

유전 알고리즘 처리속도 향상을 위한 프로세서 구조

Processor-Architecture for the Faster Processing of Genetic Algorithm

윤한얼, 정재원, 심귀보
중앙대학교 전자전기공학부

Han-Ul Yoon, Jae-Won Jung, and Kwee-Bo Sim

School of Electrical and Electronics Engineering, Chung-Ang University

E-mail: huyoon@wm.cau.ac.kr

ABSTRACT

유전 알고리즘은 NP-Hard 문제의 해결이나, 함수 최적화, 복잡한 제어기의 파라미터 값 추적 등, 광범위한 분야에 걸쳐 이용되고 있다. 일반적인 유전 알고리즘은 적합도 함수를 통해 해들의 품질을 결정하고, 해들의 품질에 따라 선택 연산을 거쳐, 교차나 돌연변이를 통해 우수한 품질의 해를 찾는 과정을 가진다. 현재 이 과정은 대부분 소프트웨어적으로 구현되어 범용 프로세서를 통해 수행 된다. 그러나 높은 소프트웨어 의존성은 해집단의 크기가 커질수록 교차/변이 연산과 해들의 품질비교에 수행되는 시간을 크게 증가시키는 약점이 있다. 따라서 본 논문에서는 순위 기반 선택과 일점 교차(one-point crossover)를 사용한다는 제약하에, 해들의 순위를 정렬 네트워크를 통해 결정하고 해들을 Residue Number System(RNS)로 표현하여 하드웨어적으로 교차연산을 처리하는 프로세서 구조를 제안한다. 이러한 접근을 통해 해들의 품질비교에 걸리는 시간을 크게 줄이고 교차/변이 연산의 효율을 높일 수 있다.

Key words : 순위 기반 선택, 일점 교차, 정렬 네트워크, RNS, GA 프로세서 구조

I. 서 론

유전 알고리즘의 가장 큰 매력을 말하자면 실제 생물학적 모델과의 유사성일 것이다. 이러한 매력은 소프트웨어 환경상에서 단순한 코드들이 실제 세계에서 일어났던 진화과정을 밟는 것을 가능하게 하였다. 유전 알고리즘이 특히 부각되는 이유는, 과거 정확한 초기 모델링과 수많은 실험에서의 시행착오를 거쳐 사람이 해결하였던 많은 문제들(Human doing)을 기계들이 스스로 알아서(Machine doing) 할 수 있도록 만들었다는 점이다. 이에 유전 알고리즘은 현재 매우 넓은 분야에서 응용되고 있다. 그 중 가장 대표적인 예를 들자면, 함수 최적화, 신경망 최적화, 퍼지 시스템 최적화, 차량 라우팅, 그래프 분할, VLSI 회로 배치 등이다.

일반적으로 유전 알고리즘 다음과 같은 전형

적인 구조를 갖는다[1].

- ① 초기 집단 생성
- ② 적합도 판정
- ③ 선택/재생
- ④ 교차
- ⑤ 변이
- ⑥ 적합도 판정
- ⑦ 종료조건을 만족하면 정지. 만족하지 않으면 다시 ③으로 돌아가 반복

③의 선택 연산에는 잘 알려진 예로, 품질 비례 룰렛 휠 선택, 토너먼트 선택, 순위 기반 선택 등이 있다. 본 논문에서 시스템 구성의 제약(Constraint)으로 정한 순위 기반 선택은 해집단 내의 해들을 품질 순으로 '순위'를 매긴 다음 가장 좋은 해부터 일차 함수적으로 적합

도를 배정하는 방법이다[2]. 여기서 품질에 따른 해들의 순위를 정하기 위해서는 해들을 오름차순 또는 내림차순으로 정렬한 뒤, 차례로 순위를 배정하면 된다. 이 경우 해집단의 크기가 매우 크다면 해들의 품질을 비교하는데 상당한 수행 시간이 요구된다.

④의 교차 연산은 다양한 해를 탐색하기 위한 유전 알고리즘의 대표적 연산자이다. 그 중 일점 교차는 가장 많이 사용되는 교차 연산으로 선택된 두 개의 해를 자름선을 기준으로 각각의 왼쪽과 오른쪽을 취하여 합침으로써 새로운 해를 만들어낸다(그림1). Residue Number

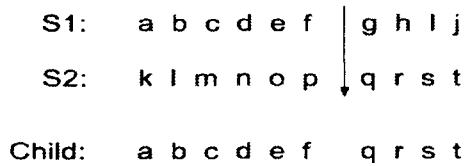


그림 1. 일점 교차

system이란 각 자릿수(digit)를, 해당 자릿수에 배정된 제수(moduli)들로 나누었을 때의 나머지로 표현하는 수 체계를 말한다[3]. 따라서 RNS에서는 교차 연산에 의한 해들의 다양성이 매우 높고 이것은 보다 넓은 해의 탐색공간을 가질 수 있다는 것을 의미한다.

본 논문에서는 순위 기반 선택에 있어서의 해들의 품질 비교에 정렬 네트워크를 이용하고, Residue Number System(RNS)으로 표현된 해들의 일점 교차를 하드웨어적으로 수행하는 프로세서 구조를 제안한다. 정렬 네트워크를 통해 해들의 품질 비교를 빠른 시간에 처리하고자 하는 것과, RNS로 표현된 해의 하드웨어적 교차를 통해 보다 빠르게 다양한 자식해를 생성하는 것이 본 논문의 목표가 된다.

II. 정렬 네트워크(Sorting Network)를 통한 순위 기반 선택

정렬 네트워크(Sorting Network)란 n개의 입력값을 정렬해 낼 수 있는 k개의 비교기(comparator) 블록으로 구성된 네트워크를 말한다[4]. 네트워크를 구성하는 하나의 블록은 2개의 Input data bus, 1개의 comparator cell(comparator + multiplexer logic), 그리고 2개의 Output data bus로 구성된다. 여기서 I/O bus의 폭은 자연히 I/O data들의 사이즈가 된다. 블록을 통한 데이터 흐름을 설명하기 위해 위쪽 bus의 input data를 A, 아래쪽 bus의 input data를 B라 부르기로 한다. 2개의 input

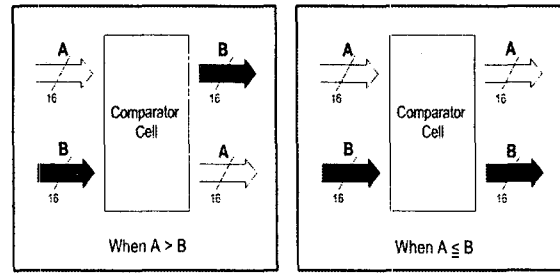


그림 2. Comparator cell을 통한 I/O Data flow

data는 각 bus를 통해 comparator cell로 들어간다. 만약 $A > B$ 라면 output bus의 data는 위쪽이 B, 아래쪽이 A가 되어 두 값은 서로 위치를 바꾸게 된다. $A \leq B$ 의 경우에는 output bus data는 그대로 위쪽이 A, 아래쪽이 B가 된다. 즉, 두 값 중에서 큰 값은 아래쪽 bus로 나가게 되고, 작은 값은 위쪽 bus로 나가게 되는 것이다(그림2). 본 논문에서는 이러한 comparator cell들 60개로 이루어진 Green 모델 정렬 네트워크를 이용한다(그림3). 그림 3에서 평행 스텝이란 어느 한 bus가 가지는 최대의 comparator cell의 개수를 의미한다. 위에서부터 9번째 bus를 보면 왼쪽에서 오른쪽으로 데이터들이 진행하는 동안 총 10번의 비교 연산이 수행되며, 이것은 자연스럽게 모든 데이터가 정렬 되기까지 걸리는 최대 시간이 된다. 결론적으로, 16개 해들의 순위를 결정하는데 10 machine clock으로 처리할 수 있는 것이다.

Input data의 개수가 16로 제한 된 것은, 최종적으로 이 프로세서를 통해 유전 알고리즘을 수행 할 때에 개체군의 개수를 16개로 고정해야만 하는 제약 조건이 된다. 그러나 적절한 문제를 선택한다면 본 논문의 목적인 하드웨어적 정렬을 통해 품질에 따른 해들의 순위 결정을 빠르게 처리하고자 하는 논문의 목적에는 크게 벗어나지 않는다. 또한 input data의 개수를 확장하는 방법이 존재한다. 그림 4와 같이 cross sorting이라는 방법을 이용하면 4개의 N-input local 정렬 네트워크를 이용하여 1개의 $2N$ -input data 정렬 네트워크를 구성할 수 있다.

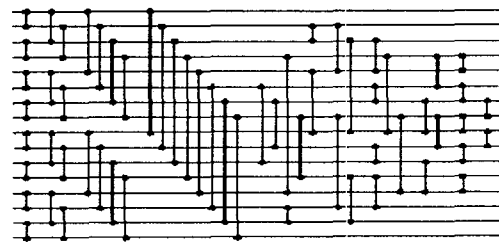


그림 3. Green 모델: 60 비교기, 10 평행 스텝, 16 입력 정렬 네트워크

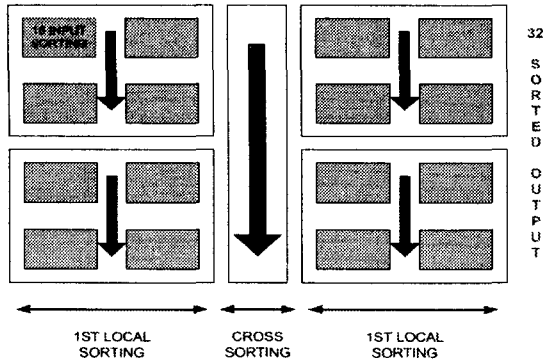


그림 4. Cross Sorting 방법

III. Residue Number System (RNS) 으로 표현한 유전자들의 교차 연산

약 1500년 전, 중국의 학자 Sun Tsu는 수 수계기를 냈는데, “7로 나누면 2가 남고, 5로 나누면 3이 남고, 3으로 나누면 2가 남는 수는 무엇인가?” 하는 내용이였다. 이것이 곧 residue number system(RNS)의 표현(2|3|2)_{RNS(7|5|3)}의 개념을 나타내고 있다[5].

RNS에서 어떤 수 x 는, k 개의 서로 소 관계의 제수들(*prime moduli* : $m_{k-1} > \dots > m_1 > m_0$)로 나누었을 때의 나머지(자연히 k 개가 생성됨)로 표현된다. 따라서 각 자릿수 x_i 는 그 자리에 해당된 prime moduli m_i 로 ($0 \leq i \leq k-1$) 나누었을 때의 나머지를 나타낸다. 기호로 나타내면,

$$x_i = x \bmod m_i = \langle x \rangle_{m_i}$$

와 같다. 모든 m_i 들의 곱을 *dynamic range*라 하고 이것은 해당 RNS수가 나타낼 수 있는 수들의 개수를 의미한다. 즉, *dynamic range*를 M 이라 하면,

$$M = m_{k-1} \times \dots \times m_1 \times m_0$$

이다. 음수는 다음과 같이 표현 하기로 정의한다.

$$\langle -x \rangle_{m_i} = \langle M - x \rangle_{m_i}$$

한 가지 예외적인 것은 0은 바뀔 없이 그대로 나타낸다[3][5].

RNS에서 덧셈, 뺄셈, 곱셈(나눗셈은 불가)은 각 자리끼리의(digit by digit) 연산으로 매우 간단히 이루어진다. 예를 들어 prime moduli $m_3=8, m_2=7, m_1=5, m_0=3$ 으로 표현되는 어떤 RNS수 $x=(6|6|1|0)_{RNS(8|7|5|3)}$ 은

+6을 나타내고, $y=(6|5|3|1)_{RNS(8|7|5|3)}$ 은 -2를 나타낸다. $x+y$ 는 $\langle 6+6 \rangle_8=4, \langle 6+5 \rangle_7=4, \langle 1+3 \rangle_5=4, \langle 0+1 \rangle_3=1$ 이므로 그 결과는 $(4|4|4|1)_{RNS(8|7|5|3)}$ 이 되고 이것은 $x+y=6+(-2)=4$ 와 같은 결과임을 알 수 있다. 뺄셈을 해보면 $x-y$ 는 $\langle 6-6 \rangle_8=0, \langle 6-5 \rangle_7=1, \langle 1-3 \rangle_5=3, \langle 0-1 \rangle_3=2$ 와 같으므로 $(0|1|3|2)_{RNS(8|7|5|3)}$ 이 되고 이것은 $x-y=6-(-2)=8$ 과 같다. 곱셈은 $(4|2|3|0)_{RNS(8|7|5|3)}$ 으로 -12가 된다.

RNS의 장점은 위와 같이 산술적 연산이 쉽게 이루어진다는 점도 있지만, 이것이 유전 알고리즘에서의 교차연산에 강력한 이점을 가지고 있다는 것이다. 첫째로, 교차 연산에서 자름선의 기준을 두는 문제에 있어서 RNS는 그 표현법 자체에서 명확한 자름선의 기준을 제공한다. 둘째로, 이전 표현이나 실수 표현에서는 자식해 역시 부모해의 형질을 일정 정도 포함하고 있어 사실상 해들의 탐색공간이 제한되나, RNS에서의 교차 연산은 부모해와 매우 상이한 자식해를 생성해 내므로 교차 연산과 더불어 변이의 효과도 가져온다는 것이다. 위의 예에서 $x=(6|6|1|0)_{RNS(8|7|5|3)}=+6, y=(6|5|3|1)_{RNS(8|7|5|3)}=-2$ 를 가지고 가운데 선을 기준으로 6|6 과 3|1을 교차 연산하면 $(6|6|3|1)_{RNS(8|7|5|3)}$ 가 되고 이것은 십진수로 +118를 의미한다. 따라서, 이를 통해 상당히 넓은 해공간을 탐색할 수 있음을 알 수 있다. 자세한 교차 연산 과정을 그림 5에 나타내었다.

IV. 유전 알고리즘 (Genetic Algorithm) 처리속도 향상을 위한 프로세서 구조

이 장에서는 2장과 3장의 내용을 바탕으로 설계한 유전 알고리즘(Genetic Algorithm: GA) 프로세서 설계에 대해 소개한다.

프로세서 설계에 있어서 가장 중요한 부분은 산술/논리 연산을 담당하는 ALU(Arithmetic and Logic Unit)의 구성이다. 전형적인 ALU는 ① Arithmetic 연산부 - PC(Program Counter)를

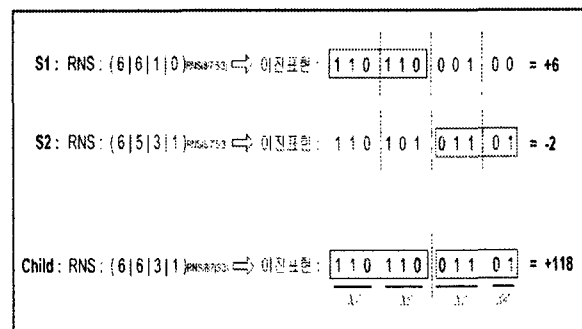


그림 5. RNS 수를 이용한 일점 교차 연산

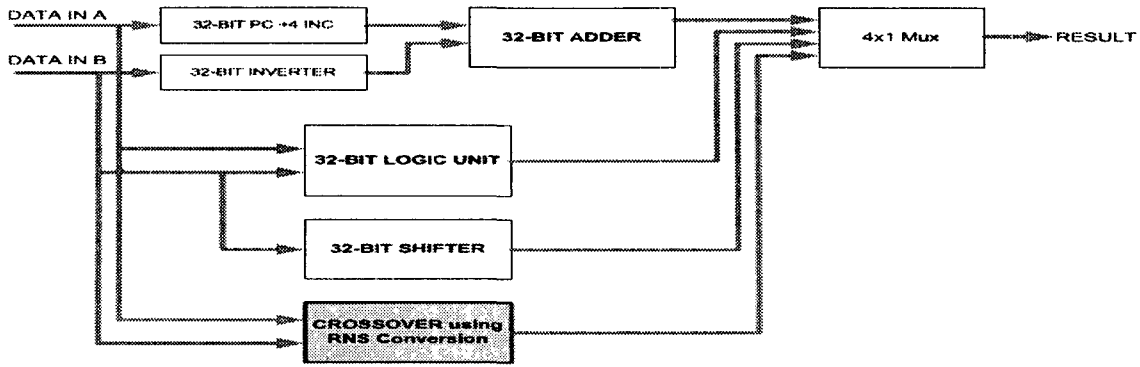


그림 6. RNS 변환을 통한 교차 연산부가 추가된 ALU 구조

증가시키는 부분, Adder에서 $A-B$ 를 수행하기 위한 inverter 부분, 그리고 Adder부분, ② Logic 연산부 - and, or, not, $C=B$, etc. ③ Shift 연산부 - shr, shra, shl, shc, ④ 어떤 연산의 결과를 Result로 내보낼 지 결정해 주는 Multiplexer로 이루어진다. 본 논문에서는 여기에 RNS를 이용한 교차 연산부를 추가하였다. 그림 6은 RNS를 이용한 교차 연산부(회색 블록 안)를 전형적인 ALU의 구조에 첨가한 구조를 보여주고 있다[6].

2장에서 해들의 순위를 정렬 네트워크를 통해 평가하기로 하였다. 16개의 해들의 비교를 위해 범용 레지스터(General Purpose Register)에 연결해 주었다. 정렬된 결과는 곧바로 데이터 메모리에 저장되고, 순위별로 저장된 해들은 일차 함수 형태의 적합도 배정 함수에 의해 적합도를 배정받고 다음 세대 연산에 들어간다.

반 선택에 있어서의 해들의 품질 비교를 수행하고, RNS로 표현된 해들의 교차 연산을 하드웨어적으로 수행하는 유전 알고리즘 프로세서 설계를 보였다.

이 구조는 아직까지 소프트웨어적으로 처리 가능한 여러 복잡한 문제들을 해결하기에는 시간적·공간적 제약이 많이 따른다. 그러나 범용 프로세서가 아닌 유전 알고리즘 처리를 위한 전용 프로세서 구조를 구성했다는 점에서 연구의 의미를 찾을 수 있다 하겠다.

감사의 글: 본 연구는 과학기술부의 뇌신경정보학연구사업의 '뇌정보처리에 기반한 감각정보 융합 및 인간행위 모델 개발'의 연구비 지원으로 수행되었습니다. 연구비 지원에 감사드립니다.

V. 결론

본 논문에서는 정렬 네트워크를 통해 순위기

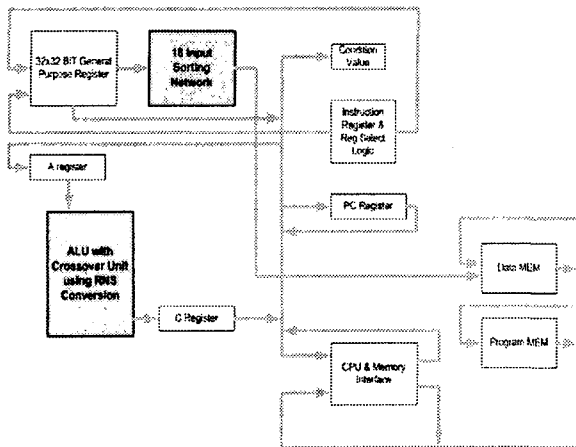


그림 7. 유전 알고리즘 프로세서의 전체 구조

VI. 참고문헌

- [1] 강훈, 심귀보, *지능정보시스템*, 브레인코리아, 2003
- [2] 문병로, *유전알고리즘*, 두양사, 2003.
- [3] Behrooz Parhami, *Computer Arithmetic*, Oxford Newyork: 2000.
- [4] H. W. Lang. (2002, Dec 12). Sorting Network. [Online]. Available: <http://www.iti.fe-flensburg.de/lang/algorithmen/sortieren/sortieren.htm>
- [5] Han-Ul Yoon, Kyoung-Taik Park, and Kwee-Bo Sim, "Improvement of processing time using residue number system and sorting network in controller design," *IECON 2004 Computer and Control Systems*, pp. 1-6, 2004.
- [6] V. P. Heuring and H. F. Jordan, *Computer System Design and Architecture*, Prentice Hall, 1997.