

# 제약만족 최적화 문제를 위한 백트래킹 탐색의 구조화

손 석 원<sup>†</sup>

요 약

모든 제약만족 최적화 문제의 해를 구하는 일반화된 알고리즘을 구하는 것은 매우 어렵다. 그러나 결정 변수의 특성에 따라 세분화된 문제는 해를 위한 알고리즘을 구하기에 더 쉽다는 가정을 할 수 있다. 이와 같은 가정 하에 문제를 세분화 시키는 문제분류규칙을 제안하고 세분화된 문제의 특성에 맞는 백트래킹 알고리즘을 개발한다. 백트래킹을 이용한 깊이우선탐색에서 해를 빨리 찾기 위한 방법 중 하나는 탐색되는 노드의 순서를 효과적으로 배열하는 것이다. 정적 특성이 우세한 무선 센서 네트워크의 클러스터 헤드 위치문제와 동적 및 정적 특성의 혼합특성을 갖는 RFID 리더 간섭 최소화 문제를 선택하여 최적의 변수 순서화 알고리즘을 개발하고 기존의 방법과 비교하였다. 결과적으로 문제를 세분화시킴으로써 체계적인 탐색을 위한 백트래킹의 구조화를 실현하였다. 또한 개발된 백트래킹 알고리즘의 성능이 우수함을 보였다.

키워드 : 백트래킹 탐색, 제약만족 최적화, 변수 순서화, 문제분류규칙, 무선자원할당

## A Backtracking Search Framework for Constraint Satisfaction Optimization Problems

Surgwon Sohn<sup>†</sup>

ABSTRACT

It is very hard to obtain a general algorithm for solution of all the constraint satisfaction optimization problems. However, if the whole problem is separated into subproblems by characteristics of decision variables, we can assume that an algorithm to obtain solutions of these subproblems is easier. Under the assumption, we propose a problem classifying rule which subdivide the whole problem, and develop backtracking algorithms fit for these subproblems. One of the methods of finding a quick solution is efficiently arrange for any order of the search tree nodes. We choose the cluster head positioning problem in wireless sensor networks in which static characteristics is dominant and interference minimization problem of RFID readers that has hybrid mixture of static and dynamic characteristics. For these problems, we develop optimal variable ordering algorithms, and compare with the conventional methods. As a result of classifying the problem into subproblems, we can realize a backtracking framework for systematic search. We also have shown that developed backtracking algorithms have good performance in their quality.

Keywords : Backtracking Search, Constraint Satisfaction Optimization, Variable Ordering, Problem Classifying Rule, Radio Resource Allocation

### 1. 서 론

현대의 복잡한 공학적 문제들의 특징 중 하나는 그것이 이산적인 (Discrete) 성질을 갖고 있다는 것이다. 따라서 선형 계획법(Linear Programming)에서 많이 사용되는 심플렉스 (Simplex) 방법이나 개선된 심플렉스 방법을 사용할 수 없고 정수 계획법(Integer Programming)을 이용하여야 한

다. 또한 이러한 문제들은 복잡도가 증가할수록 기하급수적인 시간이 걸리게 되며 정수 계획법으로 완전해를 구하기 힘든 NP-hard[1] 문제인 경우가 많고 조합 최적화 (Combinatorial Optimization)의 속성을 갖고 있다.

무선자원할당 문제는 대부분 조합 최적화 문제로 모델링 될 수 있다. 제약만족문제(Constraint Satisfaction Problem)는 결정 변수(Decision Variables)와 이 변수에 할당되는 값, 그리고 제약조건(Constraints)의 집합들로 정의된다[2]. 제약만족문제의 해는 변수의 도메인 값을 제약조건을 만족하는 상태로 변수에 할당하는 것이다. 이 제약만족문제의 확장된 개념으로서 목적함수를 추가한 것이 제약만족 최적화문제 (Constraint Satisfaction Optimization Problem)[3]이고, 이것

※ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2008-313-D00954).

† 정 회 원 : 호서대학교 벤처전문대학원 부교수

논문접수 : 2011년 2월 8일

수정일 : 1차 2011년 3월 30일, 2차 2011년 4월 6일

심사완료 : 2011년 4월 7일

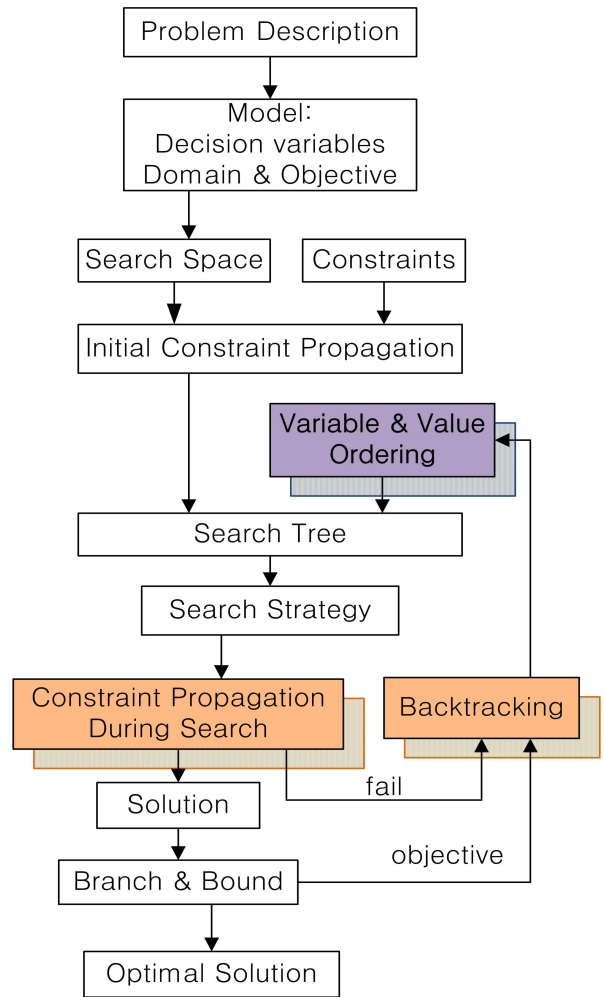
은 흔히 제약 최적화 문제(Constraint Optimization Problem)라고 부른다.

이들 문제를 풀기 위한 두 개의 주요한 방법이 있는데 첫째는 나무탐색(Tree Search) 방법이다. 이것은 백트랙(Backtrack)에 기반을 두고 있으므로 백트래킹 탐색(Backtracking Search)[4]이라고도 불린다. 두 번째 방법은 지역 탐색(Local Search)을 이용한 방법이며 언덕등반 탐색(Hill Climbing Search), Simulated Annealing, Tabu Search 등이 대표적 방법이다. 이 두 개의 방법은 모두 그 적용문제에 따라 장단점이 존재한다[5].

(그림 1)은 나무탐색을 이용하여 제약만족 최적화 문제의 해를 구하는 절차를 나타내며, 그 절차의 첫째 단계는 문제를 정확히 기술하는 것이고 두 번째 단계는 제약만족 모델링을 하는 것이다. 이 때 결정 변수, 제약 조건, 그리고 목적 함수를 결정한다. 각각의 결정 변수들은 그 변수에 할당될 수 있는 값들의 집합인 도메인(Domain)을 갖고 있다. 이 때 탐색 공간이 만들어지고 모든 값들의 조합을 일일이 열거(Enumeration)함으로써 해를 구한다. 그렇지만 이 방법은 매우 비효율적이므로 모든 변수에서 제약조건에 위배되는 값들을 미리 제거시키는 초기 제약전파를 실행한다. 그렇게 되면 탐색 공간이 대폭 감소하게 되는데 이것을 탐색 나무(Search Tree)라고 부른다. 나무탐색은 기본적으로 변수들에 값을 차례로 하나씩 할당함으로써 부분해를 서서히 완전해로 만드는 것이다. 만약에 부분해에 관계된 어떤 변수가 제약조건을 위배하면 바로 전에 탐색된 변수로 백트래킹하게 된다.

탐색 나무에서 해가 왼쪽 위에 나타나도록 변경시킬 수 있는데 이 때 필요한 것이 변수 및 값 순서화(Variable and Value Ordering)[6]이다. 이렇게 되면 다음 단계에서 실행하는 지역 탐색 방법으로서 깊이우선 탐색(Depth First Search) 방법을 사용하면 해를 빨리 찾을 수 있게 된다. 이 지역탐색을 실행하면서 제약전파를 하게 되는데 이것은 제약조건을 위배하는 미래의 값들을 미리 제거하여 탐색을 빠르게 한다. 만약에 변수에 더 이상의 값들을 할당할 수 없게 되면 변수 순서화 단계로 백트래킹을 하게 되어 다른 변수를 시도하게 된다. 백트래킹하지 않을 때에는 해가 구하여지게 된다. 이것은 모든 제약조건을 만족하였다는 의미이다.

최적해를 구하는 것은 기본적으로 모든 해를 구하고 그것들을 반복적으로 비교하는 것이다. 변수 순서화 알고리즘은 백트래킹을 최소화시켜 제한된 시간 내에 모든 해를 효율적으로 구하게 된다. 그러므로 최적해를 구하기 위해서는 Branch & Bound 알고리즘[4]을 사용하여 현재 노드에서 구해진 해를 상한 값(Upper Bound)으로 놓고 다음에 구해질 값과 계속 비교하여 최적의 해를 구하여야 한다. 지역 탐색으로서 깊이우선탐색을 사용하지 않고 Tabu Search같은 Metaheuristics를 사용하게 되면 변수 순서화나 제약전파가 필요 없게 된다. 따라서 이 방법은 제약 최적화를 푸는 두 번째 방법이 되는 것이다. 본 연구는 첫 번째 방법인 나무탐색에 있어서 효율적인 백트래킹에 관한 것이다.



(그림 1) 제약만족 최적화 문제의 해를 구하는 절차

## 2. 관련 연구

제약 최적화 문제를 푸는 일반적인 백트래킹 탐색 알고리즘을 개발하는 것은 매우 어려운 일이다[3]. 최적화 문제를 풀기 위한 백트래킹 알고리즘의 일반화 연구에 대한 시도로서 Gent는 Constrainedness라는 개념을 제안하였다[7]. 이 개념은 일반적인 문제에 있어서 상태 천이를 고려하여 변수의 도메인 크기와 변수 개수 파라미터를 이용해서 Kappa라는 값을 구하게 되는데 이것을 Constrainedness라고 한다. 이 Kappa 값을 최소화시키는 방향으로 파라미터를 조정하여 새로운 경험적 변수화 알고리즘을 구하게 된다.

많은 연구자들은 백트래킹 탐색의 핵심인 경험적(Heuristic) 변수 순서화 방법을 개발하는데 연구를 집중하였다. 그 중에서도 Golomb과 Baumert[8], Haralick과 Elliott[9]는 탐색나무의 깊이를 최소화하는 알고리즘으로 동적 변수 순서화(Dynamic Variable Ordering)의 하나인 dom에 대하여 연구하였다. 제약만족 문제가 아닌 일반적인 혼합정수 프로그래밍(Mixed Integer Programming) 모델에서

도 변수 순서화는 해를 빨리 찾는데 큰 영향을 미친다[10]. 또한 제약 가중치(Constraint Weight)을 이용한 변수 순서화 방법[11]도 제안되기도 하였다.

한편으로는 문제의 특성에 따라서 정적 변수 순서화(Static Variable Ordering) 방법이 더 좋은 성능을 보일 수 있다. 정적 변수 순서화는 Freuder[12]에 의해서 소개되었는데 변수가 갖고 있는 연결도 또는 차수의 특성을 이용하였다. 즉, 높은 차수부터 낮은 순서로 변수를 정렬한 다음에 차례대로 변수를 선택하는 것으로 deg라고 표현한다. 이것은 시간에 따라서, 즉 나무의 가지가 탐색됨에 따라서 변수의 선택이 결정되는 것이 아니고 탐색하기 전에 이미 변수선택이 결정되는 것이다. 따라서 대부분의 경우 dom 알고리즘이 deg보다 우수한 성능을 보이는 것은 자연스러운 현상이다. 실제로 deg는 단독으로 사용되기 보다는 보조적인 결정 파라미터로 사용된다. 그래프 채색문제와 같이 도메인이 같은 크기일 경우 tie-breaker로서 변수의 연결도를 사용하는 알고리즘의 성능이 더 좋은 것으로 알려졌다. 이것은 Dsatur[13]로 이름 지어 졌으며 <dom, deg>라고도 표현된다.

만약 다음 변수가 도메인의 크기가 같고 연결도도 같을 때 미래의 변수의 도메인 크기가 최소인 변수를 선택하는 것이 좋을 것이다. 이것은 Dsatur 알고리즘을 확장한 것으로 3 번째 tie-breaker로 동작하며 Smith에 의해 BZ2[14]로 명명되었다. 이것은 <dom, deg, regretdmin>으로 생각될 수 있는데 여기서 regretdmin은 현재의 변수를 제외한 미래의 변수들 중 도메인의 크기가 제일 작은 변수를 말한다.

Bessiere와 Regin은 제약조건들이 약할 때 정적 변수 순서화 방법이 동적 변수 순서화 방법인 dom보다 성능이 우수하나 제약조건이 강할 때에는 deg는 성능이 매우 좋지 않다는 것을 밝혔다. 따라서 그들은 변수의 도메인 크기와 연결도 속성을 혼합한 변수 순서화 알고리즘을 제안하였는데 이것을 (dom/deg)[15]라 부른다. Boussemart는 이것을 발전시켜 dom/wdeg를 제안하였다[11]. 이러한 알고리즘의 분류 기준은 Wallace[16]에 의해 연구되기도 하였다. 본 논문에서는 무선 센서 네트워크의 클러스터 헤드 위치문제와 RFID 리더 간섭 최소화 문제와 같은 무선통신의 자원할당문제를 조합최적화 문제로 모델링하고 제안한 문제분류규칙을 이용하여 알맞은 변수 순서화 알고리즘을 개발하여 문제의 해를 구한다.

### 3. 백트래킹 탐색의 구조화

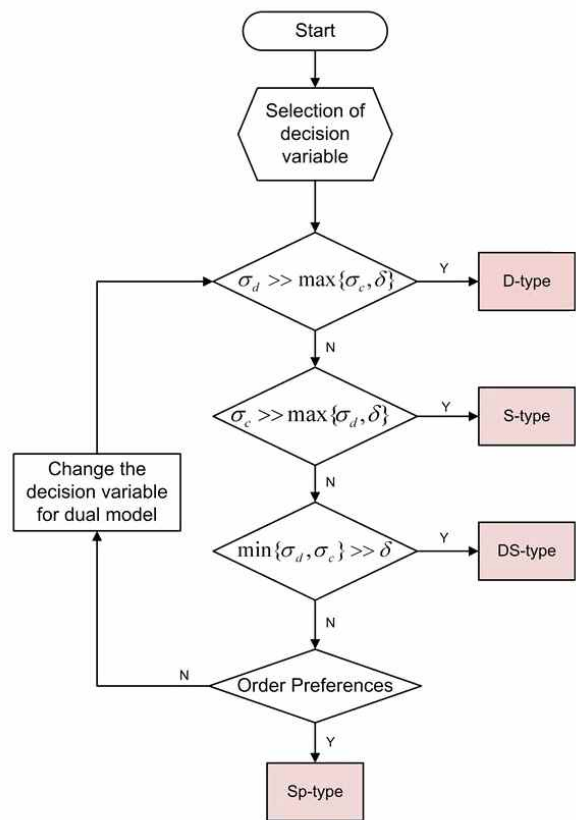
#### 3.1 문제분류 규칙 생성

모든 제약만족문제를 푸는 일반적인 백트래킹 알고리즘을 개발하는 것은 매우 어려우므로 문제를 변수 속성에 따라서 4가지로 분류하고, 각각의 세분화된 문제에 적합한 최적의 백트래킹 알고리즘을 구하기로 한다. 변수 순서화 방법에는 크게 동적 변수 순서화와 정적 변수 순서화 방법이 있지만 경우에 따라서 문제의 특성이 동적 및 정적 특성이 혼합되

어 있는 경우도 많다. 또한 정적 특성이 지배적인 경우에도 문제의 특성이 특정 파라미터에 크게 의존하는 경우가 있다. 우리는 이것을 변수의 “선호성(Preferences)”이라 부른다. 따라서 제약 최적화 문제를 분류하는데 이런 특성을 기준으로 사용해서 세분화하는 규칙을 만들 수 있다. 즉, 동적인 특성으로서 도메인 크기, 정적인 특성으로서 변수의 연결도, 그리고 또 다른 정적 특성인 선호성을 이용하여 문제를 크게 4 가지로 분류할 수 있는데 이를 문제분류규칙(Problem Classifying Rule)이라 부른다. 도메인 크기와 변수의 연결도는 서로 상관관계가 없다. 그러나 연결도와 선호성은 같은 정적 특성이기 때문에 서로 상관관계가 있다. 즉, 문제가 선호성을 갖고 있을 때에는 그렇지 않을 때보다 특정 파라미터에 의존하는 경향이 있다.

문제를 이렇게 분류하는 목적은 각각의 세분화된 문제 형태를 인식하는 것이 변수 순서화 알고리즘을 개발하는데 도움이 되기 때문이다. 결정 변수의 속성에 기반을 둔 문제분류규칙을 이용하여 문제의 형태를 분류하는 과정은 다음과 같으며 흐름도는 (그림 2)와 같다.

단계 1 : 제약 최적화 문제의 결정변수를 선택한다.



(그림 2) 문제분류규칙의 흐름도

단계 2 : 나무 탐색에 있어서 결정변수의 도메인 크기의 표준편차(od)가 연결도의 표준편차(oc)보다 매우 클 때는 이 문제를 D-type으로 규정한다. 즉, 변수의 도메인 크기가 노드를 탐색하면서 크게 변하는 문제 형태이다.

단계 3 : 변수에 있어서 연결도의 표준편차가 도메인 크기의 표준편차보다 매우 클 때에는 이 문제를 S-type으로 규정한다. 즉, 노드가 탐색됨에 따라서 도메인의 크기 변화가 기준 값 이하이면서 연결도의 크기 변화가 큰 문제 형태이다.

단계 4 : 도메인 크기와 연결도의 표준편차가 모두 실험적으로 얻어진 임의의 크기  $\delta$ 보다 매우 클 때에는 이 문제를 DS-type으로 규정한다. 즉, 동적 및 정적 특성이 혼합된 문제이다.

단계 5 : 도메인 크기와 연결도의 표준편차가 모두  $\delta$ 와 비슷하고, 그리고 변수에 선호성이 있을 때에는 이 문제를 Sp-type으로 규정한다. 이러한 문제는 D-type 및 S-type의 특징이 없는 문제로서 해결하기 어려운 문제가 될 수 있다. 그러나 선호성이 있다는 사실에 주목하여 문제형태를 결정하는 것이다.

단계 6 : 도메인 크기와 연결도의 표준편차가 모두  $\delta$ 와 비슷하고, 그리고 변수에 선호성이 없을 때에는 결정변수를 변경하여 단계 1로 되돌아간다. 이 단계에서는 문제의 형태를 규정할 변수의 속성에 관한 특징이 불분명하여 결정변수를 바꾼 다음 문제분류 규칙을 다시 적용하는 것이다. 결정변수를 바꿀 때에는 쌍대 모델(Dual Model)을 이용한다[17].

### 3.2 변수 순서화 알고리즘

문제분류규칙을 제약만족 최적화 문제에 적용하면 4 가지 형태, 즉 D, S, DS 및 Sp-type으로 세분화할 수 있으므로 이들 세부 형태에 가장 적합한 4 가지 종류의 변수 순서화 알고리즘을 개발할 수 있다. 첫 번째 변수 순서화 알고리즘은 D-type 문제를 해결하는 알고리즘으로서 DVO(Dynamic Variable Ordering) 형태이며 변수의 동적 특성에 기반을 둔 것으로서 지금까지 가장 잘 알려진 것은 dom 알고리즘이다. 이것은 다음 변수를 선택할 때 도메인의 크기가 가장 작은 변수를 선택한다. 따라서 이것은 “smallest domain first”라고도 불리며 탐색나무의 깊이를 최소화한다. 만약에 선택할 다음 변수가 같은 크기의 도메인일 경우에는 선택 지표로서 연결도 또는 차수인 deg가 가장 큰 변수를 선택하게 된다. 이것은 D<sub>sat</sub> 알고리즘[13]으로 불리며 <dom,deg>로 표시될 수 있다. 이 때 deg는 tie-breaker로 동작한다.

두 번째 알고리즘은 SVO(Static Variable Ordering) 형태이며 S-type 문제의 해에 관한 것이다. 이 형태의 알고리즘에서 가장 잘 알려진 것은 제약 그래프에서 연결도가 가장 큰 변수부터 차례로 선택하게 되는 deg 알고리즘이다. 물론 여기서도 같은 연결도일 때 tie-breaker로서 도메인 크기가 작은 변수를 선택하는 <deg, dom> 알고리즘도 생각할 수 있다. 이러한 S-type의 문제로서 클러스터링(Clustering) 문제를 들 수 있다. 응용 분야는 센서 네트워크의 클러스터 헤드의 위치문제를 들 수 있다.

세 번째 알고리즘은 DVO와 SVO 혼합 형태로서 변수의 동적 및 정적 특성을 이용하는 혼합형(Hybrid) 알고리즘으로서 DSVO로 표현할 수 있다. 제약조건이 복잡한 그래프 채

색문제는 DS-type으로 생각할 수 있다. 따라서 변수의 동적 및 정적특성을 동시에 고려하는 혼합형 알고리즘이 우수한 성능을 나타낼 것이다. 그래프 채색문제에서 기존의 D<sub>sat</sub> 알고리즘이 가장 우수한 성능을 보였는데 D<sub>sat</sub> 알고리즘은 DVO이다. 그러나 그래프 채색문제를 문제분류규칙에 적용하면 DS-type임을 알 수 있으며 지금까지 알려진 D<sub>sat</sub> 알고리즘보다 더 우수한 알고리즘이 존재함을 추측할 수 있게 된다. 이 추측되는 알고리즘은 문제의 특징이 DS-type임을 고려하여 동적 및 정적특성이 섞인 혼합형 형태의 알고리즘이 될 것이다. 이 혼합형 알고리즘으로서 지금까지 개발된 것 중 성능이 가장 좋은 것은 dom/deg과 dom/wdeg가 있다. 이것은 동적 특성인 변수의 크기와 정적 특성인 연결도의 비를 이용한 것으로 변수의 선택지표로 사용된다.

본 논문에서 제안한 wdom - deg 알고리즘 역시 DVO와 SVO 알고리즘을 결합한 일종의 혼합(Hybrid) 알고리즘이다. 그런데 dom에 가중치(weight)를 두어서 탐색속도를 빠르게 하는 역할을 한다. 적절한 w의 값을 구하는 것은 경험적으로 얻을 수 있다.

네 번째 알고리즘은 SpVO 형태로서 Sp-type의 문제에 대해서 가장 성능이 좋은 알고리즘이며 선호성을 이용한 것이다. Sp-type의 문제에는 DVO나 SVO 알고리즘이 우수한 성능을 보이지 않는다. 그것은 문제의 특성을 좌우하는 변수의 특징이 선호성에 기반을 두기 때문에 변수 순서화 알고리즘도 특정 파라미터를 찾아내어 가중치를 주어야 함을 의미한다.

### 3.3 값 순서화 알고리즘

위의 변수 순서화 알고리즘에서 변수가 선택된 다음에는 그 선택된 변수에 할당되는 값의 후보가 여러 개 있을 수 있다. 따라서 어느 값을 먼저 변수에 할당하는가 하는 문제는 첫 번째 해를 찾는 데에 있어서 시간적으로 매우 큰 영향을 미친다. 왜냐 하면 탐색 가지에서 탐색이 실패할 가지보다 해가 존재 가능한 가지를 먼저 탐색하게 되어 그 만큼 더 빨리 해를 찾을 수 있다. 그렇지만 해가 없는 문제이거나 모든 해를 찾는다면 이 값 순서화는 의미가 없게 된다. 예를 들어서 각각의 변수에 올바른 값이 할당된다면 탐색은 백트래킹이 발생하지 않고 구해질 수 있다. 주파수 할당 같은 문제에 있어서 필요한 주파수는 최소가 되어야 한다. 따라서 주파수의 최소화를 목적함수로 하는 경우에 간섭하지 않는다는 제약조건을 만족하면서 기존에 사용된 주파수의 재활용이 많을수록 전체 주파수의 최소화라는 목적함수에 적합할 것이다.

## 4. 실험

### 4.1 무선 센서 네트워크의 클러스터 헤드 위치문제

센서 노드 및 클러스터 헤드의 에너지 소비가 동시에 최소가 되도록 집단화 하는 문제는 창고위치문제와 같이 NP-hard 문제[1]가 되며 완전해를 구하기 어렵게 된다. 초

기 에너지가 동일한 노드들이 클러스터를 형성하고 노드정보를 클러스터 헤드로 전송하는 2-tier 네트워크를 고려한다. 이러한 센서 네트워크 모델은 기본적으로 LEACH[18] 모델을 따른다.

대부분의 에너지는 송신할 때 소모되며 송신주파수가 1-2GHz 일 때, 센서노드와 클러스터 헤드 사이의 거리가 약 90m 이내에서는 자유공간 모델(Free-space Model)로 해석되어 송신 에너지는  $r^2$ 에 반비례한다. 또한 거리가 90m 이상이 되면 다중경로 모델(Multi-path Model)로 해석되어 송신 에너지는 급격하게 증가하여  $r^4$ 에 반비례하게 된다. LEACH 모델에 따르면 자유공간 모델에서 k 비트의 메시지를 거리 r인 클러스터 헤드에 전송하는데 소비되는 센서노드의 전송에너지는 다음 식 (1)로 표시된다.

$$E_{tx}(k,r) = E_{elec}(k) + E_{amp}(k,r) = kE_{elec} + kP_{fs}r^2 \quad (1)$$

여기서  $E_{elec}$ 은 전자기기 회로에서 소비되는 에너지로서 거리에 관계없이 일정하게 소비된다.  $E_{amp}$ 는 송신출력 에너지로서  $P_{fs}r^2$ 로 표시되며 거리에 관계가 있는 항이다. 식(1)은 다중경로 모델에서는 식(2)로 표현된다.

$$E_{tx}(k,d) = E_{elec}(k) + E_{amp}(k,d) = kE_{elec} + kP_{mp}d^4 \quad (2)$$

여기서 d는 센서노드와 클러스터 헤드 사이의 거리인데 식(1)과 구별하기 위해 d로 나타내었다. 센서노드에서 데이터를 수신할 때에는 거리에 관계없이 일정한 에너지만 소비되며 식 (3)으로 표현된다.

$$E_{rx}(k) = E_{elec}(k) = kE_{elec} \quad (3)$$

클러스터 헤드 후보들의 집합을  $RN = \{1, \dots, m\}$  이라 하고 초기에 그 위치가 정해져 있으며 j라는 클러스터 헤드가 선택되어 설치된다. 센서노드의 집합은  $SN = \{1, \dots, n\}$  으로 표시되며 임의의 센서노드 i는 클러스터 헤드 j에 할당될 수 있다. 클러스터 헤드에서 기지국으로 전송하는데 소비되는 에너지 벡터를 X라고 한다면  $X_j$ 는 클러스터 헤드 j에서 소비되는 에너지양을 의미한다. 클러스터 헤드 j는 클러스터에 할당된  $n_j$ 개의 센서노드로부터 각각 q 비트의 메시지를 받아서 기지국으로 재전송하는데 이때 필요한 에너지는 다중경로 모델로 표현되며 식 (4)와 같다.

$$X_j = qn_j(2E_{elec} + E_{da} + P_{mp}d_j^4) \quad (4)$$

여기서  $E_{da}$ 는 클러스터 헤드가 데이터를 병합하는데 소비되는 에너지이다.  $Y_{ij}$ 는 센서노드 i가 클러스터 헤드 j에 q 비트의 데이터를 전송하는데 소비되는 에너지를 말한다.

$$Y_{ij} = q(E_{elec} + P_{fs}r_{ij}^2) \quad (5)$$

$Y_j$ 는 클러스터 j에 할당된 모든 센서노드에서 소비되는 에너지의 합을 말한다. 클러스터 내에서 센서노드로부터 클러스터 헤드까지의 거리는 비교적 가까우므로 자유공간 모델을 사용한다.  $Y_j$ 는 식 (6)으로 나타내어진다.

$$Y_j = \sum_{i \in I_j} Y_{ij} = qn_j \sum_{i \in I_j} (E_{elec} + P_{fs}r_{ij}^2) \quad (6)$$

여기서  $I_j$ 는 클러스터 j에 할당된 센서노드의 집합을 말하며 식(6)에서는  $n_j$ 개가 할당된다는 것을 의미한다. 클러스터 헤드 위치문제의 해를 구하기 위해 네트워크 전체에서 소비되는 에너지를 최소화시키는 것보다 각각의 클러스터에서 소비되는 에너지를 균형 있게 만드는 것이 중요하다. 따라서 클러스터 헤드의 위치문제를 해결하기 위한 목적함수로 아래 식(7)을 생각할 수 있다.

$$\min \{ \max_{j \in RN} Y_j \} \quad (7)$$

식(7)은 임의의 클러스터에서 소비되는 센서노드의 에너지의 최대값을 최소화시킨다는 의미이다. 이 때 제약조건들은 다음 식들로 표현된다.

$$\sum_{j=1}^m x_{ij} = 1, \quad \forall i \in SN \quad (8)$$

$$x_{ij} \leq y_j, \quad \forall i \in SN, \forall j \in RN \quad (9)$$

$$\sum_{j=1}^m y_j = k \quad (10)$$

$$\sum_{j=1}^m x_{ij} \leq C_j, \quad \forall j \in RN \quad (11)$$

$$x_{ij}, y_j \in \{0,1\}, \quad \forall i \in SN, \forall j \in RN \quad (12)$$

제약조건 (8)은 센서노드 i가 단일 클러스터 헤드 j에 할당된다는 것을 말한다. 조건식 (9)는 센서노드가 후보 클러스터 헤드 중에서 하나의 클러스터 헤드에 할당된다는 말이다. 식(10)에서 k는 클러스터의 수를 의미한다. 기지국의 위치를 고려하지 않고 순수하게 센서노드의 에너지만을 고려하는 LEACH와 같은 클러스터 모델에서는 최적의 클러스터 개수를 사전에 알 수 있다. 식(11)에서  $C_j$ 는 계산 속도를 높이기 위한 것으로 직관적으로  $N/k$  보다 커야 한다. 여기서 N은 네트워크 전체의 센서노드 개수이다. 식 (10)과 (11)로 인해서 탐색공간을 크게 줄일 수 있으며 따라서 탐색속도를 높게 된다.

실험에 사용된 네트워크의 크기는 한 번이 100m 인 정사각형이고 100개의 센서노드 들이 무작위로 설치된다. 중계노드는 센서노드보다 훨씬 큰 초기에너지를 갖고 있으며 30개의 후보 위치를 네트워크 설계하기 전에 미리 정한다. 실험의 일관성을 위해서 LEACH의 연구에서 사용했던 것과 같은 크기의 네트워크와 같은 수의 센서노드, 그리고 같은 파라미터를 사용하였다. LEACH 논문에서 최적의 클러스터

수는 6개이며, 이 때 식 (7)의 목적함수를 최적으로 만족하는 중계노드의 위치를 찾아내는 것이 본 실험의 목적이다.

알고리즘의 구현은 ILOG OPL Studio 3.51을 사용하였으며 IBM PC에서 실행하였다. OPL 언어는 제약 프로그래밍에서 많이 사용되는 것으로 본 실험에서는 식 (7)의 값으로 4.965 mJ이 나왔으며 총 탐색노드 수는 11,661개이다.

#### 4.2 RFID 리더(Reader) 간섭 최소화 문제

FDMA를 사용하는 RFID 네트워크에서 사용 채널 수보다 많은 리더가 있을 때 밀집리더모드(Dense Reader Mode)라 하고 이 때 리더 사이에 전파간섭이 발생하게 된다[19]. 리더간의 전파간섭을 최소화하여 전체 시스템의 가독률(Read Rate)을 높이는 것이 리더 간섭 최소화 문제이다. 이것은 많이 연구되어지는 리더 영역에서의 다수 태그간 충돌문제[20]와는 다른 것이다. 이 문제는 제약 최적화 문제로 모델링될 수 있으며 최적화 목적함수는 리더에게 할당되는 채널 수를 최소화하는 것이다. 이 문제는 NP-hard 문제로서 주파수 할당문제이며 k-coloring 문제와 매우 유사하게 모델링 할 수 있다.

주파수 할당 문제는 전통적으로 Brelaz가 제안한 그래프 채색에 가장 효과적인 포화도(Saturation Degree) 알고리즘인 D<sub>sat</sub>을 사용하여 해를 구하였으며 이것은 동적 변수 순서화 방법에 해당된다. 즉, 이 문제를 D-type으로 보고 문제의 해를 푸는 알고리즘을 적용한 것으로 해석된다. 이것은 주파수 값들의 도메인이 가장 작은 변수를 찾아서 먼저 주파수를 할당하고 할당된 변수를 제거하고 계속 도메인이 작은 변수를 찾아 주파수를 할당한다. 도메인 크기란 리더에 할당할 수 있는 주파수의 개수를 말한다. 만약에 같은 크기의 변수가 나타나면 연결도 또는 차수(Connectivity or Degree)가 큰 변수를 선택한다. 연결도는 Freuder에 의해서 제안된 것으로서 이것은 탐색하기 전에 이미 변수의 순서가 정해지는 정적 변수 순서화 방법이다. 즉, 연결도는 변수를 선택함에 있어서 도메인 크기가 같을 경우에 차선책으로서 변수 선택의 판단 근거가 되는 것이다.

주파수 할당 문제를 문제분류규칙에 적용하면 탐색 가지의 순서에 따라 도메인의 크기와 연결도가 크게 변화하기 때문에 이 문제를 D-type으로 보지 않고 DS-type으로 간주한다. 따라서 변수 순서화 알고리즘도 도메인뿐만 아니라 연결도를 동시에 고려하는 알고리즘이 더 우수하다는 것을 예측할 수 있으므로 도메인의 변화뿐만 아니라 정점들의 연결도를 동시에 고려한 혼합(Hybrid) 알고리즘을 제안한다. 즉 도메인의 변화에 주목하는 동적 변수 순서화법과 연결인접도의 변화에 주목하는 정적 변수 순서화법을 동시에 고려한 wdom-deg 알고리즘을 제안한다. 여기서 dom은 변수들이 가질 수 있는 값들의 집합이고 deg은 변수들의 연결도를 뜻한다. 즉, 가지마다 크기가 달라지는 동적 요소인 dom과 정적요소인 deg를 복합적으로 사용하여 변수 순서화를 진행한다. w는 도메인과 연결도의 크기가 매우 다를 경우, 변수 선택에 있어서 한 쪽으로 치우치는 것을 방지하기 위한 것으로 실험적으로 얻어진 가중치이다.

실험을 위하여 제약만족 최적화 모델링을 하면 다음과 같다. 주파수 대역은 917-923.5 MHz로서 32개 채널로 나누고 각 채널의 대역폭은 200KHz이다. 안테나에서 출력전력은 10mW이하로 하고 LBT(Listen Before Talk)를 사용하지 않는다고 가정한다. 리더가 설치되는 장소를 100개의 영역으로 나누고 리더가 설치될 수 없는 지역조건으로 인해서 몇몇 장소에는 리더가 설치되지 않는다고 가정한다. 각 채널간 거리에 따른 간섭 제약조건을 아래와 같이 나타낸다.

1) 동일지역 제약(Co-Site Constraint, CSC) - 동일 셀 안의 채널들은 적어도 지정된 주파수 거리 이상을 유지하여야 한다. 동일 셀 안에서는 적어도 3 채널 이상을 유지하여야 간섭이 생기지 않는다.

$$|f_i - f_j| \geq 3$$

2) 인접 채널 제약(Adjacent Channel Constraint, ACC) - 서로 일정거리 안에 있는 두 송신기는 인접 채널을 사용할 수 없다. 인접 채널은 보통 1 - 2 채널만큼 떨어진 채널을 의미한다. 따라서 이 제약조건은 다음과 같은 식으로 표현된다.

$$|f_i - f_j| \geq c_{ij}$$

ACC는 n 정점의 제약 그래프에서 n x n 행렬로 나타내어지며, 간섭도 같은 제약 그래프로 나타내어진다. 여기서 i ≠ j이고 c<sub>ij</sub>는 각각의 송신기 쌍 (i,j) 에 대해서 필요한 최소 주파수 이격이다.

모델링을 단순하게 하기 위하여 서비스 영역을 총 500 m x 500 m 로 이루어진 정사각형 넓이의 영역으로 하고, 이것을 다시 50 m x 50 m 의 크기로 나누어 총 100개의 셀로 구성된 서비스 영역을 정하였다. 하나의 셀 안에는 리더가 0-2개까지 설치될 수 있고 무작위로 리더기를 설치한다. 5개의 인스턴스로 실험하고 리더기는 평균 총 67개로 실험하였다. 실험에서 RFID 리더기 출력은 10mW 로 정하였다. 알고리즘의 구현은 ILOG OPL Studio 3.51을 사용하였으며 IBM PC에서 실행하였다. 본 실험에서는 필요 채널수가 24 개, 총 탐색노드 수는 56,859개이다.

### 5. 평 가

본 장에서는 4장의 실험에 대하여 적용된 알고리즘을 평가한다. 본 논문에서 제안한 백트래킹 탐색의 구조화는 제약만족 최적화 문제에 대한 일반적인 백트래킹 알고리즘은 매우 구하기 어렵다는 가정에서 출발하였다. 따라서 본 논문에서는 문제를 결정변수의 속성에 따라서 4가지로 세분화시키고 세분화된 문제의 특성에 가장 잘 맞는 백트래킹 알고리즘을 개발하는 것으로 귀결된다. 또한 문제를 세분화시키는 기준은 결정변수의 정적 및 동적 특성에 따르므로 결정변수의 순서가 어떻게 되는가 하는 문제 즉, 변수 순서화 알고리즘이 문제의 해를 구하는데 매우 중요한 사항이 된다.

4.1절에서 실험한 무선 센서 네트워크의 클러스터 헤드 위치문제는 4가지 세분화된 문제 중에서 S-type에 해당한다. 따라서 일반적인 백트래킹 알고리즘을 구하는 것보다



결정변수의 속성이 정적인 상태를 최대한 이용하는 SVO 알고리즘이 더 우수하다는 것은 예측할 수 있는 것이다. 식 (7)은 목적함수 값이 적어야 각 노드에서의 전력소비가 균등하다는 것을 의미하며 표 1은 그 결과이다. 탐색 노드 수 (Choice points)는 적을수록 알고리즘의 효율성이 높은 것이다. SB는 변수 순서화를 적용하지 않은 Simple Backtracking을 의미하며 DVO는 동적 변수 순서화 알고리즘을 나타내고 SVO는 정적 변수 순서화 알고리즘을 나타낸다.

<표 1> 변수 순서화 알고리즘의 성능비교

| 변수순서화 알고리즘 | 식(7)의 값(mJ) | 총 탐색노드 수 | 실행시간 (초) |
|------------|-------------|----------|----------|
| SB         | 5,457       | 66,680   | 63.75    |
| DVO        | 4,961       | 30,564   | 28.03    |
| SVO        | 4,955       | 11,661   | 9.11     |

이 문제는 문제형태가 정적문제(S-type)이므로 SVO 알고리즘이 가장 우수하리라 예측할 수 있었는데 실제 실험에 있어서 <표 1>에서와 같이 SVO 알고리즘이 가장 우수함을 알 수 있었다. 이와 같이 문제의 해를 구하는 일반적인 백트래킹 알고리즘을 구하는 것은 변수의 속성을 고려하지 않으므로 각각의 문제에 대해서 최적해가 아닐 수 있다.

4.2절에서 실험한 RFID 리더 간섭 최소화 문제는 주파수 할당문제와 유사하게 모델링하였으므로 4가지 세분화된 문제 중에서 DS-type에 해당한다. 따라서 일반적인 백트래킹 알고리즘을 구하는 것보다 결정변수의 속성이 동적 및 정적 상태가 혼합된 DSVO 알고리즘이 우수하다는 것은 예측할 수 있다. 다음 <표 2>는 여러 가지 변수 순서화 알고리즘의 성능을 비교한 것이다.

<표 2> 변수 순서화 알고리즘의 성능비교

| 변수순서화 알고리즘 | 필요 채널수(평균) | 총 탐색노드 수 | 실행시간 (초) |
|------------|------------|----------|----------|
| SB         | 29.8       | 8,575    | 2.08     |
| dom        | 25.2       | 136,119  | 25.2     |
| Dsatur     | 24.0       | 55,859   | 1.89     |
| wdom - deg | 21.8       | 79,027   | 2.45     |

여기서 SB는 변수 순서화 및 값 순서화를 하지 않고 단순 백트래킹을 사용하였을 경우이다. DSVO 알고리즘에도 여러 가지가 있으나 기존 Dsatur 알고리즘보다 더 우수한 wdom - deg 알고리즘을 제안하였고 표 2에서 그 결과를 보였다. 표 2에서는 필요 채널수에서 wdom - deg 알고리즘이 더 우수함을 알 수 있었지만 탐색노드 수와 실행시간에서는 기존 Dsatur 알고리즘이 우수한 것처럼 보인다. 하지만 필요 채널수에서 최적화되지 않은 것이기 때문에 의미가 없다.

dom 알고리즘은 3.2절에서 기술한 바와 같이 "smallest domain first"라고 불리는데 결정변수의 도메인 크기가 가장 작은 변수를 먼저 탐색하는 것이다. deg 알고리즘은 제약 그래프에서 변수의 연결도 또는 차수가 가장 큰 것부터 차례로 탐색하는 것이다.

Sp-type 문제는 S-type 문제의 부분집합으로 볼 수 있다. 이 문제는 4.1절 실험에서 목적함수 식 (7)을 아래와 같은 식 (13)과 같이 변경하면 얻을 수 있다.

$$\min \{ \max_{j \in RN} X_j + \max_{j \in RN} Y_j \} \quad (13)$$

여기서  $X_j$ 는 식 (4)로 표현된다. 즉, 클러스터 헤드에서 소비되는 에너지 양으로서 일반 센서 노드에서 소비되는 에너지 양보다 훨씬 크기 때문에 클러스터 헤드는 서비스 영역에서 기지국 근처로 모이게 되는 선호성으로 동작하게 된다. 그러므로 이 문제는 기본적으로 S-type의 부분집합으로 생각할 수 있으며 특정 변수에 가중치가 부여된 SVO 알고리즘으로 해결될 수 있음을 예측할 수 있다.

D-type의 문제는 일반적으로 찾기 힘들다. 2장에서 기술한 관련연구들은 문제를 분류하지 않은 상태에서 일반적인 백트래킹 알고리즘을 개발한 것인데 이들의 알고리즘은 DVO라고 분류되어도 실제 저자의 분류에 의하면 DS-type으로 분류되기 때문이다. 즉 DVO보다는 DSVO 알고리즘이 더 좋은 성능을 나타내었다. DS-type이 아니고 순수한 D-type 문제는 현실세계에서 존재하지 않을지도 모른다.

## 6. 결론

제약만족 최적화 문제의 해를 구하는 일반적인 백트래킹 알고리즘은 구하기 어렵다. 따라서 결정 변수의 속성에 기반을 두어 모든 문제를 4가지 형태로 세분화하고 이 분류된 형태에 따라 각각의 특성화된 알고리즘을 적용하였다. 결국, 무선 센서 네트워크의 클러스터링 및 RFID 리더의 간섭 최소화 같은 제약만족 최적화 문제들의 백트래킹 알고리즘을 구조적으로 효율성 있게 개발하였다.

무선 센서 네트워크의 클러스터링 문제는 정적 문제인 S-type으로서 그에 적당한 SVO 백트래킹 알고리즘이 가장 우수한 성능을 보였다. 또한 RFID 문제는 동적 및 정적 문제가 혼합된 DS-type 문제로서 그에 가장 적당한 wdom - deg 알고리즘을 개발하고 적용시킴으로써 타 알고리즘보다 성능이 우수함을 입증하였다. 따라서 모든 제약만족 최적화 문제의 해를 구하기 위해 일반적인 알고리즘을 개발하는 것보다 문제를 세분화하고 다시 이에 적합한 백트래킹 알고리즘을 구하는 것이 더 쉽고 성능 또한 우수함을 알 수 있다. 추후 연구로서 문제분류규칙을 보완하여 형태 분류가 쉽지 않은 문제들도 다룰 수 있는 보다 진보된 문제 분류규칙이 요구된다.

## 참 고 문 헌

- [1] M. Ercsey-Ravasz, T. Roska, and Z. Neda, "Cellular neural networks for NP-hard optimization," 11th Int'l Workshop on Cellular Neural Networks and Their Applications, pp.52-56, Jul., 2008.
- [2] P. Hell and J. Nešetřil, "Colouring, constraint satisfaction, and complexity," Computer Science Review, Vol.2, Issue3, pp.143-163, Dec., 2008.
- [3] F. Rossi, P. van Beek, and T. Walsh (Eds.), Handbook of Constraint Programming, Elsevier, 2006.
- [4] Z. Yujun, X. Jinyun, and S. Haihe, "A category theoretic approach to search algorithms: Towards a unified implementation for branch-and-bound and backtracking," 4th Int'l Conference on Computer Science & Education, pp.845-850, Jul., 2009.
- [5] S. Prestwich, "Local search and backtracking vs non-systematic backtracking," AAAI 2001 Fall Symposium on Using Uncertainty within Computation, 2001.
- [6] H. Terashima-Marin, J.C. Ortiz-Bayliss, P. Ross, and M. Valenzuela-Rendon, "Hyper-heuristics for the dynamic variable ordering in constraint satisfaction problems," In proceedings of the 10th annual conference on genetic and evolutionary computation, pp.571-578, ACM, 2008.
- [7] I. P. Gent, E. MacIntyre, P. Prosser, B. M. Smith, and T. Walsh, "An empirical study of dynamic variable ordering heuristics for the constraint satisfaction problem," in Principles and Practice of Constraint Programming, pp.179 - 193, 1996.
- [8] S. W. Golomb and L. D. Baumert, "Backtrack programming," Journal of the ACM, Vol.12, No.4, pp.516-524, 1965.
- [9] R. M. Haralick and G. L. Elliott, "Increasing tree search efficiency for constraint satisfaction problems," Artificial Intelligence, Vol.14, No.3, pp.263 - 313, 1980.
- [10] J. Patel, J.W. Chinneck, "Active-constraint variable ordering for faster feasibility of mixed integer linear programs," Mathematical Programming, Vol.110 Issue3, pp.445-474 Sep., 2007.
- [11] F. Boussemart, F. Hemery, C. Lecoutre, and L. Sais, "Boosting systematic search by weighting constraints." In Proc. 16th European Conference on Artificial Intelligence-ECAI'04, pp.146-150, IOS, 2004.
- [12] E. C. Freuder, "A sufficient condition for backtrack-free search," Journal of ACM, Vol.29, No.1, pp.24 - 32, 1982.
- [13] D. Brelaz, "New methods to color the vertices of a graph," Communication of the ACM, Vol.22, No.4, pp.251 - 256, 1979.
- [14] B. M. Smith, "The Brelaz heuristic and optimal static orderings." in CP series, Lecture Notes in Computer Science, J. Jaffar, Ed., Vol.1713. Springer, pp.405 - 418, 1999.
- [15] C. Bessiere and J.-C. Regin, "MAC and combined heuristics: Two reasons to forsake FC (and CBJ?) on hard problems," in CP series. Lecture Notes in Computer Science, E. C. Freuder, Ed., Vol.1118. Springer, pp.61 - 75, 1996.
- [16] R.J. Wallace, "Determining the principles underlying performance variation in csp heuristics," International Journal on Artificial Intelligence Tools, Vol.17 Issue5, pp.857-880, Oct., 2008,
- [17] B. Hnich, T. Walsh, and B. M. Smith, "Dual modelling of permutation and injection problems," Journal of Artificial Intelligence Research, Vol.21, pp.357 - 391, 2004.
- [18] W.B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks", IEEE Transactions on Wireless Communications, 1(4), pp.660 - 670, 2002.
- [19] S. Zhou, Z. Luo, E. Wong, C.J. Tan, and J. Luo, "Interconnected RFID Reader Collision Model and its Application in Reader Anti-collision," 2007 IEEE International Conference on RFID, pp.212-219, 2007.
- [20] 김원태, 안광선, 이성준, "RFID 시스템에서 비트 변화 감지를 이용한 가변 슬롯 트리 기반 충돌 방지 알고리즘," 정보처리학회논문지A, 제16-A권, 제4호, pp.289-298, 2009. 8.

## 손 석 원



email : sohn@hoseo.edu

1985년 인하대학교 전자공학과(공학사)

1987년 인하대학교 전자공학과(공학석사)

2007년 인하대학교 컴퓨터정보공학과

(공학박사)

1987년~1992년 한국원자력연구소

1999년~현 재 호서대학교 벤처전문대학원 부교수

관심분야 : 무선통신망, 센서네트워크, 인공지능