

## 다중무인운반차 시스템의 새로운 동적경로계획 알고리즘 : 비정지우선 우회알고리즘

### A New Dynamic Routing Algorithm for Multiple AGV Systems : Nonstop Preferential Detour Algorithm

신 성 영, 조 광 현

(Sung-Young Shin and Kwang-Hyun Cho)

**Abstract :** We present a new dynamic routing scheme for multiple autonomous guided vehicles (AGVs) systems. There have been so many results concerned with scheduling and routing of multiple AGV systems; however, most of them are only applicable to systems with a small number of AGVs under a low degree of concurrency. With an increased number of AGVs in recent applications, these AGV systems are faced with another problem that has never been occurred in a system with a small number AGVs. This is the stop propagation problem. That is, if a leading AGV stops then all the following AGVs must stop to avoid any collision. In order to resolve this problem, we propose a nonstop preferential detour (NPD) algorithm which is a new dynamic routing scheme employing an election algorithm. For real time computation, we introduce two stage control scheme and propose a new path searching scheme,  $k$ -via shortest path scheme for an efficient dynamic routing algorithm. Finally, the proposed new dynamic routing scheme is illustrated by an example.

**Keywords :** Automated Guided Vehicle (AGV), dynamic routing,  $k$ -via shortest path

#### I. 서론

AGVs(Automated Guided Vehicles)는 물류운송시스템(material handling systems), 유연생산시스템(flexible manufacturing systems), 그리고 자동항만(automated transportation port) 시스템 등의 응용에서 일반화되고 있다[8], [10], [11]. 지난 십 여년 동안 AGV 시스템에 관한 많은 연구가 진행되어 왔고, 팔목할만한 기술적 진보를 이룩하였다. 그 중에서 현재까지 AGV 시스템에서 가장 주목을 받아오는 기술 중에 하나가 바로 AGV의 스케줄링과 라우팅에 관한 기술이다[6], [5], [12], [8]. 이와 같이 스케줄링과 라우팅에 관한 많은 알고리즘이 제안되어 왔으나, 대부분 매우 적은 수의 AGV 시스템이거나, 낮은 수준의 동시성을 제공하는 시스템에 적용할 수 있을 뿐이다. 오늘날의 AGV 시스템은 컨테이너 터미널과 같이 많은 수의 AGV가 상호협동해야 한다는 관점에서 효율적인 스케줄링과 라우팅 기법의 개발이 그 어느 때 보다 중요하게 되었다. 특히, 운행 중인 한 대의 AGV가 정지함으로써 인해 뒤따르는 모든 AGV가 정지해야 하는 상황은 소수의 AGV 시스템에서는 고려되지 않은 새로운 문제로 대두되고 있다. 또한 AGV, 생산기계, 운영자(operators) 등과 같은 시스템 요소가 점차 복잡해지고 다양한 기능을 수행해야 하는 오늘날의 산업시스템은 구조적으로 예기치 않은 시스템의 돌발상황이 발생할 가능성이 매우 높다. 하지만 대부분의 AGV 경로 계획기법은 시스템의 환경에 유연하게 대응하지 못하는 단점을 가진다. 특히, 스케줄링기법은 선행한 모든 AGV가 주어진 시간에 고정된 궤도

를 따라 주행한다고 가정하고 AGV의 급송 계획을 수립함으로써 돌발상황으로 인한 주행경로의 변경은 시스템 전체에 치명적 혼란을 야기시킬 수 있다. 이러한 문제를 해결하기 위해 무엇보다 시스템의 변화에 강인한(robust) 알고리즘을 개발하는 것이 중요하다. 이러한 관점으로부터 본 논문에서 제안하는 트래픽제어 알고리즘은 다음과 같은 문제 해결을 목적으로 한다.

첫째, 시스템의 상태변화에 따라 동적으로 경로를 계획하는 동적경로 계획법을 제안한다. 둘째, 선행 AGV의 정지로 인해 뒤따르는 모든 AGV가 정지하는 연속정지를 방지한다. 셋째, 실시간 경로 탐색이 가능한 새로운 경로탐색알고리즘을 제안한다. 트래픽제어 알고리즘의 동적경로계획을 위해 본 논문에서는 증량적경로계획법을 도입한다[10]. 증량적경로계획이란 스케줄기반의 급송과 대조적으로 AGV가 급송 전에 시작노드에서 목적노드까지의 경로가 계획되는 것이 아니라 현재노드에서 목적노드에 도달하기 위한 일부경로만을 계획하는 방법으로 시스템의 상태변화에 융통성있게 대처할 수 있다. 이러한 증량적경로계획을 기반으로 비정지우회경로를 계획하기 위해서는 다음 부분경로가 우회가능한노드에서 결정되어야한다. 이러한 증량적경로계획의 실시간성을 보장하기 위해 온라인 연산의 일부를 오프라인에서 수행하여 온라인상에서의 계산량을 경감시키는 이단계기법을 도입한다. 트래픽제어 알고리즘 일명 비정지우선 우회알고리즘은 오프라인에서 경로탐색을 위한  $k$ -경유최단경로알고리즘과 온라인상에서 상호배타적 문제 해결위한 투표알고리즘으로 이루어진다.

지금까지 AGV시스템에 관한 관련된 연구를 살펴보면 다음과 같다. 비충돌 최단시간 AGV 경로에 대한 개념은 1985년 Broadbent[1]에 의해 처음으로 소개되었으며, AGV의 경로

논문접수 : 2001. 12. 10., 채택확정 : 2002. 6. 5.

신성영 : 울산대학교전기전자정보시스템공학과(ssy@sys.ulsan.ac.kr)

조광현 : 울산대학교전기전자정보시스템공학과(ckh@sys.ulsan.ac.kr)

※ 이 논문은 2001년도 한국학술진흥재단의 지원에 의하여 연구가 되었음. (KRF-2001-041-E00279)

점유시간(path occupation time)을 기술하는 행렬을 생성하기 위해 Dijkstra의 최단경로 알고리즘을 사용하였다. 알고리즘의 복잡도는 경로네트워크의 노드수를  $N$ 이라 할 때  $O(N^2)$ 이다. Egbelu 등[3]은 양방향경로 AGV 시스템에서 AGV 대비 생산성 면에서 주목할 만한 향상을 보여 주었다. 그러나 양방향 경로 AGV 시스템의 제어는 공유 경로 구역에 대한 다수 운반차의 경쟁으로 인해 다소 복잡해진다. Haung[4]은 허용시간 창(free time windows) 안에서 경로분선들을 공유하기 위해 라벨링 알고리즘(labeling algorithm)을 제안하였다. 이 알고리즘은 물리적 경로네트워크의 링크를 두 노드를 연결하는 두개의 링크를 가진 노드로 변환한 뒤 각 노드에 대하여 정의된 허용시간 창에 라벨을 할당하고, 모든 노드의 라벨을 비교함으로써 최단경로를 구하는 것이다. 이 알고리즘의 복잡도는 전환된 네트워크의 모든 시간 창(time windows)의 수를  $D$ 라 할 때  $O(D^2 \log D)$ 이다. Kim 등[5]은 양방향 경로 네트워크에서 AGV 경로계획을 위한 비충돌 최소시간 알고리즘(conflict free shortest time algorithm)을 제안했다. 이 알고리즘에서 시간창 그래프(time window graph)의 개념이 처음 소개되었다. 알고리즘의 복잡도는 운반차의 대수를  $V$ , 노드 수를  $N$ 이라 할 때  $O(V^4 N^2)$ 이다. Taghaboni 등[10]은 상대적으로 빠르게 AGV의 경로를 계획할 수 있는 증량경로계획기법(incremental route planning scheme)으로 불리는 동적경로계획기법을 제안하였다. Lee 등[6]은 비충돌 최단경로 탐색을 위해 이단계 기법을 도입하였으며, AGV의 잠재충돌을 탐지하기 위해 링크점유시간표(link occupation table)를 이용하였다. 알고리즘은 오프라에서  $K$ 개의 최단경로를 탐색하는 KSFP(K-Shortest Feasible Path) 알고리즘과 온라인에서 최적경로를 생성하는 MP(Motion Planning) 알고리즘으로 대별된다. 알고리즘의 복잡도는 오프라인 연산 과정인 KSFP 알고리즘은 네트워크의 노드 수를  $N$ 이라 할 때  $O(KN^3)$ 이다. 온라인 연산 연산과정인 MP 알고리즘의 복잡도는 운반차의 대수를  $K$ 라 하고 경로의 노드 수를  $S$ 라 할 때  $O(KVS^2)$ 이다.

논문의 구성은 먼저 2 장에서는 경로네트워크 모델과 다중무인운반차 시스템의 제어기 구조에 대하여 기술하며, 3 장에서는 트래픽제어 알고리즘에 대하여 기술하고, 4 장에서는 AGV의 동적경로계획의 예를 설명한다. 그리고 마지막으로 5 장에서는 결론으로 본 논문을 마무리한다.

**II. 경로네트워크 모델과 다중무인운반차 시스템의 제어기 구조**

**1. 경로네트워크 모델**

AGV 시스템의 모델은 크게 AGV가 이동하는 물리적 공간을 나타내는 경로네트워크 모델과 AGV 자체를 나타내는 모델이 있다. 편이상 AGV의 모델은 경로네트워크에서 점으로 나타내기도 한다. AGV 시스템의 물리적 공간은 AGV의 이동을 감지하는 센서[일종의 트랜스폰더 (transponder)]와 유도선(guide line) 등으로 구성되고 두 제어점 사이의 경로를 경로분선(path segment)이라 한다. 경로네트워크 모델은 다수개의 노드(node)와 링크(link)로 구성된 그래프의 특수한 형태이다. AGV 시스템의 경로분선은 네트워크에서 노드에의해

경로분선들 사이의 관계는 방향성 링크(directed link)에 의해 표시된다. AGV는 레도상 매설된 지정된 제어점을 통과할 때만 그 위치를 알 수 있는 것이 보통이므로 일반적으로 한 순간에 한대의 AGV만이 주어진 레도에 진입할 수 있게끔 제어하는 구역제한기법이 사용된다[4]. 경로네트워크 모델에서 노드는 AGV가 점유할 수 있는 하나의 경로분선(path segment)을 나타내고, 링크는 노드상호간의 연결관계를 나타낸다. 노드(node)  $p$ 로부터 인접노드  $q$ 로 연결된 링크를  $(p,q)$ 로 표기한다. 이 링크  $(p,q)$ 에 대하여  $p$ 는 '노드  $q$ 의 입력노드(input node)'라고 정의하고, 노드  $q$ 는 '노드  $p$ 의 진출노드(output node)'라고 정의한다. 또한 링크  $(p,q)$ 는 노드의 쌍  $p$ 와  $q$ 에 대하여 유일하다고 가정한다. 노드  $p$ 의 인접한 노드의 집합은  $Adj(p)$ 로 나타낸다. 노드는 네트워크의 물리적 구조에 따라, 오직 하나의 입력노드(input node)와 하나의 진출노드(output node)를 가지는 일반노드(ordinary node), 두개이상의 입력노드와 하나의 진출노드를 가지는 병합노드(merging node), 두개이상의 입력노드와 두개 이상의 진출노드를 가지는 교차노드(crossing), 하나의 입력노드와 두개이상의 진출노드를 가지는 분기노드(branch node) 등으로 구분한다. 그리고 교차노드는 병합노드와 분기노드의 합성구조로 트래픽제어 관점에서 병합노드로 혹은 분기노드로 분류된다. 경로네트워크의 일반적 가정들은 다음과 같다.

- 1) 경로네트워크에서 경로분선들의 길이는 일정하다.
- 2) 경로의 방향은 단방향이다.
- 3) AGV는 오직 하나의 경로분선에 위치한다.

이러한 가정을 바탕으로 경로네트워크에서 충돌은 병합노드와 교차노드에서 발생하게 된다.

**2. 다중무인운반차 시스템의 제어기 구조**

본 논문에서 제안하는 비정지우선 우회알고리즘은 그림 1과 같은 제어시스템을 기반으로 동작한다. 제어시스템은 운동제어기(motion controller), 트래픽제어기(traffic controller), 관리제어기(supervisor), 그리고 경로테이블(routing table)로 구성된다. 운동제어기는 트래픽제어기와 양방향 통신이 가능하며 경로네트워크상에서의 현재위치와 주어진 부분경로를 따라 주행하며, AGV 고장상태를 경로테이블생성기(routing table generator)로 전송한다. 경로테이블생성기는 AGV 고장 등과 같은 시스템 환경이 변화하면 경로테이블을 갱신한다.

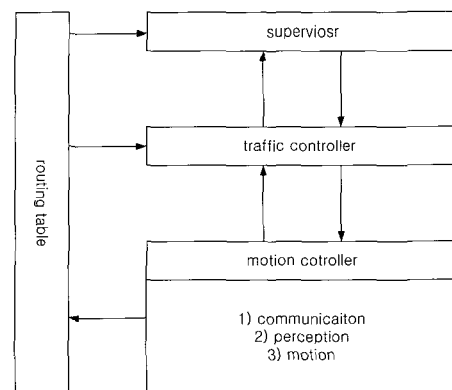


그림 1. 다중무인운반차 시스템의 제어기 구성도.  
Fig. 1. The control scheme of multiple AGV systems.

제어시스템에서 운동제어기는 일반적으로 제공되는 AGV 동작제어 알고리즘을 이용할 수 있으며, 트래픽 제어기는 다음 3 장에서 설명된 셀 기반의 투표알고리즘을 기반으로 동작한다. 관리제어기는 AGV 가 다수의 셀에서 동시에 진입이 허용된 경우 목적지까지 최단경로계획이 가능한 셀을 선택하는 기능을 수행한다. AGV 시스템에서 AGV 에 대한 일반적 가정들은 다음과 같다.

- 1) 경로네트워크에서 AGV 는 등속주행을 한다.
- 2) AGV 는 다른 AGV 의 위치는 알지 못하고 트래픽제어기와의 통신에 의해 주행경로가 결정된다.

### III. 트래픽제어 알고리즘

본 논문에서 제안하는 비정지우선 우회알고리즘은 증량적경로계획법으로 분기노드에서 다음분기노드까지의 부분경로를 계획하는 방법으로 AGV 가 급송된 후 전체경로를 부분적으로 계획해 나간다. 분기노드에서 다음 분기노드를 선택하는 방법으로는 생성될 부분경로에서의 트래픽 상황을 고려하여 목적지까지의 최단경로가 되는 노드를 선택해야 한다.

이 절에서는 k-경유최단경로 기법을 이용하여 경로테이블을 생성하는 방법과 경로네트워크에서 AGV 의 연속흐름을 보장할 트래픽제어기법과 알고리즘에 대하여 알아본다.

#### 1. k-경유최단경로 탐색알고리즘

AGV 경로계획법은 이미 주행 혹은 계획이 이미 완료된 경로의 의존 여부에 따라 정적경로계획과 동적경로계획으로 나눌 수 있다. 먼저 정적 경로계획은 일반적인 경로계획법으로서 경로탐색을 수행하는 라우팅과 충돌 회피를 위한 스케줄링으로 구성된다. 동적경로계획법은 정적경로 계획법과는 달리 충돌회피를 위한 스케줄링 기능을 별도로 수행하지 않고 시스템의 주어진 상태에 따라 경로를 선택한다. AGV 시스템에서 동적경로계획을 수행해야 하는 이유는 네트워크의 규모와 AGV의 대수가 증가하게 되면 계산량의 폭주로 인해 모든 AGV들을 스케줄링하는 것이 불가능해 질 수 있으며, 시스템과 시스템의 요소 등에 의한 돌발사태로 AGV 스케줄이 흐트러질 경우 모든 AGV를 다시 스케줄링해야 하는 극단적인 문제가 발생할 수 있기 때문이다. 그러므로 다수 AGV 시스템의 경우 동적경로계획을 수행하는 것이 타당하다.

AGV 시스템의 동적경로계획은 고도의 실시간성을 요구하는 기법으로 기존의 온라인 경로탐색알고리즘으로는 계산량의 문제를 해결할 수 없다. 이러한 계산량의 문제를 해결하기 위해 이단계기법의 도입이 필요하다. 이단계기법에서 일반적으로 사용되는 경로탐색기법은 다수최단경로 탐색기법으로 각 노드는 목적노드까지 최대 K 개의 최단경로를 가진다[6]. 하지만 증량적경로계획법에서 분기노드의 AGV 가 부분경로를 계획할 때, 요구되는 정보는 목적지까지의 k 개의 최단경로가 아니라 이웃노드를 경유해 목적지에 도달할 수 있는 최단경로의 리스트이다.

본 논문에서는 이러한 목적에 맞게 분기노드에서 이웃노드를 경유해 목적노드까지의 경유최단경로를 탐색할 수 있는 k-경유최단경로탐색기법을 제안하였다. 또한 경유최단 경

로테이블은 k-경유최단경로 탐색트리 생성함으로써 작성된다. k-경유최단경로탐색알고리즘은 초기화, 릴렉세이션(relaxation), 탐색알고리즘으로 구성된다.

다음의 의사코드(pseudo code)는 k-경유최단경로 탐색트리 알고리즘을 초기화하기 위한 것이다.

```
Initialize(G,v)
1. for each node n∈N[G]
2. do d[n]:=∞
3. π[n]:=NIL
4. d[v]=w(s,v)
5. π[n]:=s
```

여기서  $G=(N, L)$ 는 경로네트워크를 나타내고, N 은 노드의 집합을, L 은 방향성 링크의 집합을 나타낸다. v 는 임의의 노드를, s 는 탐색시점에 정해진 시작노드를 나타내고, d[v]는 시작노드에서 노드 v까지의 추정상의 경로가중치의 합을, π[v]는 노드 v의 선행노드를 각각 나타낸다. w(s,v)는 링크 (s,v)의 가중치(weight)를 나타낸다. 초기화는 주어진 노드의 이웃노드에 대하여 선행경로와 가중치를 명시하며, 주어진 시작노드의 이웃노드의 개수만큼 수행된다. 초기화의 연산시간은 경로네트워크의 모든 노드에 대하여 일회씩 수행되므로 계산량은  $O(N)$ 이다.

릴렉세이션은 링크(u,m)에 대하여 노드 u 를 통하여 노드 m 에 도달할 수 있는 최단경로를 개선할 수 있는지를 평가하고 그렇다면 d[v]와 π[v]를 갱신하는 것이다. 다음은 릴렉세이션을 위한 의사코드이다.

```
Relax(u,m,w)
1. if d[v]>d[u]+w(u,v)
2. then d[v]:=d[u]+w(u,m)
3. π[v]:=u
```

마지막으로 k-경유최단경로 탐색알고리즘에 대한 의사코드가 다음과 같다.

```
k-via_shortest_path(G, w, s)
1. for each v∈Adj(s)
2. do Initialize(G,v)
3. S:=∅ // initialize set S
4. Q:=N[G] // initialize the Min_Priority Q
5. while Q≠∅ // invariant Q=N-S
6. do u:=Extract-Min(Q)
7. S:=S∪{u}
8. for each node v∈Adj(u)
9. do Relax(u,v,w)
```

알고리즘의 단계 1-2 는 시작노드의 이웃노드에 대하여 초기화를 수행하고 초기화가 완료되면 시작노드와 이웃노드들까지의 부분트리가 완성된다. 단계 3 은 임의의 노드 집합 S 를 ∅로 초기화하고, 단계 4 에서는 경로네트워크의 노드집합을 우선순위대기열(priority queue) Q 에 삽입한다. 단계 5-9 에서 while 루프를 통하여 노드 u 는  $Q=V-S$  로부터 추출되고 집합 S 에 더해진다. 그리고 while 루프는 경로네트워크의 노드 수 즉, |N|번 반복된다. 알고리즘은 우선순위 대기열

(priority queue) Q에 대하여 다음과 같은 불변식  $Q=N-S$  을 가진다.

전체알고리즘의 계산량 분석: 알고리즘은 다음의 3 가지 우선순위대기열(priority queue) 연산을 수행함으로써 대기열(queue)의 최소값-우선순위를 유지한다: 경로네트워크의 노드 집합을 우선순위대기열에 삽입[insertion(단계 4)], 우선순위 대기열에서 최소값 추출[extraxt\_min (단계 6)], 대기열의 데이터를 지시하는 키값 감소[decrease key(단계 8)]. 삽입과 최소값 추출 연산은 노드당 일회 호출된다. 인접 리스트(adjacent list) Adj(v)에 있는 각각의 링크는 단계 8-9 의 for 루프동안 일회 평가된다. 인접리스트(adjacent list)에 있는 모든 링크의 수는  $|I|$ 이므로 for 루프동안 전체 키값 감소연산은  $|I|$ 회 반복된다. 알고리즘의 연산시간은 우선순위 대기열을 어떻게 구현하느냐에 따라 달라진다. 구현방식에는 노드에 번호를 부여하는 방식(node numbering)과 피보나치 힙(Fibonacci heap)이 있다. 노드에 번호를 부여하는 방식으로 우선순위 대기열을 구현할 경우 최소값 추출 연산시간은  $O(N)$ 이다. 전체알고리즘의 연산시간은 시작노드의 인접노드 Adj(s)에 대하여 인접노드의 수를  $k$ 라 할 때  $k$ 회 반복되므로  $O(kN^2+kN)=O(kN^2)$ 이다. 만약 피보나치 힙을 이용하여 우선순위 대기열을 구현하였을 경우 연산시간은  $O(kMogN+kN)=O(kMogN)$ 가 된다.

그림 2는 경로네트워크의 예를 나타내고 그림 3은 생성된 경유최단경로 트리를 도시한 것이다. 그림 2에서 원은 노드를, 원과 원을 잇는 방향성 직선은 링크를 나타낸다. 그리고 노드에 부여된 번호는 노드의 번호를, 링크에 번호는 노드사이의 가중치를 나타낸다. 분기노드 1에 이웃하는 노드는 {2, 3, 4}이고 경유최단경로알고리즘은 이들을 경유해 목적노드에 도달할 수 있는 3개의 최단경로를 계산한다. 그림 3은 이러한 방법으로 계산된 k-경유최단경로 트리이다. 그림 3에

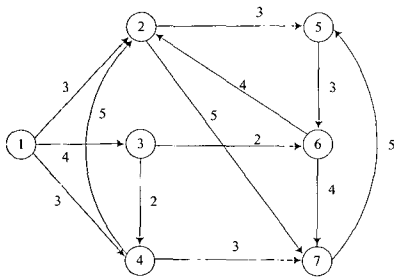


그림 2. k-경유최단경로 탐색을 위한 예.  
Fig. 2. An example for k-via shortest path search.

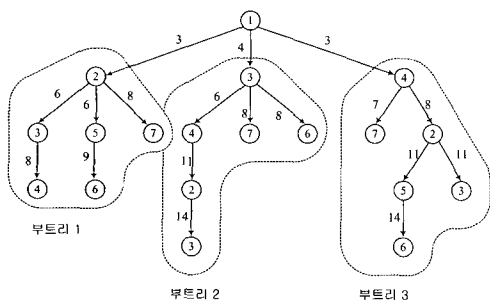


그림 3. k-경유최단경로에 의해 생성된 트리.  
Fig. 3. The k-via shortest path tree.

서 링크에 부여된 번호는 노드까지의 가중치이다. 3개의 부트리(sub-tree)는 부트리의 루트 노드를 경유노드로 선택했을 때 목적노드까지의 최단경로트리를 각각 나타낸 것이다. 예를 들어 부트리 1은 그림 2의 노드 1에서 이웃노드 2를 경유노드로 선택했을 경우 목적노드까지의 최단경로를 나타낸 것이다. 노드 4를 목적노드로 했을 때, 목적노드까지의 가중치는 8이다.

2. 상호배타적 구역: 셸

앞 절에서는 동적경로계획에 이용되는 경로테이블의 구성과 새로운 경로탐색 알고리즘에 대하여 기술하였다. 이 절에서는 이러한 탐색표를 이용하여 어떻게 AGV의 연속흐름을 보장할 수 있는 비정지우선 우회알고리즘을 생성할 것인가에 대하여 알아본다.

경로네트워크에서 최단경로는 분기노드에 계획되지만, AGV의 연속적 흐름을 보장하기위한 트래픽제어는 다수 AGV가 동시에 한점을 통과할 수 있는 병합노드관점에서 이루어져야 한다.

경로네트워크 모델은 기본적으로 구역제한 기법을 기반으로 하고 있으며, 모든 AGV가 동속주행을 한다고 가정하고 있다. 좁은 의미의 동속주행은 전체네트워크에서의 동속주행을 의미하며, 넓은 의미의 동속주행은 노드 진입속도와 노드 퇴출속도가 동속이라는 의미만을 포함한다. 여기서는 좁은 의미의 동속주행을 가정한다.

경로네트워크가 완전연결네트워크(fully connected network) 이라면 최소 하나 이상의 병합노드가 존재한다. 구역제한기법에서는 이러한 병합노드에서 물리적 충돌이 발생할 수 있다.

모든 AGV가 동속주행 한다고 가정하므로 병합노드에서 동일한 시간반경에 놓인 노드를 통과중인 AGV는 충돌을 회피할 수 없다. 즉, 우리는 이러한 상태를 AGV 시스템의 포괄적금지상태로 정의하고 경로네트워크 모델에서 상호배타적 구역의 집합으로 나타낸다. 이러한 상호배타적구역(mutual exclusive region)을 셸(shell)이라 정의하고 셸을 형성하는 교차 또는 병합노드를 셸의 중심이라 한다. 셸은 임의의 분기노드의 선행노드로 교차 또는 병합노드에 인접한 일반노드의 집합과 이들 노드의 집합에 인접한 일반노드의 집합으로 정의 된다. 예를 들어, 경로네트워크의 분기노드의 집합을 B라 할 때 병합노드  $n_j$ 에 인접한 일반노드의 집합을 첫번째 셸  $S(n_j,1)=\{n \in Adj(n_j) \setminus B\}$ 로 나타내고, 첫번째 셸에 인접한 일반노드의 집합을 두 번째 셸  $S(n_j,2)=\{n \in Adj(n_q) \setminus B, \forall n_q \in S(n_j,1)\}$ 로 나타낸다. 마찬가지로 k번째 셸은 k-1 번째 셸에 인접한 일반노드의 집합  $S(n_j,k)=\{n \in Adj(n_o) \setminus B, \forall n_o \in S(n_j,k-1)\}$ 으로 나타낸다. 교차 또는 병합노드를 셸의 중앙이라 하고  $S(n_j,0)$ 로 나타낸다.

그림 4는 병합 노드 1을 중심으로 하는 셸의 예를 나타낸다. 셸 중앙  $S(1,0)$ 은 노드 1이고 첫번째 셸  $S(1,1)=\{2, 3, 4, 5\}$ 이며 두번째, 세번째 셸은 각각  $S(1,2)=\{7, 8, 9\}$ ,  $S(1,3)=\{10, 11\}$ 이다.

트래픽제어 알고리즘의 목적은 다수의 AGV가 동시에 셸에 진입을 하는 것을 방지함으로써 AGV의 연속흐름을 보장하는 것이다.

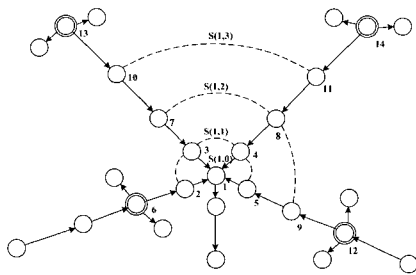


그림 4. 셸.  
Fig. 4. The shells.

다음 절에서는 하나의 셸에 다수의 AGV가 동시에 진입을 요구할 때 이중 한대만을 허용하고 나머지는 우회시키는 방법에 대하여 알아본다.

3. 상호배타적 구역: 셸

셸은 병합점을 중심으로 구성된 상호배타적 구역으로 정의되고, AGV의 연속흐름을 위해 한순간에 오직 한대의 AGV를 허용해야 한다. 그렇다면 다수대의 AGV가 동시에 하나의 셸에 진입을 요구할 경우 어떠한 것을 허용하고 우회시킬 것인가를 결정해야 한다. 이 절에서는 이러한 상호배타적 문제에 주로 이용되는 투표 알고리즘과 이를 AGV 시스템에 어떻게 적용할 것인가를 서술하고 이에 따른 구체적인 알고리즘을 기술한다.

투표알고리즘[17]은 분산시스템에서 특별한 작업을 수행하기 위하여 프로세스 집합에서 하나의 프로세스를 선택하는 일반적 형태의 연산방법이다. 일반적으로 투표알고리즘에는 두 단계의 절차가 있는데, 먼저 최고의 우선순위를 가지는 리더의 선택하고, 다음으로 다른 프로세스에 리더의 선출을 알리는 것이다. 이들 단계에서 시스템에 id를 분배하는 것이 필요하다. 본 논문에서는 목적노드까지의 최단경로를 가지는 AGV가 최고의 우선순위를 가지는 것을 트래픽 제어정책으로 정한다. 이러한 우선순위 정책은 전체 시스템 관점에서 다음과 같은 논리적 근거에서 효율적이다. AGV 시스템의 혼잡도를 AGV 대수의 함수로 정의하면, AGV의 대수가 증가하면 혼잡도가 증가하고, 혼잡도가 증가하면 충돌을 회피하기 위한 우회빈도수가 증가한다. 우회빈도수가 증가하면 목적노드까지의 경로가 길어지고, 개별 AGV에 대하여 목적노드까지의 경로가 길어진다는 것은 전체 시스템의 성능이 감소한다는 것이다. 그러므로 시스템의 성능을 향상시키기 위해서는 시스템의 혼잡도를 감소시켜야 하는 데, 시스템의 혼잡도를 줄이기 위한 제어정책으로 목적노드까지의 거리가 가장 적게 남은 AGV에 우선순위를 주어 현재 임무수행중인 AGV의 대수를 줄이는 것이다.

AGV 시스템에서 투표구는 셸에 진입을 요구하는 다수 AGV의 집합으로 구성되며, 투표참가자에게 부여되는 id는 현재노드에서 목적노드까지의 다음 경유노드를 경유했을 때의 경유최단경로이다. 이는 이미 만들어진 k-경유최단경로 테이블을 이용한다.

투표를 시작하기 전에 셸 입력노드의 AGV들은 투표를 위한 논리적 링을 형성하는데 노드에 진입한 시간 순서에 따라 번호가 부여된다. 투표를 시행하기 전에 다음과 같은 3단

계의 투표사전활동을 수행한다.

첫째, 각 AGV는 최소한 하나의 투표구역에 속하게 된다. 투표발의는 분기노드에서 하기 때문에 최소 두개 이상의 투표구역에 속하게 된다. 둘째, 소속된 투표구에서 다른 AGV에 의해 투표 발의가 발생하면 투표에 참여하게 된다. 셋째, 분기노드 진입 후 다음 분기노드를 결정하기 위해 AGV는 투표를 발의한다.

AGV 시스템을 위한 투표알고리즘을 기술하기 위해 다음과 같은 가정을 한다. 첫째, 트래픽 제어기는 셸의 입력노드에서 진행중인 AGV의 노드 진입시간, 목적노드, 진입속도 등에 관한 정보를 수신 받는다. 노드 진입시간은 투표구의 논리적 링을 형성하기 위해 이용된다. 둘째, 투표구에서 id의 값이 적을수록 높은 우선순위를 가진다. 셋째, 투표구의 논리적 링은 노드에 진입한 시간순으로 번호를 부여하고 AGV는 논리적 링에서 자신의 번호를 인식할 수 있다. 논리적 링의 이웃은 다음 순번의 AGV를 나타낸다. 즉 노드 i를 진행하는 AGV를 AGV(i)라고 할 때 AGV(i)의 이웃이웃은 AGV(i+1)이 된다. 분기노드에서 진행중인 AGV는 최소 한 개 이상의 투표구에 속하므로 한 개 이상의 투표구에서 선출될 수 있다. 트래픽 제어기는 투표구에서 선출된 결과를 관리제어기에 전송하고 관리제어기는 모든 트래픽 제어기로부터 전송되어온 결과를 평가한다. 만약, 한대의 AGV가 두개 이상의 셸에서 진입이 허용된다면, 그 중 최단경로를 가지는 것을 선택한다.

AGV 시스템에서의 투표알고리즘을 상세하게 기술하면 다음과 같다. 투표를 발의하는 AGV는 셸 진입을 위해 자신의 id와 투표를 위한 논리적 링 안에서의 자신의 번호가 포함된 메시지를 이웃AGV에게 전송한다. 이웃에서 투표메시지를 수신 받으면, 자신의 id와 비교한다. 만약 id의 값이 자신의 것보다 크다면, 자신을 투표참가자로 명시하고, 수신 받은 메시지와 자신의 메시지를 교체하고 이웃에게 전송한다. 만약 id의 값이 자신의 값보다 작다면, 자신을 투표참가자로 명시하고 받은 메시지를 이웃에게 전송한다. 만약 도착한 메시지의 id가 수신받은 AGV의 id와 동일하다고 flag가 자신의 것과 일치한다면, id의 값이 최소라는 것을 의미하고 수신받은 AGV가 투표에서 선출된다. AVG가 투표에서 선출되었으면, 이웃에게 선출되었다는 메시지를 전송한다. 만약 선출이 끝났다는 메시지를 받으면, 자신을 투표비참가자로 명시하고, 이웃에게 투표결과를 전송한다. 이러한 알고리즘을 의사코드로 기술하면 다음과 같다.

```

1. procedure AGV(i)
2.   if state=election then
3.     participant:=T
4.     send (election, i, id(i), flag(i)) to AGV((i+1) mod n)
5.   if state=receive(election) then
6.     if id(j) < id(i) then
7.       send (election, j, id(j), flag(j)) to AGV((i+1) mod n)
8.       participant:=T
9.     else if id(j) > id(i) ^ participant then
10.      null
11.     else if id(j) > id(i) ^ ~participant then
12.      send (election, i, id(i)) to AGV((i+1) mod n)
    
```

13. participant :=T
14. else if id(j) == id(i) ^ flag(j) == flag(i) then
15. send (elected, i) to P((i+1) mod n)
16. if state == receive (elected) then
17. winner :=j
18. participant :=F
19. if j ≠ i then
20. send(elected, j) to AGV((j+1) mod n)
21. end procedure AGV(i)

AGV 투표알고리즘의 계산량 분석: 계산량은 두가지 측면에서 분석된다. 투표에서 하나의 AGV를 선출하는데 걸리는 시간과 알고리즘에서 전송되는 메시지의 수이다. 투표에서 하나의 AGV를 선출하는데 소요되는 시간은 최대 메시지가 논리적 링을 두바퀴 되는 데 걸리는 시간이다. 알고리즘에서 하나의 AGV가 "elected" 메시지를 전송하는 수는 투표에 참여하는 AGV를 수를 n이라 할 때 n번이고, "election" 메시지를 전송하는 수는 다음 두 극단에 따라 달라진다. 첫째, id값이 최소인 AGV에 의해 투표가 발의된 경우 전송 메시지의 수는 O(n)이다. 둘째, id값을 기준으로 내림차순으로 논리적 링이 정렬되어 있고, 모든 AGV가 동시에 투표를 발의할 때 메시지의 수는  $\sum_{i=0}^{n-1} i+1 = \frac{1}{2}n \cdot (n+1)$  이다. 즉, O(n<sup>2</sup>)이다.

IV. AGV 동적경로 계획의 예

이 절에서는 그림 4의 AGV 시스템을 통하여 본 논문에서 제안한 비정지우선 우회알고리즘을 적용해본다. 그림 5는 전체 AGV 시스템의 일부분으로 증량적경로계획법에 따라 분기노드에서 다음 부분노드를 선택하는 과정을 설명하기 위해 제시되었다.

그림 5에서 상단 그리고 좌측에 부여된 번호는 AGV 시스템의 구역을 나타내기 위한 것이다. 예를 들어 lane 3, 5, 7이 분기하는 점은 <8,2>로 나타내고, 경로네트워크 모델의 노드가 된다. 각 lane위에 등 간격으로 그려진 점선은 AGV가 구역으로의 진출입을 감지하기위해 설치된 감지기를 나타내고 모든 제어기는 AGV가 감지기를 통과하는 순간에만 이동을 알 수 있다. 한편, 시스템에서 노드<a,b>를 점유하는 AGV는 AGV(<a,b>)로 나타낸다.

비정지우선 우회알고리즘은 오프라인에서 k-경유최단경로탐색기법을 이용하여 경로테이블을 생성하는 부분과 분기노드에서 다음 분기노드를 선택하는데 이용되는 투표알고리즘으로 구성되고, 투표는 AGV의 연속흐름을 위한 조건으로 제시되는 셀에 진입하고자 하는 다수 AGV들 중에서 하나를 선택하기 위해 행해진다. 이러한 우회알고리즘은 경로네트워크상에서 모든 노드들에 대한 경유최단경로테이블을 생성하는 것으로 시작된다. 하지만, 본 예제에서는 경로테이블이 완성되어있다고 가정하고 분기노드에서 셀 진입을 위한 투표절차에 대해서만 기술한다.

분기노드<8,2>에서 AGV(<8,2>)는 다음 분기노드를 선택하기 위해 투표과정을 수행하는데, AGV가 진입을 요구하는 셀은 노드 <8,7>를 중심으로 하는 셀 S(<8,7>,4)과 노드 <3,2>를 중심으로 하는 셀 S(<3,2>,4) 그리고 노드 <13,2>를

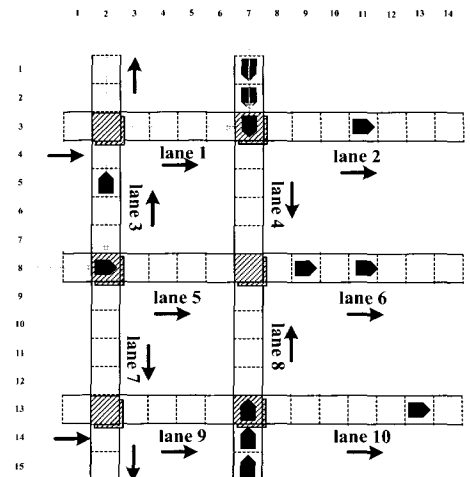


그림 5. AGV 동적경로계획의 예.  
Fig. 5. An example of dynamic routing for AGV.

중심으로 하는 셀 S(<13,2>,4)로 모두 3 개의 셀이다. 투표는 각각의 셀에 대하여 시행되고, 그 중에서 진입이 허용된 셀을 통해 다음 부분경로가 결정된다. 투표에 참여하기 위해 AGV(<8,2>)는 현재 노드진입시간, 목적노드 등에 대한 정보를 셀이 소속된 트래픽 제어기로 전송한다. 여기서 트래픽 제어기는 서로 독립된 기능을 수행하며 셀 제어기로 불리기도 한다.

다음에 기술된 내용은 AGV(<8,2>)가 S(<8,7>,4)에 대한 투표과정만을 나타내지만, 각각의 트래픽제어기는 동시에 S(<3,2>,4)과 S(<13,2>,4)에 대해서도 투표과정을 진행한다. 각 트래픽제어기의 투표결과는 관리제어기에 의해 최종적으로 결정됨으로써 AGV의 부분경로 계획이 완료된다. 투표절차를 설명하기에 앞서 다음과 같이 가정한다.

가정 1. 각 분기노드에 진입한 순서는 AGV(<8,2>), AGV(<3,7>), AGV(<13,7>) 순이다.

가정 2. AGV(<8,2>)가 노드 <8,3>을 점유할 때 목적노드까지 최단경로를 가지고, AGV(<3,7>), AGV(<13,7>)가 다음 순이다.

**초기화:** 투표구에 포함된 AGV는 자신의 id를 초기화 시키고, id는 현재노드에서 목적노드까지의 경유최단경로가 되고, 경로테이블에서 참조한다. 본 예제에서 AGV(<8,2>)가 S(<8,7>,4)를 점유하여 목적노드까지의 최단경로 d(<8,2>, <d1,d1>)=12로 한다. 그리고 AGV(<3,7>)와 AGV(<13,7>)은 각각 d(<3,7>, <d2,d2>)=15, d(<13,7>, <d3,d3>)=20로 한다.

**단계 1.** AGV(<8,2>)는 S(<8,7>,4) 진입을 위해 투표를 발의한다. 투표참가자는 S(<8,7>,4)의 입력노드를 점유하는 AGV의 집합으로 {AGV(<8,2>), AGV(<3,7>), AGV(<13,7>)}이 된다.

**단계 2.** 3.3 절에서 명시된 알고리즘에 따라 id를 순번에 따라 전송하고, id가 최소인 AGV를 선택하고, 투표 절차를 마친다.

본 예제에서는 AGV(<8,2>)가 id=12를 가져 투표에서 선출된다.

그림 6은 투표진행 과정을 단계별로 나타내고 있다. 큰 원안에 부여된 번호는 논리적 링의 번호를 나타내고, 작은 원은 각각의 투표참가자를 나타낸다. 본 예제에서는 총 9

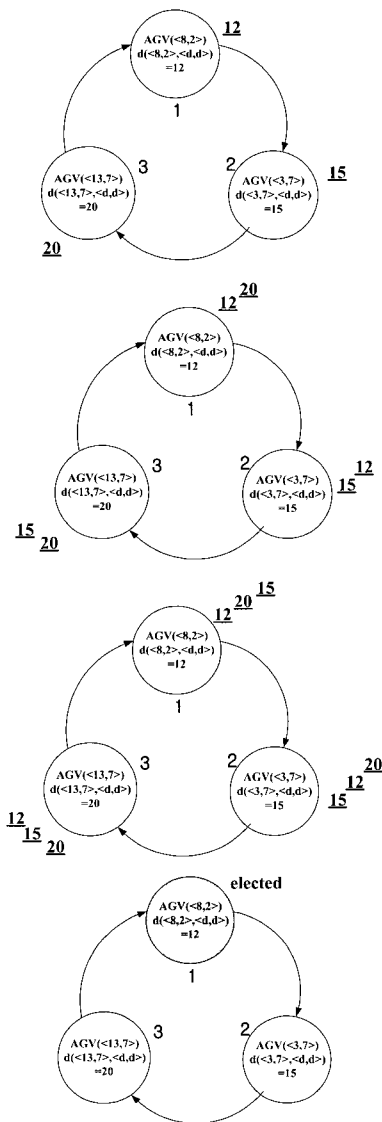


그림 6. AGV 시스템에서의 투표과정.  
Fig. 6. The election processing for AGV systems.

번의 메시지 전송으로 AGV(<8,2>)가 목적노드까지의 최단경로를 가진다는 것을 알 수 있다.

트래픽제어기(또는 셀제어기)는 투표구에서의 선출결과를 관리제어기로 전송하고, 분기노드에서 AGV 가 다수 셀에서 동시에 선출된 경우 목적노드까지의 최단경로가 가능한 하나의 셀 만을 허용한다. 이런 경우 다른 셀 제어기는 선출된 AGV 를 제외하고 재투표를 실시한다.

분기노드<8,2>에서 AGV(<8,2>)가 S(<8,7>,4)에 대한 진입이 허용되었으므로 AGV(<3,7>)과 AGV(<13,7>)은 우회노드를 통해 다음 부분경로가 계획된다. 이러한 일련의 투표과정을 통해 비록 AGV(<3,7>)과 AGV (<13,7>)은 우회시켰지만, 이를 뒤따르는 AGV(<2,7>), AGV (<1,7>), AGV(<14,7>), AGV(<15,7>) 들은 경로계획을 위해 정지하지 않고 투표과정을 통해 주어진 상황에서의 최적부분경로를 선택할 수 있다.

본 논문에서 제안한 제어시스템의 목적은 AGV 가 병합노드를 통과할 때 다음 3가지 조건을 만족하는 부분경로계

획이 가능한 노드를 결정하는 것이다. 첫째, 병합노드에서 충돌하지 않는다. 둘째, 충돌을 회피하기위해 정지하지 않고 우회경로를 선택한다. 셋째, 계획된 부분경로는 주어진 상황에서 최적경로이다. 지금까지 라우팅 및 스케줄링에 관한 많은 연구가 있어 왔으나 대부분 병합노드에서 충돌을 회피하기 위해 정지후 재출발하는 방식의 제어기법을 사용하거나[6], 경로네트워크에서 허용가능한 노드만을 탐색하여 라우팅하는 시간창기반의 경로탐색 기법을 사용하였다[4],[6]. 하지만 이러한 기법들은 스케줄링에 기반하므로 시스템의 변화에 의해 스케줄에 혼란이 발생할 경우 전체 시스템에 치명적인 영향을 줄 수 있다. 본 예제를 통해 비정지우선 우회알고리즘은 스케줄에 기반하지 않으므로 시스템의 변화에 영향을 받지 않는다는 것을 간접적으로 보일 수 있다. 또한 투표알고리즘을 주어진 상황에서 목적노드까지의 최적경로계획과 목적지까지 AGV 의 연속흐름이 가능하다는 것을 보일 수 있다.

V. 결론 및 향후과제

본 논문에서는 AGV 시스템의 새로운 동적경로문제를 다루었다. 특히 연속정지문제를 해결하기 위해 비정지우선 우회알고리즘을 제안하였다. AGV 시스템에서 비정지 제어문제는 확장된 금지영역의 설정을 요구하게 되는데 이를 셀이라 정의하였다. 투표 알고리즘의 투표 참가자의 id는 다음 경우 노드에서 목적노드까지의 최단경로로 하였다. 이를 위해 k-경유 최단경유노드 알고리즘을 도입하여 경로태이블을 이용하는 이단계기법을 이용하였다. k-경유 최단경로는 기존의 최단경로 알고리즘의 제한성과 k-최단경로 알고리즘의 복잡성을 보완한 것으로 동적경로계획 알고리즘에 적합하다. 마지막으로 사례연구를 통해 다수 AGV시스템에서 비정지우선 우회경로 알고리즘의 타당성을 입증하였다. 차후 연구과제로는 AGV의 확장된 동속주행 개념을 도입하여 AGV 시스템의 비정지 문제를 해결하고, 투표알고리즘에서 분기노드로 부터 목적노드까지의 최단경로 AGV에 우선순위를 부여하는 제어정책이 전체 시스템에 얼마나 효율적인지를 컴퓨터 시뮬레이션을 통해 정량적으로 분석하고 이에 따라 제안된 비정지우선 우회알고리즘의 전체 효율성을 검토하는 점 등이 있다.

참고문헌

[1] A. J. Broadbent, C. B. Besant, S. K. Premi, and S. P. Walker, "Free ranging AGV systems: promises, problems and pathways", in *Proc. of Int Conf. Automated Material Handling*, pp. 221-237, 1985.  
 [2] E. G. Chang and R. Roberts, "An improved algorithm for decentralized extrema-finding in configurations of processors", *Communications of the ACM*, Vol. 22, No. 5, pp. 281-283, May 1979.  
 [3] P. J. Eglelu and J. M. A. Tanchoco "Potentials for bidirectional guided for automated guided vehicle systems", *Int. J. Prod. Res.*, Vol. 24, No. 5, pp. 1075-1097, 1986.  
 [4] J. Huang, U. S. Palekar, and S. G. Kapoor, "A labeling algorithm for the navigation of automated guided vehicles", *Tran. ASME J. Eng. Ind.*, Vol. 115, No. 1, pp. 315-321, Aug. 1993.  
 [5] C. W. Kim and J. M. A. Tanchoco, "Conflict-free shortest-time

- bi-directional AGV routing". *Int J. Prod. Res.*, Vol. 29, No. 12, pp. 2377-2391. 1991.
- [6] J. H. Lee and B. H. Lee, "A real-time traffic control scheme of multiple AGV systems for collision free minimum time motion: a routing table approach", *IEEE Trans. Syst., Man, and Cybern. A*, Vol. 28, No. 3, May. 1998.
- [7] B. A. Peters, J. S. Smith, and S. Venkatesh, "A control classification of automated guided vehicle systems", *International Journal of Industrial Engineering*, Vol. 3, No. 1, pp. 29-39, 1996.
- [8] L. Qiu and W. W. Hsu, "Scheduling and routing algorithms for AGVs: a survey." *Technical Report: CAIS-TR-99-26*, Center for Advanced Information Systems, School of Applied Science, Nanyang Technological University, Singapore, Oct. 1999.
- [9] D. R. Shier, "On algorithms for finding the K shortest paths in a network", *Networks*, Vol. 9, pp. 195-214, 1979.
- [10] F. Taghaboni and J. M. A. Tanchoco, "Comparison of Dynamic Routing Techniques for Automated Guided Vehicle Systems". *Int. J. Prod. Res.*, Vol. 33, No. 10, pp. 2653-2669, 1995.
- [11] G. Tharna, G. H. Nicholas, and S. Chelliah, "Design and operational issue in AGV-served manufacturing systems", *Annals of Operations Research*, Vol. 76, pp. 109-154, 1998.



신성영

2000년 울산대학교 전자공학과 졸업. 동대학원 석사(2002). 2002년~현재 울산대학교 전기전자정보시스템공학과 박사 과정. 관심분야는 멀티에이전트시스템의 관리제어, 생산시스템의 자동화, 바이오정보공학 응용 등.



조광현

1993년 한국과학기술원 전기및전자공학 졸업. 동대학원 석사(1995), 동대학원 박사(1998), 1998년~1999년 동대학원 위촉연구원 및 연구연구원, 1999년 3월~현재 울산대학교 전기전자 및 정보시스템 공학부 조교수. 관심분야는 이산사건시스템의 해석 및 관리제어, 생산시스템 자동화(반도체 생산시스템 자동화), 통신망 분석 및 제어, 광통신망 시스템(스위칭 및 라우팅), 바이오정보공학 응용 등.