



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

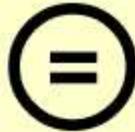
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학박사 학위논문

유전자 알고리즘을 이용한 센서 네트워크의
수명연장에 관한 연구



2007년 8 월

부 경 대 학 교 대 학 원

정 보 시 스템 학 과

성 기 택

공학박사 학위논문

유전자 알고리즘을 이용한 센서 네트워크의
수명연장에 관한 연구

지도교수 우 종 호

이 논문을 공학박사 학위청구논문으로 제출함

2007년 8 월

부 경 대 학 교 대 학 원

정 보 시 스템 학 과

성 기 택

성기택의 공학박사 학위논문을 인준함

2007년 8 월 30 일

주 심 공 학 박 사 성 길 영 ㉠

위 원 경 영 학 박 사 어 윤 양 ㉠

위 원 공 학 박 사 서 경 룡 ㉠

위 원 공 학 박 사 권 오 흠 ㉠

위 원 공 학 박 사 우 중 호 ㉠

목 차

Abstract	6
제 1 장 서 론	8
1.1 연구의 필요성과 목적	8
1.2 연구배경	10
1.3 연구의 내용	12
제 2 장 관련 연구	14
2.1 센서 네트워크의 커버문제	14
2.2 센서 네트워크의 수명	17
2.3 목표지점커버와 지역커버 문제	18
제 3 장 목표지점 커버문제	20
3.1 목표지점 커버문제	20
3.2 최대커버집합 문제	22
3.3 MSC 문제를 해결하는 기존의 방법들	25
제 4장 센서노드 스케줄링 메카니즘의 고찰	27
4.1 센서노드 스케줄링 프로세스	27
4.2 기존의 스케줄링 알고리즘 고찰	28
제 5 장 유전자 알고리즘을 이용한 MSC 문제의 해	33
5.1 유전자 알고리즘	33
5.1.1 유전자 알고리즘의 역사	33
5.1.2 유전자 알고리즘의 기본구조	34

5.1.3	스키마 정리	36
5.1.4	병렬 유전자 알고리즘	38
5.1.4.1	병렬 유전자 알고리즘의 개요	38
5.1.4.2	병렬 유전자 알고리즘의 분류	39
5.2	중앙처리방식의 유전자 알고리즘	43
5.2.1	염색체의 표현	43
5.2.2	선택연산	47
5.2.3	교차연산	49
5.2.4	돌연변이 연산	53
5.2.5	평가함수	54
5.3	병렬처리방식의 유전자 알고리즘	57
5.3.1	병렬 유전자 알고리즘의 구성	57
5.3.2	시스템의 동작	59
5.3.3	센터노드의 구성 및 동작	61
5.3.4	클라이언트의 구성 및 동작	61
제 6 장	시뮬레이션 및 고찰	63
6.1	유효성 검정 시뮬레이션	64
6.1.1	유전자 알고리즘방식의 유효성	65
6.1.2	유전자 알고리즘의 특성 평가	66
6.1.2.1	교차율 변화에 따른 수명 평가	67
6.1.2.2	돌연변이율 변화에 따른 수명 평가	68
6.2	중앙처리방식과 병렬처리방식의 비교	69
6.2.1	세대수에 따른 네트워크 수명값의 비교	70

6.2.2 센서 수에 따른 수명	72
6.3 센서노드의 모드전환	73
제 7 장 결론	78
참 고 문 헌	80



그림 목 차

그림 1.1	센서노드의 구조	8
그림 2.1 (a)	지역 커버	16
그림 2.1 (b)	지점 커버	17
그림 3.1	3개의 목표지점과 4개의 센서로 구성된 MSC 문제	22
그림 4.1	센서노드 스케줄링 메카니즘	28
그림 4.2	센서노드 스케줄링 메카니즘(중앙처리방식)	29
그림 4.3	MSC 문제에 대한 greedy-heuristic 알고리즘	30
그림 4.4	센서노드 스케줄링 메카니즘 (greedy-heuristic)	31
그림 5.1	유전자 알고리즘의 동작	35
그림 5.2	병렬 유전자 알고리즘의 분류	39
그림 5.3	전역병렬 유전자 알고리즘의 개념도	40
그림 5.4	분산 유전자 알고리즘의 구조	42
그림 5.5	염색체의 부호화	44
그림 5.6	염색체 부호화의 예	46
그림 5.7	룰렛-휠 선택기법의 공간배정	48
그림 5.8	교차연산 과정	49
그림 5.9	교차연산의 상세절차	51
그림 5.10	교차연산의 예	52
그림 5.11	돌연변이연산의 과정	54
그림 5.12	돌연변이연산의 상세 과정	55

그림 5.13 돌연변이연산의 예	56
그림 5.14 네트워크에서 센터노드와 클라이언트 노드구조	58
그림 5.15 전체 시스템의 동작	60
그림 5.16 센터노드의 구성	61
그림 5.17 클라이언트의 구성	62
그림 6.1 센서의 수에 따른 수명 비교(target 수 : 5)	66
그림 6.2 CGA에서 교차율 변화에 따른 수명	67
그림 6.3 CGA에서 돌연변이율 변화에 따른 수명	69
그림 6.4 세대 수에 따른 네트워크 수명(노드 수 : 100, target 수: 10)	71
그림 6.5 노드 수에 대한 네트워크 수명(target 수 : 50)	72

표 목 차

표 6.1 모드전환횟수와 수명(단일 노드 에너지)	75
표 6.2 모드전환횟수와 수명(다양한 노드 에너지)	76

A Study on Prolonging the Lifetime of Sensor Networks by Genetic Algorithm

Ki-Taek Seong

Interdisciplinary Program of Information System,

Graduate School

Pukyong National University

Abstract

A critical aspect of applications with wireless sensor networks is network lifetime. Power-constrained wireless sensor networks are usable as long as they can communicate sensed data to a processing node. Therefore, judicious power management and scheduling of sensor node's operation can effectively extend network lifetime.

Active node selection protocol based MSC(maximum set cover) problem was suggested for prolonging the lifetime of the sensor networks in target coverage applications. MSC model is NP-complete and for this, some optimization

algorithms were proposed.

These algorithms have following features that

- 1) algorithms are centralized
- 2) algorithms need too much run-time in huge sensor networks
- 3) change of network's operation modes between sensing and scheduling

The genetic algorithm is a global optimum random search algorithm based on the principles of evolution and natural genetics.

In this thesis, applied GA(genetic algorithm) as a new method to MSC problem for extending networks lifetime and reducing the number of mode exchange on networks operations. To apply the genetic algorithm to MSC problem, designed a new chromosome encoding, crossover and mutation process and then, evaluated the of effectiveness suggested methods (GA based a centralized and a parallel version) by the software simulation.

Simulation results showed that suggested GA based algorithms were suitable and effective on MSC problem and produced better results on QoS of network's continuous sensibility.

The contributions of this thesis are

- i) applied GA on NP-complete MSC problem
- ii) developed parallel GA on considering network topology
- iii) verified that GA based methods are suitable and produced better results on QoS of network's continuous sensibility

제 1 장 서 론

1.1 연구의 필요성과 목적

유비쿼터스에 대한 관심과 최근의 전자 및 기계기술의 발달로 인하여, 무선 통신장치와 다양한 종류의 센서 및 마이크로프로세서의 집적화가 가능하게 되었으며, 이와 함께 센서노드의 발전과 함께 무선 센서 네트워크(WSN: wireless sensor networks)에 관한 연구가 활발히 진행되고 있다. 다수의 센서노드들에 의하여 무선 통신 네트워크를 형성하는 센서 네트워크는 지역의 감시, 생물학적 환경감시, 지형지물의 감시, 건강관리 등의 다양한 분야에서 적용되고 있다 [1 - 9]. 그림 1.1은 하나의 센서노드에 대한 기본적인 구조를 보여주고 있다 [10].

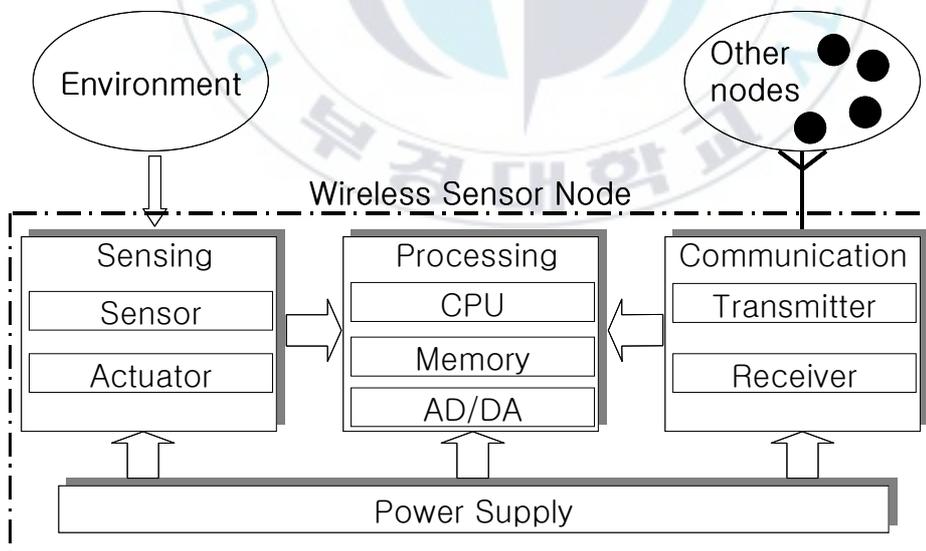


그림 1.1 센서노드의 구조

Fig. 1.1 Sensor node architecture

그림에 나타난 바와 같이 하나의 센서노드는 외부환경에서 발생하는 다양한 이벤트를 탐지하는 감지기능, 감지된 데이터의 국부적 처리(local processing)와 그 이외 기타 연산을 담당하는 정보처리기능, 그리고 이웃하는 센서노드와의 정보전달을 담당하는 무선 통신기능 등을 지원하는 하드웨어와 소프트웨어로 구성되며, 각 기능의 동작에 필요한 에너지는 자체 보유하고 있는 전원을 이용한다.

센서노드는 외부로부터의 전원공급이 불가능하기 때문에 전적으로 자체 전원을 이용하여 운영되며, 개개의 센서노드의 수명은 전체 네트워크의 수명을 결정하는 주 요인이 된다. 센서노드의 전원을 효율적으로 관리함으로써 네트워크의 수명은 연장될 수 있으므로 이에 대한 다양한 방법이 연구되어 왔다.

본 연구는 센서노드의 동작을 적절하게 스케줄링하여 네트워크 수명을 연장하는 방법에 관한 것이다. 센서노드의 동작에 있어서 통신에 사용되는 에너지 소비량은 전체 에너지 소비에 있어서 많은 부분을 차지하고 있으므로 무선통신의 적절한 제어를 통한 에너지절약 방안이 다양하게 연구되어왔다.

센서노드의 라디오는 동작에 따라 네 가지 상태 즉, 무선 송신, 무선 수신, 아이들(idle) 상태, 슬립(sleep)상태를 가지며, 이에 따라 소비되는 에너지량도 다르다. Rockwell의 WINS에 탑재된 센서의 전력소비량은, 송신상태에서는 0.38W에서 0.7W, 수신 상태에서는 0.36W, 아이들 상태에서는 0.34W, 그리고 슬립 상태에서는 0.03W이다[11].

센서 네트워크를 적용함에 있어서, 일반적으로 임무에 필요한 최소의 센서노드 수 이상의 노드가 분포된다. 따라서 잉여의 노드가 존재하게 되며 이러한 잉여의 노드를 적절하게 활용함으로써 네트워크의 수명 연장이 가능하다. 잉여 노드가 포함된 센서 네트워크에서 각 노드의 동작모드를 결정하는 것을

노드 스케줄링이라 한다.

기존의 관련연구에 의하면 노드 스케줄링 문제는 NP-complete 문제이며 이를 해결하는 최적화 알고리즘이 다양하게 제안되었다. 기존의 제안된 방법들은 센서 네트워크의 전체 운영시간을 일정 시간간격을 갖는 라운드(round)로 구분하여 각 라운드 시작전에 노드 스케줄링을 수행한다. 즉 하나의 라운드는 스케줄링 시간과 센서 네트워크의 주 임무인 감지시간으로 구성된다. 라운드의 횟수와 스케줄링 시간이 길어지면 센서 네트워크의 지속적인 감지기능에 부정적인 영향을 주기 때문에 라운드의 횟수와 스케줄링 시간의 단축이 요구된다.

본 연구에서는 노드 스케줄링 문제에 유전자 알고리즘을 적용하였다. 주어진 문제에 대하여 여러가지 유전자 연산을 통하여 최종적인 노드 스케줄링 결과를 도출하게 함으로써 네트워크의 감지 지속성을 향상시켰다. 또한 노드 스케줄링 문제에 대하여 유전자 알고리즘을 적용하여 해가 갖는 알고리즘적 특성을 분석하였다.

1.2 연구배경

에너지 자원의 적절한 관리는 센서 네트워크의 수명과 성능에 직접적인 영향을 주는 요소이므로 대단히 중요하다.

에너지를 절약하는 기술은 일반적으로 다음과 같이 분류된다.

- 1) 센서노드 동작모드 스케줄링을 통한 에너지의 효율적 관리

- 2) 통신전송 거리의 조정을 통한 에너지 제어
- 3) 효율적인 통신 라우팅 및 데이터 수집

본 연구는 1)의 분류에 속하는 것으로 다수의 센서노드가 분포된 상태에서 잉여의 센서노드를 슬립모드로 유도하는 메커니즘의 설계에 관한 것이다. 일반적인 센서 네트워크의 응용에서는 감지하고자하는 지점에 대한 위치를 알고 있는 상태에서 다수의 센서노드를 분포시킨다. 이때 특정 지점을 커버하는 복수 개의 노드가 존재할 수 있으며, 따라서 잉여의 센서노드가 발생하게 된다. 이러한 잉여 센서노드를 적절하게 운영함으로써 센서 네트워크의 수명을 연장하는 것이 가능하다.

Mihaela C. 등은 감지하고자하는 대상의 위치정보를 알고 있는 상태에서 감지대상을 커버함과 동시에 센서 네트워크의 수명을 최대화하는 문제를 정의하고 이러한 문제가 NP-complete임을 증명하였다[12]. 또한 정의한 최적화 문제에 대하여 다양한 최적화 알고리즘을 이용한 방법들을 제시하였다.

My T. Thai 등은 기존의 방식에 대하여 다음과 같은 사항을 지적하였다[13].

- 1) 알고리즘의 높은 수행시간(선형 프로그래밍 방식)
- 2) 알고리즘의 중앙처리방식(선형 프로그래밍, 발견적 greedy 방식)

기존의 방법들은 최적의 해를 구하는 최적화식을 기반으로 제안된 방식으로 대용량의 센서노드와 다수의 감지지점(target point)이 존재하는 응용환경에 적용될 경우, 많은 수행시간을 요구하게 된다. 또한 기존의 방식에서 스케줄링의 결과를 실제 센서 네트워크 운영에 반영할 때 감지기능을 중단하는 횟수가 많

아저 센서 네트워크의 지속적인 감지기능에 부정적인 영향을 준다. 이에 대한 사항은 4장에서 상세히 설명한다.

대용량의 센서 네트워크 환경을 고려할 때 네트워크 수명에 대한 최적의 값을 구하는 것도 중요하지만, 센서 네트워크의 주 임무인 감지기능을 하기 전에 노드 스케줄링에 많은 시간을 할애하는 것은 바람직하지 않다. 따라서 네트워크의 수명을 연장하는 스케줄링을 확립하되 지속적인 감지가 가능한 알고리즘의 개발이 필요하다.

1.3 연구의 내용

본 연구는 센서 네트워크의 수명을 최대화하는 최적화 문제 MSC 문제에 유전자 알고리즘 방식을 적용하는 방안에 관한 것이다.

유전자 알고리즘(genetic algorithm)은 생물의 유전시스템을 모방하여 개체군에 선택, 교차, 돌연변이 연산들을 반복적으로 적용하여 근사 최적해를 구하는 다점검색기법으로 비교적 간단하게 구현할 수 있으므로 함수의 최적화, 조합 문제, 제어문제, 기계학습 등의 다양한 분야에 적용되고 있다[14-18].

본 연구의 주요 내용은 다음과 같다.

첫째, 주어진 문제에 대한 유전자 알고리즘 모델을 제시한다.

둘째, 센서 네트워크가 갖는 구조적 특성을 이용한 병렬 유전자 알고리즘 모델 제시한다.

셋째, 시뮬레이션을 통해 제안한 방법이 유효함을 보인다.

본 논문의 구성은 다음과 같다.

제2장에서는 센서 네트워크의 수명을 연장하는 방안과 관련된 연구에 대하여

알아본다. 제3장에서는 센서 네트워크에서, 특정 목표지점을 커버하는 노드의 집합을 구하는 문제인 TCP(target coverage problem)와 함께 네트워크 수명을 최대화하는 MSC(maximum set cover) 문제에 대하여 소개한다. 제 4장에서는 MSC 문제에 대한 유전자 알고리즘 모델의 설계에 관하여 설명하며 제 5장에서는, 센서노드의 스케줄링 메카니즘에 관한 고찰을 기술하였다. 제 6장에서는 MSC 문제에 대하여, 제안한 유전 알고리즘 방식의 유효성을 확인하는 시뮬레이션의 수행 결과 및 분석에 관하여 기술한다. 제 7장에서는 시뮬레이션 분석 결과를 근거로 연구의 결론과 함께 향후 연구방향을 기술한다.



제 2 장 관련 연구

센서노드는 크기, 무게, 그리고 구조적 특성으로 인하여 가용 자원의 한계를 갖는다. 센서 네트워크 응용분야의 특수한 조건에 따라 전원을 교환하거나 보충하는 방안은 유효한 방법이 아니므로 에너지의 효율적인 관리와 함께 가능한 작은 에너지로서 동작하는 것은 센서 네트워크가 갖추어야 할 가장 중요한 기능 중 하나이다[19].

다양한 에너지 효율적인 소비 방안들이 문헌에 소개되고 있으며, 이러한 방안들은 센서노드의 하드웨어구조뿐만 아니라 알고리즘을 개발하거나 네트워크의 프로토콜을 설계하는 측면에서도 다양하게 고려되고 있다[20]. 이러한 에너지 소비문제와 함께 다루어져야 할 문제는 센서 네트워크의 커버문제이다. 커버문제란 특정영역을 대상으로 하여 센서 네트워크가 구축되었을 때, 해당지역에서 발생하는 이벤트의 감지능력을 의미하며 센서 네트워크 감지성능에 대한 QoS 척도로 활용된다.

2.1 센서 네트워크의 커버문제

센서 네트워크 커버문제의 주목적은 감지 대상으로 하는 물리적인 공간에서, 특정 목표지점 또는 영역에 대하여 최소 하나 이상의 노드에 의하여 감지 영역에 포함될 수 있도록 각 노드의 위치를 확보하는 것이다.

M. Cardei 등은 커버문제에 대하여 다음과 같이 분류하였다[21].

- 1) 문제의 대상 : 네트워크 수명의 최대화 혹은 분포되는 센서노드의 수를 최소화
- 2) 센서노드 분포방법 : 결정 혹은 랜덤
- 3) 센서노드의 감지반경과 통신반경과의 관계
- 4) 부가적인 요구사항: 에너지 효율성, 연결성
- 5) 알고리즘의 특성 : 중앙처리방식과 분산처리방식

커버문제는 이러한 분류와 함께 커버의 대상에 따라 다음과 같이 분류된다.

- 지역 커버: 센서 네트워크에 의하여 커버되는 영역(area)이 관심대상이다.[22 - 26].
- 지점(혹은 목표지점) 커버 : M. Cardei 등과 K. Kar 등 의하여 연구되었으며 커버의 목표는 특정 지점이다[27][28].
- path 커버 : X. Li 등과 S. Meguerdichian 등에 의하여 연구되었으며 관심 부분은 움직이는 노드에 의하여 발생하는 감지 path이다[29][30].

그림 2.1은 지역 커버 문제와 지점 커버 문제를 보여주고 있다. 그림에서 ○는 동작을 하지 않는 유효노드를, ●는 동작노드를 의미한다. 여기서 노드간의 통신거리는 감지반경의 2배 이상으로 하였다. 그림에서 원 모양의 실선은 센서노드에 의하여 감지되는 감지영역을 나타내고 있으며 동작노드사이를 연결하는 실선은 노드간 통신 통신 패스를 의미한다. 그림 2.1의 (a)에서, 네트워크의 감지 대상은 실선의 사각형의 영역이며 9개의 동작노드에 의하여 사각형 전역이 커버됨을 보여주고 있다.

그림 2.1의 (b)에서, 감지 대상은 □으로 나타낸 지점들이며, 동작노드 3개의 감지영역 내에 7개의 감지지점이 위치하고 있으므로 3개의 노드에 의하여 모

든 지점이 커버됨을 보이고 있다.

커버문제의 공통적인 목적은 특정 시점에 동작하는 센서노드들을 적절하게 스케줄링하여 네트워크의 수명을 최대화하는 것이다.

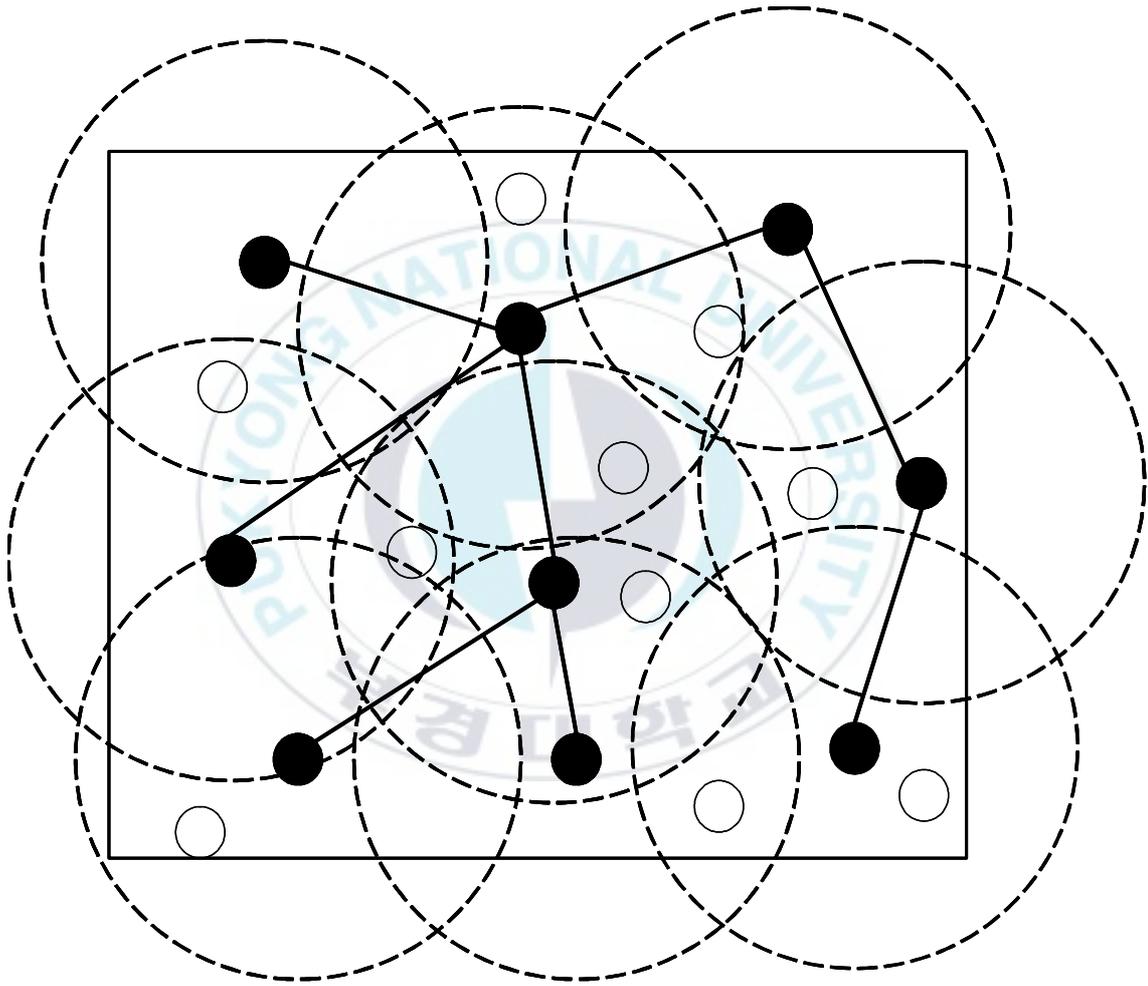


그림 2.1 (a) 지역 커버

Fig. 2.1 (a) area cover

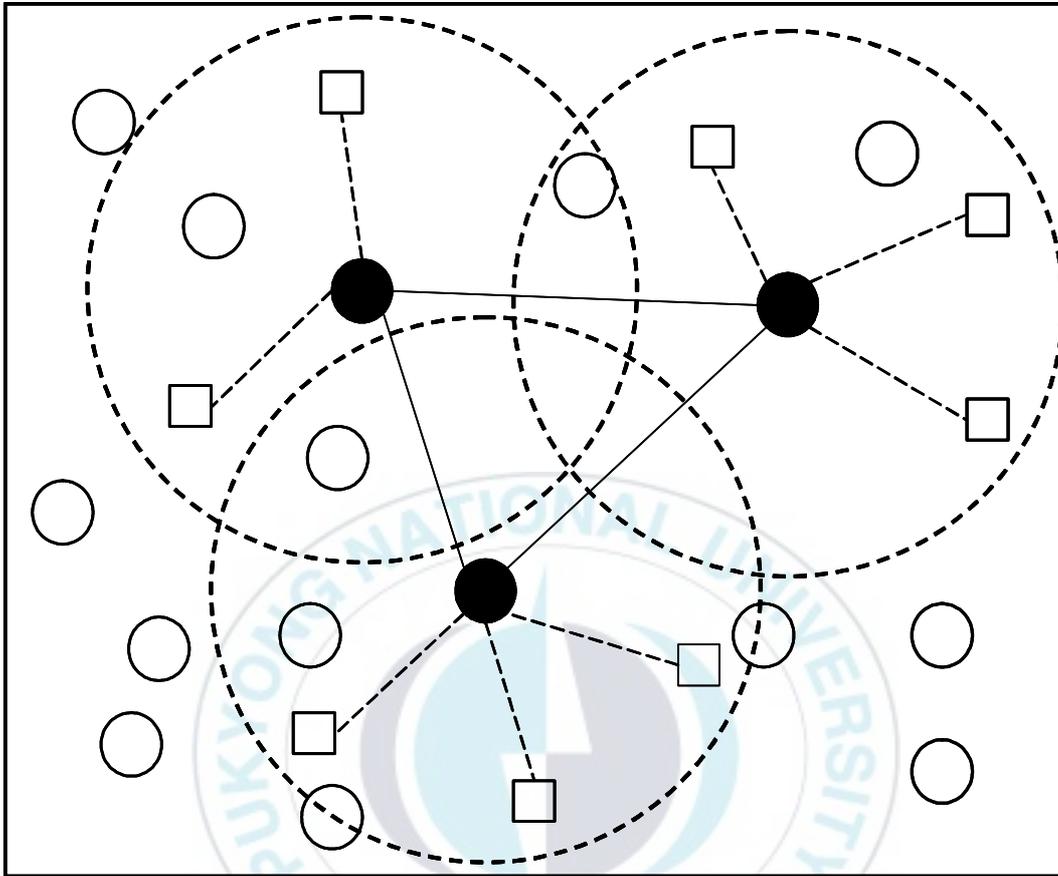


그림 2.1 (b) 지점 커버
 Fig. 2.1 (b) point coverage

2.2 센서 네트워크의 수명

그림 2.1에 나타난 바와 같이 센서 네트워크의 대상이 특정지점이든, 특정 영역이든 분포된 모든 센서노드가 동작될 필요는 없다. 각 목적에 따라 필요한 노드만 선택하여 동작시키고 나머지 노드는 이후에 필요한 시점(이전에 동

작하던 노드가 보유하고 있는 에너지를 모두 소모하여 기능을 할 수 없는 경우 등)에 사용할 수 있다. 이와 같이 노드 스케줄링은 분포된 센서노드 중에서 동작해야만 하는 노드의 집합을 선택하여 사용하고 나머지 노드는 유히노드로 두어 이후 필요한 시점에 사용하는 절차이다.

노드 스케줄링 알고리즘은 동작노드 선택 프로토콜(active node selection protocol)을 기반으로 국부적(local)이면서 분산처리 되는 것이 바람직하며, 이렇게 운영됨으로서 네트워크의 크기가 증가하거나 네트워크의 토폴로지(topology)가 동적일 경우에도 잘 적용될 수 있다. 그러나 이러한 방법이 갖는 단점도 있는데 이것에 대해서는 제 4장에서 기술한다. 동작노드 선택 문제는 특정영역의 모니터링, 연결성, 에너지 효율성과 같은 요구사항에 따라 다양한 방법으로 해결가능하다 [22-25].

2.3 목표지점커버와 지역커버 문제

M. Cardei 등은 특정지점들을 감지할 수 있는 모든 노드들의 집합을 disjoint set 조건으로 구하여 각각 집합에 대하여 서로 다른 시간동안 감지를 수행하게 함으로서 네트워크의 수명을 연장하는 방안을 제시하였다. 제안한 방법을 간략히 설명하면 다음과 같다. 특정지점들을 감지하는 하나의 노드집합이 구해지면 이 집합은 active 모드로 되어 감지기능을 수행하고, 나머지 집합들은 sleep 모드로 된다. active 모드로 선택된 집합의 모든 에너지가 소모되면 나머지 집합들에 대하여 순차적으로 sleep 모드에서 active 모드로 전환시켜 가면서 운영하면 네트워크의 수명이 연장된다. 이러한 방식의 핵심은 disjoint 집합의 수를 최대화하여 네트워크의 수명을 연장하는 것이다. 여기서 모든 목

표지점을 커버하는 노들의 집합을 disjoint 집합으로 구하는 문제를 disjoint set cover 문제로 정의하여 목표지점 커버문제를 다루었다. disjoint set cover 문제는 NP-complete 문제임을 증명하였으며 정수프로그램을 이용한 해를 소개하였다[27].

S. Slijepcevic 등은 주어진 감지대상영역을 필드의 집합으로 간주하여 특정 영역을 커버하는 노드의 집합을 disjoint 집합으로 구하여, 각 집합을 순차적으로 동작모드로 적용시킴으로써 네트워크의 수명을 연장하는 방안을 연구하였다[26].

M. Cardei 등은 모든 목표지점을 커버하는 노들의 집합을 non-disjoint 집합으로 구하여 이 집합들에 대하여 적절한 동작모드 수행시간을 할당함으로써 네트워크의 수명을 연장하는 방안을 제안하였다[12]. 여기서, 목표지점들을 커버하는 센서노드의 집합을 non-disjoint set으로 구하는 문제를 MSC(maximum set cover)문제로 정의하였으며 이 문제는 NP-complete 임을 증명하였다. 정의한 MSC 문제에 대하여 최적화 방법인 정수 프로그램(IP : integer programming), 선형 프로그램(LP : linear programming), 그리고 greedy 방식을 적용한 해결방안을 소개하였다.

제 3 장 목표지점 커버문제

3.1 목표지점 커버문제

센서 네트워크의 주 기능인 커버 문제에 있어서 특정 지점을 감지대상으로 하는 목표지점 커버문제(TCP : target cover problem)를 정의하기 위하여 다음과 같은 가정을 한다.

- 1) 지속적인 감지가 요구되는 임의의 목표지점(target)들과 이들 주위에 랜덤하게 분포된 다수의 센서노드로 구성된 센서 네트워크를 가정한다.
- 2) 센서 네트워크에는 하나의 BS(base station)가 있으며, 이 BS는 범용의 클러스터 헤드(cluster-head)의 기능과 함께 이웃하는 다른 센서노드로부터의 데이터 수집 기능을 수행한다.
- 3) 감지된 데이터는 센서노드 자체적 혹은 BS에 의하여 처리되어 사용자에게 전달된다.
- 4) 센서노드는 자신의 위치를 인지하는 기능을 갖는다.

일반적으로 센서노드는 주어진 다수의 target을 커버하기 위하여 요구되는 최소의 수 이상으로 분포되므로 하나의 target을 복수개의 센서노드에 의하여 감지되는 경우가 발생한다. 이때 복수개의 센서노드들 중에서 target을 커버하는 센서노드들 동작을 하면 되므로 나머지 노드들은 수면(sleep) 모드로 설정할 수 있다. 즉 잉여의 센서노드에 대하여 동작모드를 sleep/active로 전환하는 노드 스

케줄링을 적용하여 센서노드들의 에너지를 효율적으로 사용할 수 있다. 이러한 개념을 기반으로 목표지점 커버문제를 정의한다.

목표지점 커버문제 : m 개의 목표지점이 존재하며 각 목표지점에 대한 위치 정보는 알려져 있다. 목표지점의 주위에 n 개의 센서노드가 분포되어 있을 때, 모든 목표지점이 연속적으로 감지되면서 네트워크의 수명이 최대가 되도록 센서노드의 동작모드를 스케줄링 한다[12].

센서노드의 동작 스케줄링은 다음과 같은 절차로 진행된다.

- 단계 1) 센서노드들은 자신위치정보를 BS에 전송한다.
- 단계 2) BS는 센서노드들로부터 받은 정보를 이용하여 노드 스케줄링 알고리즘을 수행하고 그 결과에 따라 각 노드에 대하여 해당하는 동작정보를 전송한다.
- 단계 3) 모든 센서는 BS로부터 수신한 정보에 따라 동작모드를 설정한다.

이러한 절차에서는 BS와 센서노드 사이의 데이터의 전송방식과 동기 (synchronization)방법 등이 고려되어야 한다. 노드와 BS사이에 효율적인 데이터 전송에 사용될 수 있는 프로토콜로서는 LEACH[31], PEGASIS[32]의 방법이 사용될 수 있다. 노드의 동기는 BS가 주기적으로 노드에게 짧은 동기신호를 전송함으로써 구현가능하다.

3.2 최대커버집합 문제

MSC 문제를 정의함에 있어서 다음과 같은 사항을 가정한다. m 개의 목표지점 r_1, r_2, \dots, r_m 을 커버하기 위하여 n 개의 센서 s_1, s_2, \dots, s_n 가 임의의 위치에 분포되어 있다. TCP를 정의함에 있어서 BS가 센서노드와 목적지점들의 위치에 대한 정보를 갖는다고 가정하였으므로 이를 이용하여 각 센서노드가 감지할 수 있는 target 정보를 구할 수 있다. 한 가지 방법로서는, 센서와 target 간의 유클리드 거리를 구하여 센서노드의 감지반경보다 작으면 센서노드는 그 목표지점을 커버한다고 판단한다. MSC 문제의 예를 살펴보면 그림 3.1과 같다.

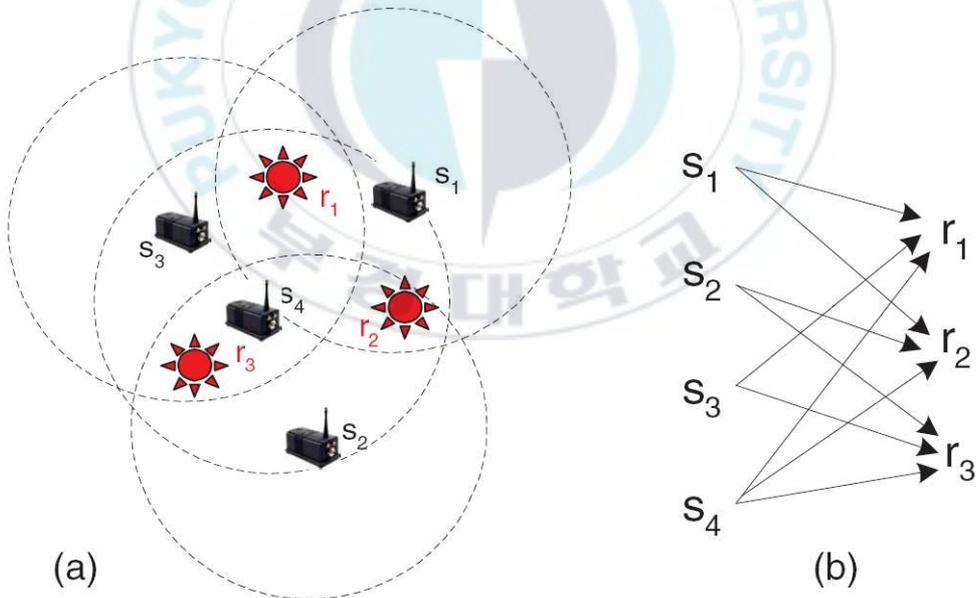


그림 3.1 3개의 목표지점과 4개의 센서로 구성된 MSC 문제

Fig. 3.1 MSC problem with three targets and four sensors

그림 3.1의 (a)에서는 3개의 목표지점을 나타내는 집합 $R=r_1, r_2, r_3$ 과 4개의 센서를 나타내는 집합 $C= s_1, s_2, s_3, s_4$ 가 적용된 센서 네트워크보이고 있다. 여기서 각 센서에 의해서 감지되는 target 지점의 판별은 센서노드를 중심으로 감지 반경을 갖는 원 내부에 존재 유무로서 가능하다. 센서에 의하여 감지되는 영역은 감지 길이를 반경으로 하는 정확한 원은 아니지만 문제를 일반화시키기 위하여 원의 모양을 갖는 것으로 한다.

센서노드와 target들 간의 관계를 각 센서노드에 의하여 커버되는 목표 지점의 집합으로 나타내면 그림 3.1의 (b)와 같이 각 센서노드에 대하여 $s_1=(r_1, r_2)$, $s_2=(r_2, r_3)$, $s_3=(r_1, r_3)$, $s_4=(r_1, r_2, r_3)$ 와 같은 나타낼 수 있다.

초기에 모든 센서노드는 동일하게 동작시간 1의 에너지를 갖는다고 가정한다.

네트워크 수명을 모델링함에 있어서 다음의 같은 두 가지 방법을 고려한다.

첫 번째, 모든 목표지점을 커버하는 센서노드의 집합을 disjoint 집합으로 구하여 각 노드집합에 센서노드의 동작시간을 적용하여 네트워크의 수명을 표현한다. 즉 센서노드들로 구성된 각 집합에서 하나의 센서노드가 반드시 하나의 집합에서만 존재해야 한다. 따라서 하나의 집합이 갖는 에너지 값은 하나의 센서노드가 갖는 에너지 값과 같다. 그림 2의 (a)에서, $R=(r_1, r_2, r_3)$ 을 커버하는 노드의 집합을 disjoint set으로 구하면 $S_1 = \{s_1, s_2\}$, $S_2 = \{s_3, s_4\}$ 가 되며, 이 경우, 구해진 집합에 1의 동작시간을 적용할 수 있으므로 전체 네트워크의 수명은 2가 된다.

두 번째, 모든 목표지점을 커버하는 노드의 집합을 non-disjoint set으로 표현하여 네트워크의 수명을 표현한다. 즉 모든 target을 커버하는 센서노드 집합에 대하여 하나의 센서노드가 복수개의 집합에 존재하여도 무방하다. 그림 2의

(a)에서, $R=(r_1, r_2, r_3)$ 을 커버하는 노드의 집합을 non-disjoint set으로 구하면, $S_1 = \{s_1, s_2\}$, $S_2 = \{s_2, s_3\}$, $S_3 = \{s_1, s_3\}$, $S_4 = \{s_4\}$ 이 된다. 각 센서노드의 수명이 1이고 첫 번째 경우와는 달리 하나의 센서노드는 여러 집합에서 원소로 사용되므로, 각각의 집합에서 사용되는 센서노드의 동작시간은 다음과 같이 결정된다. 센서노드 s_i 에 대한 동작시간은 식 (3.1)과 같이 표현가능하다.

$$t_{s_i} = \frac{1}{\text{모든 집합에서 } s_i \text{가 원소로 사용되는 횟수}} \quad (3.1)$$

식(3.1)에 따라 각 센서 s_1, s_2, s_3, s_4 에 적용되는 동작시간은 각각 $t_{s_1} = \frac{1}{2}$, $t_{s_2} = \frac{1}{2}$, $t_{s_3} = \frac{1}{2}$, $t_{s_4} = \frac{1}{1}$ 이 된다. 여기서 집합 S_i 에 적용되는 수명은 그 집합을 구성하는 센서노드에 할당된 동작시간 중에서 최소값이 된다. 즉 집합 S_i 에 적용되는 수명은 t_{S_i} 값은 식 (3.2)와 같이 정의된다.

$$t_{S_i} = \min(t_{s_i}), \forall s_i \in S_k \quad (3.2)$$

따라서 네트워크의 전체 수명은 T_{WSN} 식 (3.3)과 같다.

$$T_{WSN} = \sum_{p=1}^k t_{S_p} \quad (3.3)$$

여기서 k 는 non-disjoint 집합 수이다.

식 (3.3)를 이용하여 집합 S_1, S_2, S_3, S_4 에 각각 적용하면 아래와 같다.

$$\begin{aligned}
T_{WSN} &= t_{S_1} + t_{S_2} + t_{S_3} + t_{S_4} \\
&= \min(0.5, 0.5) + \min(0.5, 0.5) + \min(0.5, 0.5) + \min(1) \\
&= 0.5 + 0.5 + 0.5 + 1 \\
&= 2.5
\end{aligned}$$

disjoint set과 non-disjoint set 의 두 가지 방법을 비교하면 두 번째 방법이 첫 번째 방법에 비하여 25% 네트워크 수명이 향상됨을 보인다.

이상의 내용으로 MSC 문제를 정의하면 다음과 같다.

MSC 문제 : 센서의 집합 C 와 목표지점 집합 R 이 주어졌을 때 목표 지점 R 을 커버하는 센서들의 집합 S_1, \dots, S_p 와 이에 대응하는 $t_1, \dots, t_p (t_i \in [0,1])$ 을 구하되 $t_1 + \dots + t_p$ 가 최대값이 되도록 한다.

여기서 C 의 부분집합인 s 는 S_1, \dots, S_p 에 존재하고 전체 집합에 대하여 할당되는 값의 합은 1이하를 갖는다. 즉 하나의 센서노드 수명은 1이다.

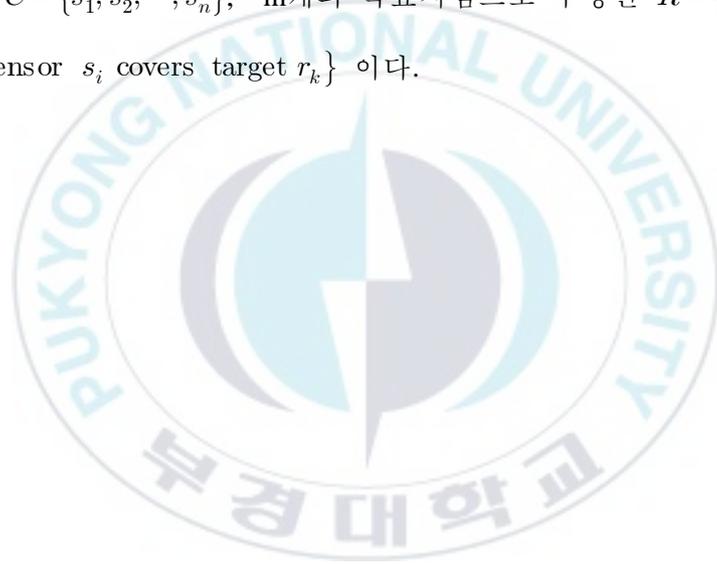
3.3 MSC 문제를 해결하는 기존의 방법들

M. Cardei 등은 MSC 문제를 해결하는 방법으로 정수프로그래밍, 경험적 방법 기반의 선형 프로그래밍과 경험적 방법의 greedy 등 3가지 최적화 기반의 알고리즘을 제안하였다.

정수 프로그래밍 방식의 적용을 위한 MSC 문제의 정형화된 식은 (3.4)과 같다.

$$\begin{aligned}
& \text{Maximize} && t_1 + \cdots + t_p \\
& \text{subject to} && \sum_{j=1}^p x_{ij} t_j \leq 1 \quad \forall s_i \in C \\
& && \sum_{i \in C_k} x_{ij} \geq 1 \quad \forall r_k \in R, j = 1, \dots, p \\
& \text{where} && x_{ij} = 0, 1 \quad (x_{ij} = 1 \text{ if and only if } s_i \in S_j) \quad (3.4)
\end{aligned}$$

여기서 p 는 목표지점 모두를 커버하는 노드들의 집합, n 개의 센서로 구성된 센서집합 $C = \{s_1, s_2, \dots, s_n\}$, m 개의 목표지점으로 구성된 $R = \{r_1, r_2, \dots, r_m\}$, $C_k = \{i \mid \text{sensor } s_i \text{ covers target } r_k\}$ 이다.



제 4장 센서노드 스케줄링 메카니즘의 고찰

4.1 센서노드 스케줄링 프로세스

센서 네트워크의 에너지 효율적인 운영을 위하여 네트워크를 구성하는 센서노드의 동작을 스케줄링한다. 진술한 바와 같이 센서노드 스케줄링이란 충분히 분포된 센서노드에 대하여 감지에 필요한 노드를 **active**, 불필요한 노드는 **sleep** 상태로 설정하는 메카니즘이다.

일반적으로 스케줄링 알고리즘은 하나의 센서노드에서 수행되는 중앙처리방식과 각 노드에서 국부적으로 처리되는 분산처리방식으로 구분된다. 중앙처리방식은 모든 센서노드의 위치, 보유에너지 그리고 주어진 **target** 지점의 정보를 하나의 센서노드(일반적으로 **BS**)가 종합하여 각 센서노드의 동작모드를 결정한다. 분산처리방식은, 임의의 센서노드에서 통신 가능한 이웃하는 노드와의 정보교환을 통하여 주변의 감지영역내에 있는 **target**을 효율적으로 감지할 수 있도록 자신과 이웃노드의 동작모드를 국부적으로 선택한다. 동작모드가 결정된 센서노드는 아직 결정되지 않은 이웃하는 노드에게 자신의 정보를 전달하여 동작상태를 결정할 수 있도록 한다. 이러한 과정이 순차적으로 네트워크 전체에 전달되면 전체 센서노드의 동작모드가 결정된다.

그림 4.1은 센서 네트워크에서 스케줄링 시간과 노드의 동작을 나타낸 것이다. 그림에서 나타난 바와 같이 네트워크 전체 수명시간은 여러개의 라운드로 구성되며 하나의 라운드는 노드 스케줄링 알고리즘 수행시간과 스케줄링에 의거한 센서노드의 동작시간(**active time**)으로 구성된다. 센서 네트워크에서는 그

림과 같이 스케줄링시간과 동작시간으로 나누어져 있으므로 스케줄링하는 동안은 감지를 할 수 없는 상태가 된다. 이러한 모드전환지점이 많으면 지속적인 감지를 할 수 없게 되어 감지기능의 QoS에 부정적인 요인으로 작용하게 된다.

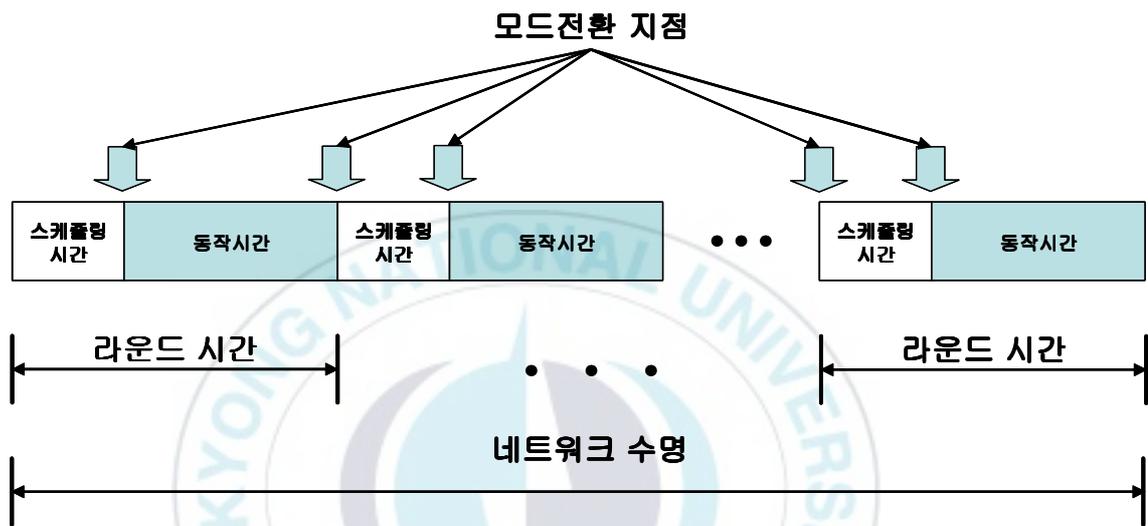


그림 4.1 센서노드 스케줄링 메카니즘

Fig. 4.1 scheduling mechanism of sensor networks

4.2 기존의 스케줄링 알고리즘 고찰

분산처리방식의 스케줄링 알고리즘은 그림 4.1과 같이 스케줄링 시간에 특정 노드로부터 전체 노드에 이르기까지 센서노드의 동작모드를 설정하는 과정을 거친다. 전체 센서 네트워크 운영과정에 라운드 개념이 적용되기 때문에 모드전환지점은 반드시 발생하게 된다. 그러나 중앙처리방식의 알고리즘에서 하나의 센서노드에서 일괄적으로 스케줄링이 수립되는 특성으로 인하여, 알고리즘을 적용함에 있어서 센서노드의 정확한 에너지소비 모델을 반영하면 단

한번의 스케줄링 알고리즘의 수행으로 전체 센서노드의 동작 스케줄이 가능하다. 따라서 중앙처리 방식의 스케줄링 알고리즘은 그림 4.1과는 다른 형태의 모드전환이 발생한다.

그림 4.2는 중앙처리방식의 알고리즘을 적용하였을 때 스케줄링 메카니즘을 나타낸 것이다. 여기서의 라운드 개념은 결정된 동작 스케줄에 맞추어 전체 센서노드가 일시에 적용되도록 하기 위한 동기 신호 (synchronization signal) 발생을 의미한다. 이러한 동기 신호는 전체 센서 노드에게 전달되어야 하므로 모든 센서 노드에 공통적으로 적용되어야 하기 때문에 전체 네트워크 관점에서는 대규모의 에너지 소모를 초래하는 요인이 된다. 일반적으로 센서 노드에서 전체 에너지소모에 대하여 통신에너지 소모가 많은 비중을 차지하고 있으므로 가능한 통신에너지 소모는 줄여야 한다.

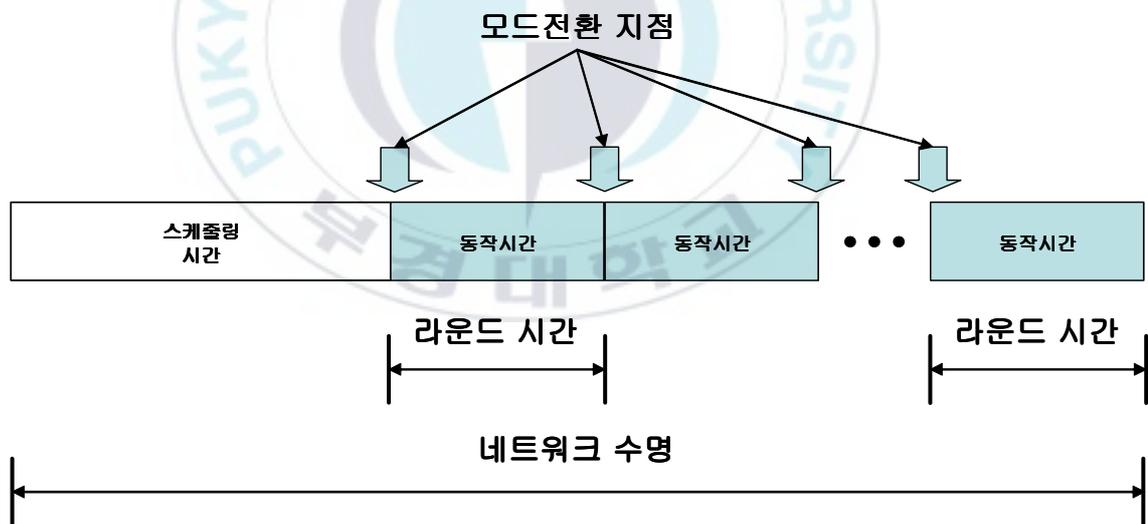


그림 4.2 센서노드 스케줄링 메카니즘(중앙처리방식)

Fig. 4.2 scheduling mechanism of sensor nodes(centralized)

M. Cardei 등이 [12]에서 제안한 greedy-heuristic 알고리즘은 이와 같이 모든

센서노드의 소비에너지가 일정하다는 가정에서 제안한 중앙처리 방식의 노드 스케줄링 알고리즘이다. 그림 4.3은 greedy-heuristic 알고리즘이다.

```

Greedy-MS C Heuristic ( $C, R, w$ )
1: set lifetime of each sensor to 1
2: SENSORS = C
3: i=0
4: while each target is covered by at least one sensor in SENSORS do
5: /*. a new set cover  $C_i$  will be formed */
6: i = i + 1
7:  $C_i = \emptyset$ .
8: TARGETS = R
9:   while TARGETS =  $\emptyset$ . do
10:    /*. more targets have to be covered */
11:    find a critical target  $r_{critical} \in$  TARGETS
12:    select a sensor  $s_u \in$  SENSORS with greatest contribution,
        that covers  $r_{critical}$ 
13:     $C_i = C_i \cup s_u$ 
14:    for all targets  $r_k \in$  TARGETS do
15:      if  $r_k$  is covered by  $s_u$  then
16:        TARGETS = TARGETS -  $r_k$ 
17:      end if
18:    end for
19:  end while
20:  for all sensors  $s_j \in C_i$  do
21:    lifetime  $s_j =$  lifetime  $s_j - w$ 
22:    if lifetime  $s_j == 0$  then
23:      SENSORS = SENSORS -  $s_j$ 
24:    end if
25:  end for
26: end while
27: return i-number of set covers and the set covers  $C_1, C_2, \dots, C_i$ 

```

그림 4.3 MSC 문제에 대한 greedy-heuristic 알고리즘

Fig. 4.3 greedy-heuristic algorithm for MSC problem

알고리즘의 입력값은 target 집합 R과 하나의 라운드 시간에 센서노드의 동작시간 또는 사용되는 에너지인 w 이다. 알고리즘의 출력 값은 각 라운드에서 동작해야만 하는 노드의 집합인 C_i 이다. 그림 4.3에서 스케줄링 결과값인 C_i 는 라인 4 - 26까지의 while do 를 통하여 구하여진다. 또한 라인 21에서는, 선택된 센서노드가 w 값의 에너지를 소비함을 반영하고 있다. 라인 26까지 한번의 while do 를 수행하면 1 라운드에 해당하는 노드 집합이 구해진다. 그림 4.4는 MSC 문제에 대한 greedy-heuristic 알고리즘을 적용한 네트워크 스케줄링 메카니즘이다.

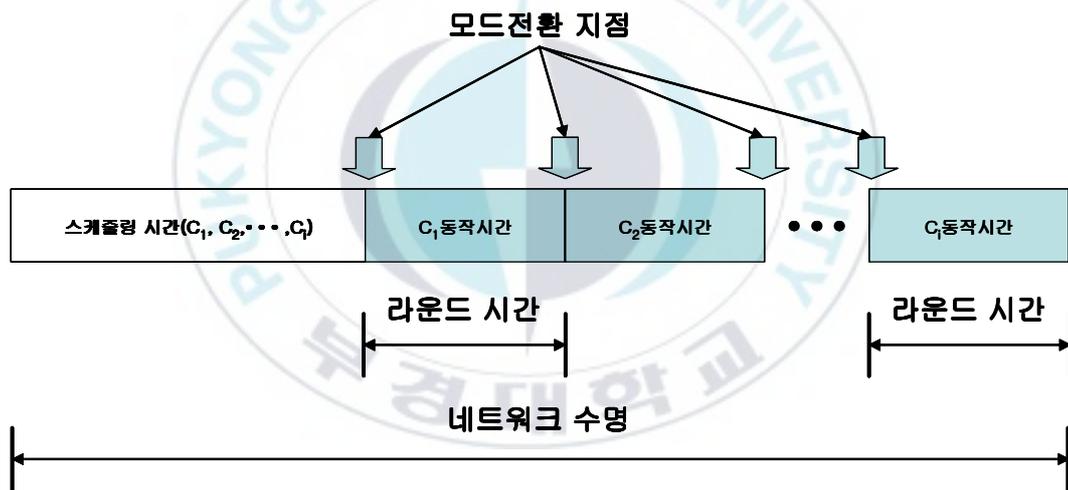


그림 4.4 센서노드 스케줄링 메카니즘 (greedy-heuristic)

Fig. 4.4 scheduling mechanism of sensor nodes (greedy-heuristic)

그림에서와 같이 스케줄링알고리즘에 의하여 구해지는 일련의 노드집합에서 현재 구한 노드 집합 C_j 가 이전에 구한 노드집합 C_{j-1} 과 동일하다면 C_j 와 C_{j-1} 를 하나의 라운드에 적용하는 것이 가능하다. 따라서 동일한 노드 집합이

연속적으로 나타나면 동일한 모든 집합을 하나의 라운드로 적용하고, 서로 다른 집합에 대해서는 각각을 구분하여 적용함으로써 모드 전환지점의 수를 최소화할 수 있다.

중앙처리방식의 스케줄링 알고리즘은 분산처리 방식과는 달리 노드의 스케줄링이 일괄적으로 처리됨에 따라 센서 네트워크의 지속적인 감지기능면에서 상대적으로 우수한 면을 보이고 있다. 반면 모드전환횟수를 최소화하여 전체 노드의 동기화에 필요한 통신 에너지의 소모를 최소화하는 것이 필요하다.



제 5 장 유전자 알고리즘을 이용한 *MSC* 문제의 해

MSC 문제는 *NP-complete* 이며 기존의 방법들은 최적화기법을 이용한 해결 방안을 제안하였다. 이러한 최적화 알고리즘 기반의 방법들은 알고리즘을 통하여 스케줄링을 적용하는 과정에서 라운드 횟수 즉, 네트워크의 모드 전환 횟수가 많아지는 단점이 있다. 응용분야에 따라 센서 네트워크의 최대의 동작 시간을 확보하는 노드 스케줄링도 중요하지만 지속적인 감지를 유지하도록 하는 스케줄링도 필요하다.

본 논문에서는 *NP-complete* 문제에 유전자 알고리즘을 적용하였다. *MSC* 문제에 유전자 알고리즘을 적용하여 최적의 값에 수렴하는 경향을 분석에 가능한 적은 수행시간 내에 유효한 해를 구하는 방안을 도출하고자 한다.

유전자 알고리즘에 관하여 배경 기술에 대한 소개와 함께 주어진 *MSC* 문제에 대하여, 일반적인 방식인 중앙처리방식의 유전자 알고리즘 설계에 관하여 설명하고, 다음으로 센서 네트워크의 구조적 특성을 이용한 병렬화 기법을 적용한 병렬 유전자 알고리즘의 설계에 관하여 기술한다.

5.1 유전자 알고리즘

5.1.1 유전자 알고리즘의 역사

1950년대 후반 J. Holland[14]는 진화 생물학자인 Fisher의 저서 "*The Genetic Theory of Natural Selection*"를 접하면서 물리학 도구를 이용하는 시도를 하였다. 그 결과, 그는 학습과 같이 진화도 환경에 적응해 가는 한 형태이지만 단일 세대 기간에서만 동작하기보다는 세대를 통해 동작한다는 사실을 알게 되었다. 그는 진화가 유기체에서 동작한다면 컴퓨터 프로그램 상에서 구현하는 것도 가능할 것이라는 가정 하에 유전자도구를 적응 문제에 적용하는 연구를 수행하게 되었다.

이 시기에 J. Holland가 수행한 연구는 적응 시스템과 함께 환경을 다루게 되었다. 그의 목적은 임의 환경에 적응하는 무제한적인 능력을 가진 프로그램을 만드는 데 필요한 이론과 절차를 개발하는 것이었다. 그는 최우량인자의 인위적인 적자생존과 같은 비자연적인 선택의 근본적인 역할을 인식하게 되었고, 탐색을 위해서는 단일구조 대 단일구조 접근법보다는 집단 접근법을 사용하게 되었다. 그리고 1965년에는 교차, 돌연변이, 재조합 등의 유전연산자들의 중요성에 대해 인식하게 되었다. 이러한 연구결과를 토대로 J. Holland는 유전자 알고리즘의 표준모델을 제안하였다.

5.1.2 유전자 알고리즘의 기본구조

유전자 알고리즘에서 해의 후보는 유전자형(genotype) 혹은 염기(gene)로서 염색체에 일차원적으로 표현된다. 그리고 각 세대는 개별 염색체를 나타내는 개체(individual)들의 집합인 개체군(population)으로 표현된다. 유전자 알고리즘은 유전연산자들을 개체군에 속해 있는 염색체들에 반복 적용함으로써 세대를 진행해 감에 따라 더 나은 해를 구하게 된다. 이 때 사용되는 유전연산자는 선택연산, 교차연산, 돌연변이연산이 있다. 선택연산은 도태의 압력을 가하는

경우에 중요한 것으로 선택연산을 통해 개체군에서 교차연산의 대상이 될 염색체들을 선택한다. 교차연산은 선택된 두 부모의 염색체를 조합하여 유전정보를 교환함으로써 새로운 염색체를 만들고, 돌연변이연산은 유전자를 일정한 확률로 변화시킴으로써 지역 해에서의 탈출과 함께 더 넓은 해 공간을 탐색을 가능하게 한다.

일반적인 유전자 알고리즘의 동작은 순서는 그림 5.1과 같다.

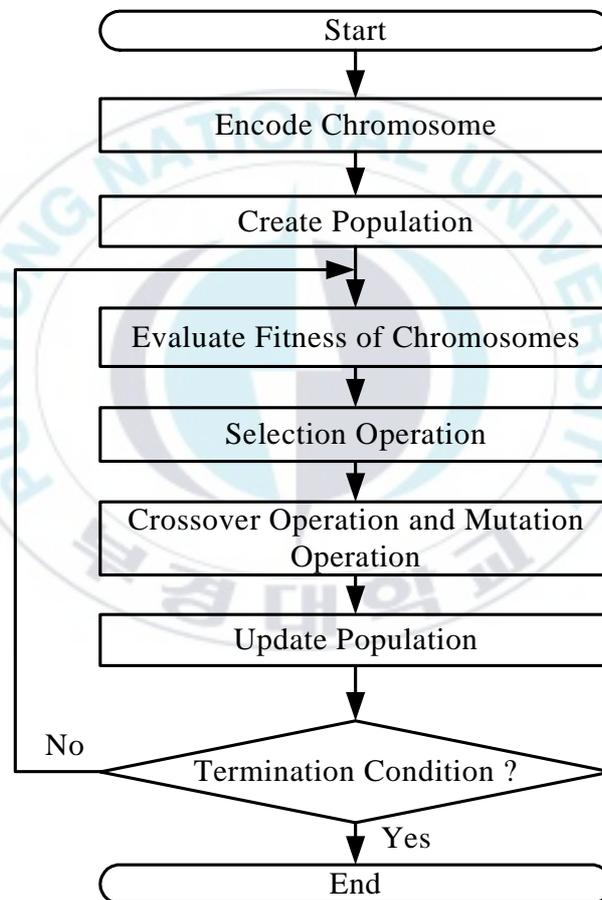


그림 5.1 유전자 알고리즘의 동작

Fig. 5.1. The operation of genetic algorithm

유전자 알고리즘의 동작은 먼저 탐색대상이 되는 해집합에서 해의 후보들을 선택하여 염색체를 인코딩하고, 이 염색체들을 이용하여 초기 모집단(initial population)을 생성한다. 평가함수(evaluation function)를 통해 개체군 내에 있는 각 염색체들의 적합도를 평가하고 평가된 적합도를 선택연산에서 활용하여 교차연산과 돌연변이연산을 수행할 염색체를 선택한다. 이 때, 선택압력을 조절함으로써 도태확률을 변화시킬 수 있다. 이를 통해 최적해에 대한 수렴속도를 조절할 수 있다. 선택연산을 통해 선택된 부모염색체들에 교차연산과 돌연변이연산을 적용하여 자식염색체를 생성한다. 부모염색체와 자식염색체 중 적합도가 높은 염색체로 다음 세대의 개체군을 재생성하고 종료조건에 이를 때까지 선택연산, 교차연산, 돌연변이연산의 적용과 새로운 개체군의 재생성을 반복 수행한다. 알고리즘의 종료조건은 일반적으로 세대수로 결정된다.

5.1.3 스키마 정리

유전자 알고리즘에서는 개체군에 유전연산자들을 반복 적용함으로써 최적해를 탐색하고 탐색과정이 진행되는 동안 일정한 패턴들이 유지되어 최적해를 찾게 된다. 스키마 정리는 유전자 알고리즘의 각 세대가 진행됨에 따라 어떤 패턴들이 살아남을지를 보여주는 정리이다[33].

스키마의 최초고정부분과 최후고정부분 사이의 거리를 나타내는 정의길이를 $\delta(H)$, 스키마 중에서 값이 결정되는 부분의 수를 나타내는 차수를 $o(H)$, 세대 t 에서 개체군에 존재하는 스키마 H 의 개수를 $m(H, t)$, 스키마 H 를 포함하고 있는 개체의 평균적응도를 $f(H)$, 개체군에 포함되어 있는 개체들의 평균적응도를 \bar{f} 라고 할 때, 세대 $t + 1$ 에 존재하는 스키마 H 의 개수에 대한 기대치 $m(H, t + 1)$ 은 식 (5.1)과 같다[14,15].

$$m(H, t+1) = m(H, t) \frac{f(H)}{f} \quad (5.1)$$

이 때, 스키마는 교차연산이나 돌연변이연산에 의해서 파괴될 가능성이 있으므로 이 영향을 고려하여야 한다. 교차확률이 P_c , 각 염색체의 길이가 l 일 때 어떤 스키마 H 가 교차연산에 의해서 파괴될 확률 P_{HC} 는 식 (5.2)와 같다 [14].

$$P_{HC} = P_c \frac{\delta(H)}{l-1} \quad (5.2)$$

돌연변이확률이 P_m 일 때 어떤 스키마 H 가 돌연변이연산에 의해서 파괴될 확률 P_{HM} 은 식 (5.3)과 같다[14].

$$P_{HM} = o(H)P_m \quad (5.3)$$

식 (5.1)에 식 (5.2)의 P_{HC} 와 식 (5.3)의 P_{HM} 을 적용하여 $m(H, t+1)$ 을 구하면 식 (5.4)와 같이 된다[14].

$$m(H, t+1) \geq m(H, t) \times \frac{f(H)}{f} \left[1 - P_c \frac{\delta(H)}{l-1} - o(H)P_m \right] \quad (5.4)$$

여기서 $\delta(H) / (l-1)$ 는 염색체에서 교차점이 놓일 수 있는 모든 경우에 대한 스키마 길이의 비율, 즉 스키마가 파손될 확률을 나타낸다. 그러나 교차점이

스키마 내부에 놓이는 경우라 하더라도 상대 부모해가 똑같은 특정 기호를 갖고 있다면 그 스키마는 파괴되지 않는다. 즉, 스키마 정리는 스키마의 생존 가능성에 대한 하한선을 제시한다[34].

식 (5.4)의 스키마 정리는 변이율과 교차율을 줄이면 적용된 스키마가 집단 중에 증가한다는 것을 나타내고 가능한 한 변이율이나 교차율을 줄이는 것이 좋으나 변이율과 교차율이 0이 되면 진화는 일어나지 않음을 나타낸다.

5.1.4 병렬 유전자 알고리즘

유전자 알고리즘은 구현과 동작과정이 간단하고 유효시간 내에 근사 최적해를 얻을 수 있기 때문에 함수의 최적화, 조합문제, 제어문제, 기계학습 등의 다양한 분야에 적용된다. 근래에 유전자 알고리즘은 더 크고 복잡한 문제들에 대한 적용이 시도되고 있으며 이를 위하여 계산의 고속화 및 보다 나은 근사해를 구하는 방법들이 요구된다. 고속으로 근사해를 얻는 가장 자연적인 방법은 유전자 알고리즘을 병렬화하는 것이며 이를 병렬 유전자 알고리즘이라 한다[35,36].

5.1.4.1 병렬 유전자 알고리즘의 개요

병렬유전자 알고리즘에서는 각 처리기가 하위집단 혹은 부개체군(subpopulation)을 가짐으로써 지역해가 전체 개체군으로 확산되는 것을 피하고 보다 좋은 해를 얻고자 하는 접근방법과 병렬처리를 통해 탐색의 고속화를 꾀하는 접근방법이 있다. 그러나 어느 쪽이든 한 가지 방법만을 고려하는 경우는 거의 없고 실제로는 해의 질을 향상시키고 탐색도 고속으로 수행하고자 하고 있다[35 - 38].

5.1.4.2 병렬 유전자 알고리즘의 분류

병렬 유전자 알고리즘은 구현되는 구조, 부 개체군의 유무 및 크기에 따라 그림 5.2와 같이 기본 모델들, 소규모 병렬 유전자 알고리즘, 추가 모델들로 분류될 수 있다[35,36].

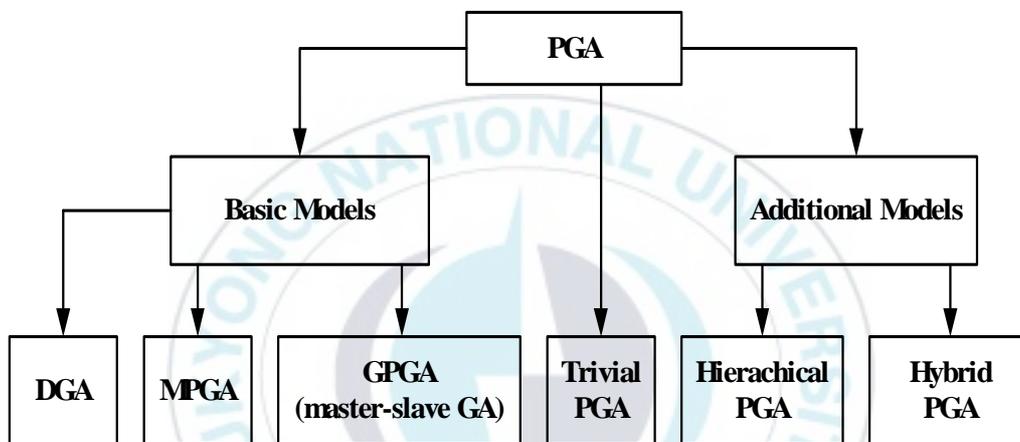


그림 5.2 병렬 유전자 알고리즘의 분류

Fig. 5.2 The classification of parallel genetic algorithms

기본 모델들은 전역 병렬 유전자 알고리즘(GPGA: global parallel genetic algorithms), 대규모 병렬 유전자 알고리즘(MPGA: massively parallel genetic algorithms), 분산 유전자 알고리즘(DGA: distributed genetic algorithms)으로 세부적으로 분류될 수 있고, 기본 모델들을 조합하거나 다른 최적화 방법을 조합한 추가 모델들은 계층적 병렬 유전자 알고리즘(HPGA: hierarchical parallel genetic algorithms)과 하이브리드 병렬 유전자 알고리즘(hybrid parallel genetic algorithms)으로 분류될 수 있다.

전역 병렬 유전자 알고리즘은 마스터-슬레이브 유전자 알고리즘이라고도 불리며 하나의 개체군으로 구성된다. 그림 5.3은 전역 병렬 유전자 알고리즘 알고리즘의 개념을 나타낸다.

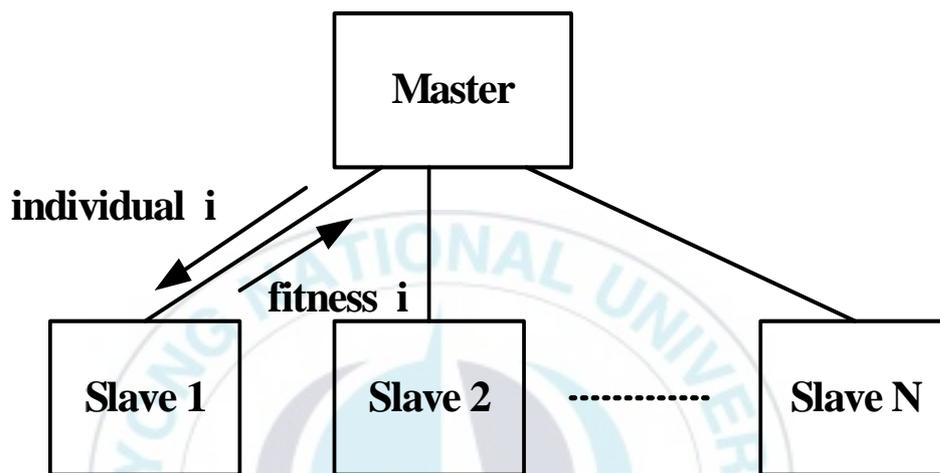


그림 5.3 전역병렬 유전자 알고리즘의 개념도

Fig. 5.3 A schematic view of global parallel genetic algorithm

이 병렬 유전자 알고리즘에서는 개별 염색체의 평가가 프로세서들에 분산되기 때문에 계산 속도가 빠르다. 이 때 적합도의 평가는 각 프로세서들에 분산되고, 선택연산과 교배연산은 전역적으로 이루어진다. 또한, 구현이 용이하고 성능향상을 기대하기 쉽다[39,40].

대규모 병렬 유전자 알고리즘은 세립적(fine-grained) 병렬 유전자 알고리즘이라고도 불리며 대규모 병렬 컴퓨터에 적합하다. 이 병렬 유전자 알고리즘은 하나 혹은 소수의 처리기를 이용하여 각 처리기가 대규모의 개체군을 가지고 유전자 알고리즘을 수행한다. 이 때 각 프로세서에서 개선된 염색체들은 인접

한 처리기들과만 교환된다[41 - 45].

분산 유전자 알고리즘은 다중 개체군 유전자 알고리즘(multi-deme or multi-population genetic algorithm), 조립적(coarse-gained) 병렬 유전자 알고리즘 혹은 섬 모델(island model)이라고도 불린다.

병렬 유전자 알고리즘의 구조는 그림 5.4와 같다. 병렬 유전자 알고리즘은 여러 개체군을 이용하여 연산을 처리하고 필요한 경우 염색체 이주 과정을 통해 염색체를 교환한다. 이는 미성숙 조기 수렴의 문제점을 해결하고 개체의 다양성을 유지하기 위해서 고안된 이론으로서 그 성능의 우수함을 인정받고 있다[46 - 51]. 최근에 가장 많이 이용되는 네트워크기반 병렬 유전자 알고리즘도 병렬 유전자 알고리즘의 일종이다.

소규모 병렬 유전자 알고리즘은 프로세서들이 독립된 문제들을 개별적으로 처리하는 기법이다. 이 병렬 유전자 알고리즘은 프로세서간의 통신이 존재하지 않으며 개별 처리가 가능한 통계처리 등을 위해 주로 사용된다[35,38].

계층적 병렬 유전자 알고리즘은 여러 개의 병렬화 기법들을 함께 사용하면서 각 병렬화 기법들 사이에 계층구조를 적용하여 상위계층은 속도를 향상시키고 하위계층은 더 넓은 해 공간을 탐색하는 방식으로 주로 이용된다[35,36].

하이브리드 병렬 유전자 알고리즘은 병렬화 기법들을 개별적으로 사용하는 것보다 나은 성능을 얻기 위해서 둘 이상의 병렬화 기법을 결합시켜서 사용한다[36].

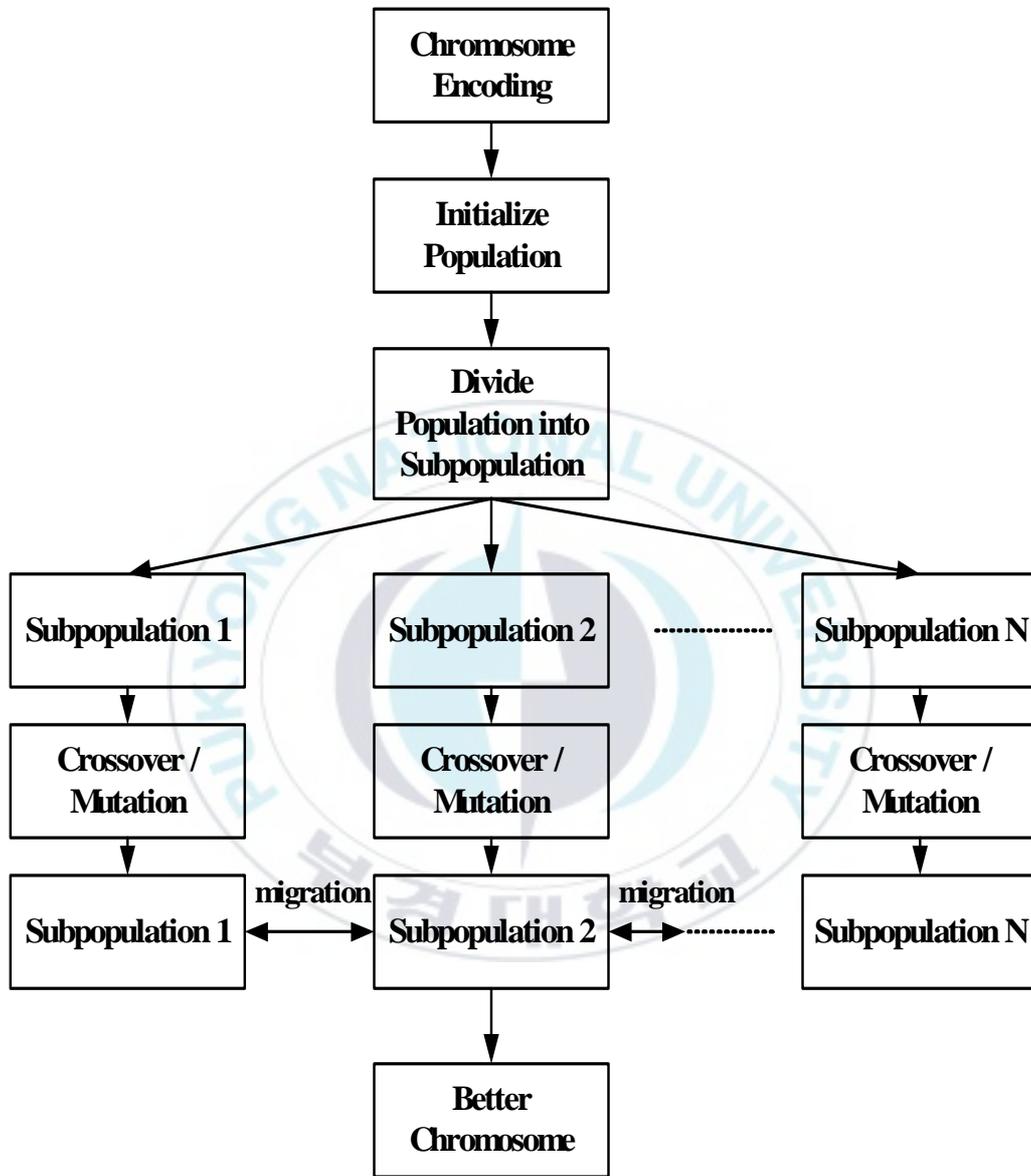


그림 5.4 분산 유전자 알고리즘의 구조

Fig. 5.4 The architecture of distributed genetic algorithm

5.2 중앙처리방식의 유전자 알고리즘

유전자 알고리즘이 하나의 BS에서 수행되는 단일모델의 중앙처리방식의 모델에 관하여 설명한다. 유전자 알고리즘을 이용하여 MSC 문제를 해결하기 위하여 문제에 적합한 염색체 표현방법과 함께 새로운 유전연산자들을 설계한다.

주어진 센서노드와 목표지점들에 대한 정보를 알고 있을 때, 염기는 모든 목적지점을 커버하는 하나의 노드들의 집합을 의미하며, 염색체는 MSC 문제의 잠재 해로서 염기들의 집합을 의미한다.

5.2.1 염색체의 표현

유전자 알고리즘에서 문제의 잠재 해를 표현하기 위한 기본적인 개체를 염색체라 하고 각 개체를 이루는 요소를 염기라 한다. 유전자 알고리즘에서 모든 해는 염색체로 표현될 수 있어야 하므로, 가장 보편적으로 사용하는 염색체의 표현(또는 인코딩)방식은 이진비트열을 이용한 방법이다. 이진비트열 방식은 가능한 모든 해를 표현할 수 있어야 하므로 주어진 문제에 대한 모든 해의 구성요소가 결정된 상태에서 적용가능하다.

MSC 문제에서, 해를 구성하는 염기는 모든 target을 커버하는 노드의 집합이므로 표현되고, 모든 센서노드들로 구성된 집합의 멱집합 중 모든 target 커버를 하지 못하는 노드의 집합을 제외한 집합으로 표현된다.

염색체 인코딩방식을 이진비트열로 표현하기 위해서는, MSC 문제에서 해에 해당하는 염기 즉, 모든 target을 커버하는 노드의 집합을 우선하여 구하여야 하지만 대용량의 노드 및 target이 적용될 경우에는 target을 커버하는 센서노드의 집합을 구하는 자체만으로 많은 계산량을 요구하므로 이진 비트열을 이용한 인코딩방식은 적합하지 않다.

본 연구에서 제안한 방법에서는, 염색체를 부호화하기 위해서 미리 모든 **target** 커버를 수행하는 센서들의 집합들을 구하지 않고, 센서들을 하나씩 임의로 선택하여 모든 **target**이 커버될 때 까지 반복 수행함으로써 하나의 염기를 구성하게 하고, 이러한 염기들로 구성된 염색체를 문제의 해로 표현한다. 염색체의 인코딩 과정을 그림 5.5에 나타내었다.

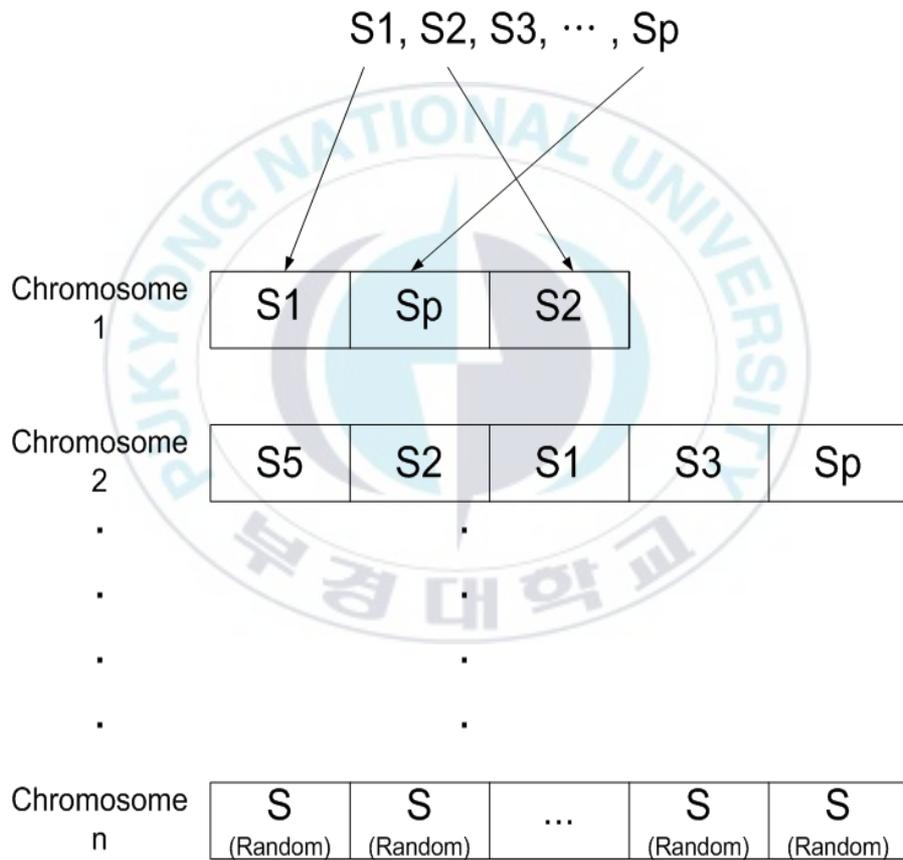


그림 5.5 염색체의 부호화

Fig. 5.5 The Chromosome Encoding

그림에서 $S_1, S_2, S_3, \dots, S_p$ 는 모든 **target**을 커버하는 센서노드 집합인 염기에 해당된다. 여기서 S_i 는 앞서 언급한바와 같이 미리 구하지 않으며, 임의의 센서노드를 선택하여 **target** 커버여부를 결정하게 되므로, S_i 집합의 크기, 구성요소 그리고 집합의 수는 다르게 나타날 수 있다. 실제 $S_1, S_2, S_3, \dots, S_p$ 는 염색체 인코딩 과정에서 만들어지며 이러한 절차에 따라 각 염기는 다음 그림 5.5와 같이 임의의 염색체에 추가되어 새로운 염색체를 생성하는 결과가 된다. 만약 중복되는 센서집합이 염기로 선택된 경우에는 축약을 통해 배제한다. 하나의 센서가 모든 **target**을 모두 커버하는 경우에는 그 센서만을 요소로 갖는 하나의 집합이 된다. 하나의 염색체 인코딩과정을 무한정 반복할 수는 없으므로 임의의 시행횟수를 두어 그 동안에 구성된 집합들을 하나의 해로 간주하여 염색체로 설정한다. 제안한 염색체의 부호화 방법을 그림 3.1의 예를 적용하면 그림 5.6과 같다.

그림 5.6에서, 센서 s1, s2로 구성된 센서집합 S1, 센서 s2, s3로 구성된 센서집합 S2, 센서 s3, s1로 구성된 센서집합 S3, 센서 s4로만 구성된 센서집합 S4가 염색체의 각 염기를 구성할 수 있는 예이다.

본 논문에서는 최적의 해를 구하는 것이 목적이 아니라 임의의 시간 이내에 유효한 값을 도출하는 것을 목적으로 하고 있다. 따라서 S1, S2, S3, S4가 의미하는 모든 **target**을 커버하는 모든 노드의 집합을 먼저 구하지 않는다. 즉, 분포된 모든 노드에 대하여 임의로 노드를 선택하되 전체 **target**을 커버할 때까지 노드를 중복되지 않도록 선택하여 하나의 염기를 완성한다. 이러한 과정을 이용함으로써 모든 **target**을 커버하는 집합을 구하는 계산은 하지 않게 된다. 그림 5.6에서 염색체의 길이는 일정하지 않다. 그 이유는 주어진 문제에서 노드의 수와 **target**의 수가 정해져 있지 않으며, 또한 주어진 **target**을 모두 커버하는 노드의 집합들은 경우에 따라 다르므로 해의 조합을 특정 길이로 한정시킬 수 없

기 때문이다.

Example :

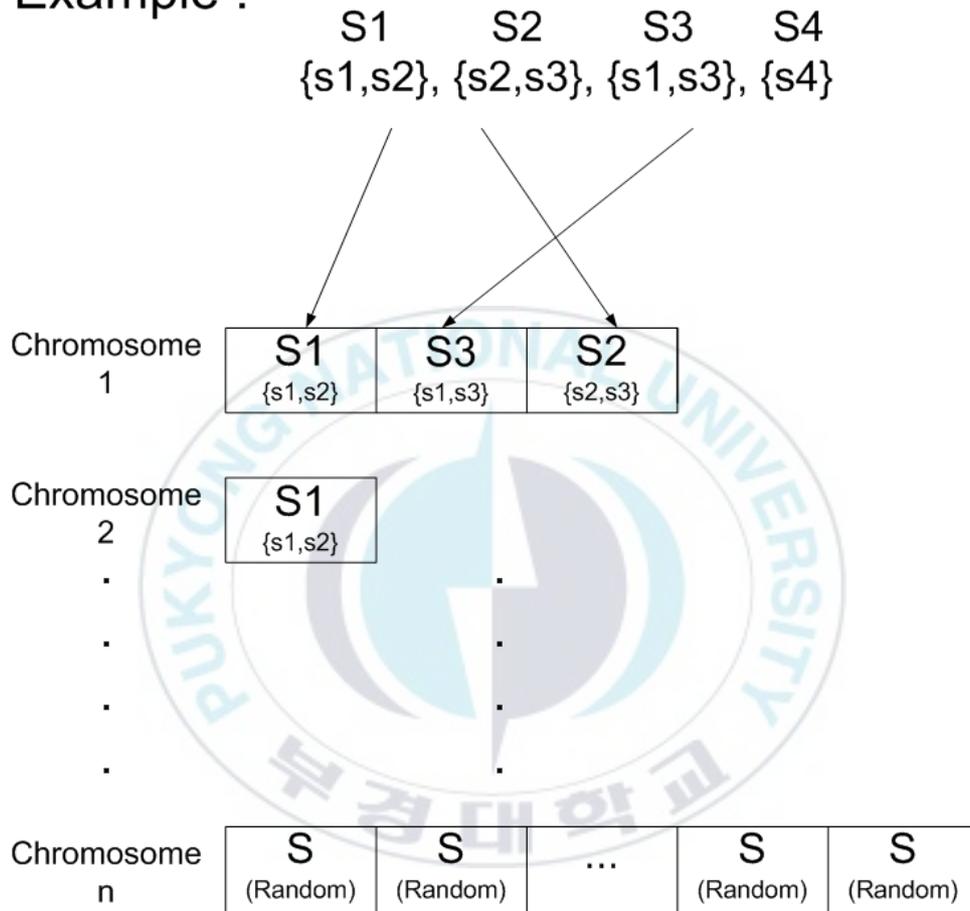


그림 5.6 염색체 부호화의 예

Fig 5.6 The Example of Chromosome Encoding

제안한 인코딩 방법의 적절성을 Kobayashi 등이 제안한 코드화의 평가규범 [34]에 따라 평가하였다. 코드화의 평가규범은 완비성(completeness), 건전성(soundness), 비중복성(non-redundancy)이다.

완비성은 문제 공간에서 해의 후보는 모두 탐색체로 표현 가능하다는 것을 나타내는 것으로, 제안한 방법에서는 모든 **target**을 커버하는 센서 집합을 얻기로 하여 탐색체가 형성되므로, 모든 해의 후보가 탐색체로 표현될 수 있다.

건전성은 유전자 알고리즘 공간의 탐색체는 모두 문제공간에서 해의 후보에 대응될 수 있음을 나타내는 것으로, 제안한 방법의 탐색체는 그 크기를 노드와 **target** 수를 고려한 최대 길이로 구성할 경우에 모든 경우를 표현할 수 있으므로 건전성을 만족한다.

비중복성은 탐색체와 해의 후보는 1 대 1로 대응될 수 있음을 나타내고, 제안한 방법에서는 하나의 탐색체가 하나의 해 후보에 대응될 수 있다.

이상과 같이 제안한 방법은 완비성, 건전성, 비중복성을 모두 만족하므로 유전자 알고리즘에서 사용되는 인코딩방법으로 적절한다.

5.2.2 선택연산

제안한 유전자 알고리즘에서는, 최적해 수렴속도를 향상시키기 위한 선택연산 방법으로서 엘리트 보존 기법을 적용하였다. 또한, 각 탐색체의 적합도를 기준으로 하여 룰렛-휠(roulette-wheel) 선택기법을 병행하여 사용함으로써 우수형질의 탐색체 요소가 다음 세대에서도 살아남을 수 있는 확률을 높였다.

룰렛-휠 선택기법은 각 탐색체의 적합도에 따라서 선택연산을 통해 선택될 확률이 비례적으로 할당된다. 즉, 각 탐색체의 적합도를 모두 합한 값만큼의 크기를 가진 룰렛-휠을 가정한다. 각 탐색체는 이 룰렛-휠 상에 자신의 적합도만큼의 공간을 배정받는다.

본 연구에서 사용한 적합도함수는 구성된 탐색체 즉 **MSC** 문제의 최적화식

에서 최대화하고자 하는 네트워크 수명 값이다.

그림 5.7은 8개의 염색체가 적합도에 비례해서 룰렛-휠 상의 공간을 배정받은 모양의 예를 보인다. 이 할당받은 공간의 면적에 비례하여 교차연산을 적용할 때 그 염색체가 선택되어질 확률이 다르다.

엘리트 보존 기법은 현 세대의 엘리트 염색체는 무조건 다음 세대에 선택되는 방법이다. 이렇게 함으로써 각 세대의 엘리트들의 유전형질이 빨리 전파될 수 있다.

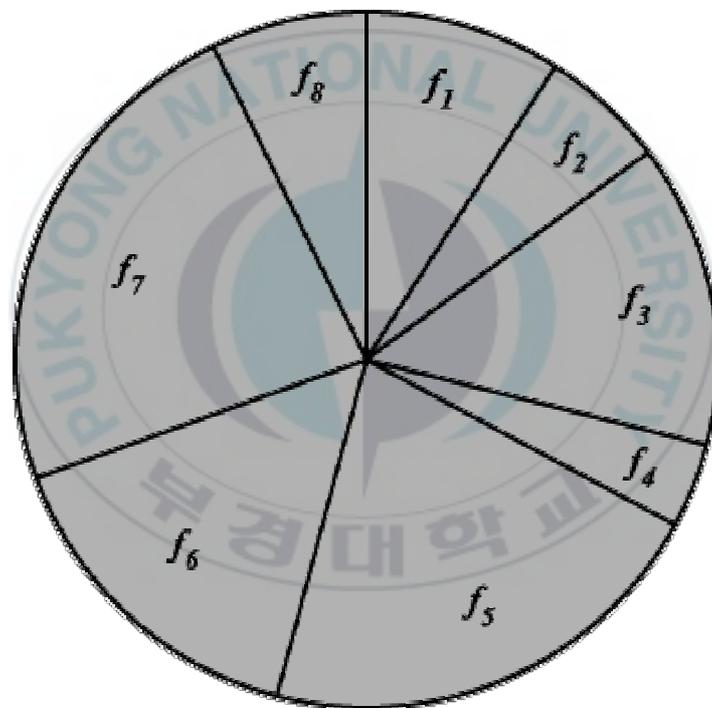


그림 5.7 룰렛-휠 선택기법의 공간배정

Fig 5.7 space assignment for roulette-wheel selection

본 논문에서는 룰렛-휠 선택연산을 통해 교차연산과 돌연변이연산에 사용될 염색체를 선택하여 새로운 염색체를 생성시키는 이용하고, 엘리트 보존 기법

을 적용하여 이전 세대의 엘리트 염색체가 현 세대에 남도록 하여 결과적으로 룰렛 휠 기법과 엘리트 보존 기법을 병행하였다. 이를 통해 최적해에 대한 수렴속도를 향상시킬 수 있다. 이러한 선택방법은 유전자 알고리즘의 다양성을 급속히 떨어뜨릴 수 있다. 제안한 유전자 알고리즘에서는 이를 보완하도록 돌연변이 연산을 재설계하였다.

5.2.3 교차연산

MSC 문제에서 염색체의 각 염기들은 모든 target을 커버하는 노드의 집합이다. 이들의 조합은 다양하게 생성될 수 있으며 이러한 집합들을 구성하는 센서노드의 수도 다양하다. 따라서 유전자 알고리즘을 이용하여 MSC 문제를 해결할 때 염색체의 인코딩은 가변 길이를 가질 수 있도록 하여야 하고, 이를 제안한 방법에서는 구현하였다. 이와 더불어서 교차연산에서도 가변길이를 갖는 염색체들에 대한 교차연산이 수행될 수 있어야 하기 때문에 기존의 교차연산으로는 한계가 있어 새로운 교차연산이 요구된다. 본 연구에서 제안한 교차연산은 그림 5.8과 같이 2단계의 수행과정을 갖는다.

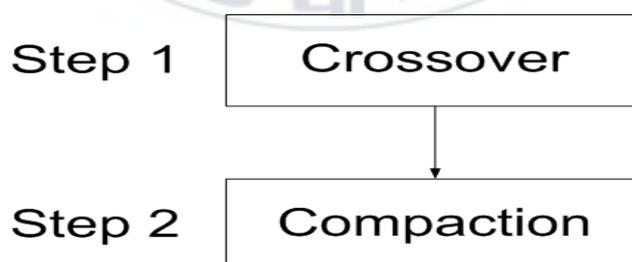


그림 5.8 교차연산 과정

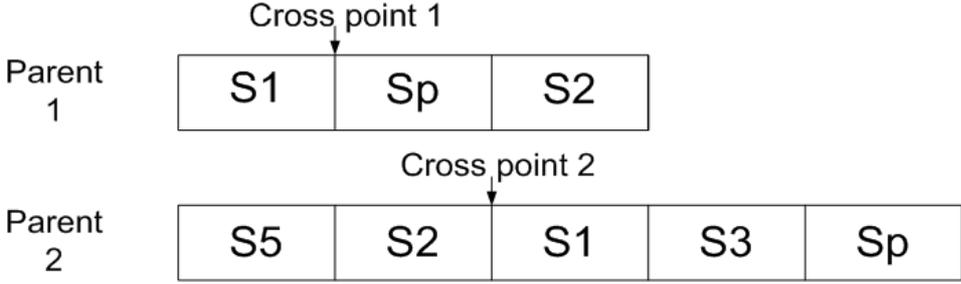
Fig. 5.8 The process of crossover operation

먼저 교차연산의 계산시간을 단축시키기 위하여 가장 간단한 1점 교차를 수행한다. 즉, 2개의 부모염색체를 선택하여 각 부모염색체에 대해 임의의 교차점을 하나씩 선택한 후 그 교차점을 기준으로 교차점 이후의 염기들을 교환함으로써 교차를 수행한다. 생성된 자식염색체에는 중복되는 염색체가 존재할 수 있기 때문에 축약 과정을 통해 중복 염색체를 제거한다. 자세한 과정은 그림 5.9와 같다.

그림 5.8의 step 1에서와 같이 parent 1, parent 2 에 대하여 임의의 교차점을 선택한다. child 1을 생성하는데 있어서 parent 1의 교차점을 기준으로 parent 2의 교차점 기준 이후와 교환하여 이루어진다. child 2도 동일한 방법으로 교환하여 구하여 질수 있다. step2 에서는 step 1의 결과로 구성된 child1,2에 대하여 중복되는 염색체 즉, child 1에서는 S1, child 2에 대해서는 S2가 중복되므로 축약과정을 통하여 중복되는 염색체를 배제한다.

3.2절의 MSC 문제의 예를 적용하여 교차연산과정을 수행하면 그림 5.10과 같다. 그림 5.10에서는 목표지점 $R = r_1, r_2, r_3$ 과 4개의 센서집합 $C = s_1, s_2, s_3, s_4$, 그리고 각 센서에 의하여 커버되는 target 집합 $s_1 = (r_1, r_2)$, $s_2 = (r_2, r_3)$, $s_3 = (r_1, r_3)$, $s_4 = (r_1, r_2, r_3)$ 을 예로 나타내었다.

Step 1. 1-point crossover



Crossover



Step 2. Compaction

Compaction

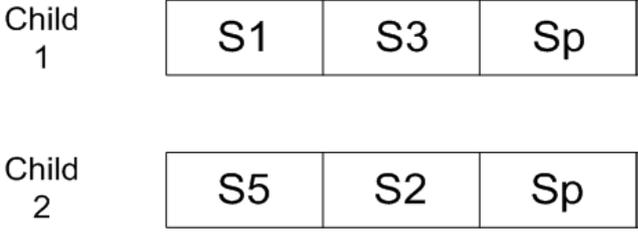
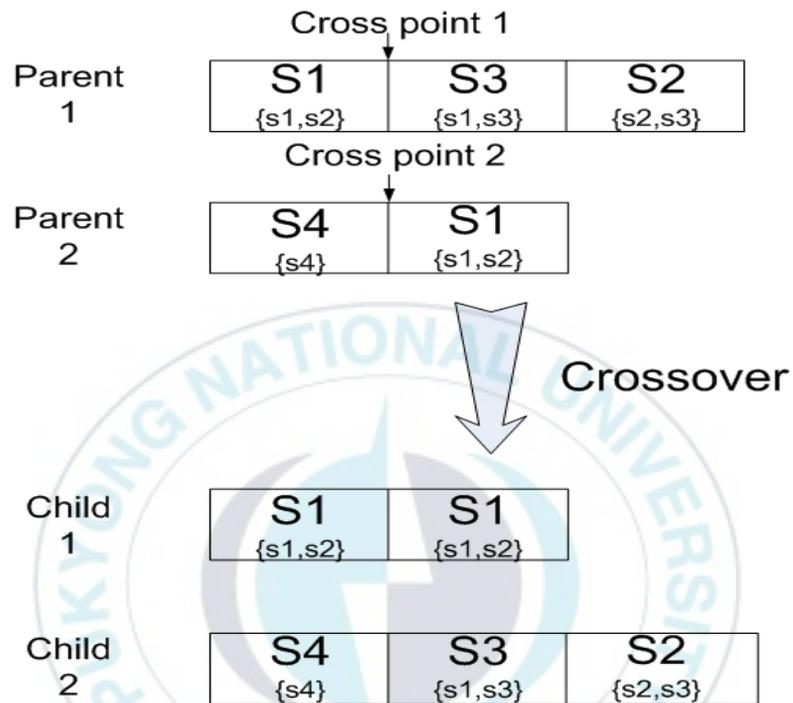


그림 5.9 교차연산의 상세절차

Fig. 5.9 The detail process of crossover

Step 1. 1-point crossover



Step 2. Compaction

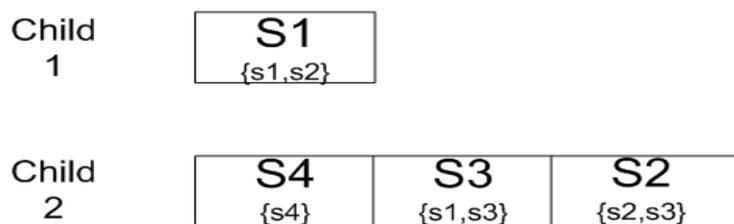


그림 5.10 교차연산의 예

Fig 5.10 The Example of crossover

5.2.4 돌연변이 연산

돌연변이 연산은 염색체를 구성함에 있어서 다양성과 가능한 넓은 해의 공간을 탐색하여 최적의 값을 찾기 위한 목적으로 사용된다.

본 연구에서 사용한 돌연변이 연산은 여러 개의 염기로 구성된 염색체에서 임의의 한 염기를 다른 염기와 교체함으로써 구현한다. 제안하는 유전자 알고리즘방식에서, 선택연산에 룰렛-휠 선택과 엘리트보존기법을 병용함으로써 최적해로의 수렴속도가 향상되는 반면 유전자 알고리즘의 다양성이 떨어질 수 있다. 이러한 단점을 보완할 수 있도록 돌연변이 연산을 설계한다.

MSC 문제에서, 유전자 알고리즘의 염색체의 길이가 길어진다는 의미는 전체 target을 커버하는 노드의 집합수가 많다는 것을 의미하고 노드 집합의 수가 많을수록 노드 자체의 에너지를 나누어서 사용할 수 있으므로 전체 네트워크의 수명이 길어진다는 가정을 한다.

전형적인 돌연변이 연산을 MSC 문제에 적용하게 되면 염기의 중복이 발생할 수 있으므로, 이를 해결하고 염색체의 크기가 클수록 최적해 집합에 가까워질 확률이 높아지는 MSC 문제의 특성을 고려하여 새로운 돌연변이연산을 설계하고 제안한다.

본 논문에서 제안한 돌연변이연산은 그림 5.11과와 같이 3단계의 수행과정을 가진다. 먼저 임의의 염색체 2개를 선택하여 두 염색체를 결합하는 결합과정을 통해 염색체의 크기를 키운다. 이는 MSC 문제에서는 한 센서가 더 많은 집합에 참여할수록 그 평가도가 높아지게 되므로, 염색체를 구성하는 염기의 수가 증가하게 되면 최적해에 근접하게 될 확률이 높아지게 된다.

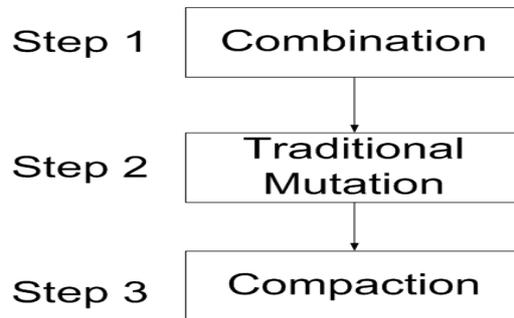


그림 5.11 돌연변이연산의 과정

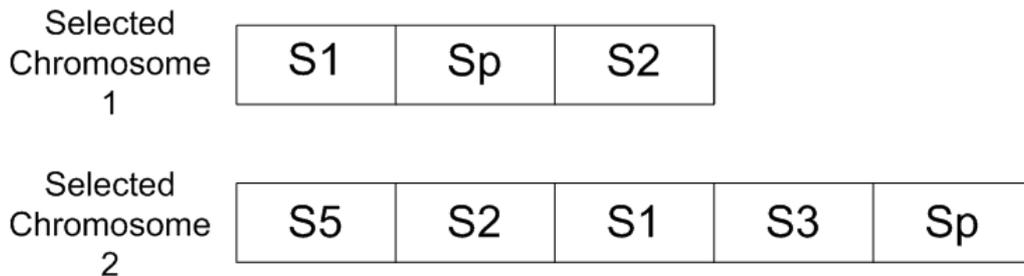
Fig. 5.11 The process of mutation

두 번째 단계에서 임의의 한 염기를 선택하여 다른 염기로 치환하는 전형적인 돌연변이연산을 적용함으로써 새로운 염색체가 생성된다(Step 2). 두 염색체를 결합하면 중복되는 염기가 발생할 수 있으므로 축약을 통해서 중복염색체를 제거한다(Step 3). 이 과정을 상세하게 설명하면 그림 5.12와 같다. 3.2 절의 MSC 문제에 대해서 제안한 돌연변이연산을 적용한 예는 다음 그림 5.13과 같다.

5.2.5 평가함수

유전자 알고리즘에서 임의의 염색체에 대한 적합도는 평가는 주어진 문제의 최적화 값에 의하여 결정된다. MSC 문제는 네트워크 수명을 최대화하는 것이 목적이며 이에 대한 최적화 식은 (3.1)에 나타나 있다. 식의 최적화함수에서 최대값을 구하는 함수의 의미는 다음과 같다. target을 커버하는 모든 집합에 대하여 각 집합에 적절한 동작시간 값을 설정한다. 모든 집합에 설정된 동작시간을 합하면 주어진 네트워크의 수명시간이 되며 이 값을 최대화하는 것이다. MSC 문제의 목적이다.

Step 1. Combination



Combination

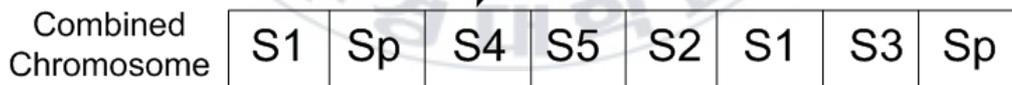
Combined Chromosome

Step 2. Traditional Mutation

Mutation

S1, S2, S3, S4, ..., Sp

1. Select 1 Gene by Random
2. Replace selected gene to 1 gene in chromosome



Step 3. Compaction

Compaction

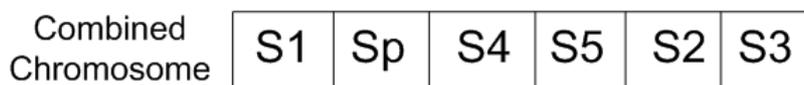
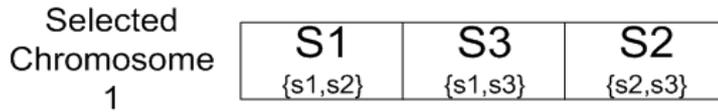


그림 5.12 돌연변이연산의 상세 과정

Fig 5.12 The detail process of mutation

Step 1. Combination



Step 3. Traditional Mutation



S1, S2, S3, S4, ..., Sp

S1 Selected by Random

1. Select 1 Gene by Random
2. Replace selected gene to 1 gene in chromosome



Step 3. Compaction

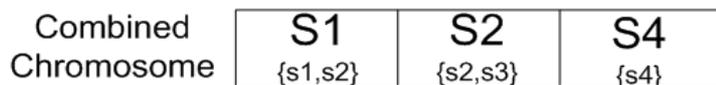


그림 5.13 돌연변이연산의 예

Fig 5.13 The Example of mutation

따라서 본 연구에서는 검색체를 평가함에 있어서 그 검색체가 나타내는 수명 값을 평가함수로 한다. 유전자 알고리즘에서 사용된 평가함수는 식(5.5) 과 같다.

$$f = \sum_{i=1}^p t_i \quad (5.5)$$

여기서, t_i 는 선택된 염기의 서비스 시간, 즉 노드집합들의 동작시간이고, p 는 검색체 수이다. 즉, 한 검색체를 구성하는 염기들의 집합에 대하여 동작시간을 설정하고 이 염기들의 동작시간 합산한 것이 선택된 검색체의 수명이다.

5.3 병렬처리방식의 유전자 알고리즘

MSC 문제의 해에 대하여 계산 성능을 향상시키기 위하여 5.2에서 제시한 중앙처리방식의 유전자 알고리즘을 개선한 병렬 유전자 알고리즘의 설계에 관하여 기술한다.

5.3.1 병렬 유전자 알고리즘의 구성

센서 네트워크 환경에서는 많은 노드가 존재하고 각각의 노드들이 계산능력을 갖추고 있으며 무선 통신장치를 이용하여 다른 노드와의 정보전달이 가능하다. 이러한 센서 네트워크의 구조적 특성으로 인하여, 센서 네트워크에 참여하는 노드들의 계산능력과 각 노드들 간의 통신 기능 동시에 사용하는 것이 가능하게 된다. 따라서 이러한 센서 네트워크의 topology 특성을 이용하여 유전자 알고리즘을 병렬로 수행함으로써 MSC 문제를 효과적으로 해결할 수 있다.

유전자 알고리즘의 병렬처리를 위한 방법은 전술한 바와 같이 분산처리형과 중앙집중형으로 분류할 수 있다. 분산처리형 유전자 알고리즘은 각 노드들이 독립적으로 유전자 알고리즘을 수행하는 방식으로서, MSC 문제의 경우 이웃하

는 노드의 상태에 대한 정보를 가져야하는 등의 알고리즘 수행에 있어서 이웃 노드의 데이터 의존성이 높기 때문에 적절한 방법이 되지 않는다. 따라서 본 논문에서는 중앙 집중형 병렬 유전자 알고리즘을 이용하여 MSC문제를 해결하는 방법을 제안한다. 제안하는 중앙 집중형 병렬 유전자 알고리즘에서는, 각 노드들의 정보를 하나의 센터노드에서 수신하여 염색체를 생성하고, 생성된 염색체를 이용하여 개체군을 생성하여 문제 해결에 참여하는 클라이언트 노드들에 분배한 후, 각 클라이언트 노드들에서 정해진 이주주기에 맞춰서 자신의 엘리트를 센터노드에 전달한다. 센터노드에서는 이렇게 전달된 각 클라이언트 노드들의 지역 엘리트를 이용하여 전역 엘리트를 구하고, 구해진 전역 엘ittest를 다시 각 클라이언트 노드들에 분배함으로써 전역 엘ittest에 대한 수렴속도를 향상시킬 수 있다. 실제 센서 네트워크를 적용시키는 응용환경을 고려하여 센터노드와 클라이언트 노드로 구성된 센서 네트워크 환경은 그림 5.14와 같다.

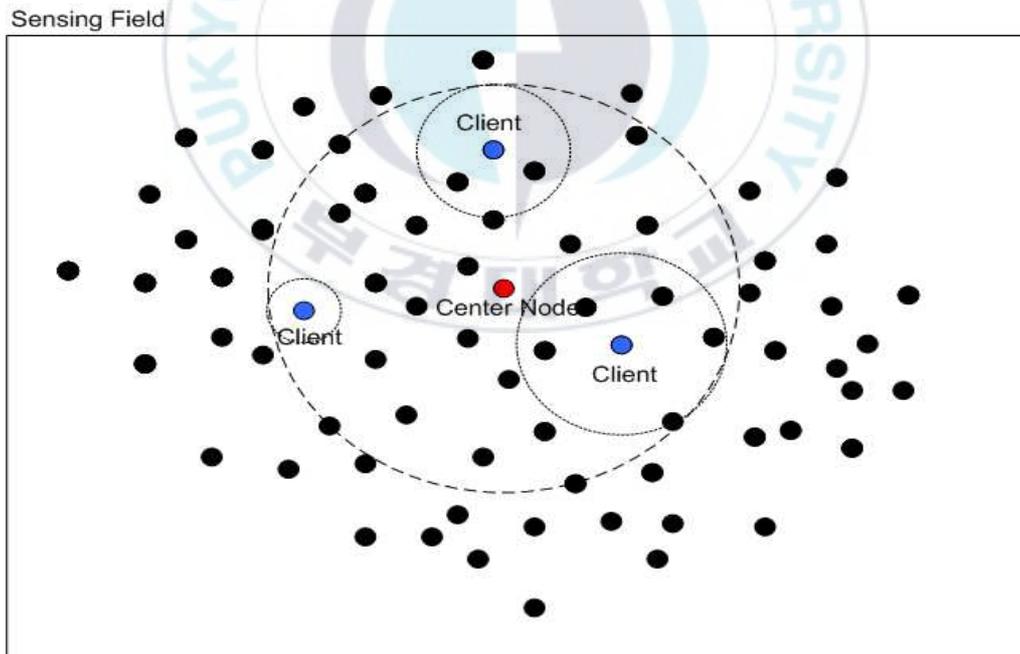


그림 5.14 네트워크에서 센터노드와 클라이언트 노드구조
 Fig 5.14 A center and client nodes on sensor networks structure

그림에서, 감지 대상영역(sensing field)에 분포된 각 노드들은 초기에 자신의 target 커버 정보를 센터노드에 전달하고, 센터노드에서 병렬 유전자 알고리즘의 유전자 부호화에 따라 초기 개체군을 생성하고 이를 클라이언트에 전달하여 처리한다. 센터노드의 통신 범위 내의 노드들 중에서 임의로 클라이언트를 선정한다. 이 때 클라이언트는 하나의 노드로 구성되거나 노드군이 될 수 있다. 단, 클라이언트 노드군의 범위는 센터노드의 통신 유효거리 안에 있는 노드여야 한다.

5.3.2 시스템의 동작

제안한 병렬 유전자 알고리즘을 적용하여 MSC 문제를 해결하는 시스템의 동작은 그림 5.15와 같다. 센터노드의 동작과 클라이언트들의 동작에 준하여서 각 동작을 단계 1~ 단계 3으로 로 구분하여 동작순서를 나눌 수 있다.

그림 5.15에서 단계 1, 단계 3은 센터노드에서 수행되고 단계 2는 클라이언트들에서 수행된다.

단계 1: 센터 노드에서 수행

- 1) 염색체를 부호화하여 생성
- 2) 생성된 염색체를 이용하여 초기개체군을 생성
- 3) 생성된 초기 개체군을 클라이언트 수를 고려하여 부개체군으로 나눔
- 4) 부개체를 각 클라이언트 노드에 이주

단계 2: 클라이언트 노드에서 수행

- 1) 센터노드로부터 개체군 받음
- 2) 유전자 알고리즘을 수행
- 3) 엘리트를 선정 : 지역(local) 엘리트
- 4) 엘리트를 센터노드에 전달

단계 3: 센터 노드에서 수행

- 1) 각 지역 엘리트들 중에서 최고의 적합도를 만족하는 엘리트 선정 :
전역(global) 엘리트
- 2) 전역 엘리트를 클라이언트에 전달

여기서, 단계 2의 엘리트 선정과정과 단계3의 전역엘리트 선정과정을 최종 세대수에 도달할 때까지 반복한다. 이와 같은 과정을 그림 5.15에 나타내었다.

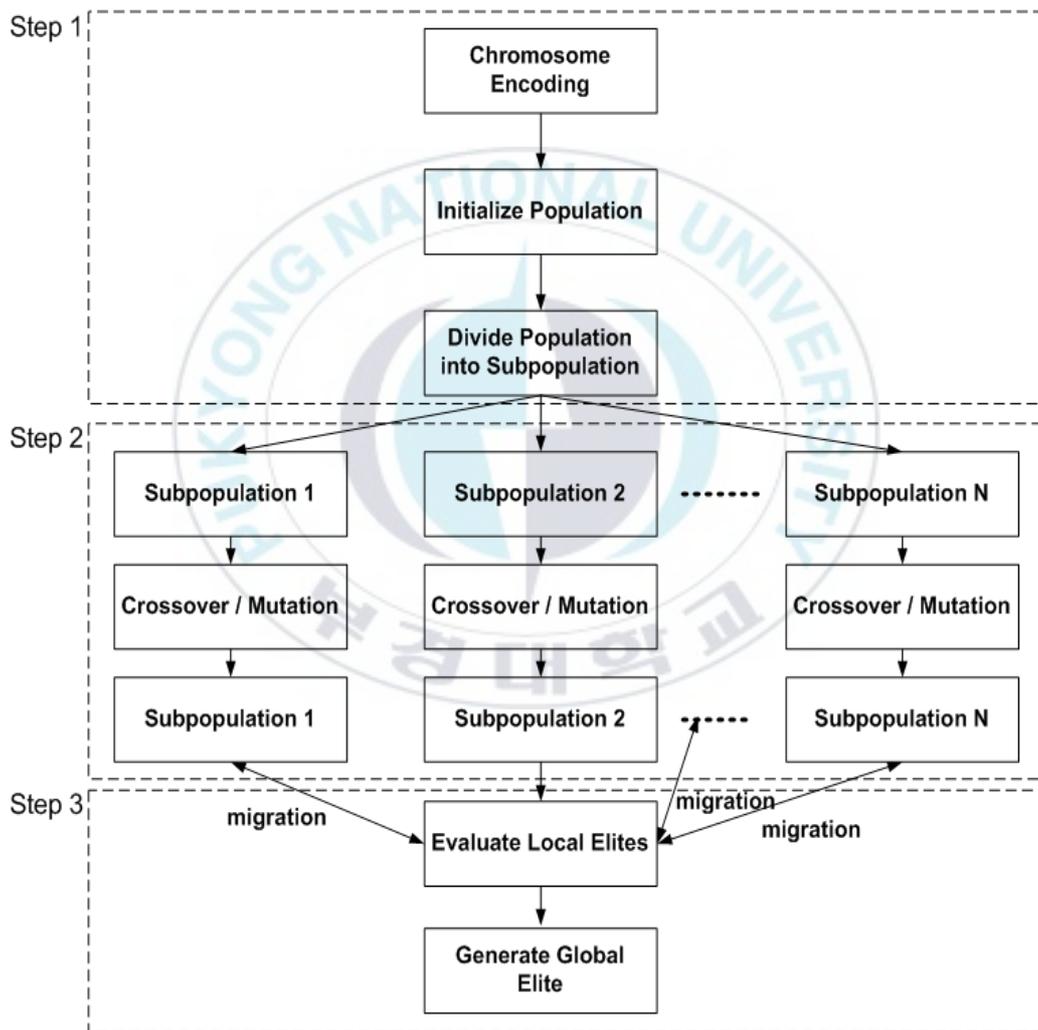


그림 5.15 전체 시스템의 동작

Fig. 5.15 The operation of system

5.3.3 센터노드의 구성 및 동작

센터노드는 주 기능은 전역(global) 최적해의 선택과 이를 각 클라이언트에 분배하는 것이다. 이러한 동작을 위하여 센터노드는 다음 그림 5.16과 같이 초기화 프로세스(Initialize process), 이주관리자(Migration Manager), 평가 프로세스(Evaluation Process)로 구성된다.

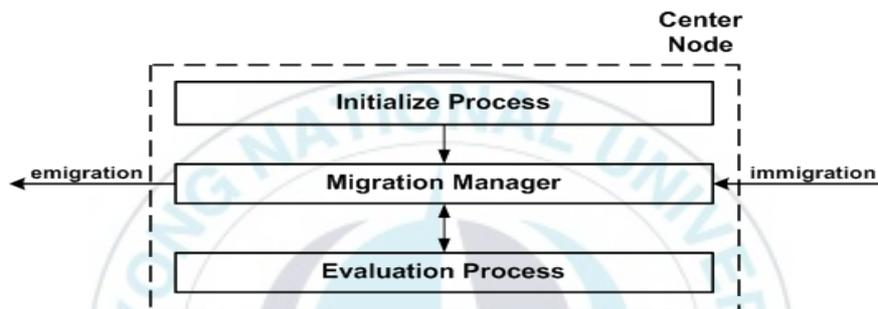


그림 5.16 센터노드의 구성

Fig 5.16 The consist of center node

그림 5.16에서, 초기화 프로세스에서는 염색체 부호화 및 초기 개체군 생성을 담당하고, 이주관리자에서 각 지역 엘리트의 이주와 전역 엘리트의 이주를 담당한다. 평가 프로세스에서는 각 지역 엘리트의 평가를 수행하고 이들 중 적합도가 가장 높은 지역 엘리트를 전역 엘리트로 설정한다.

5.3.4 클라이언트 노드의 구성 및 동작

각 클라이언트들은 지역(local) 최적해를 구하기 위해서 개별적으로 유전자 알고리즘을 수행하며 이를 위해서 클라이언트는 다음 그림 5.17과 같이 이주

관리자와 유전자 알고리즘 실행부로 구성된다.

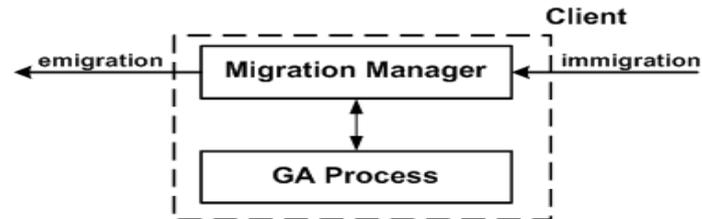


그림 5.17 클라이언트의 구성

Fig. 5.17 The consist of client

그림의 클라이언트 구조에서, 이주관리자(migration manager)는 지역 엘리트
를 센터노드로 전송하고(emigration), 센터노드로부터 받은 전역 엘ittest를 유전
자 알고리즘 실행부(GA process)에 전달한다. 실행부는 전달받은(immigration)
전역 엘ittest를 이용하여 유전자 알고리즘을 적용하여 지역 엘ittest를 갱신한
다.

제 6 장 시뮬레이션 및 고찰

MSC 문제의 해결방안으로 제안한 유전자 알고리즘의 유효성을 확인하기 위한 시뮬레이션을 수행하고 이를 고찰한다. 시뮬레이션의 목적은 다음과 같다.

- 1) MSC 문제에 대한 유전자 알고리즘의 유효성 확인
- 2) 유전자 알고리즘 방식의 특성관찰
- 3) 불균일한 에너지를 갖는 센서노드와 구성된 네트워크 환경에서 제안한 방식과 기존의 방법과의 비교

1)의 이유는, NP-complete 인 MSC 문제에 대하여 최적화 기법인 유전자 알고리즘을 적용함에 있어서 타당성을 확인하기 위함이다. 2)의 이유는, 연구의 주요 목적이 알고리즘에 대한 계산시간을 줄임과 함께 유효한 값을 획득하는 방법을 유도하는 것이므로 유전자 알고리즘 방식에서 최적값으로 수렴하는 과정을 분석하여 유효한 값에 이르게 하는 요소 중에서 연산수행 횟수와 관련이 깊은 항목을 관찰하고자 한다. 3)의 이유는 다음과 같다.

센서 네트워크는 한번 분포되어 수명이 다 할 때까지 운영하고 종료하는 경우도 있지만 여러번 반복적으로 센서노드를 분포시켜 연장 운영할 수 있으며 또한 각 센서노드에 따라 각각 수행되는 무선통신, 감지, 국부적인 연산활동 등으로 인하여 모든 센서노드에 대하여 균일한 에너지 소모는 가능하지 않다. 이와 같은 불균일한 에너지를 갖는 센서노드를 고려하면 기존의 방법은 센서 네트워크를 정상적인 운영에 앞서 노드 스케줄링을 행하는 전처리 과정이 수시로 발생할 수 있다. 이러한 전처리 횟수의 증가는 센서네트워크의 지속적인 감지 기능의 QoS를 저하시키는 요인으로 작용한다. 따라서 보다 현실적인 에너지 분포환경을 고려한, 센서 네트워크의 QoS 관점에서 제안한 방법과 기존의 방법을 비교하였다.

중앙처리 방식의 유전자 알고리즘을 CGA(centralized genetic algorithm)라 하고, 병렬처리방식의 유전자 알고리즘을 PGA(parallel genetic algorithm)라 한다.

시뮬레이션에 사용된 프로그램은 Java (SE version 1.6.0_01)로 구현하였으며, HP proliant ML350(INTEL zeon 3.0GHz, 4 CPU)에서 MS Server 2003 EE 운영 체제에서 수행되었다.

6.1 유효성 검정 시뮬레이션

기존의 알고리즘과의 비교를 통하여 제안한 유전자 알고리즘방식의 유효성을 확인하였다. 비교에 사용된 알고리즘은 MSC 문제를 발견적 greedy (greedy-heuristic) 방식[12]과 간단한 greedy 방식을 사용하였다. 이하에서는 발견적 greedy 방식을 GH-MSC, 간단한 greedy 방식을 SG-MSC라 한다.

SG-MSC 방식은 3.2에서 예에 대한 해결방법 중 하나이다. 즉, 모든 target을 커버하는 노드의 집합을 disjoint 조건으로 하여 집합의 수를 최대화하는 방법이다. 이 방법은 disjoint 조건을 만족하는 집합의 수를 최대화문제를 greedy 방법으로 구현한 것이다. 즉 집합수의 수가 최대가 된다는 것은 네트워크의 수명이 최대가 된다는 의미이다. SG-MSG 방식은 다음과 같은 절차로 수행된다.

절차 1) 1개의 target을 감지하는 센서노드 하나를 선정한다.

절차 2) 절차 1에서 선정한 target과 노드를 제외한 나머지 target과 센서노드에 대하여 절차 1을 수행한다.

3) 모든 target이 커버되면 선정된 노드를 하나의 집합으로 하여 동작 모드로 설정하여 모든 에너지가 소모될 때까지 active 노드로 한다.(수명: 1)

절차 4) 남아 있는 센서노드에 대하여 절차 1부터 3까지를 반복 수행한다.

6.1.1 유전자 알고리즘방식의 유효성

GH-MSC 방식은 MSC 문제의 해로서 짧은 수행시간과 우수한 해를 도출하였다[12]. 이와 함께 SG-MSG 방식과 제안한 CGA 방식과의 비교를 통하여 유효성을 확인하였다.

시뮬레이션에 적용된 환경은 다음과 같다.

노드가 분포되는 지역은 500m x 500m 정방형으로 노드와 목적지점이 랜덤하게 위치한다.

- 노드의 수 : 100, 200, 300
- target 수 : 5
- 노드의 감지반경 : 100

그리고 CGA방식에 적용한 요소로서, 교차율은 0.9, 돌연변이율은 0.7, 세대수는 100으로 설정하였다. 탐색체 길이의 경우, 적용되는 노드의 수에 따라 해의 크기도 같이 증가하여야 하므로 노드의 수의 2배가 되도록 설정하였다.

그림 6.1은 3가지 방식에 대하여 상기의 환경에서 적용하여 노드 수 변화에 따른 센서 네트워크 수명에 대한 값이다. 그림에 의하면 제안한 CGA방식이, 최적의 값을 도출하는 GH-MSC에 의한 결과에 근접하는 값을 보이고 있으며 SG-MSC 방식에 비해서는 상대적으로 우수한 결과를 나타내고 있음 알 수 있다. 또한 CGA방식의 경우 세대수를 충분히 늘리고, 탐색체의 길이를 길게 설정하면 GH-MSC의 결과에 근접하는 결과를 기대할 수 있다.

따라서 제안한 유전자 알고리즘의 방식이 MSC 문제에 대하여 유효함이 입증된다.

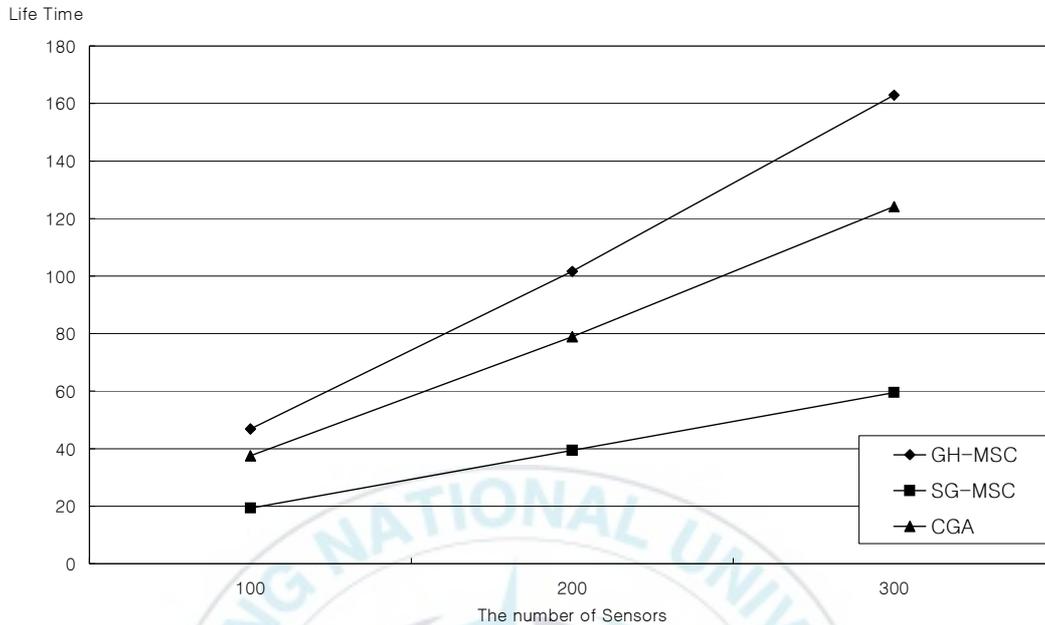


그림 6.1 센서의 수에 따른 수명 비교(target 수 : 5)

Fig 6.1 The comparison of lifetime as the number of sensors (no. of target: 5)

6.1.2 유전자 알고리즘의 특성 평가

유효성이 확인된 유전자 알고리즘이 MSC 문제에 적용되었을 때의 특성을 관찰하였다. 적용된 문제에 대하여 유전자 알고리즘만이 갖는 교차율, 돌연변이율, 세대수, 탐색체길이에 따라 최적값으로 수렴되는 조건을 분석하여 유효한 값에 수렴하는 과정에서 알고리즘 수행시간과 관계하는 요소를 중심으로 관찰하였다. 이러한 관찰을 통하여 제안된 유전자 알고리즘이 일반 유전자 알고리즘이 갖는 특성을 갖는가를 확인함과 동시에 특정문제에서(여기서는 MSC 문제) 유효한 값을 도출하는 요소값(교차율과 돌연변이율)을 예상할 수 있다. 시뮬레이션 환경은 유전자 알고리즘 요소로서, 세대수는 100, 탐색체 길이는

10, 그리고 교차율과 돌연변이율을 0.1부터 0.9까지의 값을 갖도록 하였다. 네트워크 환경은 target 수를 5로 고정하고 센서노드 수를 10, 100, 400로 변화시켰다.

본 연구에서 제안한 CGA와 PGA 방식 두 가지가 모두 유전자 알고리즘을 기반으로 하고 있으며, 본 시뮬레이션의 목적이 유전자 알고리즘 특성 분석이므로 여기서는 CGA 방식만 적용하였다.

6.1.2.1 교차율 변화에 따른 수명 평가

교차율변화에 따른 네트워크 수명 값의 변화를 그림 6.2에 나타내었다.

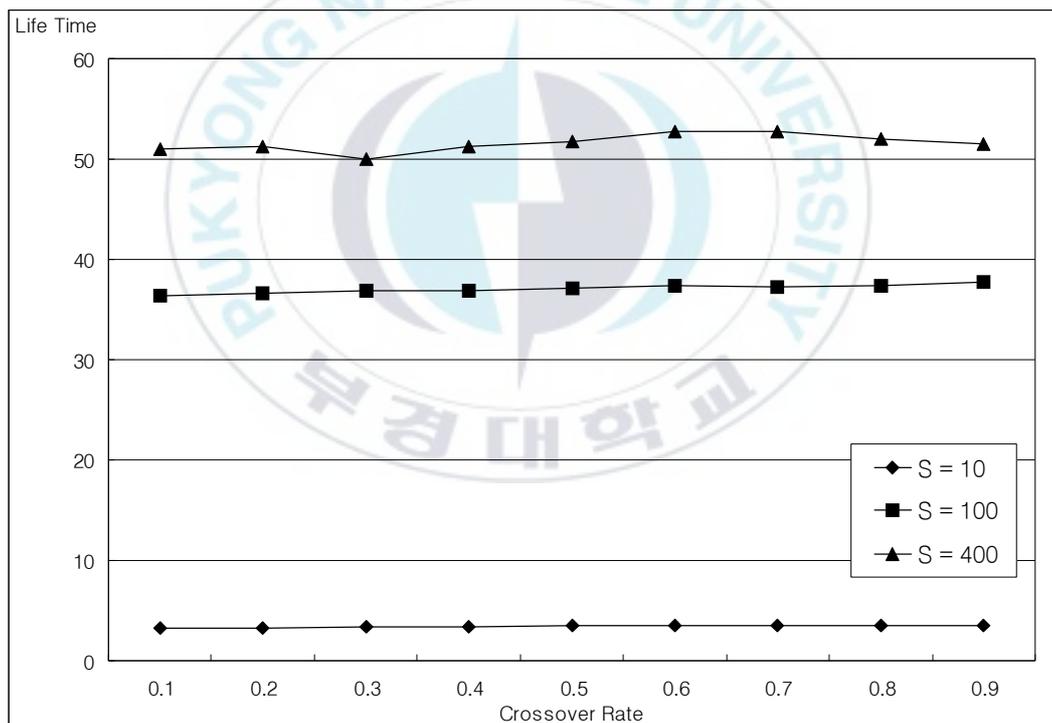


그림 6.2 CGA에서 교차율 변화에 따른 수명

Fig. 6.2 The comparison of lifetime to crossover rate in CGA

그림에서 노드의 수가 적을 경우에는 네트워크의 수명에 크게 좌우되는 해가 없어서 교차율을 증가 시켜도 네트워크 수명에는 변화가 거의 없다.

그러나 너무 낮은 교차율은 미세하지만 수명의 차이를 나타내므로 유전자 알고리즘을 적용할 경우 0.3이상의 교차율을 적용하는 것이 바람직하다.

6.1.2.2 돌연변이율 변화에 따른 수명 평가

돌연변이 연산은 한정된 해의 공간에서 보다 넓고 다양하게 다양한 해를 탐색하기 위해 사용된다. 알고리즘의 수행에 있어서 돌연변이율을 높인다는 것은 보다 다양하게 해를 찾는다는 의미이다. 유전자 알고리즘에서 돌연변이율이 높으면 그 만큼 더 많은 해공간의 탐색이 가능하고, 지역 해에서 탈출할 수 있는 확률이 높아지나 돌연변이율이 너무 높은 경우 랜덤 탐색과 유사한 특성을 나타내게 된다. 돌연변이율 변화에 따른 네트워크 수명에 대한 시뮬레이션 결과는 그림 6.3과 같다.

일반적으로 해의 공간을 구성하는 해의 요소수가 적으면 돌연변이율을 높게 설정하여도 최적해를 찾는 확률에는 큰 영향을 주지 않는다. 그러나 해 공간을 구성하는 요소의 수가 많게 되면 다양한 탐색을 통하여 최적해를 찾는 것이 필요하다. 그림 6.3에 의하면 센서노드 수가 적을 경우에는 돌연변이율의 증가에 따른 네트워크 수명 값의 변화가 거의 없음을 보여준다. 그러나 센서노드의 수가 많은 경우에는 돌연변이율이 높아지면 네트워크의 수명 값도 증가함을 보여준다. 즉 다양하고 광범위하게 해를 찾는다는 것이다.

센서노드가 많을 경우, 시행횟수를 줄이기 위한 작은 값의 교차율 적용(0.1에서 0.3)은 상대적으로 수명 값을 감소시키는 결과를 발생하게 한다. 그림 6.3에 의하면 노드의 수가 많을 경우에는 적절한 돌연변이율(0.3 이상의 값)을 유지하는 것이 좋을 수 있다. 따라서 센서노드 수가 많을 경우 적절한

돌연변이율을 유지함으로써 유효한 네트워크 수명값을 기대할 수 있다.

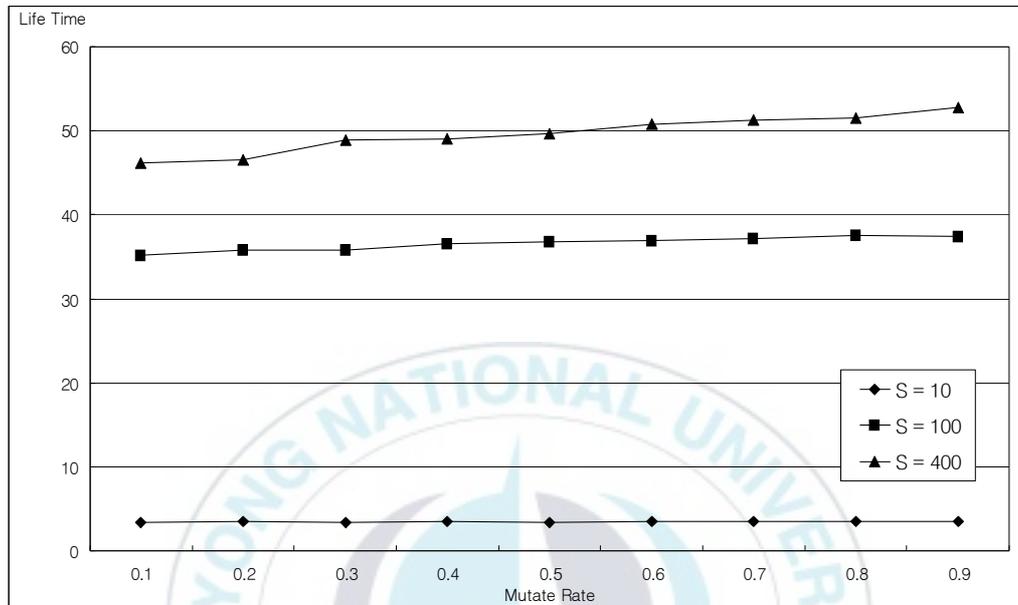


그림 6.3 CGA에서 돌연변이율 변화에 따른 수명
 Fig. 6.3 The comparison of lifetime to mutation rate in CGA

6.2 중앙처리방식과 병렬처리방식의 비교

MSC 문제에 대하여 유전자 알고리즘을 적용한 두 가지 방식 즉, CGA와 PGA의 성능을 비교하고자 한다. 전술한 바와 같이 PGA 방식의 경우 센서 네트워크의 topology 특성을 활용한 방식으로 CGA 방식에 비하여 보다 나은 성능을 위하여 개발한 방식이다. 두 가지 방식이 유전자 알고리즘을 기반으로 하고 있으므로, 공통적으로 적용됨과 함께 성능에 영향을 주는 세대수 변화에 따른 네트워크의 수명 값을 관찰한다. 또한 노드 증가에 따른 두 가지 방식의 성능을 비교한다.

시뮬레이션 환경은 6.1.1과 동일하게 교차율은 0.9, 돌연변이율은 0.7, 염색체 길이는 100으로 하였다. PGA에서 클라이언트 노드의 수는 3개로 하였으며 클라이언트에서의 지역엘리트 이주주기는 10세대로 고정하였다.

6.2.1 세대수에 따른 네트워크 수명값의 비교

먼저 CGA 방식과 PGA 방식에 대하여 세대수의 증가에 따른 네트워크 수명값이 최적값으로의 수렴하는 과정을 관찰한다. 일반적으로 유전자 알고리즘에서 세대수가 많을수록 좋은 해를 구하는 확률이 높아진다.

PGA의 경우, 여러 개의 클라이언트 노드에 의하여 지역 엘리트를 만들어내며 센터노드에서는 클라이언트로부터 전달받은 지역 엘리트로부터 가장 우수한 염색체를 선정하여 다시 각 클라이언트로 전달하여 우수한 염색체를 만드는 과정을 하기 때문에 CGA 방식에 비하여 좋은 결과값을 도출할 확률이 높다.

그림 6.4는 네트워크 환경에서 노드수가 100, target 수가 10 이고, 유전자 알고리즘 요소에 대하여 교차율 0.9, 돌연변이율은 0.7, 염색체 길이는 100으로 설정하였으며, PGA에 대해서는 3개의 클라이언트 노드와 지역 엘리트의 이주 주기를 10세대로 하였다. 그림에 나타난 바와 같이 PGA방식의 경우 10세대 당 이주하는 특성으로 인하여 매 이주 시점에 CGA 방식보다 빠른 수렴도를 보이고 있다. PGA방식에서 클라이언트 노드수를 많이 설정하면 보다 나은 개선도를 보일 수 있음이 예측된다. 따라서 유전자 알고리즘방식에서 세대수가 한정된 조건이라면 PGA 방식이 나은 결과를 보임을 알 수 있다.

본 시뮬레이션에서는 통신에 따른 에너지 소모를 고려하지 않았다. CGA 방식과는 달리 PGA 방식은 병렬로 수행되지만 엘리트의 이주에 따른 통신을 요구하게 된다. 클라이언트 노드수가 많으면 많은 에너지를 소모하게 되며 이에 따라 네트워크 전체에 대한 수명에 영향을 미칠 수 있다. 또한 엘리트 이주 주기를 짧게하여 가능한 지역 엘리트 이주횟수를 증가시키면 보다 우수한 개선도

가 기대되나 이주횟수도 통신횟수와 관계있으므로 무조건 많이 할 수는 없다.

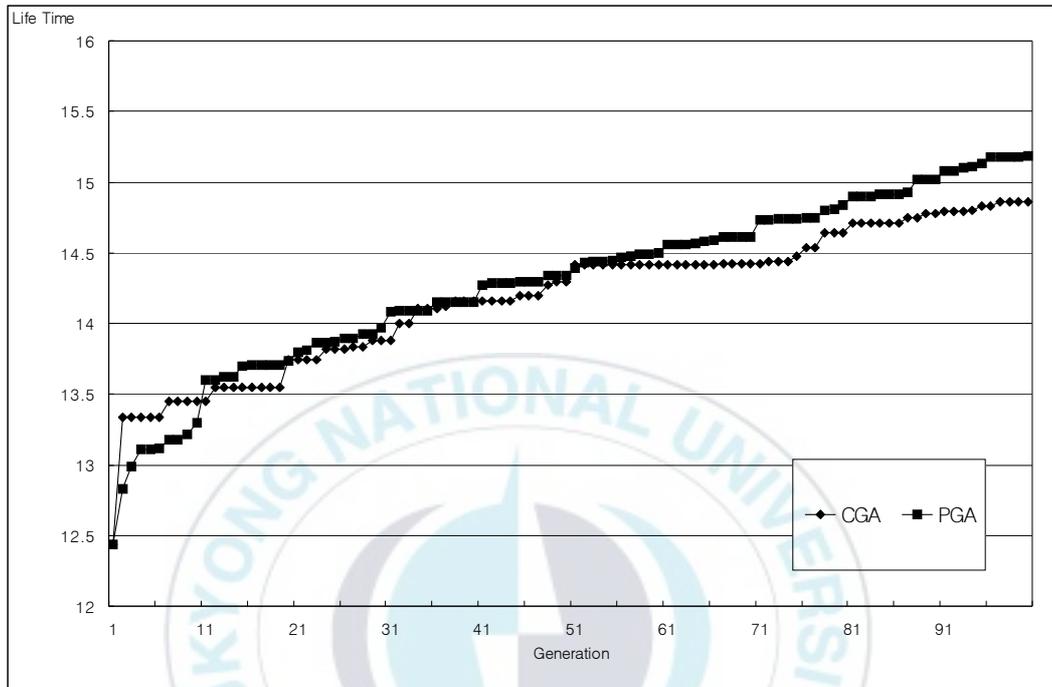


그림 6.4 세대 수에 따른 네트워크 수명(노드 수 : 100, target 수: 10)

Fig. 6.4 Lifetime on the number of generations
(no. of nodes: 100, no. of targets: 10)

결론적으로, 제한된 세대수에서는 PGA방식이 CGA 방식보다 우수하나, PGA의 경우 클라이언트 노드의 수와 이주주기에 따라 통신량이 발생하게 되어 통신에너지 소모에 따른 네트워크 수명에 영향을 줄 수 있다. 따라서 이러한 유효 세대수와 클라이언트 노드 수, 이주주기와 통신 횟수에 따른 에너지소모를 종합적으로 고려하여 효과적인 유전자 방식의 알고리즘 설계가 가능하다.

6.2.2 센서 수에 따른 수명

센서노드 수 변화에 따른 두 가지 유전자 알고리즘 기법을 적용하여 네트워크의 수명 값을 비교하였다. 네트워크 환경에서 센서 수를 600개에서 1400개까지 100개 단위로 변화시킨 반면, target 수는 50으로 고정하였다. 유전자 알고리즘 관련 요소의 값으로, 교차율은 0.9, 돌연비율은 0.7, 염색체 길이는 100으로 하였다. 또한 비교대상의 알고리즘이 동일한 유전자 알고리즘 기반이므로 세대수는 10으로 하였다. PGA에서 사용한 클라이언트 노드 수는 3개 설정하였다. 그림 6.5는 시뮬레이션 결과이다.

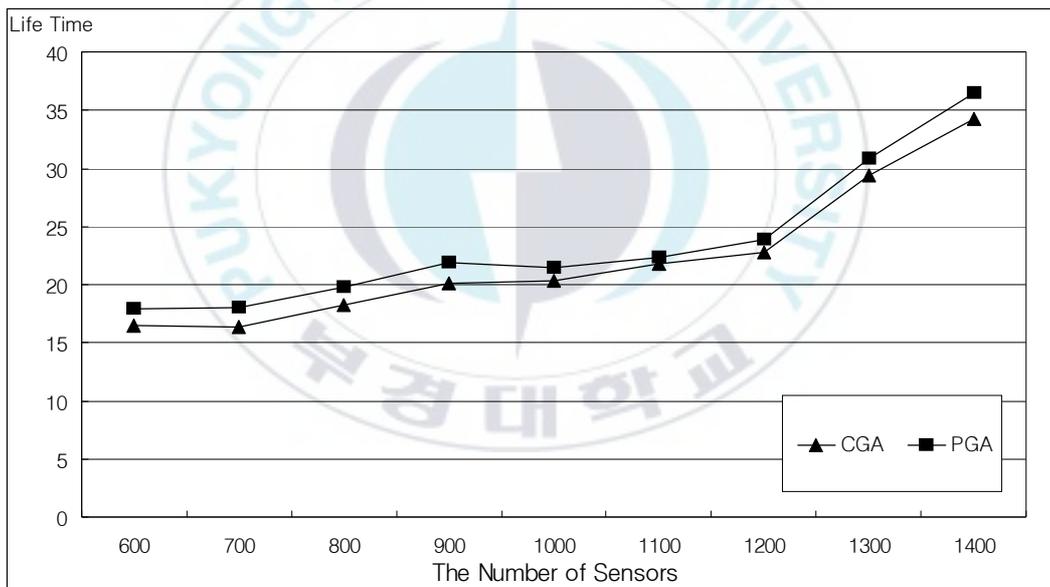


그림 6.5 노드 수에 대한 네트워크 수명(target 수 : 50)

Fig. 6.5 수명 on number of nodes(no. of targets : 50)

그림에 나타난 바와 같이 전반적으로 PGA 방식이 CGA방식에 비하여 우수한 결과를 보이고 있다. 그 이유는 PGA 방식의 경우 동일한 알고리즘이 3개의 클

라이언트 노드에서 병렬적으로 수행되며 각 클라이언트에서 제공하는 지역 엘리트들 기반으로 검색체가 구성되므로 상대적으로 우수한 결과를 도출하기 때문이다. 그러나 PGA 방식의 경우 센터 노드와 클라이언트 노드사이에 지역 엘리트의 이주를 위한 데이터 전송이 필요하기 때문에 CGA 방식에 비하여 클라이언트 노드와 센터 노드의 에너지 소모가 부가적으로 발생하는 단점이 있다. 이러한 부가적인 통신 에너지 소모는 이주주기와 밀접한 관계가 있다. 따라서 이주주기를 충분히 길게 설정함과 동시에 가능한 많은 수의 클라이언트 노드를 알고리즘 수행에 참여시킴으로서 보다 나은 결과의 도출이 가능함을 알 수 있다.

6.3 센서노드의 모드전환

본 장에서는, 중앙처리 방식의 스케줄링 알고리즘인 GH-MSc 방식과 CGA 방식에 대하여 4장에서 기술한 운영모드 전환 횟수에 대한 시뮬레이션을 수행하고 결과를 기술하였다.

센서 네트워크를 운영함에 있어서 주 임무인 감지기능을 수행하기에 앞서 노드 스케줄링이 선행된다. 4장에서 기술한 바와 같이 네트워크 전체 운영과정을 스케줄링시간과 동작시간으로 구분하였을 때, 분산처리방식의 경우 그림 4.1과 같이 각 라운드마다 감지기능을 중단하고 스케줄링을 수행해야하는 운영모드의 전환(operation mode switching)이 발생한다. 중앙처리방식의 알고리즘에서는 그림 4.4와 같이 한번의 스케줄링 알고리즘 수행을 통하여 얻어진 정보를 전체 센서노드들에게 전달한 후 매 라운드마다 모든 센서노드의 동작을 동기화하기 위한 신호의 전송이 필요하다. 이러한 동기신호의 전송을 위하여 운영모드의 전환이 일어나야 한다.

센서 네트워크의 운영모드 전환 횟수는 지속적인 감지 기능의 QoS 척도로 사용될 수 있으며 운영모드 전환 빈도는 작을수록 좋다.

운영모드 전환에 대한 시뮬레이션은 GH-MSC 방식과 CGA 방식에 대하여 다음과 같이 수행되었다. GH-MSC 방식에서는 그림 4.3의 알고리즘을 구현하여 집합 $C_1, C_2 \dots C_i$ 를 구한다. 4.3절에서 기술한 바와 같이 서로 다른 집합에 대하여 1 라운드씩 적용하며, 연속적으로 동일한 집합에 대해서는 동일 라운드로 취급 가능함으로 현재 구해진 노드집합과 이전의 노드집합을 비교하여 서로 상이하면 모드전환 횟수를 1씩 증가한다. CGA 방식에서는 알고리즘이 수행되면 최적의 검색체가 구성되며 검색체를 구성하는 염기가 모든 target을 커버하는 노드의 하나의 집합이 된다. 염기 하나 하나가 순서에 따라 네트워크에 적용되기 때문에 결과적으로 검색체를 구성하는 염기의 수가 모드전환 횟수가 된다.

센서 네트워크는 임무를 수행함에 있어서 한번의 센서노드 분포로 응용이 끝나지 않고 필요에 따라 여러번 반복하여 분포시키는 경우도 있다. 초기에는 모든 센서노드가 동일한 에너지량을 갖는다고 가정할 수 있지만 여러번 반복하여 분포된 경우 각 센서노드가 갖는 에너지량은 동일하지 않다. 또한 데이터 전송의 경우 동기신호, 국부적 데이터 전달에 있어서 통신량은 서로 다를 수 있으므로 시간이 경과함에 따라 사용하는 에너지량은 차이가 난다. 따라서 시뮬레이션은 두가지 경우를 고려하여 수행하였다. 첫 번째, 센서 네트워크가 처음 적용되는 경우와 같이 모든 센서노드의 보유에너지가 동일하다는 조건의 경우와 두 번째, 각 센서노드가 갖는 에너지량이 서로 다른 경우이다.

시뮬레이션 환경의 네트워크 관련 요소 중 target 수는 5, 센서노드 수는 100, 200, 300, 400으로 설정하였다. CGA 관련 요소 중 교차율은 0.9, 돌연변이율은 0.7, 세대수는 100, 검색체 길이는 노드수의 2배로 설정하였다.

시뮬레이션은 센서노드 수 변화에 따른 GH-MSC 방식과 CGA 방식에서 운영모드 전환 횟수(K) 및 네트워크 수명, 그리고 각각에 대한 비율을 $\alpha_K, \alpha_{lifetime}$

으로 나타내었다. 여기서 CGA의 K와 수명값은 시행수 100회에 대한 100세대 값의 평균이다. 먼저 각 센서노드가 갖는 초기 에너지량을 100으로 단일한 에너지량을 갖는 환경에서의 시뮬레이션 결과를 표 6.1에 나타내었다.

표 6.1 모드전환횟수와 수명(단일 노드 에너지)

Table 6.1 No. of mode switching and lifetime (unified node energy)

적용방식 노드 수	GH-MSK		CGA		비율(CGGA/GH-MSK)	
	K	수명	K	수명	$\alpha_K(\%)$	$\alpha_{lifetime}(\%)$
100	4822	4824	233	3897	4.83202	80.783582
200	10451	10452	330	8156	3.157593	78.032912
300	16760	16762	362	12856	2.159905	76.697291
400	22544	22544	402	16864	1.78318	74.804826

표 6.2는 서로 다른 노드 에너지를 갖는 환경을 고려한 시뮬레이션 결과이다. 각 센서노드가 갖는 에너지량은 [1..100]의 비정규분포를 갖도록 랜덤하게 발생하도록 하였다.

표 6.1과 6.2에 의하면 네트워크 수명에 있어서는 GH-MSK 방식이 우수하나 모드 전환횟수는 CGA방식이 우수함을 보이고 있다. 두 방식에서 이러한 K값에 대한 차이가 발생하는 이유는 다음과 같이 분석된다. GH-MSK 방식은 알고리즘의 수행에 있어서 target을 커버하는 집합에 대한 순서가 중요하므로 $C_1, C_2 \dots C_i$ 를 구함에 있어서 동일한 집합이 있다 하더라도 연속적으로 재배열할 수 없다. 따라서 각기 다른 라운드에 적용하여야 하므로 모드 전환횟수가 증가하게 된다. 유전자 알고리즘의 경우 염색체를 구성하는 염기를 선정할 때

동일한 엮기는 미리 배제된다. 유전자 알고리즘에서의 엮기 1개가 GH-MSC 방식에서 1개의 집합 C_i 에 해당되므로 엮기의 수가 모드 전환횟수가 된다. 엮색체의 길이가 길면 엮기의 수도 증가할 수 있다. 그러나 5.2.1에 기술한 바와 같이 본 연구에서 제안한 유전 알고리즘에서는 엮색체의 길이를 일정하게 제한하지 않고 가변적으로 구성되도록 하였다.

표 6.2 모드전환횟수와 수명(다양한 노드 에너지)

Table 6.2 No. of mode switching and lifetime (various node energy)

적용방식 노드 수	GH-MSC		CGA		비율(CGA/GH-MSC)	
	K	수명	K	수명	$\alpha_K(\%)$	$\alpha_{lifetime}(\%)$
100	2318	2336	146	1649	6.298533	70.590753
200	5202	5210	284	3582	5.459439	68.748562
300	7940	7947	356	5253	4.483627	66.102478
400	11353	11358	393	7655	3.46164	67.397429

GH-MSC 방식에서 모드 전환횟수가 에너지가 균일하지 않은 상태에서 균일한 상태보다 적게 일어나는데 그 이유는 다음과 같다. 알고리즘에서 target을 커버하는 노드를 선정함에 있어서 상대적으로 에너지가 많은것을 우선하여 선택한다. 따라서 에너지가 많은 노드가 연속적으로 선택됨으로서 에너지가 많은 노드들로 구성된 집합이 연속적으로 선택되기 때문이다.

CGA 방식과 GH-MSC 방식에서 네트워크 수명의 차이가 나는 이유는 네트워크에 분포된 센서노드의 평균에너지 차이에 기인하는 것이다.

CGA 방식에서는 전환의 횟수에 있어서 센서노드 에너지량이 균일하거나, 불

균일할 경우에도 큰 차이가 없을 알 수 있다. 따라서 유전자 알고리즘 방식은 에너지량의 분포방식에 관계없이 적용가능하며 네트워크의 운영모드 전환을 고려한 응용분야에서 우수한 결과를 나타냄을 확인하였다.



제 7 장 결론

본 논문에서는, 센서노드에 대한 동작모드 스케줄링함으로써 네트워크 수명을 연장하는 방법에 대한 고찰과 함께 유전자 알고리즘을 이용한 해결방안을 제시하였다.

MSC문제는 모든 target 지점을 커버하는 센서노드의 집합을 non-disjoint 집합으로 구하여 각 센서노드에 적절한 동작시간을 할당함으로써 네트워크의 수명을 연장하는 방법을 정의한 것이다. MSC 문제는 NP-complete 문제이며 이를 해결하는 기존의 방법들은 최적화 기법을 적용한 방법들이 제시되었다.

본 연구에서는 이러한 NP-complete 문제에 유전자 알고리즘을 적용하여 그 특성 분석을 통하여 최적의 값에 근접하는 유효한 값을 도출함과 네트워크의 지속적인 감지성을 높이는 방안을 제안하였다. 또한 기존의 알고리즘과 같이 하나의 노드에서 수행되는 중앙처리방식의 유전자 알고리즘과 함께 센서 네트워크가 갖는 네트워크 topology 특성을 이용한 병렬 유전자 알고리즘 방법도 함께 개발하였다.

시뮬레이션을 통하여 MSC 문제에 대하여 제안한 방식이 유효한 결과를 도출함을 확인하였으며, 교차율, 돌연변이율, 염색체 길이, 세대수 등의 값 설정에 따라 알고리즘의 수행시간과 결과값 사이의 상관관계를 갖는 일반적인 유전자 알고리즘의 특성도 확인하였다. 이러한 특성은 유전자 알고리즘방식이 주어진 문제에 대하여 수행시간과 결과값에 대한 유연한 적용성을 가짐을 보여준다. 또한 센서노드들이 갖는 에너지량을 단일화한 환경과 균일하지 않은환경을 고려하여, 네트워크 운영상 발생하는 운영모드 전환 횟수를 관찰하는 시뮬레이션을 수행하였다. 센서 네트워크의 운영모드 전환이 빈번히 일어날 경우 지속적인 감지기능을 방해하는 요인이 되며 노드동작을 스케줄링함에 있어서 이러한

모드전환은 반드시 고려되어야 할 사항이다.

시뮬레이션 결과, 기존의 방법에 비하여 제안한 유전자 알고리즘방식이 적은 모드전환 횟수를 보임에 따라 센서 네트워크의 지속적인 감지기능의 QoS 에 대하여 우수성이 입증되었다.

본 연구의 의의는 다음과 같다.

- 1) 유전자 알고리즘을 이용한 MSC 문제의 해를 제안
- 2) 센서 네트워크가 갖는 구조적인 특성을 이용한 병렬처리가 가능한 유전자 알고리즘의 개발
- 3) 감지기능 QoS 측면에서 유전자 알고리즘방식이 좋은 결과를 보임을 확인

향후 연구방향으로는, 병렬 유전자 알고리즘 방식에서 클라이언트 노드와 센터노드 간의 엘리트 이주특성에 따라 발생하는 통신기능에서 발생하는 에너지 소비모델을 적용하여 실질적인 스케줄링 모델로의 발전을 도모한다. 아울러 주어진 문제의 분석과 통신 에너지 소모를 고려한 최적의 클라이언트 노드의 수를 선정하는 방법에 관한 방안에 대한 연구도 이루어져야 할 것이다.

참 고 문 헌

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks. March 2002.
- [2] A. Bharathidasan and V. A. S. Ponduru, "Sensor networks: an overview", <http://www.csif.cs.ucdavis.edu/~bharathi/sensor/survey.pdf>, page accessed on Nov. 30, 2006.
- [3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," Proc. IEEE/ACM MobiCom Conference, pp. 263-270, 1999.
- [4] D. Estrin, L. Girod, G. Pottie and M. Srivastava, "Instrumenting the world with wireless sensor networks," Proc. International Conference on Acoustics, Speech, and Signal Processing , vol. 4, pp. 2033-2036, 2001.
- [5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Piste, "System architecture directions for network sensors," Proc. International Conference on Architectural Support for Programming Languages and Operating Systems, 2000.
- [6] G. J. Pottie and W. J. Kaiser, "Wireless sensor networks", Communications of the ACM, vol. 43, pp. 51-58, May 2000.
- [7] H. Qi, S. S. Iyengar, K. Chakrabarty, "Distributed sensor networks - a review of recent research," Journal of the Franklin Institute, vol. 338, pp. 655-668, 2001.

- [8] L. Schwiebert, S.D.S. Gupta and J. Weinmann," Research challenges in wire-less networks of biomedical sensors," Proc. IEEE/ACM MobiCom, pp. 151-165, 2001.
- [9] S. Tilak, N. B. Abu-Ghazaleh and W. B. Heinzelman, "A taxonomy of wire-less micro-Sensor network models," ACM Mobile Computing and Communications Review, vol. 6, no. 2, 2002.
- [10] Yi Zou,"COVERAGE-DRIVEN SENSOR DEPLOYMENT AND ENERGY-EFFICIENT INFORMATION PROCESSING IN WIRELESS SENSOR NETWORKS", PhD Thesis, Duke University,2004.
- [11] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, Energy- Aware Wireless Microsensor Networks, IEEE Signal Processing Magazine, 19 (2002), pp 40-50.
- [12] M. Cardei, M. Thai, Y. Li, and W. Wu, Energy-Efficient Target Coverage in Wireless Sensor Networks, IEEE INFOCOM 2005, Mar. 2005.
- [13] M. T. Thai, F. Wang and Ding-Zhu Du, Coverage Problems in Wireless Sensor Networks: Designs and Analysis, International Journal of Sensor Networks, special issue on Coverage problems in Sensor networks, 2005
- [14] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975
- [15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, New York, NY, 1989
- [16] D. E. Goldberg, *Simple genetic algorithms and the minimal, deceptive problem*, In L. Davis, editor, *Genetic Algorithms and simulated annealing*,

- Morgan Kaufmann, Los Altos, CA. pp.74-88, 1987
- [17] L. Chambers, *Practical Handbook of Genetic Algorithms*, CHC Inc., pp.60-62, 1995
- [18] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, second ed. Berlin: Springer-Verlag, 1994
- [19] J. Kahn, R. H. Katz and K. Pister, Next Century Challenges: Mobile Networking for Smart Dust, ACM MOBICOM Conference (Aug. 1999).
- [20] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, A survey of energy efficient network protocol(I), *Wireless Networks*, 7 (2001), pp 343-358.
- [21] M. Cardei, J. Wu, Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks, accepted to appear in *Computer Communications*, special issue on Sensor Networks.
- [22] J. Carle and D. Simplot, Energy Efficient Area Monitoring by Sensor Networks, *IEEE Computer*, Vol 37, No 2 (2004),pp 40-46.
- [23] D. Tian and N. D. Georganas, A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks, *Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications* (2002).
- [24] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks, *First ACM Conference on Embedded Networked Sensor Systems* (2003).
- [25] H. Zhang and J. C. Hou, Maintaining Sensing Coverage and Connectivity in Large Sensor Networks, *NSF International Workshop on Theoretical and*

Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks
(Feb. 2004).

- [26] S. Slijepcevic, M. Potkonjak, Power Efficient Organization of Wireless sensor Networks, IEEE International Conference on Communications, (Jun. 2001).
- [27] M. Cardei, D.-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, accepted to appear in ACM Wireless Networks.
- [28] K. Kar and S. Banerjee, Node Placement for Connected Coverage in Sensor Networks, Proc. of WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (2003).
- [29] X.-Y. Li, P.-J. Wan, and O. Frieder, Coverage in Wireless Ad-hoc Sensor Networks, IEEE Transactions on Computers, Vol 52 (2002), pp 753-763.
- [30] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, Coverage Problems in Wireless Ad-Hoc Sensor Networks, IEEE Infocom (2001), pp 1380-1387.
- [31] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, HICSS 2000.
- [32] S. Lindsey, C. S. Raghavendra, PEGASIS: Power Efficient GATHERing in Sensor Information Systems, IEEE Aerospace Conference (Mar. 2002).
- [33] Z. H. Cui, and Z. C. Zeng, "Schema Theorem of Real-Coded Nonlinear Genetic Algorithm," *Proc. of the First Int'l Conf. on Machine Learning and Cybernetics*, pp.1429-1431, Nov. 2002
- [34] 문병로, *유전알고리즘*, 두양사, 2003
- [35] E. Cantù-Paz, "A survey of parallel genetic algorithms," IllGAL Report

- 97003, The University of Illinois, 1997
- [36] M. Nowostawski, and R. Poli, "Parallel genetic algorithm taxonomy," *Knowledge-Based Intelligent Information Engineering Systems, 1999. Third International Conference*, pp.88-92, Aug. 1999
- [37] C. C. Petty and M. R. Leuze, "Theoretical Investigation of a Parallel Genetic Algorithm," *Proc. of ICGA-89*, pp.398-405, 1989
- [38] L. Wang, A. A. Maciejewski, H. J. Siegel and V. P. Roychowdhury, "A Comparative Study of Five Parallel Genetic Algorithms Using the Traveling Salesman Problem," *1998 IPPS/SPDP. Proc. of the First Merged International*, pp.345-349, Apr. 1998
- [39] M. Golub and D. Jakobovic, "A New Model Of Global Parallel Genetic Algorithm," *Information Technology Interfaces ITI 2000*, pp.363-368, Jan. 2000
- [40] E. Cantù-Paz, "Designing Efficient Master-Slave Parallel Genetic Algorithms," *Genetic Programming: Proc. of the Third Annual conference*, pp.455-460, 1998
- [41] B. Manderick and P. Spiessens, "Fine-gained Parallel Genetic Algorithms," *Proc. of ICGA-89*, pp.428-433, 1989
- [42] P. Spiessens and B. Manderick, "A Massively Parallel Genetic Algorithm, Implementation and First Analysis," *Proc. of ICGA-91*, pp.279-285, 1991
- [43] R. J. Collins and D. R. Jefferson, "Selection in Massively Parallel Genetic

- Algorithms," *Proc. of ICGA-91*, pp.249-256, 1991
- [44] S. Baluja, "A massively distributed parallel genetic algorithm(mdpga)," *Technical Report CMU-CS-92-196R*, Carnegie Mellon University, Pittsburg, PA, 1992
- [45] S. Baluja, "The evolution of genetic algorithms: Towards massive parallelism," in *Proceedings of the Tenth International Conference on Machine Learning*, San Mateo, CA, pp.1-8, 1993
- [46] L. Tan and K. A. Smith, "A New Parallel Genetic Algorithm," *Proc. of the International Symposium on Parallel Architectures, Algorithm and Networks*, IEEE, 2002
- [47] R. Hauser and R. Männer, "Implementation of Standard Genetic Algorithm on MIMD Machines," *Parallel Problem Solving from Nature, PPSN III*, pp.504-513, 1994
- [48] J. P. Cohoon, W. N. Martin, and D. S. Richards, "A Multi-population Genetic Algorithm for Solving the k-Partition Problem on Hypercubes," *Proc. of ICGA-91*, pp.244-248, 1991
- [49] R. Tanese, "Distributed Genetic Algorithms," *Proc. of ICGA-89*, pp.434-439, 1989
- [50] T. Starkweather, D. Whiteley, and K. Mathias, "Optimization using distributed genetic algorithm," *Parallel Problem Solving from Nature*, Springer Verlag, 1991

- [51] M. Munetomo, T. Yoshiaki, and S. Yoshiharu, "An Efficient Sigma Exchange Algorithm for a Subpopulation-Based Asynchronously Parallel Genetic Algorithm and Its Evaluation," *IPSI*, vol.35, no.9, pp.1851-1827, 1994

