

# 유전자 알고리즘을 이용한 보안 대책의 생성

박준형<sup>o</sup> 방영환 이강수\* 남기호\*\*

한남대학교 컴퓨터공학과\*, 프롬투 정보통신(주)\*\*

{junsido, bangyh}@se.hannam.ac.kr, gslee@mail.hannam.ac.kr, khnam@from2.co.kr

## Generation of Security Measure by Using Simple Genetic Algorithm

Junhyoung Park<sup>o</sup> Younghwan Bang Gangsoo Lee\* Keehyo Nam\*\*

Dept. of Computer, Hannam University\*, From2\*\*

### 요 약

현재 많은 조직에서 위험 분석을 통해 현재 자신들의 보안상의 문제점을 파악하고 그에 따른 대책을 적용하고 있다. 기존의 보안 대책을 평가하고 새로운 보안 대책을 적용하는 데 많은 어려움이 따르므로 본 연구에서는 Simple Genetic Algorithm을 이용하여 현재 조직의 상황에 적절한 보안 대책을 제시할 수 있는 방법을 연구하고자 한다.

### 1. 서론

오늘날 많은 조직에서 이루어지고 있는 위험 분석은 최종적으로 위협으로부터 보호해야할 조직의 자산 보호 및 조직 업무 연속성의 보장이다. 보안 대책은 위험 분석의 하위 단계에서 분석된 조직을 위협으로부터의 위협을 최소로 줄이기 위해 조직에 적용되어 지는 활동으로, 이런 활동이 적절하게 제시되고 활동이 이루어져야만 노출된 위협으로부터 자산을 보호할 수 있고 업무 연속성 또한 보장할 수 있다.

본 논문은 위험 분석 프로세스 중 위험 분석 후 보안 대책의 제시에 유전자 알고리즘을 이용하고 있으며, 이 알고리즘을 통해 제시된 대책이 정확하다고 하기보다는 최적의 대책을 제시하는 것을 목적으로 한다. 보안 대책의 제시 문제는 조직에서 미리 정의한 총 대책 비용의 범위 안에서 적절한 대책을 선택하는가 하지 않는가의 문제이다. 이런 최적화의 문제는 곧 비결정적인 알고리즘에 의해 다항식 시간 내에 해결되는 결정 문제들의 집합인 NP(Nondeterministic polynomial) 문제로 정확한 답이 없고 가장 최적의 해를 구하는 문제이다. 그리고 많은 후보 대책들이 존재함으로 인하여 문제의 복잡도가 증가하게 되기 때문에 문제의 해를 일일이 열거해 나가는 단순한 방법을 통해서 해를 구하기 어렵다. 따라서 이러한 문제는 조합 최적화(combinatorial optimization)의 문제이며 0/1 Knapsack 문제와 같다. 곧 최적 해에 근접한 해를 찾는 문제이면서 변수들의 값이 이진 값이면 뉴럴 네트워크 또는 유전자 알고리즘을 이용하게 되는데 뉴럴 네트워크는 학습이 필요하게 되므로, 이진 값을 통해 값을 결정하는 유전자 알고리즘을 이용하여야 한다. 이러한 Knapsack 문제의 경우 단순한 것 같지만 수많은 조합형 특성을 탐색할 수 있게 해주는 NP-hard 문제로서 그 해를 얻기가 대단히 어렵지만 유전자 알고리즘을 이용하여 해결하는 것이 가능하다[1],[2],[3].

본 연구의 2장에서는 기존의 Knapsack 문제와 본 연구의 보안 대책 생성 문제를 비교하여 수학적으로 나타내었으며, 3장에서는 제시된 문제를 유전자 알고리즘을 이용하여 보안 대책을 생성한다. 마지막으로 4장에서는 본 연구의 결론으로 제시된 알고리즘을 이용한 보안 대책 생성이 가지는 의미에 대해 알아보고 추가적으로 보완할 점을 제시한다.

### 2. 보안 대책 생성 문제와 Knapsack 문제

기존의 많은 탐색알고리즘이 존재하지만 ①얼마나 빨리 해를 발견하는가? ②얼마나 적은 메모리를 사용하는가? ③발견된 해가 얼마나 좋은 해인가? 에 의해 탐색 알고리즘을 결정한다. 하지만 요즘 컴퓨터의 발전으로 인해 메모리의 문제는 해결되었다고 본다

면, 얼마나 빨리 그리고 좋은 해를 발견하는가 좋은 알고리즘의 관건이라고 할 수 있다. 즉, 최소의 노력으로 어떤 하나의 해를 발견하는 것도 중요하지만 최적 해에 가까운 해를 발견하는 것도 중요하다는 것이다[4]. 본 연구에서 제시된 보안 대책 선택 제시 문제는 0/1 Knapsack 문제라고 볼 수 있는데, 보안 대책의 선택이 조직에서 적용할 수 있는 최대 보안 대책 비용의 범위 안에서 가장 적절한 최적의 대책 집합을 선택하는 것이기 때문이다. 즉, 선택하거나 아니면 선택하지 않는 문제이다. 따라서 공식으로 나타낸다면 다음과 같이 Knapsack 공식에 적용할 수 있다[5].

$$P : \quad \text{maximize} \quad \sum_{i=0}^n pixi$$

$$\text{subject to} \quad \sum_{i=1}^n wixi \leq C$$

$$xi = \{0, 1\}, i = 1 \dots n$$

$n$  = 후보 대책 수,  $i$  = 후보 대책 군,  $pi$  = 비용 대비 효과

$xi$  = 선택 여부,  $wi$  = 대책 비용,  $C$  = 최대 대책 비용

따라서 비용 대비 효과에 의해 선택된 대책들을 총 대책 비용 안에서 선택된 대책들의 비용을 더하여 최적의 해를 구하게 된다. 하지만 본 연구에서의 보안 대책 생성 문제는 이와는 다르게 추가적인 제약조건이 형성된다. 따라서 전형적인 Knapsack 문제에서 약간의 수정이 된다. 추가적인 제약 조건은 다음과 같다.

1. 하나의 후보 대책이 선정이 되면 반드시 선정되어야 하는 다른 후보 대책이 존재하는 경우가 있다.
2. 위에서 동시에 선정된 후보대책은 실제로 같은 후보 대책이므로 비용을 중복하여 계산하면 안 된다.

위에 제시된 제약 조건을 적용하여 공식을 다시 구성하면 다음과 같다.

$$P' : \quad \text{maximize} \quad \sum_{i=0}^n pi' xi'$$

$$\text{subject to} \quad \sum_{i=1}^n wi' xi' \leq C$$

$$xi' = \{0, 1\}, i = 1 \dots n'$$

$n'$  = 동일한 후보 대책들을 하나로 묶은 경우의 후보 대책군의 수  
 $pi'$  =  $pi \times i$  후보 대책의 수,  $xi'$  = 선택여부,  $wi$  = 대책 비용,

$wi' = \frac{wi}{\text{후보대책 } i \text{ 군의 후보대책수}}, C = \text{최대 대책 비용}$

조합 최적화 문제는 유전자 알고리즘의 응용 예들 중 가장 많은 결과를 낸 것들 중의 하나인데[6], 따라서 Knapsack 문제를 유전

자 알고리즘을 이용하여 해결 할 수 있으므로 본 논문의 보안 대책 생성 문제 또한 유전자 알고리즘을 통하여 최적의 해를 구할 수 있다.

유전자 알고리즘은 유전학과 자연 진화를 흉내낸 적응 탐색 법으로 복잡한 최적화 문제를 해결하기 위해서 집단을 사용하고, 요기에 모의 진화를 일으켜 이를 점진적으로 개선해 나간다. 집단은 다수의 염색체로 구성되고 염색체는 문제 공간상의 한 점(잠정적인 해)을 대표하게 된다. 흔히 염색체들은 유전자의 역할을 반영하도록 비트 형태로 표현된다[7]. 유전자 알고리즘은 자연 세계의 진화과정을 컴퓨터 상에서 시뮬레이션 함으로써 복잡한 실세계의 문제를 해결하고자 하는 계산 모델이다. 최적화하고자 하는 파라미터들을 염색체로 표현하여 주어진 함수의 적합 정도에 따라 적응도(fitness)를 할당하고, 적응도가 높은 인자의 유전자를 추출하여 교배와 돌연변이 등을 통해 새로운 염색체를 생성해 낸다[8]. 이렇게 만들어지는 염색체는 세대를 거치면서 최적화가 이루어지게 된다. 이러한 유전 알고리즘은 특히 적응적 탐색과 학습 및 최적화를 통한 공학적인 문제의 해결에 많이 응용된 바 있으며, 최근 들어 특히 신경망과 퍼지 로직과의 결합으로 그 응용 범위는 점점 늘어나고 있다[9].

**3. 유전자 알고리즘을 이용한 보안 대책 생성**

위험 분석에서 위협의 식별과 함께 식별된 위협에 따른 취약성들의 목록이 제시된다. 각 제시된 취약성에 따라서 보안 대책들이 제시되게 된다. 위협과 취약성은 1 : n 으로 매핑되고 되고 취약성과 보안 대책은 1 : 1로 매핑이 된다. 각 취약성에 따른 보안 대책은 대책의 구현에 따른 비용과 그 위협 감소의 효과 값을 포함하고 있다.

초기 모집단 설정에 있어서, 1)사용자의 입력, 2)최소 비용 대책, 3) 위협 감소치가 큰 순서로 모집단을 구해 예를 구현하도록 하겠다. 예를 위하여 위협과 취약성은 1 : n 으로 매핑되고 되고 취약성과 대책은 표1에 제시하며, 대책의 구현에 따른 비용과 대책의 구현으로 얻어지는 위협 감소의 효과를 수치로 표현하여 제시한다.

- 위협 - 취약성a - 대책1 - 비용, 효과
- 취약성b - 대책2 - 비용, 효과
- 취약성c - 대책3 - 비용, 효과
- 취약성d - 대책4 - 비용, 효과 등

**단계 1 : 초기화**

우선 모집단의 크기(N<sub>p</sub>)는 10이고, 각 개체의 원소 수(N<sub>e</sub>)는 10개로 선택된 모집단은 표2와 같다. 선택된 경우는 1, 선택되지 않은 경우는 0으로 세팅한다.

V1(최소비용)	= ( 1 0 1 0 1 1 1 0 0 1 )
V2(사용자선택)	= ( 1 1 0 0 1 0 1 0 1 0 )
V3(위협의 감소치)	= ( 1 1 1 0 1 0 1 0 0 1 )

**단계 2 : 적응도 평가**

적응도는 비용 대비 효과의 값을 가지고 각 개체에 대해 적응도 값을 구하여 순위를 매긴다. 표2에서 보다 시피 총 비용에 대한 위협의 총 감소치는 큰 감소로 선택한 경우(V3)가 1.446으로 1위, 최소비용으로 선택했을 경우(V1)가 1.292로 2위, 사용자 선택에 의해 선택된 개체(V2)가 1.173으로 3위이다.

**단계 3 : 선별**

선별은 확률 바퀴방법을 사용한다. 먼저 모집단 개체들의 적응도 총합을 구한다. 현재의 적응도 총합은 다음과 같고, 선별확률을 구한다.

<표 1> 제시된 대책 리스트 예

위협	취약성	대책	대책구현비용	위협감소치	위협감소치/대책비용
A	a	1	8	10	1.25
	b	2	12	20	1.66
	c	3	5	10	2.00
B	a	4	10	5	0.50
	b	5	8	8	1.00
	c	6	6	5	0.83
C	a	7	6	8	1.33
	b	8	10	6	0.60
D	a	9	12	8	0.66
	b	10	8	12	1.50
계			77	80	9.83

$$F = \sum_{i=1}^3 eval(V_i) = 3.911$$

$$P1 = eval(V1) / F = 0.330$$

$$P2 = eval(V2) / F = 0.300$$

$$P3 = eval(V3) / F = 0.370$$

다음으로 선별을 하기 위하여 각 개체 V<sub>i</sub>, i = 1, 2, 3에 대한 누적 확률을 구하는 데 그 결과가 다음과 같다.

$$q1 = 0.330, q2 = 0.629, q3 = 1.000$$

누적 확률이 구해 졌으면 [0, 1] 범위를 갖는 난수를 발생시킨다. 그 난수의 범위가 [q<sub>i-1</sub>, q<sub>i</sub>] 이면 i 번째 개체를 선택한다. 이 과정을 모집단 크기만큼(여기서는 3) 반복하여 새로운 모집단을 구성한다. 다음과 같은 순서로 난수가 발생하였다고 하자.

$$1 : 0.31190 \quad 2 : 0.76160 \quad 3 : 0.92145$$

첫 번째 난수 0.31190은 q1 보다 작으므로, V1이 새로운 모집단을 위해 선택된다. 두 번째 난수 0.76160은 q2보다 크고 q3보다 작으므로, V3이 새로운 모집단을 위해 선택된다. 이런 방식으로 난수에 대해 선별을 하면 아래와 같은 새로운 모집단이 만들어진다.

$$V'1 = ( 1 0 1 0 1 1 1 0 0 1 ) [V1]$$

$$V'2 = ( 1 1 1 0 1 0 1 0 0 1 ) [V3]$$

$$V'3 = ( 1 1 1 0 1 0 1 0 0 1 ) [V3]$$

여기에서 주목할 점은 극단적으로 구성을 하긴 하였지만 선별에 있어서 여러 번 선택되는 개체가 있는가 하면 선택되지 않고 사라져 버리는 개체가 있다는 점이다. 이것은 적응도에 따라서 확률적으로 선택되기 때문이다. 적응도가 높으면 그만큼 다음 세대에 유전될 확률이 높다.

**단계 4 : 교차**

선별에 의해 생성된 새로운 모집단(개체 V<sub>i</sub>)에 교차 연산자를 적용하는 단계이다. 교차율(P<sub>c</sub>)이 0.25라고 정하면 만약 지금까지의 예와는 달리 전체 개체수가 20개라면 평균적으로 25%(20개 × 0.25 = 5개)가 교차에 참여하리라 예상할 수 있다. 하지만 지금까지의 예에서는 3개를 가지고 하였으므로, 3개를 기준으로 작성한다. 먼저 모집단 (V<sub>i</sub>)에 있는 각 개체에 대해 [0, 1]사이의 난수를 발생시킨다. 만약 이 난수가 교차율보다 적으면, 교차연산을 위해 이 개체를 교차 대상 개체로 선택한다. 난수가 아래와 같이 발생하였다고 가정한다.

$$1 = 0.01258 \quad 2 = 0.32860 \quad 3 = 0.21580$$

위 난수 중에서 1번과 3번 개체가 0.25보다 작으므로 이는 개체 V'1과 V'3이 교차대상 개체로 선택되었다는 의미이다. 선택된 개체들을 임의로 두 개씩 쌍을 만든다. 위의 예에서는 단 두 개뿐이므로 이것을 쌍으로 한다. 많은 개체를 가지고 위의 작업을 행하였을 경우에 만약 홀수 개의 개체가 선택되었다면 선택되지 않은 개체 중 임의의 한 개체를 추가하거나 선택된 개체 중에서 임의로

<표 2> 선택된 대책 예

선택	대책1	대책2	대책3	대책4	대책5	대책6	대책7	대책8	대책9	대책10	총비용	총감소	감소치/비용
최소비용	1	0	1	0	1	1	1	0	0	1	41	53	1.292
사용자선택	1	1	0	0	1	0	1	0	1	0	46	54	1.173
큰 감소치	1	1	1	0	1	0	1	0	0	1	47	68	1.446

<표 3> 교차와 돌연변이 후 선택된 대책

선택	대책1	대책2	대책3	대책4	대책5	대책6	대책7	대책8	대책9	대책10	총비용	총감소	감소치/비용
V1	1	0	1	0	0	0	1	0	0	1	27	40	1.481
V2	1	0	1	0	1	0	1	0	0	1	35	48	1.371
V3	1	1	1	1	1	1	1	0	0	1	63	78	1.238

하나를 제거해 짝수로 만든 다음 쌍을 형성하여야 한다. 그러면 결정된 V1과 V3을 이용하여 교차시키도록 한다. 각 쌍에 대해 교차시킬 위치(pos), 즉 절단점을 결정하여야 하는 데 그것은 범위 [2, 10](단약 pos의 절단점의 뒷부분을 교차시킨다면 범위 [1, 9])의 정수의 난수를 발생시켜 결정한다. V1과 V3의 쌍에서 pos = 5이 라면 다음과 같은 V1과 V3의 자손들이 생산된다.

$$V^1 = (10101 \parallel 11001) \\ V^3 = (11101 \parallel 01001)$$

$$V^{*1} = (10101 \parallel 01001) \\ V^{*3} = (11101 \parallel 11001)$$

이렇게 만들어진 개체는 부모의 개체들과 교체가 된다. 그러면 모집단의 개체들은 다음과 같이 구성된다.

$$V^1 = (1010101001) \\ V^2 = (1110101001) \\ V^3 = (1110111001)$$

**단계 5 : 돌연변이**

돌연변이율(P<sub>m</sub>)을 0.02라고 한다면 모집단의 원소 중에서 평균 2%가 돌연변이 연산을 행할 것이다. 지금까지의 예에서 전체 모집단의 크기(N<sub>p</sub>) = 3, 개체의 원소 수(N<sub>g</sub>) = 10이므로 한 세대에서의 평균 0.6(3 × 10 × 0.02)개의 원소가 돌연변이가 된다. 모든 원소는 같은 확률로 돌연변이의 기회를 갖는다. 따라서 모집단의 각 원소에 대해 [0, 1]의 범위를 갖는 난수를 발생시켜 발생된 난수가 P<sub>m</sub>보다 적으면, 그 원소는 돌연변이를 수행한다. 전체 원소에 대해 난수를 발생 시켜 돌연변이율(P<sub>m</sub>) = 0.02보다 작은 원소번호가 다음과 같다고 하자.

원소번호	난수	개체번호	원소위치
5	0.01254	1	5
12	0.00124	2	12
24	0.19845	3	24

여기에서 3개의 개체가 돌연변이 영향을 받는다는 것을 알게 되었고, 아래 개체에서 음영 부분이 돌연변이 된 원소를 나타낸다.

$$V^1 = (1010 \blacksquare 01001) \\ V^2 = (1 \blacksquare 10101001) \\ V^3 = (111 \blacksquare 111001)$$

지금까지의 교차와 돌연변이 후의 모집단은 아래와 같다.

$$V1 = (10100001001) \\ V2 = (1010101001) \\ V3 = (1111111001)$$

**단계 6 : 적응도 검사**

이 단계에서는 선별과 유전 연산에 의해 새로이 구성된 개체들의 적응도를 평가한다. 평가된 적응도에 따라 다음 세대를 위한 선별이 이루어지고 일련의 유전 과정이 반복된다. 적응도는 초기 모집단에서의와 같은 방식으로 평가된다. 표3 참고. 한 세대가 지난 후 새로이 생성된 3개 개체의 적응도 총합은 4.090으로 초기 모집단에서의 3.911 보다 높다는 것을 알 수 있다. 또한 가장 좋은 개체의 적응도가 1.481로써 역시 초기 모집단에서 적응도가 가장 높은 적응도(1.446)보다 높다.

**단계 7 : 종료조건**

종료 조건이 만족되지 않으면 단계 3으로 가서 선별 과정과 유전 연산, 평가를 반복하게 된다. 종료조건이 만족되면 끝내는데 종료조건은 여러 가지가 있을 수 있다. 본 연구에서는 새로 생성된 세대수(돌연변이 연산 후 생성되는 세트 기준)가 100세대가 되면 종료되도록 하였다[10].

**4. 본 연구의 평가 및 결론**

기존의 많은 위험 분석의 프로세스 속에서 보안 대책의 제시는 평가자에 주관적인 판단에 의해 제시되고 결정이 되었다. 하지만 주관적인 생각들이 많이 개입하게 되면서 대책 제시의 정확도가 떨어지고 판단이 흐려질 수 있는 단점이 있었다. 이러한 문제점을 유전자 알고리즘을 이용하여 보안 대책을 제시하여 보았으며, 이를 통하여 평가자의 평가를 지원할 수 있게 되었고, 제시된 보안 대책에 대하여 타당성이 부여되었다. 이 알고리즘을 통해 대책이 제시되었다 하더라도 평가자는 각 보안 대책의 구현이 조직의 위험 감소와 가지는 연관 관계를 이해하고 적절한 대책의 구현을 통해서 조직이 가지는 대책 구현에 따른 비용의 한도 문제도 고려하여 대책을 선택하여야 하고, 적은 비용을 통해 가장 큰 위험 감소의 효과를 얻을 수 있는 대책으로 선택하여야 할 것이다.

유전자 알고리즘 수행에 있어서 속도와 탐색 성능은 알고리즘 수행을 통해 생성된 해 집합에 대하여 최적 해를 비교하여 확인할 수 있다. 유전자 알고리즘은 일반적으로 탐색공간 내에서 유망한 영역을 빨리 찾아내는 데에 유리하다[11]. 또 이런 성능은 최적 해의 비용대비 효과에 비하여 유전 알고리즘을 통해 생성된 대책의 비용대비 효과를 비교함으로써 확인할 수 있고, 이러한 탐색 성능과 속도는 집단 크기와 교배확률, 돌연변이 확률에 의해 영향을 받는다.

본 연구에서는 목적함수가 하나를 사용하였으나 여러 개의 목적 함수를 사용하여 알고리즘을 수행함으로써 좀더 최적 해에 가까운 값을 구할 수 있을 것이다. 또한 본 연구에서 적용한 유전자 알고리즘을 이용하여 위험 분석 도구에서도 적용할 수 있을 것이다.

**참고 문헌**

- [1] 진강규, 유전 알고리즘과 그 응용, 교우사, 2002. p.282
- [2] Genetic Algorithms - Knapsack Problem <http://www.functologic.com/info/Knapsack.html>
- [3] Introduction to the Knapsack Problem <http://www.ifors.ms.unimelb.edu.au/tutorial/knapsack/introduction.html>
- [4] 이상용, 인공지능, 상조사, 1998. p.74
- [5] Ellis Horowitz, Sartaj Sahni, Fundamentals of Computer Algorithms, Computer Science Press Inc. Computer Software Engineering Series, pp. 198 ~ 200
- [6] 문병로, 유전 알고리즘, 두양사, 2003. p.150
- [7] Guanrong Chen, Young Hoon Joo, Introduction to Fuzzy Control Systems in Handbook of Intelligent Control, CRC Press, 2000. p.24
- [8] 장욱 외 2인, 유전 알고리즘을 이용한 퍼지-신경망의 모델링 및 비선형 시스템 모델링에의 응용, 한국 퍼지 및 지능 시스템학회 논문지, Vol. 9, No.1, 1999. pp. 71 ~ 81
- [9] 노기갑 외 2인, 유전 알고리즘을 이용한 규칙 기반 퍼지 제어기의 최적 설계, 전기학회논문지, 48A권, 2호, 1999. 2, pp.145~152
- [10] 김여근 외 2인, 메타 휴리스틱, 영지문화사, 2003. p.4
- [11] 공성곤 외 3인(역), 유전자 알고리즘 입문, 진영사, 1997. p.179