

입력 도메인 확장을 이용한 반복 분할 기반의 적응적 랜덤 테스트 기법

신 승 훈[†] · 박 승 규^{††}

요 약

적응적 랜덤 테스트 (Adaptive Random Testing, ART)은 입력 도메인 내에 테스트 케이스를 넓고 고르게 분산시키는 방법을 통해 입력 도메인 내에 존재하는 오류 패턴을 순수 랜덤 테스트 (Random Testing, RT)보다 효율적으로 찾아내기 위한 테스트 케이스 선택 기법이다. 테스트 케이스 선택에 많은 연산량을 필요로 하는 초기 ART 기법인 거리 기반 ART (Distance-based ART, D-ART)와 제한 영역 기반 ART (Restricted Random Testing, RRT)의 개선을 위해 입력 도메인을 반복 분할하는 기법들이 제안되었고, 이 기법들은 낮은 연산량 및 성능 향상 등의 효과를 가져왔다. 하지만, 입력 도메인 반복 분할 기반 기법에서도 기존 ART 기법에서 나타나는 테스트 케이스 분포 불균일 문제가 존재하고, 이는 기법의 확장성에 장애 요소로 작용한다. 따라서 본 논문에서는 반복 분할 기반 기법에서 나타나는 테스트 케이스 분포의 특성을 파악하고, 이를 적정 수준으로 제어하기 위한 입력 도메인 확장 정책을 제안하였으며, 실험을 통해 2차원 입력 도메인에서 3%, 3차원 입력 도메인에서 10% 수준의 성능 향상을 확인하였다.

키워드 : 랜덤 테스트, 적응적 랜덤 테스트, 테스트 케이스 선택

Adaptive Random Testing through Iterative Partitioning with Enlarged Input Domain

Shin, Seung-Hun[†] · Park, Seung-Kyu^{††}

ABSTRACT

An Adaptive Random Testing(ART) is one of test case generation algorithms, which was designed to get better performance in terms of fault-detection capability than that of Random Testing(RT) algorithm by locating test cases in evenly spreaded area. Two ART algorithms, such as Distance-based ART(D-ART) and Restricted Random Testing(RRT), had been indicated that they have significant drawbacks in computations, i.e., consuming quadratic order of runtime. To reduce the amount of computations of D-ART and RRT, iterative partitioning of input domain strategy was proposed. They achieved, to some extent, the moderate computation cost with relatively high performance of fault detection. Those algorithms, however, have yet the patterns of non-uniform distribution in test cases, which obstructs the scalability. In this paper we analyze the distribution of test cases in an iterative partitioning strategy, and propose a new method of input domain enlargement which makes the test cases get much evenly distributed. The simulation results show that the proposed one has about 3 percent of improvement in terms of mean relative F-measure for 2-dimension input domain, and shows 10 percent improvement for 3-dimension space.

Key Words : Random Testing, Adaptive Random Testing, Test Case Generation

1. 서 론

소프트웨어 제품의 품질 보장을 위해 수행되는 소프트웨어 테스트는 소프트웨어 개발 과정에서 매우 중요한 역할을 담당한다. 소프트웨어 테스트 자동화는 이와 같은 소프트웨

어 테스트가 보다 효율적으로 수행될 수 있도록 하며, 여기에는 테스트를 위한 테스트 케이스의 자동 생성 및 생성된 테스트 케이스를 이용한 소프트웨어의 자동 실행 등이 포함된다. 접근이 가장 용이한 소프트웨어 테스트 자동화 방법의 하나인 랜덤 테스트는 입력 도메인 내에서 임의의 값을 선택하여 테스트 케이스를 생성하고, 이를 이용하여 주어진 소프트웨어에 대한 테스트를 수행하도록 한다. 이와 같은 랜덤 테스트는 소프트웨어 테스트 자동화를 위한 가장 기본적인 중요한 방법의 하나로[7], 다양한 어플리케이션에 성공적으로 적용되어왔다[9, 11]. 하지만 랜덤 테스트는 테스트되는 소프

[†] 준 회 원 : 아주대학교 정보통신공학과 박사과정
^{††} 정 회 원 : 아주대학교 정보통신대학 교수
논문접수 : 2008년 3월 4일
수정일 : 1차 2008년 5월 19일
심사완료 : 2008년 5월 20일

트웨어에 대한 정보를 전혀 이용할 수 없다는 제약을 가지기 때문에, Chen 등은 소프트웨어 내의 오류들이 특정한 형태를 이루는 경우, 테스트 케이스를 입력 도메인 내에 넓고 고르게 분산시켜 순수 랜덤 테스트에서보다 더 효율적으로 오류를 찾아낼 수 있도록 하는 적응적 랜덤 테스트(Adaptive Random Testing, ART)를 제안했고, 이를 기반으로 하는 다양한 응용 기법들이 제안되어 왔다.[4, 6-10].

기존에 제안된 ART 기법들은 순수 랜덤 테스트와 비교 시 30%에서 최대 50% 정도의 향상된 성능을 보이지만[7], 테스트 케이스 사이의 거리 계산에서 발생하는 많은 연산량으로 인한 테스트 케이스 선택 지연과[4, 8], 테스트 케이스의 입력 도메인 내 분포 불균일성으로 인해 입력 도메인 확장 및 오류 영역 형태 변화 시 기법의 성능이 저하되는 등의 단점을 가진다[4, 8, 10]. 이중 많은 연산량에 의한 테스트 케이스 선택 지연은 입력 도메인의 반복 분할 기법을 통해 효율적으로 개선되었으며[7], 테스트 케이스의 입력 도메인 내 분포 불균일성 문제 해결을 위한 노력들이 시도되어 왔다[9-10]. 하지만 입력 도메인 내 분포 불균일 문제 해결을 위해 제안된 기존 정책들은 테스트 케이스 사이의 거리 계산을 이용하는 기법을 위한 것으로, 입력 도메인의 반복 분할을 이용하는 기법에 적용하기에는 무리가 있다. 따라서 본 논문에서는 입력 도메인의 반복 분할 기법에 적용 가능한 입력 도메인 내 분포 불균일성 완화 정책을 제시하고, 제안된 정책의 효율성을 실험을 통해 검증한다.

2. 정의 및 관련 연구

2.1 정의

하나의 소프트웨어에 대한 유한 크기의 입력 데이터 집합을 입력 도메인이라 하고, 입력 도메인 내에 존재하는 전체 입력 데이터 중 소프트웨어의 오류를 유발하는 입력 데이터의 비율을 오류 비율(θ)이라고 한다. 소프트웨어의 오류를 유발하는 입력은 입력 도메인 내에서 몇 가지 유형으로 군집을 이루며, 이를 오류 영역 혹은 오류 패턴이라고 한다. Chan 등에 따르면 오류 패턴은 Block 패턴, Strip 패턴 및 Point 패턴 등으로 구분되는데, 이 중 Block 및 Strip 패턴이 상대적으로 빈번하게 나타나는 오류 패턴[5]이다. 따라서 기존 ART 관련 연구에서는 Block 패턴을 중심으로 제안된 정책의 성능 평가가 이루어졌다.

F-measure[8]는 첫 번째 오류를 발견하는 데까지 소요된 테스트 케이스 개수를 의미하는데, 이는 오류 발견 시 테스트의 진행을 멈추고 오류의 원인을 파악하는 작업을 수행하는 일반적인 소프트웨어 테스트 환경에 비추어 볼 때, P-measure¹⁾나 E-measure²⁾보다 직관적인 성능 평가 지수라고 할 수 있다. 이에 따라 거의 모든 ART 관련 연구에서 기법의 성능 평가에 F-measure가 이용되었으며, 본 논문에서도 기법들의 성능 비교에 F-measure를 사용한다.

순수 랜덤 테스트의 이론적 F-measure는 $F_{RT} = 1/\theta$ 이고, F-measure는 입력 도메인 내의 오류 비율에 영향을 받기 때문에 기법들의 성능의 비교에는 순수 랜덤 테스트가 갖는 이론적 F-measure에 대한 평균 상대 지수가 이용된다.

$$\text{평균 상대 F-measure} = \text{mean } F_{ART} / F_{RT}$$

즉, 평균 상대 F-measure가 1인 경우, 순수 랜덤 테스트와 같은 성능을 가짐을 의미하며, 0에 가까운 값일수록 우수한 성능을, 1보다 큰 값은 순수 랜덤 테스트보다 낮은 성능을 가짐을 나타낸다.

2.2 관련 연구 및 문제점

2.2.1 테스트 케이스 자동 생성 기법

소프트웨어 테스트 단계 중 가장 많은 노력이 소모되며 자동화가 어려운 단계가 테스트 케이스를 생성하는 단계[3]이기 때문에, 테스트 케이스의 생성의 자동화는 효율적인 소프트웨어 테스트를 위해 매우 중요한 위치를 차지한다고 할 수 있다. 이러한 테스트 케이스의 자동 생성을 위한 연구를 살펴보면, 우선 블랙박스 테스트를 기반으로 하는 오류 패턴을 이용하는 랜덤 기법[4, 6-10]과 유전자 알고리즘 기반 기법[3] 등이 있으며, 화이트박스 테스트를 기반으로 하는 기법에는 알로이 명세를 이용한 기법[2] 등이 있다.

유전자 알고리즘을 이용하는 기법은 뮤테이션 테스트를 위해 생성되는 뮤턴트 프로그램에 대한 테스트 케이스의 뮤테이션 점수를 유전자 알고리즘의 적합성 함수(fitness function)로 이용하여, 좀 더 많은 수의 뮤턴트를 식별해낼 수 있는 새로운 테스트 케이스를 생성하는 것을 목적으로 한다. 이 기법은 소프트웨어의 소스 코드 분석 없이 단지 소프트웨어에 대한 입출력 관계만을 이용하는 블랙박스 테스트에 기반을 두고 있으며, 랜덤 테스트보다 우수한 성능을 보인다[3]. 하지만 뮤턴트 프로그램의 작성 및 새로운 테스트 케이스 생성에 뮤테이션 점수가 이용되므로 테스트 되는 소프트웨어에 대한 지식이 전혀 없다고 보기 어려워 순수한 블랙박스 테스트를 전제로 하는 랜덤 테스트와는 차별성을 가진다.

한편 알로이(alloy) 명세를 이용한 기법은 화이트 박스 테스트를 기반으로, 소프트웨어의 실행을 요구하지 않는 변형된 목적 지향적 테스트 케이스 생성 기법(goal-oriented)이다. 이 기법은 입력 소프트웨어를 관계 논리 언어인 알로이 명세로 변환하고, 생성된 명세로부터 알로이 분석기를 통해 소프트웨어의 경로 및 이를 실행할 수 있는 입력 값을 생성한다[2]. 이 기법은 소프트웨어 상의 특정 제어점의 일부만으로도 테스트 케이스 생성이 가능하다는 점과 사용자가 소프트웨어의 경로를 선택해야하는 부담을 덜어주는 등의 장점을 가지지만, 화이트박스 테스트를 위한 기법이기 때문에 소프트웨어의 소스 코드를 분석하여 계산 그래프(computation graph) 및 유한 상태 모델을 작성하는 등 블랙박스 테스트 기반 테스트 케이스 자동 생성 기법에 비해 상대적으로 복잡한 절차가 요구된다.

1) 주어진 테스트 케이스 세트를 이용해 오류를 발견할 확률

2) 주어진 테스트 케이스 세트를 이용했을 때 발견되는 오류의 평균 개수

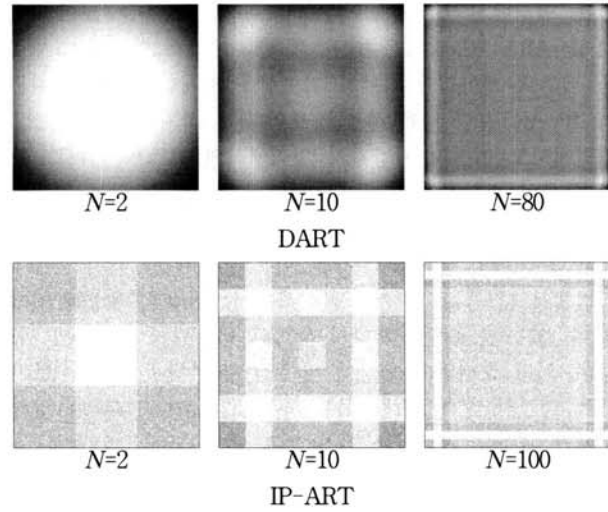
2.2.2 적응적 랜덤 테스트

테스트 케이스의 생성에 유한 개수의 테스트 케이스 후보 데이터를 이용하는 거리 기반 ART(Distance-based ART, D-ART)[8]와 이전 테스트 케이스를 중심으로 설정되는 제한 영역을 기반으로 하는 제한 영역 기반 ART (Restricted ART, RRT)[4]는 ART의 가장 기초적이며 대표적인 기법이다. 하지만 D-ART와 RRT의 테스트 케이스 선택 정책은 테스트가 진행될수록 테스트 케이스 선택의 기준이 되는 거리 계산에 필요한 연산량이 증가하여 테스트 케이스 선택에 긴 연산 시간을 요구하는 단점을 가진다. 이러한 단점을 극복하기 위해 입력 도메인을 그리드 형태로 분할하고, 분할된 영역과 영역 내의 입력 데이터를 임의로 선택하여 테스트 케이스를 생성하는 입력 도메인 분할 기법들이 제안되는데, 이러한 기법들 가운데 입력 도메인 반복 분할 기법(ART through Iterative Partitioning, IP-ART)은 D-ART나 RRT보다 아주 적은 연산 시간을 사용하지만, 효율성은 이들 기법과 유사한 특성을 보인다[7]. IP-ART는 테스트의 진행에 따라 입력 도메인의 분할 수준(P)을 점점 높여가면서, 그리드를 구성하는 한 셀의 크기를 줄여가는 방식을 기본으로 하고, 테스트 케이스가 선택된 셀과 그 셀의 이웃 셀을 제한 영역으로 설정한다. 즉, 입력 도메인 분할을 통해 RRT에서의 제한 영역을 거리 계산 없이 적용할 수 있도록 한 방법이다.

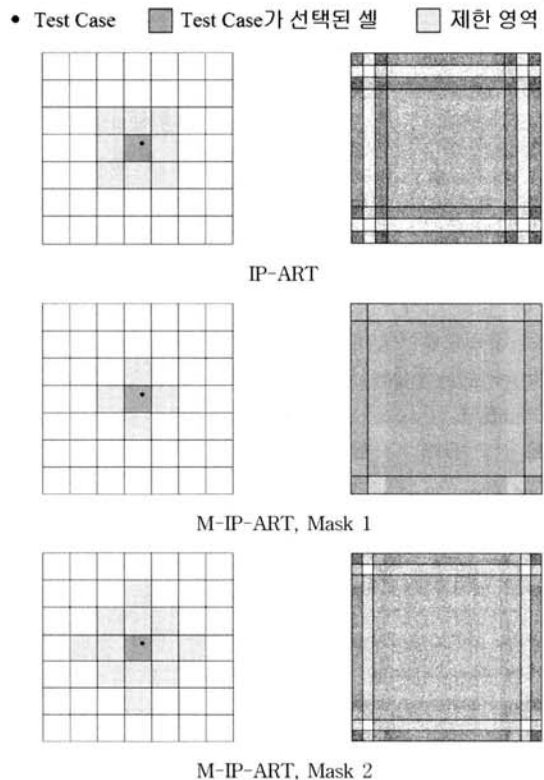
하지만 D-ART, RRT 및 IP-ART 기법 모두에서는 선택된 테스트 케이스의 입력 도메인 내 분포 불균일 현상이 나타나는데, 이는 오류 비율의 변화, 오류 영역의 입력 도메인 내 위치 및 입력 도메인의 차원수 증가에 따른 기법의 성능 편차 혹은 성능 저하와 같은 단점을 야기한다. (그림 1)은 D-ART와 IP-ART를 사용했을 때 나타나는 입력 도메인 내 테스트 케이스 분포이며, 총 5000회의 테스트 수행 과정에서 선택된 N 번째 테스트 케이스의 공간 분포를 나타낸다. 그림에서 테스트 케이스는 점으로 표시되었고, 따라서 그림의 어두운 부분에서 상대적으로 더 높은 빈도의 테스트 케이스 선택이 이루어지고 있음을 나타낸다[1, 10].

D-ART나 RRT 기법 사용 시 나타나는 입력 도메인 내 테스트 케이스 분포 불균일성을 해결하기 위해 Mayer 등은 오류 영역의 가상 이동[9]과 입력 도메인의 가상 확장 정책[10] 등을 제안하였으며, 이들은 높은 차원수를 가지는 입력 도메인에서 기법의 성능 저하를 완화시키는 효과를 가져왔다. 하지만 이와 같은 정책들은 D-ART 및 RRT 기법의 사용 시 나타나는 테스트 케이스 편중 현상을 대상으로 하기 때문에 이와는 다른 형태의 테스트 케이스 편중 현상을 보이는 기법에서는 테스트 케이스 분포 편중 현상을 적절히 제어할 수 없으며, 이 역시 D-ART와 RRT 기법을 바탕으로 하기 때문에 테스트 케이스 생성에 긴 시간이 요구되는 단점을 가진다.

이에 따라 입력 도메인 분할 기반 정책인 M-IP-ART (Modified IP-ART)가 제안되었으며, 이는 IP-ART 기법이 사용하는 제한 영역의 변경을 통해 IP-ART 기법 사용 시 나타나는 테스트 케이스 분포 편중 현상의 완화를 시도하였



(그림 1) N 번째 테스트 케이스의 분포 [1, 10]



(a) 제한 영역 마스크 (b) 테스트 케이스 편중 현상($N=50$)

(그림 2) 제한 영역 마스크 및 테스트 케이스

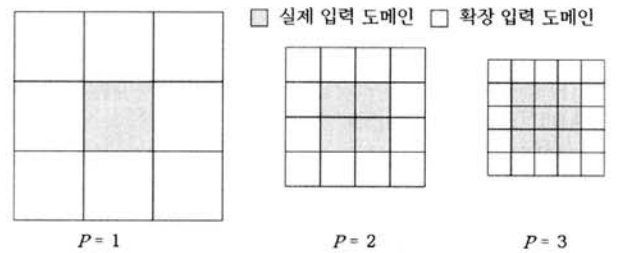
다[1]. M-IP-ART에서 사용하는 제한 영역과 이 제한 영역을 사용했을 때 나타나는 테스트 케이스 분포가 (그림 2)에 IP-ART를 이용한 경우와 비교하여 제시되어 있다. 그림에서와 같이 M-IP-ART에서 사용하는 두 가지 종류의 제한 영역 모두는 IP-ART의 제한 영역이 유발하는 테스트 케이스 편중 현상보다 낮은 수준의 편중 현상을 유발한다. M-IP-ART, Mask 2의 경우, 2차원 입력 도메인을 대상으로

하는 실험에서 IP-ART보다 더 우수한 성능을 보이지만, M-IP-ART, Mask 1은 이전 테스트 케이스 인근에서 새로운 테스트 케이스가 선택될 가능성을 배제할 수 없기 때문에, IP-ART보다 낮은 성능을 보인다[1]. 하지만 M-IP-ART 기법에서도 여전히 가장자리 영역에서의 테스트 케이스 분포 불균일 현상을 확인되며, 이를 완화시킬 수 있는 정책이 요구된다.

입력 도메인의 가상 확장을 이용한 정책은 기존의 ART 기법에서 나타나는 테스트 케이스 편중 현상을 적절히 제어할 수 있는 효율적인 하나의 방법이 될 수 있다. 하지만 이 경우, 입력 도메인의 확장에 따른 관리 부하 증가가 수반되므로 적절한 형태로 구성하는 것이 필요하다. 따라서 입력 도메인 확장을 위해서는 가장 먼저 입력 도메인의 반복 분할 방식을 사용할 때 나타나는 테스트 케이스 분포 편중 현상에 대한 분석이 수행되어야 하고, 확장된 입력 도메인 내에 존재하는 가상 분할 영역과 실제 입력 도메인 내에 존재하는 실제 분할 영역간의 구분을 위한 정책 및 가상 테스트 케이스의 생성으로 인한 테스트 케이스 관리와 입력 도메인 관리를 위한 논리 행렬 관리 등에 수반되는 자원 사용량 증가 등이 고려되어야 한다. 또한 입력 도메인이 테스트 진행에 따라 이전과 다른 크기를 가진 셀로 분할되므로, 기법의 성능 향상 및 용이한 입력 도메인 관리를 위해 이들을 적절히 수용 가능한 형태로 입력 도메인 확장이 이루어져야 한다. 이와 같은 요구 조건이 충족되면 테스트 케이스를 입력 도메인 내에 더욱 균일하게 분포시킬 수 있어 기법의 성능 개선 및 성능 편차 감소 효과를 얻을 수 있으며, 또한 일반적으로 소프트웨어에 대한 입력이 다차원임을 고려할 때, 기법의 다양한 형태의 입력 도메인에 대한 확장성도 기대할 수 있게 된다.

3. 테스트 케이스 선택 정책

(그림 3)은 본 논문에서 제안하는 입력 도메인 반복 분할 방식 기반의 기법에 적용 가능한 입력 도메인 확장 정책을 나타낸다. 확장되는 영역을 구성하는 한 셀의 폭은 실제 입력 도메인을 구성하는 한 셀의 폭과 동일하게 구성하되, 테스트 케이스 분포를 고려해 확장 지수(Enlargement Factor, EF)를 조절하도록 하였다. 따라서 입력 도메인이 2차원일 경우, 확장된 입력 도메인의 분할은 $(1+2EF) \times (1+2EF)$ 그리드에서 시작하여 1씩 증가하는 형태가 되고, 실제 입력 도메인은 확장된 가상 도메인 내부에 위치하게 되며, 그 크기는 입력 도메인 분할에 의한 영향을 받지 않는다. 이러한 구성에서 가상 입력 도메인 영역은 셀의 폭을 기준으로 하는 고정 확장 지수를 사용하기 때문에 테스트 진행에 따라 그 폭이 줄어드는 특징을 보이며, 이 가상 확장 영역 내에서 입력 도메인 반복 분할 방식 기반의 기법에서 나타나는 입력 도메인 가장자리의 테스트 케이스 불균일 현상이 나타나게 된다. 즉, 실제 입력 도메인의 크기 변화 없이 가상 입력 도메인의 정의만을 통하여, 일반 입력 도메인에서 테스트가 진행됨에

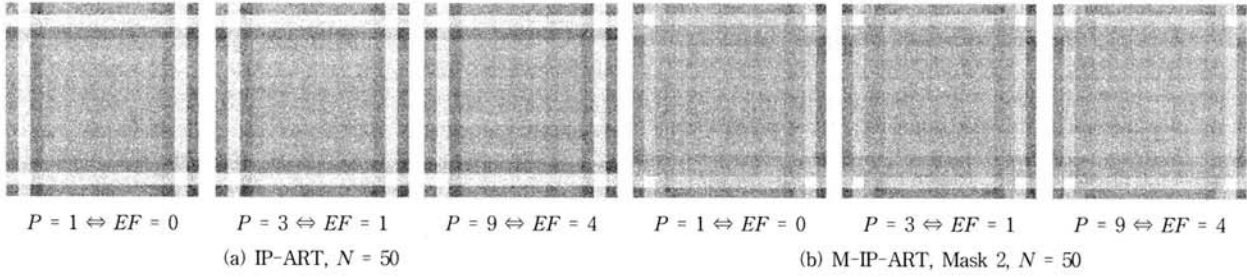


(그림 3) 입력 도메인 확장 ($EF = 1$)

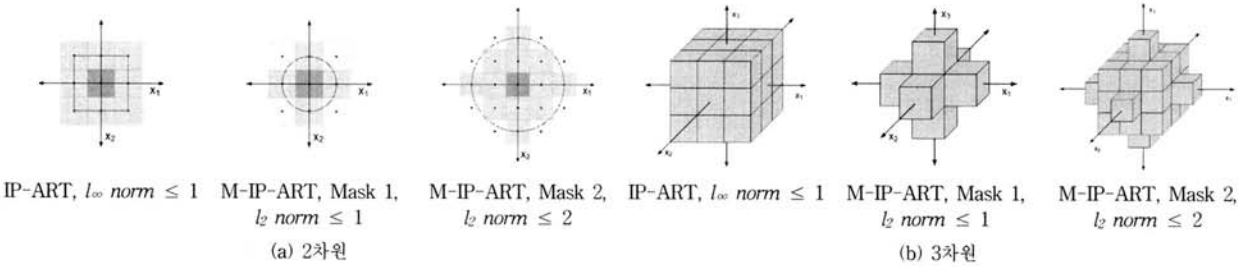
따라 점차 넓어지는 특징을 가지는, 입력 도메인 중심부에 위치한 균일한 테스트 케이스 분포가 이루어지는 영역에 실제 입력 도메인이 위치하도록 할 수 있게 된다.

이러한 입력 도메인 확장 방법을 IP-ART와 M-IP-ART 기법에 적용하기 위해 각각의 기법이 사용되었을 때의 테스트 케이스 분포를 살펴보면, 우선 IP-ART의 경우 입력 도메인 외곽을 둘러싸는 3단의 영역에서 테스트 케이스 분포 불균일 현상이 발생되고, 그 내부는 순수 랜덤 테스트에서 나타나는 것과 유사한 형태의 균일한 분포를 보인다. M-IP-ART의 Mask 1에서는 전체 입력 도메인에서 대체적으로 균일한 분포를 보이나, 입력 도메인 외곽의 1단의 영역에서 안쪽 영역보다 상대적으로 높은 테스트 케이스 불균일 현상을 보이고, M-IP-ART의 Mask 2에서는 IP-ART 기법보다 불균일 현상이 뚜렷하지 않지만, 입력 도메인 외곽을 둘러싸는 2단의 영역이 다른 영역보다 상대적으로 불균일 현상이 심한 편이며, 그 내부는 대체로 균일한 분포를 보이지만, 이 영역 역시 IP-ART의 내부 영역 보다 균일성이 낮다. 이러한 특징을 고려할 때 IP-ART 기법에서는 $EF = 3$ 을, M-IP-ART의 Mask 1에서는 $EF = 1$, Mask 2에서는 $EF = 2$ 를 사용하는 것이 바람직할 것으로 보이며, IP-ART 기법에 의해 생성되는 높은 균일성을 가지는 내부 영역의 이용은 기법의 성능 개선에 상대적으로 더 큰 영향을 미칠 것으로 예상된다. 즉, 입력 도메인의 확장을 고려하지 않은 환경에서는 전체적인 입력 도메인 내 테스트 케이스 분포 균일성이 높고, 선택된 테스트 케이스 사이의 거리가 적절하게 유지되는 M-IP-ART의 Mask 2가 더 우수한 성능을 보였지만[1], 입력 도메인 확장을 통한 테스트 케이스 분포 불균일 영역 배제는 내부 영역에서 균일한 분포를 가지는 IP-ART 기법의 성능에 더 긍정적인 영향을 줄 수 있을 것으로 예상된다.

또한 입력 도메인의 확장이 이루어지는 경우, 각 기법은 자신이 사용하는 제한 영역 마스크로부터 많은 영향을 받게 된다. 입력 도메인 분할 기법은 제한 영역 마스크를 이용하여 제한 영역에 해당하는 영역에서 테스트 케이스가 선택되지 않도록 설정하는 작업을 수행하므로, 넓은 제한 영역 마스크의 사용은 입력 도메인 확장과 입력 도메인의 차원 수 증가에 따라 작업 부하 증가를 유발하여 테스트 케이스 선택 속도 저하에 영향을 주게 된다. 따라서 확장된 입력 도메인을 사용하는 기법과 높은 차원수를 가진 입력 도메인을 대상으로 한 테스트에서는 상대적으로 넓은 제한 영역을 사용하는 M-IP-ART의 Mask 2보다 IP-ART 기법이 더 빠른



(그림 4) 초기 P값 변화에 따른 테스트 케이스 분포



(그림 5) 노름(Norm)을 이용한 제한 영역 마스크

테스트 케이스 선택 시간과 더 높은 확장성을 가질 것이다. M-IP-ART의 Mask 1은 IP-ART 기법보다 더 좁은 제한 영역을 사용해 테스트 케이스 선택은 IP-ART보다 신속하게 이루어지겠지만, 선택된 테스트 케이스 사이의 거리가 충분치 못할 가능성을 배제할 수 없다는 단점으로 인한 성능 저하가 입력 도메인 확장 정책에서도 나타날 것이다.

3.1 최초 분할 수치 변화에 따른 테스트 케이스 분포

본 논문에서 제안한 입력 도메인 확장 정책의 적용을 위해서는 최초 도메인 분할 수치(P)의 변화가 테스트 케이스 분포에 미치는 영향이 거의 없다는 사실이 전제되어야 한다. 즉, 테스트 케이스 분포는 입력 도메인의 확장에 따라 최초 입력 도메인 분할 값이 P=1에서 P=1+2EF로 증가하는데 대해 독립적이어야 한다. 이에 따라 최초 분할 수치가 테스트 케이스 분포에 미치는 영향을 파악하기 위한 실험을 P의 최초 값이 1, 3, 9일 때에 대해 각각 수행하였고, 그 결과가 (그림 4)에 제시되어 있다. (그림 4-(a))는 IP-ART에서의, (그림 4-(b))는 M-IP-ART Mask2에서의 실험 결과이다. 그림에서 나타난 것과 같이 최초 분할 수치의 증가는 테스트 케이스 분포에 거의 영향을 주지 않는 것으로 확인되었으며, 이에 따라 입력 도메인 확장 정책은 입력 도메인 반복 분할 기반 기법의 성능 개선을 위한 정책으로 활용 가능함을 알 수 있다.

3.2 노름(Norm)을 이용한 제한 영역 설정

기존의 IP-ART 기법에서는 2차원 이상의 높은 차원수를 가지는 입력 도메인에 대한 제한 영역 설정에 2차원에서의 제한 영역 설정 방법을 단순 확장하여 이용하도록 하고 있다[7]. 즉, 테스트 케이스가 선택된 셀과 그 셀에 인접한(adjacent) 셀을 제한 영역으로 설정하는 방법을 고차원 입력 도메인에 대해서도 적용하도록 하고 있다. 하지만 입력 도메인의 차

원수가 4차원 이상이 되면, ‘인접한 셀’의 정의가 모호해지기 때문에 이를 명확한 형태로 정의할 필요가 있으며, 더욱이 M-IP-ART 기법의 경우에는 단순한 ‘인접 셀’ 개념을 통해 표현하기 어려운 제한 영역을 사용하고 있기 때문에, 입력 도메인의 차원수가 증가하게 되면, 제한 영역의 표현이 매우 어렵게 된다. 이에 따라 본 논문에서는 입력 도메인의 차원수에 무관하게 적용 가능한 노름(Norm) 기반의 제한 영역 설정 방법을 정의한다.

분할된 입력 도메인을 테스트 케이스가 선택된 셀을 중심점을 원점으로 하는 벡터 공간이라고 할 때, 각 셀의 중심점을 셀을 지칭하는 좌표성분으로 하고, 각 셀의 좌표 성분은 원점으로부터 멀어질수록 1씩 증가한다고 가정하면 입력 도메인 내의 각 셀은 다음과 같은 벡터로 표현이 가능해진다.

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^t$$

이 때, 테스트 케이스가 선택된 셀에서의 거리 즉, 각 셀을 나타내는 벡터의 크기는 노름을 이용해 표현이 가능하며, 이를 이용해 간단한 방법으로 제한 영역을 나타낼 수 있게 된다. 우선 IP-ART 기법에서 사용하는 제한 영역은 테스트 케이스가 선택된 셀을 둘러싼 모든 셀을 포함해야하므로 l_∞ 노름이 사용 가능하고, M-IP-ART 기법의 제한 영역은 셀의 중심점이 테스트 케이스가 선택된 셀로부터 일정 반경 내에 존재하는 셀로 바꿔 표현 가능하므로 l_2 노름을 사용하면 Mask1, Mask2 모듈을 간단히 표현할 수 있다. l_∞ 와 l_2 노름의 정의는 다음과 같고, 이를 이용해 정의한 제한 영역은 (그림 5)와 같은 형태를 취한다.

$$l_\infty \text{ norm} = \|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|,$$

$$l_2 \text{ norm} = \|\mathbf{x}\|_2 = \left\{ \sum_{i=1}^n x_i^2 \right\}^{\frac{1}{2}}$$

<표 2> 입력 도메인 확장 기반의 IP-ART 알고리즘

<p>INPUT (입력 도메인 확장 지수 EF, 제한 영역 마스크 M) D : 실제 입력 도메인, D' : 확장 입력 도메인, P : 입력 도메인 분할 지수 E : 이전에 테스트 된 테스트 케이스의 집합, $E = \{e_1, e_2, \dots, e_N\}$ G : $(P+2EF) \times (P+2EF)$ - 논리 그리드</p> <p>1. $P = 1, E = \emptyset$로 초기화.</p> <p>2. $(P+2EF) \times (P+2EF)$인 그리드 G 생성. 그리드 G를 수용하는 확장 입력 도메인 D' 생성 G를 구성하는 모든 셀을 T로 초기화. $\{e_i \mid e_i \in (E \cap D')\}$인 모든 e_i를 포함한 셀을 기준으로 M 적용. M에 해당되는 G의 셀을 F로 변경.</p> <p>3. while (G 내에 T로 설정된 셀의 수 > 0) { T인 셀 중 임의의 셀 선택. 선택된 셀 내에서 임의의 테스트 케이스를 선택 (TC_{N+1}) TC_{N+1}을 E에 등록 if (TC_{N+1}이 D내에 존재) { TC_{N+1}을 사용하여 테스트 수행. if (테스트 결과 오류 발견) 테스트 종료. } 선택된 셀을 기준으로 M 적용 M에 해당되는 모든 G의 셀을 F로 변경 } <p>4. $P = P + 1$. 2번 단계에서 다시 수행.</p> </p>
--

3.3 수행 알고리즘

본 논문에서 사용하는 입력 도메인 확장을 이용한 알고리즘은 IP-ART 수행 알고리즘[7]을 바탕으로 입력 도메인 확장에 따른 작업이 부가된 형태를 가지며, <표 1>에 기술된 것과 같은 형태로 동작한다.

실제 입력 도메인을 D , 가상 확장된 입력 도메인을 D' 이라할 때, 입력 도메인의 확장을 이용하는 정책은 입력 도메인의 가상 확장에 따라 발생하는 실제 도메인 외부 영역 즉, $D'-D$ 에서 선택된 테스트 케이스에 대한 처리 및 관리를 필요로 한다. 제안하는 알고리즘에서는 실제 도메인 외부에서 테스트 케이스가 선택된 경우, 이를 이용한 실제 테스트는 수행하지 않고, 추후 제한 영역 설정에 이용하기 위해 이전 수행한 테스트 케이스 리스트(E)에 등록만 한다. 또한 P 값의 증가에 따른 확장 입력 도메인의 재분할은 더 좁아진 ($D'-D$) 영역을 생성하도록 하기 때문에, 재분할 후 확장 입력 도메인(D') 외부에 존재하는 이전 테스트 케이스 ($e_i \mid e_i \in (E - D')$)가 존재하게 되고, 이러한 테스트 케이스 e_i 는 이전 수행된 테스트 케이스를 중심으로 생성되는 제한 영역 설정 과정에서 배제된다.

4. 성능 평가

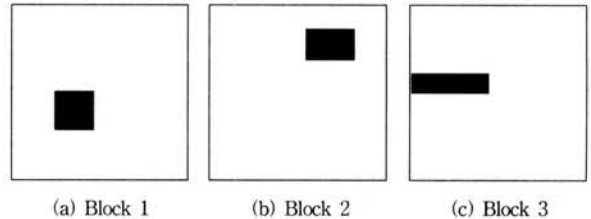
4.1 실험 환경

입력 도메인 확장 정책을 이용한 기법의 성능 평가를 위한 실험은 <표 2>와 같은 입력 파라미터를 이용하여 다양

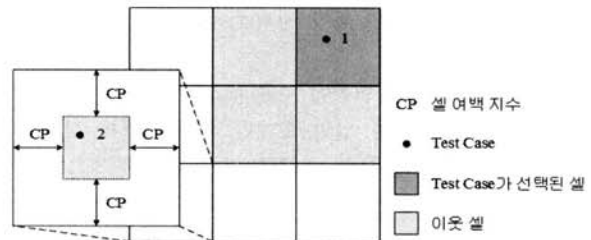
<표 3> 실험 환경

입력 도메인	2차원 : 10000 × 10000 3차원 : 5000 × 5000 × 5000
오류 비율(θ)	0.01, 0.005, 0.002, 0.001
오류 패턴	Block1, Block2, Block3
테스트 회수	5000

기법	확장 지수 및 셀 여백 지수
ED-IP-ART	$EF = 3$
ED-M-IP-ART, Mask1	$EF = 1$
ED-M-IP-ART, Mask2	$EF = 2$
ED-M-IP-ART, Mask2 with CP	$EF = 2, CP = 0.4$



(그림 6) 2차원 입력 도메인에서의 오류 패턴



(그림 7) 셀 여백 지수

한 오류 비율(θ) 및 Block 패턴을 가진 가상 입력 도메인을 구축하여 진행되었다. 오류 비율은 실제 입력 도메인에서 오류 영역이 차지하는 비율이며, 이에 따라 오류 영역의 크기는 '실제 입력 도메인의 크기 × 오류 비율'이 된다.

(그림 6)은 시각적인 특성을 고려해 임의로 설정한 오류 비율인 $\theta = 0.05$ 를 가정했을 때 생성된 오류 패턴의 예이다. 오류 영역은 하나의 클러스터를 이루고 있으며, 오류 비율에 맞춰 입력 도메인 내부의 임의 위치에 생성된다. 실험을 위해 사용된 Block 패턴의 생성은 오류 영역을 구성하는 변의 비율이 서로 다른 세 가지 형태를 갖도록 하였다.

또한, 그리드 방식이 Block 패턴에 대해 가지는 장점을 이용한 셀 여백 지수(CP)[1]를 입력 도메인 확장 정책에 추가로 적용한 실험이 일부 수행되었으며, 셀 여백 지수의 적용은 (그림 7)과 같은 형태로 이루어진다. 셀 여백 지수는 임의로 선택된 셀 내에서 테스트 케이스 선택 시, 셀 여백 지수를 통해 일정한 비율로 배제된 영역에서는 테스트 케이

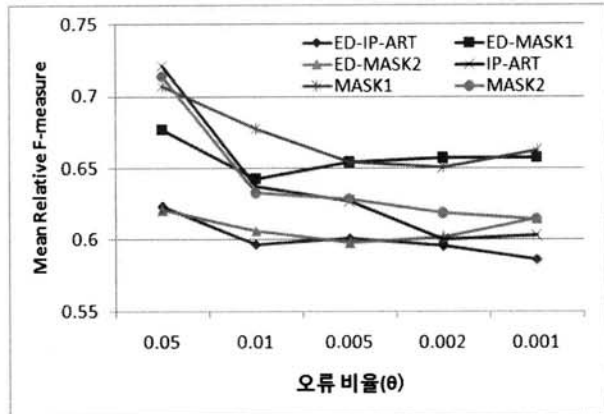
스가 선택되지 않도록 하는데 이용이 되며, 한 셀을 구성하는 변의 길이가 w 일 때, $0 < CP < 0.5w$ 값을 가진다. 셀 여백 지수가 적용되면, 셀 여백 지수의 값이 클수록 테스트 케이스는 셀의 중심부에서 선택되게 되어, Block 1 패턴에 대해 장점을 가지게 되며, 값이 낮아질수록 다른 기법과 유사한 성능을 보이게 된다. 셀 여백 지수의 성능 평가를 위한 실험에서는 Block 패턴에 대해 가장 높은 성능을 유도하는 것으로 알려진 $CP = 0.4$ 를 사용한다[1].

4.2 실험 결과

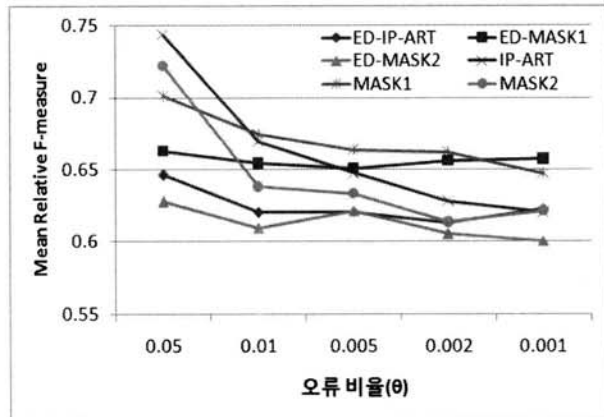
(그림 8)은 2차원 입력 도메인에서 오류 비율 변화에 따른 각 ART 기법의 성능 변화를 Block 패턴 1, 2 및 3에 대해서 각각 나타내고 있다. 이 실험 결과에 따르면 오류 비율이 높은 환경에서 낮은 성능을 보이는 기존의 현상이 입력 도메인 확장 정책을 사용하는 경우에도 유사하게 재현되고 있음을 알 수 있으며, 입력 도메인 확장 정책을 사용하는 경우, 그렇지 않은 경우에 비해 전체적으로 평균 3% 정도의 성능 향상을 보이는 것으로 나타났다.

오류 영역의 형태에 따른 성능은 Block 1 패턴의 경우 확장 입력 도메인 기반의 IP-ART 기법(ED-IP-ART)이 가장 우수한 성능을 보이며, Block 2와 Block 3 패턴에서는 Mask2를 이용한 확장 입력 도메인 기반의 M-IP-ART 기법(ED-MASK2)이 가장 우수한 성능을 보이는 것으로 나타났다. 이는 IP-ART 기법이 사용하는 제한 영역의 형태로 인한 Block 2 및 Block 3 패턴에서의 성능 저하[1]가 확장 입력 도메인 방식에서도 그대로 나타나고 있음을 의미한다. 하지만 입력 도메인 확장을 통해 상대적으로 고른 테스트 케이스 분포를 가질 수 있게 됨에 따라, 2차원 입력 도메인에서는 Block 3 패턴을 제외하면 입력 도메인 확장 정책을 이용하는 기법별 성능 차이가 최대 5% 정도로 그리 크지 않음을 알 수 있다. 특히 제한 영역으로 Mask1을 사용하는 M-IP-ART 경우에는 입력 도메인 확장 정책을 사용했을 때(ED-Mask1)와 그렇지 않은 때(Mask1) 모두에 대해서 오류 비율에 따른 성능 편차가 가장 작은 것으로 나타났으며, 이는 Mask1을 통해 생성되는 테스트 케이스 분포가 다른 기법에 비해 상대적으로 균등한 분포를 유발하기 때문에 오류 비율 변화에 대한 영향을 다른 기법에 비해 적게 받고 있음을 의미한다.

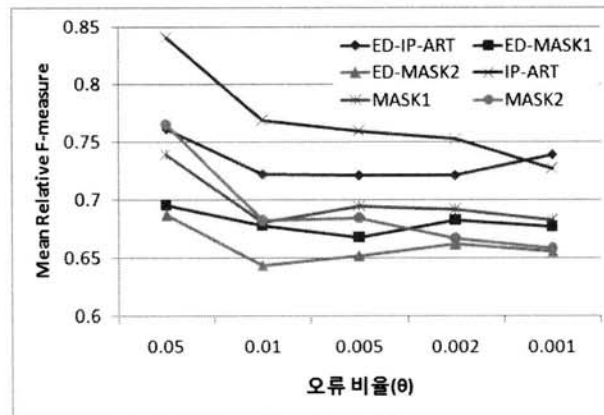
(그림 9)는 3차원 입력 도메인에서 오류 비율 변화에 따른 각 ART 기법의 성능 변화를 Block 패턴 1, 2 및 3에 대해서 각각 나타내고 있다. 이 실험 결과에 따르면 3차원 입력 도메인을 사용하는 경우, 입력 도메인 확장 정책을 이용할 때, 그렇지 않을 때에 비해 평균 10% 정도의 성능 향상을 보이는 것으로 나타났다. 즉, 테스트 케이스 분포 불균일성이 각 기법의 성능에 미치는 영향은 입력 도메인의 차원 수가 증가할수록 더 커지는 것을 고려할 때, 입력 도메인 확장 정책은 기존 기법에 비해 상대적으로 균일한 테스트 케이스 분포를 유발하고, 이를 통한 성능 향상이 있었음을 의미한다. 특히 Block 3 패턴이 $\theta = 0.05$ 로 분포된 환경에서는 IP-ART 기법과 Mask 2의 평균 상대 F-measure가 1



(a) Block 1



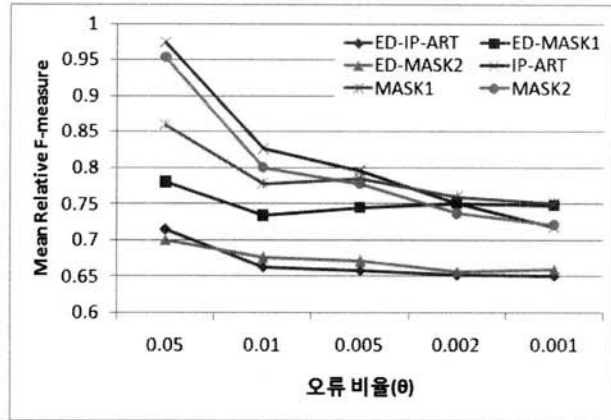
(b) Block 2



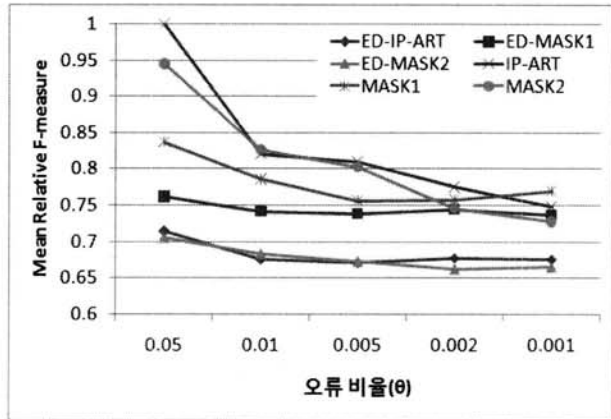
(c) Block 3

(그림 8) 오류 비율에 따른 성능 변화 - 2차원 입력 도메인

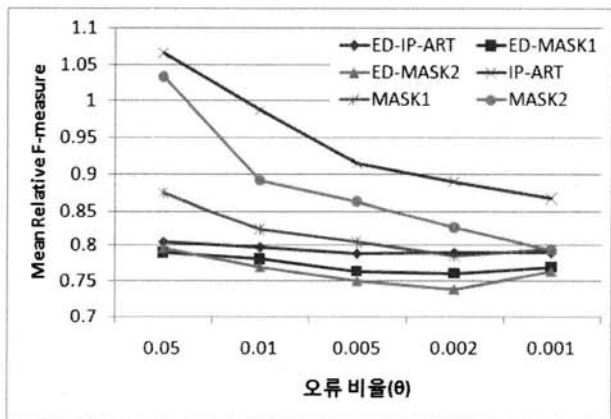
이상이 되어 순수 랜덤 테스트보다 낮은 성능을 보이지만, 입력 도메인 확장 정책을 통해 1 이하의 값을 갖도록 성능이 개선되는 등, 오류 비율이 상대적으로 높은 환경 즉, 기존 기법들이 낮은 성능을 보였던 환경에서 입력 도메인 확장 정책의 영향이 더욱 크게 나타났다. 또한 입력 도메인 확장 정책이 사용되지 않은 경우, 상대적으로 고른 테스트 케이스 분포를 보였던 M-IP-ART Mask1을 사용했을 때의 성능 평균은 동일한 환경에서의 IP-ART 기법이나 M-IP-ART Mask2보다 3~7% 정도 우수한 것으로 나타났으며, 이러한 결과



(a) Block 1



(b) Block 2

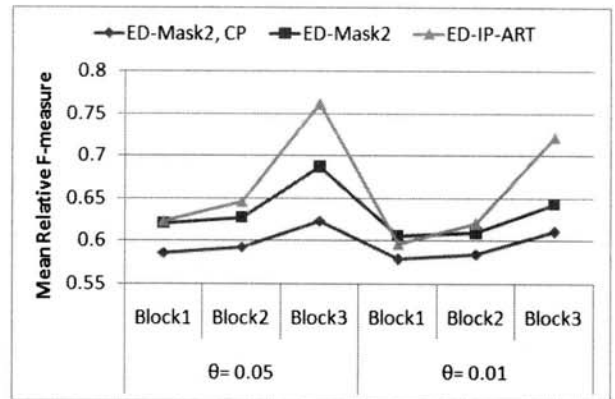


(c) Block 3

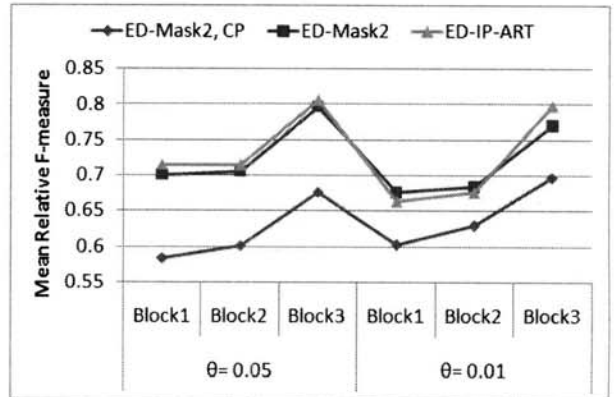
(그림 9) 오류 비율에 따른 성능 변화 - 3차원 입력 도메인

역시 테스트 케이스 분포의 불균일성이 입력 도메인 차원수 증가 시 기법의 성능에 영향을 미치고 있음을 의미한다.

각 기법별 성능 평균을 살펴보면, 기법별 성능 차이는 1~5% 수준으로 거의 비슷하지만, 그 중 ED-Mask2가 가장 우수한 성능을 보인다. 오류 비율 변화에 따른 성능 편차는 2차원 입력 도메인에서와 마찬가지로 Mask1을 사용하는 경우가 가장 작았고, 특히 ED-Mask1의 경우 Mask1이 보이는 성능 편차의 1/3 수준의 성능 편차를 보여 오류 비율에 의한 영향이 가장 작음을 알 수 있다. Block 패턴에 따른



(a) 2차원 입력 도메인

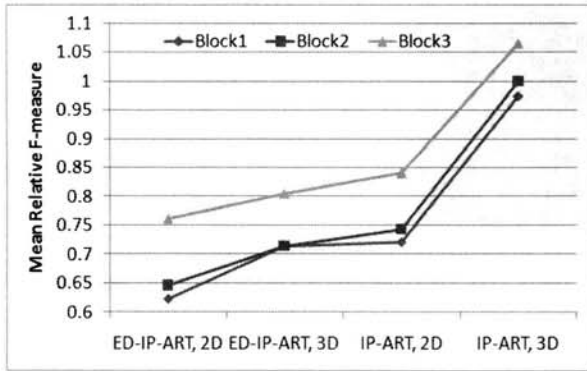


(b) 3차원 입력 도메인

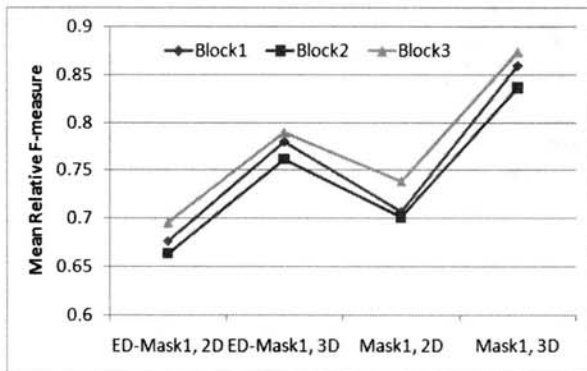
(그림 10) 셀 여백 지수 적용 시 성능 변화(CP = 0.4)

성능은 2차원 입력 도메인에서와 마찬가지로 Block 1 패턴의 경우 ED-IP-ART가 가장 우수하며, Block 2와 Block 3 패턴에 대해서는 ED-Mask2가 상대적으로 더 우수한 것으로 나타나 제한 영역 형태의 영향이 3차원 입력 도메인에서도 반영되고 있음을 보이고 있다.

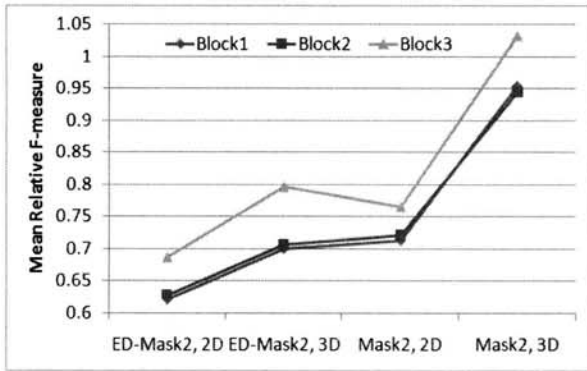
(그림 10)은 입력 도메인 확장 정책이 사용된 경우, 셀 여백 지수의 적용이 기법의 성능에 미치는 영향 평가를 위한 실험 결과를 제시하고 있다. 이 실험은 이전 실험에서 다른 오류 비율에 비해 기법들의 성능이 상대적으로 낮게 나타났던 $\theta = 0.05$ 와 $\theta = 0.01$ 일 때, 타 기법보다 더 우수한 성능을 보였던 ED-Mask2와 ED-IP-ART 기법과 셀 여백 지수가 적용된 ED-Mask2 (ED-Mask2, CP)의 성능 비교를 입력 도메인이 2차원일 때와 3차원일 때에 대해 각각 수행하였다. 실험 결과에 따르면 셀 여백 지수가 적용된 경우, 2차원 입력 도메인에서 평균 4~7%, 3차원 입력 도메인에서 평균 7~10% 정도 성능 향상이 이루어진 것으로 나타났으며, 오류 비율과 Block 패턴 변화를 포함한 모든 실험 환경에서 셀 여백 지수가 적용된 경우의 성능이 더 우수한 것으로 확인되었다. 특히 셀 여백 지수가 적용된 ED-Mask2 (ED-Mask2, CP)는 실험 환경 변화에 따른 성능 편차도 다른 기법에 비해 낮은 수준으로 유지되어 환경 변화에 대한 적응력이 뛰어난 것을 알 수 있다. 이와 같은 현상은 셀 여백 지수가 가지는 Block 패턴에 대한 특성이 다양한 환경에서 기법의 성능에 긍정적인



(a) IP-ART



(b) M-IP-ART, Mask1



(c) M-IP-ART, Mask2

(그림 11) 입력 도메인 차원수에 따른 성능 변화

영향을 미치고 있기 때문이다.

(그림 11)은 입력 도메인 차원수 증가에 따른 주요 기법별 성능 변화를 나타낸 것이다. 실험 결과에 따르면 기법별로 차이는 있으나, 모든 기법에서 입력 도메인 확장 정책 이용 시 입력 도메인 차원수 증가에 따른 성능 저하가 감소되는 것을 확인할 수 있으며, 특히 IP-ART 기법에 대한 입력 도메인 확장 정책 적용이 가장 큰 효과를 보이는 것을 알 수 있다. 실험 환경에 따른 성능 편차가 가장 작은 Mask1의 경우, 입력 도메인의 확장으로 인한 성능 향상 효과는 있지만, 입력 도메인 차원수 증가에 따른 성능 감소를 또한 입력 도메인 확장 정책의 사용 전과 유사하게 나타나, 입력 도메인의 차원수가 추가로 증가하는 경우에는 좋은 성능을 기대하기 어렵다. 따라서 입력 도메인 확장 정책 적용

시 입력 도메인 차원수 증가로 인한 성능 저하율이 가장 낮은 ED-IP-ART가 비교하는 세 기법 중 입력 도메인 차원수 증가에 따른 확장성이 가장 우수할 것으로 예측된다.

5. 결 론

IP-ART와 M-IP-ART 기법은 D-ART나 RRT 기법의 단점인 테스트 케이스 선택에 소요되는 많은 연산량을 그리드 방식을 이용하여 효율적으로 개선하고, 다양한 환경에서 대체로 기존 기법에 비해 우수한 성능을 보인다. 특히 M-IP-ART 기법의 경우 IP-ART 기법에서 사용하는 제한 영역 형태 변경을 통해 2차원 입력 도메인에서 IP-ART 기법보다 우수한 성능을 보인다. 하지만 IP-ART와 M-IP-ART 두 기법 모두 테스트 케이스 분포의 균일성 향상에 한계가 있어, 입력 도메인 차원수 증가에 따른 성능 저하가 문제로 지적되어 왔다. 따라서 본 논문에서는 각 기법을 사용했을 때 고른 테스트 케이스 분포를 보이는 것으로 나타난 영역에 실제 테스트 케이스를 분포시켜, 입력 도메인의 차원수 증가에 따른 성능 저하를 줄이도록 하는 입력 도메인 확장 정책을 제안하고, 그 성능을 평가하였다.

실험 결과에 따르면 입력 도메인 확장 정책이 각 기법에 사용되는 경우, Block 패턴의 형태에 따른 영향 감소 및 입력 도메인의 차원수 증가에 따른 적응성 증가로, 2차원 입력 도메인에서 3%, 3차원 입력 도메인에서 10% 수준의 향상된 성능을 보이는 것으로 나타나, 입력 도메인 확장 정책이 입력 도메인 차원수 증가에 따른 각 기법의 성능 저하 감소에 긍정적인 영향을 미치고 있음을 확인할 수 있었다.

하지만 입력 도메인 확장 정책은 기존의 기법에서 보이는 테스트 케이스 분포에 의한 영향을 완전히 배제할 수 없다는 특징을 가지고 있기 때문에, 기초가 되는 기법에서 더욱 고른 테스트 케이스 분포를 발생시키도록 하는 추가 연구가 필요하며, 진행된 실험 또한 3차원 입력 도메인 수준까지만 수행되어, 입력 도메인의 차원수가 추가로 확장되는 경우의 성능 변화를 살펴볼 필요가 있다.

참 고 문 헌

- [1] 이광규, "반복 분할 기반의 적응적 랜덤 테스트 향상 기법", 석사학위논문, 아주대학교 정보통신전문대학원, 2008
- [2] 정인상, "Alloy 명세 기반 자동 테스트 데이터 생성 기법", 정보처리학회논문지D, pp.1598-2866, 제14-D권 2호, pp.191-202, 2007.
- [3] 정인상, 창병모, "유전자 알고리즘을 이용한 뮤테이션 테스트의 테스트 데이터 자동 생성", 정보처리학회논문지D, pp.1598-2866, 제8-D권 1호, pp.81-86, 2001
- [4] K. P. Chan, T. Y. Chen, D. Towey, "Restricted Random Testing," In Proceedings of the 7th European Conference on Software Quality(ECSQ 2002), Volume 2349 of Lecture Notes in Computer Science, pages 321-330, Springer, 2002.

- [5] F. T. Chan, T. Y. Chen, I. K. Mak, and Y. T. Yu, "Proportional Sampling Strategy: Guidelines for Software Testing Practitioners," *Information and Software Technology*, 38:775-782, 1996
- [6] T. Y. Chen, G. Eddy, R. Merkel, P. K. Wong, "Adaptive Random Testing Through Dynamic Partitioning," In *Proceedings of the 4th International Conference on Quality Software (QSIC 2004)*, pages 79-86. IEEE Computer Society, 2004.
- [7] T. Y. Chen, D. H. Huang, Z. Q. Zhou, "Adaptive Random Testing Through Iterative Partitioning," In *Proceedings of the 11th International Conference on Reliable Software Technologies*, volume 4006 of *Lecture Notes in Computer Science*, pages 155-166. Springer-Verlag, 2006.
- [8] T. Y. Chen, H. Leung, and I. K. Mak, "Adaptive Random Testing", In *Proceedings of the 9th Asian Computing Science Conference (ASIAN 2004)*, volume 3321 of *Lecture Notes in Computer Science*, pages 320-329. Springer, 2004.
- [9] J. Mayer, "Adaptive Random Testing with Randomly Translated Failure Region," In *Proceedings of the 1st International Workshop on Random Testing (RT 2006)*, pages 70-77, ACM, 2006
- [10] J. Mayer, C. Schneckenburger,, "Adaptive Random Testing with Enlarged Input Domain", In *Proceedings of the Sixth International Conference on Quality Software (QSIC '06)*, pages 251-258. IEEE Computer Society, 2006.
- [11] J. Mayer, C. Schneckenburger, "An Empirical Analysis and Comparison of Random Testing Techniques," In *Proceedings of the 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE 2006)*, pages 105-114, ACM, 2006.



신 승 훈

e-mail : sihnsh@ajou.ac.kr

2000년 아주대학교 정보 및 컴퓨터공학부 (학사)

2002년 아주대학교 정보통신공학과(석사)

2002년~현 재 아주대학교 정보통신공학과 박사과정

관심분야: 소프트웨어 테스트 자동화, 멀티미디어 서비스 정책 등



박 승 규

e-mail : sparky@ajou.ac.kr

1974년 서울대학교 응용수학과(공학사)

1976년 한국과학기술원(KAIST) 전산학과(석사)

1982년 Institut National Polytechnique de Grenoble 전산학과(박사)

1976년~1977 한국과학기술연구소(KIST) 연구원

1977년~1978 KIET (현ETRI) 연구원

1978년~1982 프랑스 그레노블 IMAG 연구원/학생

1982년~1984년 KIET (현ETRI) 실장 / 선임연구원

1984년~1985년 미국 IBM 왓슨연구소 연구원

1985년~1992년 ETRI 연구위원 / 책임연구원.

1992년~현 재 아주대학교 정보통신대학 교수.

관심분야: 임베디드 테스트, 자가 컴퓨팅/치료 시스템, 차세대 컴퓨터 구조 등