

테스트 케이스 분포 조절을 통한 IP-ART 기법의 성능 향상 정책

(Improving Performance of ART with Iterative Partitioning using Test Case Distribution Management)

신승훈[†] 박승규^{**} 최경희^{**}
(Seung-Hun Shin) (Seung-Kyu Park) (Kyung-Hee Choi)

요약 적응적 랜덤 테스트(Adaptive Random Testing, ART)은 테스트 케이스의 효율적인 선택을 통해 순수 랜덤 테스트(Random Testing, RT)보다 더 적은 수의 테스트 케이스를 이용해 입력 도메인 내의 오류를 찾는 것을 목적으로 한다. ART 기법 중 하나인 입력 도메인 반복 분할 기법(ART through Iterative Partitioning, IP-ART)은 초기 ART 기법의 단점인 많은 연산량을 입력 도메인 분할에 의해 효율적으로 개선되도록 하였으며, 입력 도메인 확장을 이용한 IP-ART(IP-ART with Enlarged Input Domain, EIP-ART)는 IP-ART의 테스트 케이스 분포 특징을 이용하여 추가적인 성능 향상과 확장성을 제공하였다. 하지만 EIP-ART는 입력 도메인 확장에 따라 발생하는 부하로 인해 테스트 케이스 생성에 오랜 시간을 요구하기 때문에 이의 개선이 필요하다. 따라서 본 논문에서는 두 가지의 추가 부하를 유발하지 않는 테스트 케이스 분포 조절 기법을 제안하고, 이들의 성능 개선 가능성을 실험을 통해 확인하였으며, 실험 결과, 제안된 두 기법 중 입력 도메인 타일링 기법이 모든 환경에서 더 우수한 성능 및 확장성을 갖는 것으로 확인되었다.

키워드 : 랜덤 테스트, 적응적 랜덤 테스트, 테스트 케이스 선택

Abstract The Adaptive Random Testing(ART) aims to improve the performance of traditional Random Testing(RT) by reducing the number of test cases to find the failure region which is located in the input domain. Such enhancement can be obtained by efficient selection algorithms of test cases. The ART through Iterative Partitioning(IP-ART) is one of ART techniques and it uses an iterative input domain partitioning method to improve the performance of early-versions of ART which have significant drawbacks in computation time. And the IP-ART with Enlarged Input Domain(EIP-ART), an improved version of IP-ART, is known to make additional performance improvement with scalability by expanding to virtual test space beyond real input domain of IP-ART. The EIP-ART algorithm, however, have the drawback of heavy cost of computation time to generate test cases mainly due to the virtual input domain enlargement. For this reason, two algorithms are proposed in this paper to mitigate the computation overhead of the EIP-ART. In the experiments by simulations, the tiling technique of input domain, one of two proposed algorithms, showed significant improvements in terms of computation time and testing performance.

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

[†] 학생회원 : 아주대학교 정보통신공학과
sihsh@ajou.ac.kr
^{**} 종신회원 : 아주대학교 정보 및 컴퓨터공학부 교수
sparky@ajou.ac.kr
khchoi@ajou.ac.kr
논문접수 : 2008년 11월 12일
심사완료 : 2009년 4월 6일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지: 소프트웨어 및 응용 제36권 제6호(2009.6)

1. 서론

랜덤 테스트(Random Testing)은 균일 분포(uniform distribution)를 따르도록 테스트 케이스를 생성하고, 이를 이용해 테스트를 수행하는 블랙박스 테스트의 한 정책이다. 따라서 랜덤 테스트는 테스트 대상이 되는 소프트웨어에 대한 사전 정보 없이 테스트 케이스를 생성하기 때문에, 기능적(functional) 혹은 구조적(structural) 테스트에 대한 상대 개념으로 이해되어 왔으며[1], 이러한 이유로 랜덤 테스트의 효율성을 회의적으로 바라보는 시각도 있다[2]. 하지만 랜덤 테스트가 제공하는 테스트 케이스 생성 알고리즘 작성의 용이성 및 생성된 테스트 케이스 간의 독립성은 각각 방대한 양의 테스트의 정의와 테스트 결과를 바탕으로 한 소프트웨어의 신뢰도 및 통계치 추정을 가능하도록 하기 때문에[1], 랜덤 테스트는 다양한 애플리케이션의 테스트에 성공적으로 적용되어 왔으며[3-5], 이와 더불어 랜덤 테스트의 효율성 검증과[6-8] 효율적인 랜덤 테스트 케이스 생성을 통한 소프트웨어 오류 검출 성능 향상을 위한 노력[9-13] 또한 계속되어 왔다.

효율적인 랜덤 테스트 케이스 생성 기법 중 하나인 적응적 랜덤 테스트(Adaptive Random Testing, ART)은 소프트웨어 내의 오류가 군집하여 입력 도메인 내에 위치한다는 특성을 이용하여, 새롭게 선택되는 테스트 케이스가 이전 테스트 케이스 인근에서 선택되는 것을 방지하고, 선택된 테스트 케이스가 입력 도메인 내에 가능한 고르게 분포되도록 하여, 소프트웨어 내의 오류를 순수 랜덤 테스트보다 효율적으로 찾아낼 수 있도록 한다[9-13]. 이와 같은 테스트 케이스의 선택을 위하여 초기의 ART 기법은 테스트 케이스 선택에 테스트 케이스 간 거리를 이용하였고, 이를 통해 순수 랜덤 테스트에 비해 최대 50% 수준의 향상된 오류 검출 성능을 보였다[11]. 하지만 초기의 ART 기법들은 테스트가 진행될수록 요구되는 거리 연산량이 늘어나고, 또한 선택된 테스트 케이스가 입력 도메인 내의 특정 위치에 더 밀집되는 특성을 보이는데[13], 이는 각각 제한적인 확장성과 입력 도메인 차원 수 증가에 따른 성능 저하 등의 단점을 유발한다. 이러한 단점의 보완을 위해 입력 도메인의 반복 분할 기법(ART through Iterative Partitioning, IP-ART)이 제안 되었으며, IP-ART는 테스트 케이스 간 거리 연산이 아닌 입력 도메인 분할과 테스트 케이스 선택 제한 구역의 설정을 통해 테스트 케이스 생성에 필요한 연산량을 상당부분 줄였다[10]. 하지만 IP-ART 기법도 테스트 케이스의 입력 도메인 내 분포 불균일 현상은 충분히 제어하지 못했으며, 이는 기법의 확장성에 제약 요소가 되어, 이를 해결하기 위해

IP-ART 고유의 입력 도메인 내 테스트 케이스 분포에 근거한 입력 도메인 확장 기반 IP-ART 기법(IP-ART with Enlarged Input Domain, EIP-ART)[14]과 IP-ART의 제한 영역 형식에 변화를 주는 기법(Modified IP-ART, M-IP-ART)[15] 등이 제안되었다. 우선 EIP-ART는 입력 도메인 차원 수 증가에 대한 확장성 및 우수한 성능을 보장하지만, 확장된 입력 도메인에 수반되는 작업 부하가 입력 도메인 차원 수 증가에 따라 급격히 증가하여, 고차원 입력 도메인에서는 테스트 케이스 생성에 오랜 시간을 요구하며, M-IP-ART는 IP-ART에서 나타나는 분포 불균일 현상을 완화시키긴 하였지만 충분한 수준의 균일성을 이끌어내지 못해, 입력 도메인 차원 수 증가에 따른 확장성이 제한적이기 때문에 개선을 필요로 한다. 이에 따라 본 논문에서는 IP-ART 고유의 특성을 분석하고, 그 결과를 기반으로 입력 도메인의 확장 등의 추가 부하를 필요로 하지 않는 IP-ART 개선 방안을 제시하고, 이들이 IP-ART의 성능과 테스트 케이스의 입력 도메인 내 분포에 미치는 영향을 실험을 통해 비교 분석한다.

2. 정의 및 관련 연구

2.1 정의

한 소프트웨어에 대한 유한 크기의 입력 데이터 집합을 입력 도메인이라 하고, 입력 도메인 내에 존재하는 전체 입력 데이터 중 소프트웨어의 오류를 유발하는 데이터의 비율을 오류 비율(θ)이라 정의한다. 또한 소프트웨어의 오류를 유발하는 결함들은 입력 도메인 내에서 군집을 이루어 오류 영역을 형성하며, 이 오류 영역은 몇 가지 유형을 가지고 입력 도메인 내에 존재하는 성향을 가지는데 이를 오류 패턴이라고 한다. 오류 패턴은 크게 블록(Block), 스트립(Strip) 및 포인트(Point) 등의 세 가지로 분류되고[16], 이 중 블록 패턴과 스트립 패턴이 상대적으로 빈번하게 나타나는 오류 패턴이기 때문에[17], 기존 ART 관련 연구에서는 블록 패턴을 중심으로 제안된 기법의 성능을 평가하였으며, 본 논문에서도 블록 패턴을 대상으로 기법의 성능을 평가한다.

F-지수(F-measure)는 소프트웨어 테스트 수행 중 첫 번째 오류를 발견할 때까지 소요된 테스트 케이스의 수를 의미하는데[11], 소프트웨어에 대한 테스트를 수행하는 과정에서 오류가 발견되었을 때, 이에 대한 원인 파악 및 조치가 수행되는 소프트웨어 테스트 환경을 고려할 때, F-지수는 테스트 케이스 생성 기법의 성능에 대한 직관적인 평가 기준이라고 할 수 있다. 하지만 F-지수는 입력 도메인 내의 오류 비율에 직접적인 영향을 받기 때문에 그 자체로 사용하기는 어렵다. 따라서 순수 랜덤 테스트의 성능을 기준으로 하는 상대 F-지수를 이

용하여 기법의 성능을 평가한다. 입력 도메인의 내의 오류 비율이 θ 일 때, 이에 대한 순수 랜덤 테스트의 이론적 F-지수는 $F_{RT}=1/\theta$ 가 되고, 다른 기법들은 F_{RT} 에 대한 상대 수치인 평균 상대 F-지수(Mean Relative F-Measure)로 성능을 평가하게 되며, 평균 상대 F-지수는 다음과 같이 정의된다.

$$\text{평균 상대 F-지수} = (\text{평균 } F_{ART}) / F_{RT}$$

따라서 평균 상대 F-지수는 특정 기법을 사용했을 때 구해지는 F-지수 평균값과 순수 랜덤 테스트를 사용했을 때의 F-지수 비율이 된다. 즉, 평균 상대 F-지수가 1인 경우, 이 기법이 순수 랜덤 테스트와 동일한 성능을 가짐을 의미하게 되고, 평균 상대 F-지수가 1보다 클수록 낮은 성능을, 0에 가까울수록 높은 성능을 가짐을 의미한다. 본 논문에서는 이 평균 상대 F-지수를 이용하여 기존 기법과 제안 기법 간의 성능 비교 평가를 수행한다.

2.2 관련 연구 및 문제점

초기 ART 기법은 새로운 테스트 케이스의 생성에 테스트 케이스 간 거리를 이용하기 때문에, 테스트가 진행될수록 연산량이 늘어나는 단점을 가진다. 이와 같은 거리 연산을 배제하기 위해 제안된 IP-ART는 입력 도메인을 $n \times n$ 파티션으로 분할하고, 분할된 파티션 중 임의의 파티션에서 임의의 데이터를 선택하여 테스트 케이스로 이용한다. 이 과정에서 IP-ART는 이전의 테스트 케이스가 위치한 파티션과 해당 파티션을 둘러싸는 파티션은 제한 영역(Restriction Area)으로 설정하고, 제한 영역에서는 테스트 케이스가 선택될 수 없도록 하여 이전 테스트 케이스 인근에서 새로운 테스트 케이스가 선택되는 것을 방지하며, 그림 1과 같은 형태로 동작한다[10].

IP-ART는 최초, 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 오류가 발견되지 않은 경우, 해당 파티션을 제한 영역으로 설정한다. 이때, 입력 도메인 내에 더 이상 선택 가능한 파티션이 없

기 때문에, 분할 수준을 1 증가시켜 입력 도메인을 분할하고($P=2$), 이전 테스트 케이스를 포함하는 파티션과 그 이웃 파티션을 제한 영역으로 설정한다(그림 1(b)). $P=2$ 인 경우에는 항상 모든 파티션이 제한 영역에 포함되므로 입력 도메인을 재분할하고($P=3$), 이전 테스트 케이스를 포함한 파티션과 그 이웃 파티션을 제한 영역으로 설정한다(그림 1(c)). 이후 분할 영역 중 제한 영역으로 설정되지 않은 파티션 하나를 임의로 선택하고, 해당 파티션 내에서 임의의 데이터를 선택하여 다음 테스트 케이스(2)로 이용하고(그림 1(d)), 선택된 테스트 케이스를 이용한 테스트에서 오류가 발견되지 않으면, 테스트 케이스가 선택된 파티션과 그 이웃 파티션을 제한 영역으로 지정한 후(그림 1(e)), 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 배제하여, 테스트 케이스 생성에 요구되는 시간 비용을 감소 시켰으며, 성능은 초기 ART 기법 수준으로 유지한다[10].

하지만 IP-ART 기법에서도 그림 2와 같은 IP-ART 고유의 테스트 케이스 분포 불균일 현상을 찾아볼 수 있는데, 이는 기법의 확장성에 대한 장애 요소로 작용하게 된다[14,15]. 그림 2는 IP-ART를 이용해 오류영역을 찾아내는 작업을 50,000회 반복 수행하는 동안, 입력 도메인 분할 수준이 각각 $P=8$, $P=15$ 일 때의 파티션 선택 빈도 분포 가운데 입력 도메인의 가장자리에서 첫 번째(X1)부터 다섯 번째(X5)까지의 라인에 위치한 파티션에 대한 누적 선택 수를 나타낸다. 이에 따르면 테스트 케이스는 입력 도메인 가장자리를 둘러싸는 첫 번째(X1)와 세 번째 라인(X3)에 위치한 파티션에서 높은 빈도로 선택되며, 네 번째 라인의 안쪽 영역(X4~)은 전체 평균 선택 빈도와 유사하거나 약간 낮고, 외곽 영역에 비해 상대적으로 고른 선택 빈도를 보인다. IP-ART에 의해 생성되는 테스트 케이스 분포의 문제점은 첫 번째 라인(X1)과 세 번째 라인(X3) 및 이 라인들의 교차점

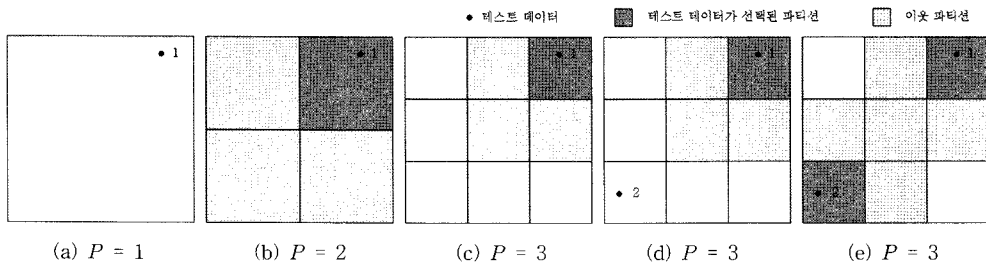
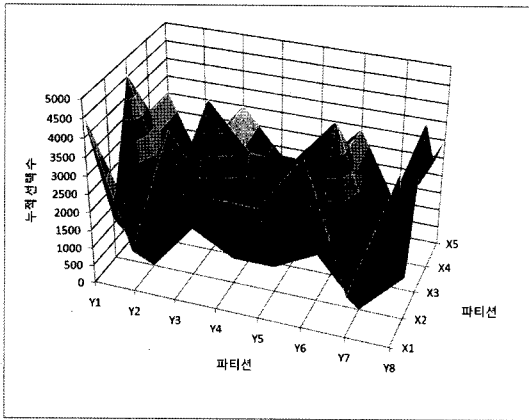
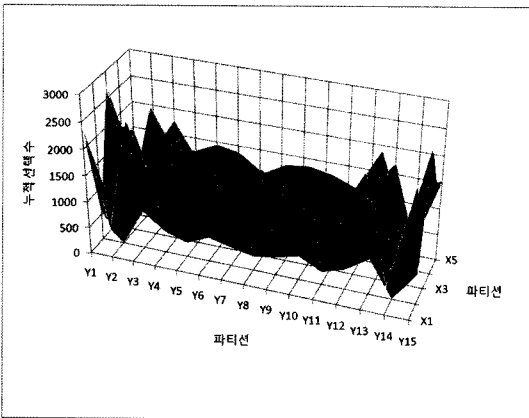


그림 1 IP-ART 수행 예



(a) $P = 8$



(b) $P = 15$

그림 2 테스트 케이스의 분포 - IP-ART

(예, $\{X1, Y1\}, \{X1, Y3\}, \dots$)에 위치하는 파티션에 대한 선택 빈도가 다른 파티션에 대한 선택 빈도보다 훨씬 높다는 것과, 이로 인해 두 번째 라인($X2$) 및 두 라인의 교차점(예, $\{X2, Y2\}, \{X2, Y7\}$)에 제한 영역이 높은 빈도로 생성되어 해당 영역에서의 테스트 케이스 선택 빈도가 매우 낮아진다는 점이다. 특히 두 번째 라인의 교차점, $\{X2, Y2\}, \{X2, Y7\}$ 에서는 단 하나의 테스트 케이스 선택도 이루어지지 않아, 이와 같은 문제의 해결이 요구된다.

IP-ART가 가진 이와 같은 문제의 해결을 위해 입력 도메인 확장 기반 IP-ART 기법(IP-ART with Enlarged Input Domain, EIP-ART)이 제안되었는데, EIP-ART는 입력 도메인을 가상 확장하여, 테스트 케이스 분포 불균일 현상이 보이는 영역을 실제 입력 도메인 둘레에 가상 확장한 영역에 위치하도록 하여, 실제 입력 도메인에서는 상대적으로 고른 테스트 케이스 분포를 얻을 수 있도록 한 기법이다[14]. IP-ART에 대해 EIP-ART 기법을 적용하는 경우에는 IP-ART의 테스트 케

이스 분포를 고려하여, 한 파티션 넓이의 세 배에 해당하는 확장 영역(확장 지수 = 3)을 실제 입력 도메인 가장자리에 위치하도록 하고, 이를 통해 테스트 케이스 분포 불균일 현상이 가장 확장된 입력 도메인 영역에서 발생되도록 유도한다. 이와 같은 방법으로 EIP-ART는 실제 입력 도메인 내에서는 상대적으로 고른 테스트 케이스 분포를 보이도록 하여, 고차원 입력 도메인에서도 우수한 성능을 보이는 확장성을 갖는다. 하지만 입력 도메인의 가상 확장으로 인해 실제 입력 도메인에 위치한 파티션에 대한 선택이 상대적으로 줄어들게 됨에 따라(2차원 입력 도메인, $P=15$ 일 때, 33%) 실제 테스트 케이스 선택에 더 많은 시간을 필요로 하게 되고, 또한 실제 도메인이 확장 영역을 포함한 전체 도메인에서 차지하는 비율이 입력 도메인 차원 수(D)가 증가할수록 $(P/(2 * \text{확장 지수} + P))^D$ 로 급격하게 줄어들기 때문에, 확장 입력 도메인에 위치하는 가상 테스트 케이스에 대한 관리 비용 뿐만 아니라, 테스트 케이스 선택에 요구되는 시간 비용이 입력 도메인 차원 수가 증가할수록 급격하게 늘어나는 단점을 유발한다.

그림 3은 입력 도메인 차원 수 증가에 따른 IP-ART와 EIP-ART의 오류 영역 검출 시간 증가율을 나타내고 있다. 오류 비율이 0.01인 입력 도메인에서 총 1000회의 테스트를 통해 두 기법이 오류영역을 찾아내는 데 소요된 평균 시간을 구하고, 2차원 입력 도메인에서 소요된 시간을 기준으로 입력 도메인 차원 수 변화에 따른 오류 영역 검출 시간 증가율을 나타낸 것이다. 그림 3에 나타난 것과 같이 지수 형태로 증가하는 EIP-ART의 오류 영역 검출 시간으로 인해 EIP-ART는 4차원 입력 도메인에서 IP-ART보다 63배 이상의 시간을 필요로 하게 되며, 이로 인해 입력 도메인의 차원 수가 증가할수록 EIP-ART의 적용이 어려워진다. 따라서 입력 도메인의 확장과 같은 추가 자원을 요구하는 기법이 아닌, IP-ART 기법 고유의 특징을 적절히 조절하여 테스트 케이스 분포 불균일 현상을 제어하고, 이를 통해 성능을 향상시킬 수 있는 기법이 요구된다.

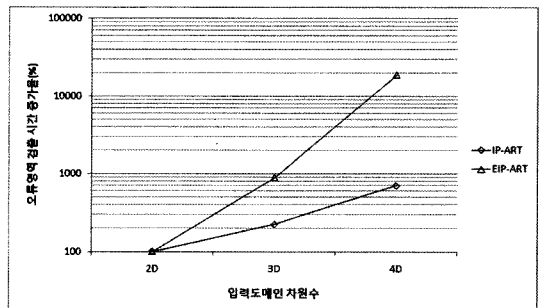


그림 3 오류 영역 검출 시간 증가율

표 1 제한 영역 관리 알고리즘 - 2차원 입력 도메인

```

P : 입력 도메인 분할 수준
E : 이전 테스트 케이스 집합,  $E = \{e_1, e_2, \dots, e_i\}$ 
 $e_i = \{\text{입력 데이터 } Ix_i, \text{ 입력 데이터 } Iy_i\}$ 
D : 입력 도메인,  $D = \{0 \leq Ix_i < \text{MaxX}, 0 \leq Iy_i < \text{MaxY}\}$ 
M : 불린(Boolean) 매트릭스 ( $P \text{ cells} \times P \text{ cells}$ )
 $e_i = \{Ix_i, Iy_i\} \mid e_i \in E$ 을 포함하는 M의 셀의 좌표 =  $(\text{PosX}, \text{PosY})$ ,
 $\text{PosX} = \lfloor Ix_i \times P / \text{MaxX} \rfloor, \text{PosY} = \lfloor Iy_i \times P / \text{MaxY} \rfloor$ 

IP-ART 알고리즘의 제한 영역 설정과정에서.. (* : RAMs 추가 사항)

if PosX or PosY이 1 or P (or 3 or P-2)* 이면
    AdjCell  $\in \{(PosX-1 \leq \text{curPosX} \leq \text{PosX}+1, \text{PosY}-1 \leq \text{curPosY} \leq \text{PosY}+1)\}$ 인
    모든 AdjCell at  $(\text{curPosX}, \text{curPosY})$ 에 대해,
    if curPosX or curPosY가 1 or P (or 3 or P-2)* 이면,
        if AdjCell이 T이면,
            AdjCell := F, CntCandCell 수정
else
    AdjCell  $\in \{(PosX-1 \leq \text{curPosX} \leq \text{PosX}+1, \text{PosY}-1 \leq \text{curPosY} \leq \text{PosY}+1)\}$ 인
    모든 AdjCell at  $(\text{curPosX}, \text{curPosY})$ 에 대해,
    if AdjCell이 T이면,
        AdjCell := F, CntCandCell 수정
    
```

3. 테스트 케이스 분포 조절 기법

3.1 제한 영역 관리(Restriction Area Management)

제한 영역 관리는 테스트 케이스가 선택된 파티션 주위의 제한 영역 형태를 파티션 선택 빈도를 고려하여 설정하도록 하는 방법이며 표 1의 알고리즘과 같이 동작한다. 즉, IP-ART 기법 적용 시 선택 빈도가 높았던 파티션을 관리 대상 파티션으로 정의하고, 테스트 수행 과정에서 관리 대상 파티션이 선택되면, 선택된 파티션 주위의 모든 파티션을 제한 영역으로 설정하는 대신, 그림 4와 같이 관리 대상 파티션만 제한 영역으로 설정한다. 이와 같은 방법은 선택 빈도가 낮은 파티션이 선택 빈도가 높은 파티션의 제한 구역으로 설정되어 빈번하게 파티션 선택 대상에서 제외되는 현상을 억제함으로써 선택 빈도가 낮은 파티션에 선택 가능성을 제공하게 된다. 따라서 제한 영역 관리 기법은 제한 영역의 관리를 통해 선택 가능성을 제공 받는 파티션 수가 많은 높은 분할 수준에서 즉, 낮은 오류 비율을 가진 환경 및 고차원 입력 도메인 환경에서 상대적으로 더 큰 영향력을 가질 것으로 예상된다.

한편 IP-ART를 사용할 때 나타난 테스트 케이스 분포 중 입력 도메인의 가장자리에서 세 번째 라인에 위치한 파티션의 선택 빈도가 높은 원인이 첫 번째 라인에 위치한 파티션에 의해 두 번째 라인에 생성된 제한

영역에 의한 것일 수도 있지만, IP-ART에서 사용하는 제한 영역의 형태에 의해 유발된 것일 가능성도 가지고 있다. 즉, 그 원인이 첫 번째 라인의 높은 빈도에 의한 것이라면 첫 번째 라인에 위치한 파티션의 제한 영역 관리만으로 파티션 선택 빈도 제어가 가능해질 수 있으며, IP-ART에서 사용하는 제한 영역에 의해 생성된 것이라면 세 번째 라인도 관리 대상에 포함되어야 한다. 따라서 이들 간의 영향 평가를 위하여 제한 영역 관리 기법은 첫 번째 라인에 의해 생성되는 제한 영역만을 대상으로 하는 약한 제한 영역 관리(Weak Restriction Area Management, RAMw)와 첫 번째 및 세 번째 라인에 의해 생성되는 제한 영역 모두를 관리하는 강한 제한 영역 관리(Strong Restriction Area Management, RAMs)로 나누어 테스트 케이스 분포 파악 및 성능 평가를 수행한다.

3.2 입력 도메인 타일링(Input Domain Tiling, IDT)

입력 도메인 타일링은 IP-ART에서 사용하는 제한 영역에 대한 직접 개입 없이 구조적인 방법을 통해 입력 도메인 외곽에서 발생하는 테스트 케이스 분포 불균일 문제를 해결하고자 하는 기법이다. IP-ART에서 발생하는 테스트 케이스 분포 불균일 현상은 IP-ART에서 사용하는 제한 영역과 입력 도메인 형태의 상관관계에 의해 발생하는 현상이므로, 실제 입력 도메인의 외부에 가상의 파티션이 존재한다면 실제 입력 도메인 가장

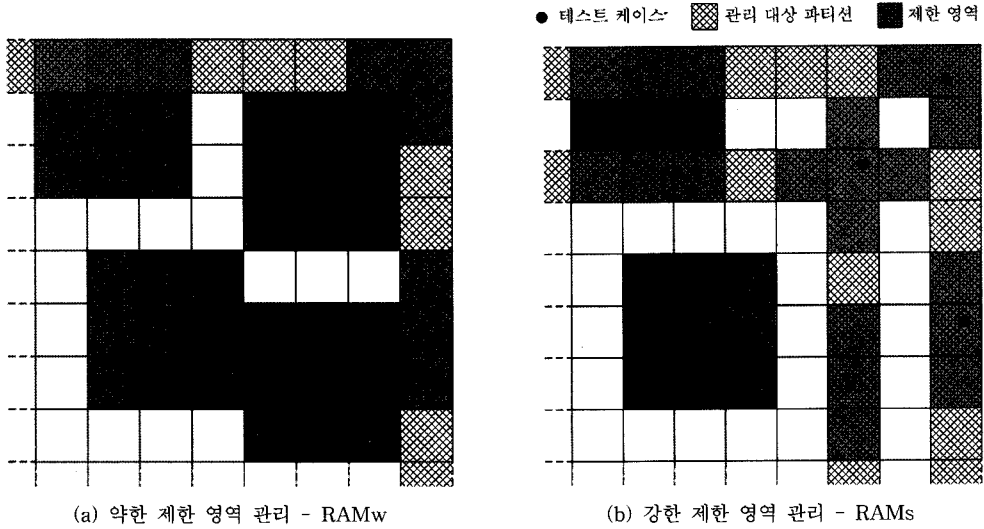


그림 4 제한 영역 관리

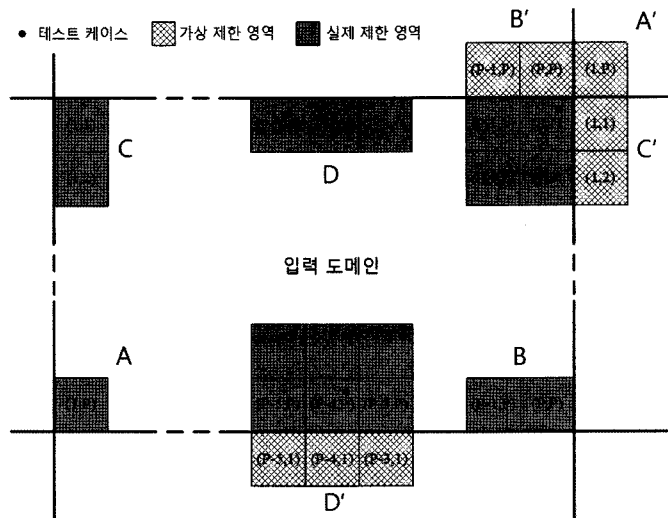


그림 5 입력 도메인 타일링의 제한 영역 설정 (2차원 입력 도메인)

자리에서 발생하는 테스트 케이스 분포 불균일 현상이 제어될 수 있다. 하지만 입력 도메인 가상 확장 기법에서와 같이 입력 도메인의 외부에 가상 테스트 케이스가 존재하도록 하는 기법은 또 다른 부가적인 부하를 요구하기 때문에, 실제 입력 도메인에 대한 구조적인 접근을 통해 이를 해결할 필요가 있다. 따라서 입력 도메인 타일링 기법에서는 그림 5와 같이 입력 도메인을 타일 형태로 가상 배치하되, 가상 배치된 입력 도메인 내에는 테스트 케이스가 위치하지 않도록 하여, 입력 도메인의 외곽에서 발생하는 테스트 케이스 분포 불균일 현상을 제어하며, 표 2에 제시된 알고리즘과 같이 동작한다.

입력 도메인 타일링이 적용되면 입력 도메인은 자신과 동일한 분할 수준을 가진 입력 도메인에 의해 둘러싸이게 되고, 이로 인해 입력 도메인의 모든 경계는 가상 입력 도메인의 경계와 서로 인접하게 된다. 입력 도메인이 이러한 구조를 가지게 되면, IP-ART에서는 무시되었던 입력 도메인 외부의 제한 영역이 인접한 가상 입력 도메인 내의 특정 파티션에 설정되고, 가상 입력 도메인에 제한 영역으로 설정된 파티션의 좌표를 실제 입력 도메인에 적용하여 해당 파티션이 제한 영역으로 설정되도록 한다. 그림 5의 A~D는 각각 가상 입력 도메인에 설정된 A'~D'가 실제 입력 도메인에 적용된

표 2 입력 도메인 타일링 알고리즘 - 2차원 입력 도메인

P : 입력 도메인 분할 수준
 E : 이전 테스트 케이스 집합, $E = \{e_1, e_2, \dots, e_i\}$
 $e_i = \{\text{입력 데이터 } Ix_i, \text{ 입력 데이터 } Iy_i\}$
 D : 입력 도메인, $D = \{0 \leq Ix_i < \text{MaxX}, 0 \leq Iy_i < \text{MaxY}\}$
 M : 불린(Boolean) 매트릭스 ($P \text{ cells} \times P \text{ cells}$)
 $e_i = \{Ix_i, Iy_i\} \mid e_i \in E$ 을 포함하는 M 의 셀의 좌표 = $(\text{PosX}, \text{PosY})$,
 $\text{PosX} = \lfloor Ix_i \times P / \text{MaxX} \rfloor, \text{PosY} = \lfloor Iy_i \times P / \text{MaxY} \rfloor$

IP-ART 알고리즘의 제한 영역 설정과정에서..

$\text{AdjCell} \in \{(PosX-1 \leq \text{curPosX} \leq PosX+1, PosY-1 \leq \text{curPosY} \leq PosY+1)\}$ 인 모든 AdjCell at $(\text{curPosX}, \text{curPosY})$ 에 대해,
 if AdjCell 이 T 이면, $\text{AdjCell} := F$
 이 과정에서 $\text{AdjCell} \notin \{(1 \leq \text{curPosX} \leq P, 1 \leq \text{curPosY} \leq P)\}$ 이면, 타일링 수행 즉, $1 \leq n \leq P$ 일 때, $(0, 0) \rightarrow (P, P), (P+1, P+1) \rightarrow (1, 1),$
 $(0, n) \rightarrow (P, n), (P+1, n) \rightarrow (1, n), (n, 0) \rightarrow (n, P), (n, P+1) \rightarrow (n, 1)$
 CntCandCell 수정

것이고, 가상 입력 도메인에서의 좌표 그대로 실제 입력 도메인에 반영된 것이다. 이와 같은 구성은 입력 도메인을 동일 크기를 가지는 파티션으로 구성된 가상 무한 공간 형태로 만들기 때문에, 입력 도메인의 경계가 없는 것과 같은 효과를 얻게 되고, 따라서 제한 영역의 영향이 입력 도메인 내의 파티션 위치에 상관없이 동일하게 유지되도록 한다. 이에 따라 IP-ART 기법 사용 시 나타났던 입력 도메인 가장자리의 테스트 케이스 분포 불균일 현상이 억제되고, 전체 입력 도메인에 테스트 케이스가 고르게 분포되는 효과를 기대할 수 있게 된다. 입력 도메인 타일링은 그림 5에서 나타난 것과 같이 가상 입력 도메인에 설정된 제한 구역이 실제 입력 도메인에서 기존 테스트 케이스의 인근의 파티션이 아닌 원거리에 위치한 파티션 즉, 선택 가능성이 있는 파티션에 설정되는 단점을 가지지만, 테스트 케이스의 고른 분포 유도를 통해 이와 같은 단점이 극복될 수 있을 것으로 예상된다.

4. 성능 평가

4.1 실험 환경

제안된 기법이 테스트 케이스 분포와 IP-ART 기법의 성능에 미치는 영향 평가를 위한 실험은 표 3과 같은 환경에서 수행되었다. 실험은 다양한 오류 비율을 가진 블록 패턴의 오류 영역을 입력 도메인 내 임의의 위치에 생성하여 수행되었고, 블록의 크기는 입력 도메인의 크기와 오류 비율에 의해 결정하되, 블록의 형태는

블록을 구성하는 모든 변의 길이가 동일하도록 구성하였다. 기법의 성능 평가를 위한 실험은 입력 도메인 차원 수를 증가시켜가며, 오류 비율 당 10,000회를 수행하고, 이 과정에서 얻은 F-지수의 평균을 이용하여 평균 상대 F-지수를 구해 성능을 비교 평가하였으며, 테스트 케이스의 분포 확인을 위한 실험은 표 4에 정의된 것과 같이 2차원 입력 도메인에 대해 수행하였고, 가능한 많은 테스트 케이스의 생성을 위해 오류 비율을 실험 환경의 가장 낮은 오류 비율인 0.0005로 설정하고, 각 기법별로 50,000회의 테스트를 수행하였다.

표 3 성능 평가 실험 환경

입력 도메인	한 입력 데이터의 변위 = 0~10,000 (예 : 3차원 - 10,000×10,000×10,000) EIP-ART의 변위 = 0~1000
입력 도메인 차원	2차원~8차원 (EIP-ART는 2차원~6차원)
오류 비율 (θ)	0.01, 0.005, 0.002, 0.001, 0.0005
오류 패턴	블록
테스트 회수	10,000회 (EIP-ART는 1000회)

표 4 테스트 케이스 분포 확인 실험 환경

입력 도메인	한 입력 데이터의 변위 = 0~10,000 단, EIP-ART의 변위 = 0~1000
입력 도메인 차원	2차원
오류 비율 (θ)	0.0005
오류 패턴	블록
테스트 회수	50,000회

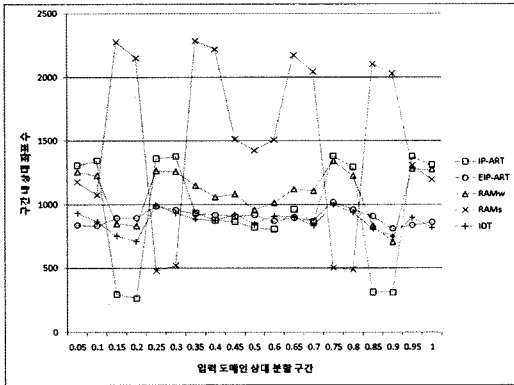
4.2 실험 결과

4.2.1 테스트 케이스 분포

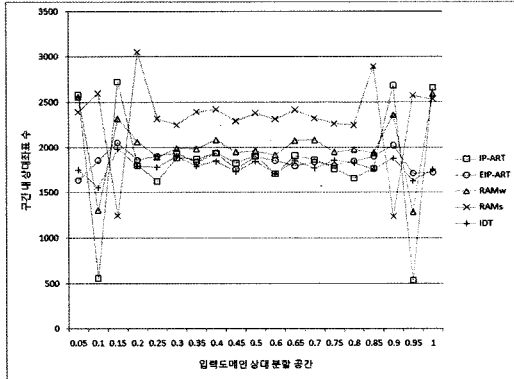
그림 6은 IP-ART와 EIP-ART 및 본 논문에서 제안한 기법인 제한 영역 관리 기법(RAMw, RAMs)과 입력 도메인 타일링(IDT)을 이용했을 때, 2차원 입력 도메인, 오류 비율 0.0005에서 오류 영역을 찾는 작업을 50,000회 반복 수행하는 과정에서 입력 도메인 분할 수

준이 각각 10, 20, 30일 때 선택된 테스트 케이스 분포를 나타낸 것이다. 입력 도메인은 한 입력 데이터의 범위를 [0,1]구간으로 변환하고, 이를 200개의 단위 구간으로 나눈 후, 각 단위 구간 내에 위치하는 테스트 케이스의 첫 번째 입력 데이터의 좌표를 누적하여 표현한 것이다. 두 번째 입력 데이터의 분포는 첫 번째 입력 데이터의 분포와 거의 동일하기 때문에 용이한 비교를 위해 첫 번째 데이터의 분포만을 사용하였다. 또한 첫 번째와 두 번째 입력 데이터의 분포가 유사하기 때문에, 단위 구간별 편차가 심한 경우, 실제 입력 도메인 내에서는 더욱 큰 테스트 케이스 분포 편차를 보이게 된다.

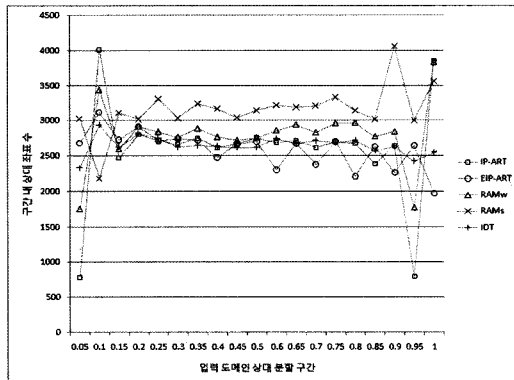
앞서 기술한 것과 같이 IP-ART는 입력 도메인 가장 자리에서 첫 번째와 세 번째 라인에 위치한 파티션에 대한 선택이 많고, 두 번째 라인에 위치한 파티션의 선택 빈도는 낮으며, 네 번째 라인 안쪽 영역은 편차가 그리 크지 않은 분포를 보인다. 또한 입력 도메인을 가상 확장하여 실제 도메인이 IP-ART의 네 번째 라인 안쪽에 위치하도록 한 EIP-ART는 입력 도메인의 전체 영역과 모든 분할 수준에서 상대적으로 고른 분포를 보인다. 한편 제한 영역 관리 기법 중 입력 도메인의 가장자리 첫 번째 라인에서만 제한 영역을 관리하는 RAMw는 IP-ART와 유사한 편차 모양을 보이며, 모든 입력 도메인 분할 수준에서 입력 도메인 가장자리에서 두 번째와 세 번째 라인에 위치하는 파티션의 선택 빈도 편차를 IP-ART의 50% 수준으로 조절하고 있다. 하지만 첫 번째와 세 번째 라인 모두에서 제한 영역을 관리하는 RAMs는 IP-ART의 분포와 다른 형태를 보이며, 입력 도메인 분할 수준 증가에 따라서도 역시 변화하는 모습을 보인다. RAMs는 관리 대상이 되는 라인이 상대적으로 많기 때문에 낮은 입력 도메인 분할 수준에서는 관리 대상과 인접한 라인에 위치한 파티션이 높은 빈도로 선택되도록 유도하게 되며, 이러한 현상은 입력 도메인 분할 수준이 증가할수록 감소하는 경향을 보인다. 이에 따라 RAMs는 입력 도메인 분할 수준이 20 이상일 때에는 IP-ART보다 균일한 테스트 케이스 분포를 생성하므로, 테스트 과정에서 분할 수준이 높게 유지될 가능성이 높은 환경 즉, 낮은 오류 비율을 가진 입력 도메인과 입력 도메인 분할에 따라 파티션의 수가 빠르게 증가하는 고차원 입력 도메인에서 상대적으로 더 우수한 성능을 보일 것으로 예상할 수 있다. IDT는 입력 도메인 전체 영역과 모든 분할 수준에서 EIP-ART와 유사하거나 더 균등한 테스트 케이스 선택 빈도 분포를 보인다. 따라서 이전 테스트 케이스와 무관한 영역에 대한 제한 영역 설정으로 인한 일정 수준의 역효과는 존재하겠지만, 오류 비율 및 입력 도메인의 차원 수 증가에 대한 확장성을 가질 것으로 예상된다.



(a) P=10



(b) P=20



(c) P=30

그림 6 파티션 선택 분포

4.2.2 수행 시간

그림 7은 입력 도메인 차원 수 증가에 따른 각 기법의 오류, 영역 검출 시간 증가율을 제시하고 있다. 오류 비율이 0.01인 입력 도메인에서 각 기법을 이용해 1000 회씩의 테스트를 수행하여, 오류영역을 찾아내는데 소요된 평균 시간을 구하고, 2차원 입력 도메인에서 소요된 시간을 기준으로 각 입력 도메인 차원에서의 오류 영역 검출 시간 증가율을 나타낸 것이다. 그리고 표 5는 각 기법을 사용했을 때 오류 영역 검출에 소요된 시간과 IP-ART를 사용했을 때 소요되는 시간의 비율이다.

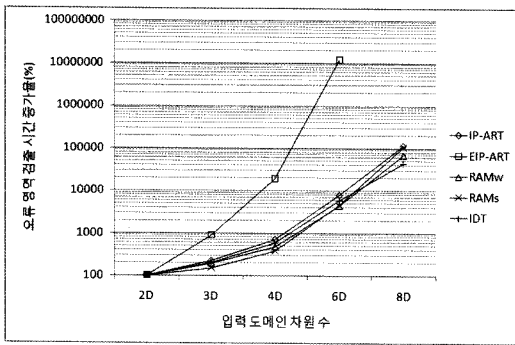


그림 7 오류 영역 검출 시간 증가율

결과에 따르면 EIP-ART를 제외한 모든 기법이 IP-ART보다 낮은 수준의 오류 검출 시간 증가율을 보이고, 또한 2차원 입력 도메인을 제외한 나머지 차원에서 대체로 IP-ART보다 짧은 수행 시간을 요구한다. 이는 이들이 IP-ART보다 입력 도메인 차원 수 증가에 대한 적응성이 우수함을 의미한다. 6차원 이하의 입력 도메인에서는 제한 영역 관리 기법이 다른 기법에 비해 상대적으로 더 낮은 증가율을 보이지만, 6차원 이상에서는 요구 시간이 급격히 증가하여 RAMs의 경우 IP-ART와 유사한 증가율을 갖는다. IDT는 입력 도메인 차원 수 증가에 대한 요구 시간 증가율 변화가 다른 기법에 비해 심하지 않아, 8차원 입력 도메인에서는 다른 기법에 비해 요구 시간 증가율이 가장 낮아진다. 이와 같은 IDT의 입력 도메인 차원 수 증가에 대한 안정성은 표 5에 잘 나타나 있으며, 8차원 입력 도메인에서 IP-ART의 42% 수준의 시간으로 오류 영역을 검출하는 등 다른 기법에 비해 고차원 입력 도메인에서도 가장 안정적인 수행 시간을 보인다. EIP-ART의 경우는 4차원 이상의 입력 도메인에서 더 급격한 수행 시간 증가율을 보이고 있으며, 이에 따라 표 5에 제시된 것과 같이 6차원 이상의 입력 도메인에서는 실험에 너무 오랜 시간이 요구되어 8차원 입력 도메인에 대한 실험은 수행되지 못했다.

표 5 IP-ART 기준 오류 영역 검출 시간 비율(IP-ART = 1)

	EIP-ART	RAMw	RAMs	IDT
2D	2.38	1.07	1.07	1.07
3D	9.52	0.93	0.72	0.97
4D	63.92	0.73	0.58	0.89
6D	3708.46	0.60	0.62	0.80
8D		0.65	1.01	0.42

4.2.3 평균 상대 F-지수

표 6은 입력 도메인 및 오류 비율 변화에 따른 각 기법의 평균 상대 F-지수를 나타낸다. 평균 상대 F-지수는 오류 비율이 낮을수록 그리고 입력 도메인의 차원 수가 낮을수록 대체로 더 낮은 값을 가지며, 2차원 입력 도메인에서는 각 기법 간 성능 격차가 그다지 크지 않지만 입력 도메인의 차원 수가 증가할수록 격차는 심해지고, 평균 상대 F-지수의 증가율도 대체로 더 커진다.

입력 도메인 내의 오류 비율 변화에 가장 민감한 기법은 RAMs로, 일정하지 않은 형태의 제한 구역 사용이 오류 비율에 따른 성능 편차를 유발하며, 또한 입력 도메인 차원 수가 증가할수록 이러한 제한 구역의 사용량이 늘어나게 되면서 성능 편차가 심해지는 것으로 보인다.

표 6 평균 상대 F-지수

입력도메인	오류 비율	IP-ART	EIP-ART	RAMw	RAMs	IDT
2D	0.01	0.639	0.568	0.652	0.722	0.589
	0.005	0.617	0.571	0.645	0.673	0.580
	0.002	0.606	0.551	0.620	0.667	0.587
	0.001	0.593	0.553	0.614	0.648	0.578
	0.0005	0.591	0.530	0.595	0.634	0.578
3D	0.01	0.834	0.628	0.783	0.713	0.670
	0.005	0.801	0.610	0.768	0.729	0.658
	0.002	0.756	0.590	0.754	0.734	0.653
	0.001	0.711	0.550	0.717	0.740	0.643
	0.0005	0.698	0.563	0.702	0.735	0.637
4D	0.01	1.126	0.692	0.889	0.809	0.754
	0.005	1.045	0.691	0.853	0.726	0.730
	0.002	0.993	0.605	0.835	0.712	0.723
	0.001	0.937	0.524	0.814	0.714	0.690
	0.0005	0.894	0.489	0.813	0.743	0.709
6D	0.01	1.840	0.665	1.309	1.443	0.859
	0.005	1.780	0.665	1.185	1.202	0.852
	0.002	1.671	0.625	1.118	1.013	0.827
	0.001	1.570	0.574	1.080	0.887	0.848
	0.0005	1.518	0.366	1.041	0.778	0.833
8D	0.01	3.298		2.488	3.224	0.979
	0.005	2.977		2.008	2.592	0.986
	0.002	2.705		1.742	1.865	0.996
	0.001	2.868		1.666	1.641	0.982
	0.0005	2.795		1.665	1.494	0.980

다. IP-ART, EIP-ART, RAMw는 오류 비율 변화에 대한 적응성이 대체로 유사한 것으로 나타났으며, 전체 입력 도메인 영역을 고르게 사용하는 IDT의 오류 비율에 따른 성능 편차가 2% 수준으로 가장 적었다. 입력 도메인 차원 수 증가에 따른 성능 편차는 IP-ART가 39%로 가장 크며, 제한 영역 관리 기법도 IP-ART보다는 작지만 1차원 증가 당 19~20% 수준의 성능 하락을 보였다. 한편 EIP-ART는 6차원 입력 도메인까지의 실험에서 차원 수 증가 당 약 1% 수준으로 가장 좋은 안정성을 보였으나, 8차원 입력 도메인에 대한 실험 결과가 없어 타 기법과의 정확한 비교가 어려우며, IDT는 1차원 증가 당 약 7% 수준의 성능 저하를 보여, EIP-ART를 제외한 타 기법에 비해 입력 도메인 차원 수 증가에 대한 안정성이 가장 우수한 것으로 나타났다. 이에 따라 IDT는 실험 환경으로부터 받는 영향이 작고, 따라서 적용 환경에 대한 적응성이 우수함을 알 수 있다.

평균 상대 F-지수를 기준으로 살펴보면, IP-ART는 4차원 입력 도메인의 높은 오류 비율에서와 6차원 이상의 입력 도메인에서는 평균 상대 F-지수가 1 이상이 되어 순수 랜덤 테스트보다 효율성이 떨어지기 때문에 이와 같은 환경에 적용하기는 어렵다. 한편 RAMw는 4차원 입력 도메인의 모든 오류 비율까지, RAMs는 6차원 입력 도메인의 낮은 오류 비율까지 1 이하의 평균 상대 F-지수를 보여 입력 도메인 차원 수 증가에 대한 적응성이 IP-ART보다 향상되기는 하였지만, 추가적으로 입력 도메인이 확장되는 경우에는 적용이 어려움을 확인할 수 있었다. 또한 RAMs와 RAMw의 성능 비교를 통해 일부 환경을 제외하면 입력 도메인의 외곽 첫 번째 라인에 위치한 파티션에 대한 높은 선택 빈도의 영향이 사용하는 오류 영역의 영향보다 크게 작용하고 있음을 알 수 있다. IDT는 8차원 입력 도메인까지의 모든 실험 환경에서 1 이하의 평균 상대 F-지수를 보여, 실험이 수행된 전체 환경에서 순수 랜덤 테스트보다 더 우수한 성능을 보이고 있다. IDT의 이와 같은 성능은 IP-ART를 기준으로 했을 때, 전체 입력 도메인에서 최소 2%(2차원)에서 최대 194%(8차원)의 성능 향상이 있었음을 의미하며, 고차원 입력 도메인에 대한 테스트에서 수행 속도가 느린 EIP-ART를 보완하는 기법으로서 IDT의 가능성이 확인되었다.

5. 결론

IP-ART는 초기 ART 기법의 단점인 테스트 케이스 간의 거리 계산에 필요한 많은 연산량을 입력 도메인 분할 기법을 이용해 효과적으로 개선하였고, EIP-ART는 IP-ART의 테스트 케이스 분포 특징을 적절하게 이용하여, IP-ART 기법의 적응성과 성능을 확장시켰다.

하지만, EIP-ART는 우수한 성능에 비해 테스트 케이스 생성에 오랜 시간을 요구하기 때문에, EIP-ART와는 다른 접근 방법을 사용하는 정책이 요구되었다. 따라서 본 논문에서는 테스트 케이스 분포 조절을 통해 IP-ART의 성능 개선을 시도하는 두 가지 정책을 제안하고, 이들의 성능을 비교 평가하였다.

실험 결과에 따르면 제안된 두 가지 정책 모두가 오류 영역 검출 속도면에서 IP-ART와 유사하거나 짧은 시간을 요구하고, 기존의 IP-ART보다 입력 도메인 차원 수 증가에 대해 더 나은 적응성을 보이며, 3차원 이상의 입력 도메인에서는 모든 제안 기법이 IP-ART 기법보다 우수한 성능을 보인다. 특히 입력 도메인 타일링을 이용한 정책은 모든 실험 환경에서 순수 랜덤 테스트보다 우수한 성능을 보여 고차원 입력 도메인에서 EIP-ART를 대신해 이용될 수 있음을 확인할 수 있었다.

하지만 제안된 정책은 EIP-ART와 비교 시, 오류 영역 검출 속도만을 개선하고 있을 뿐, 입력 도메인의 확장에 따른 적응력이나 평균 상대 F-지수는 EIP-ART에 미치지 못하고 있기 때문에 이들을 적절하게 가공하여 추가적인 성능 개선을 이끌어 낼 수 있는 정책이 요구된다.

참고 문헌

- [1] R. Hamlet, "Random testing," Encyclopedia of Software Engineering, pp. 970-978, Wiley, 1994.
- [2] G. J. Myers, "The Art of Software Testing," 2nd ED., John Wiley & Sons, 2004.
- [3] H. Bati, L. Giakoumakis, S. Herbert, A. Surna, "A genetic approach for random testing of database systems," in Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB '07), pp. 1243-1251, Sep. 2007.
- [4] 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 (QSIC '03), pp. 20-23, Nov. 2003.
- [5] J. Regehr, "Random Testing of Interrupt-Driven Software," in Proceedings of the 5th ACM international conference on Embedded software (EMSOFT '05), pp. 290-298, Sep. 2005.
- [6] J. Duran and S. Ntafos, "An Evaluation of Random Testing," IEEE Transactions on Software Engineering, volume 10, no. 4, pp. 438-444, Jul. 1984.
- [7] D. Hamlet, "When Only Random Testing will do," in Proceedings of the First International Workshop on Random. Testing (RT '06), pp. 1-9, Jul. 2006.
- [8] 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.
- [9] K. P. Chan, T. Y. Chen, and D. Towey, "Restricted Random Testing," in Proceedings of the 7th European Conference on Software Quality (ECSQ 2002), vol. 2349 of Lecture Notes in Computer Science, pp. 321-330, Springer, Jun. 2002.
 - [10] 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 (Ada-Europe 2006), volume 4006 of Lecture Notes in Computer Science, pp. 155-166. Springer-Verlag, Jun. 2006.
 - [11] T. Y. Chen, H. Leung, and I. K. Mak, "Adaptive Random Testing," in Proceedings of the 9th Asian Computing Science Conference (ASIAN 2004), vol. 3321 of Lecture Notes in Computer Science, pp. 320-329. Springer, Dec. 2004.
 - [12] J. Mayer, "Adaptive Random Testing with Randomly Translated Failure Region," in Proceedings of the 1st International Workshop on Random Testing (RT 2006), pp. 70-77, Jul. 2006.
 - [13] J. Mayer, C. Schneckenburger, "Adaptive Random Testing with Enlarged Input Domain," in Proceedings of the Sixth International Conference on Quality Software (QSIC '06), pp. 251-258, Aug. 2006.
 - [14] 신승훈, 박승규, "입력 도메인 확장을 이용한 반복 분할 기반의 적응적 랜덤 테스트 기법", 정보처리학회논문지D, volume 15D, no. 4, pp. 531-540, 2008년 8월.
 - [15] 이광규, 신승훈, 박승규, "반복 분할 기반의 적응적 랜덤 테스트 향상 기법", 대한전자공학회 논문지, 제45권, CI편, 제5호, pp. 180-191, 2008년 9월.
 - [16] 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, Dec. 1996.
 - [17] 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.



박 승 규

1974년 서울대학교 응용수학과 공학사
 1976년 한국과학기술원(KAIST) 전산학과 석사.
 1982년 Institut National Polytechnique de Grenoble 전산학과 박사.
 1976년~1977년 한국과학기술연구소(KIST) 연구원.
 1982년~1983년 KIET(현ETRI) 실장/선임연구원.
 1984년~1985년 미국 IBM 왓슨연구소 연구원.
 1985년~1992년 ETRI 연구위원/책임연구원.
 1992년~현재 아주대학교 정보통신대학 정보 및 컴퓨터공학부 교수.
 관심분야는 임베디드 테스트, 자가 컴퓨팅/치료 시스템, 차세대 컴퓨터 구조 등



최 경 회

1976년 서울대학교 수학교육과 학사
 1979년 프랑스 그랑테폴 ESEIHT 석사.
 1982년 프랑스 Paul Sabatier 대학 정보공학과 박사.
 1982년~현재 아주대학교 정보통신대학 정보 및 컴퓨터공학부 교수.
 관심분야는 운영체제, 분산시스템, 실시간, 임베디드 시스템 테스트 등



신 승 훈

2000년 아주대학교 정보 및 컴퓨터공학부 학사.
 2002년 아주대학교 정보통신공학과 석사.
 2002년~현재 아주대학교 정보통신공학과 박사과정.
 관심분야는 소프트웨어 테스트 자동화, 멀티미디어 서비스 정책 등