

논문 2008-45CI-5-23

반복 분할 기반의 적응적 랜덤 테스트 향상 기법

(Modified Adaptive Random Testing through Iterative Partitioning)

이 광 규*, 신 승 훈**, 박 승 규***

(Kwang-Kyu Lee, Seung-Hun Shin, and Seung-Kyu Park)

요 약

적응적 랜덤 테스트 (Adaptive Random Testing, ART)은 입력 도메인 내의 오류 패턴을 순수 랜덤 테스트 (Random Testing, RT)보다 좋은 효율로 찾아내기 위해 고안된 테스트 케이스 선택 알고리즘이다. 대표적인 ART 기법인 거리 기반 ART (Distance-based ART, D-ART)와 제한 영역 기반 ART (Restricted Random Testing, RRT) 등은 좋은 성능을 보이지만, 테스트 케이스 선택에 필요한 많은 양의 거리 계산으로 인한 느린 테스트 케이스 생성과 거리 기반 방식의 사용으로 인한 테스트 케이스 분포의 불균일성이라는 단점을 가진다. 반복 분할 기반 ART (ART through Iterative Partitioning, IP-ART)는 입력 도메인을 반복 분할하는 방식을 통해 D-ART와 RRT가 가진 계산 부하를 크게 감소시켰다. 하지만 IP-ART의 경우에도 테스트 케이스 분포 문제는 여전히 존재하여 기법의 확장 적용에 대한 장애 요소로 작용하고 있다. 따라서 본 논문에서는 이와 같은 IP-ART의 단점 완화 및 성능 개선을 위한 방법을 제안하고, 실험을 통해 평균 9% 정도의 성능 향상을 확인하였다.

Abstract

An Adaptive Random Testing (ART) is one of test case generation algorithms that are designed to detect common failure patterns within input domain. The ART algorithm shows better performance than that of pure Random Testing (RT). Distance-based ART (D-ART) and Restriction Random Testing (RRT) are well known examples of ART algorithms which are reported to have good performances. But significant drawbacks are observed as quadratic runtime and non-uniform distribution of test case. They are mainly caused by a huge amount of distance computations to generate test case which are distance based method. ART through Iterative Partitioning (IP-ART) significantly reduces the amount of computation of D-ART and RRT with iterative partitioning of input domain. However, non-uniform distribution of test case still exists, which play a role of obstacle to develop a scalable algorithm. In this paper we propose a new ART method which mitigates the drawback of IP-ART while achieving improved fault-detection capability. Simulation results show that the proposed one has about 9 percent of improved F-measures with respect to other algorithms.

Keywords : Random Testing, Adaptive Random Testing, Test Case Generation

I. 서 론

소프트웨어의 품질은 소프트웨어 개발 프로세스의 질에 의존적이며, 소프트웨어 테스트는 소프트웨어 개발 프로세스에서 매우 중요한 역할을 담당한다^[1-2]. 따라서 높은 품질을 가지는 소프트웨어의 개발을 위해서

는 효율적인 소프트웨어 테스트 정책이 요구되며, 이에 따라 다양한 형태의 소프트웨어 테스트 관련 연구가 진행되어 왔다. 다양한 소프트웨어 테스트 정책 중 하나인 랜덤 테스트는 기초적이고 단순한 형태의 접근을 시도한다^[3-4]. 랜덤 테스트에서는 입력 도메인 내에서 임의의 값을 선택하여 테스트 케이스를 생성하고, 이를 이용하여 주어진 소프트웨어에 대한 테스트를 수행한다. 이와 같은 랜덤 테스트는 소프트웨어 테스트 자동화를 위한 가장 기본적이고 중요한 방법의 하나이며^[2], 실제로 자바 JIT 컴파일러, SQL 데이터베이스군, 윈도 우즈 NT 응용 프로그램 등과 같은 어플리케이션에 성공적으로 적용되었다^[5-6].

* 정회원, (주) NHN QA센터
(QA center, NHN Co.)

** 정회원-교신저자, *** 평생회원, 아주대학교 정보통신
전문대학원

(Graduate School of Information & Communication,
Ajou University)

접수일자: 2008년5월22일, 수정완료일: 2008년9월8일

랜덤 테스트는 테스트 대상이 되는 소프트웨어의 소스코드 분석 없이 적용 가능한 블랙박스 테스트 기법의 하나이기 때문에 다른 소프트웨어 테스트 정책과 비교할 때, 상대적으로 소프트웨어 테스트 자동화가 용이하다는 장점을 갖지만, 소프트웨어 내의 오류를 찾아내는 성능은 다른 소프트웨어 테스트 정책에 비해 낮다는 단점이 있다. 이에 따라 소프트웨어 내의 오류 패턴^[2]에 근거하여 순수 랜덤 테스트 정책보다 더 효율적으로 테스트 케이스를 생성해 내기 위한 다양한 적응적 랜덤 테스트 (Adaptive Random Testing, ART) 기법들이 제안되어 왔다^[2, 6~10].

기존에 제안된 ART 기법들은 순수 랜덤 테스트와 비교할 때 30%에서 경우에 따라 최대 50%까지의 향상된 성능을 보이지만^[2], 일부 ART 기법은 테스트 케이스 선택에 매우 많은 연산량을 요구하여 테스트 케이스 선택에 오랜 시간이 소요되는 단점을 가지고 있으며^[7~8], 또한 테스트 케이스의 입력 도메인 내 분포가 균일하지 않아 성능의 추가 향상 및 적용 대상 확장을 어렵게 하는 현상이 기존 ART 기법 모두에서 나타나고 있다^[10]. 이러한 기존의 ART 기법이 가지는 단점을 보완하기 위해 미러링(Mirroring), 지역화(Localization) 및 분할(Partitioning) 등을 이용한 다양한 방법들이 제안되어 왔으며^[2, 9, 12~14], 이 중 일부는 D-ART와 RRT가 가지는 많은 연산량 문제는 해결했으나, 오류 발견 능력은 D-ART나 RRT보다 낮아지는 특성을 보인다^[9]. IP-ART (ART through Iterative Partitioning)는 빠른 테스트 케이스 선택 속도와 높은 오류 발견 성능 두 가지를 모두 만족시키고 있지만^[2], 테스트 케이스 분포 문제는 여전히 존재해 입력 도메인 내에 위치하는 오류 패턴의 형태와 위치에 따른 성능 편차를 유발하는 것으로 나타났다.

본 논문에서는 그리드 방식이 가지는 장점을 효과적으로 살릴 수 있도록 하는 셀 여백 지수의 사용을 통한 IP-ART 성능 개선 방법을 제안하고, 기존 IP-ART에서 사용한 제한 영역에 다양한 변화를 주어, 제한 영역의 형태가 테스트 케이스 분포와 기법의 성능에 미치는 영향을 실험을 통해 평가한다.

II. 정의 및 관련 연구

1. 정의

하나의 소프트웨어에 대한 입력 데이터의 집합인 입력 도메인은 유한 공간으로 가정하며, 입력 도메인 내

에 존재하는 전체 입력 데이터 중 소프트웨어의 오류를 유발하는 입력 데이터의 비율을 오류 비율(θ)이라고 정의한다. 따라서 입력 도메인의 크기가 d , 오류 유발 입력 데이터의 크기가 m 일 때, 오류 비율, $\theta = m/d$ 이 된다.

F-measure는^[7] 첫 번째 오류가 발견될 때까지 소요된 테스트 케이스의 개수를 의미한다. 통상적으로 소프트웨어에 대한 테스트를 수행할 때, 오류가 발견되면 진행되던 테스트를 멈추고 해당 오류에 대한 대처 작업을 수행하는 것이 일반적이기 때문에, F-measure는 소프트웨어 테스트 기법의 성능 평가 시 직관적인 지표로 이용될 수 있을 뿐 아니라, 기존 ART 기법의 성능 평가에도 F-measure가 이용되므로 기법 간 상호 성능 비교에도 용이하게 사용 가능하다.

순수 랜덤 테스트의 F-measure 이론 평균치는 $F_{RT} = 1/\theta$ 이 되고, 각 기법의 성능은 순수 랜덤 테스트의 성능을 기준으로 하는 평균 상대 F-measure를 사용하여 평가하며, 평균 상대 F-measure는 다음과 같이 정의된다.

$$\text{평균 상대 F-measure} = \text{mean } F_{ART} / F_{RT} \quad (1)$$

즉, 평균 상대 F-measure가 1인 경우 순수 랜덤 테스트의 성능과 같다는 것을 의미하고, 0에 가까운 값일 수록 우수한 성능을, 1보다 큰 값인 경우 순수 랜덤 테스트 기법보다 낮은 성능을 가짐을 의미한다.

2. 오류 패턴 (Failure Pattern)

소프트웨어의 오류를 유발하는 입력은 입력 도메인 상에서 몇 가지 형태로 무리를 이루어 오류 영역을 구성하는 특징(이하 오류 패턴)을 보이며, 이 오류 패턴에는 블록 패턴, 스트립 패턴, 포인트 패턴 등이 있다^[11]. 그림 1은 오류 패턴의 종류를 2차원 입력 도메인 상에서 표현한 것이다.

포인트 패턴은 오류를 유발하는 입력이 한 점 혹은 아주 작은 크기의 군집을 이루어 입력 도메인의 여러 곳에 분포되어 있는 패턴이고, 스트립 패턴은 오류를 유발하는 입력의 영역이 폭이 좁은 직선 형태를 가지는 패턴을 의미하며, 블록 패턴은 오류를 유발하는 입력이 하나 혹은 소수의 영역에 군집되어 있는 형태를 가진다. Chan 등에 따르면 블록 및 스트립 패턴이 상대적으로 더 빈번하게 나타나는 오류 패턴이고^[11], 따라서 기존의 ART 관련 연구에서는 제안 기법의 성능 평가 시

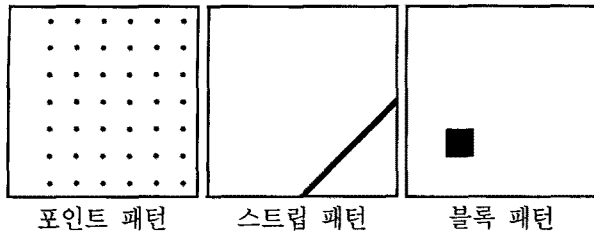


그림 1. 오류 패턴
Fig. 1. Failure Patterns.

주로 블록 패턴을 사용했다.

ART 기법에서는 이와 같은 소프트웨어가 가지는 오류 패턴에 근거하여, 오류를 유발하지 않는 테스트 케이스와 근접한 입력 데이터는 오류를 유발할 확률이 상대적으로 낮다고 보고, 테스트 케이스가 입력 도메인 내에 가능한 넓게 분포될 수 있도록 테스트 케이스를 선택한다. 그림 1^[7]과 같은 형태의 오류 패턴을 유발하는 프로그램 예제는 [9]에서 찾아볼 수 있다.

3. 기존 ART 기법

테스트 케이스의 생성에 이전 테스트 케이스와의 거리를 이용하는 거리 기반 ART(Distance-based ART, D-ART)^[7]와 이전 테스트 케이스를 중심으로 제한 영역을 설정하고 그 외부에서 새로운 테스트 케이스가 선택되도록 하는 제한 영역 기반 ART(Restricted ART, RRT)^[8]는 ART의 가장 기초적이며 대표적인 기법이다. 하지만 D-ART와 RRT의 테스트 케이스 선택 정책은 테스트가 진행될수록 이전에 사용했던 테스트 케이스의 개수가 증가하게 되고, 이에 따라 새로운 테스트 케이스 선택에 수반되는 거리 계산 및 제한 영역 설정에 긴 연산 시간을 요구하는 단점을 가진다. 이러한 단점을 극복하기 위해 입력 도메인을 분할하여 그리드를 만들고, 작성된 그리드 중 하나의 셀과 선택된 셀 내에 존재하는 입력 데이터를 임의로 선택하여 테스트 케이스를 생성하는 입력 도메인 분할 기법들이 제안되었는데, 이 중 입력 도메인 반복 분할 기법 (ART through Iterative Partitioning, IP-ART)은 D-ART나 RRT보다 아주 낮은 연산량을 이용하지만 이들과 유사한 성능을 보인다^[2]. 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

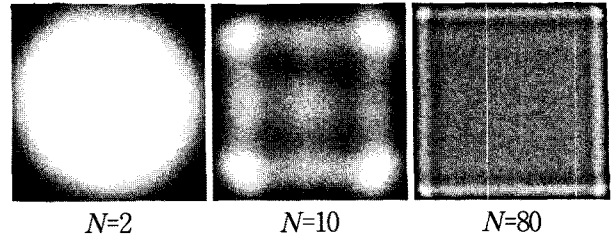


그림 2. N 번째 테스트 케이스의 분포 - DART^[10]
Fig. 2. The distribution of N -th test case - DART^[10].

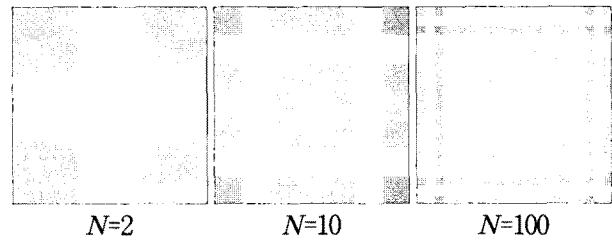


그림 3. N 번째 테스트 케이스의 분포 - IP-ART
Fig. 3. The distribution of N -th test case - IP-ART.

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 2^[10]와 그림 3은 각각 DART와 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.

D-ART나 RRT 기법 사용 시 나타나는 입력 도메인 내 테스트 케이스 분포 불균일성을 해결하기 위해 Mayer 등은 오류 영역의 가상 이동^[6]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 거리를 기반으로 하는 D-ART 및 RRT 기법의 사용 시 나타나는 테스트 케이스 편중 현상을 대상으로 하기 때문에 이와는 다른 형태의 테스트 케이스 편중 현상을 보이는 기법에 적용하는 것이 불가능하며, 개선된 기법^[6,10] 역시 D-ART와 RRT 기법을 바탕으로 하기 때문에 테스트 케이스 생성에 긴 시간을 요구하는 단점을 가진다. 따라서 입력 도메인 분할 방식을 사용하는 기법을 기반으로 하는 새로운 기법이 요구된다.

III. Modified-IP-ART (M-IP-ART)

IP-ART를 사용하는 경우에 발생하는 테스트 케이스 분포의 불균일 현상의 원인은 선택된 셀과 선택된 셀의 8개 이웃 셀을 제한 영역으로 지정하여 해당 영역에서 테스트 케이스를 선택하지 못하도록 하는데 있다. 또한 IP-ART는 사각형 형태로 사상 가능한 입력 도메인 상에 존재하는 정사각형 형태의 오류 패턴에 특화된 구조를 사용하고 있기 때문에, 오류 패턴이 비록 블록 패턴이라고 할지라도 패턴의 모양이 정사각형에서 벗어날수록 성능이 저하되는 특징을 가진다. 이와 같은 IP-ART가 가지는 단점의 보완을 위해 본 논문에서는 새로운 제한 영역 마스크 사용과 셀 여백 지수 사용을 제안하고, 이를 통해 테스트 케이스 분포 불균일 현상의 완화 및 다양한 형태의 블록 패턴에 대한 알고리즘의 적응성을 높이고자 한다.

1. 제한 영역 마스크

그림 4의 (a)와 (b)는 각각 기존의 ART by Bisection^[9]과 IP-ART^[2]에서 사용한 제한 영역 마스크를, (c)와 (d)는 제한 영역 마스크가 테스트 케이스 분포 불균일성에 미치는 영향을 파악하기 위해 새롭게 정의한 제한 영역 마스크이다. ART by Bisection에서 사용되는 그림 4의 마스크 (a)의 경우, D-ART와 RRT가 가지는 테스트 케이스 선택에 필요한 많은 양의 거리 계산을 그리드를 이용해 효과적으로 제거할 수 있어 테스트 케이스 선택 속도의 개선은 이루어졌으나, 제한 영역이 테스트 케이스가 선택된 셀로 국한되어 그림 5와 같이 다음 테스트 케이스가 이전 테스트 케이스 주위에서 생성되는 단점을 가지고, 이로 인해 D-ART나 RRT 보다 낮은 성능을 보인다^[15].

그림 4의 (b)는 IP-ART에서 사용하는 제한 영역 마스크로 그림 5에서 제시한 것과 같은 인접한 테스트 케이스 선택 문제를 해결하기 위해, 테스트 케이스가 선택된 셀과 그 이웃 셀을 모두 제한 영역으로 정의하여 테스트 케이스 선택 시 충분한 거리가 유지될 수 있도록 하였다. 하지만 이 경우 그림 3에 제시된 것과 같은 테스트 케이스 분포의 불균일 현상이 발생되어 성능 향상에 대한 장애 요소로 작용한다. 이와 비교하여, (c) Mask 1은 대각선 방향의 셀을 (b)의 제한 영역에서 제외하여 더 좁아진 제한 영역을 가지고, (d) Mask 2는 4개의 셀을 (b)의 제한 영역에 추가하여 더 넓어진 마름모 형태의 제한 영역을 총 12개의 셀을 이용해 지정하

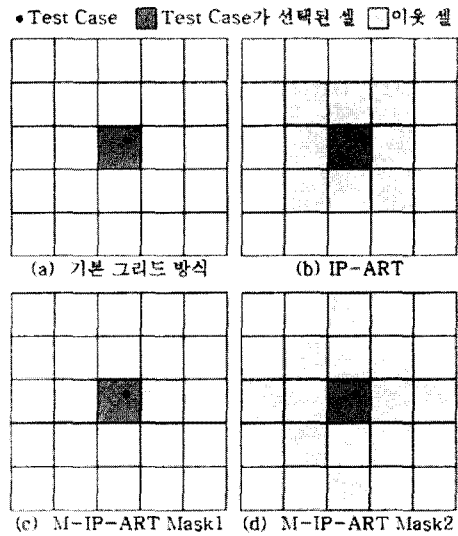


그림 4. 제한 영역 마스크
Fig. 4. Masks for generate restriction area.

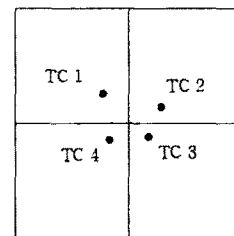


그림 5. ART by Bisection의 테스트 케이스 선택 예
Fig. 5. A case of test case selection of the ART by Bisection.

고 있다. 마스크 (c)를 이용하는 경우, 마스크 (a)를 이용했을 때의 단점인 인접한 테스트 케이스 선택이 테스트 케이스가 선택된 셀의 대각선 방향의 셀에서 발생되어 알고리즘의 성능에 영향을 미치겠지만, 테스트 케이스 분포는 마스크 (b)를 사용했을 때보다 균일하게 분포될 것이다. 그리고 마스크 (d)를 이용하는 경우에는 마스크의 모양이 마름모꼴을 형성하고 있어 마스크 (b)를 사용했을 때 볼 수 있는 '井' 형태의 테스트 케이스 선택에서 자주 이용되지 않는 영역에 영향을 주어 좀 더 균일한 분포를 얻을 수 있을 것이다.

2. 셀 여백 지수 (CP)

그림 4의 (a)와 같은 기본 그리드 방식을 사용하는 ART by Bisection을 사용할 경우, 그림 6-(a)와 같이 그리드를 구성하는 셀의 크기가 오류 영역의 크기와 완벽하게 일치하면, 상대적으로 적은 수의 테스트 케이스로 오류 영역을 발견할 확률이 높아진다. 그러나 그림 6-(b)와 같이 오류 영역이 셀의 영역과 일치하지 않는 경우에는 오류 영역을 벗어나는 테스트케이스를 선택할

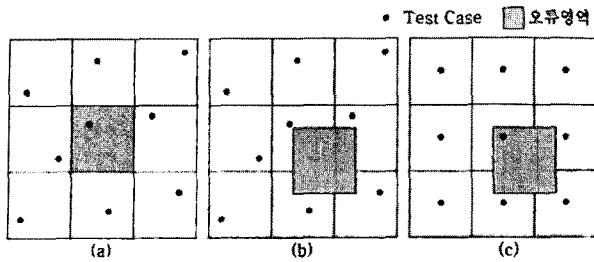


그림 6. 그리드 방식의 특징
Fig. 6. A typical characteristic of a grid.

확률이 상대적으로 높아지기 때문에 알고리즘의 효율성이 떨어지게 된다. 따라서 그림 6-(c)와 같이 셀 내부의 일정한 위치에서 테스트 케이스를 선택하게 되면, 오류 영역과 셀 영역이 일치하지 않더라도 그림 6-(a)의 경우와 유사한 효율성을 갖게 된다.

단, 이러한 특성은 오류 영역의 크기가 셀의 크기와 일치하는 경우에 한정되며, 일반적인 상황에서는 오류 영역의 크기가 셀의 크기와 일치하지 않기 때문에 오류 영역의 크기가 셀의 크기와 일치하는 경우에 비해 효율성이 떨어지게 된다. 오류 영역의 크기가 다양한 일반적인 상황을 전제로 한 실험에서 셀 내부의 특정 위치에서 테스트 케이스를 생성했을 때의 평균 상대 F-measure는 0.59로, 셀 내부의 임의의 위치에서 테스트 케이스를 생성했을 때의 평균 상대 F-measure는 0.71로 나타났다. 즉, 기본 그리드 방식 기반의 ART by Bisection을 사용하는 경우, 셀 내부의 특정 위치에서 테스트 케이스를 선택하는 방법이 그렇지 않은 방법보다 약 12% 적은 수의 테스트 케이스를 사용하여 오류를 검출할 수 있음을 나타낸다. 이에 따라 본 논문에서는 이와 같은 그리드의 특성을 기법에 반영하기 위한 셀 여백 지수(CP)를 정의하여 이용한다. 셀 여백 지수는 셀 너비에 대한 상대 비율로 표현되는데, 한 셀의 너비가 w 일 때, $0 < CP < 0.5w$ 값을 가지며, 셀의 모든 경계에 적용이 되어 셀의 네 가장자리 영역이 테스트 케이스 선택 시 배제되도록 한다. 이와 같은 방법은 테스트 케이스의 선택이 셀 중심부의 일정 공간 내에서 이루어지도록 하기 때문에, 셀 여백 지수가 커질수록 블록 패턴 검출 성능에 긍정적 영향을 줄 것이며, 셀 여백 지수가 작아질수록 기존의 기법과 유사한 성능을 보일 것이다.

3. 알고리즘

M-IP-ART는 IP-ART 알고리즘^[2]을 바탕으로 하여, 사용되는 마스크 양식의 변경 및 셀 여백 지수가 적

표 1. M-IP-ART 알고리즘
Table 1. The M-IP-ART algorithm.

<p>INPUT {셀 여백 지수 CP, 제한 영역 마스크 M}</p> <p>P : 입력 도메인 분할 수준</p> <p>E : 이전에 테스트 된 테스트 케이스의 집합, $E = \{e_1, e_2, \dots, e_N\}$</p> <p>$G$: $P \times P$ 불린 그리드 (Boolean Grid)</p> <ol style="list-style-type: none"> $P := 1, E := \emptyset$ 로 초기화. a) $P \times P$ 그리드 G를 생성. b) G의 모든 셀을 T로 초기화. c) $\forall e_i \in E, e_i$를 포함하는 모든 셀을 기준으로 M에 해당하는 셀을 F로 변경. T인 셀이 존재하면, a) T인 셀 중 임의의 셀을 선택. b) 선택된 셀에 셀 여백 지수(CP) 적용 c) 셀 여백을 제외한 영역에서 임의의 테스트 케이스 선택 d) 테스트 수행. - 오류 발견 시 테스트 중지 e) 테스트 케이스를 이전 테스트 케이스 집합(E)에 등록 f) 선택된 셀을 기준으로 M에 해당하는 셀을 F로 변경. g) 단계 3 반복 T인 셀이 더 이상 없으면, a) 분할 수준을 1 증가시키고, ($P := P + 1$) b) 단계 2부터 다시 반복. 오류가 발견될 때까지 단계 2~4를 반복 수행.
--

용되었으며, 표 1에 기술된 것과 같이 동작한다.

M-IP-ART는 입력 도메인을 $P \times P$ ($P > 0$)개의 셀로 분할하여 그리드를 구성하되 분할 수준(P)은 1부터 시작하여 1씩 증가하도록 한다. 테스트 케이스는 아직 테스트 케이스 선택이 이루어지지 않은 셀 중 임의로 선택된 셀에 셀 여백 지수(CP)를 적용하여 셀 외곽의 셀 여백을 배제하고, 셀 여백에 해당되지 않는 셀의 중심 영역에서 임의로 선택한다. 이러한 과정을 통해 선택된 테스트 케이스를 이용하여 테스트를 수행한 결과 오류가 발견되지 않으면, 이 테스트 케이스를 이전 테스트 케이스 집합(E)에 등록하고, 테스트 케이스의 선택이 이루어진 셀과 이 셀 주위의 제한 영역에 해당하는 셀을 F 로 설정하여 이들 셀에서 추가의 테스트 케이스 선택이 이루어지는 것을 방지한다. 이러한 과정을

반복적으로 수행하여 테스트 케이스를 선택하되, 그리드를 구성하는 셀 중 T 로 정의된, 즉 아직 테스트 케이스 선택이 이루어지지 않은 셀이 존재하지 않으면 분할 수준을 1 증가시키고 ($P := P+1$) 입력도메인을 재분할한다. 재분할 후에는 이전 테스트 케이스 집합에 포함된 테스트 케이스를 가지는 모든 셀과 이 셀 주위의 제한 영역에 해당하는 셀을 F 로 설정하여 이미 테스트 케이스 선택이 이루어진 것으로 정의하며, 새로운 테스트 케이스 선택은 테스트 케이스 선택이 이루어지지 않은 것으로 정의된 셀(T)에서 이루어지게 된다. 이는 이전 테스트 케이스 인근에서 새로운 테스트 케이스가 선택되는 것을 방지하기 위한 것이며, 이 과정은 오류를 발견할 때까지 반복 수행된다.

IV. 성능 평가

1. 실험 환경

새로 정의된 제한 영역 마스크가 테스트 케이스 분포에 미치는 영향과 셀 여백 지수가 알고리즘의 성능에 미치는 영향 및 각 ART 기법 간 성능 비교를 위한 실험은 표 2와 같은 입력 파라미터를 이용하여 다양한 오류 비율(Θ) 및 오류 패턴을 가진 가상 입력 도메인을 구축하여 진행되었다. 오류 비율은 전체 입력 도메인에서 오

표 2. 실험 환경
Table 2. Simulation Environment.

입력 도메인	10000 × 10000 (2차원)
오류 비율 (Θ)	0.01, 0.005, 0.002, 0.001
오류 패턴	Block1, Block2, Block3, Strip, Point
테스트 회수 (N)	5000

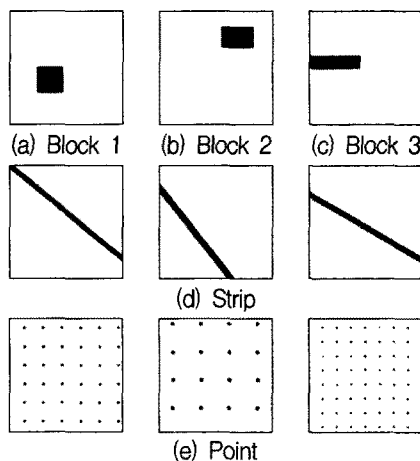


그림 7. 오류 패턴
Fig. 7. Generated failure patterns for simulation.

류 영역이 차지하는 비율을 의미하며, 이에 따라 오류 영역의 크기는 '전체 입력 도메인 크기 × 오류 비율'로 계산된다.

그림 7은 시각적인 특성을 고려해 가상으로 생성한 $\Theta = 0.5$ 일 때의 오류 패턴이다. 오류 패턴은 블록, 스트립 패턴의 경우 오류 영역이 하나의 클러스터를 이루고 있다고 가정하였으며, 오류 비율에 맞춰 입력 도메인 내부의 임의 위치에 생성된다.

블록 패턴은 폭과 높이를 달리하는 세 가지 유형으로 생성될 수 있도록 하였으며, 스트립 패턴의 경우, 기존 ART 기법에서는 입력 도메인을 구성하는 두 인접한 경계와 각 경계 상에 위치한 한 점을 임의로 선택한 후, 두 점을 지나는 선분을 구성하고, 주어진 오류 비율에 맞추어 선분의 두께를 조절하는 형태로 생성했는데, 이때 각 경계에서 임의로 선택된 점의 위치가 두 경계의 접점과 근거리에서 위치하여 선분의 길이가 짧아지면 선분의 두께가 너무 두꺼워지므로 적정거리를 유지하도록 구성하였다^[16]. 하지만 '인접한 두 경계' 혹은 '두 경계의 접점에서 근거리에서 위치하는 점의 생성 불가'와 같은 생성 조건은 실제 소프트웨어에서 충분히 발생 가능한 조건임에도 불구하고 이를 배제하도록 하고 있기 때문에 사실적으로 맞게 수정될 필요가 있다. 따라서 본 논문에서의 스트립 패턴은 스트립 패턴이 시작되는 시작점과 선분의 기울기를 임의로 생성하여 선분을 구성하고, 오류 비율에 따라 폭을 조절하는 형태로 생성하였다. 포인트 패턴은 오류 영역이 여러 개의 작은 클러스터로 구성되어 있다고 가정하기 때문에, 모든 클러스터의 합이 오류 영역의 크기가 되도록 하였고, 각 클러스터는 임의의 위치에 임의의 개수가 생성되도록 했다. 단, 오류 영역의 개수는 일정 범위 내로 제한하였다.

2. 제한 영역 마스크의 영향

새롭게 정의된 제한 영역 마스크를 사용한 경우의 테스트 케이스 분포는 그림 8, 9와 같은 형태로 나타났다. 그림 8과 9는 총 5000번의 테스트를 수행하고, 각 테스트 과정에서 생성된 N 번째 테스트 케이스의 입력 도메인 내 위치를 점으로 나타낸 것이다. 즉, 선택 빈도가 높은 영역은 상대적으로 어둡게 표현된다.

그림 8은 Mask 1을, 그림 9는 Mask 2를 적용했을 때의 결과이다. 이들 모두에서 IP-ART의 경우보다 편중 현상이 확연히 줄어들기는 했지만, 여전히 테스트 케이스 선택 빈도가 다른 영역에 비해 상대적으로 낮은

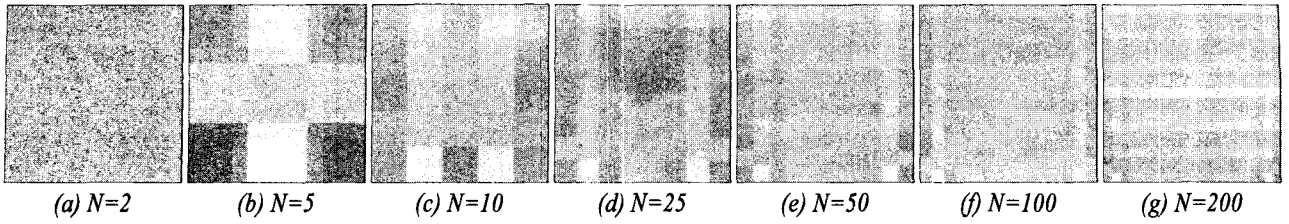


그림 8. Mask 1에 의한 테스트케이스 분포
Fig. 8. The test case distributions that generated with mask1.

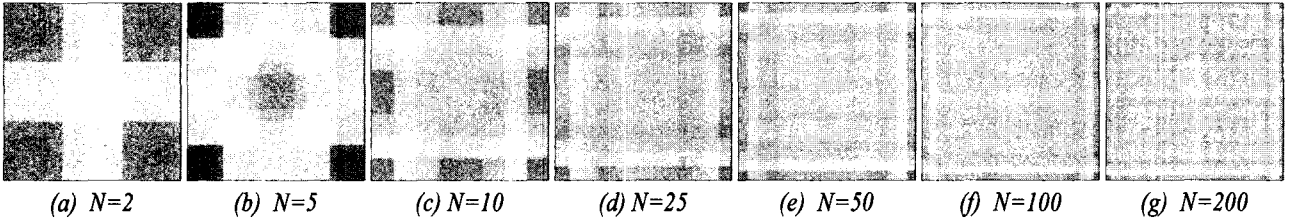


그림 9. Mask 2에 의한 테스트케이스 분포
Fig. 9. The test case distributions that generated with mask2.

영역이 존재하고 있음을 알 수 있다. Mask 2의 경우에는 테스트가 진행될수록 편중 현상이 상당히 완화되었으며, 실제로 IP-ART에서는 테스트 케이스 선택이 거의 이루어지지 않던 영역에서도 어느 정도 테스트 케이스 선택이 이루어지고 있는 것을 알 수 있다.

그림 10은 2차원 도메인 상에서 $\Theta = 0.005$ 일 때 각 오류 패턴에 대한 IP-ART와 M-IP-ART의 성능을 비교 제시하고 있다. 그림 10의 실험 결과를 살펴보면 평균 성능은 기본 그리드 < Mask 1 < IP-ART < Mask 2를 사용했을 때의 순으로 나타났다. 이러한 결과는 테스트 케이스 편중 현상이 기본 그리드 < Mask 1 < Mask 2 < IP-ART 순으로 심해지는 것을 고려할 때, 기본 그리드 방식과 Mask 1은 다른 마스크를 사용할 때보다 고른 테스트 케이스 분포를 생성시키지만, 오히려 성능은 저하되고 있음을 알 수 있다. 이 같은 결과는

기본 그리드 방식을 사용하는 ART by Bisection의 단점인 인접한 테스트 케이스 선택 가능성이 성능에 영향을 미친 것으로 파악된다. 즉, 제한 영역이 좁아지면 테스트 케이스의 분포는 상대적으로 균일하게 되지만, 인접한 테스트 케이스 선택으로 인한 성능 저하가 발생되며, 제한 영역이 일정 수준으로 커지면 어느 정도 성능 향상은 기대할 수 있으나, 테스트 케이스 분포가 불균일해져 오류 패턴과 오류 영역의 입력 도메인 내 위치 및 입력 도메인의 차원 수 증가에 따른 성능 변동 폭이 심화되는, 제한 영역의 넓이와 알고리즘의 성능 간 트레이드오프가 존재함을 알 수 있다.

이에 따라 제한 영역의 넓이와 알고리즘의 성능 간의 관계 확인을 위한 실험이 수행되었고, 그림 11에 그 결과가 제시되어 있다. 이 실험 결과에 따르면 제한 영역이 13개의 셀로 구성된 Mask 2가 가장 좋은 성능을 보

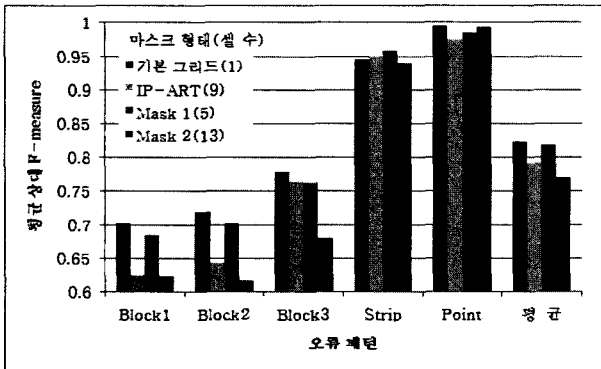


그림 10. IP-ART와 M-IP-ART의 성능 비교 ($\Theta = 0.005$)
Fig. 10. Mean relative F-measures of IP-ART and M-IP-ART ($\Theta = 0.005$).

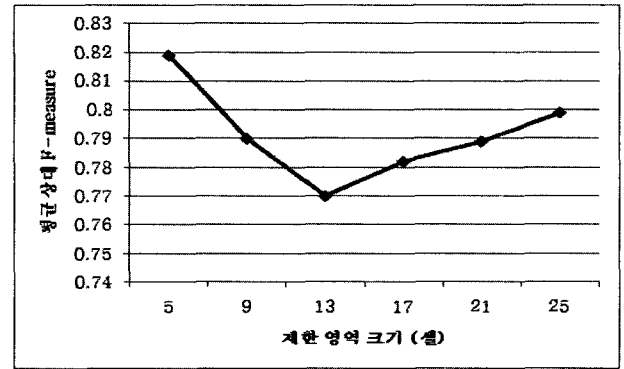


그림 11. 제한 영역의 크기에 따른 성능 변화
Fig. 11. Mean relative F-measure with various size of restriction area.

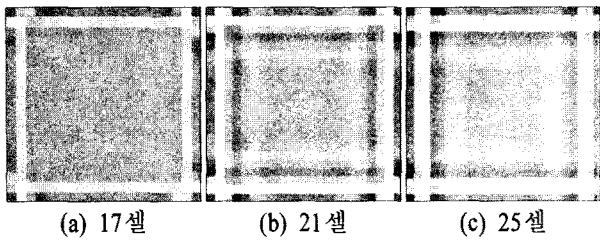


그림 12. 제한 영역 크기 변화에 따른 테스트 케이스 불균일 현상 (N=50)

Fig. 12. Non-uniform test case distributions that are generated by various size of restriction area (N=50).

이고 있으며, 그보다 제한 영역이 더 커지면 테스트 케이스 분포의 불균일 정도가 그림 12와 같이 심화되어, 오히려 성능이 저하된다. 즉, 제한 영역의 크기와 형태 및 그 제한 영역으로 인해 생성되는 테스트 케이스 분포가 종합적으로 알고리즘의 성능에 영향을 미치는 것을 알 수 있다.

3. 셀 여백 지수(CP)에 따른 성능 평가

그림 13과 14는 각각 Mask 1과 Mask 2를 적용했을 때, 셀 여백 지수 변화 ($0.2 \leq CP \leq 0.45$, 0.05 간격)에 따른 M-IP-ART의 성능 변화를 평가한 실험 결과이며, 각 환경에 대해 5,000회 실험한 평균 상대 F-measure를 제시하고 있다. 이 실험 결과에는 그리드 방

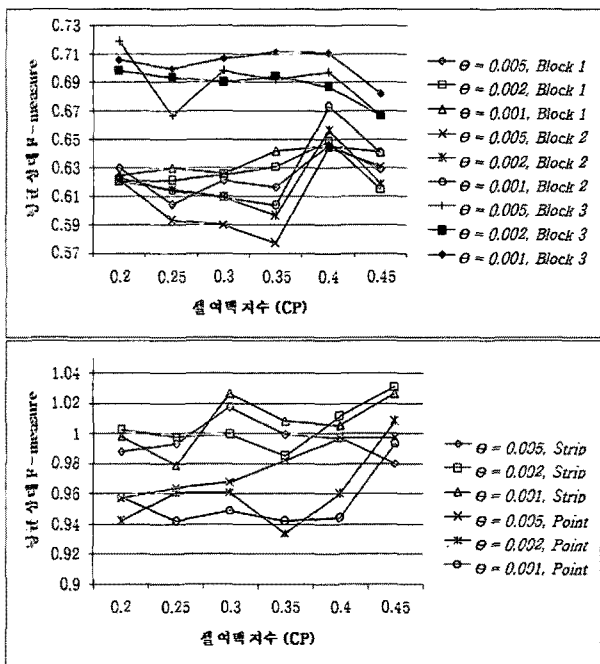


그림 13. 셀 여백 지수에 따른 성능 변화 (Mask 1)
Fig. 13. Mean relative F-measure with various CP-Mask1.

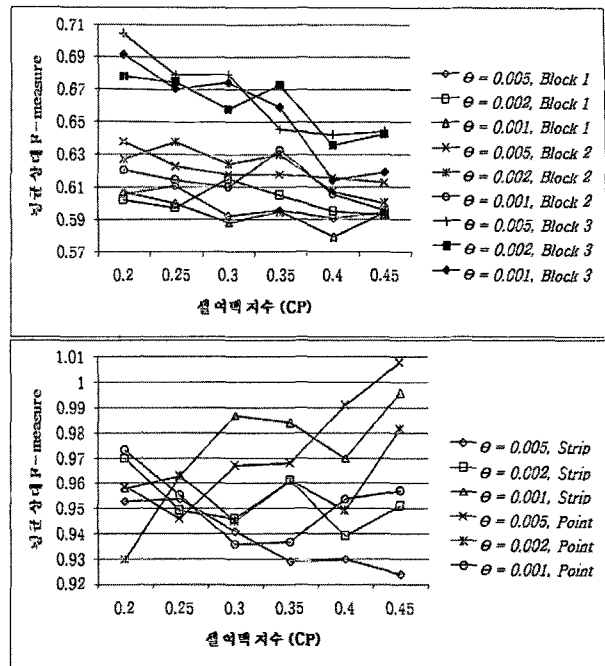


그림 14. 셀 여백 지수에 따른 성능 변화 (Mask 2)
Fig. 14. Mean relative F-measure with various CP-Mask2.

식이 가지는 특성이 그대로 잘 나타나있는데, 셀 여백 지수(CP)가 클수록 블록 패턴에서의 성능은 대체로 좋아지지만, 스트립 패턴과 포인트 패턴에서는 성능은 급격히 저하된다. 또한 오류 영역이 정사각형에 가까울수록 즉, Block 1 형식의 오류 영역을 가진 입력 도메인 경우 더 우수한 성능을 보였다. 하지만 스트립 패턴과 포인트 패턴의 경우, 마스크의 모양에 관계없이 1에 가깝거나 1보다 큰 값을 보였다. 즉, 셀 여백 지수가 적용이 되는 경우, 스트립 패턴과 포인트 패턴에서는 순수 랜덤 테스트와 유사하거나 더 낮은 성능을 보이게 된다. 하지만, 다른 ART 기법에서도 스트립 패턴과 포인트 패턴의 경우 유사한 성능을 보이고, 소프트웨어에서 발견되는 오류들이 대체로 블록 패턴과 스트립 패턴이라고 할 때^[11], 블록 패턴에 대한 성능이 해당 기법의 주요 평가 지표가 되므로, 적절한 수준의 셀 여백 지수의 적용은 기법의 전체적인 성능에 긍정적인 영향을 줄 수 있을 것으로 예상된다.

4. ART 기법 성능비교

4.1 시뮬레이션 환경

D-ART, RRT, IP-ART 및 M-IP-ART의 블록 패턴과 스트립 패턴에 대한 성능 비교를 위한 실험 환경은 표 3과 같다. 실험은 주어진 환경에서 각 ART 기법에 대해 5000회 테스트를 수행했을 때의 평균 상대

표 3. 실험 환경

Table 3. Simulation Environment.

테스트 회수	5000
입력 도메인	10000×10000, 2차원
오류 비율 (θ)	0.01, 0.005, 0.002, 0.001
오류 패턴	Block1, Block2, Block3, Strip
D-ART	$N = 10$
RRT	$R = 1.5$
M-IP-ART	$M = \text{Mask 2}, CP = 0.4$

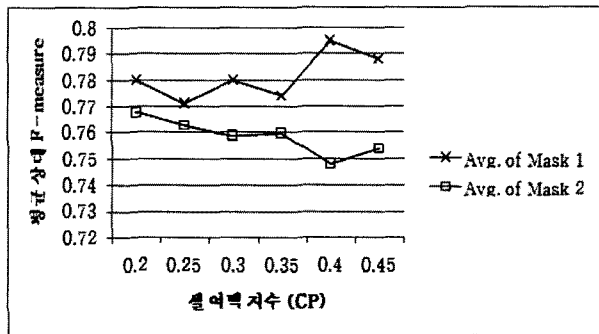


그림 15. 셀 여백 지수 변화에 따른 성능 변화 평균
Fig. 15. Average of mean relative F-measure.

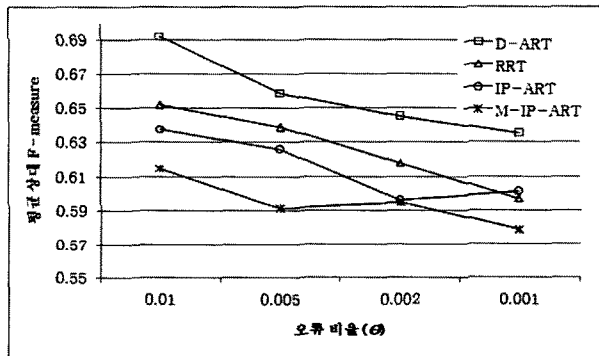


그림 16. 오류 비율에 따른 ART 기법별 성능 - Block 1
Fig. 16. Mean relative F-measure of each ART algorithm with various failure rate - Block 1.

F-measure를 구하도록 진행되었으며, 각 ART 기법에서 요구하는 파라미터는 각 기법에서 최적 성능을 보이는 것으로 알려진 값을 사용했다. 즉, D-ART는 후보 테스트 케이스 개수 $N = 10$ 을, RRT의 제외 영역 크기는 $R = 1.5$ 로 정의하였다^[7-8]. 그림 15는 전체 오류 비율 및 다양한 오류 패턴에 대해 수행한 셀 여백 지수 영향 평가 결과치의 평균을 나타낸 것이다. 이와 같은 결과에 따라 M-IP-ART는 $M = \text{Mask 2}, CP = 0.4$ 를 사용하도록 정의하였다.

그림 16은 Block 1 패턴에서, 오류 비율 변화에 따른 각 ART 기법의 성능 변화를 나타낸 것인데, 이 실험 결과에 따르면 모든 ART 기법에서 공통적으로 오류

영역을 구성하는 폭과 높이의 길이 차이가 클수록 (Block 1 < Block 2 < Block 3) 성능이 저하되는 특성을 보였으며, 특히 IP-ART의 경우 성능 저하 폭이 가장 큰 것으로 나타났다. 반면, M-IP-ART는 블록의 형태 변화에 따른 성능 편차가 다른 ART 기법보다 작아 전체적으로 비슷한 성능을 유지하고 있으며, 특히 Block 3 패턴의 경우 20% 정도 향상된 성능을 보이고 있다. 즉, M-IP-ART는 모든 블록 패턴에서 가장 좋은 성능을 보이는 것으로 나타나 다양한 블록 모양에 대한 적응성이 뛰어난 것을 알 수 있다.

4.2 블록 패턴에서 ART 기법별 성능 비교

그림 17은 블록 패턴에 대한 성능 비교 평가 실험 결과이다. 이 실험 결과에 따르면 모든 ART 기법이 공통적으로 오류 비율이 높아질수록 성능이 저하됨을 알 수 있다. 이는 각 기법의 고유한 테스트 케이스 선택 방식

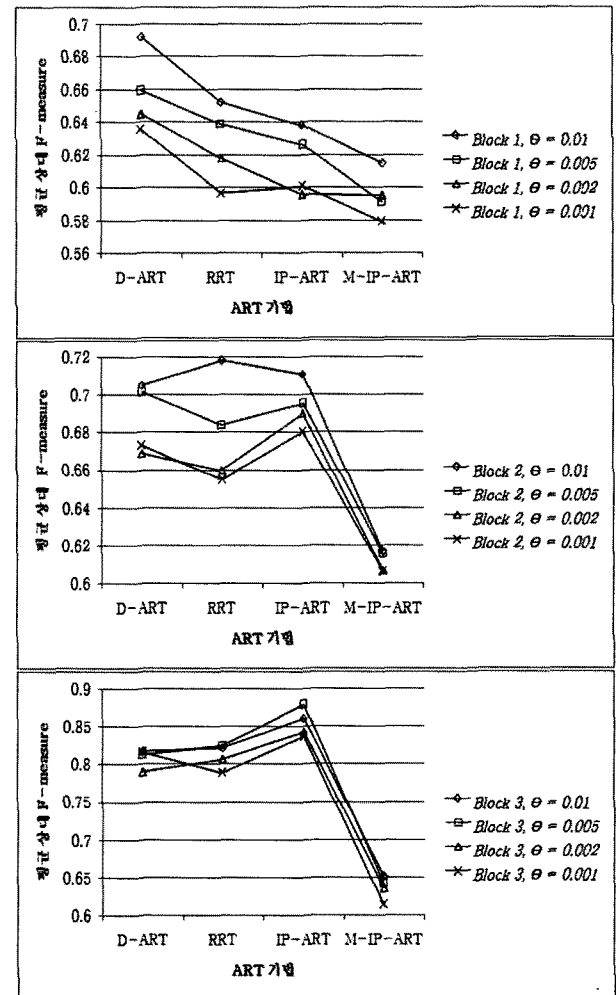


그림 17. ART 기법 성능 비교 - 블록 패턴
Fig. 17. Mean relative F-measure of each ART algorithm - Block pattern.

특성으로 인해 발생하는 테스트 케이스 분포의 불균일 현상 때문이며, 이러한 현상은 높은 오류 비율과 고차원 도메인에서 기법의 성능을 저하시키는 직접적인 원인이 된다^[10]. M-IP-ART 역시 높은 오류 비율을 가지는 환경에서 성능이 낮아지는 결과를 보이고 있지만, 모든 오류 비율에서 다른 ART 기법에 비해 좋은 성능을 유지하고, 또한 오류 비율 변화에 따른 성능 저하 격차가 다른 ART 기법에 비해 작음을 알 수 있다.

4.3 스트립 패턴에서 ART 기법별 성능 비교

그림 18은 스트립 패턴에 대한 각 ART 기법의 성능 평가 실험 결과이다. 스트립 패턴에서 각 ART 기법들은 모두 비슷한 결과를 보여주고 있으나, 오류 비율 0.001인 경우 D-ART가 가장 좋은 성능을 보이며, 그 외의 환경에서는 M-IP-ART가 가장 좋은 성능을 가지는 것으로 확인되었다. 이 실험 결과에 따르면, D-ART와 RRT는 오류 비율의 변화에 상관없이 항상 비슷한 수준의 성능을 유지하지만, IP-ART와 M-IP-ART는 오류 비율이 낮아질수록 성능도 저하되는 현상을 보인다. 이는 IP-ART와 M-IP-ART가 그리드 방식을 기본으로 하는 기법이기 때문에 선형이나 소규모의 클러스터를 대상으로 하는 환경에서는 그리드 방식이 가지는 장점이 충분히 발휘될 수 없기 때문인 것으로 추정되며, 특히 입력 도메인 내 오류비율이 0.001보다 더 낮은 환경일 경우에는 순수한 랜덤 테스트보다 낮은 성능을 보일 것이라 예상된다.

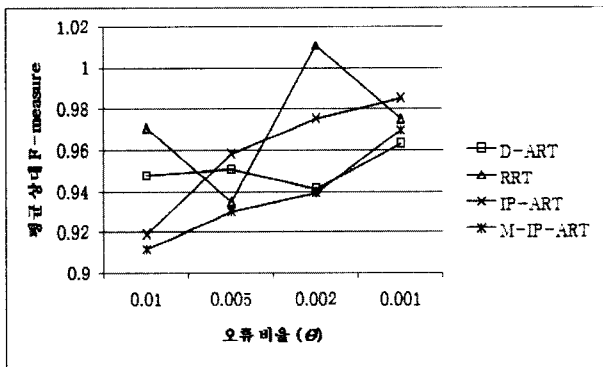


그림 18. 오류 비율에 따른 ART 성능 - 스트립 패턴
 Fig. 18. Mean relative F-measure of each ART algorithm with various failure rate - Strip pattern

V. 결 론

IP-ART는 D-ART나 RRT 기법의 단점인 테스트 케이스 선택에 소요되는 계산 부하를 그리드 방식을 이

용하여 효율적으로 제어하고, 다양한 환경에서 대체로 우수한 성능을 보인다. 하지만 IP-ART에서 사용하는 제한 영역이 가지는 특징에서 기인하는 입력 도메인 내 테스트 케이스 분포의 불균일 문제는 여전히 존재하여 해당 기법의 다차원 입력 도메인에 대한 적용이나 특정 오류 패턴에 대한 성능 저하 등을 야기한다. 이에 따라 본 논문에서는 IP-ART에서 사용하는 제한 영역의 크기와 모양이 입력 도메인 내 테스트 케이스 분포에 미치는 영향을 새로운 제한 영역 마스크의 정의 및 여러 다른 넓이를 가진 제한 영역 마스크를 사용해 확인하고, 그리드 방식이 가지는 특징을 효과적으로 적용하기 위한 셀 여백 지수를 정의했으며, 실험을 통해 이의 사용 가능성 여부를 실험을 통해 확인하였다.

제한 영역 마스크의 형태 변경에 따른 실험에서는 Mask 1의 경우 근거리에서 위치한 테스트 케이스 선택 가능성으로 인해 기존 IP-ART에 비해 낮은 성능을 보이지만, Mask 2의 경우 약 4% 정도 향상된 성능을 보였다. 즉, 제한 영역으로 사용하는 마스크의 크기는 테스트 케이스 분포와 알고리즘 성능 사이의 트레이드오프를 유발하기 때문에, 적절한 크기와 모양을 가지는 제한 영역 마스크를 사용할 때 더 우수한 성능을 얻을 수 있음을 의미한다. 한편 Mask 1과 2에 추가적으로 셀 여백 지수를 적용했을 때에는 Mask 1의 경우 평균 7%, Mask 2의 경우 평균 9% 정도 향상된 성능을 보이는 것으로 나타났으며, 특히 블록의 폭과 높이의 차이가 큰 Block 3에 대해서는 최고 20%까지의 성능 차이를 보였다. 이는 그리드 방식이 블록 패턴에 대해 가지는 특징이 이용된 결과이며, 특히 Mask 1은 근거리에서 위치한 테스트 케이스 생성 가능성으로 인해 여백 지수의 적용이 없으면 IP-ART보다 낮은 성능을 보임을 비추어볼 때, 이와 같은 결과는 여백 지수의 적용이 블록 패턴의 검출에 매우 큰 영향을 미치고 있음을 나타낸다.

셀 여백 지수와 제한 영역 마스크의 적용은 부수적인 연산을 요구하지 않는 방법이다. 따라서 제안된 제한 영역 마스크를 사용할 경우 2차원 입력 도메인에서는 탁월한 성능 향상을 보이지는 않지만, 테스트 케이스 분포의 균일화를 통한 기법의 확장 적용성 확충을 위해 사용될 필요가 있으며, 셀 여백 지수는 주된 오류 패턴인 블록 패턴에 대한 성능 향상을 목적으로 적용이 필요해지므로, 이 두 가지 방법을 개별적으로 적용하는 것보다는 모두 적용하는 것이 바람직할 것으로 보인다.

스트립 패턴과 포인트 패턴에 대한 제안된 기법의 성

능은 블록 패턴에 대한 성능보다 상대적으로 낮은 것으로 나타났다. 이러한 현상은 대부분의 기존 ART 기법에서 나타나는 현상이며, 모든 ART 기법에서 해결해야 할 과제로 남아있는 부분이다.

본 논문에서는 셀 여백 지수 적용과 다양한 제한 영역 마스크 사용이 IP-ART 기법에 미치는 영향을 평가하였으며, 이를 통해 제안된 두 가지 방법을 IP-ART에 적용할 경우 대부분의 환경에서 기존의 ART 기법들보다 우수한 성능을 보이는 것으로 나타났다. 하지만 제안된 기법의 다차원 입력 도메인에 대한 적용 가능성 확인과 오류 비율이 아주 낮은 환경 및 스트림 패턴과 포인트 패턴에 대한 ART 기법의 적응력 향상, 그리고 제한 영역이 테스트 케이스 분포에 미치는 영향 평가는 추가 연구가 필요한 과제이다.

참고 문헌

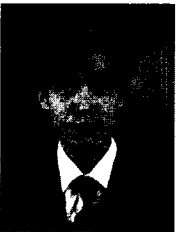
- [1] B. Hailpern, P. Santhanam, "Software Debugging, Testing, and Verification", IBM Systems Journal, vol. 41, no. 1, pp. 4-12, June 2002.
- [2] T. Y. Chen, D. H. Huang, Z. Q. Zhou, "Adaptive Random Testing Through Iterative Partitioning", in Proc. of the 11th Intl. Conf. on Reliable Software Tech., vol. 4006 of LNCS, pp. 155-166. Porto, Portugal, June 2006.
- [3] R. Hamlet, "Random testing", Encyclopedia of Software Engineering, Wiley, pp. 970-978, 1994.
- [4] P. S. Loo, W. K. Tsai, "Random Testing Revisited", Information and Software Technology, vol. 30, No. 9, pp. 402-417, September 1988.
- [5] J. Mayer, C. Schneckenburger, "An Empirical Analysis and Comparison of Random Testing Techniques", in Proc. of the 2006 ACM/IEEE Intl. Symposium on Empirical Software Engineering (ISESE 2006), pp. 105-114, Rio de Janeiro, Brazil, September 2006.
- [6] J. Mayer, "Adaptive Random Testing with Randomly Translated Failure Region", in Proc. of the 1st Intl. Workshop on Random Testing (RT 2006), pp. 70-77, Maine, USA, July 2006.
- [7] T. Y. Chen, H. Leung, and I. K. Mak, "Adaptive Random Testing", in Proc. of the 9th Asian Computing Science Conf. (ASIAN 2004), vol. 3321 of LNCS, pp. 320-329. Chiang Mai, Thailand, December 2004.
- [8] K. P. Chan, T. Y. Chen, D. Towey, "Restricted Random Testing", in Proc. of the 7th European Conf. on Software Quality (ECSQ 2002), vol. 2349 of LNCS, pp. 321-330, Helsinki, Finland, June 2002.
- [9] T. Y. Chen, G. Eddy, R. Merkel, P. K. Wong, "Adaptive Random Testing Through Dynamic Partitioning", in Proc. of the 4th Intl. Conf. on Quality Software (QSIC 2004), pp. 79-86. Braunschweig, Germany, September 2004.
- [10] J. Mayer, C. Schneckenburger, "Adaptive Random Testing with Enlarged Input Domain", in Proc. of the Sixth Intl. Conf. on Quality Software (QSIC 2006), pp. 251-258, Beijing, China, October 2006.
- [11] 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, vol. 38, no. 12, pp. 775-782, February 1996.
- [12] T. Y. Chen, F. C. Kuo, R. G. Merkel, S. P. Ng, "Mirror Adaptive Random Testing", Information and Software Technology, vol. 46. issue 15, pp. 1001-1010, December 2004.
- [13] J. Mayer, "Adaptive Random Testing by Bisection and Localization", in Proc. of the 5th Intl. Workshop on Formal Approaches to Testing of Software (FATES 2005), vol. 3997 of LNCS, pp. 72-86, Scotland, UK, July 2005.
- [14] T. Y. Chen, D. H. Huang, "Adaptive Random Testing by Localization", in Proc. of the 11th Asia-Pacific Software Engineering Conf. (APSEC 2004), vol. 00, pp. 292-298, Busan, Korea, December 2004.
- [15] J. Mayer, "Adaptive Random Testing by Bisection with Restriction", in Proc. of the 7th Intl. Conf. on Formal Engineering Methods (ICFEM 2005), vol. 3785 of LNCS, pages 251-263, Manchester, UK, November 2005.
- [16] J. Mayer, "Lattice-based Adaptive Random Testing", in Proc. of the 20th IEEE/ACM Intl. Conf. on Automated Software Engineering (ASE 2005), pp. 333-336, CA, USA, November 2005.

저 자 소 개



이 광 규(정회원)
 2006년 아주대학교 정보 및
 컴퓨터공학부 학사
 2008년 아주대학교
 정보통신공학과 석사
 2008년 3월~현재
 (주) NHN QA센터

<주관심분야 : 소프트웨어공학, 소프트웨어 테스
 팅, 테스트 자동화>



신 승 훈(정회원)
 2000년 아주대학교 정보 및
 컴퓨터공학부 학사
 2002년 아주대학교
 정보통신공학과 석사
 2004년 아주대학교
 정보통신공학과 박사수료

<주관심분야 : 소프트웨어 테스트, 테스트 자동
 화, 멀티미디어 전송 정책>



박 승 규(평생회원)
 1974년 서울대학교
 응용수학과 학사
 1976년 한국과학원 (KAIST)
 전산학과 석사
 1982년 Institut National
 Polytechnique de
 Grenoble 전산학과 박사

1976년~1977년 한국과학기술연구소(KIST)
 연구원

1977년~1978년 KIET (현ETRI) 연구원

1978년~1982년 프랑스 그레노블 IMAG 연구원
 학생

1982년~1984년 KIET (현ETRI) 실장/
 선임연구원

1984년~1985년 미국 IBM 왓슨연구소 연구원

1985년~1992년 ETRI 연구위원/책임연구원

1992년~현재 아주대학교 정보통신대학 교수

<주관심분야 : 임베디드 테스트, 자가 컴퓨팅/치
 료 시스템, 차세대 컴퓨터 구조>