져 왔으며, 아래에는 이 논문에서 다루는 HRM 문제와 관련이 있는 SONET-WDM 링 설계에 관한 주요 연구 결과를 소개한다.

ADM 비용을 최소화하는 간단한 형태의 수요 할당 연구에서는 링간 트래픽과 각 링에서의 노드간 거리 비용은 고려하지 않았다[12, 13]. ADM 비용 대신 링의 개수를 최소화하는 수요 할당 문제를 다룬 연구로는 링의 용량만 고려하여 최적해의 2배수 이내의 해를 제공하는 휴리스틱[7] 개발 및 경로 재설정(path relinking)과 분산탐색(scatter search) 기법을 이용한 휴리스틱[1] 개발이 있다. 하나의 노드가 서로 다른 링에 중복 할당되지 않는 수요 할당 문제를 branch-and-price 기법을 이용하여 해결한 연구[14], 용량이 다른 링을 혼합하여 사용할 수 있고 하나의 수요를 분할하여(split) 여러 개의 링에 분산 할당하여 링 구축비용을 최소화하는 문제를 다룬 연구[5]가 있다. 또한, 미리 정해진 허브 노드를 공유하면서 OADM 비용과 허브 연결비용을 최소화하는 문제를 열생성(column generation) 기법으로 해결한 연구도 있으나[9], 각 링의 OADM 개수 제약과 OXC의 용량 제약은 고려하지 않았다. 이 밖에도 링간 트래픽을 고려한 SONET-WDM 링 설계 문제를 다룬 다수의 연구가 있으나[2, 3, 4, 6, 15], 메쉬망에서의 OXC간 거리비용, OADM간 및 OXC간 거리비용을 고려한 연구는 찾아볼 수 없다.

3. 정수계획법 모형

계층형 링메쉬 광통신망 설계문제의 정수계획법 모형 개발에 사용하는 기호를 아래에 나타낸다.

• 집합

- N : 노드 집합,
- E : 수요가 정의된 노드쌍의 집합 (N 과 E 로 구성된 그래프 $G(N, E)$ 는 단절되어 있지 않다고 가정한다),
- K : 링의 집합.

• 파라메타

- $R (\geq 3)$: 하나의 링에 설치할 수 있는 OADM의 최대 개수,
- b^A : OADM의 트래픽 처리 용량,

- b^X : OXC의 트래픽 처리 용량,
- d_{ij} : 노드쌍 i 와 j ($> i$)간 수요,
- a : OADM 설치 비용,
- β : OADM을 OXC로 변경할 때 소요되는 비용(즉, OXC 비용 - OADM 비용),
- g_{ij} : 노드 i 와 j ($> i$)를 동일한 링에 할당할 때 발생하는 거리 비용,
- r_{ij} : 노드 i 와 j ($> i$)가 서로 다른 링의 허브 노드 (OXC 설치 노드)일 때 발생하는 거리 비용.

거리비용 g 와 r 은 두 노드 i 와 j ($> i$)를 동일한 링에 할당하거나 서로 다른 링의 허브 노드로 설정하는 것을 방지하기 위한 페널티로 간주할 수 있다. 가령, 두 노드 i 와 j ($> i$)가 지리적으로 매우 멀리 떨어져 있어 하나의 링으로 연결하면 네트워크 관리에 어려움이 있거나 정확한 비용은 예측할 수 없지만 케이블 비용이 지나치게 증가하는 것이 명확한 경우에는(가령, 서울의 A전화국과 제주도의 B전화국을 하나의 링으로 직접 연결) 해당 g_{ij} 값을 매우 큰 값으로 설정할 수 있다.

• 이진 변수

- x_{ik} : 노드 i 가 링 k 에 할당되면(즉, 링 k 에 할당된 노드 $i \in N$ 에 OADM 또는 OXC를 설치하면) 1, 그렇지 않으면 0,
- v_{ik} : 노드 i 가 링 k 의 허브 노드로 선택되면(즉, 링 k 에 할당된 노드 i 에 OXC를 설치하면) 1, 그렇지 않으면 0,
- f_{ijkl} : 수요 d_{ij} 가 링 k 에 할당된 노드 i 와 j ($> i$)사이의 링 내부(intra-ring) 트래픽으로 처리되면 1, 그렇지 않으면 0,
- f_{ijkl} : 수요 d_{ij} 가 링 k 에 할당된 노드 i 와 링 l ($\neq k$)에 할당된 노드 j ($> i$)사이의 링간(inter-ring) 트래픽으로 처리되면 1, 그렇지 않으면 0,
- u_{ijk} : 수요 d_{ij} 가 링 k 에 관련된 링간 트래픽으로 처리되고($\sum_{l(\neq k) \in K} f_{ijkl} = 1$) 링 k 에 할당된 노드 i 에 OADM이 설치되면(즉, 노드 i 가 링 k 의 허브 노드가 아니면: $x_{ik} = 1, v_{ik} = 0$) 1, 그렇지 않으면 0,
- y_{ijk} : 링 k 에 노드 i 와 j ($> i$)가 모두 할당되면($x_{ik} = 1, x_{jk} = 1$) 1, 그렇지 않으면 0,
- z_{ij} : 노드 i 와 j ($> i$)가 서로 다른 링의 허브 노드이면 ($v_{ik} = 1, v_{jl} = 1: k, l (\neq k) \in K$) 1, 그렇지 않으면 0,
- w_k : 링 k 가 사용되면 1, 그렇지 않으면 0.
링 k 에 설치된 OXC가 처리하는 링간 트래픽의

양은 $\sum_{(i,j) \in E} d_{ij} \sum_{l(\neq k) \in K} (f_{ijkl} + f_{jikl})$ 으로 나타낼 수 있으며, 링 k 에 설치된 OADM이 처리하는 링 내부 트래픽의 양은 $\sum_{(i,j) \in E} d_{ij} f_{jikl} + \sum_{(i,j) \in E} d_{ij} (u_{ijk} + u_{jik})$ 으로 나타낼 수 있다. 여기서 주목할 점은 $u_{ijk} = 1$ 은 수요 d_{ij} 가 링 k 를 포함한 링간 트래픽으로 처리됨을 나타내지만, 노드 i 가 링 k 의 허브 노드가 아니므로 링 k 에 할당된 노드 i 에서 링 k 의 허브 노드 $i' (\neq i)$ 까지 수요 d_{ij} 를 전달하기 위해서는 링 k 의 OADM 용량을 사용하게 된다는 것이다. 따라서, $u_{ijk} = 1$ 인 경우에도 링 k 의 OADM 트래픽은 d_{ij} 만큼 증가한다. $u_{jik} = 1$ 일 때도 $u_{ijk} = 1$ 인 경우와 마찬가지로 링 k 의 OADM 트래픽은 d_{ij} 만큼 증가한다.

한편, 링 k 를 사용할 경우($w_k = 1$), 링 구조의 SONET 망 구성을 위해서는 최소 3개의 OADM 및 OXC가 링 k 에 설치되어야 한다. 만일, 임의의 링 k 에서 하나의 수요 d_{ij} 만 링내 트래픽으로 처리하는 해가 존재하면, 링 k 에는 두개의 OADM만 설치하면 된다. 그러나, 이 경우 최소 3개의 OADM 및 OXC 설치해야 하는 제약 조건을 만족시키기 위해 트래픽 처리가 수반되지 않는 임의의 노드 $i' \in N \setminus \{i, j\}$ 를 링 k 에 할당하는 해가 생성될 수 있다. 이와 같은 의미 없는 해를 방지하기 위해서는 링 k 에 노드 i 가 할당되면 노드 i 와 관련된 수요가 반드시 링 k 에서 처리되도록 해야 하며, 다음과 같은 식으로 표현할 수 있다:

$$\sum_{j \in N: (i,j) \in E} \sum_{l \in K} f_{ijkl} + \sum_{j \in N: (j,i) \in E} \sum_{l \in K} f_{jilk} \geq x_{ik} \quad i \in N, k \in K.$$

위에 설명한 주요 제약조건을 포함한 HRM 문제는 다음과 같은 정수계획법 모형으로 나타낼 수 있다.

$$\text{HRM: Minimize } \sum_{i \in N} \sum_{k \in K} (a x_{ik} + \beta v_{ik}) + \sum_{i,j(>i) \in N} \sum_{k \in K} g_{ij} y_{ijk} + \sum_{i,j(>i) \in N} r_{ij} z_{ij}$$

Subject to

$$\sum_{k \in K} \sum_{l \in K} f_{ijkl} = 1 \quad (i, j) \in E, \quad (1)$$

$$f_{ijkl} \leq x_{ik} \quad (i, j) \in E, k, l \in K, \quad (2)$$

$$f_{ijkl} \leq x_{jl} \quad (i, j) \in E, k, l \in K, \quad (3)$$

$$\sum_{i \in N} x_{ik} \leq R w_k \quad k \in K, \quad (4)$$

$$\sum_{i \in N} v_{ik} \leq w_k \quad k \in K, \quad (5)$$

$$v_{ik} \leq x_{ik} \quad i \in N, k \in K, \quad (6)$$

$$\sum_{(i,j) \in E} d_{ij} (f_{ijkk} + u_{ijk} + u_{jik}) \leq b^A w_k \quad k \in K, \quad (7)$$

$$u_{ijk} \geq \sum_{l(\neq k) \in K} f_{ijkl} - v_{ik} \quad (i, j) \in E, k \in K, \quad (8)$$

$$u_{ijk} \geq \sum_{l(\neq k) \in K} f_{ijlk} - v_{jk} \quad (i, j) \in E, k \in K, \quad (9)$$

$$\sum_{(i,j) \in E} d_{ij} \sum_{l(\neq k) \in K} (f_{ijlk} + f_{ijlk}) \leq b^x \sum_{i \in N} v_{ik} \quad k \in K, \quad (10)$$

$$\sum_{i \in N} x_{ik} \geq 3w_k \quad k \in K, \quad (11)$$

$$\sum_{j \in N: (i,j) \in E} \sum_{l \in K} f_{ijlk} + \sum_{j \in N: (j,i) \in E} \sum_{l \in K} f_{jilk} \geq x_{ik} \quad i \in N, k \in K, \quad (12)$$

$$x_{ik} + x_{jk} \leq y_{ijk} + 1 \quad i, j (> i) \in N, k \in K, \quad (13)$$

$$v_{ik} + v_{jl} \leq z_{ij} + 1 \quad i, j (> i) \in N, k, l (\neq k) \in K, \quad (14)$$

$$f, x, u, v, y, z, w \in \{0, 1\}. \quad (15)$$

제약식 (1)은 모든 수요가 링에 할당되어야 함을 나타낸다. 제약식 (2)와 (3)은 수요 d_{ij} 가 링 k 와 l 에서 처리되면($f_{ijlk} = 1$), 링 k 에 포함된 노드 i 와 링 l 에 포함된 노드 j 에 OADM을 설치해야 함을 나타낸다. 제약식 (4)는 각 링에는 R 개 이하의 OADM 및 OXC를 설치할 수 있음을 나타낸다. 제약식 (5)는 각 링에 설치하는 OXC 수가 1을 넘지 못함을 나타내며, 제약식 (6)은 OXC 설치 위치는 해당 링에 OADM이 설치된 노드중에서 선택해야 함을 나타낸다. 일반적으로 OADM보다 OXC가 비싸므로, β 는 OXC 비용과 OADM 비용의 차이로 생각할 수 있다. 제약식 (7) ~ (9)는 각 링에서 내부 트래픽으로 처리하는 수요는 OADM의 처리용량 b^A 보다 클 수 없음을 나타낸다. 제약식 (10)은 각 링에서 링간 트래픽으로 처리하는 수요는 OXC의 처리용량 b^X 보다 클 수 없음을 나타낸다. 제약식 (11)은 임의의 링 k 를 사용할 경우 링 k 에 할당된 3개 이상의 노드에 OADM 및 OXC를 설치해야 함을 나타낸다. 제약식 (12)는 OADM이 링 k 의 노드 i 에 설치될 경우, 노드 i 와 관련된 수요가 반드시 링 k 에서 처리되어야 함을 나타낸다. 제약식 (13)은 링 k 에 할당된 노드 i 와 $j (> i)$ 에 모두 OADM이 설치되면, 링 k 구축 시 발생하는 노드 i 와 $j (> i)$ 의 연결 비용을 표현하며, 제약식 (14)는 링 k 에 포함된 노드 i 와 링 $l (\neq k)$ 에 포함된 노드 $j (> i)$ 에 모두 OXC가 설치되면, 메쉬망 구축시 발생하는 노드 i 와 $j (> i)$ 간 OXC 연결 비용을 표현 한다. 제약식 (15)는 모든 변수가 0 또는 1의 값만을 가질 수 있음을 나타낸다.

정수계획법 모형 HRM은 NP-hard이다. 왜냐하면, 모형 HRM에서 $b^X = 0, g_{ij} = 0 \forall i, j (> i) \in N$ 으로 설정하면 NP-hard 문제로 알려진 SONET 문제[12]가 되기 때문이다.

3. 모형 분석

모형 HRM에서 f, v 와 w 변수만 이진(binary) 변수로 선

언하여도, 최적해에서 x, u, y 및 z 변수는 자동으로 이진 값을 갖는다. 또한, f 와 v 변수의 값만 주어지면 w 변수의 값은 자동적으로 결정된다.

Remark 1. 제약식 (2)와 (3)은 다음과 같이 나타낼 수 있으며,

$$\sum_{l \in K} f_{ijlk} \leq x_{ik} \quad (i, j) \in E, k \in K, \quad (16)$$

$$\sum_{l \in K} f_{ijlk} \leq x_{jk} \quad (i, j) \in E, k \in K, \quad (17)$$

식 (16)과 (17)로부터 다음 관계가 성립함을 알 수 있다.

$$\sum_{i \in N} \sum_{k \in K} x_{ik} \geq |N|. \quad (18)$$

또한, 제약식 (11)로부터 모든 가능해가 만족하는 식 (19)를 구할 수 있으며

$$\sum_{i, j (> i) \in N} y_{ijk} \geq 3w_k \quad k \in K, \quad (19)$$

식 (19)는 다음과 같이 개선할 수 있다.

$$\sum_{i, j (> i) \in N} y_{ijk} \geq 3/2 (\sum_{i \in N} x_{ik} - 1) \quad k \in K. \quad (20)$$

만일, $|K| < |N|/R$ 이면 HRM의 선형완화(LP-relaxation) 모형의 가능해도 존재하지 않으므로, 이 논문에서는 $|K| \geq |N|/R$ 을 가정한다. 한편, $|M|/R \leq |K| < \sum_{(i,j) \in E} d_{ij}/b^A$ 이면, 최소 2개 이상의 링에 OXC를 설치해야 하므로 식 (21)을 만족하며,

$$\sum_{i \in N} \sum_{k \in K} v_{ik} \geq 2, \quad (21)$$

제약식 (14)로부터 식 (22)를 구할 수 있다.

$$\sum_{i, j (> i) \in N} z_{ij} \geq \sum_{i \in N} \sum_{k \in K} v_{ik} - 1. \quad (22)$$

NP-hard인 HRM 문제를 더욱 어렵게 만드는 요인은 링의 대칭성(symmetry)이다. 링의 대칭성을 제거하기 위해 식 (23) 또는 (24)를 적용할 수 있으며, 식 (23)과 (24)를 만족하는 최적해가 반드시 존재함을 알 수 있다.

$$w_k \leq w_{k+1} \quad k = 1, \dots, |K| - 1, \quad (23)$$

$$\sum_{i \in N} x_{ik} \leq \sum_{i \in N} x_{i, k+1} \quad k = 1, \dots, |K| - 1. \quad (24)$$

식 (23)과 달리 식 (24)는 링의 사용 여부뿐만 아니라 링에 설치하는 OADM 개수까지 링별로 정렬한다. 따라서, 링의 대칭성 제거에는 식 (24)가 식 (23)보다 효과적인 것으로 예상된다. 아래에는 식 (24)로부터 몇 가지 절단평면을 추가로 개발한다.

Proposition 1. $w = |M|/|K|$ 로 정의하자. $\lfloor w \rfloor < w$ 이면, 식 (24)를 만족하는 모든 가능해는 식 (25)를 만족한다.

$$(1 - \lfloor w \rfloor) \sum_{i \in N} x_{i,1} + \lfloor w \rfloor \sum_{i \in N} x_{i,|K|} \geq \lfloor w \rfloor. \quad (25)$$

Proof. 식 (25)는 식 (26)으로 표현할 수 있다.

$$\sum_{i \in N} x_{i,1} \geq \lfloor w \rfloor (\sum_{i \in N} x_{i,1} - \sum_{i \in N} x_{i,|K|+1}). \quad (26)$$

$\sum_{i \in N} x_{i,|K|} > \sum_{i \in N} x_{i,1}$ 이면 식 (26)이 성립함은 명백하므로 $\sum_{i \in N} x_{i,|K|} = \sum_{i \in N} x_{i,1}$ 이라고 가정한다. 이때, 식 (26)은 $\sum_{i \in N} x_{i,1} \geq \lfloor w \rfloor$ 으로 표현된다. 나머지 증명에서는 $\sum_{i \in N} x_{i,1} < \lfloor w \rfloor$ 을 만족하는 가능해가 존재하지 않음을 보인다. $\lfloor w \rfloor < w$ 이므로 $\lfloor w \rfloor = \lfloor w \rfloor + 1$ 이다. $\sum_{i \in N} x_{i,1} \leq \lfloor w \rfloor$ 을 만족하는 가능해가 존재하면, $\sum_{i \in N} x_{i,|K|} = \sum_{i \in N} x_{i,1}$ 이므로 $\sum_{i \in N} \sum_{k \in K} x_{ik} \leq \lfloor w \rfloor \times |K|$ 을 만족한다. 또한, $\lfloor w \rfloor \times |K| < |M|$ 이므로(왜냐하면 $\lfloor w \rfloor < w$ 이므로), $\sum_{i \in N} \sum_{k \in K} x_{ik} < |M|$ 이 성립해야 한다. 그러나, 이 결과는 식 (18)과 상충된다. 즉, $\sum_{i \in N} x_{i,1} < \lfloor w \rfloor$ 을 만족하는 가능해는 존재하지 않으며, $\sum_{i \in N} x_{i,|K|} = \sum_{i \in N} x_{i,1}$ 일 때 모든 가능해는 $\sum_{i \in N} x_{i,1} \geq \lfloor w \rfloor$ 을 만족함을 의미한다.

Proposition 1의 결과를 부연 설명하면, 식 (24)를 모형 HRM에 적용하지 않더라도 식 (25)를 만족하는 최적해가 반드시 존재한다는 것이다. 즉, 식 (25)만 적용하여도 식 (24)를 만족하는 해를 찾을 수 있다. 아래에서는 식 (25)로부터 새로운 절단평면을 유도하기 위해 최소의 링 개수 $MinK$ 를 찾는 문제를 고려한다.

$$MinK = \text{Minimize } \sum_{k \in K} w_k$$

Subject to

$$\sum_{k \in K} \sum_{l \in K} f_{ijkl} = 1 \quad (i, j) \in E, \quad (25)$$

$$f_{ijkl} \leq x_{ik} \quad (i, j) \in E, k, l \in K, \quad (26)$$

$$f_{ijkl} \leq x_{jl} \quad (i, j) \in E, k, l \in K, \quad (27)$$

$$\sum_{i \in N} x_{ik} \leq R w_k \quad k \in K, \quad (28)$$

$$\sum_{i \in N} v_{ik} \leq w_k \quad k \in K, \quad (29)$$

$$v_{ik} \leq x_{ik} \quad i \in N, k \in K, \quad (30)$$

$$\sum_{(i,j) \in E} d_{ij} (f_{ijkk} + u_{ijk} + u_{jik}) \leq b^A w_k \quad k \in K, \quad (31)$$

$$u_{ijk} \geq \sum_{l (\neq k) \in K} f_{ijkl} - v_{ik} \quad (i, j) \in E, k \in K, \quad (32)$$

$$u_{jik} \geq \sum_{l (\neq k) \in K} f_{jikl} - v_{jk} \quad (i, j) \in E, k \in K, \quad (33)$$

$$\sum_{(i,j) \in E} d_{ij} \sum_{l (\neq k) \in K} (f_{ijkl} + f_{jikl}) \leq b^X \sum_{i \in N} v_{ik} \quad k \in K, \quad (34)$$

$$\sum_{i \in N} x_{ik} \geq 3 w_k \quad k \in K, \quad (35)$$

$$\sum_{j \in N: (i,j) \in E} \sum_{l \in K} f_{ijkl} + \sum_{j \in N: (j,i) \in E} \sum_{l \in K} f_{jikl} \geq x_{ik} \quad i \in N, k \in K, \quad (36)$$

$$\sum_{i \in N} x_{ik} \leq \sum_{i \in N} x_{i,k+1} \quad k = 1, \dots, |K| - 1. \quad (37)$$

$$x, f, u, v, w \in \{0, 1\}.$$

$MinK$ 를 찾는 위 문제에서 식 (25) ~ (36)은 식 (1) ~ (12)와 동일하며, 식 (37)은 식 (24)에 나타난 링의 대칭현상을 제거하기 위한 제약식을 표현한다. $MinK$ 를 찾는 문제는 일반적으로 쉽게 풀 수 없는 조합최적화 문제이다. 따라서, 이진변수 w 를 제외한 나머지 변수를 모두 선형완화한 문제의 목적함수 값을 $MinK'$ 로 나타내면, $MinK'$ 는 비교적 쉽게 구할 수 있다. 즉, $MinK'$ 를 찾기 위해 위 문제에서 모든 변수를 선형완화한 다음 $w_k = 1$ 인 링의 개수를 1부터 $|K|$ 까지 하나씩 증가시키면서 가능해를 구할 때까지 선형완화한 모형의 최적해를 구하면 된다. 이 경우, 최대 $|K|$ 번의 선형계획법 문제를 풀면 $MinK'$ 값을 확인할 수 있다. $MinK$ 또는 $MinK'$ 값으로부터 다음과 같은 새로운 절단평면을 유도할 수 있다.

Proposition 2. 모든 가능해는 다음 식 (38)을 만족한다.

$$\sum_{i \in N} \sum_{k \in K} (x_{ik} + v_{ik}) \geq |N| + MinK - 1. \quad (38)$$

Proof. 식 (18)에 의해 $\sum_{i \in N} \sum_{k \in K} x_{ik} \geq |M|$ 임을 알 수 있고, $MinK$ 의 정의에 따라 최소 $MinK$ 개의 링을 사용한다. 또한, 그래프 $G(N, E)$ 가 연결되어 있으므로 $MinK$ 개의 링을 사용하면 최소 $MinK - 1$ 개의 OXC 및 $|M|$ 개 이외의 OADM이 추가로 설치되어야 함을 알 수 있다.

Proposition 3. $\zeta = (|M| + MinK - 1)/|K|$ 로 정의하자. $\lfloor \zeta \rfloor < \zeta$ 이면, 식 (24)를 만족하는 모든 가능해는 다음 식 (39)를 만족한다.

$$(1 - \lfloor \zeta \rfloor) \sum_{i \in N} x_{i,1} + \lfloor \zeta \rfloor \sum_{i \in N} x_{i,|K|} + \lfloor \zeta \rfloor / 2 \sum_{i \in N} \sum_{k \in K} v_{ik} \geq \lfloor \zeta \rfloor. \quad (39)$$

Proof. 식 (39)는 식 (40)으로 나타낼 수 있다.

$$\sum_{i \in N} x_{i,1} + \lfloor \zeta \rfloor / 2 \sum_{i \in N} \sum_{k \in K} v_{ik} \geq \lfloor \zeta \rfloor (\sum_{i \in N} x_{i,1} - \sum_{i \in N} x_{i,|K|+1}). \quad (40)$$

$\sum_{i \in N} x_{i,1} < \sum_{i \in N} x_{i,|K|}$ 이면, 식 (40)의 우변은 최대 값이 0이므로 식 (40)이 성립함은 명백하다. 따라서, $\sum_{i \in N} x_{i,1} = \sum_{i \in N} x_{i,|K|}$ 이라고 가정하면, 식 (40)은 식 (41)로 표현할 수 있다.

$$\sum_{i \in N} x_{i,1} + \lceil \zeta \rceil / 2 \sum_{i \in N} \sum_{k \in K} v_{ik} \geq \lceil \zeta \rceil. \quad (41)$$

또한, $\sum_{i \in N} \sum_{k \in K} v_{ik} > 0$ (즉, $\sum_{i \in N} \sum_{k \in K} v_{ik} \geq 2$)이면, 식 (41)이 성립함은 명백하므로 $\sum_{i \in N} \sum_{k \in K} v_{ik} = 0$ 이라고 가정한다. 이 때, 식 (41)은 $\sum_{i \in N} x_{i,1} \geq \lceil \zeta \rceil$ 이 된다. 나머지 증명에서는 $\sum_{i \in N} x_{i,1} = \sum_{i \in N} x_{i,|K|}$ 이고 $\sum_{i \in N} \sum_{k \in K} v_{ik} = 0$ 일 때 $\sum_{i \in N} x_{i,1} < \lceil \zeta \rceil$ 을 만족하는 가능해가 존재하지 않음을 보인다. $\lfloor \zeta \rfloor < \zeta$ 라고 가정하였으므로 $\lceil \zeta \rceil = \lfloor \zeta \rfloor + 1$ 이다. $\sum_{i \in N} x_{i,1} \leq \lfloor \zeta \rfloor$ 을 만족하는 가능해가 존재하면, $\sum_{i \in N} x_{i,1} = \sum_{i \in N} x_{i,|K|}$ 이므로 $\sum_{i \in N} \sum_{k \in K} x_{ik} \leq \lfloor \zeta \rfloor \times |K|$ 을 만족한다. 또한, $\lfloor \zeta \rfloor \times |K| < |N| + \text{Min}K - 1$ 이므로(왜냐하면 $\lfloor \zeta \rfloor < \zeta$ 이므로), $\sum_{i \in N} \sum_{k \in K} x_{ik} < |N| + \text{Min}K - 1$ 이 성립해야 한다. 그러나, 이 결과는 식 (38)과 상충된다. 즉, $\sum_{i \in N} x_{i,1} < \lceil \zeta \rceil$ 을 만족하는 가능해는 존재하지 않으며, $\sum_{i \in N} x_{i,1} = \sum_{i \in N} x_{i,|K|}$ 이고 $\sum_{i \in N} \sum_{k \in K} v_{ik} = 0$ 일 때 모든 가능해는 $\sum_{i \in N} x_{i,1} \geq \lceil \zeta \rceil$ 을 만족함을 의미한다.

MinK를 구할 수 없을 때 Proposition 2와 Proposition 3에서 MinK대신 MinK'값을 사용하여도 식 (38)과 식 (39)는 유효(valid)하다.

4. 휴리스틱

현재의 가능해를 (f^o, x^o, v^o) 로 나타내고 집합 K^X 와 $N(k)$ 을 다음과 같이 정의한다: $K^X = \{k \in K: \sum_{i \in N(k)} v^o_{ik} = 1\}$, $N(k) = \{i \in N: x^o_{ik} = 1\} \quad \forall k \in K$.

4.1 초기해 발견 절차

Step 0. $k = 1, L = E, V = \emptyset$.

Step 1. OADM 용량 제약과 개수 제약을 고려하여 L 에 속한 가능한 많은 수의 수요 (i, j) 를 링 k 에 할당한다. 이때, g_{ij} 가 작은 (g_{ij} 가 같으면 d_{ij} 가 큰) (i, j) 를 먼저 할당한다. 링 k 에 할당한 수요는 L 에서 삭제한다.

Step 2. 링을 추가한다($k = k + 1$). 추가한 링 k 에 대해서 Step 1의 절차를 수행하고 Step 3으로 이동한다.

Step 3. OADM 용량 제약과 개수 제약 및 OXC 용량 제약을 고려하여 L 에 속한 가능한 많은 수의 (i, j) 를 기존에 설치된 임의의 두 링 k 와 l 에 링간 트래픽으로 할당한다. 이때, OADM을 추가할 수 있으면 r_{ij} 가 작은 수요를 먼저 할당한다. r_{ij} 가 같으면 d_{ij} 가 큰 수요를 먼저 할당한다. 각 링의 OXC 위치는 해당 링에 최초로 할당되는 링간 트래픽 (i, j) 의 수요 노드(end node) i 와 j 로 정한다. 링간 트래픽으로 할당한 모든 수요를 L 에서 삭제한다.

Step 4. $L \neq \emptyset$ 이면 Step 2로 이동한다. $L = \emptyset$ 이면 모형 HL의 최적해를 구하여 가능해 (f^o, x^o, v^o) 의 v^o 값을 갱신하고 종료한다.

HL: Minimize $\sum_{i, j (> i) \in N} r_{ij} z_{ij}$

Subject to

$$\sum_{i \in N(k)} v_{ik} = 1 \quad k \in K^X, \quad (42)$$

$$v_{ik} + v_{jl} \leq z_{ij} + 1 \quad i \in N(k),$$

$$j (> i) \in N(l), k, l (\neq k) \in K^X, \quad (43)$$

$$\begin{aligned} & \sum_{i \in N(k)} \sum_{j \notin N(k): (i,j) \in E} \sum_{l (\neq k) \in K} d_{ij} f^o_{ijkl} (1 - v_{ik}) \\ & + \sum_{j \notin N(k): (j,i) \in E} \sum_{l (\neq k) \in K} d_{ji} f^o_{jilk} (1 - v_{ik}) \\ & \leq b^A - \sum_{(i,j) \in E} d_{ij} f^o_{ijkk} \quad k \in K^X, \quad (44) \end{aligned}$$

$$v, z \in \{0, 1\}.$$

모형 HL에서 제약식 (42)와 (43)은 쉽게 이해할 수 있는 부분이므로 설명을 생략한다. 제약식 (44)는 링간 트래픽이 존재하는 링마다 OXC를 설치하는 위치에 따라 각 링의 내부 트래픽으로 처리할 수요가 달라질 수 있으므로, 내부 트래픽의 양이 OADM의 처리용량을 만족시킬 수 있는 위치에 OXC를 설치해야 함을 나타낸다. 초기해 발견 절차의 Step 4에서 링별 최적의 OXC 위치 선정을 위해 모든 조합을 검사하는 방법(total enumeration)은 비현실적이다. 왜냐하면, 링의 수($|K|$)가 증가하면 링별 OXC 위치의 가능한 조합의 수는 최대 $|N(1)| \times |N(2)| \dots |N(|K|)$ 개로 기하급수적으로 증가하기 때문이다. 반면, 모형 HL 역시 NP-hard 유형에 속하는 조합최적화 문제이지만 모형 HL의 최적해를 구하여 링별 최적의 OXC 위치를 결정하는 방법은 링의 수가 많더라도 현실적인 조건에서는(예를 들면 $|K| < 100$), 매우 효과적이다. 초기해 발견 절차에서는 거리 비용 g 와 r 을 최소화하는 방향으로 수요를 할당하였으나, 지역해 개선(local improvement) 절차에서는 비용 α 와 β 를 포함한 총비용을 최소화하는 방향으로 해를 변화시킨다. 또한, 초기해 절차에서는 링별 OADM 및 OXC 개수의 최소 개수 제약을 위반할 수

있으나, 아래에 설명하는 지역해 개선 절차에서는 가능해를 생성하도록 설계한다.

4.2 지역해 개선 절차

각 링에 할당된 수요를 변경하기 위해 서브루틴 Relocation(ij, k, l)를 정의하며, Relocation(ij, k, l)은 다음과 같이 2가지 경우로 구분한다.

(a) ($f_{ijk} = 1$ or $f_{ijl} = 1$ or $f_{jlk} = 1$) \rightarrow $f_{jil} = 1$: OADM 위치를 확인하여 $N(k)$ 과 $N(l)$ 을 갱신하고, 최대 $|N(k)| \times |N(l)|$ 개의 경우의 수를 검토하여 r 비용을 최소화하는 링 k 와 l ($\neq k$)의 OXC 위치를 결정한다.

(b) ($f_{ijk} = 1: l' \in K \setminus \{k, l\} \rightarrow f_{jil'} = 1$ or ($f_{jlk} = 1: l' \in K \setminus \{k, l\} \rightarrow f_{jil'} = 1$: 이 경우, 링 l' 에는 아무런 변화도 없다. 따라서, (a)경우와 마찬가지로 $N(k)$ 과 $N(l)$ 만을 갱신하고 r 비용을 최소화하는 링 k 와 l ($\neq k$)의 OXC 위치를 결정한다.

Relocation(ij, k, l)을 수행하면 OADM 및 OXC 트래픽이 링 k 에서는 감소하며 링 l 에서는 증가한다. 한편, (b)의 경우에는 링 l' 의 OADM 및 OXC 트래픽의 변화가 없으므로 링 k 에서는 제약식 (11)을, 링 l 에서는 제약식 (4), (7), (10)을 검사하여 가능해 생성 여부를 확인해야 한다. 가능해를 생성하는 Relocation(ij, k, l)을 찾기 위해서는 최대 $|K|^2 \times |E|$ 회의 연산이 필요하며, 수요 할당이 정해졌을 때 OXC 위치 최적화에는 최대 R^2 ($\geq |N(k)| \times |N(l)|$) 회의 추가적인 연산이 필요하므로 Relocation(ij, k, l)을 1회 수행하기 위해서는 최대 $R^2 \times |K|^2 \times |E|$ 회의 연산이 필요하다. 그러나, OXC 위치 최적화는 수요 할당 변경으로 가능해를 구한 경우에만 수행하며 링 k 의 OADM 용량 제약을 만족하는 OXC 설치 가능 위치는 $|N(k)|$ 보다 클 수 없으며, 일반적으로 실제 연산 횟수는 $R^2 \times |K|^2 \times |E|$ 보다 훨씬 적다. 아래에는 Relocation(ij, k, l)을 이용하여 지역해를 개선하기 위한 절차를 나타낸다.

Step 0. $cnt = 0$.

Step 1(Backward Improvement I). 두 링 k 와 l ($l < k$)에 할당된 수요 (i, j)의 모든 조합에 대해서 총비용을 줄이는 임의의 Relocation($ij, k, l: l < k$)을 수행한다. 이때, $k, (i, j), l$ ($l < k$)의 순으로 선택하며, k 는 클수록, l 은 작을수록 먼저 검사한다. k 가 정해지면 (i, j)는 임의의 순서로 검사한다. 총비용을 줄이는 Relocation($ij, k, l: l < k$)이 없으면 Step 2로 이동하고, 그렇지 않으면 $cnt = 0$ 으로 설정하고 Step 1을 한번 더 수행한다.

Step 2(Backward Improvement II). 총비용 증가가 최소인 Relocation($ij, k, l: l < k$)을 수행하고 OXC가 설치된

링만을 대상으로 수확모형 HL의 최적해를 구하여 최적의 OXC 위치를 결정한다. 이 때 총비용이 감소하면 $cnt = 0$ 으로 설정하고 Step 1으로 이동하며, 그렇지 않으면 $cnt = cnt + 1$ 로 설정한다. $cnt = cntMax$ 이거나 수행시간이 제한시간($timeLimit$)을 초과하였으면 종료하고, 그렇지 않으면 Step 3으로 이동한다.

Step 3(Forward Diversification). $f_{ijk} = 1$ 을 만족하는 임의의 수요 (i, j)와 링 k 를 선택하여 $f_{jil} = 1$ ($l > k$)로 변경하고 Step 1로 이동한다. 이때, 링 l 은 가능해 조건을 만족하는 최소 인덱스를 나타낸다. 만일, 기존 링 중에서 가능해 조건을 만족하는 링 l ($> k$)이 없으면 l 은 새로운 링을 나타낸다.

Step 1에서는 링 인덱스가 큰 쪽에서 작은 쪽으로 수요 할당을 변경하며, Step 3에서는 링 인덱스가 작은 쪽에서 큰 쪽으로 수요 할당을 변경한다. 따라서, Step 3에서는 새로운 링이 생성될 수 있지만, Step 1에서 링의 수가 감소할 수 있다.

5. 실험 결과

이 장에서는 HRM 문제의 해를 구하기 위해 이 논문에서 개발한 휴리스틱 알고리즘의 성능을 평가한다. 이를 위해 다양한 크기의 문제를 랜덤으로 생성하였으며, 파라메타는 다음과 같이 설정하였다.

Step 1. $R = 8, b^A = 64, b^X = 64, d_{ij} = \text{Uniform}[0, 5], a = 10, \beta = 10$.

Step 2. $|N|$ 개의 노드를 100×100 평면에 랜덤하게 분포시키고, 모든 노드쌍 간의 거리 $\text{distance}(i, j)$ 를 계산한다. 거리 $\text{distance}(i, j)$ 가 50보다 큰 노드쌍만을 대상으로 거리 비용 g 와 r 을 다음과 같이 설정한다: $g_{ij} = r_{ij} = wgt \times \lceil \text{distance}(i, j)/50 \rceil$.

wgt 는 OADM 비용 $a = 10$ 을 기준으로 거리 비용 g 와 r 값의 크기를 조절하기 위한 가중치로서 1과 20 두 가지 값을 사용하였다. 즉, $wgt = 1$ 인 경우에는 OADM과 OXC 비용의 합을 우선적으로 최소화하며, 이 값이 동일할 경우에 거리비용을 최소화하는 의미로 해석할 수 있으며, $wgt = 20$ 인 경우에는 거리비용을 우선 최소화하며, 거리비용이 동일한 경우에 OADM과 OXC비용을 최소화하는 해를 선호하는 것으로 해석할 수 있다.

휴리스틱 알고리즘은 C언어로 코딩하였으며, 성능 비

교를 위해 최적화 소프트웨어 CPLEX 버전 10.0을 사용하였다. 모든 실험은 Pentium IV 2.8 GHz PC에서 수행하였다. 휴리스틱 알고리즘에는 *cntMax*와 *timeLimit* 두 가지 파라미터를 사용하며, 이 값은 예비실험을 실시한 후 다음과 같이 설정하였다. *cntMax* = 10, *timeLimit* = 1,000 초. 즉, 휴리스틱 알고리즘의 수행시간은 문제마다 1,000 초로 설정하였는데, 그 이유는 이 논문에서 실험한 가장 큰 사이즈의 문제에서 휴리스틱 알고리즘을 1,000초 동안 수행한 결과 충분히 우수한 품질의 해를 구할 수 있었기 때문이다. 휴리스틱 알고리즘의 수행도 평가를 위해서 CPLEX는 최적해를 구하는 것이 바람직한데, 사이즈가 큰 문제는 수 십 시간이 지나도 최적해를 확인할 수 없는 경우가 대부분이기 때문에 현실적인 시간 여건을 고려하여 CPLEX의 수행시간을 문제마다 10,000초(약 2시간 47분)로 제한하였다. 실험결과는 표 1 ~ 표 6에 나타내며, 표 1 ~ 표 6에 나타낸 기호 설명은 다음과 같다.

- Lower bound HRM: 모형 HRM에서 정수 조건을 완화한 모형의 최소의 목적함수 값,
- Lower bound EHRM: 모형 HRM의 제약식 (2)와 (3)을 식 (16)과 (17)로 바꾸고, 링 대칭성을 제거하기 위한 식 (24)와 (39)를 추가한 모형 EHRM (Enhanced HRM)에서 정수 조건을 완화한 모형의 최소의 목적함수 값,
- Lower bound Final: 모형 HRM 및 EHRM의 최적해를 구하기 위해 10,000초 동안 CPLEX를 수행하는 과정에서 확인한 가장 큰 값의 LP-relaxation Lower bound 값,
- Upper bound HRM: CPLEX를 이용하여 10,000초 동안 모형 HRM를 돌렸을 때 구한 가장 좋은 해의 목적함수 값,
- Upper bound EHRM: CPLEX를 이용하여 10,000초 동안 모형 EHRM를 돌렸을 때 발견한 가장 좋은 해의 목적함수 값,
- Upper bound Heuristic: 휴리스틱 알고리즘을 1,000초 동안 수행하여 구한 최선해(best solution)의 목적함수 값,
- Gap = 100% × (Upper bound Heuristic - Lower bound Final) / Lower bound Final,
- Ratio = 100% × (Upper bound Heuristic - Min{Upper bound HRM, Upper bound EHRM}) / Min{Upper bound HRM, Upper bound EHRM}.

표 1과 표 2에는 비교적 크기가 작은 20문제에 대한 실험결과를 나타낸다. OADM비용(α) 및 OXC 비용(β)과

비교하여 OADM간 거리비용(g) 및 OXC간 거리비용(r)의 상대적인 크기에 따른 실험결과의 차이를 확인하기 위해 각 문제마다 거리비용에 대한 가중치 $wgt = 1$ 일 때의 실험결과를 표 1에 표시하며, $wgt = 20$ 일 때의 실험결과는 표 2에 표시한다. 즉, 동일한 문제지만 표 2에는 거리비용 g 와 r 의 값을 표 1의 문제보다 20배 크게 설정하였을 때 얻은 실험결과를 나타낸다.

표 1에서 HRM과 EHRM의 ‘CPU Time’을 비교하면, HRM은 20문제 중에서 17문제의 최적해를 10,000초 이내에 확인하였으나 EHRM은 20문제 모두 최적해를 확인하였다. 또한, EHRM이 최적해를 찾는 데 걸리는 시간은 HRM의 약 10분의 1로 단축시켰다. 반면, $wgt = 20$ 로 증가시킨 표 2에서도 HRM은 20문제 중에서 17문제의 최적해를 10,000초 이내에 확인하였으나 EHRM은 20문제 모두 최적해를 확인하였다. $wgt = 20$ 인 경우, 모형 EHRM을 이용할 때 최적해를 찾는 데 걸리는 시간이 오히려 증가한 경우도 있으며 (#1, #9, #10, #13, #20), 모형 EHRM이 최적해를 찾는 데 걸리는 시간은 HRM의 약 2.4분의 1 수준이다. 표 1과 표 2의 실험결과만 비교하면, wgt 가 작을 때 식 (24)와 (39)에 의한 링 대칭성 제거 효과가 뚜렷함을 알 수 있다. 반면, 휴리스틱 알고리즘의 성능은 표 1과 표 2에서 모두 Gap = 3%를 넘지 않으며, 특히 20문제중에서 표 1에서는 12문제, 표 2에서는 14문제의 최적해를 발견하였다.

표 3과 표 4에는 크기를 증가시킨 20문제의 실험결과를 제시한다. 앞의 경우와 마찬가지로, $wgt = 1$ 일 때의 실험결과는 표 3에 표시하며, $wgt = 20$ 일 때의 실험결과는 표 4에 표시한다. 표 3과 표 4에서는 제한시간 10,000초 동안 최적해를 확인할 수 없었다. 그러나, 표 3과 표 4의 ‘Ratio’를 살펴보면, HRM 및 EHRM의 해보다 휴리스틱 알고리즘이 제공하는 해가 최대 9.8%나 비용을 감소시킨다. 또한, HRM의 가능해가 EHRM의 가능해보다 우수한 경우는 표 3에서는 단 2문제(#16, #20), 표 4에서는 단 4문제(#3, #4, #7, #17)에 불과하다. 즉, 비교적 크기가 큰 문제에서도 HRM보다 EHRM이 주어진 시간 동안 더 좋은 해를 얻는데 도움이 됨을 알 수 있다.

표 5와 표 6에는 표 3과 표 4에서 다른 문제보다 더 큰 문제의 실험결과를 나타낸다. 이 경우에도 제한시간 10,000초 동안 단 한 문제의 최적해도 확인할 수 없었으며, HRM 및 EHRM의 최선해보다 휴리스틱 알고리즘의 해가 우수함을 ‘Ratio’ 값으로부터 알 수 있다. HRM 및 EHRM의 최선해와 비교할 때 휴리스틱 해의 개선 효과는 최대 19%에 이른다.

[표 1] 실험결과: $|N| = 20$, $|E| = 30$, $|K| = 5$, $wgt = 1$

No	Lower bound			Upper Bound			Gap	Ratio	CPU Time(Seconds)		
	HRM	EHRM	Final	HRM	EHRM	Heuristic			HRM	EHRM	Heuristic
1	240	240	286	286	286	286	0%	0%	2464	91	1K
2	240	240	290	290	290	290	0%	0%	10K	1349	1K
3	240	240	293	293	293	297	1.36%	1.36%	2007	119	1K
4	240	240	284	284	284	291	2.46%	2.46%	1591	126	1K
5	240	240	290	290	290	296	2.06%	2.06%	7982	1402	1K
6	240	240	286	286	286	286	0%	0%	1270	206	1K
7	240	240	284	284	284	289	1.76%	1.76%	562	178	1K
8	240	240	279	279	279	279	0%	0%	651	90	1K
9	240	240	288	288	288	288	0%	0%	512	187	1K
10	240	240	292	292	292	298	2.05%	2.05%	382	110	1K
11	240	240	292	292	292	292	0%	0%	7981	739	1K
12	240	240	290	290	290	290	0%	0%	10K	207	1K
13	240	240	284	284	284	284	0%	0%	1051	213	1K
14	240	240	291	291	291	296	1.71%	1.71%	2669	230	1K
15	240	240	296	296	296	298	0.67%	0.67%	1500	382	1K
16	240	240	285	285	285	285	0%	0%	1327	251	1K
17	240	240	288	288	288	288	0%	0%	5791	680	1K
18	240	240	283	283	283	290	2.47%	2.47%	4094	74	1K
19	240	240	292	292	292	292	0%	0%	3664	883	1K
20	240	240	288	288	288	288	0%	0%	10K	304	1K

[표 2] 실험결과: $|N| = 20$, $|E| = 30$, $|K| = 5$, $wgt = 20$

No	Lower bound			Upper Bound			Gap	Ratio	CPU Time(Seconds)		
	HRM	EHRM	Final	HRM	EHRM	Heuristic			HRM	EHRM	Heuristic
1	240	240	530	530	530	530	0%	0%	556	1818	1K
2	240	240	630	650	630	630	0%	0%	10K	6670	1K
3	240	240	730	730	730	730	0%	0%	54	40	1K
4	240	240	490	490	490	490	0%	0%	1293	231	1K
5	240	240	670	670	670	690	2.98%	2.98%	10K	613	1K
6	240	240	530	530	530	530	0%	0%	627	181	1K
7	240	240	550	550	550	550	0%	0%	487	265	1K
8	240	240	450	450	450	450	0%	0%	63	65	1K
9	240	240	630	630	630	630	0%	0%	244	872	1K
10	240	240	710	710	710	720	1.40%	1.40%	76	215	1K
11	240	240	710	710	710	710	0%	0%	6830	82	1K
12	240	240	650	650	650	650	0%	0%	723	731	1K
13	240	240	530	530	530	540	1.88%	1.88%	2035	3092	1K
14	240	240	690	690	690	690	0%	0%	657	302	1K
15	240	240	790	790	790	790	0%	0%	10K	1750	1K
16	240	240	550	550	550	560	1.81%	1.81%	112	90	1K
17	240	240	610	610	610	610	0%	0%	1125	232	1K
18	240	240	490	490	490	500	2.04%	2.04%	2332	548	1K
19	240	240	710	710	710	730	2.81%	2.81%	278	159	1K
20	240	240	630	630	630	640	1.58%	1.58%	160	1967	1K

[표 3] 실험결과: $|M| = 30, |E| = 50, |K| = 7, wgt = 1$

No	Lower bound			Upper Bound			Gap	Ratio	CPU Time(Seconds)		
	HRM	EHRM	Final	HRM	EHRM	Heuristic			HRM	EHRM	Heuristic
1	368	368	401	436	435	425	5.98%	-2.3%	10K	10K	1K
2	368	368	400	432	432	422	5.5%	-2.3%	10K	10K	1K
3	368	368	403	434	433	430	6.69%	-0.7%	10K	10K	1K
4	368	368	392	439	435	430	9.69%	-1.1%	10K	10K	1K
5	368	368	401	435	434	426	6.23%	-1.8%	10K	10K	1K
6	368	368	400	431	431	425	6.25%	-1.4%	10K	10K	1K
7	368	368	399	439	435	430	7.76%	-1.1%	10K	10K	1K
8	368	368	395	433	435	430	8.86%	-0.7%	10K	10K	1K
9	368	368	399	435	432	428	7.26%	-0.9%	10K	10K	1K
10	368	368	404	431	431	423	4.71%	-1.8%	10K	10K	1K
11	368	368	403	436	432	424	5.21%	-1.9%	10K	10K	1K
12	368	368	399	432	432	421	5.51%	-2.5%	10K	10K	1K
13	368	368	405	435	434	422	4.19%	-2.7%	10K	10K	1K
14	368	368	401	433	433	430	7.23%	-0.7%	10K	10K	1K
15	368	368	404	434	434	439	8.66%	1.15%	10K	10K	1K
16	368	368	399	435	437	432	8.27%	-0.7%	10K	10K	1K
17	368	368	403	437	436	425	5.45%	-2.5%	10K	10K	1K
18	368	368	398	431	431	422	6.03%	-2.1%	10K	10K	1K
19	368	368	402	432	432	427	6.21%	-1.2%	10K	10K	1K
20	368	368	403	432	433	418	3.72%	-3.2%	10K	10K	1K

[표 4] 실험결과: $|M| = 30, |E| = 50, |K| = 7, wgt = 20$

No	Lower bound			Upper Bound			Gap	Ratio	CPU Time(Seconds)		
	HRM	EHRM	Final	HRM	EHRM	Heuristic			HRM	EHRM	Heuristic
1	368	368	398	510	480	460	15.5%	-4.1%	10K	10K	1K
2	368	368	400	470	470	450	12.5%	-4.2%	10K	10K	1K
3	368	368	397	450	470	430	8.31%	-4.4%	10K	10K	1K
4	368	368	420	510	530	480	14.3%	-5.8%	10K	10K	1K
5	368	368	400	470	470	450	12.5%	-4.3%	10K	10K	1K
6	368	368	400	430	430	430	7.5%	0%	10K	10K	1K
7	368	368	402	510	520	460	14.4%	-9.8%	10K	10K	1K
8	368	368	400	470	470	450	12.5%	-4.2%	10K	10K	1K
9	368	368	398	450	450	450	13.1%	0%	10K	10K	1K
10	368	368	398	450	450	440	10.5%	-2.2%	10K	10K	1K
11	368	368	400	450	450	440	10.0%	-2.2%	10K	10K	1K
12	368	368	400	470	470	440	10.0%	-6.4%	10K	10K	1K
13	368	368	400	490	470	450	12.5%	-4.3%	10K	10K	1K
14	368	368	401	470	470	430	7.23%	-8.5%	10K	10K	1K
15	368	368	418	450	450	450	7.65%	0%	10K	10K	1K
16	368	368	400	510	510	460	15.0%	-9.8%	10K	10K	1K
17	368	368	400	490	510	460	15.0%	-6.1%	10K	10K	1K
18	368	368	400	430	430	430	7.5%	0%	10K	10K	1K
19	368	368	400	450	450	450	12.5%	0%	10K	10K	1K
20	368	368	400	470	470	450	12.5%	-4.3%	10K	10K	1K

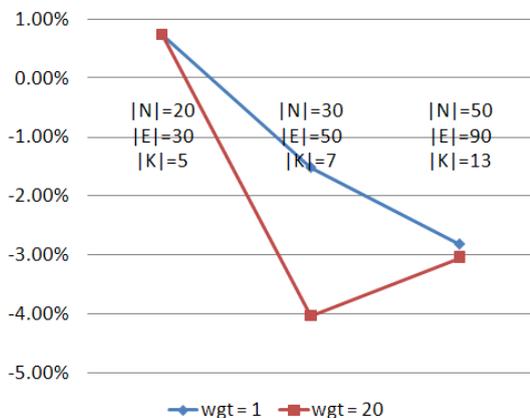
[표 5] 실험결과: $|N| = 50$, $|E| = 90$, $|K| = 13$, $wgt = 1$

No	Lower bound			Upper Bound			Gap	Ratio	CPU Time(Seconds)		
	HRM	EHRM	Final	HRM	EHRM	Heuristic			HRM	EHRM	Heuristic
1	592	592	608	732	718	673	10.7%	-6.3%	10K	10K	1K
2	592	592	606	703	703	680	12.2%	-3.3%	10K	10K	1K
3	592	592	611	712	691	678	10.9%	-1.9%	10K	10K	1K
4	592	592	603	782	694	664	10.1%	-4.3%	10K	10K	1K
5	592	592	603	708	721	678	12.4%	-4.2%	10K	10K	1K
6	592	592	605	698	702	688	13.7%	-1.4%	10K	10K	1K
7	592	592	615	792	700	684	11.2%	-2.3%	10K	10K	1K
8	592	592	604	697	698	685	13.4%	-1.7%	10K	10K	1K
9	592	592	603	692	707	670	11.1%	-3.2%	10K	10K	1K
10	592	592	605	696	692	678	12.1%	-2.0%	10K	10K	1K
11	592	592	609	698	713	678	11.3%	-2.9%	10K	10K	1K
12	592	592	603	701	693	685	13.6%	-1.1%	10K	10K	1K
13	592	592	613	694	704	690	12.5%	-0.6%	10K	10K	1K
14	592	592	607	738	730	674	11.0%	-7.6%	10K	10K	1K
15	592	592	608	693	706	690	13.5%	-0.4%	10K	10K	1K
16	592	592	605	692	719	678	12.1%	-2.0%	10K	10K	1K
17	592	592	611	693	717	685	12.1%	-1.1%	10K	10K	1K
18	592	592	608	705	701	680	11.8%	-2.9%	10K	10K	1K
19	592	592	609	699	702	679	11.5%	-2.9%	10K	10K	1K
20	592	592	606	692	699	662	9.24%	-4.3%	10K	10K	1K

[표 6] 실험결과: $|N| = 50$, $|E| = 90$, $|K| = 13$, $wgt = 20$

No	Lower bound			Upper Bound			Gap	Ratio	CPU Time(Seconds)		
	HRM	EHRM	Final	HRM	EHRM	Heuristic			HRM	EHRM	Heuristic
1	592	592	616	690	690	680	10.3%	-1.5%	10K	10K	1K
2	592	592	612	810	820	700	14.3%	-13%	10K	10K	1K
3	592	592	613	900	690	690	12.5%	0%	10K	10K	1K
4	592	592	618	880	690	680	10.0%	-1.4%	10K	10K	1K
5	592	592	614	720	810	700	14.0%	-2.7%	10K	10K	1K
6	592	592	609	720	840	700	14.9%	-2.7%	10K	10K	1K
7	592	592	617	690	690	690	11.8%	0%	10K	10K	1K
8	592	592	612	740	690	690	12.7%	0%	10K	10K	1K
9	592	592	612	740	780	670	9.47%	-9.5%	10K	10K	1K
10	592	592	614	690	700	690	12.3%	0%	10K	10K	1K
11	592	592	641	860	890	690	7.64%	-19%	10K	10K	1K
12	592	592	612	970	690	690	12.7%	0%	10K	10K	1K
13	592	592	612	730	730	690	12.7%	-5.4%	10K	10K	1K
14	592	592	606	690	710	690	13.8%	0%	10K	10K	1K
15	592	592	607	720	720	700	15.3%	-2.7%	10K	10K	1K
16	592	592	614	690	690	690	12.3%	0%	10K	10K	1K
17	592	592	613	700	700	690	12.5%	-1.4%	10K	10K	1K
18	592	592	618	790	690	690	11.6%	0%	10K	10K	1K
19	592	592	614	690	690	690	12.3%	0%	10K	10K	1K
20	592	592	613	690	750	680	10.9%	-1.5%	10K	10K	1K

표 1 ~ 표 6의 'Ratio' 값에 대한 비교 결과를 그림 4에 요약하여 나타낸다. 그림 4에서 세로축은 각 표에 나타난 20개 문제의 Ratio 평균값을 나타낸다. 가장 작은 크기의 문제($|M| = 20$, $|E| = 30$, $|K| = 5$)인 표 1과 표 2의 Ratio 평균은 $wgt = 1$ 또는 $wgt = 20$ 인 경우 모두 0.74%를 넘지 않는다. 즉, 휴리스틱 해가 최적해보다 평균적으로 0.74% 정도 더 많은 비용이 발생하는 해를 발견하고 있다. 작은 크기의 문제에 대해서 CPLEX의 수행도가 더 우수하게 나타나는 것은 CPLEX가 제한시간 10,000초 이내에 모두 최적해를 찾았기 때문이다. 한편, 문제의 크기를 조금 증가시킨 표 3과 표 4의 20문제($|M| = 30$, $|E| = 50$, $|K| = 7$)에 대해서는 $wgt = 1$ 및 $wgt = 20$ 인 경우 모두 Ratio 평균이 음수값을 보인다. 즉, 문제의 크기가 조금만 커져도 CPLEX는 최적해를 찾지 못하며 제한시간 10,000초 동안 CPLEX가 찾는 가장 좋은 해보다 휴리스틱 알고리즘이 더 좋은 해를 발견한다는 것이다. 특히, $wgt = 20$ 인 경우에는 CPLEX가 10,000초 동안 찾는 가장 좋은 해보다 휴리스틱 알고리즘이 찾는 해가 평균적으로 4%가량 비용을 절감함을 알 수 있다. 이 논문에서 실험한 가장 큰 크기의 표 5와 표 6의 문제($|M| = 50$, $|E| = 90$, $|K| = 13$)에 대해서도 휴리스틱 알고리즘은 안정적인 동작을 보여준다. 즉, $wgt = 1$ 인 경우나 $wgt = 20$ 인 경우 모두 CPLEX가 10,000초 동안 찾는 가장 좋은 해보다 평균적으로 휴리스틱 알고리즘이 3%정도 더 좋은 해를 제공한다.



[그림 4] Ratio 평균값 비교: [표 1] ~ [표 6]

6. 결론

이 논문에서는 계층형 구조를 갖는 링메쉬 광통신망

구축 문제를 다루었으며, 제안한 광통신망 구축 문제를 표현하는 정수계획법 모형을 개발하였다. 또한, 정수계획법 모형의 구조적 특성을 분석하여 링의 대칭성을 제거하는 효과적인 절단평면을 개발하였으며, LP-relaxation 값을 개선하는 효과는 없으나 작은 사이즈의 문제에서는 최적해를 구하는 데 소요되는 시간을 크게 감소시키며, 최적해를 구하는 데 소요되는 시간을 크게 감소시키며, 시간 감소 효과는 거리비용의 크기가 작을수록 커짐을 확인하였다. 또한, 이 논문에서는 사이즈가 큰 문제의 우수한 해를 구하기 위해 휴리스틱 알고리즘을 개발하였으며, 대부분의 문제에 있어서 최적해 대비 3%이내의 차이를 보이는 우수한 품질의 해를 제공한다. 최적해를 확인할 수 없는 사이즈가 큰 문제에 대해서는 휴리스틱 알고리즘이 CPLEX보다 우수한 품질의 해를 제공함을 확인하였다.

추후 연구과제는 각 링마다 하나 이상의 OXC를 설치하는 경우 HRM 모형과 비교하여 비용 절감 효과를 분석하는 것이다.

참고문헌

- [1] Aringhieri, R. and M. Dell'Amico, "Comparing metaheuristic algorithms for SONET network design problems", *Journal of Heuristics*, Vol. 11, pp. 35-57, 2005.
- [2] Chang, M. and S. Chang, "A heuristic method for self-healing ring design in a single-homing cluster", *Telecommunication Systems*, Vol. 14, pp. 175-195, 2000.
- [3] Cho, W., J. Wang, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: uniform-traffic case", *Photonic Network Communications*, Vol. 3, pp. 245-254, 2001.
- [4] Chung, S., H. Kim, Y. Yoon, and D. Tcha, "Cost-minimizing construction of a unidirectional SHR with diverse protection", *IEEE/ACM Transactions on Networking*, Vol. 4, pp. 921- 928, 1996.
- [5] Fortz, B., P. Soriano, and C. Wynants, "A tabu search algorithm for self-healing ring network design", *European Journal of Operational Research*, Vol. 151, pp. 280-295, 2003.
- [6] Gerstel, O., R. Ramaswami, and G. Sasaki, "Cost-effective traffic grooming in WDM rings", *IEEE/ACM Transactions on Networking*, Vol. 8, pp. 618-630, 2000.

- [7] Goldschmidt, O., A. Laugier, and E. Olinick, "SONET/SDH ring assignment with capacity constraints", *Discrete Applied Mathematics*, Vol. 129, pp. 99-128, 2003.
- [8] Grover, W. and R. Martens, "Combined ring-mesh optical transport networks", *Cluster Computing*, Vol. 7, pp. 245-258, 2004.
- [9] Kang, D., K. Lee, S. Park, K. Park, and S. Kim, "Design of local networks using USHRs", *Telecommunication Systems*, Vol. 14, pp. 197- 217, 2000.
- [10] Kim, Y., Y. Lee, and J. Han, "A ring-mesh topology design problem for optical transport network", *Journal of Heuristics*, Vol. 14, pp. 83- 202, 2008.
- [11] Lee C. and S. Koh, "Assignment of add-drop multiplexer rings and digital cross-connect systems mesh in telecommunication networks", *Journal of the Operational Research Society*, Vol. 52, pp. 440-448, 2001.
- [12] Lee Y., H. Sherali, J. Han, and S. Kim, "A branch-and-cut algorithm for solving an intra-ring SONET design problem", *Networks*, Vol. 35, pp. 223-232, 2000.
- [13] Sherali, H. D., J. C. Smith, and Y. H. Lee, "Enhanced model representation for an intra-ring SONET design problem allowing demand splitting", *INFORMS Journal on Computing*, Vol. 12, pp. 284-298, 2000.
- [14] Thomadsen, T. and T. Stidsen, "Hierarchical ring network design using branch-and-price", *Telecommunication Systems*, Vol. 29, pp. 61-76, 2005.
- [15] Wang, J., W. Cho, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: non-uniform traffic and bi-directional ring", *IEEE/OSA Journal of Lightwave Technology*, Vol. 19, pp. 1645-1653, 2001.

한 정 희(Junghee Han)

[정회원]



- 1994년 2월 : 고려대학교 산업공학과 (공학사)
- 1996년 2월 : 고려대학교 산업공학과 (공학석사)
- 1999년 8월 : 고려대학교 산업공학과 (공학박사)
- 2000년 3월 ~ 2004년 2월 : LG 전자 이동통신 연구소 선임연구원
- 2004년 3월 ~ 현재 : 강원대학교 경영대학 경영학과 교수

<관심분야>

정보통신, 최적화, 수요예측