# 터치 스크린 유형 한글자판을 위한 유전자 알고리즘

강태원°, 한수경
강릉대학교 컴퓨터과학과
twkang@kangnung.ac.kr
skhan@kangnung.ac.kr

# Genetic Algorithm for Korean Keyboard of Touch Screen Style

TaeWon Kang°, SooKyung Han
Dept. of Computer Science, Kangnung National University

## 요 약

오늘날 전자수첩에서부터 셀룰라 폰(휴대폰), PDA, 그리고 HPC 등에 이르기까지 휴대형 정보기기의 사용이 급증하고 있다. 이러한 장비는 한글을 입력하기 위한 한글자판을 제공하는 경우가 대부분인데, 전자 수첩과 같이 물리적 자판이 제공되는 경우와 터치 스크린 상에 소프트웨어적으로 제공되는 경우가 있으며, 두 경우 모두 자판의 크기 때문에 한 두 손가락만을 이용하여 글자를 입력하게 된다. 이 논문에서는 이러한 경우, 보다 빠르게 한글을 입력할 수 있는 한글 자판 배열을 유전자 알고리즘을 이용하여 찾는 것에 대하여 연구한다. 실험 결과 유전자 알고리즘을 이용하여 생성한 자판이 기존의 자판에 비하여 평균적으로 약 33% ~ 49% 정도 빠르게 글자를 입력할 수 있다.

## 1. 연구 배경

전자수첩, 셀룰라 폰, PDA, 그리고 HPC 등과 같은 휴대형 정보기기 사용이 증가되는 정보화 사회에서, 인간의 문자 생활은 손으로 글씨를 쓰는 것에서 벗어나 점점 자판을 통한 입력에 의존하게 되었다. 오늘날 가장 보편적으로 사용되고 있는 표준 한글 자판 배열은 한국공업표준협회가 한국과학기술원에 용역을 주어 개발 공표한 KSC-5715이다[1]. 이 자판은 다양한 기존의 타자기 자판과 비교하여, 받침을 치기 위해 윗글쇠(Shift-Key)를 쳐야하는 등 많은 비효율적인 요소를 가지고 있다고 알려졌지만[2], 컴퓨터가 타자기를 대체하면서 거의 대부분의 사람들이 이 자판에 익숙해져 있다. 그러한 이유에서 위의 휴대형 정보기기는 모두 표준 자판 배열을 따르고 있다.

그러나, 이 자판은 양손을 모두 사용하는 경우를 전제로 한 것인 반면에, 휴대형 정보기기는 크기가 작기 때문에 자판의 크기가 작고, 따라서 일반적으로 스타일러스 펜이나 손가락 한 개를 이용하여 입력하게된다. 이 논문에서는 이러한 경우 가장 빠르게 글자를 입력할 수 있는 한글 자판 배열을 유전자 알고리즘을 이용하여 찾아본다. 물론 현재 자판에 익숙한 사용자 입장에서 자판을 새로이 외워야하는 문제점을 생각할 수 있으나, 휴대형 정보기기 자판의 작은 크기 때문에 자판이 한 눈에 들어와서 조금만 익숙해지면 외우지 않고도 효율적으로 사용할 수 있을 것이다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 한글 자판 배열 문제와 유전자 알고리즘에 대하여 살펴보고, 3장에서는 터치 스크린 유형 한글 자판 배열을 위한 유전자 알고리즘에 대하여 설명하고, 4장에서는 실험을 통하여 결과를 분석하고, 마지막으로 5장에서 결론을 맺는다.

## 2. 한글 자판 배열과 유전자 알고리즘

한글 자판 배열을 결정한다는 것은 한글 자소를 자판의 글쇠 위에 배치시키는 것이다. 즉, 한글 자소를 정의역으로 하고 글쇠 조합을 치역으로 하는 적절한 함수를 찾는 문제이다. 여기서 글쇠 조합이라는 것은 윗글쇠가 안 눌린 상태의 글쇠-자판의 각 키-집합에 윗글쇠를 누른 상태의 글쇠 집합을 더한 것을 말한다[3]. 먼저, 영문 자판의 경우 하나의 자판 배열은 영문 자소 52자(대문자와 소문자를 구분)를 26개의 글쇠에 대응시키는 함수에 해당한다. 그렇지만, 현실적으로 대문자와 소문자를 다른 글쇠에 배정한다는 것이 좋지 않기 때문에, 대문자와 소문자를 동일한 글쇠에 배정한다면 가능한 영문 자판 배열의 수는 다음과 같은 정의역과 치역을 갖는 단사 함수의 개수와 같다.

$$f : A \rightarrow K,\ A = \text{영문자소집합},\ K = \text{글쇠집합},$$
$$|A| = 26, \qquad |K| = 26$$

즉, 영문 자판 배열 문제는 가능한 26!의 배열 중에서 하나를 선택하는 문제다.

영어 자소와는 다르게 한글 입력을 위한 한글 자소 집합이 정해져 있지 않으며, 예전의 3벌식, 4벌식, 그리고 5벌식 타자기들에서 알 수 있듯이, 한글 자판을 위한 글쇠 조합 역시 정해져있지 않기 때문에 한글 자판 배열 문제는 훨씬 복잡하다. 예를 들어, 2벌식 자판인 KSC-5715는 33개의 자소 집합을 사용하고 3벌식 자판은 52개의 자소 집합을 사용하고, 북한의 표준 자판은 26개의 자소 집합을 사용한다. 한글 자판 배열 문제도 영문 자판 문제와 마찬가지로 한글 자소 집합에서 글쇠 조합으로 대응되는 함수를 찾는 문제인데, 한글의 경우 정의역과 치역이 정해지지 않았기 때문에 더 많은 경우의 수를 따져야 한다.

한글을 구성하는 자소 집합은 다음과 같으며, 한글 자소 전체 집합은 초성(단자음+쌍자음) 19자, 중성(단모음+복모음) 21자, 그리고 종성(단자음+복자음+{ㄲ,ㅆ}) 27자를 더하여 모두 67가 된다.

* 단자음(14개): {ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ,ㅅ,ㅇ,ㅈ,ㅊ,ㅋ,ㅌ,ㅍ,ㅎ}
* 단모음(10개): {ㅏ,ㅑ,ㅓ,ㅕ,ㅗ,ㅛ,ㅜ,ㅠ,ㅡ,ㅣ}
* 쌍자음(5개): {ㄲ,ㄸ,ㅃ,ㅆ,ㅉ}
* 복자음(11개): {ㄳ,ㄵ,ㄶ,ㄺ,ㄻ,ㄼ,ㄽ,ㄾ,ㄿ,ㅀ,ㅄ}
* 복모음(11개): {ㅐ,ㅒ,ㅔ,ㅖ,ㅘ,ㅙ,ㅚ,ㅝ,ㅞ,ㅟ,ㅢ}

여기서, 기본 자소인 단자음과 단모음을 제외한 자소는 기본 자소의 조합으로 표현이 가능하기 때문에, 한글 자판을 위한 자소 집합의 크기는 최소 25(기본 자소+ 문자 완성을 위한 메타문자)에서 최대 67(세벌식 자판의 경우)이다[3]. 그렇다면, 한글 자판 배열에서 가능한 자소 집합의 개수는, 2벌식과 3벌식의 경우로 제한 하더라도, $2^{67-25} = 2^{42}$에 이르는 엄청난 양이 된다[4]. 결국 영문 자판 배열 문제의 경우 자소 집합 즉, 정의역이 1개인데 반하여, 한글 자판 배열의 경우 정의역의 가짓수만 하더라도 $2^{42}$가지나 된다.

결론적으로, 한글 자판 배열 문제는 경우의 수가 대단히 많은 최적화 문제에 해당하는 것이며, 이러한 문제에 유전자 알고리즘이 매우 효과적으로 응용될 수 있다.

## 3. 한글 자판 배열을 위한 유전자 알고리즘

유전자 알고리즘(Genetic Algorithm)은 자연선택과 유전자에 기초를 둔 일종의 탐색 알고리즘이다[5]. 이것은 탐색 공간에 대한 어떠한 지식도 사용하지 않으면서도 알고리즘이 간단하고, 강하며(robust), 또한 일반적이기 때문에 많은 분야에서 응용되고 있다[6]. 자연은 제한된 자원에 대한 개체간의 경쟁 시에 적응성이 강한 개체가 더 많이 살아남을 수 있게 하는 것으로 알려져 있다. 또한, 그들의 상대적인 우수성은 유전자로 특징 지워지며 이들은 다음 세대에 전달되어 결국 우수한 개체의 우수한 특성이 세대를 건너가며 유지되는 것이다.

일반적으로 단순 유전자 알고리즘은 한 개의 모집단을 구성하는 것으로 시작된다. 여기서 모집단 내의 개체는 문제에 대한 잠정적인 해를 나타내는 것이며, 초기 모집단 내의 개체는 임의로 정한다. 초기 모집단이 생성되면 각 개체를 평가하여 그들의 상대적 적합도를 계산하고, 그 값에 근거하여 다음 세대에 자손을 남길(즉, 자신의 유전자를 다음 세대에 넘길) 개체를 선택한다. 선택된 개체들은 자신의 유전자를 모두 전달하거나, 교차를 통하여 다른 개체와 부분적으로 혼합된 유전자를 전달하거나, 돌연변이를 통하여 자신의 것과 다른 유전자를 포함하도록 전달한다. 그렇게 만들어진 새로운 개체들로 구성된 모집단이 다음 세대를 형성하는 것이다. 이러한 진화 과정을 되풀이하면 모집단의 개체들을 적합도가 높은 쪽으로 수렴하게 된다. 그리고 최종적으로 가장 적합도가 높은 개체를 최적해로 사용한다.

### 3.1 한글 자판 배열 개체의 표현

어떤 문제에 유전자 알고리즘을 적용하기 위해서는 개체의 표현방법, 모집단 구성 방법과 적용할 진화 연산자 및 유전 연산자를 정하여야 한다.

다양한 한글 자소 집합과 글쇠 조합 집합에 대한 자판 배열을 표현하기 위해서는 기본적으로 자판 배열을 나타내는 개체의 유전자 개수가 가변적이어야 한다. 이 논문에서는 서술의 편의를 위하여 현재의 자판에 나타나는 한글 자소와 글쇠를 중심으로 설명한다. 즉, 한글 자소를 현재 자판에서 영문자와 구두점이 위치한 30개의 글쇠에 배정하기 위하여 한글 자소 집합은 기본 자소 24자에 세로 모음 ㅐ,ㅒ,ㅔ,ㅖ와 쌍자음을 더하여 총 33자로 하되, 쌍자음은 해당 단자음의 상단에 위치토록 하며, 복모음 ㅙ와 ㅞ는 각각 ㅚ와 ㅝ의 상단에 위치토록 한다. 즉, 한글 자소는 26개이며, 여기에 추가로, 상단의 자소를 입력하기 위한 윗글쇠(Shift-Key), 공백 문자(Space-Bar), 그리고 구두점 ",","와"."를 포함하도록 한다. 결국 자소는 모두 30개이며 글쇠 역시 30개다(그림 3.1). 그림에서 글쇠 위 숫자는 글쇠 번호를 나타낸다.

자소 집합 = {ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ,ㅅ,ㅇ,ㅈ,
ㅊ,ㅋ,ㅌ,ㅍ,ㅎ,ㅏ,ㅑ,ㅓ,ㅕ,ㅗ,ㅛ,ㅜ,ㅠ,
ㅡ,ㅣ,ㅐ,ㅔ,Shift,",",".", ␣}

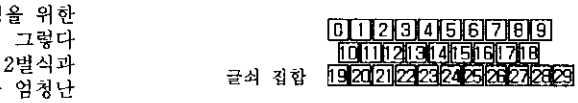


그림 3.1 자소 집합과 글쇠

이러한 전제하에 모집단 내의 각 개체는 하나의 자판 배열을 나타내는 것으로 각각 30개의 유전자를 가지며, 각 유전자는 글쇠 번호를 값으로 갖는데, 그 값은 현재 개체가 나타내는 자판에서 해당 자소의 위치를 나타낸다.

유전자 알고리즘에서 개체의 표현법이 정해지면 그러한 개체들의 모임인 모집단을 구성하게 된다. 단순 유전자 알고리즘에서 모집단은 일정한 개수(모집단의 크기)의 임의의 개체들의 모임이며, 모집단 내의 개체들은 단지 같은 세대 구성원이라는 것 이외의 어떠한 관계도 가지고 있지 않다.

### 3.2 적합도 및 유전 연산자

유전자 알고리즘을 적용할 때 사용하는 진화 연산자 즉, 선택 연산자 및 교차와 돌연변이로 대표되는 유전 연산자의 종류는 매우 다양하며, 각각의 방법은 나름대로의 특징을 갖는다[7,8]. 선택 연산을 위해서는 먼저 각 개체의 적합도를 평가해야 하는데, 한글 자판 배열 문제에서 개체의 적합도 즉, 자판의 적합도는 임의의 문서 입력을 위하여 자판 위를 이동한 총 이동거리로 한다. 글쇠들 사이의 이동거리는 임의로 정한 자판에서의 실제 거리를 측정하여 사용한다.

이 문제에서는, 순회 외판원 문제에서와 같이, 한 개체의 유전자 값은 유전형 내에 두 번 이상 나올 수 없다[6]. 따라서, 일반적인 교차 연산을 수행하면 안되고, PMX 등과 같이 교차 연산 후에 새롭게 생겨난 자식 개체 역시 정당한 자판이 되는 연산자를 사용해야 한다. 돌연변이 연산을 적용하는 경우도, 어떤 방법을 사용해도 무관하지만 돌연변이 후의 결과가 항상 정당한 자판이 되는 연산자를 사용해야 한다. 임의의 두 유전자를 교환하는 치환이나, 연속된 유전자 토막에서 유전자 값의 순서를 반대로 하는 전위 등의 돌연변이 연산자를 사용하면 된다.

## 4. 실험 및 분석

유전자 알고리즘을 이용하여 생성한 한글 자판의 효율성을 평가하기 위하여 실험을 수행하였다. 실험은 3장에서 설명한 상황을 구현하여 수행하였다. 즉, 현재의 2벌식 자판에서 사용되는 26개의 한글 자소(실제로는 위치가 강제적으로 지정되는 7개의 자소를 포함하여 33개의 한글 자소를 표현한다.)에 4개의 자소를 추가한 총 30개의 자소를 그림 3.1과 같이 배치된 30개의 글쇠에 배열하는 경우에 대하여 실험하였다.

각 개체 즉, 자판의 적합도를 평가하기 위해서 정치, 경제, 사회에 관한 글, 관동별곡 전문, 시, 소설, 노래 가사, 및 채팅 대화록 등 총 8개의 성격이 다른 문서를 사용하였다. 각각의 문서에 가장 적합한 자판 배열 8개를 생성한 후, 나머지 7

개의 문서에 대하여 효율을 평가하고, 현재의 표준 자판과 비교하였다. 또한, 한글 자소가 영어와는 다르게 자음과 모음으로 구성되고, 한글이 초성, 중성, 종성으로 되어있는 점을 고려하여 자음은 자판의 좌측에, 모음은 우측에만 위치하도록 하는 실험을 수행하여 결과를 비교하였다. 모든 경우에 모집단의 크기는 20, 교차율은 0.3, 돌연변이율은 0.01, 반복횟수는 2000으로 하였다. 대부분의 경우 0.5이하의 낮은 교차율에서 더 좋은 결과가 나왔으며, 대부분 2000번 이후에는 아주 오랜 세대가 지나야 더 좋은 개체가 찾아졌다.

다음 <그림 4.1>은 8개의 문서를 이용하여 생성한 8개의 자판과 현재의 표준 자판에서의 자소간 평균 이동거리를 비교한 것이다. 자소간 평균 이동거리는 문서 전체의 자소를 입력하는데 필요한 전체 이동 거리를 문서에 포함된 자소 개수로 나눈 것이다. 실험에 사용한 자판은 글쇠 중심 사이의 수평 방향 거리가 1.3cm, 수직 방향 거리가 1.4cm정도인 가상의 자판을 사용하였다.

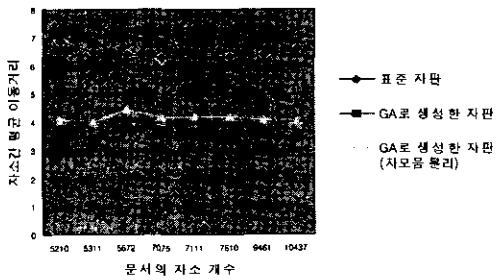


그림 4.1 각 문서별 자판의 자소간 평균 이동거리

그림에서, 예를 들어 자소 수가 5210인 경우가 나타내는 것은 자소 수가 5210개인 문서를 사용하여 생성한 자판의 자소간 평균 이동거리, 동일한 문서에 대한 표준 자판의 자소간 평균 이동거리이다. 그림에서 알 수 있는 것처럼 유전자 알고리즘을 이용하여 생성한 자판들 전체의 자소간 평균 이동거리(=3.59)가 기존 자판의 모든 문서에 대한 자소간 평균 이동거리(=6.455)에 비하여 글자를 입력하기 위한 이동거리가 약 45%가량 단축된다. 표준 자판과 같이 자음과 모음을 각각 자판의 좌측과 우측에만 할당되도록 한 자판들의 자소간 평균 이동거리(=4.14) 역시 약 36%정도 효과적임을 보인다. 그리고, 자판 생성에 사용한 문서의 자소 개수에도 영향을 받지 않음을 알 수 있다.

다음으로 각각의 문서를 이용하여 생성된 자판들에 대하여 다른 문서를 입력할 때의 자소간 평균 이동거리를 조사하였다(표 4.1). 전체적으로 특정 문서에 종속됨이 없다는 것을 알 수 있다. 표에서 회색 부분은 각 자판을 생성할 때 사용한 문서에 대한 자소간 평균 이동거리를 나타내며, 테두리가 진한 부분은 모든 문서에 대한 최소의 자소간 평균 이동거리를 값을 나타낸다. 결국, 자음과 모음을 분리하지 않는 경우는 자판6, 분리하는 경우는 자판2*가 가장 효과적인 것으로 나타난다. 다음 <그림 4.2>는 표준 자판과 유전자 알고리즘으로 생성한 한글 자판을 나타낸다.

## 5. 결론

휴대형 정보기기는 크기가 작기 때문에 자판의 크기가 작고, 따라서 일반적으로 스타일러스 펜이나 손가락 한 개를 이용하여 입력하게된다. 이 논문에서는 이러한 경우 가장 빠르게 글자를 입력할 수 있는 한글 자판 배열을 유전자 알고리즘을 이용하여 찾아보았다. 실험 결과 유전자 알고리즘을 이용하여 생성한 자판이 기존의 자판에 비하여 평균적으로 약 33% ~ 49% 정도 빠르게 글자를 입력할 수 있음을 알았다. 연구진은 현재 양손을 모두 사용하는 일반적인 자판과, 컴퓨터 키보드가 아닌 자소 집합과 글쇠 조합에 유전자 알고리즘을 적용하는 연구를 수행 중이다.

표 4.1 각 자판의 문서별 자소간 평균 이동거리

| 표준 자판 | | 문서1 | 문서2 | 문서3 | 문서4 | 문서5 | 문서6 | 문서7 | 문서8 | 평균 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 6.38 | 6.28 | 6.38 | 6.33 | 6.60 | 6.49 | 6.33 | 6.85 | 6.46 |
| 유전자 알고리즘에 의한 자판 | 자판1 | 3.59 | 3.66 | 3.72 | 3.59 | 3.57 | 3.55 | 3.68 | 3.60 | 3.62 |
| | 자판2 | 3.92 | 3.59 | 3.97 | 3.55 | 3.77 | 3.75 | 3.78 | 3.76 | 3.78 |
| | 자판3 | 3.88 | 3.81 | 3.59 | 3.85 | 3.80 | 3.79 | 3.86 | 3.81 | 3.83 |
| | 자판4 | 3.67 | 3.67 | 3.76 | 3.59 | 3.55 | 3.56 | 3.58 | 3.63 | 3.62 |
| | 자판5 | 3.45 | 3.62 | 3.67 | 3.56 | 3.59 | 3.31 | 3.63 | 3.52 | 3.53 |
| | 자판6 | 3.40 | 3.54 | 3.67 | 3.48 | 3.41 | 3.59 | 3.50 | 3.40 | 3.47 |
| | 자판7 | 3.65 | 3.82 | 3.86 | 3.66 | 3.59 | 3.56 | 3.59 | 3.62 | 3.67 |
| | 자판8 | 3.70 | 3.63 | 3.87 | 3.70 | 3.62 | 3.58 | 3.58 | 3.59 | 3.65 |
| | 자판1* | 3.59 | 4.25 | 4.20 | 4.15 | 4.07 | 4.10 | 4.10 | 4.35 | 4.16 |
| | 자판2* | 3.98 | 3.59 | 4.22 | 4.12 | 4.01 | 3.85 | 4.07 | 4.07 | 4.06 |
| | 자판3* | 4.24 | 4.32 | 3.59 | 4.44 | 4.36 | 4.37 | 4.40 | 4.52 | 4.38 |
| | 자판4* | 4.21 | 4.23 | 4.48 | 3.59 | 4.17 | 4.08 | 4.22 | 4.10 | 4.20 |
| | 자판5* | 4.09 | 4.21 | 4.36 | 4.09 | 3.59 | 3.98 | 4.22 | 4.09 | 4.13 |
| | 자판6* | 4.16 | 4.23 | 4.48 | 4.20 | 4.21 | 3.59 | 4.06 | 4.11 | 4.18 |
| | 자판7* | 4.39 | 4.30 | 4.49 | 4.29 | 4.32 | 4.22 | 3.59 | 4.30 | 4.31 |
| | 자판8* | 4.05 | 4.09 | 4.25 | 4.06 | 4.05 | 3.93 | 3.98 | 3.59 | 4.06 |

*: 자음과 모음을 분리한 자판

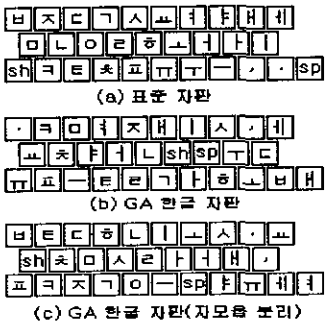


그림 4.2 표준 자판 및 GA 한글 자판 배열

## 6. 참고 문헌

[1]. 한국공업표준협회, KSC 5715 정보처리용 건반배열, 한국공업표준협회, 1982.

[2]. 이만영, 표준한글자판 문제 해결을 위한 정책결정 모형의 개발, 통신학술 연구과제, 1992.

[3]. 정승훈, 박진우, 이일병, 컴퓨터 모의 실험에 의한 자판 배열의 성능 평가, 제 3 회 한글 및 한국어 정보처리 학술대회 논문집, 1991.

[4]. 오길록, 최기선, 박세영, "한글공학", 3장, 대영사, 1994.

[5]. D. E. Goldberg , "Genetic Algorithms in Search, Optimization & Machine Learning,, pp. 1-25,1989.

[6]. Z. Michalewicz, "Genetic Algorithms+Data Structures = Evolutionary Programs",pp. 209-237, 1999.

[7]. Hancock, Selection Methods for Evolutionary Algorithms, in Practical Handbook of Genetic Algorithms, Vol. 2, pp. 67-92, 1995.

[8]. Pawlowsky, Crossover Operator, in Practical Handbook of Genetic Algorithms, Vol. 1, pp. 101-114, 1995.