

유전자 알고리즘을 이용한 고정 셀에서 글자 폰트(font) 최적화

김상원*, 김승희*, 김우제**

Word Processor font optimization in Fixed-function cell Using a Genetic Algorithm

Sang-Won Kim *, Seung-Hee Kim *, Woo-Je Kim**

요약

본 연구는 유전자알고리즘을 사용하여 표의 크기에 맞게 가장 최적화된 글자로 표현할 수 있는 방법을 실험하였다. 그 결과 셀의 넓이와 높이, 입력받을 글자의 개수를 계산하여 글자 크기, 줄 간격, 자간 간격의 최적의 값을 찾아 길이가 서로 다른 글자를 최적화된 상태로 표현할 수 있도록 폰트를 제공할 수 있었다. 본 연구는 유전자알고리즘을 통하여 현재 사용하고 있는 다양한 워드프로세스에서 발생되고 있는 셀 고정 상태에서의 글자 최적화 문제에 대한 해결 방법을 제시하였는데 그 의의가 있다.

▶ Keywords : 유전자알고리즘, 폰트, 최적화, 한글, 워드프로세스, 셀 최적화

Abstract

This study was conducted to explore a method of displaying optimized letters that fit the size of tables using a genetic algorithm. As a result, fonts with optimized letters of different lengths were offered through optimum values of the font size, line spacing, and letter spacing by calculating the width and height of the cell and number of letters to be entered. This study is significant in that it provides a solution to letter optimization issues in fixed cells that occur in various word processors that are currently used, through the genetic algorithm.

▶ Keywords : GA, font, Optimization, Hangeul, word process, cell optimization

•제1저자 : 김상원 •교신저자 : 김우제

•투고일 : 2013. 7. 25, 심사일 : 2013. 8. 19, 게재확정일 : 2013. 8. 27.

* 서울과학기술대학교 IT정책전문대학원 산업정보시스템전공 박사과정(Graduate School of Public Policy and Information Technology, Seoul National University of Science and Technology)

** 서울과학기술대학교 기술경영융합대학 글로벌융합산업공학과(School of Global Convergence of Industrial Engineering, Seoul National University of Science and Technology)

※ 본 논문은 2012년 한국경영과학회 추계학술대회 / 2012년 방위사업청 무기체계 시험 평가 세미나에 발표한 논문을 확장한 것입니다.

I. 서론

우리가 사용하고 있는 워드프로세서의 표 기능에서 셀 고정기능을 이용해야하는 경우가 종종 발생한다. 그런데 셀 고정 기능을 사용해서 글자를 입력하게 되면 셀 범위를 벗어나는 경우가 발생한다. 이런 경우 흔히 글자크기를 줄이거나 줄 간격 또는 자간 간격을 일일이 조정하여 글자가 셀 내부에 적절히 표현되도록 편집한다. 대개 이런 경우의 편집은 편집자마다 다르게 나타난다.

예컨대 글자크기를 조정하는 것을 선호하는 경우도 있고 줄 간격, 자간간격을 적절히 조정하는 경우도 있다. 각각의 경우 모두 편집자가 선호하는 방식으로 편집하게 된다. 이러한 방법의 사용자로 하여금 불편함을 느끼게 한다. 즉, 사용자가 글자 크기, 줄 간격, 자간 등, 글자 속성을 여러 번 조정해서 맞춰야 하며, 사용자마다 느끼는 최적의 글자 속성이 다르게 나타날 수 있다. 글자크기를 일정비율로 줄이거나 크게 하는 상대크기 조정기능은 모든 글자속성이 동시에 적용된 비율로 조정되어 사용자가 원하는 편집방향과 다를 수 있다.

표에서 셀의 크기를 고정한 후 길이가 다른 글자들을 표시하기 위해서, 대부분의 표기능이 있는 워드프로세서, 엑셀과 같은 스프레드시트, 기타 리포팅 툴들에서는 표에서 글자의 상대크기를 조정하는 기능, 표 글자 고정기능, 장평 고정 등의 기능을 제공하고 있다.

이러한 기능들은 자동으로 글자크기가 셀의 크기에 맞춰지는 능동형 방식과 입력된 글자가 셀 범위를 벗어나면 글자 크기조정여부를 확인한 후 글자의 크기를 조정하는 대화형 방식으로 나눌 수 있다. 이 기능은 크기가 고정된 셀의 크기를 계산하여 입력된 글자의 크기가 큰 경우 일정 비율로 글자 크기를 조정하는 방법을 사용한다. 이 방법은 고정된 셀에 입력 받은 글자를 빠르게 축소하는 데는 효과적이거나, 일정 비율로 일괄 축소되면서 글자 수가 많은 경우 지나치게 작은 크기로 축소되는 문제가 발생한다.

어떻게 하면 셀 내부에 최적의 글자를 표현할 수 있을 것인가? 이 문제는 글자를 표현하는 글자크기, 줄 간격, 자간 간격을 조합하여 최적의 값을 찾는 문제로 정리될 수 있다. 기존의 일정비율로 조정하는 선형화된 방법은 앞서 언급한 것과 같은 문제가 발생한다.

유전자 알고리즘은 이러한 조합 최적화 문제에 적합하다고 알려져 있다. 이 실험은 유전자알고리즘을 이용하여 표의 고정 크기 셀에 길이가 서로 다른 글자를 최적화된 상태로 표현될 수 있도록 글자의 속성들을 최적의 상태로 조합할 수 있는

유전자 알고리즘을 개발하여 적용해보고 이러한 제한적인 문제에도 유전자 알고리즘을 이용한 방법이 효과적인지 검증하는데 목적이 있다.

본 연구를 위해 2장에서는 현재 상용화된 제품에서의 글자 최적화 방법 및 문제점을 살펴보고, 3장에서 수리모형을, 4장에서 유전자알고리즘을 설계한다. 5장에서 실험을 통해 글자 최적화 과정을 분석하고 6장에서 결론을 맺는다.

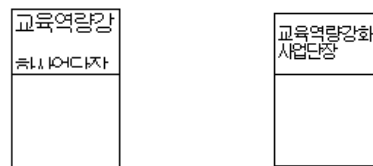
본 논문은 기존 연구[1]의 돌연변이율에 대한 적합도와 해의 개선정도를 높이기 위하여 교차연산과 산술평균 교차에 대한 적합도 분석을 새롭게 연구하였다.

II. 선행 연구 고찰

2.1 상용 워드프로세서 사례 분석

2.1.1 아래한글(한글과 컴퓨터사)

아래한글에서 표의 셀 고정 기능을 이용하는 경우 그림 0과 같은 문제가 발생한다. 이런 경우 사용자가 직접 글자크기, 줄 간격, 자간 등의 글자 속성을 여러 번 반복 적용해서 각자 기준에 따른 최적의 속성 조합으로 편집하곤 한다. 아래한글에서는 이런 경우의 최적화 기능을 지원하지 않는다.



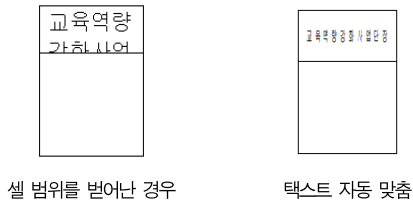
셀 범위를 벗어난 경우

사용자 편집

그림 4. 아래한글 셀 편집 사례
Fig. 4. Hangeul Cell compilation Example

2.1.2 MS Word(Microsoft사)

Microsoft사의 워드프로세서인 MS Word에서 표 기능을 이용하는 경우 셀 옵션에서 "텍스트 자동 맞춤"이라는 기능으로 셀에서 글자가 모두 표시될 수 있도록 지원하는 기능이 있다. 그러나 이 경우 글자 크기가 셀을 기준으로 일괄 축소되어 글자 수가 셀 크기보다 많은 경우, 그림 2에서처럼 셀 내 표시된 글자를 알아보기 어려운 상황이 종종 발생하게 된다.



셀 범위를 벗어난 경우 텍스트 지동 맞춤
 그림 5. MS Word 셀 편집 사례
 Fig. 5. MS-Word Cell compilation Example

2.1.3 엑셀(Microsoft사)

대표적인 스프레드시트 엑셀에서 이와 유사한 기능을 지원 하는지 살펴보았다. 엑셀에서는 셀 서식에서 “셀에 맞춤”이라 는 기능 이용하여 셀 내 표시된 글자를 최적화하는 방법을 사 용하는데 이 기능도 그림 3과 같이 셀 크기에 비례해서 축소 하는 방법을 사용하고 있어서 지나치게 크기가 작아진 글자는 알아보기 어려운 경우가 발생한다.

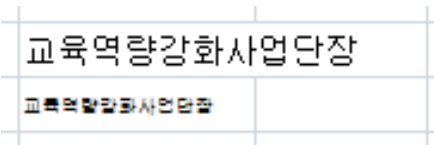


그림 3. 엑셀 셀 편집 사례
 Fig. 3. Excel Cell compilation Example

2.1.4 RDServer(M2Soft사)

웹 리포팅 툴 전문회사 M2Soft의 RDServer에서 지원되 는 표 기능에서 셀 내 글자 최적화 방법은 셀 크기를 정한 후 셀 범위 내부에 모든 글자가 표시되게 하기위하여 “폰트크기 조정”이라는 방법을 이용한다. 이 방법도 다른 제품들과 같이 셀 크기에 비례하여 글자 속성을 축소하는 방법을 사용하기 때문에 글자 수가 많아지면 그림 4와 같이 글자를 알아볼 수 없을 정도로 축소되는 현상이 종종 발생한다.

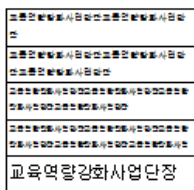


그림 4. RDServer 셀 편집 사례
 Fig. 4. RDServer Cell compilation Example

이처럼 셀 내 글자 크기를 최적화하려는 시도는 표 기능을

지원하는 워드프로세서, 스프레드시트, 웹리포팅 툴 등 거의 모든 제품에서 확인되고 있으나, 여러 가지 글자 속성을 조합 하여 최적화하려는 시도는 거의 없는 것으로 확인 되었다. 이 러한 문제점을 조합 최적화 관점에서 접근하고 해결하기위해서 최적화된 조합을 찾는데 유용한 유전자 알고리즘을 이용해서 글자 속성 조합을 최적화할 수 있는 방안을 찾아보고자 한다.

2.2 유전자알고리즘

유전자알고리즘은 유전학과 자연 진화의 개념을 도입한 적 응 탐색법으로 1975년 Holland에 의해 개발되었는데 유연 한 성격 때문에 복잡한 최적화 문제를 해결하고 강건성을 높 이기 위한 방법으로 많이 사용되고 있다.[2]

이는 전통적으로 선형계획법이 목적함수의 최적해를 구하 는 방법으로 많이 사용되고 있으나 목적함수가 비선형, 비연 속, 미분 불가 등의 제약이 있는 경우에는 비선형계획법으로 도 최적해를 구하는 것이 어렵기 때문이다.

유전자 알고리즘은 미분 방법 대신 확률적 탐색 방법을 사 용하여 해를 탐색하며, 제약식을 통해 자손 세대의 유전자 진 화를 통제하도록 설계할 수 있고, 만약 비가능해가 있는 경우 라도 페널티 함수 등을 통해 자손 단계에서 도태시킬 수 있다. 이러한 확률의 원리를 활용하여 가장 우수한 최적의 해를 도출 하게 되는데 알고리즘의 그림 5와 같은 구조로 표현할 수 있다.

본 연구에서는 교배를 위한 방법으로 산술교차를 활용하였 고, 돌연변이 실험을 통해 최적의 돌연변이 비율을 도출하는 실험을 수행하였고 이를 통해 최적의 글자 속성으로 표현하였다.

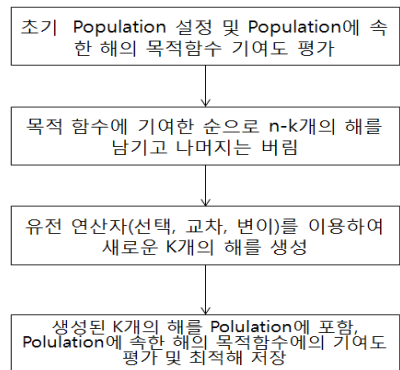


그림 5. 유전자 알고리즘의 구조(2)(3)
 Fig. 5. Structure of the genetic algorithm(2)(3)

제한된 유전자 알고리즘은 표 기능을 지원하는 고정 셀에 서 최적의 글자 속성을 찾아내는데 효과적으로 사용될 수 방 법을 제시하였다.

2.3 Font 연구 현황

글자 Font와 관련한 연구는 국내외 적으로 그 수가 작으며, 주제 영역으로 보면 표 1과 같이 폰트개발 및 디자인과 이와 관련한 정보시스템 관련 논문이 다수를 차지하고 있으며, 그 외 폰트 인식, 저장장치와 관련하여 폰트 압축 및 복원, 호환 처리, 가독성 향상 등의 연구가 진행되어 지고 있다.

표 1. Font 연구 현황
Table 1. Font study status

연구 주제	관련연구
폰트(디지털 폰트) 개발 및 디자인, 관련 정보시스템	{6}, {7}, {8}
폰트 별 효율적 저장방식, 압축 및 복원	{8}, {10}
글꼴(폰트) 별 호환 처리	{11}
폰트에 따른 글자 인식 및 가독성	{9}, {12}

본 연구에서 다루고자 하는 실제 워드프로세스 작업 시 폰트 크기를 최적화하여 표현하는 것과 관련하여서는 국내외 적으로 연구된 바가 없다.

III. 문제수리모형

3.1 알고리즘 수리 모형

최적화에 사용할 글자 속성은 글자크기(pt), 줄 간격(pt), 자간(%) 총 세 가지 속성을 사용한다. 모든 속성 값들은 각종 연산에 동일한 크기를 적용하기 위하여 mm로 변환하여 적용하며, 글자크기와 동일한 크기로 표현하기 위하여 줄 간격은 자간속성의 “고정값” 기능을 이용한다.

3.1.1 제약조건

본 연구가 알고리즘의 유효성을 실험하기 위한 목적임을 감안하여 글자 크기, 줄 간격, 자간 범위, 글자체에 대하여 다음과 같이 제약사항을 전제로 한다.

- 글자크기 제약 : 실험에 사용하는 글자크기는 6pt ~ 20pt로 제한한다.
- 줄 간격 제약 : 줄 간격이 글자크기를 초과하면 글자 겹침 현상이 발생하게 됨으로 줄 간격은 글자 크기를 초과할 수 없도록 제한한다.
- 자간범위제약 : 일반적인 워드프로세서의 자간범위는 기본 값이 0으로 설정되어있다. 이보다 작아지는 경우 가독성이 떨어지므로, 자간 범위는 0보다 크거나 작고 -20보다 작도록 제한한다.(아래한글 허용범위의 20%만 사

용)

- 글자체 제약 : 글자체는 굴림체로 제한하여 실험한다.

3.1.2 목적함수

글자의 각 속성들을 최적의 상태로 조합할 수 있는 속성조합을 만들기 위하여 고정 셀의 가로, 세로 길이를 구한 후, 입력 받은 글자 크기와 한 줄당 표시 가능한 최대 글자 수, 표시 가능한 최대 글자 높이를 차례대로 계산하여 연산이 종료될 때, 남은 셀 높이가 크고 줄 간격이 0에 가까우며 글자 크기가 큰 조합이 좋은 해가 되도록 한다.

3.1.3 수리모형

수리 모형은 유전자 알고리즘을 위한 입력 상수, 표 내 글자 최적화를 위한 조건인 글자 크기와 간격, 자간의 입력 변수, 이들 간의 최적 조건을 연산하기 위한 수식으로 구성된다.

[입력상수]

- POPULATIONS_SIZE(PS) : 모집단크기
- GENE_CNT : 유전자의 개수
- END_CNT : 수행횟수
- ELITISM_SIZE(ES) : 엘리트보관 개수
- mutationRate(MR) : 돌연변이비율
- PT_mm : 단위변환(pt → mm) 0.35mm
- CW : 셀의 넓이
- CH : 셀의 높이
- WC : 입력 받은 글자 개수

[입력변수]

- fts : 글자 최소크기(pt)
- fte : 글자 최대크기(pt)
- rss : 줄 최소간격(pt)(=fts)
- rse : 줄 최대간격(pt)(=fte)
- lse : 자간 최소간격
- lss : 자간 최대간격
- g_0 : 글자 크기 속성 값(mm)
- g_1 : 줄 간격 속성 값(mm)
- g_2 : 자간 속성 값(mm)
- r_0 : 남은 글자 수 적합도
- r_1 : 남은 셀 높이 적합도
- r_2 : 남은 줄 간격 적합도
- r_3 : 남은 줄 간격 적합도($r_3 \leq 0$)
- CRH_MAX : Cell에 표시 가능한 최대 글자 높이
- CRC_MAX : Cell에 한 줄당 표시 가능한 최대

글자 수

RWS_{mm} : 환산 줄 간격(mm)

CXX : 현재 남아있는 글자 수

RRC : 현재 남아있는 줄 수

RWT_{mm} : 누적 총 높이

CWS : 한 글자 당 자간 너비 환산(mm)

$n : CXX - CRC_MAX, n > 0$

$k : RRC$

한 글자 당 자간 너비 환산 표현식은 식(1)과 같으며 자간의 너비를 mm로 환산한다.

$$CWS = (g_0 \times PT_mm) \times \left(\frac{g_2}{100}\right) + PT_mm \dots\dots\dots (1)$$

한 줄당 표시 가능한 최대 글자 수 표현식은 식(2)와 같다. 현재의 글자 속성 조합으로 셀에 한 줄당 표시 가능한 최대 글자 수를 얻기 위하여 고정 셀의 가로길이를 환산된 글자너비와 환산된 자간너비를 합한 값으로 나눈다.

$$CRC_MAX = \text{floor}\left(\frac{CW}{(g_0 \times PT_mm) + CWS}\right) \dots\dots\dots (2)$$

현재 남아있는 글자 수 표현식은 식(3)과 같다. 글자 속성 조합으로 표시할 수 있는 입력받은 글자 수에서 한줄 표시 가능한 글자 수를 빼면 현재 남아있는 글자 수를 얻을 수 있다.

$$CXX = \sum_k^n (CXX - CRC_MAX) \dots\dots\dots (3)$$

누적 총 높이 표현식은 식(4)와 같다. 누적 총 높이 연산은 셀 내부에 글자를 최적화해서 표현위한 중요한 연산이다.

$$RWT_mm = \sum_k^n (FTH_mm \times k + RWS_mm \times (k-1)) \dots\dots\dots (4)$$

셀에 글자를 최적화하기 위해서는 글자 속성 조합이 셀 내부에서 어떻게 표현되고 있는지 수리적으로 판단할 수 있어야 적합도를 판단할 수 있다. 이 실험에서는 생성된 글자 속성조합으로 셀 범위를 얼마나 초과하는지 누적된 총 높이를 연산

하여 셀 높이와 비교해서 초과되면 그 셀 범위를 벗어난 것으로 판단하고 페널티를 부여하는 방식으로 처리한다.

적합도는 남은 글자수 적합도(식(5)), 남은 셀 높이 적합도(식(6)), 남은 줄 간격 적합도(식(7)), 남은 줄 간격이 음수인 경우의 줄 간격 적합도(식(8))로 나누어 계산된다.

$$r_0 = \sum_k^n (WC - (CRC_MAX \times k)) \dots\dots\dots (5)$$

$$r_1 = \sum_k^n (CH - RWT_mm) \dots\dots\dots (6)$$

$$r_2 = \sum_k^n (g_1 - g_0) \dots\dots\dots (7)$$

$$r_3 = \sum_k^n (g_0 - g_1) \dots\dots\dots (8)$$

즉, r_0 은 연산이 종료된 후 남은 글자 수를 의미한다. r_1 은 연산이 종료된 후 남아있는 셀의 높이를 의미하며 적합도 첫 번째 요소로 산정에 사용된다. r_3 는 남은 줄 간격으로 연산된 값은 줄 간격이 크면 적합도를 낮게 처리하기 위하여 글자크기에서 줄 간격을 뺀 값을 사용한다. 이렇게 처리하면 글자 크기보다 줄 간격이 큰 경우 음수가 나오게 되어 적합도는 낮아지게 된다. 적합도의 두 번째 요소로 사용된다.

최종 목적함수는 적합도를 모두 합산한 값으로서, 계산식은 식(9)와 식(10)과 같다.

$$Z = MAX(r_1 + r_3 + g_0), r_1 + r_3 \geq 0 \dots\dots\dots (9)$$

또는

$$Z = MAX(r_1 + r_3), r_1 + r_3 < 0 \dots\dots\dots (10)$$

적합도는 연산 후 남아있는 셀의 높이와 줄 간격 가중치 그리고 글자크기를 더하여 얻는다. 따라서 최적해는 남은 셀 높이가 크고 줄 간격이 0에 가까우며 글자 크기가 큰 조합으로 이중 제일 큰 적합도를 갖고 있는 유전자가 된다.

IV. 유전자 알고리즘 분석 및 설계

4.1 알고리즘을 적용하기 위한 순서도

그림 6의 알고리즘 순서도를 간단히 요약하면, 모집단을

생성한 후 적합도 평가와 정렬을 수행한다. 이후 종료조건을 만족하는지 비교한 후 만족하는 경우 종료 시키고 만족하지 않는 경우 엘리트 유전자를 보관 처리한다. 우성유전자 보관이 끝나면 교차연산을 수행하고 돌연변이처리를 수행한다. 돌연변이가 끝나면 임시저장소에 기록되어 있는 자료를 새로운 개체집단으로 구성한 후 일련의 과정을 종료 조건이 완성될 때까지 반복한다.

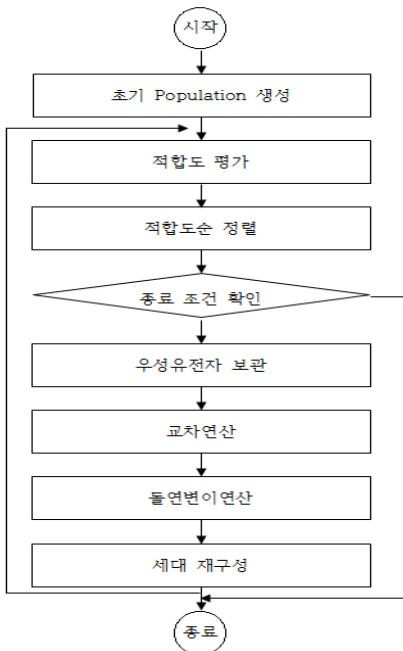


그림 6. 알고리즘 흐름도
Fig. 6. Algorithm Flow Chart

4.2 해 표현 방식

4.2.1 실수유전자

무엇보다도 인자들의 속성이 실수일 경우 인자들과 유전자들이 일대일 대응되는 간명함이 실수 표현의 큰 장점이다. 또한 수의 '크기' 개념을 연산자에 담을 수 있는 장점도 있다. 동일 위치에 있는 두 부모의 유전자들의 값을 평균 내어 자식의 동일 위치 유전자 값으로 삼는 산술적 교차 (arithmetical crossover)가 한 예로, 실수 표현에 대한 Michalewicz의 실험[3]에 의하면 비교적 타 연산방법에 비해 우수한 것으로 나타나있다.

산술적 교차는 수의 '크기' 라는 개념을 교차 행위에 사용하는 점에 큰 매력이 있다. 그렇지만 수들의 산술 평균을 지향하므로 매우 빠른 수렴을 보인다. 변이 등을 적절히 조절하여 설익은 수렴이 되지 않도록 주의해야 한다.[5]

4.2.2 모집단 구성

모집단은 글자크기와 줄 간격, 자간 속성을 실수형으로 정의하고 반복 실험을 거쳐 최적의 유전자 개체 수를 정하는 방식을 취하고자 한다. 그림 7은 이번 실험에 사용된 유전자 설계이다.

Population (A generation)	A chromosomes		
	글자크기 g1	줄간격 g2	자간 g3
Chromosome 1	10.2	10.4	-10
Chromosome 2	16.5	20.4	-1
Chromosome 3	12.2	6.4	-2
⋮	⋮	⋮	⋮
Chromosome n	7.7	9.5	0

그림 7. 유전자 설계
Fig. 7. Gene Architecture

4.2.3 적합도 평가

초기화된 세대에 속하는 개체들은 적합도 함수에 의해 주어진 문제의 최적해(Optimum Solution)에 얼마나 근접해 있는가에 대한 평가를 받는데 이를 적합도라 한다[13]. 본 연구에서 적합도는 크가 정해진 표의 셀 안에 표시할 수 있는 글자 수, 줄 간격, 글자 간격을 계산하여 각 변수들의 합이 셀 범위를 얼마나 초과하는지를 적합도 함수로 사용한다. 계산 결과가 0보다 크면 셀 범위를 초과하지 않는 것을 의미하고 0보다 작으면 셀 범위를 벗어난 것을 의미한다. 셀 범위를 초과하지 않는 속성 조합들 중 최대값을 최적의 유전자로 선택한다.

4.2.4 교차연산

교차연산은 실수형 유전자의 장점을 최대한 활용하고자 두 개의 염색체의 각 유전자의 속성 값을 산술평균하여 교차연산(교배)한다. 교차연산은 엘리트 유전자를 제외(10%)한 범위에서 이루어지도록 한다. 그림 8은 이번 실험에서 적용되는 산술 평균 교차 연산 방법을 설명하고 있다.

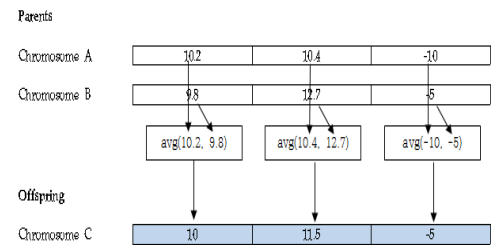


그림 8. 산술 평균 교차 연산
Fig. 8. An arithmetic mean crossover operation

4.2.5 돌연변이연산

돌연변이는 전체 개체 중에서 일정 비율을 정하여 임의로 선택된 염색체의 유전인자를 임의로 선택하여 처리한다.

돌연변이는 지역최적화를 회피하고 유전자의 다양성을 확보하여 양질의 유전인자를 얻기 위해 임의로 선택된 유전인자를 조작하는 과정이다. 이러한 돌연변이 연산은 주어진 환경에서 돌연변이 결과가 최선의 방향으로 돌출되도록 각각의 매개변수를 조정하여 보다 유익한 결과가 나오는 변수를 찾는 것이 중요하다.

돌연변이 확률을 높이면 보다 다양한 해를 생성할 수 있지만 해 집단의 수렴성은 떨어진다. 반대로 확률을 낮게 하면 수렴성은 높아지지만 다양한해를 생성하기가 힘들어진다. 이번 실험에서는 변이율을 조정하여 각각 반복해서 측정한 후, 이 중에서 해의 다양성을 보장할 수 있고 수렴도가 높은 변이율을 찾아서 기준 돌연변이율로 선택한다.

V. 실험 및 결과분석

5.1 실험 종류 및 환경

유전자 알고리즘 개발 언어는 java 1.5SDK, 데이터를 처리를 위해 Oracle9i와 엑셀 2010을 기반으로 구현하였고, 파라미터 설정을 조정하며 반복실험을 진행하였다.

신규개체 생성비용, 교차연산 방법, 돌연변이 비율은 표 2와 같다.

표 2. 실험 환경 변수
Table 2. Experimental Environment Variables

구분	값
모집단(PS)	100개체
엘리트보존비율(ER)	10%
돌연변이비율(MR)	10%(고정)
종료조건	3,000회
셀 높이	10mm
셀 너비	35mm
글자 수	20자

5.2 개체의 크기와 수행횟수 분석

개체의 크기와 수행횟수와의 상관관계를 분석하여 유전자 연산에서 최적의 개체수를 정하기 위하여 개체 수는 50개체부터 200개체까지 50씩 증가시키고 수행횟수는 10회부터 200회까지 10씩 증가 시켜서 반복 수행하였다. 수행 결과, 그림 9와 같이 개체 집단이 60개체인 경우와 100개체일 때

적합도가 최대값으로 수렴되었다. 이후 실험은 모두 100개체를 기준으로 수행하였다.

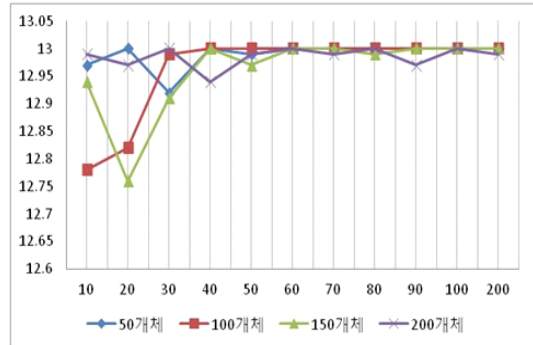


그림 9. 개체 크기와 수행횟수 수렴도의 상관관계
Fig. 9. Correlations of objects size and convergence ratio of number of executing

5.3 교차연산 방법에 따른 적합도 분석

이 실험에서는 실수형 유전자의 장점을 최대한 활용하고자 두 개의 염색체의 각 유전자의 속성 값을 산술평균하여 교차연산하는 방법을 사용하고 있다. 이 방법이 일반적인 부모 유전자 교차방법과 비교했을 때 효과적이지 검증하기 위하여 부모 유전자 교차 방법을 구현하여 동일한 매개변수를 적용하여 실험했다.

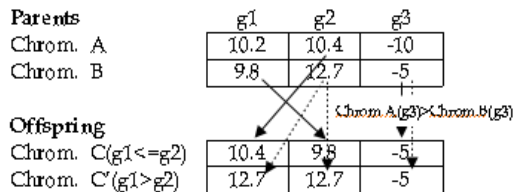


그림 10. 부모 유전자를 활용한 교차연산
Fig. 10. Cross Operation Using Parent gene

그림 10은 실험에서 적용한 부모 유전자 교차 연산 구성도이다. 임의의 부모유전자 A와 B를 선택하여 A의 줄 간격(g2)은 자식C의 글자크기로 B의 글자크기(g1)은 자식C의 줄 간격으로 교차하고 자간간격은 A와 B를 비교하여 자간간격이 큰 유전자로 교차 연산하는 구조이다. 단, 글자크기가 줄 간격보다 큰 경우 글자 겹침이 발생하므로 이를 적절히 조정하는 과정이 추가로 필요하다.

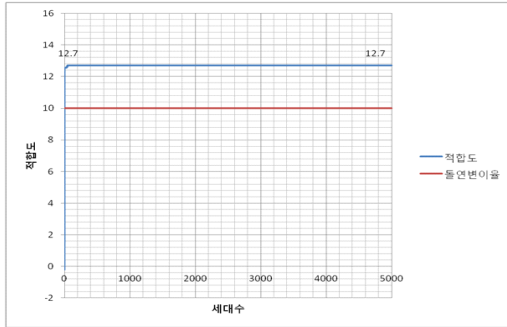


그림 11. 부모 유전자 교차 연산방법의 적합도 분포(개체:100, 변이율:10%, 세대수 5,000)

Fig. 11. The fitness distribution of Parents genetic crossover operation method(object: 100, changed rate: 10%, the number of generation : 5000)

그림 11은 개체 수 100개, 돌연변이율 10%를 적용하고 기타 매개변수도 동일한 값을 적용하여 5,000회까지 반복하여 수행한 결과중 하나이다. 동일한 실험을 총 10회 수행, 수렴 정도는 모두 그림 11과 비슷한 분포를 나타냈다. 실험 결과 부모 유전자 교차연산 방법은 그림 12의 산술평균 교차 연산방법 보다 적합도가 낮아 수렴 정도가 좋지 않게 나타나고 있다. 이와 같은 결과는 줄 간격보다 큰 경우 글자 겹침을 방지하기 위해 글자크기를 줄 간격의 크기로 제한하는 제약조건 연산으로 다양한 유전자의 출현이 어렵게 되면서 해의 개선 정도가 약해진 결과로 분석된다. 반면에 그림 12는 산술평균으로 적당한 유전자의 분포가 가능해지면서 해의 개선정도가 부모유전자의 교차 연산보다 좋은 것으로 나타나고 있다.

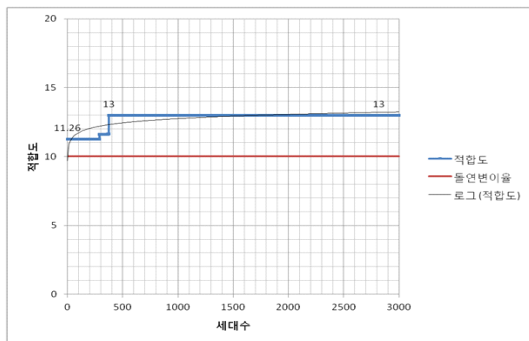


그림 12. 산술 평균 교차 연산방법의 적합도 분포(개체:100, 변이율:10%, 세대수 3,000)

Fig. 12. The fitness distribution of arithmetic mean crossover operation method(object: 100, changed rate: 10%, the number of generation : 3000)

5.4 돌연변이율과 적합도 분석

앞서 언급한 것과 같이 돌연변이 비율은 해의 수렴성과 밀접한 관계가 있다. 이번 실험은 돌연변이 비율에 따른 적합도 수렴 정도를 비교하여 적합한 돌연변이 비율을 확인하는 실험이다. 실험은 돌연변이 발생확률의 변이를 각각 10%, 30%, 50%로 조정하면서 개체의 수는 100개체, 세대수는 3,000으로 실험했다. 실험은 각10회씩 실시하였다. 실험결과 그림 13에서처럼 100개체에서 10% 변이가 발생하였을 때 수렴정도가 가장 좋은 것으로 나타났다. 변이율 30%(그림 14)와 50%(그림 15)의 경우 초기 값의 영향이 크고 해의 개선 정도가 상대적으로 낮은 것으로 확인되었다.

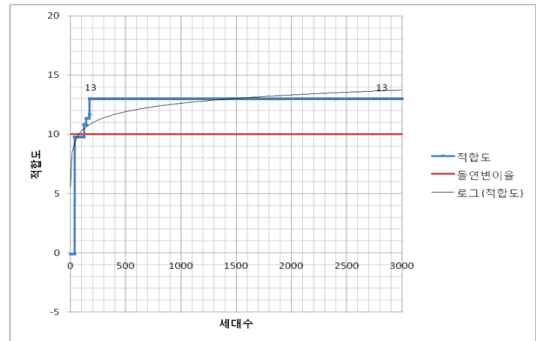


그림 13. 돌연변이율과 적합도 분포(개체:100, 변이율:10%, 세대수 3,000)

Fig. 13. The fitness distribution versus mutation ratio(object: 100, changed rate: 10%, the number of generation : 3,000)

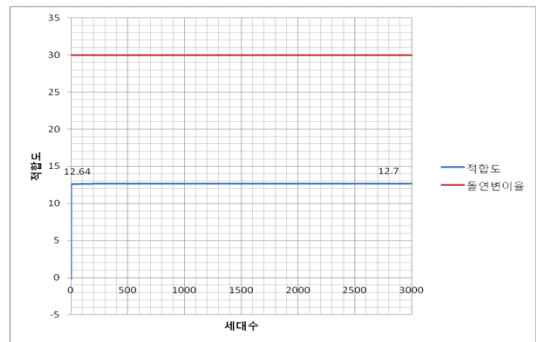


그림 14. 돌연변이율과 적합도 분포(개체:100, 변이율:30%, 세대수 3,000)

Fig. 14. The fitness distribution versus mutation ratio(object: 100, changed rate: 30%, the number of generation : 3,000)

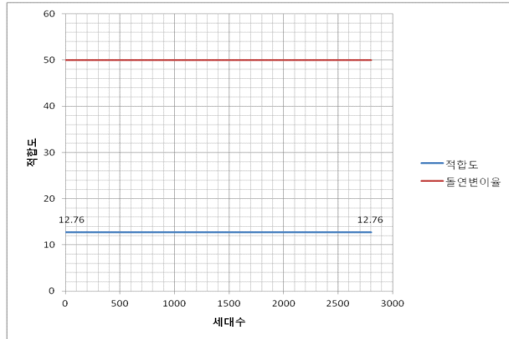


그림 15. 돌연변이율과 적합도 분포(개체:100, 변이율:50%, 세대수 3,000)

Fig. 15. the fitness distribution versus mutation ratio(object: 100, changed rate: 50%, the number of generation : 3,000)

5.5 실험 결과

이번 실험에서 목적함수 설계 조건은 줄 간격이 0에 가까우며 글자 크기가 큰 조합으로 셀 범위를 벗어나지 않는 글자 속성조합 중 제일 큰 적합도 값을 갖는 유전자를 찾는 것이다. 적용 실험 환경은 한글 2010, 셀 안 여백 0mm, 셀 높이 10mm, 셀 너비 35mm, 글자 수 20, 글자체는 굴림체이다. 그림 16은 앞서 실험한 유전자알고리즘에 적용할 매개변수들의 적정성 실험결과를 토대로 개체 수:100, 변이율:10%, 세대수(종료조건):3,000회를 실제 한글 셀 속성 값으로 적용한 모습이다. 글자 속성이 유전자 연산으로 조합되어 근사 최적해로 표현되는 것을 확인할 수 있었다.

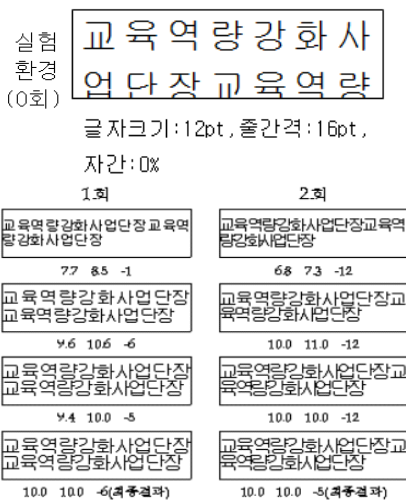


그림 16. 글자 속성 선택 적용 결과
Fig. 16. The result of applying the character properties cell

VI. 결론

이번 연구는 널리 사용되어지고 있는 워드프로세서의 표 기능 중 셀 고정 기능에서 글자 크기를 최적화하기 위한 조합 최적화 문제에 효과적인 유전자알고리즘을 이용하는 방법과 실제로 구현된 알고리즘을 최적의 상태로 적용하기 위한 실험 및 분석, 그리고 실제로 적용한 결과 대한 내용이다.

고정 셀을 벗어나지 않는 최적화된 글자크기를 산출하기 위해 셀을 고정된 공간으로 설정하고 글자의 속성들을 동일한 수치로 환산하여 글자를 직렬화한 후 일반적인 워드프로세서의 자동계행 기능을 전제로 현재 입력받은 글자의 최대 줄 수를 산정하여 셀의 범위를 얼마나 초과하는지를 계산하여 이를 적합도로 적용하였다. 문제의 핵심속성인 셀 범위를 크기로 표현하여 유전자 연산에 적용하기위하여 “크기” 표현에 적합한 실수유전자를 사용하였다. 실험에 사용된 실수형 유전자는 일반적인 이진유전자와 비교했을 때, 상대적인 크기와 양의 개념을 쉽게 표현할 수 있어서 이번 연구와 유사한 성격의 조합 최적화 표현에 유용하게 응용될 수 있다는 것을 보여주고 있다.

그러나 본 연구에서 구현된 프로그램으로 도출된 최적해는 각 속성의 적합도를 더한 값으로 사용자들의 글자 속성 선호도에 대한 반응이 없으며, 글자체에 따른 고유 속성들과 특수문자, 외래어 등의 각종 글자 폰트 특성 등에 관한 세부적 항목과 셀에 대한 자세한 속성들을 반영하지 못하고 제한된 연구에 그치고 있다. 향후에는 이번 연구에서 나타는 문제점들을 보완하여 셀 내 글자의 최적화를 위한 수리 모형에 관한 연구와 글자와 셀의 다양한 속성을 함께 반영하여 최적화할 수 있는 추가적인 연구가 필요하다.

참고문헌

- [1] Sang-Won Kim, Woo-Je Kim, "Font optimization in Fixed-function cell Using a Genetic Algorithm," Proceedings of the Korean Operations Research and Management Science Society Seminar, pp. 1337-1343, Nov. 2012.
- [2] Seung-Hee Kim, Woo-Je Kim, "Development of a Predictive Model of Satisfaction Survey Respondents Classified Information Service using GA," Journal of KIIT. Vol 11, No. 6, pp.

- 111-126, June. 2013.
- [3] V. Kecman, "Learning and Soft Computing," MIT Press, Cambridge, Mass, pp. 499, 2001.
- [4] Michalewicz, Z. "Genetic Algorithms + Data Structures = Evolution Programs," Springer-Verlag., 1992.
- [5] Seung-Gon, Gong, "Genetic Algorithms(Genetic Algorithms + Data Structures = Evolution Programs)," 3rd., green, Oct. 1996.
- [6] Young-Deuk Moon, "Digital Font Design for Effective Multimedia Contents Description," Journal of KIIT, 8(1), pp. 326-329, May. 2010
- [7] Byung-Gook Lee, Yun-Beon Park, Woan-Kyu Lee, "An Effective Representation of Combination Rule for the Hangeul Typeface Design," Journal of the Korea Information Processing Society, 3(6), pp. 1580-1587, Nov. 1996.
- [8] Eun-hwe Kim, Geub-ho Jeong, Jae-young Choi, "Minimizing Duplicates of Hangeul Fonts using Composite Glyph of TrueType," Journal of KIISE(B), 26(10), pp. 1230-1236, Dec. 1999.
- [9] Young-Woong Song, Chang-Wook Lim, In-Seok Lee, Myung -Chul Jung, Seung-Min Mo, Yong-Ku Kong, "Effects of the Syllable Number, Font Type, Color Contrast, Display Type, Letter Size and Age Group on the Legibility of the Korean Characters," Journal of the Korean Society of Safety, 24(5), pp. 92-100, Oct. 2009.
- [10] Gyung-Yun Cho, "A Study on Compression and Decompression of Bit Map Data by NibbleRLE Code," Journal of the Korea Information Processing Society, 2(6), pp. 857-865, Nov. 1995.
- [11] Nenad Marovac, "handling fonts in electronic publishing systems," Computers & Graphics, 11(3), pp. 289-295, Nov. 1987.
- [12] Chandler, Scott Bondurant, "Comparing the legibility and comprehension of type size, font selection, and rendering technology of on-screen type," University Virginia Polytechnic Institute and State University

ProQuest, UMI Dissertations Publishing, pp. 1-108, Aug. 2001.

- [13] Sang-Sung Park, Dong-Kyu A, "Image Feature Extraction using Generic Algorithm," Journal of The Korea Society of Computer and Information Vol. 11 No. 3, pp. 133-139, Aug. 2006.

저 자 소 개



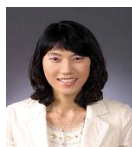
김 상 원

1994~현재: 한림성심대학교
정보전산원

2012: 서울과학기술대학교
IT정책전문대학원 박사 과정

관심분야: IT서비스 최적화

Email : swkim@hsc.ac.kr



김 승 희

2003: 동국대학교

컴퓨터멀티미디어공학 공학사

2005: 연세대학교

산업정보경영 공학석사

2011: 서울과학기술대학교

IT정책전문대학원 박사 과정

현 재: 한국토지주택공사

경영정보처 근무

관심분야: 품질공학, IT서비스 최적화

Email : kim.sh@LH.or.kr



김 우 제

1986: 서울대학교

산업공학과 공학사

1988: 서울대학교

산업공학과 공학석사

1994: 서울대학교

산업공학과 공학박사

현 재: 서울과학기술대학교

글로벌융합산업공학과 교수

관심분야: 소프트웨어공학, IT서비스

Email : wjkim@seoultech.ac.kr