

고립 픽셀 제거 변조부호 선택을 위한 절단평면 알고리즘

박태형*, 이재진^o

Cutting Plane Algorithm for the Selection of Non-Isolated Pixel Modulation Code

Taehyung Park*, Jaejin Lee^o

요 약

본 연구에서는 이차원 인접 심볼간 간섭(2D ISI) 및 인접 페이지간 간섭 (IPI)을 줄이며, 고립된 픽셀이 발생하지 않는 홀로그래픽 저장장치를 위한 이차원 변조부호를 선택하는 문제를 고려한다. 본 연구에서 고려하는 2D ISI 및 IPI를 줄이기 위한 변조부호 선택문제에서는 코드 내부에서 인접 셀간 심볼값의 차이를 최소화하고, 사용된 심볼의 분포가 균등하고, 선택된 코드워드들이 임의로 이차원에 배열될 때, 고립된 픽셀이 발생할 수 없는 코드워드를 선택한다. 코드워드 선택 문제와 선택된 코드워드에서 고립된 픽셀의 발생여부를 검색하는 모형은 정수계획법 모형으로 형성되고, 두 모형을 결합하는 절단평면(cutting plane) 알고리즘을 개발하였다. 제안된 알고리즘을 사용하여 4x2 6/8 변조부호에서 고립된 픽셀이 발생하지 않으면서, 2D ISI와 IPI를 줄일 수 있는 코드워드를 계산하였다.

Key Words : Holographic data storage, Intersymbol interference, Integer Programming, Cutting Plane Algorithm

ABSTRACT

In this paper, we introduce a modulation code design problem where best selection of two-dimensional codewords are determined to reduce two-dimensional (2D) Intersymbol Interference (ISI) and Interpage Interference (IPI), while when these codewords are randomly arranged on the storage, isolated pixel cannot be formed. Codeword selection problem and isolated pixel detection problem are formulated as integer program models and we develop a cutting plane algorithm where a valid cut is generated to remove current feasible solution to avoid isolated pixel by solving the isolated pixel detection subproblem. Using the proposed method, 4x2 6/8 codewords with non-isolated pixel are found.

1. 서 론

홀로그래픽 저장장치는 미디어의 체적에 페이지 단위로 기록과 재생을 하는 광기록 저장장치이다. 미디어의 표면에 정보를 저장하는 광기록 저장장치나 자기기록 저장장치에 비해 저장용량을 현저히 증가시킬 수 있다. 또한 페이지 단위로 정보를 처리하기 때문에

데이터 처리 속도 및 전송율이 매우 높아서 차세대 저장장치로 주목받고 있다.

하지만 홀로그래픽 저장장치는 기존 저장장치와 다른 오류요인이 있다. 홀로그래픽 저장장치는 CCD(charge-coupled device) 배열을 이용하여 2차원 픽셀 이미지를 읽기 때문에, 2차원 인접심볼간 간섭(Intersymbol Interference, ISI)이 생긴다. 또한 다수

* 본 연구는 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2012-17220364).

주저자 : 송실대학교 산업정보시스템공학과, tpark@ssu.ac.kr, 정회원

교신저자 : 송실대학교 정보통신전자공학부 정보저장및통신 연구실, zlee@ssu.ac.kr, 종신회원

논문번호 : KICS2013-04-193, 접수일자 : 2013년 4월 25일, 최종논문접수일자 : 2013년 5월 19일

의 페이지를 같은 공간에 저장하기 때문에 인접 페이지 간 간섭(Interpage Interference, IPI)이 생긴다. 2D ISI와 IPI를 줄이기 위한 변조부호, 오류정정 코드, 등화기(equalization) 및 신호검출 기법에 대한 연구가 진행되었다^{1,2)}.

인접 페이지간 간섭을 줄이기 위해서는 각 페이지마다 신호빔으로 부터 발생하는 빛의 양이 비슷해야 한다. 즉, 각 페이지마다 ON 픽셀과 OFF 픽셀의 분포가 비슷하면 신호빔의 세기의 변화가 적기 때문에 기록조건이 유사하게 되어 페이지간 간섭을 줄일 수 있다³⁾. 또한 2차원 ISI를 줄이기 위해서는 이웃한 픽셀간의 심볼값 차이가 작아야 하고, 페이지에 배열되었을 때, “0” 픽셀 주변을 “1” 픽셀이 에워싼 모양의 고립(isolated) 픽셀의 발생이 없어야 한다^{4,6)}. 본 연구에서는 정수계획법(integer programming)을 이용하여 고립픽셀이 발생하지 않으며 2D ISI와 IPI를 줄일 수 있는 변조부호의 디자인 방법을 소개한다.

II. 고립픽셀 방지를 위한 정수계획법 모형

위의 조건들을 만족하는 변조부호를 디자인하기 위한 정수계획법 모형을 이진 6/8 변조부호를 이용하여 기술한다. 아래에서 $[n]$ 은 집합 $\{1, \dots, n\}$ 을 나타낸다.

6/8 변조부호는 6비트를 입력으로 받아 8비트의 4×2 행렬의 코드워드를 내보낸다. 4×2 행렬에 2개의 심볼을 사용하는 경우, 표현할 수 있는 코드워드의 수는 총 256개이고, 이 중 64개의 코드워드를 사용한다.

$N=[256]$ 은 가능한 총 코드워드들의 집합, $M=[8]$ 은 코드워드 안의 셀의 집합이다. 코드워드 j 의 셀 i 에 기록된 심볼을 a_{ij} , 코드 j 에서 심볼 $s \in \{0,1\}$ 의 발생회수를 b_{sj} 로 표시한다. 코드워드 j 내부의 셀간 심볼값 차이의 합을 c_j , 코드워드 j 와 k 가 수평 혹은 수직으로 이웃할 때, 두 코드워드가 만나는 점점의 셀간 심볼값 차이의 합을 c_{jk} , $j < k \in N$ 로 표시한다. 또한 두 코드워드 j 와 k 의 해밍(Hamming)거리를 $h(j,k)$ 로 할 때, $h(j,k) < d_{\min}$ 이 되는 코드워드쌍들의 집합을 H_1 으로 표시한다. $c_j, c_{jk}, h(j,k)$ 는 a_{ij} 의 함수로 표현할 수 있다.

2D ISI와 IPI를 줄이기 위해서는 인접한 셀간 심볼값의 차이가 작고, 심볼들의 사용분포가 비슷한 코드워드들의 선택이 필요하다. 이러한 특성을 갖는 64개의 코드워드를 전체 가능한 256개의 코드워드에서 선택하는 문제를 노드의 집합 N , 링크의 집합 $E = \{(j,k) : j < k \in N\}$ 인 그래프에서 노드, 링크의

가중치합이 최소가 되고 사용한 심볼의 분포가 비슷한, 64개의 노드로 구성된 부분그래프(subgraph) S 를 찾는 문제로 모형화할 수 있다. 이러한 부분그래프를 찾는 문제는 [7]에서 정수계획법 모형으로 수식화하였다. 식 (1)은 [7]의 이차정수계획법 모형을 추가변수를 도입하여 선형정수계획법으로 변환한 모형이다.

$$\begin{aligned} \text{Min} \quad & \alpha_1 \left(\sum_{j \in N} c_j x_j + \sum_{j < k} c_{jk} z_{jk} + \sum_s u_s \right) \quad (1a) \\ & + \alpha_2 \left(\sum_{(j,k) \in H_1} w_{jk} \right) \\ \text{s.t.} \quad & \sum_{j \in N} x_j = 64 \quad (1b) \\ & \sum_{j \in N} b_{sj} x_j - u_s \geq 256, \quad s \in \{0,1\} \quad (1c) \\ & z_{jk} \leq x_j, z_{jk} \leq x_k, \quad j < k \in N \quad (1d) \\ & x_j + x_k - z_{jk} \leq 1, \quad j < k \in N \quad (1e) \\ & x_j + x_k - w_{jk} \leq 1 \quad (j,k) \in H_1 \quad (1f) \\ & x_j \in \{0,1\}, \quad j \in N \quad (1g) \\ & u_s \geq 0, \quad s \in \{0,1\} \quad (1h) \\ & z_{jk} \geq 0, \quad w_{jk} \geq 0, \quad j < k \in N \quad (1i) \end{aligned}$$

식 (1a)는 정수계획법 문제의 목적함수를 나타내고, 식 (1b)-(1i)는 제약식을 나타낸다. 식 (1b)는 64개의 코드워드가 선택됨을 나타내고, (1c)는 각 심볼이 균등하게 발생함을 나타낸다. (1c)의 $64 \times 8/2 = 256$ 으로서, 두 개의 심볼이 50% 균등하게 발생하는 회수를 표시한다. 변수 u_s 는 두 심볼이 정확하게 50% 균등하게 발생하지 않을 때의 페널티를 부여하기 위한 보조변수이다. (1d),(1e)는 두 코드워드 x_j, x_k 가 선택되었을 때, $z_{jk} \geq 1$ 을 보장하기 위한 제약식으로 목적함수의 $\sum_{j < k} c_{jk} z_{jk}$ 계산을 위해 도입되었다. (1f)는 선택된 코드워드간의 최소 해밍거리 만족제약식이고, (1c)와 마찬가지로 변수 w_{jk} 는 페널티 계산 보조변수이다. $w_{jk} = 1, (j,k) \in H_1$ 인 경우, 두 코드워드 j 와 k 가 둘 다 최적해에 포함되었음을 의미한다. 목적함수에서 α_1, α_2 는 사용자 지정 가중치이다. (1g)-(1i)는 x_j, u_s, z_{jk}, w_{jk} 에 대한 이진값 제약식 및 비음부호 제약식이다. (1a)의 c_j, c_{jk} 의 정의는 [7]에 수록되었다.

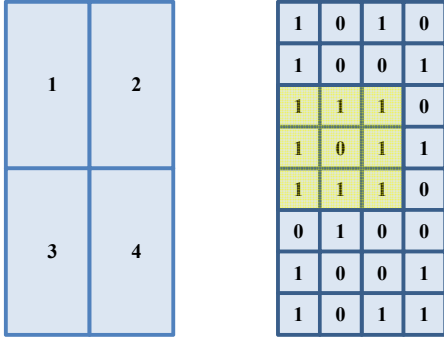


그림 1. 고립된 픽셀 계산을 위한 코드워드의 배열 및 고립된 픽셀의 예.
Fig. 1. Arrangement of arbitrary 4 codewords and example of isolated 3x3 pixel pattern.

다음의 정수계획법 모형은 모형 (1)에서 선택한 코드워드들이 임의로 배열된 경우 고립된 픽셀이 발생할 수 있는지 여부를 결정하는 모형이다. 그림 1의 왼쪽과 같이 4개의 코드워드가 2x2 형태로 배열된 경우를 고려하자. 고립된 픽셀의 발생여부는 2x2 형태로 임의의 코드워드를 배열하였을 때, 그림 1 오른쪽처럼, 전체 8x4개의 행렬에서 3x3 고립된 패턴이 부분행렬(submatrix)로 포함되었는지를 검색하여 판단한다. 0-1 변수 y_{kj} , $k \in K = [4], j \in N$ 은 그림 1의 왼쪽 2x2 행렬의 k 에 코드워드 j 가 위치함을 나타낸다. z_{mn} , $m \in [8], n \in [4]$ 은 8x4 행렬의 (m, n) 항의 심볼을 나타낸다. 3x3 고립된 픽셀 패턴은 3x3 행렬 $P_{ij}, i \in [3], j \in [3]$ 을 이용하여 표시한다. 고립된 픽셀의 검색은 3x3 행렬 P 와 8x4행렬의 모든 3x3 부분행렬과의 L_1 -거리의 최소, 즉

$$\min_{s \in [6], t \in [2]} \left\{ \sum_{i,j=1}^3 |P_{ij} - z_{r(i,s),c(j,t)}| \right\} \quad (2)$$

가 0이 되는 부분행렬이 있는지 여부를 판단하는 문제이다. 여기에서 $r(i,s), c(j,t)$ 는 변수 z 의 행렬을 표시한다.

8x4 행렬에서 행렬 P 와의 L_1 -거리가 최소가 되는 3x3 부분행렬을 찾는 문제는 다음과 같은 혼합정수계획법 모형으로 수식화 할 수 있다.

$$\begin{aligned} \text{Min } & \gamma & (3a) \\ \text{s.t. } & \sum_{ij} t_{ij}^{st} - \gamma \leq M(1 - \beta_{st}), \quad s \in S, t \in T & (3b) \\ & -\sum_{ij} t_{ij}^{st} + \gamma \leq M(1 - \beta_{st}), \quad s \in S, t \in T & (3c) \end{aligned}$$

$$t_{ij}^{st} \geq P_{ij} - z_{s-1+i, t-1+j} \quad (3d)$$

$$t_{ij}^{st} \geq -P_{ij} + z_{s-1+i, t-1+j} \quad (3e)$$

$$\sum_{s,t} \beta_{st} \geq 1 \quad (3f)$$

$$\sum_{j \in N} y_{kj} = 1, k \in K \quad (3g)$$

$$y_{kj} \leq x_j, k \in K, j \in N \quad (3h)$$

$$z_{r(i,k), c(i,k)} = \sum_{j \in N} a_{ij} y_{kj}, k \in K, i \in M \quad (3i)$$

$$y_{kj} \in \{0,1\}, k \in [4], j \in N \quad (3j)$$

$$\beta_{st} \in \{0,1\}, s \in S, t \in T \quad (3k)$$

모형 (3)에서 (3d), (3e)에 의해, 고립된 픽셀패턴 P 와 (s,t) 위치에 있는 부분행렬과의 L_1 거리는 $\sum_{ij} t_{ij}^{st} = \sum_{ij} |P_{ij} - z_{s-1+i, t-1+j}|$ 가 되며, 0-1 변수 β_{st} 는 8x4 행렬에서 비교된 패턴과 일치하는 부분행렬의 위치를 표시한다. 식 (3b), (3c), (3f)에 의해, γ 는 $\sum_{ij} t_{ij}^{st}$ 의 최소값과 일치하게 된다. 선택된 코드워드들의 조합에서 배제하고 싶은 다수의 고립픽셀 패턴이 있는 경우에도 식 (3a)-(3f)를 확장하여 적용할 수 있다.

모형 (3)을 모형 (1)과 결합하려면, 모형 (3)이 제약식 형태로 표현되어야 한다. 하지만, 64개의 코드워드에서 임의로 배열된 4개의 코드워드에서의 고립된 픽셀을 검사하는 모형 (3)을 제약식 형태로 표현하려면, 각각의 ${}_{256}C_4$ 조합에 개별적인 변수를 도입하고 식 (3b)-(3k)를 각각의 조합별로 포함하여야 하므로, 불가능하다. 본 연구에서는 모형 (1)과 (3)이 결합된 모형을 위한 해법으로 절단평면(cutting plane) 알고리즘을 개발하였다. 문제 (1)에서 선택한 코드워드에 고립된 픽셀이 발생하는 경우는 모형 (3)의 최소값이 0이 되는 경우이다. 따라서, 문제 (1)에서 고립된 픽셀이 발생하지 않는 64개의 코드워드는 다음 조건을 만족해야 한다.

$$1 \leq \min \{ \gamma \mid (3b) - (3k) \} \quad (4)$$

절단평면 알고리즘에서는 모형 (1)을 푸는 단계 n 에서 정수해 $x^{(n)} = (x_1, \dots, x_N)$ 가 주어진 경우, 식 (4)을 위반하는 경우, 현재의 $x^{(n)}$ 를 잘라내는(cutting) 부등식을 현재 단계 모형에 추가하는 알고리즘이다. 단계 n 의 모형 (1)의 최적정수해 $x^{(n)}$ 은 (3h)에 의해

표 1. 6/8 코드의 계산결과.
Table 1. Computation result for 6/8 code.

Problem	α_1	α_2	d_{\min}	obj. value	$ W / H < d_{\min} $	$ E_s / E $	# user cut
8B	1	1	2	2,563.5	147/1,024	2,016/32,640	
8B	0	1	2	0	0/1,024	2,016/32,640	
8B	1	0	3	2,409	531/4,608	2,016/32,640	
8B	0	1	3	192	192/4,608	2,016/32,640	
8B	0	1	2	110	110/1,024	2,016/32,640	156
8B	0	1	2	124	124/1,024	2,016/32,640	1008

변수 y_{kj} 의 상한으로 (3)에 입력된다. 이 경우 모형 (3)의 최적해를 $\beta_{st}^{(n)}, y_{kj}^{(n)}, z_{mn}^{(n)}$ 로 부른다. (3g)에 의해, 각 k 에 대하여, $y_{kj}^{(n)} = 1$ 이 되는 j 는 1개 발생한다. $j(k) = \{j | y_{jk} = 1\}, k \in K$ 로 정의하고, 집합 $J = \{j(k), k \in K\}$ 으로 정한다. 집합 J 는 고립된 픽셀 패턴과의 L_1 -거리가 최소가 되는 코드워드들의 조합을 나타내므로, 동일한 $j(k)$ 가 반복해서 수록될 수도 있다. 예를 들어, $J = \{1, 45, 45, 247\}$ 인 경우, $y_{1,1} = y_{45,2} = y_{45,3} = y_{247,4} = 1$ 을 나타낸다. (3)에서 계산한 하위문제의 목적함수값이 0인 경우, 즉, 고립된 픽셀패턴이 발생한 경우, 다음 단계에서 하위문제의 목적함수값이 0보다 크게 하려면, 다음과 같은 부등식이 성립한다.

$$\sum_{j(k), k \in K} x_{j(k)} \leq |J| - 1 \quad (5)$$

여기에서 $|J|$ 는 집합 J 에서 반복되지 않는 인덱스들의 개수를 나타낸다. 따라서 $J = \{1, 45, 45, 247\}$ 인 경우, $|J| = 3$ 으로 정의하고 식 (5)는 $x_1 + x_{45} + x_{247} \leq 2$ 이 된다. (10)이 현재 단계의 모형 (1)에 추가됨으로써, 다음 단계에서는 현재단계와 동일한 J 가 발생하지 않는다. 이러한 cut을 모형 (1)에 추가하여 다음 단계로 진행한다. 절단평면 알고리즘의 단계 (n)은 다음과 같다.

Step 1. 모형 (1)을 풀어서 $x^{(n)} = (x_j^{(n)})$ 을 계산한다. (1)에 가능해가 없는 경우, 중단한다. 이 경우, 고립픽셀이 배제된 코드워드의 조합은 불가능하다. (1)의 최적해가 존재하는 경우, Step 2로 간다.

Step 2. $x^{(n)}$ 이 (3h)의 상한으로 주어진 모형 (3)을

풀어서 최소값 γ 를 계산한다.

Step 3. $\gamma \geq 1$ 인 경우, 정지하고 현재의 해 $x^{(n)}$ 은 고립픽셀이 발생하지 않는 코드워드들의 집합이다. $\gamma = 0$ 인 경우, cut (5)를 (1)에 추가한다. Step 1로 돌아간다.

III. 계산 결과

모형 (1)을 위한 절단평면 알고리즘은 Visual Studio 2008과 IBM ILOG의 CPLEX callable library와 AMPL 소프트웨어를 이용하여 구현되었다^{8,9)}. 모형 (1)의 경우, $d_{\min} = 2$ 인 경우, 사용된 변수의 수는 1,282개, (1e), (1f)를 제외한 제약식의 수는 1,027개이다. 절단평면을 생성하는 모형 (3)의 경우 사용된 변수는 1,069개, 제약식은 1,301개이다.

표 1에는 6/8 코드에 대하여 $\alpha_1, \alpha_2, d_{\min}$ 의 값에 따른 목적함수와 해밍거리를 위반한 정도를 기록하였다. 표 1에서 "obj. value" 는 모형 (1)의 목적값을 표시하고, $|W|$ 는 최적해에서 $\sum_{(j,k) \in H_1} w_{jk}$ 를 표시하고, $|H < d_{\min}|$ 은 $h(j,k) < d_{\min}$ 이 되는 (j,k) 의 개수를 표시한다. $|E_s|$ 는 선택된 코드워드쌍의 개수, $|E|$ 는 전체 가능한 (j,k) 조합의 개수를 나타낸다. "# user cut"은 모형 (3)을 부분제로 사용하는 절단평면 알고리즘에서 사용한 cut의 개수를 나타낸다.

(1)을 푸는 과정에서 (5)의 절단평면은 총 156번 생성되었고, (1)의 목적값은 110, CPU 시간은 11.26초, $\sum_j b_{sj} x_j = 64$ 로 심볼 "0"과 "1"과 50% 씩 균등하게 사용되었다. 그림 2에는 고립된 픽셀패턴이 발생하지 않을 때, 식 (3)의 최적해에서의 변수 $z_{r(i,k), c(i,k)}$ 를 8x4 행렬로 값을 배열한 그림이다. 이 그림에서 고립

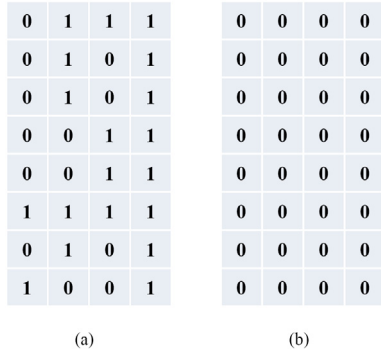


그림 2. 고립된 픽셀이 발생하지 않는 코드워드의 부분제 (5)의 (z) 값.
 Fig 2. (z) values in the subproblem (5) for codewords with no isolated pixel pattern.

된 픽셀이 발생하지 않았음을 알 수 있다. 그림 2(a)는 $y_{85,1} = y_{216,2} = y_{55,3} = y_{246,4} = 1$ 이 되었을 때 발생하였다. 고립된 픽셀과의 L_1 -거리는 1이 되었다. 그림 2(b)에는 중심이 “1”, 주변이 “0”으로 된 고립된 픽셀이 발생하지 않는 6/8 코드의 변수 z_{ij} 행렬을 보여준다.

IV. 결 론

본 연구에서는 4×2 행렬에 0-1 심볼을 저장하는 홀로그래픽 저장장치에서 고립된 픽셀패턴이 발생하지 않는 코드워드들을 선택하는 정수계획법 모형을 제시하고, 절단평면 알고리즘을 제안하였다. 제안된 알고리즘은 고립된 픽셀을 검색하는 부분제를 제외한 정수계획법 모형으로 셀간 심볼값의 변화가 적고, 사용된 심볼의 분포가 균등한 코드워드를 각 단계별로 계산하고, 단계별로 현재의 정수해에 대하여 고립된 심볼이 발생가능한지 검토하고, 고립된 심볼이 발생하는 경우, 현재 해를 절단(cutting)하는 부등식을 추가하는 절차로 구성된다. 본 연구에서 제안한 알고리즘을 사용하여 주위가 1, 중심이 0인 고립된 픽셀패턴이 발생하지 않는 6/8 코드를 계산하였다. 본 연구에서 제안한 모형은 심볼값이 다수의 값, 예를 들어 {0,1,2,3}과 같이 multi-level 코드에서 고립된 픽셀패턴이 발생하지 않는 코드워드 선택 문제에도 바로 적용 가능하다. 또한 배제할 패턴은 본 연구에서 예시한 3×3 패턴 이외에 다른 형태의 패턴도 적용할 수 있다. 장래 연구에서는 본 연구에서 개발한 방법론을 멀티레벨 코드에서 다양한 고립 패턴제거에 대하여 적용할 예정이다.

References

- [1] L. Hesselink, S. S. Orlov, and M. C. Bashaw, “Holographic data storage systems,” in *Proc. IEEE*, vol. 92, no. 8, pp. 1231-1280, July 2004.
- [2] V. Vadde and B. V. K. V. Kumar, “Channel modeling and estimation for intrapage equalization in pixel-matched volume holographic data storage,” *Appl. Opt.*, vol. 38, no. 20, pp. 4374-4386, July 1999.
- [3] J. Kim, J.-K. Wee and J. Lee, “Error correcting 4/6 modulation codes for holographic data storage,” *Japanese J. Appl. Phys.*, vol. 49, no. 8, pp. 08KB04, Aug. 2010.
- [4] J. Kim and J. Lee, “Error-correcting 6/8 modulation code for reducing two-dimensional intersymbol interference,” *Japanese J. Appl. Phys.*, vol. 50, no. 9, pp. 09MB06, Sep. 2011.
- [5] J. Kim and J. Lee, “Two-dimensional non-isolated pixel modulation code for holographic data storage,” *J. KICS*, vol. 34, no. 2, pp. 163-168, Feb. 2009.
- [6] J. Kim and J. Lee, “Two-dimensional 5:8 modulation code for holographic data storage,” *Japanese J. Appl. Phys.*, vol. 48, no. 3, pp. 03A031, Mar. 2009.
- [7] T. Park and J. Lee, “Integer programming models for the design of two-dimensional holographic storage modulation code,” *J. KICS*, vol. 38, no. 6, pp. 459-464, June 2013.
- [8] IBM, *IBM ILOG CPLEX V12.1: User’s Manual for CPLEX*, Retrieved Dec., 21,, 2009, from ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmanplex.pdf.
- [9] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Scientific Press, 1993.

박 태 형 (Taehyung Park)



1986년 2월 고려대학교 산업
공학과 학사

1989년 2월 고려대학교 산업
공학과 석사

1998년 6월 Virginia Tech 산
업시스템공학과 박사

2001년~현재 송실대학교

산업정보시스템공학과 부교수

<관심분야> 정수계획법, 통신네트워크

이 재 진 (Jaejin Lee)



1983년 2월 연세대학교 전자
공학과 학사

1984년 12월 U. of Michigan,
Dept. of EECS 석사

1994년 12월 Georgia Tech.
Sch. of ECE 박사

1995년 1월~1995년 12월

Georgia Tech. 연구원

1996년 1월~1997년 2월 현대전자 정보통신 연구
소 책임 연구원

1997년 3월~2005년 8월 동국대학교 전자공학과
부교수

2005년 9월~송실대학교 정보통신전자공학부 교수

<관심분야> 통신이론, 채널코딩, 기록저장 시스템