논문 2009-46CI-3-9

입력 도메인 반복 분할 기법 성능 향상을 위한 고려 사항 분석

(Revision of ART with Iterative Partitioning for Performance Improvement)

신 승 훈*, 박 승 규**, 정 기 현***

(Seung-Hun Shin, Seung-Kyu Park, and Ki-Hyun Jung)

요 약

적응적 랜덤 테스팅 (Adaptive Random Testing, ART) 기법 중 입력 도메인 반복 분할을 이용한 기법(ART through Iterative Partitioning, IP-ART)은 초기 ART 기법이 가지는 단점인 많은 거리 연산량을 입력 도메인 반복 분할 기법을 이용해 효율적으로 개선하였다. 또한 입력 도메인 확장을 이용한 IP-ART 기법(IP-ART with Enlarged Input Domain, EIP-ART)은 IP-ART가 가지는 특징을 효과적으로 이용해 IP-ART에서 보이는 테스트 케이스 분포 불균일 현상을 완화시켰으며, 다양한 환경에서 좋은 성능을 보인다. 하지만 EIP-ART는 입력 도메인 확장에 의해 발생되는 부수적인 부하로 인해 테스트 케이스 선택이 지연된다는 단점을 가진다. 이에 따라 본 논문에서는 입력 도메인의 확장 없이, IP-ART의 변화를 통한 성능 개선 가능성 확인을 위해 테스트 케이스 분포에 영향을 줄 수 있는 세 가지 조건을 정의하고 이들 각각이 기법에 미치는 영향을 평가하였으며, 실험을 통해 정의된 세 가지 조건 중 제한 영역 관리 조건이 기법의 성능에 가장 긍정적인 영향을 주는 것을 확인하였다.

Abstract

Adaptive Random Testing through Iterative Partitioning(IP-ART) is one of Adaptive Random Testing(ART) techniques. IP-ART uses an iterative partitioning method for input domain to improve the performances of early-versions of ART that have significant drawbacks in computation time. Another version of IP-ART, named with EIP-ART(IP-ART with Enlarged Input Domain), uses virtually enlarged input domain to remove the unevenly distributed parts near the boundary of the domain. EIP-ART could mitigate non-uniform test case distribution of IP-ART and achieve relatively high performances in a variety of input domain environments. The EIP-ART algorithm, however, have the drawback of higher computation time to generate test cases mainly due to the additional workload from enlarged input domain. For this reason, a revised version of IP-ART without input domain enlargement needs to improve the distribution of test cases to remove the additional time cost. We explore three smoothing algorithms which influence the distribution of test cases, and analyze to check if any performance improvements take place by them. The simulation results show that the algorithm of a restriction area management achieves better performance than other ones.

Keywords: 랜덤 테스팅, 적응적 랜덤 테스팅, 테스트 케이스 선택

I. 서 론

랜덤 테스팅(Random Testing)은 한 소프트웨어에 대한 입력 도메인 내에서 균일 분포(uniform distribution)를 따르도록 임의의 테스트 케이스를 생성하고, 이를 이용한 테스트를 통해 소프트웨어 내에 존재하는 오류를 찾아내는 기본적인 테스팅 전략이다. 랜덤 테스팅이 가지는 장점 중 하나인 테스트 케이스 생성의 용이성으로 인해 랜덤 테스팅은 소프트웨어 테스팅 자동화 및 다양한 애플리케이션 테스트에 성공적으로 적용되어 왔지만[1~2], 소프트웨어에 대한 사전 지식이 전혀 없는 상태에서 테스트 케이스를 선택한다는 특징 때문에 랜

* 정회원, 아주대학교 정보통신전문대학원
(Graduate School of Information & Communication, Ajou University)
** 평생회원, *** 정회원, 아주대학교 정보통신대학
(College of Information Technology, Ajou University)
접수일자 : 2009년11월17일, 수정완료일: 2009년4월28일

덤 테스팅의 효율성에 대한 회의론 또한 존재한다[3]. 이러한 회의론에도 불구하고 랜덤 테스팅이 가지는 특징인 테스트되는 소프트웨어의 구조 및 명세서(specification)에 대한 독립성[4]을 이용한 다양한 시도를 통해 랜덤 테스팅의 효율성은 꾸준히 검증되어 왔으며[5~6], 효율적인 랜덤 테스트 케이스의 생성을 통해 소프트웨어 내의 오류를 더욱 효과적으로 찾아내기 위한 시도 또한 계속되어 왔다[7~14].

효율적인 랜덤 테스트 케이스 생성 기법 중 하나인 적응적 랜덤 테스팅(Adaptive Random Testing, ART)은 소프트웨어 내에 존재하는 오류의 특성에 근거하여 오류를 더욱 효율적으로 찾아낼 수 있도록 하는 정책을 사용하고, 이를 통해 순수 랜덤 테스팅이 가지는 성능을 높이는데 목적을 두며[12], 이를 이용한 다양한 기법들이 개발되어 왔다[7~14]. 초기의 ART 기법들은 순수 랜덤 테스팅보다 높은 성능을 보이기는 하지만, 테스트 케이스 생성에 많은 연산량을 필요로 하고, 테스트 케이스의 입력 도메인 내 분포 불균일 현상과 같은 단점을 갖는데, 이러한 단점들은 테스트 케이스 생성에 소요되는 시간 비용의 증가와 입력 도메인의 차원 수 증가에 따른 성능 저하를 유발한다. 이러한 단점을 보완하기 위해 제안된 입력 도메인 반복 분할 기법(ART through Iterative Partitioning, IP-ART)은 테스트 케이스 생성에 요구되는 연산량을 상당 부분 감소시켰으며[11], 입력 도메인 가상 확장을 이용한 IP-ART 기법(IP-ART with Enlarged Input Domain, EIP-ART)은 실제 입력 도메인 내의 테스트 케이스 분포 불균일 수준이 완화되도록 하였다[7]. 하지만 IP-ART는 특정 형태의 테스트 케이스 분포 불균일 현상을 유발하여 성능 향상이 제한적이며, EIP-ART는 테스트 케이스 생성에 필요한 부수적인 연산량이 입력 도메인 차원 수 증가에 따라 급속하게 증가하는 등의 단점을 가지고 있어, IP-ART가 가지는 특성을 이용한 개선책(Modified IP-ART, M-IP-ART)[8]이 제시되었다. 여기에서는 다양한 모양의 제한 영역이 성능에 미치는 영향을 평가하였으며, 그 결과 IP-ART보다 일정 수준의 성능 개선 및 테스트 케이스 불균일 분포 현상의 완화 등을 확인할 수 있었으나, 사용되는 제한 영역의 형태에 따른 '테스트 케이스 분포'와 '선택된 테스트 케이스 간 거리' 사이의 트레이드오프 현상 또한 발견되었다. 이는 제한 영역 형태 변화 이외의 방법을 이용한 성능 개선 정책이 필요함을 의미하며, 이에 따라 본 논문에서는 IP-ART가 가지는 입력 도메인 내 테스트 케이스 분포 특성을 파악하고, 파악된 정보를 기반으로 고려할 수 있는 IP-ART의 수정 방안을 제시하며, 이들이 IP-ART의 성능과 테스트 케이스의 입력 도메인 내 분포에 미치는 영향을 실험을 통해 비교 분석한다.

Ⅱ. 정의 및 관련 연구

1. 정의

한 소프트웨어에 대한 유한 크기의 입력 데이터 집합을 입력 도메인이라고 하고, 입력 도메인 내에 존재하는 입력 데이터 가운데 소프트웨어의 오류를 유발하는 입력 데이터의 전체 입력 데이터에 대한 비율을 오류 비율(θ)이라 정의한다. 소프트웨어의 오류를 유발하는 입력 데이터는 몇 가지 유형을 가지고 입력 도메인 내에 군집을 이루어 존재하는 성향을 가지는데, 이를 오류 패턴이라고 한다. 이와 같이 군집을 이루는 오류 유발 데이터의 특성에 근거하여, ART에서는 이전에 사용된 테스트 케이스 인근에 위치하는 입력 데이터보다, 일정 거리 이상 떨어진 곳에 위치하는 입력 데이터를 선호하는 정책을 사용하며, 이와 동시에, 선택되는 테스트 케이스가 입력 도메인 내에 가능한 고르게 분포될 수 있도록 한다. 오류 패턴은 블록(Block), 스트립(Strip) 및 포인트(Point) 등 세 가지로 구분되고, 이 중 블록 패턴 및 스트립 패턴이 상대적으로 빈번하게 나타나는 오류 패턴이기 때문에[4], 기존 ART 관련 연구에서는 블록 패턴을 중심으로 제안된 기법의 성능 평가를 수행하였으며, 본 논문에서도 블록 패턴을 대상으로 기법의 성능 을 평가한다.

F-지수(F-measure)는 소프트웨어의 테스팅 수행 중 첫 번째 오류를 발견할 때까지 사용된 테스트 케이스의 수를 나타내는데[12], 오류 발견 시 해당 오류에 대한 원인 파악 및 조치가 수행되는 소프트웨어 테스팅 환경을 고려할 때, F-지수는 직관적인 성능 평가 및 비교 기준으로 사용될 수 있다. 이에 따라 기존의 ART 관련 연구에서는 제안 기법의 성능 평가에 F-지수를 사용하였고, 본 논문에서도 이를 이용하여 기법의 성능을 비교 평가한다. 하지만 F-지수는 입력 도메인 내에 존재하는 오류 비율에 영향을 받기 때문에 기법 간 성능의 직접 비교 기준으로 사용하기 어렵다. 따라서 순수 랜덤 테스팅의 성능을 기준으로 하는 상대 F-지수를 사용한다. 입력 도메인 내의 오류 비율이 θ일 때, 순수 랜덤 테스

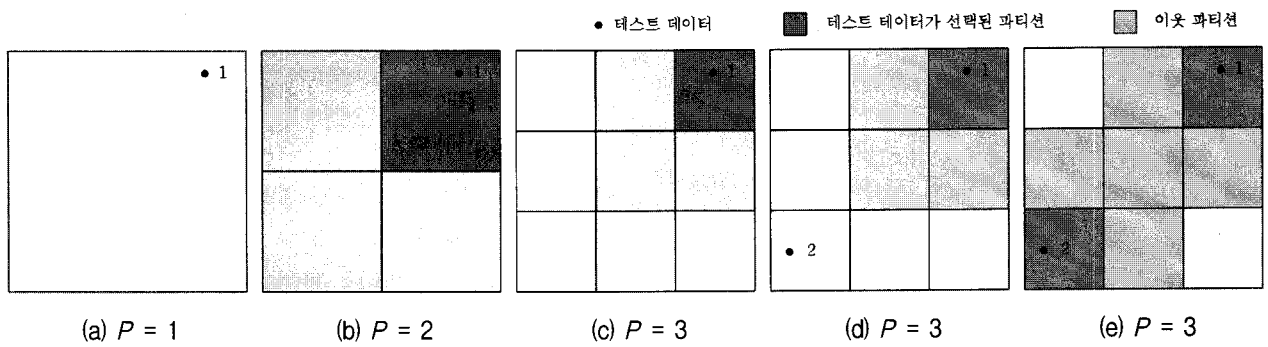


그림 1. IP-ART 수행 예
Fig. 1. An Example of The IP-ART Algorithm.

팅의 이론적 F-지수는 $F_{RT}=1/\theta$이고, 다른 기법들은 이에 대한 상대 수치인 평균 상대 F-지수(Mean relative F-measure)로 성능을 평가하게 되며, 평균 상대 F-지수는 다음과 같이 정의된다.

평균 상대 F-지수 = (평균 F_{ART}) / F_{RT}

즉, 평균 상대 F-지수는 제안된 기법을 사용하여 얻은 F-지수 평균값과 순수 랜덤 테스팅을 사용했을 때의 이론적 F-지수의 비율이 된다. 따라서 평균 상대 F-지수가 1인 경우, 이 기법은 순수 랜덤 테스팅과 동일한 성능을 가지는 것을 의미하고, 평균 상대 F-지수가 0에 가까울수록 순수 랜덤 테스팅보다 우수한 성능을, 1보다 값이 커질수록 낮은 성능을 가짐을 의미한다.

2. 관련 연구 및 문제점

초기의 거리 연산 기반 ART 기법이 가지는 단점인 과다한 거리 연산 필요성으로 인한 테스트 케이스 선택 지연 문제를 해결하기 위해 제안된 IP-ART는 테스트 케이스 간 거리 연산 대신 입력 도메인을 파티션으로 분할하고, 분할 영역과 분할 영역 내의 데이터를 임의로 선택하는 방식을 사용한다. IP-ART는 테스트의 진행에 따라 입력 도메인을 반복 분할해 입력 도메인을 구성하는 파티션의 크기를 점점 줄여가도록 하고, 테스트 케이스가 선택되었던 파티션과 해당 파티션의 인접 파티션을 테스트 케이스 선택에서 배제되는 제한 영역으로 설정하도록 하는 기법이며, 그림 1과 같은 형태로 동작한다.

IP-ART는 맨 처음 입력 도메인 전체를 포함하는 1×1 파티션을 구성한 후(P=1), 해당 파티션 내에서 임의의 테스트 케이스(1)를 선택하고(그림 1-(a)), 이를 이용하여 테스트를 수행한다. 테스트 수행 결과 소프트웨어에서 오류가 발생되지 않은 경우, 분할 수준을 높여(P=2) 입력 도메인을 분할하고, 이전 테스트 케이스를 포함하는 파티션과 그 이웃 파티션을 제한 영역으로 설정한다(그림 1-(b)). 이 때, 모든 파티션이 제한 영역에 포함되어 선택 가능한 파티션이 없으므로 입력 도메인을 재분할하고(P=3) 이전 테스트 케이스를 포함한 파티션과 그 이웃 파티션을 제한 영역으로 설정한다(그림 1-(c)). 분할된 영역 중 제한 영역으로 설정되지 않은 파티션 하나를 임의로 선택하고, 해당 파티션 내에서 임의의 데이터를 선택하여 다음 테스트 케이스(2)로 이용한다(그림 1-(d)). 선택된 테스트 케이스를 이용한 테스트에서 오류가 발견되지 않으면, 테스트 케이스가 선택된 파티션과 그 이웃 파티션을 제한 영역으로 지정하고 (그림 1-(e)), 이후 아직 제한 영역으로 설정되지 않은 파티션과 파티션 내의 입력 데이터를 임의로 선택하는 작업을 수행하며, 이와 같은 과정은 오류를 유발하는 테스트 케이스를 찾을 때까지 반복된다. IP-ART는 이와 같은 방법을 통해 초기 ART 기법에서 사용했던 거리 기반 연산을 배제하고, 테스트 케이스의 입력 도메인 내 불균일 분포 문제도 완화시켜, 테스트 케이스 생성 시간과 기법의 성능 모두에서 성과를 보였다[11].

하지만 IP-ART 기법을 사용했을 때에도 테스트 케이스의 분포 불균일 현상은 여전히 존재하여 기법의 성능 향상에 대한 장애 요소로 작용한다. 이와 같은 현상의 완화를 위하여 입력 도메인 가상 확장을 이용한 EIP-ART 기법[7]이 제안되었고, 이는 IP-ART의 수행 결과 나타나는 테스트 케이스 분포 불균일 현상의 완화와 성능 개선 등의 효과를 보였지만, EIP-ART 기법을 사용하는 경우, 입력 도메인의 차원 수가 증가함에 따라 실제 입력 도메인이 전체 (실제+가상) 입력 도메인에서 차지하는 비율이 급격히 감소되고, 이는 가상 도

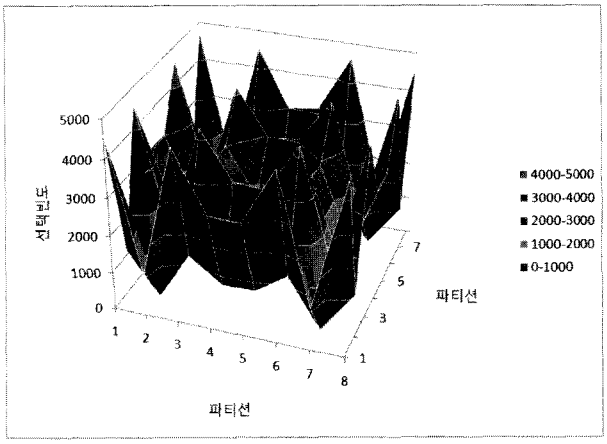
메인 내에 위치하는 가상 테스트 케이스의 관리 부하를 증가시켜 테스트 케이스 선택에 오랜 시간을 요구하는 단점을 유발한다. 이에 따라 입력 도메인의 확장과 같은 많은 부하 유발 없이 실제 입력 도메인에 대해 적용했을 때 고른 테스트 케이스 분포를 유도 가능한 정책이 요구되며, 본 논문에서는 실제 도메인 내에 적용 가능한 몇 가지 방안을 제시한다.

III. 테스트 케이스 선택 정책

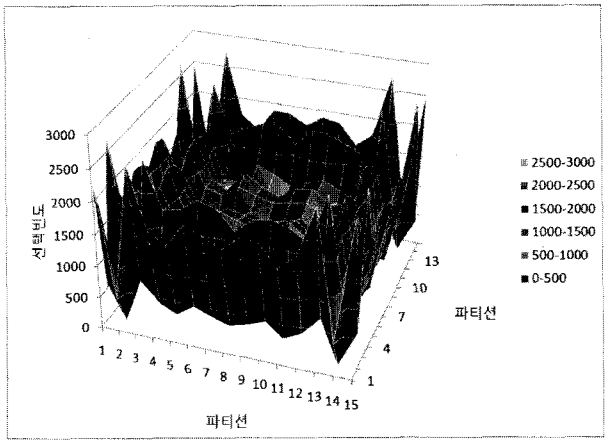
3장에서는 IP-ART 기법을 사용할 때 나타나는 테스트 케이스의 입력 도메인 내 분포를 살펴보고, 여기에서 나타나는 불균일 특성에 대한 개선 조건을 정의한다.

1. IP-ART 테스트 케이스 분포

그림 2는 2차원 입력 도메인에 대해 IP-ART 기법을 적용했을 때 생성되는 테스트 케이스의 공간 분포이다. 총 50000번의 테스트를 수행하는 동안 입력 도메인 분할 수준이 각각 8, 15일 때(P=8, P=15), 파티션이 선택된 빈도, 즉 해당 영역에서 테스트 케이스가 선택되는 빈도를 나타내고 있다. 테스트 케이스의 선택 빈도는 입력 도메인 주위를 둘러싸는 첫 번째, 세 번째 라인에 위치한 파티션에서 높게 나타났고, 네 번째 라인의 안쪽은 전체 선택 빈도 평균과 유사하거나 약간 낮으며, 선택 빈도 차이가 그리 크지 않은 특징을 보인다. 주목할 곳은 선택 빈도가 높게 나타나는 첫 번째와 세 번째 라인이 교차하는 지점과, 선택 빈도가 낮게 나타나는 두 번째 라인 및 두 번째 라인이 교차하는 곳에 위치하는 파티션이다. 첫 번째와 세 번째 라인이 교차하는 곳에 위치하는 파티션에서 테스트 케이스가 선택되는 빈도는 평균 선택 빈도의 두 배 이상이 되며, 이와 같은 현상의 영향으로 그 사이에 위치하는 두 번째 라인 상에 존재하는 파티션에서 테스트 케이스가 선택되는 빈도가 매우 낮아진다. 특히 두 번째 라인이 교차하는 곳에 위치하는 파티션에서는 단 하나의 테스트 케이스 선택도 이루어지지 않았다. 이와 같은 현상이 발생하는 원인은 각 테스트 케이스의 첫 번째 입력 데이터 좌표 분포를 표현한 그림 3에서 찾을 수 있다. (두 번째 입력 데이터의 좌표 분포 및 고차원 테스트 케이스에서도 유사한 형태의 분포를 갖는다.) 그림 3은 그림 2와 동일한 실험 조건에서 생성된 테스트 케이스의 첫 번째 입력



(a) P=8, 평균 = 2390



(b) P=15, 평균 = 1264

그림 2. 테스트 케이스 분포 - IP-ART
Fig. 2. A Test Case Distribution of IP-ART.

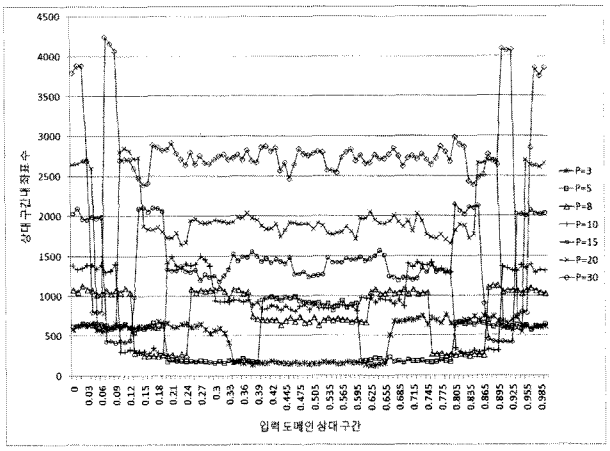


그림 3. 입력 도메인 분할 증가에 따른 첫 번째 입력 데이터 좌표의 분포
Fig. 3. A Coordinates Distribution of The First Input Data for each Partition Level.

데이터의 좌표만을 표현하였으며, 입력 도메인을 [0-1] 구간으로 변환하고, 이를 0.005 간격의 구간으로 나누었을 때, 각 구간 내에 위치하는 첫 번째 입력 데이터의

상대 좌표 빈도를 표시한 것이다. 그림 3에 따르면 파티션 분할의 영향이 없는 분할 수준(P=1 or 2)을 제외한 모든 분할 수준에서 테스트 케이스 분포의 불균일 현상이 나타난다. 이러한 불균일 현상은 해당 분할 수준에서 선택 가능한 파티션의 수가 많을수록, 즉 입력 도메인의 분할 수준이 증가할수록 심해지는 특징을 보이며, P=5 이상에서는 공통적으로 좌우측의 1, 3번째 파티션 선택 빈도는 매우 높은 반면, 그 사이에 존재하는 2번째 파티션의 선택 가능성은 매우 낮다. 그림 3은 첫 번째 입력 데이터의 좌표의 분포만을 표현한 것이기 때문에, 이와 같은 현상이 두 번째 입력 데이터의 좌표에서도 거의 동일하게 나타나는 것을 고려하면, 그림 2에 나타난 불균일 현상은 모든 입력 도메인 분할 수준에서 유사한 형태로 발생되며, IP-ART의 성능에 부정적 영향을 미치고 있음을 알 수 있다.

2. EIP-ART의 특징

EIP-ART는 그림 2, 3에 나타난 것과 같은 IP-ART의 테스트 케이스 분포 불균일 현상을 입력 도메인의 가상 확장을 통해 감소시키기 위한 기법이다. 그림 2와 3에서 알 수 있듯이 IP-ART를 이용하는 경우, 입력 도메인을 둘러싸는 외곽 영역의 세 라인은 테스트 케이스 선택 빈도 차이가 심하고, 네 번째 라인 안쪽 영역은 이의 외곽 영역보다 상대적으로 균등한 분포를 가진다. 따라서 입력 도메인을 일정 수준 확장하면 선택 빈도 차이가 심한 영역을 입력 도메인의 가상 확장 영역에 위치시킬 수 있게 되어, 실제 도메인 내에서는 심한 불균일 현상이 나타나지 않게 된다. 입력 도메인 확장 수준은 기반이 되는 제한 영역의 형태에 따라 변화되는데, IP-ART의 경우 한 파티션 폭의 세 배에 해당하는 영역을 입력 도메인 둘레에 추가하여, 첫 번째와 세 번째 라인에서 발생하는 심한 불균일 현상이 가상 확장 영역에서 발생되도록 제어한다. 그림 2에서 알 수 있듯이 네 번째 라인 내부의 파티션 별 선택 빈도는 그 외곽 영역에 비해 상대적으로 균일하기 때문에 높은 성능 향상을 기대할 수 있으나, 네 번째 라인 내부에 대해 이루어지는 선택은 확장 영역을 포함한 전체 도메인 내에서 이루어지는 선택 중 일부분(2차원 입력 도메인, P=15일 때, 33%)으로 한정되기 때문에, 확장 영역에 위치하는 테스트 케이스의 관리에 더 많은 비용이 소모되어 테스트 케이스 선택 속도가 느려진다. 또한 실제 도메인이 확장 영역을 포함한 전체 도메인에서 차지하는 부분은 입력 도메인 차원 수(D)가 증가할수록 $(P/(2*\text{확장 지수}+P))^D$로 급격하게 줄어들기 때문에, 테스트 케이스 선택 지연은 입력 도메인 차원 수가 증가할수록 매우 심해지는 단점을 가진다. 따라서 입력 도메인의 확장과 같은 추가 자원을 요구하는 정책이 아닌, IP-ART 고유의 특징을 적절히 조절하여 테스트 케이스 분포 불균일 현상을 제어하고, 이를 통해 기법의 성능을 향상 시킬 수 있는 정책이 요구된다.

3. 테스트 케이스 선택 제어 정책

표 1은 2차원 입력 도메인에서 선택된 테스트 케이스의 첫 번째 입력 데이터 좌표를 기준으로 선택 빈도가 높은 영역 선택 수에 대한 선택 빈도가 낮은 영역 선택 수의 비율을 파티션 분할 수준별로 나열한 것이다. 표 1에 따르면 P=5를 제외한 나머지 대부분의 분할 수준에서 이 비율은 유사한 값을 가진다. 즉, IP-ART의 테스트 케이스 분포 불균일 현상은 파티션 분할 수준에 독립적으로, 테스팅 과정 전체에서 유사한 영향을 미치는 것을 알 수 있다. P=5의 경우에는 입력 도메인 가장자리에서 세 번째 라인에 위치하는 파티션은 입력 도메인 중심에 위치하는 유일한 파티션이 된다. 따라서 이 파티션은 다른 분할 수준에서와는 달리 상대적으로 매우 높은 선택 빈도를 가지게 되어, 입력 도메인을 둘러싸는 첫 번째 라인에 대한 비율과 세 번째 라인에 대한 비율이 다르게 나타나게 된다.

표 1. 파티션 선택 빈도 비율
Table 1. Partition Selection Rate.

파티션 분할 수준	비율 (낮은 영역 / 높은 영역)
P=3	0.26
P=5	0.19 / 0.45
P=8	0.25
P=10	0.26
P=15	0.25
P=20	0.26
P=30	0.24
평균 (P=5 제외)	0.25

이러한 결과를 바탕으로 테스트 케이스 선택이 특정 영역에 집중되는 현상을 회피하기 위해 고려될 수 있는 조건을 정리하면 다음과 같다.

- 배제 (Rejection) : 선택 빈도가 높은 영역에서 발생되

는 선택을 일정 비율로 배제하여, 상대적으로 낮은 빈도를 가지는 영역에서 선택되도록 유도.

- 이동 (Translated Partition) : 선택 빈도가 높은 영역에서 발생되는 선택을 일정/적응적 비율로 상대적으로 낮은 빈도를 가지는 영역으로 이동.
- 제한 영역 관리 (Restriction Area Management) : 선택 빈도가 높은 영역에서 선택된 테스트 케이스의 제한 영역은 상대적으로 낮은 빈도를 가지는 영역에 주는 영향이 감소되도록 관리.

이와 같은 조건들은 공통적으로 테스트 케이스 분포 불균일 현상이 나타나는 입력 도메인 분할 수준 P=3 이상에 적용되어야 하고, 테스트 케이스가 아닌 파티션 자체에 부여되는 조건이 된다. 그리고 배제와 이동이 파티션 선택에 대한 조건이라면, 제한 영역 제어는 선택된 파티션 주위에 설정되는 제한 영역을 제어하기 위한 조건이 된다. 각 조건은 각자가 가지는 형식에 따라 IP-ART 기법에 서로 다른 영향을 미칠 것으로 예측된다.

(1) 배제 조건

배제 조건이 추가된 IP-ART 기법의 테스트 케이스 선택 과정은 표 2와 같다. 배제 조건이 추가되면, 임의로 선택된 파티션이 선택 빈도가 높은 영역에 위치하는 경우, 이를 일정 확률로 배제한다. 이와 같은 조건은 상대적으로 낮은 선택 빈도를 가지는 영역에 확률적으로 좀 더 많은 테스트 케이스 선택 기회를 부여할 수 있게 되지만, 선택 빈도가 높은 파티션이 일정 확률로 배제되기 때문에 상대적으로 테스트 케이스 선택 지연이 길어지는 단점을 가지며, 배제 후 선택되는 영역을 지정할 수 있는 기능을 가지지 않기 때문에 선택 빈도가 높은 영역이 연속적으로 선택되는 현상을 제어할 수 없고, 선택 빈도 차이를 유발하는 제한 영역의 형태가 미치는 영향에서 자유롭지 못하다는 제약점을 갖는다.

표 2. 배제 알고리즘
Table 2. A Selective Rejection Algorithm (SR).

D : 입력 도메인
P : 입력 도메인 분할 수준
M : 불린(Boolean) 매트릭스 (P *cells* × P *cells*)
CntCandCell : 제한 영역(F)이 아닌 M내의 셀 수
.
IP-ART 알고리즘의 다음 테스트 케이스 선택 과정에서
...
Step 1. if *CntCandCell* > 0 이면,
임의의 수 r ($0 < r \leq$ *CntCandCell*) 생성,
M에서 r번째 T인 셀 R 탐색
이 때, 셀 R = ($PosX$, $PosY$),
($1 \leq PosX \leq P$, $1 \leq PosY \leq P$)
else
M의 재생성 과정으로
Step 2. if ($P \leq 3$) → Step 4.로
Step 3. if ($PosX$ or $PosY$ = *1* or *3* or P-*2* or P)
임의의 숫자 *rej* 생성
if (*rej* ≤ 배제 확률 상수) → Step 1.으로
Step 4. R에 해당하는 D 내의 영역에서
임의의 테스트 케이스 *tc* 선택
Step 5. *tc*를 이용해 테스트를 수행
...

(2) 이동조건

이동 조건은 선택 빈도가 높은 영역의 파티션이 선택되는 경우, 이를 일정 확률로 선택 빈도가 낮은 영역으로 이동시킨다. 이동 조건 해당 여부는 파티션의 좌표 각각에 대해 검사되며, 이동 조건에 부합되는 경우 해당 좌표만 변환 대상이 되며, 이에 조건에 부합되지 않는 좌표는 그대로 이용하여, 선택 빈도가 낮은 영역에 고르게 분포될 수 있도록 한다. 이와 같은 방법은 적극적인 선택 빈도 조절이 가능하다는 특징을 갖지만, 이미 제한 영역으로 설정되어 있는 파티션으로의 이동 가능성도 있으며, 심한 경우, 이미 테스트 케이스가 선택된 파티션이 다시 선택될 수도 있기 때문에, 기존 테스트 케이스 인근에서 다시 테스트 케이스가 선택되어 기법의 성능에 부정적인 영향을 미칠 가능성을 제공한다. 하지만 이동 대상이 되는 영역의 제한 영역 설정 여부를 참조하여 이동을 결정하게 되면, 이 영역은 선택 빈도가 높은 파티션으로 둘러싸인 영역이므로, 이미 제한 영역으로 설정되어 있을 가능성이 높고, 따라서 이들 가운데 제한 영역으로 설정되지 않은 파티션을 선택하지 못하는 경우가 빈번하게 발생되게 된다. 따라서 본 논문에서는 이동 시 기존의 제한 영역 정보를 무시하고, 파티션 이동을 수행하도록 하며, 이과 같은 알고리즘은 표 3에 기술되어 있다.

이동 조건 알고리즘에서 파티션 이동의 기준이 되는 이동 확률 상수는 기존 IP-ART의 테스트 케이스 분포를 근거로 정의하는데, 각 파티션 분할 수준에서의 테스트 케이스 분포 특성을 따로 고려하여 정의할 것인가

표 3. 이동 알고리즘
Table 3. A Translated Partition Algorithm (TR and TRA).

```
D : 입력 도메인
P : 입력 도메인 분할 수준
M : 불린(Boolean) 매트릭스 (P cells × P cells)
CntCandCell : 제한 영역(F)이 아닌 M내의 셀 수

IP-ART 알고리즘의 다음 테스트 케이스 선택 과정에서
  ...
Step 1. if CntCandCell > 0 이면,
          임의의 수 r (0 < r ≤ CntCandCell) 생성,
        M에서 r번째 T인 셀 R 탐색
          이 때, 셀 R = (PosX, PosY),
            (1≤PosX≤P, 1≤PosY≤P)
        else
          M의 재생성 과정으로
Step 2. if ( P ≤ 3) → Step 4.로
Step 3. if ( PosX or PosY = 1 or 3 or P-2 or P )
          임의의 숫자 trn 생성
          if ( trn ≤ 이동 확률 상수 )
            해당 PosX or PosY := 2 or P-1 중
            임의의 값으로 설정
Step 4. R에 해당하는 D 내의 영역에서
        임의의 테스트 케이스 tc 선택
Step 5. tc를 이용해 테스트를 수행
  ...
```

(Adaptive Translated Partition, TRA), 혹은 전체 파티션 분할 수준에 대해 공통적인 상수(Translated Partition, TR)를 부여할 것인가에 따라 두 가지로 나뉜다.

(3) 제한 영역 관리 조건

제한 영역 관리 조건은 테스팅 수행 중 선택된 테스트 케이스가 위치한 파티션이 선택 빈도가 높은 영역 가운데 입력 도메인의 가장자리에 위치한 파티션인 경우, 즉 첫 번째 혹은 P번째 파티션인 경우에, 모든 인접 파티션을 제한 영역으로 설정하게 되면 인접한 곳에 선택 빈도가 낮은 영역이 형성되도록 영향을 주기 때문에, 선택 빈도가 높은 영역에 위치하는 파티션에 대해서만 제한 영역을 설정하도록 하는 방법이다. 이 조건은 선택 빈도가 높은 영역에 대해 배제나 이동 등의 방법을 통해 적극적으로 개입하는 방법이 아니기 때문에 기존의 테스트 케이스 분포를 일정 수준 따를 것으로 예상된다. 또한 제한 영역 관리 대상이 되는 영역에서

표 4. 제한 영역 제어 알고리즘
Table 4. A Restriction Area Management Algorithm (RAM).

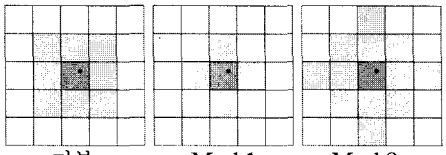
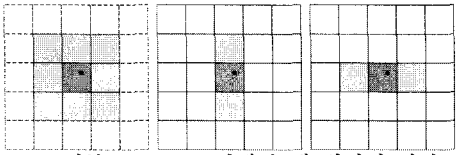
```
P : 입력 도메인 분할 수준
E : 이전 테스트 케이스 집합, E = {e1, e2, ... , ei}
    ei = {입력 데이터 Ixi, 입력 데이터 Iyi}
D : 입력 도메인,
    D = { 0 ≤ Ixi < MaxX, 0 ≤ Iyi < MaxY }
M : 불린(Boolean) 매트릭스 (P cells × P cells)
ei = {Ixi, Iyi} | ei ∈ E를 포함하는
    M의 셀의 좌표 = (PosX, PosY),
    PosX = ⌊Ixi × P / MaxX⌋,
    PosY = ⌊Iyi × P / MaxY⌋

IP-ART 알고리즘의 제한 영역 설정과정에서..
 ...
if PosX or PosY이 1 or P 이면
  AdjCell ∈ {(PosX-1 ≤ curPosX ≤ PosX+1, PosY-1 ≤
       curPosY ≤ PosY+1)}인
  모든 AdjCell at (curPosX, curPosY)에 대해,
  if curPosX or curPosY가 1 or P 이면
    if AdjCell이 T이면
        AdjCell := F, CntCandCell 수정
else
  AdjCell ∈ {(PosX-1 ≤ curPosX ≤ PosX+1, PosY-1 ≤
       curPosY ≤ PosY+1)}인
  모든 AdjCell at (curPosX, curPosY)에 대해,
  if AdjCell이 T이면,
    AdjCell := F, CntCandCell 수정
 ...
```

는 기존의 테스트 케이스 인근에서 새로운 테스트 케이스가 생성되는 현상도 발생 가능하다. 하지만 선택 빈도의 수준을 일부 제어가 가능하기 때문에, 선택 빈도 차이가 큰 환경에서는 효율성을 가질 수 있을 것으로 예상된다. 이 조건은 기존 테스트 케이스에 대한 제한 영역 설정 부분 및 수행된 테스트 케이스에 대한 제한 영역 설정 부분 모두에 적용된다.

4. 기법별 비교

IP-ART를 기초로 생성된 각 기법들은 '사용하는 제한 영역의 형태'와 '테스트 케이스 분포 조절을 위한 방법'에서 표 5에 기술된 내용과 같이 서로 차이점을 갖는다. 우선 EIP-ART와 M-IP-ART는 서로 다른 세 가지 형태의 제한 영역을 사용하는데, 일단 정의된 제한

표 5. 기법별 주요 사항 비교
Table 5. Comparison of Main Feature of Each Strategies.

기법	제한 영역 형태 (Test Case가 선택된 셀 이웃 셀)	테스트케이스 분포 조절
EIP-ART	 기본 Mask1 Mask2	제한 영역 형태에 따른 확장 지수 적용 (기본 = 3, Mask1 =1, Mask 2 = 2) + 제한 영역 형태에 따른 테스트 케이스 분포 확인
M-IP-ART	EIP-ART와 동일	제한 영역 형태에 따른 테스트 케이스 분포 확인
배제(SR)	IP-ART와 동일	IP-ART에서의 분포 기반, 배제 확률 상수 생성 적용
이동(TR/TRA)	IP-ART와 동일	IP-ART에서의 분포 기반, 이동 확률 상수 생성 적용
제한영역관리 (RAM)	 기본 선택된 셀 위치에 따라	선택된 셀의 위치에 따라 정의된 제한 영역 형태의 적용

영역은 테스트 수행 중간에 변경되지 않기 때문에, 사용되는 제한 영역의 형태에 따라 서로 다른 테스트 케이스 분포를 보인다. EIP-ART는 제한 영역의 변화와 더불어 입력 도메인 확장 지수를 추가 적용하는데, 이 확장 지수는 사용되는 제한 영역의 형태에 따라 1~3개의 셀 폭을 입력도메인의 가장자리에 가상 확장하도록 한다. 하지만 이들 모두에서 테스트 케이스 불균일 분포 수준에 대한 고려는 이루어지지 않고 있다.

본 논문에서 제안하는 '배제', '이동' 및 '제한 영역 관리' 기법 모두는 기본적으로 IP-ART의 제한 영역을 사용한다. 이들 중 '배제'와 '이동' 기법은 IP-ART에서 나타나는 테스트 케이스 분포 통계에 근거하여, 배제 확률 및 이동 확률을 정의하고, 이를 테스트 케이스 선택 빈도가 높은 셀에 적용한다. 이는 통계적으로 선택 빈도가 높은 셀이 선택되는 경우, 다시 선택하도록 하거나, 강제로 이동시키는 방법을 통하여 테스트 케이스 분포 조절을 시도하게 된다. '제한 영역 관리' 기법은 기본적으로는 IP-ART의 제한 영역 형태를 이용하되, 선택된 셀의 위치에 따라 제한 영역 형태가 변경되도록 하는데, 이는 선택 빈도가 낮은 셀의 선택 가능성을 높여 테스트 케이스 분포 조절이 이루어진다. 한편 입력 도메인의 가상 확장은 오랜 테스트 케이스 선택 지연을 유발하므로 제안된 모든 기법은 입력 도메인 확장을 사용하지 않는다.

IV. 성능 평가

1. 실험 환경

Ⅲ장에서 정의한 세 가지 테스트 케이스 분포 조절 조건이 IP-ART 기법의 성능과 테스트 케이스 분포에 미치는 영향 평가를 위한 실험은 표 6과 같은 파라미터를 이용하여, 다양한 오류 비율로 생성된 블록 패턴을

표 6. 성능 평가 실험 환경
Table 6. Simulation Settings for A Performance Check.

입력 도메인	한 입력 데이터의 변위 = 0~10000 (예 : 2차원 - 10000×10000) 단, EIP-ART의 변위 = 0~1000
입력 도메인 차원	2차원~4차원
오류 비율 (Θ)	0.01, 0.005, 0.002, 0.001, 0.0005
오류 패턴	블록
테스트 회수	10000

조건	파라미터
배제	배제 확률 상수 = 25
이동	• 고정 이동 확률 상수 = 25 • 적응적 이동 확률 상수 ▪ P=3 및 P≥6 : 25 ▪ P=4 : 38 ▪ P=5 : 19, 45

표 7. 테스트 케이스 분포 확인 실험 환경
Table 7. Simulation Settings for A TC Distribution Check.

입력 도메인	한 입력 데이터의 변위 = 0~10000 단, EIP-ART의 변위 = 0~1000
입력 도메인 차원	2차원
오류 비율 (Θ)	0.0005
오류 패턴	블록
테스트 회수	50000

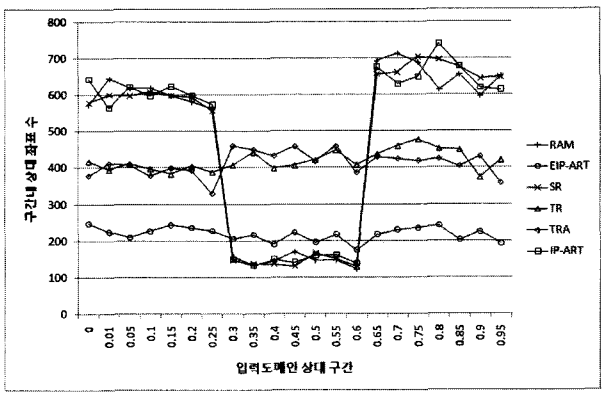
입력 도메인 내에 생성하여 진행되었다. 오류 패턴으로 이용되는 블록의 크기는 입력 도메인 크기와 오류 비율에 따라 결정되며, 블록의 형태는 블록을 구성하는 모든 변의 길이가 동일하도록 구성하였다.

정의된 조건에 대한 파라미터는 배제 조건의 경우 선택 빈도가 높은 곳과 낮은 곳의 분포 평균을 고려하여 배제 확률 상수로 25가 사용되었고, 이동 조건은 고정 이동 확률 상수와 적응적 이동 확률 상수로 나누어 정의하였으며, 고정 이동 확률 상수로는 25를, 적응적 이동 확률 상수로는 IP-ART의 테스트 케이스 분포를 기반으로 하여 파티션 분할 수준별 특징에 맞추어 각각 정의하였다.

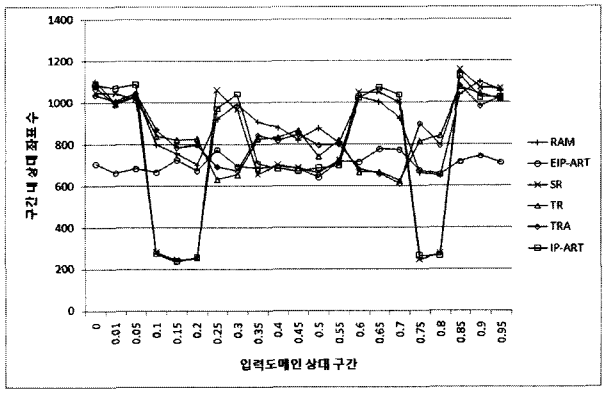
2. 실험 결과

그림 4는 IP-ART와 IP-ART에 조건을 부여한 정책 및 EIP-ART를 이용하여 2차원 입력 도메인, 오류 비율 0.0005에서 50000번의 테스팅을 수행하는 과정에서 입력 도메인 분할 수준이 각각 3, 8, 15, 30일 때 선택된 테스트 케이스 분포를 나타낸 것이다. 입력 도메인은 한 입력 데이터의 변위를 [0,1] 구간으로 변환하고, 이를 200개의 단위 구간으로 나눈 후, 각 단위 구간 내에 위치하는 테스트 케이스의 첫 번째 입력 데이터 좌표를 누적하여 나타낸 것이다. 두 번째 입력 데이터의 분포는 첫 번째 입력 데이터의 분포와 거의 동일하기 때문에, 테스트 케이스 분포 비교의 용이성을 위해 첫 번째 입력 데이터의 분포만을 사용하였다.

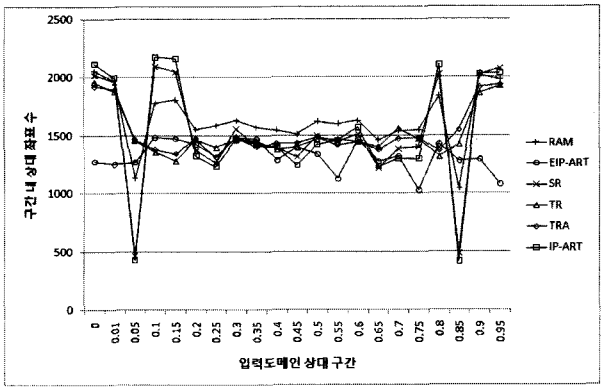
그림 4에 나타난 결과에 따르면 모든 실험 대상 기법들이 IP-ART와 유사하거나 낮은 테스트 케이스 분포 편차를 갖는다. 특히 IP-ART에서 고른 분포를 보이는 영역을 실제 영역으로 사용하는 EIP-ART는 전체 입력 도메인 분할 수준에서 상대적으로 고른 분포를 보이고 있으며, 이동 조건을 사용하는 TR과 TRA는 IP-ART



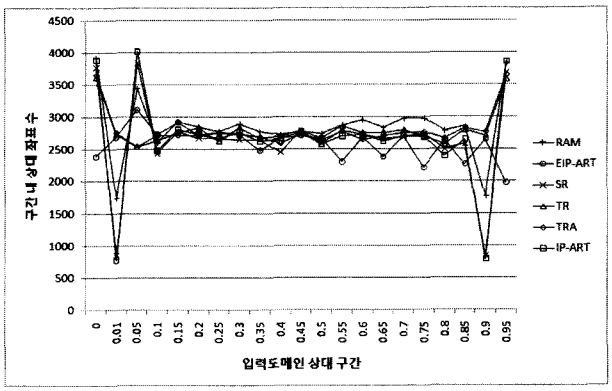
(a) P = 3



(b) P = 8



(c) P = 15



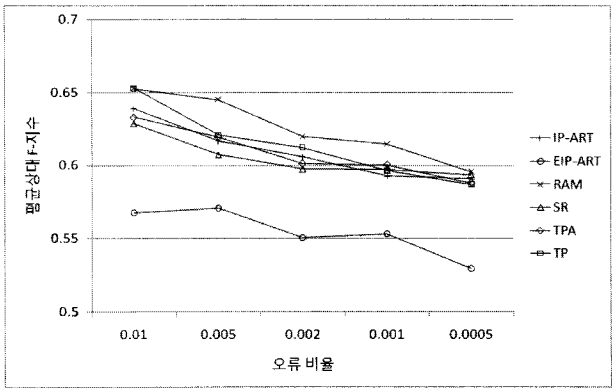
(d) P = 30

그림 4. 테스트 케이스의 첫 번째 입력 데이터 분포
Fig. 4. A Distribution of The First Input Data of TCs.

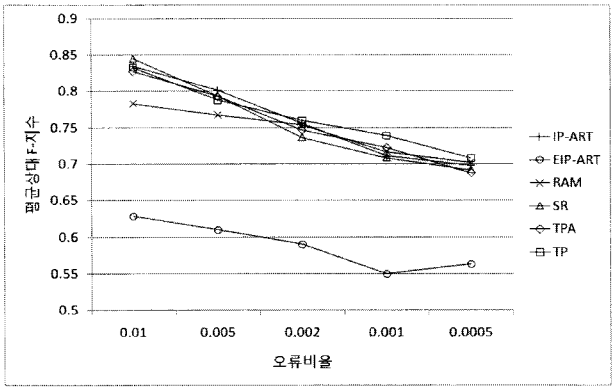
의 영역별 선택 빈도 차이를 상당 부분 감소시키고 있다. 하지만 배제 조건을 사용하는 SR은 파티션 분할 수준에 상관없이 IP-ART와 거의 유사한 수준의 분포를 보이는데, 이러한 결과는 배제 조건이 선택 빈도가 높은 영역의 파티션 선택을 일부 배제하더라도, 해당 지역에서 연속적으로 발생하는 파티션 선택을 충분히 제어할 수 없다는 점과 기존에 선택된 파티션 인근에 설정되는 제한 영역에 대한 대응책이 배제 조건에는 없기 때문인 것으로 파악된다. 한편 제한 영역 관리 조건을 사용한 RAM은 파티션 분할 수준이 낮을 때에는 IP-ART와 거의 유사한 수준의 분포를 보이지만, 분할 수준이 증가함에 따라 점차 영역별 분포 편차를 감소시키는 특징을 보이는데, 이는 분할 수준의 증가에 따라 입력 도메인 내에 제한 영역 관리 조건에 적용을 받는 파티션의 수가 증가되기 때문이다. 따라서 RAM은 낮은 차원의 입력 도메인이나 낮은 분할 수준에서 보다 높은 차원의 입력 도메인과 높은 분할 수준을 가지는 환경에서 즉, 오류 비율이 낮은 경우에 대한 적응력이 더 뛰어날 것으로 예측된다.

그림 5는 IP-ART와 EIP-ART 및 조건을 부여한 각 기법들의 성능을 F-지수를 이용해 비교한 것이다. 기존 논문의 실험 결과와 유사하게 기법들의 성능은 오류 비율이 감소할수록 좋아지고, 입력 도메인의 차원 수가 증가할수록 저하되는 것으로 나타났다.

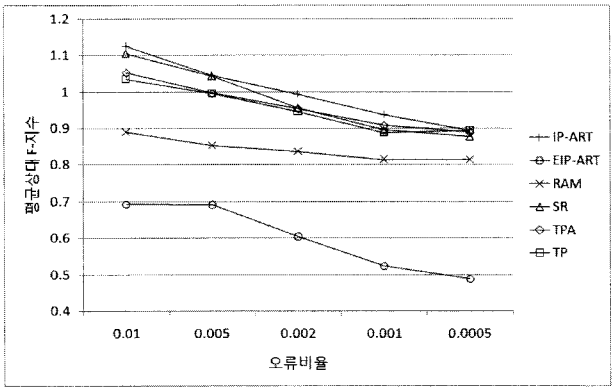
EIP-ART는 모든 차원의 입력 도메인에서 꾸준히 좋은 성능을 보이나, 조건을 반영한 다른 기법들은 IP-ART와 유사하거나 약간 향상된 성능을 보인다. 2차원 입력 도메인 환경에서는 IP-ART와 조건이 부여된 기법들 사이의 성능 차이는 평균 약 1~2% 수준의 편차를 보이는 정도로, 조건을 부여했을 때의 장점이 거의 나타나지 않았다. 하지만 입력 도메인의 차원 수가 증가할수록 부여된 조건의 영향이 나타나기 시작하는데, 이로 인해 3차원에서는 고정 이동 비율을 사용하는 방식인 TP를 제외하면 조건을 반영한 기법들의 평균이 IP-ART보다 좋아지며, 4차원에서는 모든 기법의 성능이 IP-ART보다 2~16% 향상된 것으로 나타났다. 입력 도메인 차원 수 증가에 따른 영향은 정의된 조건들 중 제한 영역 관리 조건을 사용하는 RAM에서 가장 크게 나타났는데, SR, TPA 및 TP가 2~3% 정도 수준인데 비해 RAM은 16% 향상된 성능을 보이고 있다. 이는 제한 영역 관리 조건에 의해 영향을 받게 되는 선택 빈도가 낮은 영역의 수가 입력 도메인의 차원 수가 증가함에 따라 빠르게 늘어나는데 그 원인이 있다.



(a) 2차원



(b) 3차원



(c) 4차원

그림 5. 오류 비율에 따른 평균 상대 F-지수 변화
Fig. 5. Mean Relative F-measures on various Failure Rates.

4차원 입력 도메인의 높은 오류 비율(>0.005)에서 EIP-ART와 RAM을 제외한 나머지 기법들의 F-지수는 1보다 큰 값을 갖는다. 이는 순수 랜덤 테스팅을 사용하는 경우보다 더 많은 수의 테스트 케이스를 사용해야 오류 영역이 검출될 수 있음을 의미한다. 따라서 EIP-ART와 같이 입력 도메인 내에 고른 테스트 케이스 선택 분포를 갖는 기법이나, RAM과 같이 고차원에

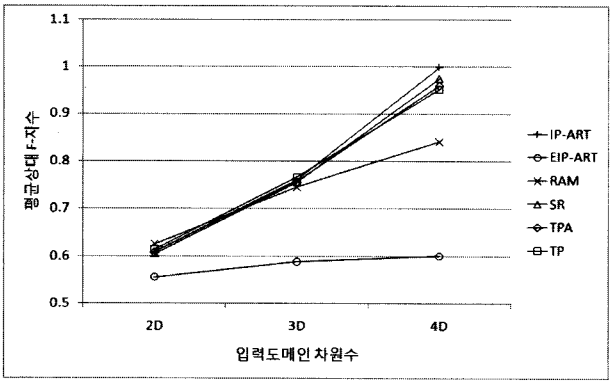


그림 6. 입력 도메인 차원 수 증가에 따른 평균 상대 F-지수 변화

Fig. 6. Mean Relative F-measures on each Input Domain Dimensions.

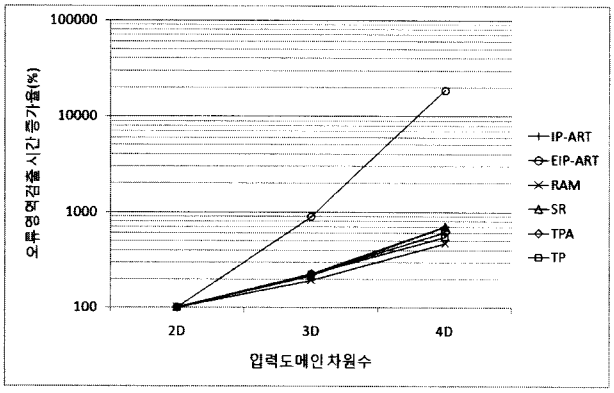


그림 7. 오류 영역 검출 시간 비용 증가율

Fig. 7. Time Cost to Detect Failure Region on each Input Domain Dimensions.

서 나타나는 현상에 적응성을 가지는 기법이 아닌 경우, 4차원 이상에서는 성능을 보장할 수 없음을 의미한다.

그림 6은 입력 도메인 차원 수의 증가에 따라 각 기법이 보이는 평균 상대 F-지수의 변화율을 나타낸 것이다. IP-ART가 차원 수 증가로부터 받는 영향이 제시된 기법 중 가장 크고, EIP-ART가 가장 작은 것으로 나타났다. IP-ART에 조건을 부여한 기법들은 대부분 입력 도메인 차원 수 증가로부터 IP-ART와 유사한 영향을 받으며, 제한 영역 관리 조건을 사용한 RAM만 상대적으로 다른 조건을 부여한 것들에 비해 입력 도메인 차원 수 증가에 따른 영향이 줄어들었다.

그림 7은 입력 도메인 차원 수 증가에 따른 각 기법의 오류 영역 검출 소요 시간 증가율을 비교한 것이다. 오류 비율이 0.01인 입력 도메인의 차원 수를 증가시켜 가며 각 기법별로 1000회의 오류 영역 검출을 수행하고, 이를 이용해 하나의 오류 영역 검출에 소요되는 평균 시간을 구한 후, 이를 2차원 입력 도메인에서의 오류 검출 요구 시간 기준 증가율로 나타낸 것이다. 결과에 따르면 배제를 제외한 다른 조건을 이용한 기법은 입력 도메인 차원 수 증가에 따른 오류 영역 검출 시간 증가율이 IP-ART보다 약간 낮은 수준이지만, 대체로 IP-ART와 유사한 증가율을 보인다. EIP-ART의 경우, 실험을 수행한 모든 입력 도메인 차원 수에서 다른 기법보다 월등한 성능을 보이기는 하지만, 입력 도메인 차원 수 증가에 따라 오류 영역 검출 시간이 급격하게 증가하여 4차원 입력 도메인에서 IP-ART의 63배 이상을 요구한다. 이러한 급격한 시간 증가의 원인은 앞서 기술한 것과 같이 EIP-ART 기법 적용 시, 실제 입력 도메인 내부보다 가상 확장된 영역에 대한 선택 빈도가 더 높아 실제 테스트 케이스 생성에 더 많은 시간을 요구하고, 확장 영역에서 선택된 가상 테스트 케이스들에 대한 부수적인 관리 비용이 입력 도메인 차원 수 증가에 따라 빠르게 증가하기 때문인데, 이와 같은 EIP-ART가 가지는 단점은 EIP-ART의 고차원 입력 도메인 환경 적용에 제약으로 작용한다. 한편 가장 짧은 시간을 요구하는 기법은 RAM이고, 이는 IP-ART의 오류 검출 시간을 기준으로, 3차원 입력 도메인에서 약 92%, 4차원 입력 도메인에서 약 72% 수준의 시간을 요구하는 것으로 나타났으며, 실험에 사용된 오류 비율이 높은 것을 고려하면 전체 환경에 대한 요구 시간은 훨씬 큰 폭으로 줄어들 것으로 예상된다. 이는 RAM의 테스트 케이스 생성 시간이 상대적으로 짧기 때문인데, RAM의 경우 선택 빈도가 높은 영역 주위에 제한 영역을 설정하지 않고, 또 그 수가 입력 도메인 차원 수 증가에 따라 늘어나, 선택 가능한 파티션의 수가 상대적으로 늘어나는데 원인이 있다.

소프트웨어의 개발 과정 중 테스팅에 주어지는 시간은 제한된 자원이므로, 테스트 케이스 선택 정책의 효율성 평가는 각 기법의 오류 검출 성능뿐만 아니라 오류 검출 속도 즉, 단위 시간 당 수행 가능한 테스트 회수와 이를 통해 검출된 오류의 개수와 함께 고려되어야 한다. 즉, 테스트 케이스 선택 정책은 우수한 오류 검출 성능과 빠른 수행 속도 모두를 동시에 만족시켜야 한다고 할 수 있다. 이에 따라 각 기법을 사용했을 때, 단위 시간당 검출 가능한 오류 영역의 수를 상대적 기준으로 비교 평가를 수행하였으며, 그 결과가 그림 8에 나타나 있다.

그림 8은 오류 비율이 0.01인 입력 도메인에서, 각 기법을 일정 단위 시간동안 사용하는 것을 가정했을

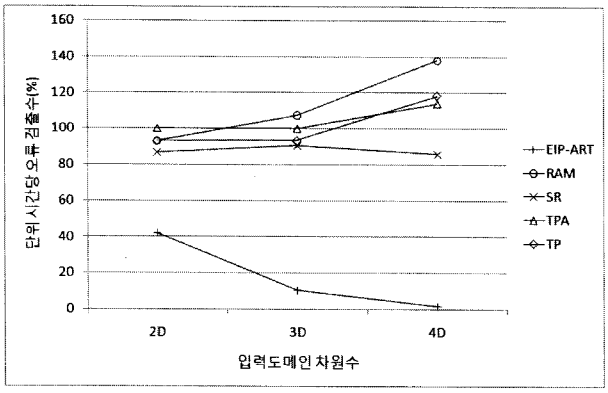


그림 8. IP-ART 대비 단위 시간당 오류 검출 수(%)
Fig. 8. Number of Failure Detections during a Unit Time(%)

때 검출 가능한 오류의 수를 IP-ART를 사용한 경우에 대한 비율로 나타낸 것이다. 이에 따르면 EIP-ART의 경우 F-지수는 타 기법에 비해 탁월한 우수성을 보였으나, 매우 오랜 수행 시간을 요구하기 때문에 차원별로 IP-ART보다 약 42~2% 수준의 아주 낮은 효율성을 보이고 있으며, SR을 제외한 제안 기법들은 IP-ART와 유사하거나 우수한 효율성을 보이는 것으로 나타났다. SR은 F-지수만을 보면 IP-ART와 유사한 성능을 갖지만, 배제 조건으로 인한 테스트 케이스의 재생성에 요구되는 시간 지연으로 인해 IP-ART의 88% 수준의 낮은 효율성을 갖는 것을 알 수 있다. 한편, TP와 TPA는 2차원과 3차원 입력 도메인에서 IP-ART와 유사하거나 조금 낮은 효율성을, 4차원 입력 도메인에서는 각각 18~14% 정도 향상된 성능을 보이는데, 이는 이들 기법이 IP-ART와 유사한 F-지수를 갖기는 하지만, 입력 도메인의 차원 수가 증가할수록 IP-ART와의 F-지수 격차 및 오류 검출 속도 차이가 커지면서, 이들이 기법의 효율성을 증가시키는 기능을 하기 때문이다. 제안된 기법 가운데 상대적으로 가장 빠른 수행 속도와 우수한 F-지수를 보였던 RAM은 2차원 입력 도메인을 제외하면, 모두 IP-ART보다 약 7~38% 향상된 효율성을 보이며, 전체 입력 도메인 차원을 고려하면 평균 13% 수준의 향상된 효율성을 갖는 것으로 나타났다. 따라서 SR를 제외한 모든 제안 기법들이 4차원 입력 도메인에서 IP-ART보다 우수한 효율성을 보여, 고차원 입력 도메인에 대한 제안 기법의 적용 가능성을 확인할 수 있었다.

V. 결 론

IP-ART는 초기 ART 기법의 단점인 테스트 케이스 선택에 필요한 거리 연산을 입력 도메인 반복 분할 방식을 이용하여 효율적으로 개선하였고, EIP-ART는 IP-ART가 가지는 테스트 케이스 분포 불균일 현상을 완화하여 다양한 환경에서 기존 기법에 비해 우수한 성능을 보인다. 하지만, EIP-ART는 확장된 가상 입력 도메인의 영향으로 테스트 케이스 선택에 오랜 시간을 요구하는 단점을 가지고 있다. 따라서 본 논문에서는 IP-ART를 바탕으로, 테스트 케이스 분포에 영향을 줄 수 있는 세 가지 조건을 정의하고, 이들 각각이 기법에 미치는 성능을 비교 평가하였다.

실험 결과에 따르면 F-지수를 기준으로 비교할 때, 정의된 세 가지 조건 중 배제와 이동 조건은 IP-ART와 거의 유사하거나 높은 차원의 입력 도메인에서 약간 우수한 성능을 보였고, 제한 영역 관리 조건은 다른 조건에 비해 테스팅 환경으로부터 받는 영향이 가장 작아 안정적이며, IP-ART의 성능을 4차원 입력 도메인에서 약 15% 정도 향상시키는 것으로 나타났다. 또한 배제 조건을 제외한 나머지 두 조건을 이용했을 때의 테스트 케이스 분포도 IP-ART를 이용한 경우보다 상당 수준 완화되기는 하였으나, 테스트 케이스 분포 불균일 현상을 EIP-ART 수준으로 제어하지는 못하고 있고, 조건에 따라 기존 테스트 케이스 인근에서 새로운 테스트 케이스가 선택될 가능성을 가지고 있어, 정의된 세 조건 중 어느 것도 EIP-ART 수준의 테스트 케이스 분포와 성능을 보이지 못했다.

하지만, 오류 영역 검출 성능과 검출 속도를 종합적으로 고려해 각 기법의 효율성을 살펴보면, EIP-ART는 오랜 오류 검출 소요 시간으로 인해 상대적으로 아주 낮은 효율성을 보이며, 제안 기법 중 가장 좋은 효율성을 보이는 RAM은 IP-ART 대비 평균 13% 수준의 효율성 향상을 확인하였다.

참 고 문 헌

[1] J. Regehr, "Random Testing of Interrupt-Driven Software", in Proceedings of the 5th ACM international conference on Embedded software, pp. 290-298, Jersey City, NJ, USA, Sep. 2005.
[2] T. Yoshikawa, K. Shimura, T. Ozawa, "Random program generator for Java JIT compiler test

system", in Proceedings of the 3rd International Conference on Quality Software, pp. 20-23, Dallas, Texas, USA, Nov. 2003.
[3] G. J. Myers, *The Art of Software Testing*, 2nd ED., John Wiley & Sons, 2004.
[4] T. Chen, T. Tse, Y. Yu, "Proportional sampling strategy: a compendium and some insights", *The Journal of Systems and Software*, vol. 58, pp. 65-81, Aug. 2001.
[5] J. Duran and S. Ntafos. "An Evaluation of Random Testing", *IEEE Transactions on Software Engineering*, vol. 10, no. 4, pp. 438-444, Jul. 1984.
[6] D. Hamlet and R. Taylor. "Partition Testing Does Not Inspire Confidence", *IEEE Transactions on Software Engineering*, vol. 16, issue 12, pp. 1402-1411, Dec. 1990.
[7] 신승훈, 박승규, "입력 도메인 확장을 이용한 반복 분할 기반의 적응적 랜덤 테스팅 기법", *정보처리학회논문지D*, vol. 15D, no. 4, pp. 531-540, 2008년 8월.
[8] 이광규, 신승훈, 박승규, "반복 분할 기반의 적응적 랜덤 테스팅 향상 기법", *전자공학회논문지*, 제 45권 CI편 제5호, pp. 180-191, 2008년 9월.
[9] K. P. Chan, T. Y. Chen, and D. Towey, "Restricted Random Testing", in Proceedings of the 7th European Conference on Software Quality, vol. 2349 of LNCS, pp. 321-330, Springer, Helsinki, Finland, Jun. 2002.
[10] 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, pp. 79-86, Braunschweig, Germany, Sep. 2004.
[11] 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, vol. 4006 of LNCS, pp. 155-166. Springer-Verlag, Porto, Portugal, Jun. 2006.
[12] T. Y. Chen, H. Leung, and I. K. Mak, "Adaptive Random Testing", in Proceedings of the 9th Asian Computing Science Conference, vol. 3321 of LNCS, pp. 320-329. Springer, Chiang Mai, Thailand, Dec. 2004.
[13] J. Mayer, "Adaptive Random Testing with Randomly Translated Failure Region", in Proceedings of the 1st International Workshop on Random Testing, pp. 70-77, Portland, Maine, USA, Jul. 2006.
[14] J. Mayer, C. Schneckenburger, "Adaptive Random Testing with Enlarged Input Domain", in Proceedings of the Sixth International Conference on Quality Software, pp. 251-258, Beijing, China, Aug. 2006.

저 자 소 개

신 승 훈(정회원)
2000년 아주대학교 정보 및 컴퓨터공학부 학사 졸업
2002년 아주대학교 정보통신 공학과 석사 졸업
2004년 아주대학교 정보통신 공학과 박사 수료
<주관심분야 : 소프트웨어 테스팅, 테스트 자동화, 멀티미디어 전송 정책>

박 승 규(평생회원)
1974년 서울대학교 응용수학과 학사 졸업
1976년 한국과학원 (KAIST) 전산학과 석사 졸업
1982년 Institut National Polytechnique de Grenoble 전산학과 박사 졸업
1976~1977 한국과학기술연구소(KIST) 연구원
1982~1983 KIET (현ETRI) 실장/선임연구원
1984~1985 미국 IBM 왓슨연구소 연구원
1985~1992 ETRI 연구위원/책임연구원
1992~현재 아주대학교 정보통신대학 교수
<주관심분야 : 임베디드 테스팅, 자가 컴퓨팅/치료 시스템, 차세대 컴퓨터 구조>

정 기 현(정회원)
1984년 서강대학교 전자공학과 학사 졸업
1988년 미국 Ilinois주립대 EECS 석사 졸업
1990년 미국 Purdue대학 전기 전자공학부 박사 졸업
1991년~1992년 현대반도체 연구소.
1993년~현재 아주대학교 전자공학부교수.
<주관심분야 : 컴퓨터구조, VLSI 설계, 멀티미디어 및 실시간 시스템>