논문 스터디 5주차

Oct 1, 2021

Genetic Algorithm

UNIST brAIns 유인재

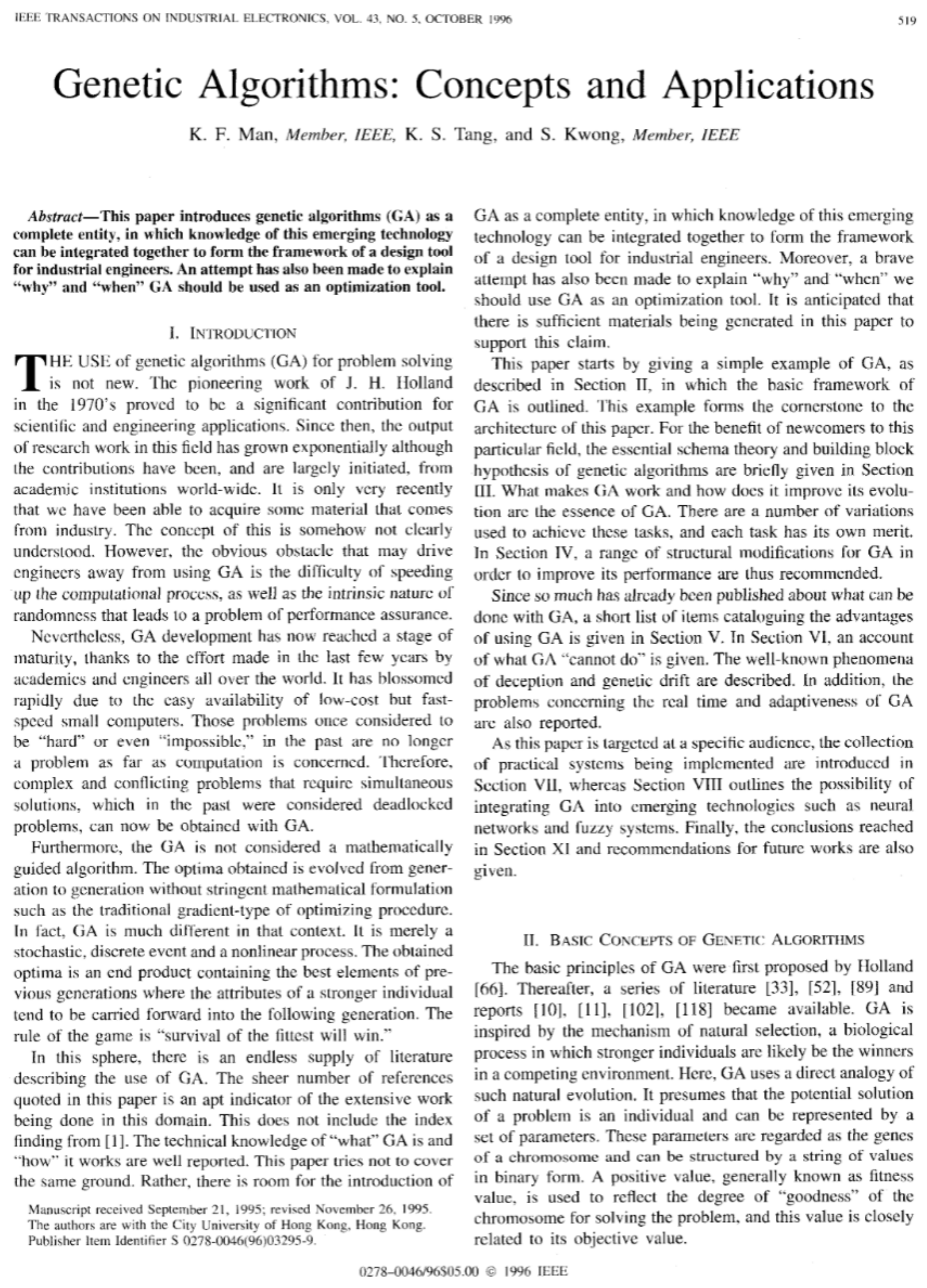
CONTENTS

Intro.

Intro.

Genetic Algorithms : Concepts and Applications (1996), IEEE

K. F. Man, K. S. Tang, and S. Kwong, Member, IEEE



IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 43, NO. 5, OCTOBER 1996 519

Genetic Algorithms: Concepts and Applications

K. F. Man, *Member, IEEE*, K. S. Tang, and S. Kwong, *Member, IEEE*

MAN *et al.*: GENETIC ALGORITHMS: CONCEPTS AND APPLICATIONS 533

Intro.

- 오늘 논문은 **review paper**입니다.

- 대부분의 내용들이 **Appendix와 reference를 바탕**으로 작성되어 있습니다.
  (자세하게 더 알아보고 싶은 내용이 있다면 번호로 지정된 해당 논문을 살펴보시면 좋을 것 같습니다)

- Genetic Algorithm (GA)에 대한 **전반적인 흐름**과
  Neural Network와의 비교/분석을 중심으로 보시면 좋을 것 같습니다.

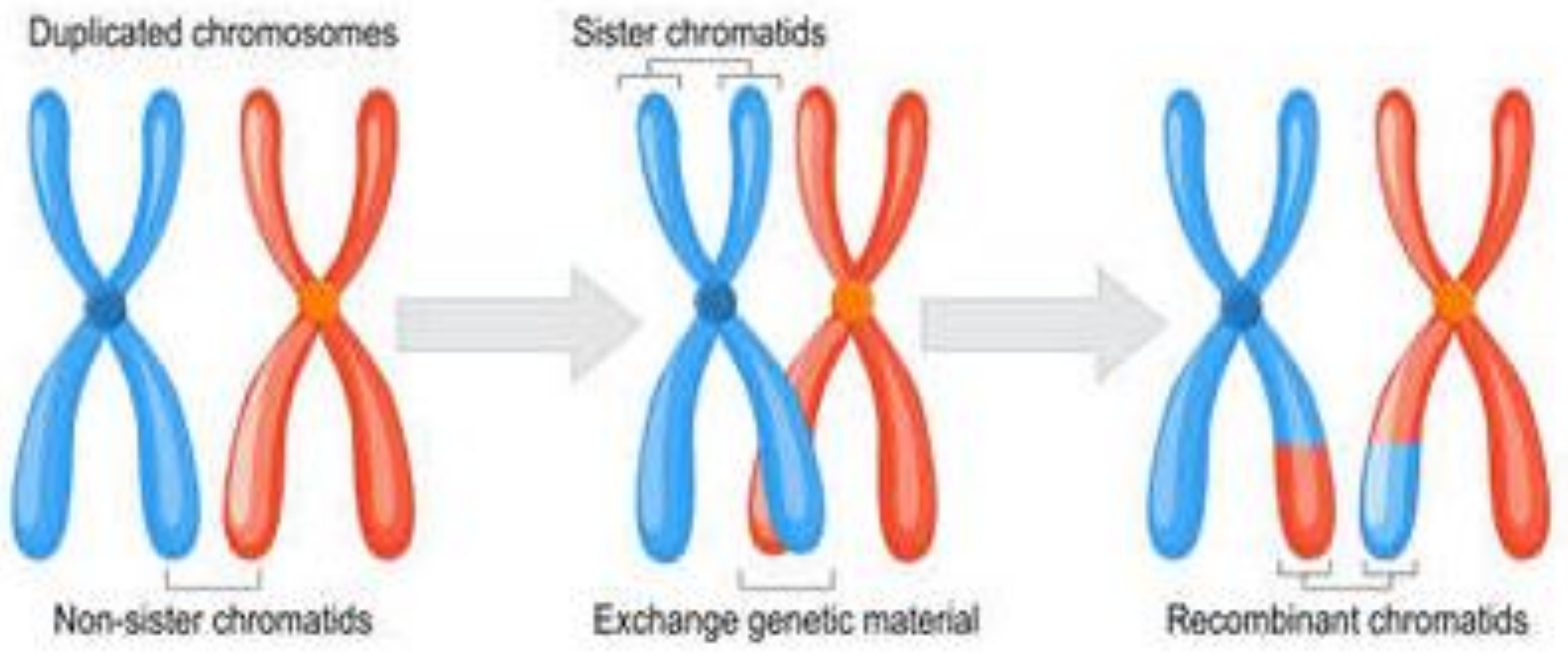
- 설명 중에 **생물학적인 내용과 단어**들이 자주 사용될 수 있습니다.
  (이해가 되지 않는 내용이 있다면 바로 질문해주시면 됩니다!)

- **1990년대 논문**이다 보니 다소 **최신 기술에 대한 적용(Application)이 부족**할 수 있습니다.
  (GA의 현재 활용방안에 대해 더 자세히 알고 싶다면 Applications of GA로 검색하시면 됩니다!)

Basic Concepts

Basic Concepts

Crossover (교차)



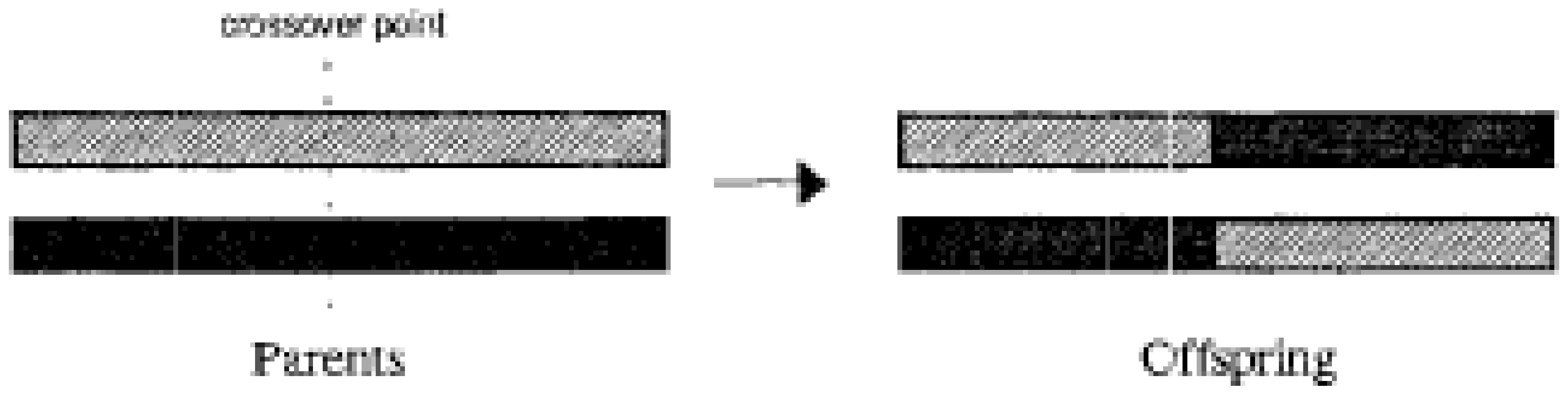
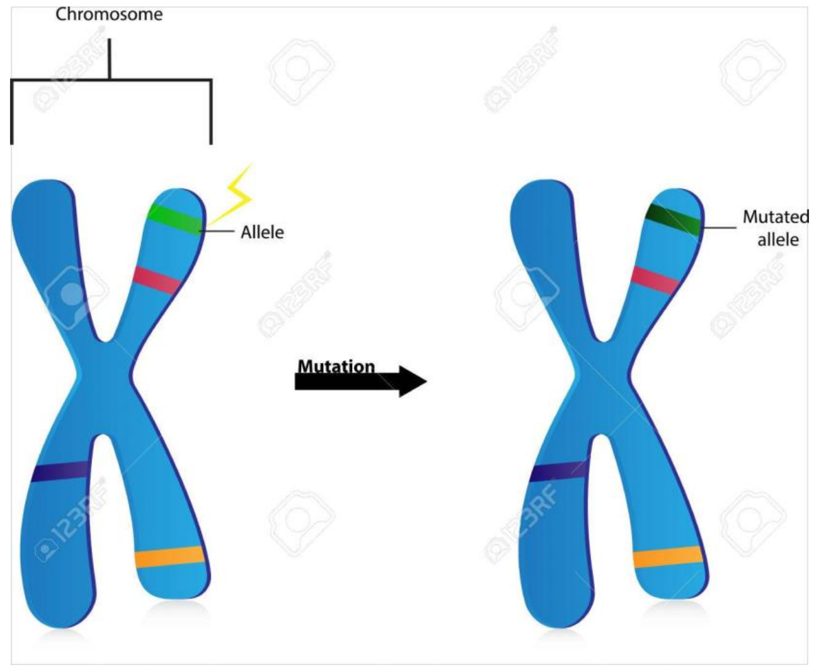


Fig. 1. Example of one-point crossover.

Crossover rate (p_c)

Mutation (돌연변이)



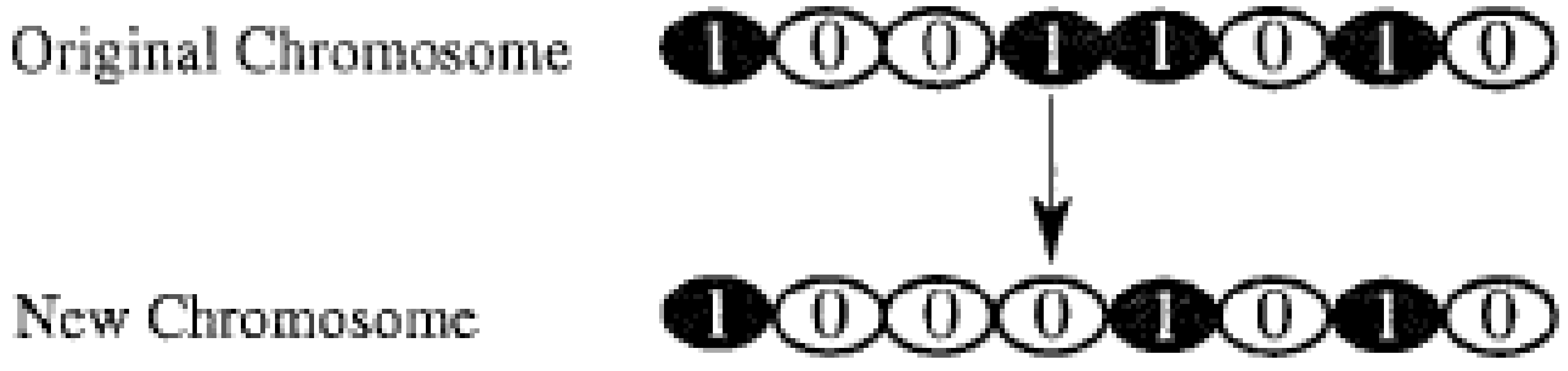


Fig. 2. Bit mutation on the fourth bit.

Mutation rate (p_m)

Basic Concepts

STEP 1: Parent Selection

	First Population	Objective Value z = f(x,y)
	1100110110101000	3.481746
	0101010110110101	3.668023
→	1000010100110110	6.261380
→	1101011111001100	12.864222

Population size = 4

X(8bit), Y(8bit) → 16bit

STEP 2: CROSSOVER

11010|11111001100 → 11010|10100110110

10000|10100110110 → 10000|11111001100

crossover point

STEP 3: MUTATION

1101010100110110 ↓ 1111010100100110

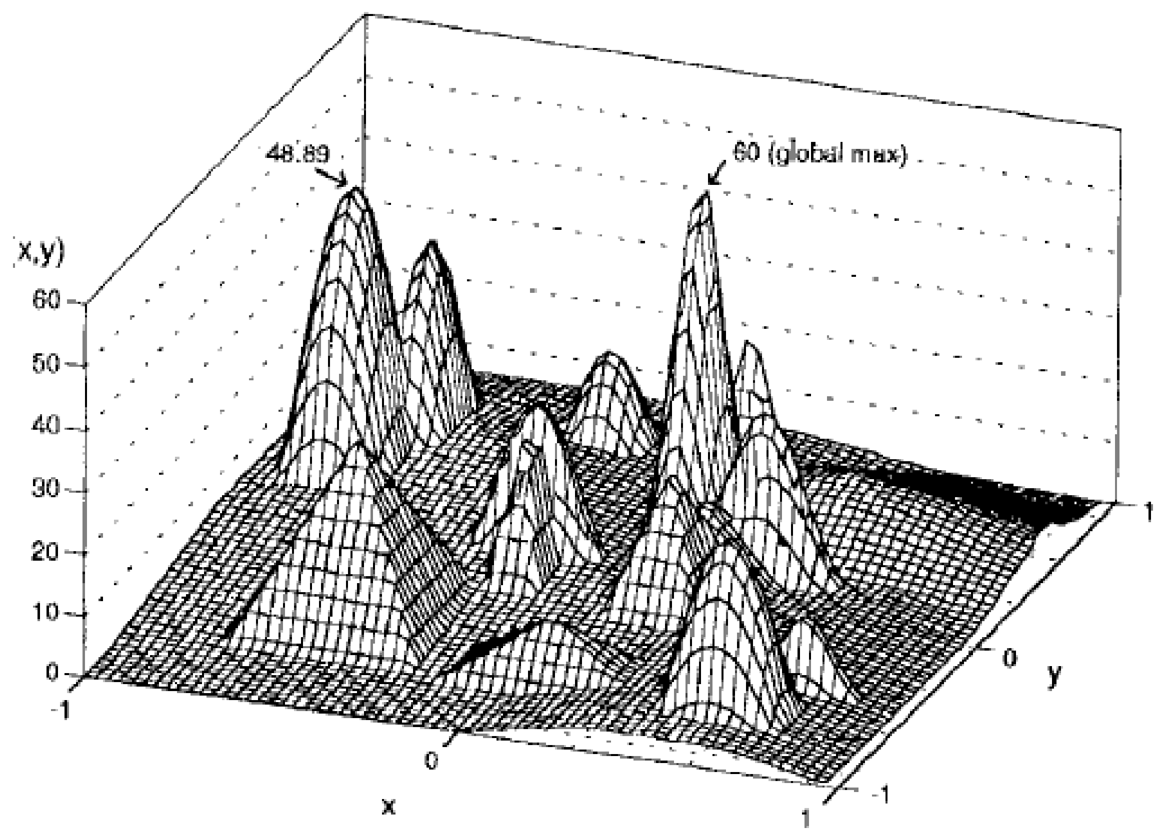
z = 8.044649

1000011111001100 ↓ 1000011111001100

z = 6.092550

STEP 4: Reinsertion

	Second Population	Objective Value z = f(x,y)
→	1111010100100110	8.044649
→	1000011111001100	6.092550
	1000010100110110	6.261380
	1101011111001100	12.864222



< Question 1 >

Crossover rate (p_c)와 Mutation rate (p_m)의 값의 변화에 따른 후손들의 발생 양상은 어떻게 변화할까?

* case 1, p_c=1이고 p_m=0 인 경우!

* case 2, p_c=0이고 p_m=1 인 경우!

Case 1,
단순 정보의 전달 혹은 재조합

Case 2,
무작위 탐색 문제

Theory & Hypothesis

Theory & Hypothesis

Schema Theory

low-order, above-average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm.

낮은 차수의 평균이상의 schemata는
유전자 알고리즘의 후속세대에서
기하급수적으로 증가하는 시도를 받는다!

Schema? sets of strings (encoded form of the chromosome) that have one or more features in common.
형질에 대한 정보를 가지고 있는 문자열

Building Block Hypothesis

A genetic algorithm seeks near-optimal performance through the juxtaposition of low -order, high-performance schemata, called the building block

유전 알고리즘은 building block이라고 하는
낮은 차수의 높은 성능을 가진 schemata의 병치를 통해
최적에 가까운 성능을 찾는다!

Structure Modification

〈 Question 2 〉

GA에서 변경 가능한 부분(조작변인)이 될 수 있는 것은 무엇이 있을까?

Structure Modification

1. Chromosome Representation
2. Objective and fitness value
3. Selection mechanism
4. Crossover operations
5. Reordering / inversion
6. Reinsertion
7. Probability rates setting
8. Parallel GA
9. Structed GA

Structure Modification

1. Chromosome Representation

First Population
1100110110101000
0101010110110101
1000010100110110
1101011111001100

- 보통의 유전자는 이진데이터로 전환하여 사용한다.
- 보다 직관적인 문자열 기반 유전자 표현도 사용 가능하지만 실제 설계된 GA가 일부 상황에서 반드시 좋은 결과를 산출하지 못할 수 있다.
- 다양한 유전자 표현 방식이 존재하지만 일부 최적화 문제에 대해 어렵고, 때로는 부자연스러운 결과를 초래할 수 있다.

Ex) Order-based representation, Embedded list

Structure Modification

2. Objective and fitness value

- 만들어진 유전자(데이이터)를 평가하는 메커니즘은 필수적인 단계
- 염색체를 입력으로 사용하고 염색체의 성능에 대한 척도로 숫자 또는 객관적인 값 생성
- Linear Scaling

$$f_i = a \cdot O_i + b$$

- Power Law Scaling

$$f_i = O_i^k$$

f_i :fitness value O_i : Objective value of chromosome i

Structure Modification

4. Crossover operations

- 교차점은 굳이 한 개일 필요가 없다. (Multipoint Crossover)
- Crossover Mask를 통해 규칙적으로 특정지점에 대해서 교차를 진행할 수 있다.
- 교차 방식에 대한 논쟁도 현재 진행중이다.

Ex) 효과적인 교차 지점의 개수, building block 보존 성능

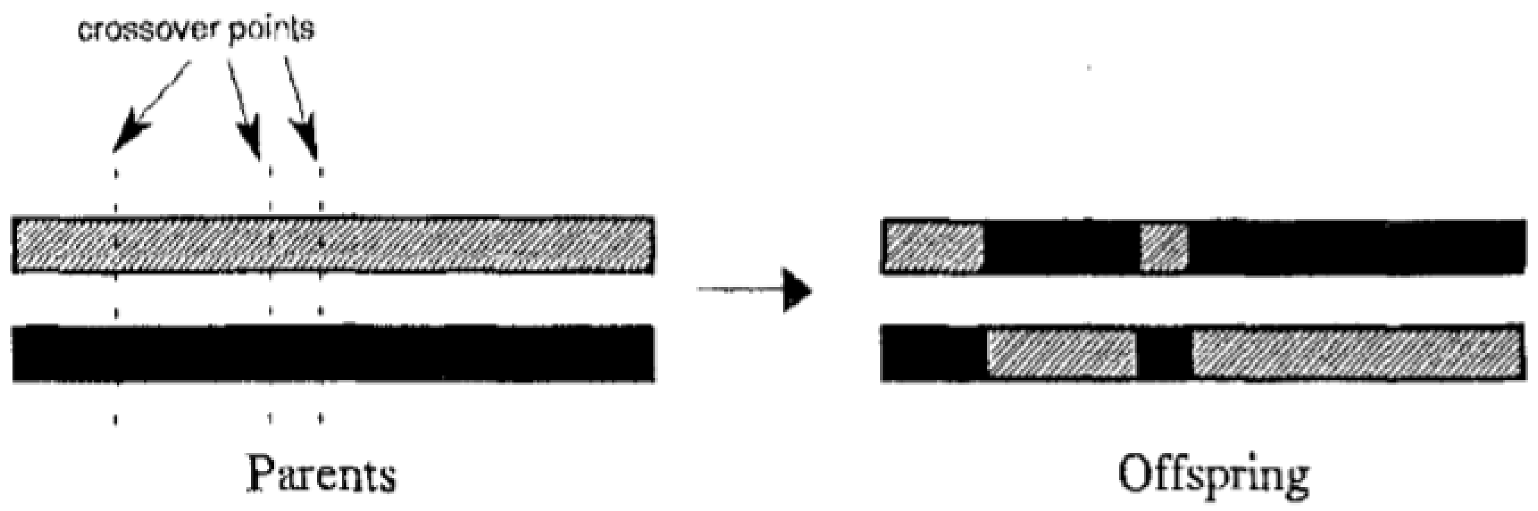


Fig. 7. Example of multipoint crossover, ($m = 3$)

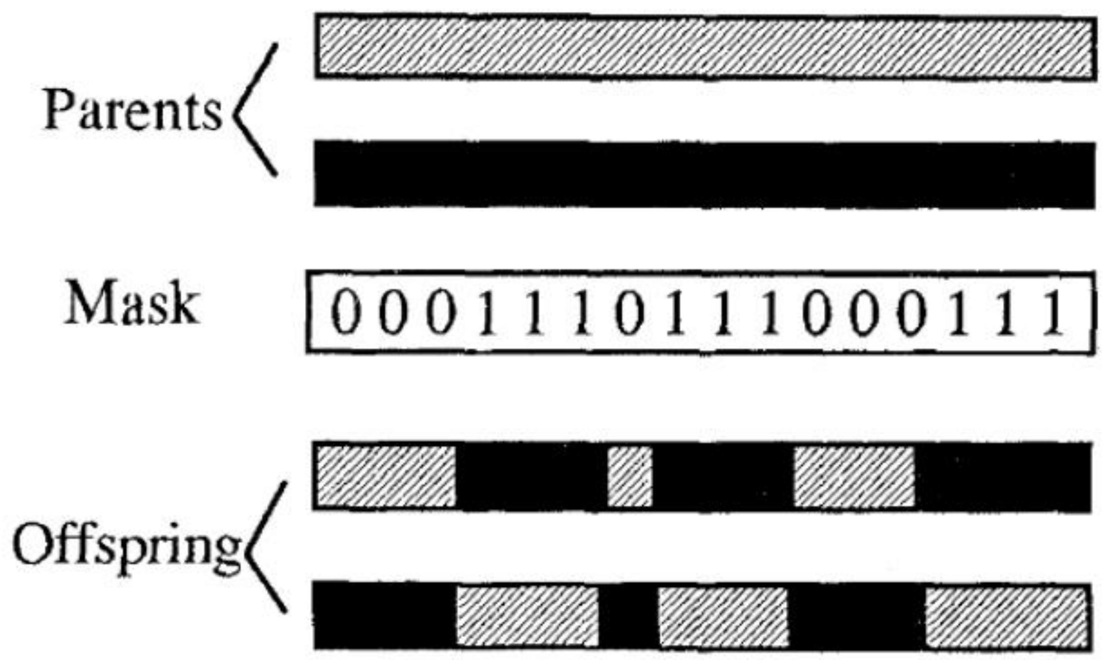


Fig. 8. Example of uniform crossover.

Structure Modification

7. Probability rates setting

- GA를 설계할 때 변경할 수 있는 확률 변수는 $\boldsymbol{p_c}$ 와 $\boldsymbol{p_m}$ 이다.

- 특정 값으로만 설정하는 것이 정답은 아니기에 상황에 맞게 적절한 조정이 필요하다.

- 보편적으로 $\boldsymbol{p_m \leq 0.1}$ 로 설정하여 과도한 무작위 문제로 바뀌는 것을 방지한다.

- 효과적인 값의 설정에 대한 논쟁도 여전히 진행중이다.

Structure Modification

8. Parallel GA

- GA가 받고 있는 비판은 바로 계산에 소요되는 시간이다.
- 하지만 해를 구하는 방정식을 찾는 것보다 수학적이지 않기에 이해가 쉬운 것은 자명하다
- 병렬구조를 활용에 이에 대한 단점을 보완할 수 있다.
- 병렬화 방법에는 전역(Global), 이주(Migration), 확산(diffusion)이 있다.

Structure Modification

8. Parallel GA (Global)

- Global GA는 전체 인구를 단일 번식 메커니즘으로 취급한다.
- 공유 메모리 다중 프로세서 또는 분산 메모리 컴퓨터에서 구현 가능하다.
- Master-Slave 관계를 기반으로 한다.
  (Master가 작업을 처리하는 동안 Slave는 대기해야 한다는 단점이 있다)

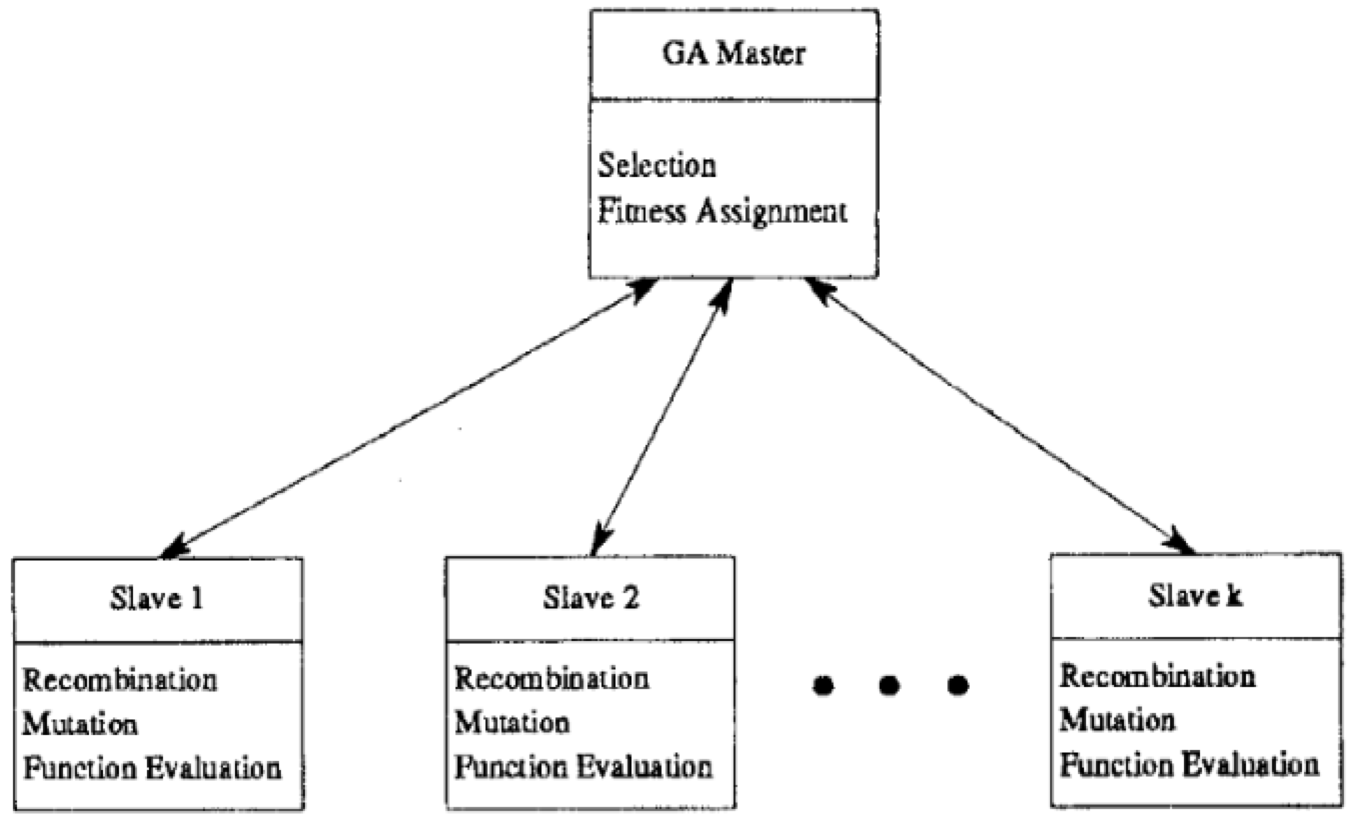


Fig. 9. Global GA.

Structure Modification

8. Parallel GA (Migration)

- Migration GA는 개체군을 여러 하위 개체군으로 나눈다.
  (각 개체군은 별도의 번식 단위로 처리된다.)
- 하위 개체군 간의 이동은 때때로 발생한다. (이동 방식에 따른 차이가 있다)

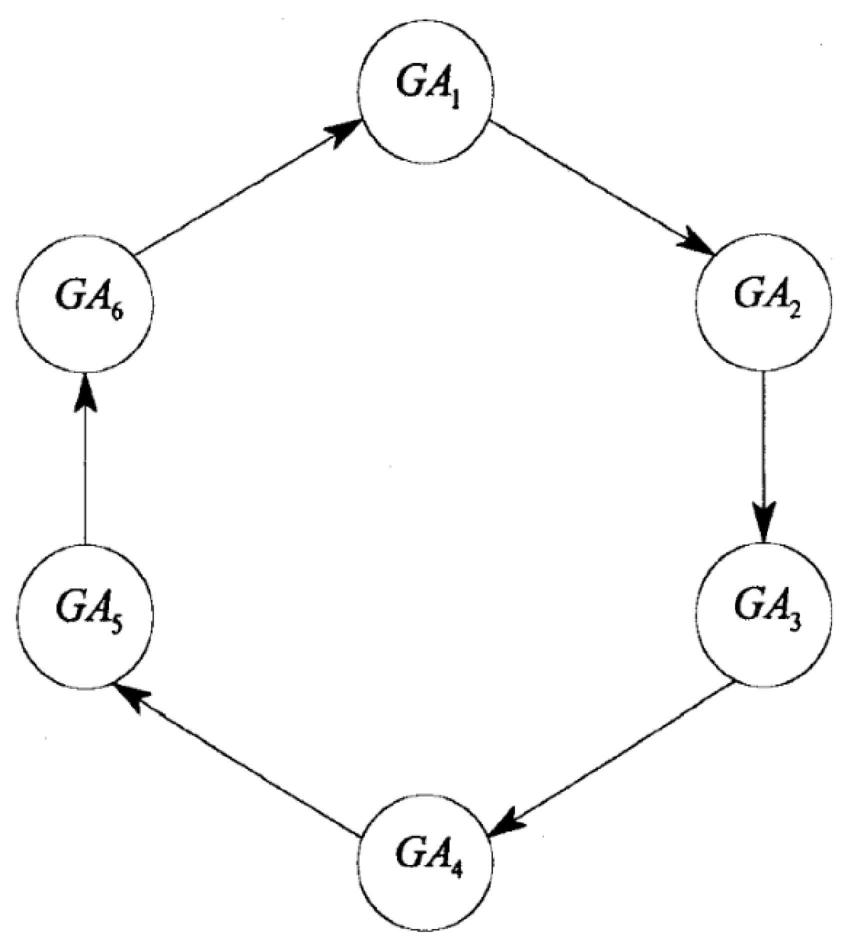


Fig. 10. Ring migration topology.

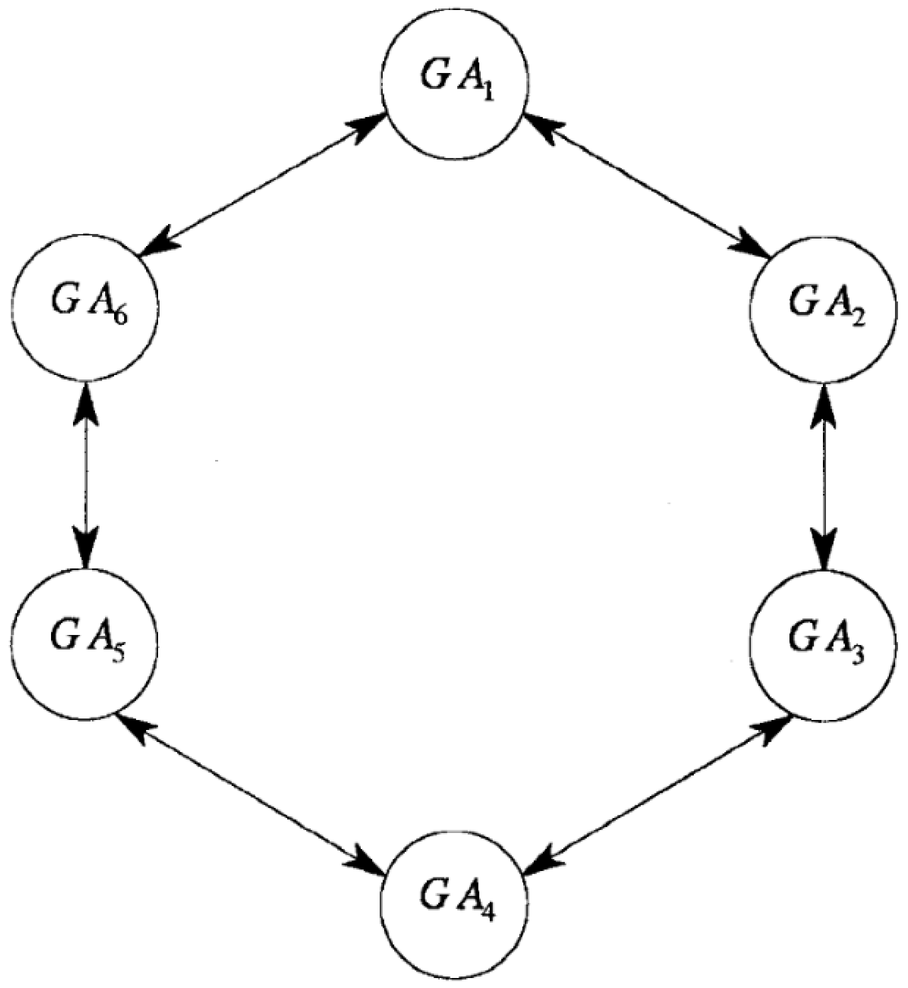


Fig. 11. Neighborhood migration topology.

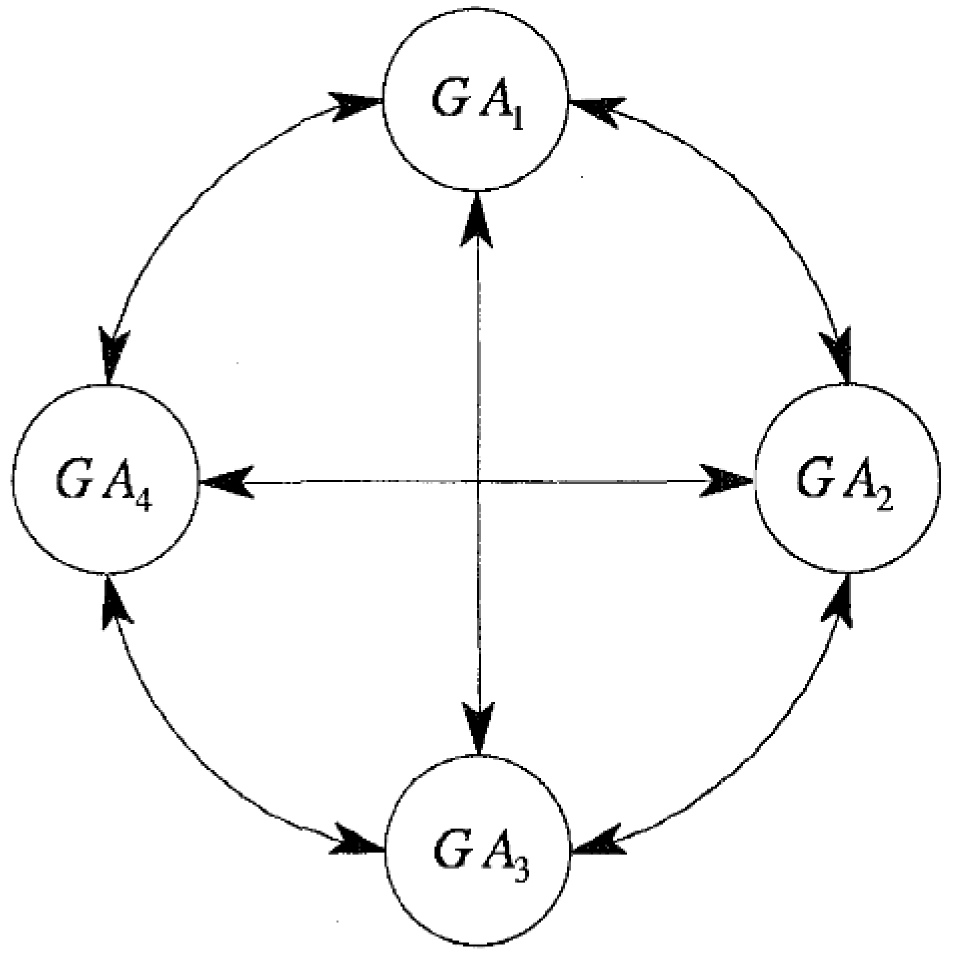


Fig. 12. Unrestricted migration topology.

Structure Modification

8. Parallel GA (Diffusion)

- Migration GA는 인구를 단일 연속구조로 간주한다.
- 2차원 그리드에서 인접한 이웃끼리 데이터의 공유가 가능한 것이 특징이다.

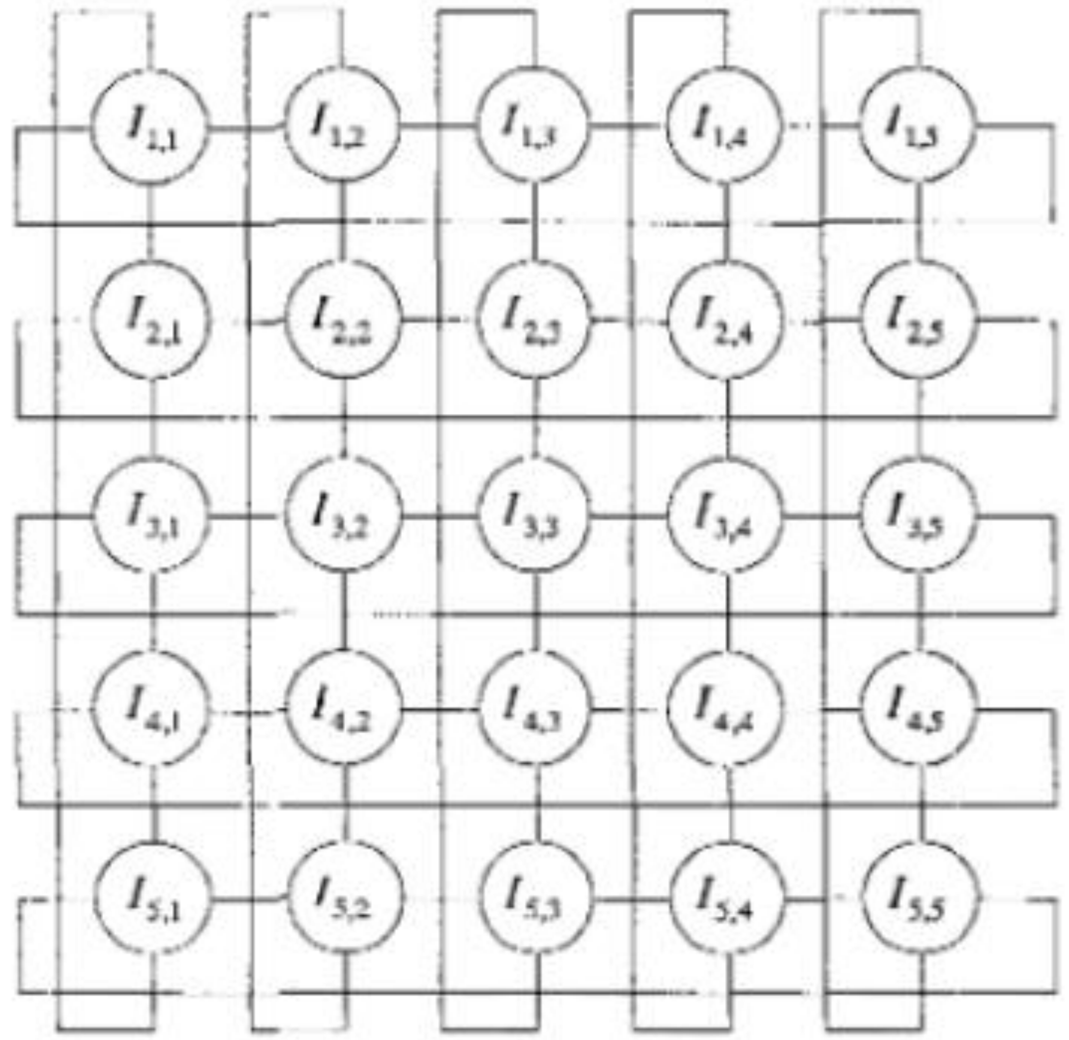
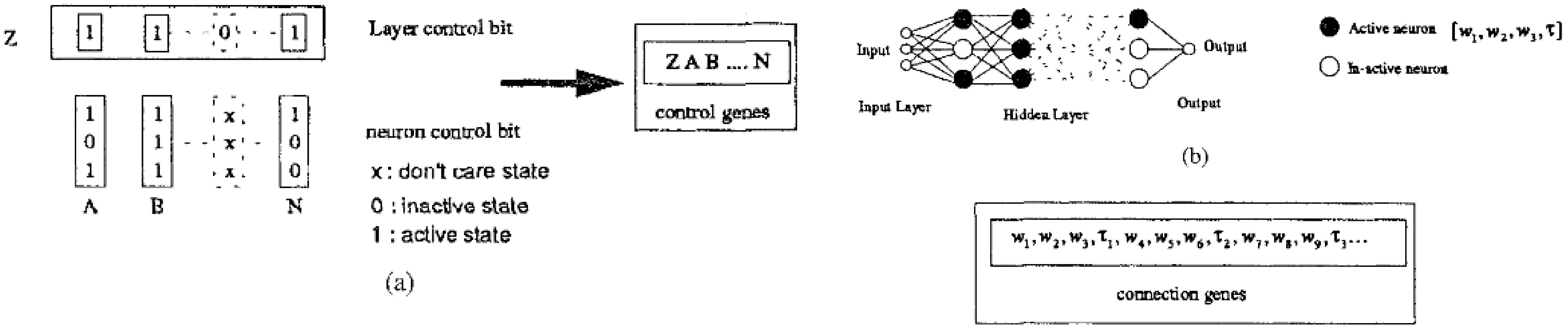


Fig. 13. Diffusion GA.

Structure Modification

9. Structed GA

- Structed GA는 계층구조로 구성된다. (Neural Network와 상당히 유사)
- 구조의 상위 수준 노드는 하위 수준 유전자의 활성화를 제어하게 된다.
  (but, 비활성화된 유전자는 변화하는 환경에 반응할 수 있는 추가 정보로 쓰인다.)
- 유전적 다양성을 보존을 위해 하위 구조로 구성된다.



Applications

Applications

1. Parameter and system identification
2. Control
3. Robotics
4. Pattern / Speech recognition
5. Engineering Designs
6. Planning and scheduling
7. Classifier system

Conclusion

Conclusion

- **GA 고유의 능력, 특성을 설명하는 시도**를 이 논문에서 찾아볼 수 있었다.
- GA에 대한 지식이 거의 없는 사람들에게 이 **기술을 소개하는 논문**이다.
- GA의 **남은 과제**는 의심할 여지 없이 **실기간 및 적응형 기능**이다.

Question?

Thank you