Journal of Institute of Control, Robotics and Systems (2012) 18(12):1096-1100
http://dx.doi.org/10.5302/J.ICROS.2012.18.12.1096 ISSN:1976-5622 eISSN:2233-4335

신장 트리 기반 표현과 MAX CUT 문제로의 응용

A Spanning Tree-based Representation and Its Application to the MAX CUT Problem

현 수 환, 김 용 혁, 서 기 성*
(Soohwan Hyun[1], Yong-Hyuk Kim[2], and Kisung Seo[1])
[1]Seokyeong University
[2]Kwangwoon University

Abstract: Most of previous genetic algorithms for solving graph problems have used a vertex-based encoding. We proposed an edge encoding based new genetic algorithm using a spanning tree. Contrary to general edge-based encoding, a spanning tree-based encoding represents only feasible partitions. As a target problem, we adopted the MAX CUT problem, which is well known as a representative NP-hard problem, and examined the performance of the proposed genetic algorithm. The experiments on benchmark graphs are executed and compared with vertex-based encoding. Performance improvements of the spanning tree-based encoding on sparse graphs was observed.

Keywords: change of basis, graph encoding, genetic algorithms, MAX-CUT, spanning tree, sparse graph

I. 서론

다양한 실제 응용 문제들이 그래프를 통해 편리하게 표현되고, 그래프 이론의 도움을 받아서 문제 해결이 용이해지는 경우가 매우 많다[1].

최대 단절(MAX-CUT) 문제는 그래프를 두 개의 부분 집합으로 나누는 단절(cut)의 크기가 가장 큰 것을 찾는 것으로, NP-complete 문제중의 하나로 알려져 있다[1,2]. MAX CUT 문제는 컴퓨터 이론 분야에서 주요하게 다루어지고 있을 뿐 아니라 다양한 분야에 응용되고 있다. VLSI와 PCB 설계 과정에서 등장하는 레이어(layer) 할당 문제와 병렬 프로세스 스케쥴링 문제 등을 MAX CUT 문제로 모델링할 수 있다[3,4]. 가장 유명한 응용분야로는 통계 물리학에서 스핀 글래스(spin glass)의 최소 에너지 상태를 결정하는 문제(ising spin glass)가 있다[4].

유전 알고리즘(GA: Genetic Algorithm) [5]은 대규모 탐색 문제에 대한 강건성이 있기 때문에 NP-hard, 또는 NP-complete 문제에 널리 적용되고 있다. GA 수행과정에서 선택, 유전연산자, 적합도등이 탐색 성능에 대부분의 영향을 미친다고 알려져 있고, 상대적으로 해의 표현(representation, 또는 encoding)은 중요성이 낮게 인식되고 왔다.

그러나 대상 문제나 적용 방식에 따라 탐색 해를 표현하는 인코딩 기법이 달라지면 그 설계 방식이 달라질 수 있고, 결과적으로 완전히 다른 탐색을 하게 되어 성능차이를 가져올 수 있다. 그 동안 제한된 범위내에서 유전 알고리즘에서 인코딩의 중요성을 주장한 연구들이 소개 되어 왔다. 비교적 단순한 형태의 인코딩 변환이지만, Kim 등[6]은 염색체를 표현하는 유전자의 위치를 최대한 관련 있는 것들끼리 가까이 재배치하도록 함으로써 다양한 문제에서 유전 알고리즘의 성능을 개선시킨 바 있다.

이를 더 일반화시켜 해를 표현하는 염색체가 가지는 유전자들 사이의 연결관계가 최대한 독립적이 되도록 가역선형변환(invertible linear transformation)을 이용해 해를 재 표현하면 유전 알고리즘의 성능이 크게 향상될 수 있음을 보인 연구들도 있었다[7,8,16]. 그런데, 이 연구들은 인코딩 변환의 중요성을 보여줄 뿐 구체적인 변환 방법에 관한 체계성이 없었다.

특히, 응용가능성이 많은 그래프 문제에 대해서는 다양한 인코딩 방법 대신, 정점(vertex) 기반 인코딩이 구현의 편의성으로 인해 대부분 선택되고 있고[9], 드물게 간선(edge) 중심의 인코딩 기법에 대한 연구도 진행된 적이 있으나, 유효한 그래프 분할 해가 구해졌는지를 확인하기가 매우 어려운 문제점이 있다[10,11].

본 논문은 기존의 정점 기반 인코딩 기법과는 달리, 주어진 그래프의 신장트리(spanning tree)를[1] 이용하여 가능해만을 찾을 수 있는 간선(edge) 중심의 인코딩 기법을 제안하고, MAX CUT [2] 그래프 문제를 대상으로 구현된 유전 알고리즘의 성능을 분석한다. 제안된 기법은 정점에 비해 간선의 수가 적은 희소(sparse) 그래프에서 보다 높은 성능이 기대된다. 특히, 요즘 부각되고 있는 소셜 네트워크 등의 실생활에 밀접하게 관련된 그래프들이 대부분 희소 그래프인 것을 감안하면, 본 논문에서 제안된 방법이 많은 가능성을 가지고 있다고 사료된다.

II 장에서는 기존의 연구와 MAX CUT 문제에 대해서 논의한다. III 장은 제안된 신장 트리 기반 인코딩 기법을 다루고, IV 장에서는 다양한 그래프 집합에 대한 실험 결과를

* 책임저자(Corresponding Author)
논문접수: 2012. 8. 14., 수정: 2012. 11. 1., 채택확정: 2012. 11. 4.
현수환, 서기성: 서경대학교 전자공학과
(xjavalov@shhyun.com/ksseo@skuniv.ac.kr)
김용혁: 광운대학교 컴퓨터소프트웨어학과(yhdfly@kw.ac.kr)

보여준다. V 장에서 결론을 정리한다.

II. MAX CUT

그래프 이론에서 컷(cut)은 두 개의 단절된 부분 집합으로 그래프의 정점을 분리하는 것을 말한다. 컷의 집합은 서로 다른 부분 집합으로 분리되는 간선의 집합이 된다. 그래프를 자르는 방식에 따라서 수많은 컷이 존재할 수 있다. (n개 정점을 갖는 그래프인 경우 2^{n-1} 가지의 컷이 존재함.)

그림 1과 같이 컷의 크기가 존재할 수 있는 다른 모든 컷보다도 크기가 크다면 이를 MAX CUT이라 하며, 이를 구하는 것은 NP-complete에 속하는 매우 어려운 문제로 알려져 있다[1,2].

서론에서 언급한 스핀 글래스(spin glass)의 최소 에너지 상태를 결정하는 문제(ising spin glass)에 대한 부연 설명은 다음과 같다. 스핀글라스의 바닥상태(groundstate)는 스핀들이 쩔쩔맴(frustration) 을 최소화 하는 원자 배열을 말하며, 이때, 스핀글라스의 바닥 상태를 찾는 과정이 조합문제에서의 MAXCUT 문제를 해결하는 것과 동치의 관계라고 알려져 있다[4].

III. GA 기반의 MAX CUT 계산

1. 정점 기반의 MAX CUT 계산

정점 기반의 GA 유전자를 구성할 경우, 그림 2와 같이 구성된 그래프의 각 정점들에 대해 GA 유전자를 일대일 매칭하여 구성한다. GA 유전자는 0 또는 1의 값을 가지며, 이는 해당 위치의 정점이 갖는 파티션 번호를 의미하게 된다(그림 3).

그리고 각각의 간선에 대해 시작 정점과 끝 정점이 서로 다른 파티션 번호를 갖는 것을 컷 간선(cut edge)으로 계산한다. 대상 그래프에 존재하는 모든 간선에 대해 컷 간선 여부를 계산하고, 이를 합산하여 총 컷 크기를 얻어낸다.

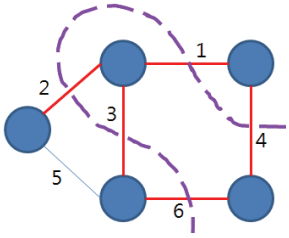

그림 1. MAX CUT 예시.

Fig. 1. Example of MAX CUT.

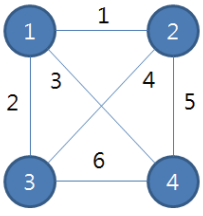

그림 2. 그래프 예시.

Fig. 2. Example of a graph.

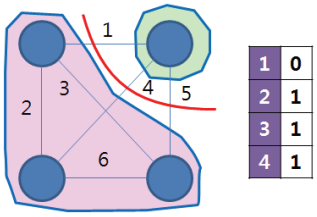

그림 3. 정점 기반의 GA 유전자 표현.

Fig. 3. A Representation of GA.

2. 간선 기반의 MAX CUT 계산

간선을 기반으로 컷을 계산하기 위해서는 우선 그림 4와 같이 대상 그래프에 대한 신장트리를 구한다. 그리고 신장트리의 각각의 간선에 대해 대상 그래프와 매칭시켜 해당 간선의 부분 집합을 구성한다. 만약 신장트리의 간선 1번의 부분 집합을 구하려면, 1번 간선이 연결되어 있는 정점과 나머지 정점들을 다른 파티션으로 계산하고, 1, 4, 5번의 간선을 1번에 대한 부분 집합으로 구성하는 형태이다(그림 5).

그림 6에서와 같이 신장트리의 간선에 대해 간선 기번의 GA 유전자를 일대일로 매칭시키며, 정점 기반의 방식과 마찬가지로 0과 1의 값으로 간선에 대한 파티션 번호를 기록한다. 컷은 간선이 1이면 해당 간선의 부분 집합에 속하는

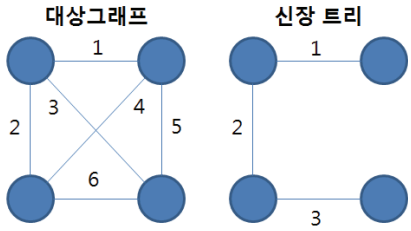

그림 4. 대상 그래프와 신장트리 예.

Fig. 4. Example of target graph and its spanning tree.

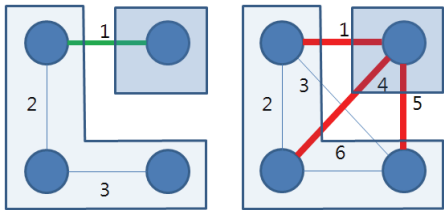

그림 5. 신장 트리의 간선과 이에 대응하는 부분 집합.

Fig. 5. Edge of spanning tree and its corresponding sub-set.

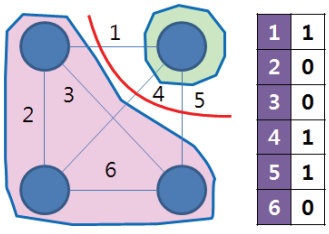

그림 6. 간선 기반의 GA 유전자 표현.

Fig. 6. A Representation of GA Chromosome for edge-based encoding.

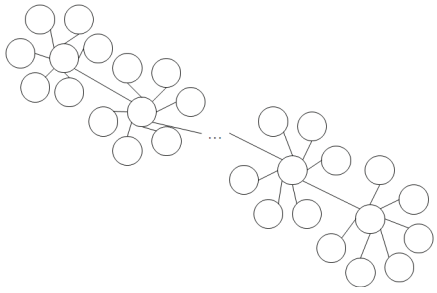

그림 7. 희소 그래프의 형태.
Fig. 7. Example of a sparse graph.

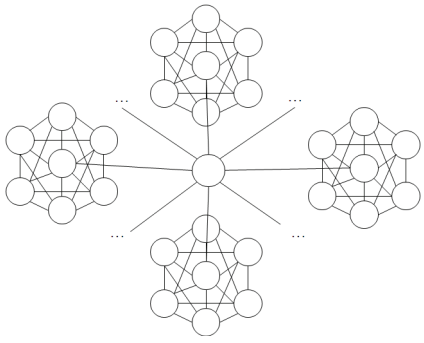

그림 8. 조밀 그래프의 형태.
Fig. 8. Example of a dense graph.

간선을 컷으로 기록한다. 여기서 만약 다른 간선의 부분 집합에도 컷으로 기록되었던 간선이 속한다면, 그것은 다시 컷이 아닌 것으로 기록한다. 이와 같이 모든 GA 유전자에 대해 컷을 계산하여 합하면 해당 그래프에 대한 총 컷의 크기를 얻어낼 수 있다.

3. 희소 그래프와 조밀 그래프

그래프는 정점과 간선의 비율에 따라서 희소 그래프와 조밀 그래프로 나눌 수 있다. 희소 그래프는 정점당 간선의 수가 적은 그래프 이며, 조밀 그래프는 정점당 연결 가능한 최대 간선의 수에 가까운 간선을 가진 그래프이다. 그림 7와 8에 각각 희소 그래프와 조밀 그래프의 예가 나와 있다. 희소 그래프와 조밀 그래프를 나누는 구체적인 기준은 없지만, 본 연구에서는 희소 그래프를 정점/간선 비율이 50 % 이상인 그래프로 구분하여 사용한다.

IV. 시뮬레이션 및 결과 고찰

1. 실험 환경

실험에 사용된 GA 파라미터는 표 1과 같다. 기본적인 일점 교배 연산자(one-point crossover)와 랜덤 변이(random mutation), 그리고 선택에는 토너먼트 선택 연산자(tournament selection)를 사용하였다. 구현은 그래프 관련 라이브러리인 Boost Graph Library [12]와 진화 연산 프레임워크인 Open Beagle [13]을 사용하였다. 신장 트리를 구하는 알고리즘으로는 Kruskal 알고리즘을 사용 하였다.

또한, 실험은 각각의 인스턴스에 대해 30회 반복 수행되었으며, 컷의 크기를 구하는 방식 외에는 어떠한 차이도 두지 않고 실험하였다.

표 1. 유전 알고리즘의 파라미터.
Table 1. Genetic parameters.

Parameter	Value
Max. # of generations	100
Population size	100
Crossover rate	0.9
Mutation rate	0.1
Tournament size	7

실험에 사용된 데이터들은 다음과 같다.

Gn.p: n 개의 정점을 가진 랜덤 그래프로 두 개의 정점 사이에 간선이 존재할 확률이 p로 주어진다. G1000.01 은 1000개 정점을 가지며, 임의의 두 정점간의 간선의 존재 확률이 p=.01 이다.

Un.d: n 개의 정점을 가진 기하적인 랜덤 그래프로 정점당 가지수 d 를 나타낸다. U500.10 은 500개의 정점으로 구성된 기하적인 랜덤 그래프로 정점당 기대 가지수가 10 을 의미한다.

pcart.n: n 개의 정점을 가진 캐터필라 그래프로 각 정점당 6개의 가지를 가진다. prcart.n 은 유사한 그래프로 각 정점이 sqrt(n) 개의 가지를 포함한다.

pgridn.b: n 개의 정점을 가진 격자형 그래프로서 최적의 최소 컷 크기가 b로 알려진 것이다. pcgridn.b 는 같은 그래프이나 경계가 감싸여져 있다.

이들 그래프 종류가 어떻게 생성되었는지에 대한 자세한 정보는 참고문헌 [14,15]에 나와있다.

2. 실험 결과

표 2는 희소 그래프에 대한 실험 결과이다. 정점과 간선의 비율이 50 % 이상으로 정점 1개당 간선 2개 혹은 그 이하로 연결되어 있는 그래프이다. 실험 결과를 보면 정점/간선의 비율이 높은 그래프에 대해 더 큰 크기의 컷을 찾아낼 수 있음을 확인할 수 있다. 정점/간선의 비율이 100 %인 cart 그래프 셋의 경우에는 평균 약 6 % 정도의 성능 향상을 볼 수 있으며, 이보다 낮은 비율의 grid 그래프 셋의 경우에는 약 1~2 %의 성능향상만을 확인할 수 있다.

표 2. 희소 그래프에 대한 실험 결과.
Table 2. Experiment results of sparse graph.

인스턴스	정점/간선 비율	컷 (정점)		컷 (간선)		성능비율 (간선/정점)
		컷	표준편차	컷	표준편차	
pcart.352	100 %	224.1	2.9	**242.1**	2.1	108.05%
pcart.702	100 %	419.0	3.3	**444.6**	3.5	106.10%
pcart.1052	100 %	606.8	5.0	**644.9**	5.6	106.28%
prcart.134	100 %	99.4	1.5	**106.1**	1.5	106.74%
prcart.554	100 %	339.0	3.1	**360.2**	2.5	106.26%
prcart.994	100 %	577.9	5.2	**611.3**	4.7	105.78%
pcgrid100.20	49.50 %	145.1	2.6	**146.8**	1.9	101.22%
pcgrid500.42	49.90 %	582.1	4.6	**594.0**	5.1	102.04%
pcgrid1000.40	49.95 %	1113.8	9.0	**1133.2**	7.0	101.74%
pgrid100.0	55.25 %	130.1	2.4	**133.2**	2.0	102.36%
pgrid500.21	52.30 %	556.9	4.3	**569.4**	4.3	102.24%
pgrid1000.20	51.79 %	1075.5	5.6	**1094.6**	7.2	101.77%

표 3. 조밀 그래프에 대한 실험 결과.

Table 3. Experiment results of dense graph.

인스턴스	정점/간선 비율	컷 (정점)		컷 (간선)		성능 비율 (간선/정점)
		컷	표준 편차	컷	표준 편차	
G500.02	21.20 %	**1303.1**	7.2	1293.3	7.4	99.25%
G500.04	9.74 %	**2747.9**	10.0	2718.3	10.5	98.92%
G1000.01	19.74 %	**2711.0**	8.5	2699.2	7.7	99.56%
u500.20	10.97 %	**2410.2**	5.7	2407.4	5.4	99.89%
u500.40	5.68 %	**4575.6**	7.1	4563.2	8.2	99.73%

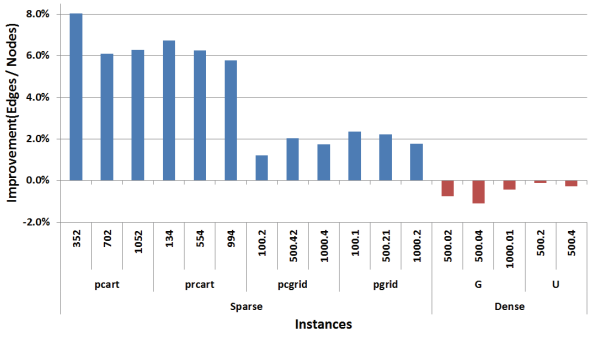

그림 9. 희소 그래프와 조밀 그래프의 성능 요약(1).

Fig. 9. Summary of Performance for sparse and dense graph(1).

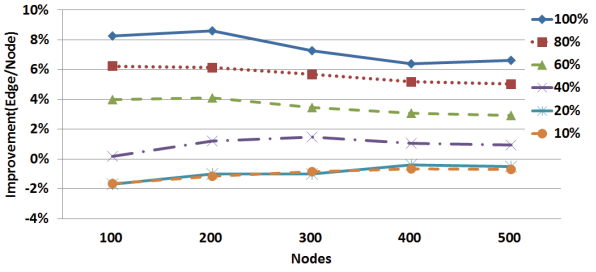

그림 10. 희소 그래프와 조밀 그래프의 성능 요약(2).

Fig. 10. Summary of Performance for sparse and dense graph(2).

표 3은 정점/간선 비율이 50 %보다 낮은 조밀 그래프에 대한 실험 결과이다. 상대적으로 정점 기반의 실험 결과보다 낮은 성능을 보여 주는 것을 확인할 수 있다. 그림 9는 희소 그래프와 조밀 그래프에 대한 표 2와 3의 성능 비교를 막대 분포도로 요약하여 나타낸 것이다. x 축은 실험 대상 그래프 집합을 희소와 조밀 그래프로 나눈 것이다. y 축은 노드 접근법에 대해서 간선 접근법이 우수한 정도를 비율로 나타내었다. 0 %를 기준으로 위와 아래에 성능 비율의 정도가 표시되어 있다.

표 4는 다양한 랜덤 그래프 집합에 대해서 실험을 수행한 결과로, 각 그래프들은 정점/간선의 비율에 따라서 임의적으로 생성되었다. 예로, r100.1.0 은 100개의 정점을 가진 랜덤 그래프로서, 정점/간선의 비율이 1.0 이다. 마지막 열에 있는 r500.0.1 그래프는 500개의 정점을 가지고 있으며, 정점/간선의 비율이 10 % 인 0.1 임을 의미한다. 정점/간선의 비율이100 % - 40 % 대에서는 최대 8 % 까지의 성능 향상이 이루어졌고, 10 %와 20 % 경우에만 1.7 % 이내의

표 4. 랜덤 그래프에 대한 실험 결과.

Table 4. Experiment results of random graph.

인스턴스	정점/간선 비율	컷 (정점)		컷 (간선)		성능비율 (간선/정점)
		컷	표준 편차	컷	표준 편차	
r100.1.0	100%	75.90	1.04	**82.17**	1.32	108.26%
r200.1.0	100%	135.70	1.73	**147.37**	1.64	108.60%
r300.1.0	100%	193.43	2.12	**207.53**	2.33	107.29%
r400.1.0	100%	250.20	3.12	**266.20**	2.68	106.39%
r500.1.0	100%	305.80	3.68	**326.07**	2.85	106.63%
r100.0.8	80%	91.47	1.45	**97.17**	1.37	106.23%
r200.0.8	80%	165.13	2.06	**175.27**	2.56	106.14%
r300.0.8	80%	236.17	3.05	**249.57**	2.30	105.67%
r400.0.8	80%	306.13	3.47	**322.00**	3.25	105.18%
r500.0.8	80%	374.57	3.89	**393.43**	4.01	105.04%
r100.0.6	60%	117.03	1.94	**121.70**	1.88	103.99%
r200.0.6	60%	212.53	2.51	**221.23**	3.09	104.09%
r300.0.6	60%	306.87	3.50	**317.50**	3.08	103.47%
r400.0.6	60%	397.83	4.16	**410.13**	4.43	103.09%
r500.0.6	60%	488.70	3.51	**503.03**	4.79	102.93%
r100.0.4	40%	167.50	3.19	**167.80**	2.39	100.18%
r200.0.4	40%	307.03	3.19	**310.77**	3.25	101.22%
r300.0.4	40%	442.63	4.25	**449.23**	3.60	101.49%
r400.0.4	40%	578.60	4.23	**584.73**	4.93	101.06%
r500.0.4	40%	713.50	4.78	**720.17**	4.60	100.93%
r100.0.2	20%	**308.07**	3.57	302.83	3.35	98.30%
r200.0.2	20%	**581.60**	5.75	575.70	5.07	98.99%
r300.0.2	20%	**848.83**	7.25	840.40	4.64	99.01%
r400.0.2	20%	**1111.1**	6.18	1107.0	6.94	99.63%
r500.0.2	20%	**1374.8**	6.71	1367.8	8.02	99.49%
r100.0.1	10%	**579.8**	4.66	570.4	4.59	98.37%
r200.0.1	10%	**1110.3**	6.24	1097.6	4.90	98.86%
r300.0.1	10%	**1635.6**	7.75	1621.7	10.56	99.15%
r400.0.1	10%	**2155.8**	7.36	2142.	13.54	99.36%
r500.0.1	10%	**2676.1**	8.84	2657.9	14.68	99.32%

저하가 나타났다. 그림 10에는 표 4의 랜덤 그래프에 실험에 대한 성능 비교 결과가 나와 있다.

V. 결론

본 논문에서는 MAX CUT 문제에 대해 새로운 형태의 인코딩 방식을 사용하여 인코딩 방식에 따른 성능 차이에 대해 알아보았다. 결과적으로 실험 환경에 따라 인코딩 방식의 변경만으로도 실험 결과가 변화되는 것을 확인하였다. 또한, 정점 기반의 계산을 사용하는 것은 조밀 그래프에 유리하고, 간선 기반의 계산을 사용하는 것은 희소 그래프에서 좋은 성능을 발휘함을 실험적으로 확인할 수 있었다.

이를 통해서 대상 문제에 따라 적합한 인코딩 기법을 사용하면 더 우수한 성능을 얻을 수 있음을 알 수 있었고, 특히 실제의 대규모 그래프 집합들이 대부분 희소 그래프 형태를 가진다는 점을 고려할 때 본 연구의 결과가 의미있는 도출이라고 볼 수 있다. 또한, 제안된 접근법은 MAX CUT 문제와 유사한, 응용 범위가 넓은 그래프 분할 문제 군에 모두 적용 가능한 기법으로 다양한 확장성을 가진다.

향후 더욱 다양한 그래프 인스턴스에 대한 실험과 신장 트리를 만드는 방식에 따른 성능의 차이에 대한 연구가 필요하다고 생각한다.

참고문헌

[1] D. Avis, A. Hertz, and O. Marcotte, *Graph Theory and Combinatorial Optimization*, Springer, New York, USA, 2005.

[2] S.-H. Kim, Y.-H. Kim, and B.-R. Moon, "A hybrid genetic algorithm for the MAX CUT problem," *In Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 416-423, 2001.

[3] R. Y. Pinter, "Optimal layer assignment for interconnect," *Journal of VLSI Computing Systems*, vol. 1, pp. 123-137. 1984.

[4] F. Barahona, M. Grotschel, M. Junger, and G. Reinelt, "An application of combinatorial optimization to statistical physics and circuit layout design," *Operational Research*, vol. 36, pp. 493-513, 1984.

[5] J. H. Holland, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison- Wesley, Reading, USA, 1989.

[6] Y.-H. Kim, Y.-K. Kwon, and B.-R. Moon, "Problem-Independent schema synthesis for genetic algorithms," *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 1112-1122, 2003.

[7] Y.-H. Kim, "Linear transformation in pseudo-boolean functions," *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 1117-1118, 2008.

[8] Y.-H. Kim and Y. Yoon, "Effect of changing the basis in genetic algorithms using binary encoding," *KSII Transactions on Internet and Information Systems*, vol. 2, no. 4, pp. 184-193, Aug. 2008.

[9] S. M. Antonio, D. Abraham, J. P. Juan, and C. Raúl, "High-performance VNS for the max-cut problem using commodity graphics hardware," *Proc. of the 18th Mini Euro Conference on VNS*, 2005.

[10] M. Armbruster, M. Fügenschuh, C. Helmberg, N. Jetchev, and A. Martin, "Hybrid genetic algorithm within branch-and-cut for the minimum graph bisection problem," *Evolutionary Computation in Combinatorial Optimization*, *Lecture Notes in Computer Science*, vol. 3906, pp. 1-12, 2006.

[11] S. Hyun, Y. Kim, and K. Seo, "A new genetic algorithm using spanning-tree-based encoding for the MAX CUT problem," *Proc. of KIIS Spring Conference 2010*, vol. 20, no. 2, pp. 231-234, 2010.

[12] Boost Library, http://boost.org.

[13] Open Beagle, http://beagle.gel.ulaval.ca.

[14] S. Poljak and Z. Tuza, "Maximum cuts and largest bipartite subgraphs," *American mathematical Society*, 20, 1993.

[15] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 841-855, 1996.

[16] S. Ko, D. Kim, and B-Y. Kang, "A matrix-based genetic algorithm for structure learning of bayesian networks," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 11, no. 3, pp. 135-142, 2011.

현 수 환

2010년 서경대학교 전자공학과 공학사. 2012년 서경대학교 전자공학과 공학석사. 현재 현대중공업 연구소 연구원. 관심분야는 진화연산, 지능로봇.

김 용 혁

1999년 서울대학교 전산과학 전공 이학사. 2001년 서울대학교 컴퓨터공학부 공학석사. 2005년 서울대학교 컴퓨터공학부 공학박사. 2005년~2007년 서울대학교 반도체공동연구소 연구원. 2007년~현재 광운대학교 컴퓨터소프트웨어학과 부교수. 관심분야는 최적화, 진화연산, 지식공학.

서 기 성

1993년 연세대학교 전기공학과 공학박사. 1999년~2003년 Michigan State University, Genetic Algorithms Research and Applications Group, Research Associate. 1993년~현재 서경대학교 전자공학과 부교수. 관심분야는 GP, GA, 진화 로보틱스, 로봇보행, 물체인식과 추적.