

# 홀로그래픽 저장장치에 대한 GS 인코딩의 정수계획법 모형 및 휴리스틱

박태형\*, 이재진<sup>o</sup>

## Integer Programming Model and Heuristic on the Guided Scrambling Encoding for Holographic Data Storage

Taehyung Park\*, Jaejin Lee<sup>o</sup>

### 요약

홀로그래픽 저장장치의 Guided Scrambling (GS) 인코딩에서는 소스비트에 제어비트를 추가하여 스램블링된 코드워드에서 변조 제약식을 만족하는 최적의 코드를 선택한다. 본 연구에서는 GS 인코딩 과정을 선형식으로 수식화하여, 제어비트를 직접 계산하는 0-1 정수계획법 모형을 소개한다. 또한 정수계획법의 계산 복잡도를 줄이기 위해 제어비트를 결정하는 이웃탐색 휴리스틱 알고리즘을 개발한다. 개발된 모형은 다양한 코드 행렬 및 제어비트를 갖는 문제들에 적용하여 성능을 비교하였다.

**Key Words** : Holographic data storage, Guided Scrambling Encoding, Integer Programming, Neighborhood Search Heuristic

### ABSTRACT

In Guided Scrambling (GS) encoding for the holographic storage, after scrambling augmented source word into codeword, the best codeword satisfying modulation constraint is determined. Modulation constraints considered in this paper are *strength* which is the minimum number of transition between '0' and '1' in each row and column of codeword array and the symbol balancedness of codeword array. In this paper, we show that GS encoding procedure can be formulated as an integer programming model and develop a fast neighborhood search heuristic for fast computation of control bits. In the simulation, we compared the performance of heuristic algorithm with the integer programming model for various array and control bit size combinations.

### I. 서론

홀로그래픽 데이터 저장장치(holographic data storage, HDS)는 데이터 페이지의 광학이미지와 참조빔(reference beam)과의 간섭패턴을 2차원 픽셀 이미지로 저장하고, charge-coupled device(CCD)로 픽셀 데이터를 읽어서 정보를 추출한다<sup>[1,2]</sup>. HDS의

신뢰도를 증가시키기 위해서는 '0'과 '1'의 패턴은 변조 제약식을 만족해야 한다. 그러한 제약식 중에서, 저장된 페이지 행렬의 각 행과 열에서 '0'에서 '1', '1'에서 '0'으로의 천이(transition)가 최소  $t$ 번은 있어야 한다는 조건이 있다. 이러한 조건을 만족하는 이진행렬을 천이강도(strength)  $t$ 를 갖는 보수적 행렬(conservative array)로 부른다. 또 다른 조건은

\* 주저자 겸 교신저자 : 숭실대학교 산업정보시스템공학과, tpark@ssu.ac.kr, 정회원  
<sup>o</sup> 숭실대학교 정보통신전자공학부 정보저장및통신 연구실, zlee@ssu.ac.kr, 종신회원  
 논문번호 : KICS2013-07-295, 접수일자 : 2013년 7월 18일, 최종논문접수일자 : 2013년 6월 23일

각 페이지의 빛의 강도의 합은 비슷해야 한다는 것이다. 즉, '0' 및 '1' 픽셀의 발생 횟수는 비슷해야 한다. 이러한 행렬을 balanced 행렬로 부른다<sup>3,41</sup>.

페이지 크기가 커짐에 따라, look-up 테이블이나, enumerative 코딩방식 이외의 인코딩 방법에 대한 관심이 증대하고 있다. 이러한 조건을 만족하는 인코딩 기법으로 멀티모드(multi-mode) 코딩 기법이 있다. 멀티모드 코딩에서는 각 소스워드  $x$ 가  $L$ 개의 코드워드 로 구성된 집합  $C_x$ 에 포함된 코드워드 중 한 개로 표현된다. multi-mode 코딩의 기본요소는 소스워드  $x$ 와  $C_x$ 간의 1-to- $L$  함수를 정의하는 것이다. 이러한 함수 중, Fair 등이 소개한 Guided Scrambling (GS), dc-free coset 코드, Reed-Solomon 스크램블 함수가 있다<sup>51</sup>.

GS 코딩에서는  $n$ 비트 전송데이터에  $p$ 비트 제어 비트(control bit)를 앞에 붙인 후,  $n + p$ 비트를 선형 케환 시프트 레지스터(linear feedback shift register, LFSR)를 통과시켜서 pseudo random 코드워드를 생성한다.  $p$  제어비트를 사용하여 생성 가능한 후보코드워드는 총  $L = 2^p$ 개가 있다. 이렇게 생성한 코드워드를 행렬로 배열한 후, 변조 제약식을 만족하는 최적의 코드워드를 선택하여 전송(기록)하는 기법이다. 그림 1은 GS 인코딩 절차를 간략하게 예시한다. 소스비트 행렬의 앞부분에 제어비트를 붙인 후, 스크램블된 코드워드에서 최적코드를 선택한 후 전송하며, 수신단에서는 동일한 다항식을 이용하여 디스크램블(descramble) 하는 단계를 보여주고 있다<sup>61</sup>. 홀로그래픽 저장장치에 GS 인코딩을 적용할 때, 코드워드의 최대 빛의 저장강도에 대한 성능평가 및 심볼 값이 균등한 코드워드에서 제어비트와 천이강도의 최소값간의 관계에 대한 모의실험이 보고되었다<sup>7,81</sup>. GS 코딩은 홀로그래픽 저장장치 이외에 OFDM의 peak-to-average power ratio(PAPR) 감소를 위한 코딩 및 광기록 저장장치를 위한 변조코드의 직류성분 제거를 위한 코딩 연구에서도 사용되었다<sup>9,101</sup>.

본 연구에서는 GS 인코딩에서 변조 제약식을 만족하는 최적의 코드워드를 찾는 과정을 정수계획법 모형으로 수식화한다. 정수계획법 모형이 변조 제약식을 만족하는 최대 천이강도를 계산할 수 있지만, 연산복잡도가 높으므로, 간단한 이웃탐색 휴리스틱(neighborhood search heuristic, NSH) 알고리즘을 개발하여 두 가지 알고리즘의 성능을 평가한다.

GS 인코딩에서 최적 코드워드를 결정하는 문제를 최적화 관점에서 모형화함으로, GS 인코딩이 적용된

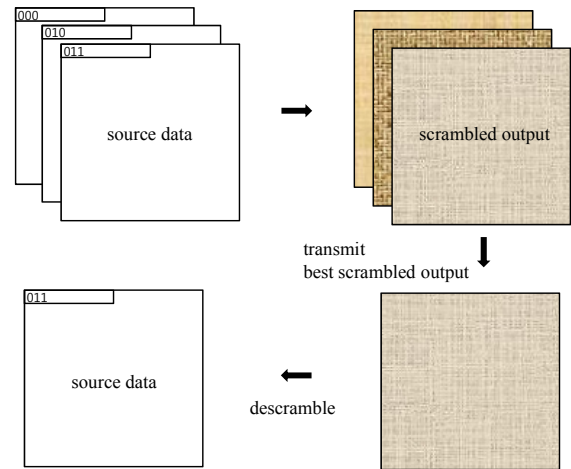


그림 1. GS 인코딩/디코딩 절차.  
Fig. 1. GS encoding/decoding procedure.

다른 문제들도 비슷하게 수식화할 수 있다. 또한 개발된 모형을 이용하면 GS 인코딩의 최악(worst case) 성능을 평가할 수 있는 모형을 개발할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 GS 인코딩과 동등한 정수계획법 모형을 개발하고, 3장에서는 간단한 이웃탐색 알고리즘을 제안한다. 4장에서는 다양한 코드워드 행렬 크기 및 제어비트수에 따른 개발된 정수계획법 모형과 휴리스틱의 성능을 비교한다.

## II. GS 인코딩의 정수계획법 수식화

GS 인코딩에서는  $n$ 비트 소스워드에  $p$ 비트의 제어비트를 붙여서 LFSR을 통과시킨 후, 스크램블된 코드워드를 행렬에 저장한다. 예를 들어,  $n = 92, p = 8, m = 10$ 인 경우, 소스워드 92비트에 8비트의 제어비트  $b_1, \dots, b_8$ 를 앞에 붙인  $b = (b_1, \dots, b_{100})$ 를 LFSR에 통과시켜서 스크램블 코드워드  $c = (c_1, \dots, c_{100})$ 를 계산한다. 벡터  $c$ 를  $10 \times 10$ 행렬로 저장할 때 행렬은  $c_{11} = c_1, \dots, c_{1,10} = c_{10}, \dots, c_{10,10} = c_{100}$ 로 저장한다.

본 논문에서는 11차 스크램블러 다항식을 사용하였고, 초기값은  $c_0 = 1, \dots, c_{-10} = 1$ 로 정하였다. 코드워드 벡터  $c$ 는 다음 식을 만족한다.

$$c_k = b_k \oplus c_{k-2} \oplus c_{k-11}, k = 1, \dots, n + p \quad (1)$$

$$c_k = b_k \oplus c_{k-2} \oplus c_{k-11} = s_k \oplus c_{k-11} \text{ 이고, 여기}$$

에서  $s_k = b_k \oplus c_{k-2}$ 로 정의한다. modulo-2 덧셈  $s_k = b_k \oplus c_{k-2}$ 은 다음과 같은 4개의 부등식으로 표현할 수 있다.

$$\begin{aligned} s_k &\leq b_k + c_{k-2}, & s_k &\leq 2 - b_k - c_{k-2} \\ s_k &\geq b_k - c_{k-2}, & s_k &\geq c_{k-2} - b_k \end{aligned} \quad (2)$$

또한  $c = (c_1, \dots, c_{n+p})$ 를  $m \times m$  행렬  $(c_{i,j})$ 로 저장할 때,  $i$ 번째 행과  $j$ 번째 열에서의 인접 심볼간 천이회수  $t_i, t_j$ 는  $c_{i,j}$ 가 0, 1이므로 다음과 같이 modulo-2 연산을 통해 계산할 수 있다.  $u_{i,j} = c_{i,j} \oplus c_{i,j+1}$ ,  $w_{i,j} = c_{i,j} \oplus c_{i+1,j}$ 로 정의하면,  $i$ 번째 행의 천이회수  $t_i$ 와  $j$ 번째 열의 천이회수  $t_j$ 는 다음 식을 만족한다.

$$t_i = \sum_{j=1}^{m-1} u_{i,j}, \quad t_j = \sum_{i=1}^{m-1} w_{i,j} \quad (3)$$

행렬  $(c_{i,j})$ 에서 '1'의 발생회수는  $\sum_{i,j} c_{i,j}$ 이므로, 두 심볼이 균등하게 사용되는 경우, 다음 식을 만족한다.

$$\sum_{i=1}^m \sum_{j=1}^m c_{ij} = m^2/2 \quad (4)$$

제어비트 값은 0-1 변수  $d_1, \dots, d_p$ 으로 나타내고, 변수  $d_{p+1}, \dots, d_{n+p}$ 는 사용자 데이터  $b_{p+1}, \dots, b_{n+p}$ 의 값을 갖는다. 식 (1)-(4)를 이용하여 천이강도  $t$ 를 최대화하는 정수계획법 모형 (GSIP)는 다음과 같다. 아래에서  $N = \{1, \dots, n+p\}$ ,  $M = \{1, \dots, m\}$ ,  $M_0 = M - \{m\}$ 를 표시한다.

아래에서 modulo-2 연산을 표시하는 부등식 (2)를  $s_k = b_k \diamond c_{k-2}$ 로 표시하였다. 모형 (5)를 정수계획법 소프트웨어에 입력할 때는 각  $\diamond$  식을 (2)의 4개의 부등식으로 대체하여야 한다.

$$\begin{aligned} \text{Max } &t & (5) \\ \text{s.t. } &s_i = d_i \diamond c_{i-2}, \quad i \in N \\ &c_i = s_i \diamond c_{i-1}, \quad i \in N \\ &u_{i,j} = c_{i,j} \diamond c_{i,j+1}, \quad \forall i \in M, j \in M_0 \\ &w_{i,j} = c_{i,j} \diamond c_{i+1,j}, \quad \forall i \in M_0, j \in M \\ &t \leq \sum_{j \in M_0} u_{i,j}, \quad i \in M \\ &t \leq \sum_{i \in M_0} w_{i,j}, \quad j \in M \\ &\sum_{i,j} c_{i,j} = m^2/2 \\ &d_{p+1} = b_{p+1}, \dots, d_{n+p} = b_{n+p} \\ &d_i \in \{0,1\}, \quad i = 1, \dots, p \end{aligned}$$

### III. 이웃탐색 휴리스틱

2장에서 소개한 GSIP의 경우, 0-1 정수계획법 문제로 NP-hard 문제이다. 이 장에서는 계산복잡도가 단순한 이웃탐색 휴리스틱을 소개한다. 제어비트  $d = (d_1, \dots, d_p)$ 에서  $j$ 번째 비트가 0에서 1, 혹은 1에서 0으로 바뀐 비트벡터를  $d$ 의 1차원 이웃벡터  $\tilde{d}_j$ 로 정의한다.

$$\tilde{d}_j = (d_1, \dots, 1 - d_j, \dots, d_p) \quad (6)$$

마찬가지로  $d$ 의 2차원 이웃벡터  $\tilde{d}_{ji}$ 는 다음과 같이 정의한다.

$$\tilde{d}_{ji} = (\tilde{d}_1, \dots, 1 - \tilde{d}_i, \dots, \tilde{d}_j, \dots, \tilde{d}_p), \quad i \neq j \quad (7)$$

벡터  $d$ 의 성능은  $d$ 가 제어비트일 때, 스크램블된 코드 워드 행렬  $(c_{i,j})$ 의 천이강도 값  $s(d)$ 와 balance 된 정도  $b(d)$ 를 합하여 정의한다.  $|\sum_{ij} c_{ij} - m^2/2| \leq 2$ 인 경우,  $b(d) = 0$ , 그렇지 않으면,  $b(d) = -2|\sum_{ij} c_{ij} - m^2/2|$ 로 정의한다. 이전 단계 제어비트  $d^{pre}$ 에 대한 제어비트  $d$ 의 성능 차이는

$$\begin{aligned} \Delta(d, d^{pre}) &= c_1(s(d) - s(d^{pre})) \\ &\quad + c_2(b(d) - b(d^{pre})) \end{aligned} \quad (8)$$

로 정의한다. 여기에서  $c_1, c_2$ 는 양의 값을 갖는데, 본 연구에서는  $c_1 = 8, c_2 = 1$ 을 사용하였다. 또한 휴리스틱 알고리즘의 최대 스텝수는 5로 제한하였다.

본 연구에서 제안하는 1차원 이웃탐색 휴리스틱 (1D NSH) 알고리즘은 다음과 같다.

- Step 1.  $k = 0$ ,  $d^{(k)} = (d_1^{(k)}, \dots, d_p^{(k)})$ , 여기에서  $d_i^{(k)} = \text{discrete uniform}(0,1)$
- Step 2.  $j(k) = \text{argmax} \{ \Delta(\tilde{d}_j, d^{(k)}) \}$
- Step 3. If  $\Delta(\tilde{d}_{j(k)}, d^{(k)}) > 0$ ,  $d^{(k+1)} = \tilde{d}_{j(k)}$ , go to Step 4. Otherwise, stop.
- Step 4.  $k = k + 1$ . If  $k < K$ , go to Step 2. Otherwise stop.

위 Step 2에서  $j(k) = \text{argmax} \{ \Delta(\tilde{d}_{ji}, d^{(k)}) \}$ 로 대체하면, 2차원 이웃탐색 휴리스틱 (2D NSH) 알고리즘이 된다.

#### IV. 실험 결과

본 논문에서 제안한 정수계획법 모형과 휴리스틱 알고리즘은 AMPL<sup>[11]</sup> 및 CPLEX<sup>[12]</sup> 소프트웨어를 사용하여 프로그래밍 하였다. 코드비트 행렬은 10x10에서 20x20까지 6가지 종류를 고려하였고, 제어비트는 최소 6에서 최대 10까지 변화하였다. 각 행렬 및 제어비트 크기 조합에 대하여 임의의 20개 소스비트를 생성하였다. 소스비트의 값은 0, 1 이산균등분포를 따른다. 표 1과 2에는 GSIP 모형을 적용한 결과를 보여준다. 표 2의 경우, 식 (4) 대신에

$$|\sum_{i,j} c_{i,j} - m^2/2| \leq 2 \quad (9)$$

를 사용하였다. GSIP 문제에서 (4)의 제약식을 갖는 코드워드를 balanced 코드워드, 식 (9)를 사용한 코드워드를 pseudo-balanced 코드워드로 부른다. 각 행렬에서 제어비트가 많을수록 평균 천이강도 값이 증가됨을 알 수 있다. 또한 행렬이 커짐에 따라 천이

표 1. balanced 코드워드의 천이강도 최소, 최대, 평균값  
Table 1. The minimum, maximum, and average strength values for the balanced codeword.

control bit array size	6			7			8			9			10		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
10x10	1	3	2.4	2	3	2.95	2	3	2.85	3	4	3.2	3	4	3.35
12x12	2	5	3.5	3	5	3.8	3	5	4.2	4	6	4.9	4	6	5.2
14x14	3	7	5.45	4	7	5.6	5	8	6.55	6	8	6.9	7	8	7.05
16x16	7	7	7.35	6	7	7.3	7	7	7.75	7	8	8.1	8	8	8.5
18x18	6	10	8.05	6	10	8.55	8	10	8.95	9	10	9.35	9	10	9.75
20x20	4	6	8	8	6	9.1	9	9	10	10	9	10.8	10	9	11.2

표 2. pseudo-balance 코드워드의 천이강도 최소, 최대, 평균값.  
Table 2. The minimum, maximum, and average strength values for the pseudo-balanced codeword.

control bit array size	6			7			8			9			10		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
10x10	2	4	3	3	4	3.35	3	4	3.45	3	4	3.6	3	4	3.8
12x12	3	5	4.2	3	6	4.25	4	6	4.55	4	6	5	4	6	5.45
14x14	4	7	5.5	4	7	5.85	5	8	6.7	5	8	6.95	6	8	7.1
16x16	6	9	7.4	6	9	7.4	7	9	7.95	7	9	8.2	8	9	8.6
18x18	7	13	8.6	8	11	8.85	8	10	9.05	9	10	9.35	9	11	9.9
20x20	8	12	9.05	8	11	9.4	9	11	10.1	10	12	10.9	11	13	11.3

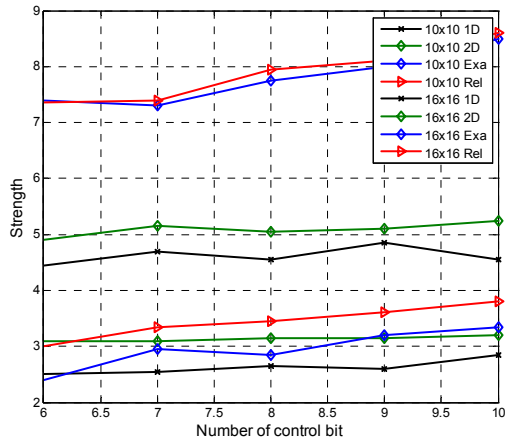


그림 2. 10x10, 16x16 행렬에서 1D, 2D 휴리스틱의 성능.  
Fig. 2. Performance of 1D, 2D heuristics for 10x10, 16x16.

강도 값이 증가한다. 표 1의 balanced 코드워드에서 계산한 천이강도 값이 pseudo-balanced 코드워드 문제의 하한이 되므로, pseudo-balanced 코드워드에서 계산한 평균 천이강도 값이 더 큰 값을 갖고 있다. 그림 2에서는 10x10, 16x16 행렬에서 1D NSH, 2D NSH 방법이 계산한 평균 천이강도 값과 정수계획법 모형의 pseudo-balanced 행렬의 천이강도 값을 비교하였다. 그림 2에서 '10x10 1D', '10x10 2D'는 1D NSH와 2D NSH의 평균 강도이고, '10x10 Exa'는 balanced 코드워드의 강도, '10x10 Rel'은 pseudo-balanced 코드워드의 값을 표시한다. 10x10의 경우, 1D NSH는 10x10 Rel에 대하여 평균 77%의 천이강도 값, 16x16의 경우는 평균 59%의 천이강도 값을 기록하였다. 2D NSH는 1D NSH에 비해 각각 12%, 6% 우수한 성능을 기록했다. 그림 2에서 고려한 생성한 전체 200개의 문제에 대하여 1D NSH가 계산한 코드워드의 '1' 픽셀의 값이 100% balanced 된 값  $m^2/2$ 와의 차이는 평균 2.65로 기록되었다. 따라서 1D NSH에서 계산한 코드워드들의 '0', '1' 픽셀의 분포는 거의 균등하였다.

### V. 결 론

본 논문에서는 GS 인코딩 과정을 0-1 변수를 사용하여 선형부등식과 등식으로 수식화한 후, 최대 천이강도 값을 갖는 코드워드를 생성하는 제어비트를 계산하는 0-1 정수계획법 모형을 개발하였다. 또한 계산복잡도가 간단한 1차원 및 2차원 이웃탐색 휴리스틱 알고리즘을 개발하였다. 개발된 모형은 다양한

코드비트 행렬 및 제어비트를 갖는 문제들에 적용하여 성능을 비교하였다.

본 논문에서 개발한 정수계획법 모형은 사용자 스위치가 입력 파라미터일 때, 최적 제어비트를 결정하는 최대화 문제이지만, GS 인코딩이 계산하는 천이강도값이 최악이 되는 최악 입력데이터를 파악하여 특정한 제어비트수, 스크램블러 다항식의 worst case 하한을 계산하는 데 이용할 수 있다.

### References

- [1] L. Hesselink, S. S. Orlov, and M. C. Bashaw, "Holographic data storage systems," *Proc. IEEE*, vol. 92, no. 8, pp. 1231-1280, Aug. 2004.
- [2] V. Vadde and B. V. K. V. Kumar, "Channel modeling and estimation for intrapage equalization in pixel-matched volume holographic data storage," *Applied Optics*, vol. 38, no. 20, pp. 4374-4386, July 1999.
- [3] A. Vardy, M. Blaum, P. H. Siegel, and G. T. Sincerbox, "Conservative arrays: multidimensional modulation codes for holographic recording," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 227-230, Jan. 1996.
- [4] R. Talyansky, T. Etzion, and R. M. Roth, "Efficient code constructions for certain two-dimensional constraints," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 794-799, Mar. 1999.
- [5] K. A. S. Immink, *Codes for Mass Data Storage Systems*, Shannon Foundation Publishers, 1999.
- [6] I. J. Fair, W. D. Grover, W. A. Krzymien, and R. I. MacDonald, "Guided scrambling: a new line coding technique for high bit rate fiber optic transmission systems," *IEEE Trans. Commun.*, vol. 39, no. 2, pp. 289-297, Feb. 1991.
- [7] W. Y. H. Wilson, K. A. S. Immink, X. B. Xi, and C. T. Chong, "An efficient coding technique for holographic storage with the method of guided scrambling," *Proc. SPIE*, vol. 4090, pp. 191-196, Sep. 2000.
- [8] N. Y. Kim, J. Lee, Y. Hong, and J. Lee,

“Optimal number of control bits in the guided scrambling method for holographic data storage,” *Japanese J. Applied Physics*, vol. 44, pp. 3449-3452, May 2005.

- [9] J. Lee and J. Lee, “New DC-suppression method of modulation codes for high density optical recording systems,” *J. KICS*, vol. 27, no. 1A, pp. 13-17, Jan. 2002.
- [10] C.-L. Wang, Y. Ouyang, and F.-H. Huang, “A low-complexity peak-to-average power ratio reduction technique for OFDM systems using guided scrambling coding,” in *Proc. IEEE VTC 2007*, pp. 2837-2840, Dublin, Ireland, Apr. 2007.
- [11] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL*, Scientific Press, 1993.
- [12] IBM, *IBM ILOG CPLEX 12.4 User’s Manual*, 2012.

이 재 진 (Jaemin Lee)



1983년 2월 연세대학교 전자공학과 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 연구소 책임 연구원

1997년 3월~2005년 8월 동국대학교 전자공학과 부교수

2005년 9월~숭실대학교 정보통신전자공학부 교수  
<관심분야> 통신이론, 채널코딩, 기록저장 시스템

박 태 형 (Taehyung Park)



1986년 2월 고려대학교 산업공학과 학사

1989년 2월 고려대학교 산업공학과 석사

1998년 6월 Virginia Tech 산업시스템공학과 박사

2001년~현재 숭실대학교 산

업정보시스템공학과 부교수

<관심분야> 정수계획법, 통신네트워크