

진화탐색을 사용한 기능점수 예측

김미숙⁰ 권기태 강태원
강릉대학교 컴퓨터공학과
{mikiss⁰, ktkwon, twkang}@kangnung.ac.kr

FP measurement using Evolutionary Search

Kim-Mi Suk⁰ Kwon-Ki Tae Kang-Tae Won
Dept. of Computer Science & Engineering, Kangnung National University

요약

성공적인 소프트웨어 개발을 위해서는 프로젝트 계획 수립단계에서 정확한 예측 모델을 연구하는 것이 중요하다. 소프트웨어 개발의 중요성이 대두됨에 따라 현재에는 많은 기존의 프로젝트 데이터를 보관하게 되었다. 기존의 예측모델에서는 개발환경에 따라 서로 다른 형태의 모델을 만들어 개발비용을 예측하였다. 모델의 형태에 따른 신뢰도 또한 주요 이슈로 작용하였다. 이 논문에서는 이러한 많은 프로젝트 데이터와 현재 개발하고자 하는 프로젝트에 대하여 과거의 데이터 중 가장 유사한 최적의 프로젝트를 찾아내기 위해 FP(Function Point)를 이용하는 많은 프로젝트 데이터에 유전자 알고리즘을 적용하여 최적의 유사 모델을 찾아내는 방법에 대하여 제안하고자 한다.

은 이미 검증되었고, 세계 각국에서 일반적으로 널리 받

1. 서론

소프트웨어 개발의 중요성이 대두되면서 예측모델에 대한 연구가 활발히 진행되고 있다[3,4]. 이러한 예측을 위해 많은 프로젝트의 데이터 수집에 대한 관심도 증가되고 있는 실정이다. 데이터 집합은 지난 수년동안 여러 분야와 다양한 프로젝트에 따라 분류되고 수집되어 왔다. 소프트웨어 개발 대가를 산정하기 위한 소프트웨어 규모 측정(Measurement)을 위한 척도(Metric)로는 프로그램 라인수(LOC, Line of code)와 대상 정보시스템의 외부입력(EI), 외부출력(EO), 외부조회(EQ), 내부논리파일(ILF), 외부인터페이스파일(EIF)의 5개 기능으로 구성되는 기능점수(Function Point) 등을 대표적으로 들 수 있다. 그러나 LOC 방식은 종래의 3세대 언어 중심의 메인 프레임(Main Frame)용 소프트웨어 개발에서는 상당한 유효성이 있었으나 LOC 척도 자체의 문제점뿐만 아니라 최근의 4세대 언어를 사용하는 개발에서는 LOC 방법에 의한 예측이 어렵게 되었다.

이 논문에서는 기능점수(Function Point)에 의한 방식으로 저장된 데이터 집합에서 새로운 프로젝트 개발을 위한 예측을 위해 가장 유사한 과거의 데이터를 찾을 수 있는 검색방법을 제안하고자 한다. 최적의 해를 찾기 위한 검색방법은 여러 가지 알고리즘이 있으나 유전자알고리즘(Genetic Algorithm)을 사용한다. 논문의 구성은 다음과 같다. 2장에서는 기능점수의 계산 방법에 대하여 알아보고, 3장에서는 유전자알고리즘의 개요에 대하여 알아본다. 4장에서는 소프트웨어 비용산정에 쓰이는 기능점수에 유전자알고리즘을 적용한 경우의 표현방법을 알아보고, 5장에서 결론과 향후 방향에 대하여 알아본다.

2. 기능점수

기능점수 분석(Function Point Analysis : FPA)은 소프트웨어 개발 프로젝트 혹은 설치된 소프트웨어 어플리케이션의 크기를 측정하는 방법론이다[3]. 기능점수 분석

아들여진 방법론이다. 새로운 기술의 발전과 동시에 재검토되고 갱신되고 있다. 최근 IFPUG 계산 실무 위원회에서 발표한 지침서로 사용되고 있다. 기능점수의 계산 과정을 보면 다음과 같다.

2.1 기능점수의 계산

일반적으로 기능점수에 대한 계산과정은 다음과 같은 절차를 따른다.

1. 기능점수의 유형 결정
2. 기능점수 계산 범위와 어플리케이션 경계를 식별
3. 데이터기능(내부논리파일, 외부인터페이스파일)과 복잡도 계산
4. 트랜잭션기능(외부입력, 외부출력, 외부조회)과 복잡도 계산
5. 미조정된 기능점수(unadjusted function point)계산
6. 일반 시스템 특성에 근거한 값 조정 인자 계산
7. 조정된 기능점수(adjusted function point) 계산

기능점수의 계산은 제안된 혹은 구현된 시스템의 다섯 가지 컴포넌트를 세는 것을 시작으로 한다. 그 다섯 가지는 외부입력, 외부출력, 논리적내부파일, 외부인터페이스파일, 외부조회 등이다. 각각의 복잡도는 일반 시스템 특성(General System Character)에서 정의된 표준안에 따라 상대적으로 저, 중, 고의 세 가지로 분류된 가중치를 부여받아 기능수(Function Count)를 계산한다. 각각의 컴포넌트의 개수는 매트릭스에 의해 가중치 값을 부여받게 된다. 가중치 값의 합계는 미조정된 기능점수(UFP)로 계산된다. 14개의 "일반적인 시스템 특성"의 영향을 평가하여 값들을 모두 합하고 값조정인자(Value Adjustment Factor)를 구한다. 마지막으로 두 개의 값을 곱해서 기능점수의 값을 구하게 된다. 즉, 기능점수의 계산은 미조정된 기능점수와 값조정인자의 곱으로 만들어진다.

2.2 기능점수의 유형

기본적으로 미조정기능점수(UFP)에 값조정인자(VAF)를 곱한 것이 일반적인 기능점수이다[3]. 조금 더 상세히 분류하면 기능점수에는 신규 개발 기능점수, 기능 확장 기능점수, 어플리케이션 기능점수의 3 종류가 있다. 이들은 각각 사용 목적이 다르고 조정을 필한 기능점수의 계산식이 다르다. 신규 개발 기능점수 및 기능 확장 기능점수는 개발 대상의 정보 시스템 기능의 양을 계측하는 것이다. 외부 사양이 결정된 단계에서 계측할 수 있으며 이후의 어느 단계에서도 계측할 수 있다. 기능 요건이 명확하게 되면 기본적으로는 계측의 정밀도가 높아지게 되고 기대할 수 있다. 개발 완료 후에 실적의 기능점수를 계측한 경우도 있다. 견적이나 실적의 기능량을 계측하는 경우에는 신규 개발 및 기능 확장 기능점수를 계산한다.

2.2.1 신규 개발 기능점수

신규 개발 시스템의 기능점수(DFP)는 미조정기능점수(UFP)에 값조정인자(VAF)를 곱하여 구한다. 프로젝트에 따라서는 데이터의 전환이나 이에 관련된 문서 작성을 위한 기능(CFP)을 포함한다. 신규개발 기능점수의 식을 보면 다음과 같다.

$$\text{신규 개발 기능점수} = (\text{미조정 기능점수} + \text{데이터의 컨버전에 의해 포함되는 기능점수}) \times \text{값조정인자}$$

2.2.2 기능 확장 기능점수

기존 시스템의 기능 확장 기능점수(EFP)는 기능의 추가(ADD), 수정(CHG-A), 삭제(DEL)하는 기능의 양을 각각 계측한다. 프로젝트에 따라서는 데이터의 컨버전이나 이에 관련된 문서 작성을 위한 기능(CFP)을 포함한다. 값조정인자도 확장 전(VAF-B ; Before)과 확장 후(VAF-A : After)로 나누어 평가하며 이를 식으로서 나타내면 다음과 같이 된다.

$$\text{기능확장 기능점수} = [(\text{추가} + \text{수정} + \text{문서작성을 위한 기능}) \times \text{확장후 값조정인자}] + (\text{삭제} \times \text{확장전 값조정인자})$$

위 식에서도 알 수 있듯이 삭제 분에 대해서도 기능점수로 가점을 하고 있다. 실제로 작업 공수를 어떻게 산출하는가에 관해서는 다양한 견해가 있어 실제 프로젝트에서는 사용자가 현실적으로 많은 고민을 하게 될 것으로 생각된다.

2.2.3 어플리케이션 기능점수

컷 오버된 시스템의 기능점수(AFP)에서 현재 그 애플리케이션이 가지고 있는 기능의 양을 나타낸다. 이 값은 신규 개발이 완료된 시점에서 초기화된다. 기능 확장에 의해 시스템의 기능이 변하면 다시 계측을 하는 것은 당연한 것이다. 이것은 어플리케이션의 자산으로서의 기능이 양을 계측하는 것이다.

먼저, 신규 개발 완료후의 어플리케이션 기능점수는

$$\text{어플리케이션 기능점수} = \text{개발프로젝트에 의해 설치된 기능의 미조정된 기능점수} \times \text{값조정인자}$$

가 되며 이행을 위한 문서 작성을 위한 기능은 갖지 않는다. 그리고 기능 확장 후의 어플리케이션 기능점수는 다음과 같다.

$$\begin{aligned} \text{어플리케이션 기능점수} = & [(\text{확장전 미조정기능점수} + \text{추가} + \text{수정후 확장된 미조정기능점수}) \\ & - (\text{수정전 확장된 미조정기능점수} + \text{삭제})] \times \text{확장후 값조정인자} \end{aligned}$$

이와 같이 프로젝트의 성격에 따라 서로 다른 계산 방법에 의해 기능점수가 계산된다. 이 논문에서는 미조정된 기능점수까지의 프로젝트 집합을 모집단으로 사용한다.

3. 유전자 알고리즘의 개요

유전자 알고리즘은 자연선택과 유전자에 기초를 둔 일종의 탐색 알고리즘이다. 자연계(system)에 있는 생물의 진화과정에 있어서, 어떤 세대(generation)를 형성하는 개체(individual)들의 집합, 즉 개체군(population)중에서 환경에 대한 적합도(fitness)가 높은 확률로 살아남아 재생(reproduction)할 수 있게되며, 이 때 교차(crossover) 및 돌연변이(mutation)로서 다음 세대의 개체군을 형성하게 된다. 유전자 알고리즘은 선택, 교차, 및 돌연변이라는 세 개의 연산을 포함하는데 이들의 수행은 모두 임의성을 포함한다.

3.1 유전 알고리즘의 동작 및 특징

유전자 알고리즘의 처리과정은 초기화 과정에서 한 모집단의 각 개체가 탐색공간에 임의적으로 흩어지게 한다. 다음으로 각 개체가 자식을 생산하는 기회를 제어하기 위하여 자연의 적자 생존의 메커니즘을 모델링 한 선택 연산을 수행한다[7]. 선택의 기반이 되는 것은 적합도 함수이며 여러 가지 선택방법들이 존재하지만 그 기본 원리는 더 좋은 개체들에게 더 많은 자식을 생산할 특권을 부여한다는 것에 있어서 공통적이다. 세 번째로 교차 연산을 수행하는데, 교차는 부모의 우수한 유전자를 자식에게 물려주기 위해 상위 개체들간에 유전자를 교환하는 것이다. 즉, 어떤 지식도 사용하지 않고 다만 선택된 개체에는 좋은 유전자가 많을 것이고, 이들을 섞으면 더 우수한 개체가 나올 것이라는 논리이다. 마지막으로 돌연변이는 부모 개체들로부터 물려받을 수 없는 유전자의 생성을 가능하게 하여 지금과는 다른 우수한 개체의 탄생 유도한다[6]. 이러한 과정을 반복하여 최적의 해를 찾아 나간다. 즉, 모집단을 진화시킨다. 지금까지의 논의에 따른 단순 유전자 알고리즘의 처리과정을 보면 다음과 같다[그림 3.1].

```

t=0
Initialize P(t)
evaluate P(t)
while(not termination-conditionk)
    t = t + 1
    select P(t) from P(t-1)
    crossover P(t)
    mutate P(t)
    evaluate P(t)
}
    
```

[그림 3.1] 단순 유전자 알고리즘 처리과정

이와 같은 특징으로 인해 다른 탐색 또는 최적화 방법 중 하나인 계산에 의존한 방법(calculus-based method :

hill-climbing)에 비하여 전역적 해를 구할 가능성이 높으며 다른 여러 탐색 방법에 비하여 효율적이다[5].

$$\text{적합도} = \sum_{\text{컴포넌트}=1}^{15} |\text{계산값} - \text{실제값}|$$

4. 기능점수를 이용한 비용산정에 쓰이는 유전자 알고리즘

어떤 문제에 유전자알고리즘을 적용하기 위해서는, 문제를 어떻게 유전형으로 표현하고, 연산자는 어떻게 구현하며, 적합도는 어떻게 계산하는가를 정해야 한다[1,2]. 여기서는 현재의 문제와 가장 유사한 기존의 기능점수와 의 최적화를 위한 유전자 알고리즘을 제안한다.

기능점수 계산에 대한 속성을 개체의 유전형으로 표현했으며, 유전연산은 복잡도의 변이를 나타내는 돌연변이만을 적용하였다.

4.1 기능점수를 위한 개체의 유전형

기능점수는 기본적으로 5개의 컴포넌트를 기반으로 저, 중, 고로 3개의 복잡도의 곱에 대한 합으로 나타낸다. 5개의 컴포넌트는 저, 중, 고에 따라 가변적으로 변할 수 있다. 그러므로 유전자알고리즘을 적용하기 위한 한 개체에 대한 염색체는 다음과 같이 표현된다[그림 4.1].



[그림 4.1] 기능점수에 대한 염색체

기능점수를 이루는 5개의 컴포넌트인 외부입력, 외부출력, 논리내부파일, 외부인터페이스파일, 외부조회 각각의 복잡도를 하나의 염색체에 기록한다. 여기서 저, 중, 고는 이미 IFPUG에서 일반적으로 사용하는 복잡도를 따른다.

4.2 기능점수를 위한 유전연산

기능점수에 의한 데이터 집합에 현재 진행하고자 하는 프로젝트의 각 컴포넌트 수와 일치하는 것이 없는 경우도 있으므로 다양한 모집단을 형성하기 위해 각 컴포넌트별로 돌연변이를 적용하는 것이 적합하다.

돌연변이는 개체의 각 유전자좌의 유전자에 대하여 일정한 돌연변이 확률을 적용하여 대립 유전자의 값으로 바꾸는 것이다. 개체에 근접한 새로운 개체를 생성하는 국소적인 랜덤 탐색의 일종이다. 또한 집단에서 잃어버린 유전형질을 복구하여 다양성을 유지하기 위한 수단으로도 사용된다. 이때 전형적인 돌연변이 확률은 0.05 이하이다.

이러한 여러 가지 돌연변이 연산에 의해 집단에서 잃어버린 유전형질을 복구하여 다양성을 유지하기 위한 수단으로도 사용된다.

4.3 기능점수를 위한 적합도

모집단에서 가장 최적의 유사한 개체를 찾기 위해 각 컴포넌트별로 기입된 컴포넌트 수를 현재 수행하고자 하는 컴포넌트의 개체 수에서 차감한 합을 적합도로 사용한다. 기능점수 예측에 쓰이기 위한 적합도는 다음과 같이 계산된다.

적합도의 값이 작을수록 가장 유사한 프로젝트의 기능 점수에 근접하게 된다.

4.4 기능점수를 위한 선택

기능점수를 위한 선택은 높은 적합도를 나타내는 것에 대한 비례 선택을 사용한다. 적합도는 0에 가까울수록 좋은 결과를 나타내므로, 비례 선택시 역수를 취한다.

5. 결 론

성공적인 소프트웨어 개발을 위해서 데이터 집합은 지난 수년동안 여러 분야와 다양한 프로젝트에 따라 분류되고 수집되어 왔다. 소프트웨어 개발 대가를 산정하기 위한 소프트웨어 규모 측정(Measurement)을 위한 척도(Metric)로는 프로그램 라인수(LOC, Line of code)와 대상 정보시스템의 입력, 출력, 조회, 내부 논리 파일, 외부 인터페이스 파일로 구성되는 5개 기능점수를 대표적으로 들 수 있다.

이 논문에서는 기능점수(Function Point)에 의한 방식으로 저장된 데이터 집합에서 새로운 프로젝트 개발을 위한 예측을 위해 가장 유사한 과거의 데이터를 찾을 수 있는 검색방법을 제안하였다. 정확한 예측을 위한 많은 프로젝트를 분류하고 정리하는 작업 또한 많은 비용과 노력이 들 것이다. 데이터 집합에서 가장 유사한 최적의 프로젝트를 찾는 것 또한 가장 중요한 작업일 것이다.

6. 참 고 문 헌

- [1] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1989.
- [2] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, pp.160-168, 1995.
- [3] 심기보, "기능점수법을 이용한 다중 회귀 분석적 소프트웨어 대가 산정 모델, 송실대학교, 박사학위논문, 2002
- [4] 이상운, 강정호, 박중양, "기능점수를 이용한 소프트웨어 개발노력 추정", 정보처리학회논문지D, 제9-D권, 제4호, pp603-612, 2002
- [5] 장병탁, "인공 진화에 의한 학습 및 최적화," 제어·자동화·시스템공학회지, 제1권, 제3호, pp. 52-61, 1995.
- [6] 진강규, 유전알고리즘과 그 응용, 교우사, p119-210, 2000
- [7] <http://rics.cie.cau.ac.kr/EAs/content.html>