

조립순서의 자동생성에 관한 연구⁺

Automatic Generation of Assembly Sequences⁺

손경준 · 정무영*

Kyoung-joon Son and Mooyoung Jung*

Abstract

It is well known that an assembly operation is usually constrained by the geometric interference between parts. These constraints are normally presented as AND/OR precedence relationships. To find a feasible assembly sequence which satisfies the geometric constraints is not an easy task because of the TSP(Traveling Salesman Problem) nature with precedence constraints.

In this paper, we developed an automated system based on Neural Network for generating feasible assembly sequences. Modified Hopfield and Tank network is used to solve the problem of AND/OR precedence-constrained assembly sequences. An economic assembly sequence can be also obtained by applying the cost matrix that contains cost-reducing factors.

To evaluate the performance and effectiveness of the developed system, a case of automobile generator is tested. The results show that the developed system can provide a "good" planning tool for an assembly planner within a reasonable computation time period.

1. 서 론

조립 및 제조관련 연구는 본질적으로 상품을 전제로 한다. 지금까지는 생산성제고를 추구하면서 다양한 제품들을 위한 조립 및 제조기술들을 개발하고 최적화를 시도하였다. 조립 및 제조 산업체의 힘은 설비자원들을 얼마나 효율적이고 효율적으로 사용하는가에 좌우되어 왔다. 그러나 사회가 발전하고 가용자원이 제한됨에 따라 우리는 가용자원의

생산성으로부터 좀더 개념적인 상품들(예컨대, 의사결정에 필요한 지식들)에 의해서 조립 및 제조 관련 산업체들의 경쟁력이 결정됨을 보고있다. 따라서 산업체가 경쟁력을 계속 유지하기 위해서는 조립 및 제조에 관련된 새로운 기술에도 개념적인 상품들의 도입이 필요하며 이에 대한 연구가 반드시 이루어져야 한다[1].

조립 및 제조관련 연구중 핵심적인 분야의 하나로 조립순서(assembly sequence) 및 절차를 계획하는 분야를 들 수 있다. 조립계획 분야에 있어서도 자동화의 필요성이 점차 증대되고 있고 이에 대한 기본적인 연구가 이루어지고 있다. 자동조립계획은 조립환경을 이루는 조립모델과 조립될 부품들

⁺본 연구는 한국과학재단의 목적기초 연구지원에 의해 수행되었음.

* 포항공과대학 산업공학과

에 대한 정보를 담고 있는 부품모델의 분석을 통해 시작된다. 조립모델과 부품모델의 분석을 통해 특정한 조립작업의 제약사항을 얻어내고 이로부터 조립계획을 생성한다. 이때 생성되는 조립계획사항으로는 공구, 고정구의 선택, 조립경로의 결정, 조립순서의 자동생성등을 들 수 있다. 특히 조립순서의 자동생성은 자동조립계획 수립시 요구되는 중요한 부분이다. 또한, 조립의 경제적 중요성은 조립작업의 자동화뿐만 아니라 조립순서의 결정에 있어서도 나타난다. 따라서 자동조립계획 프로그램은 가능한 조립순서의 생성뿐만 아니라 경제적인 조립순서도 제시할 수 있어야 한다[2].

조립작업에 있어서 적절한 조립순서를 결정하는 것은 중요하면서도 어려운 일이다. 그 이유는 첫째, 가능한 조립순서의 수는 구성 부품의 수가 적은 경우에도 많을 수 있으며[3, 4], 둘째, 부품 설계상의 작은 변화에도 선택할 수 있는 조립순서는 크게 달라지고[3], 셋째, 복잡한 조립품에 있어 가능한 조립순서는 경험있는 기술자라 하더라도 많은 가능성을 놓칠 수 있기 때문이다. 따라서 조립순서를 생성할 수 있는 체계적인 절차는 조립시스템의 계획과 프로그램을 자동화하기 위해서도 필요하다.

그러나 조립품이 아주 복잡한 topological 구조를 갖고 있거나 조립부품의 수가 많아지는 경우에는 아직도 해결해야 할 난제들이 많이 남아있다. 예를 들면, automatic geometric and physical reasoning을 적용하는데의 어려움이라던가 수 많은 가능한 조립순서중 어떤 순서를 고를 것인가 또, 조립할 부품의 수가 증가함에 따라 가능한 조립순서의 수가 지수적으로 증가하는 문제는 어떻게 해결할 것인가 하는 것 등이 있다. 이중에서 최적의 해를 찾는 것은 TSP(Traveling Salesman Problem)로 이는 NP-Complete에 해당된다. 이 문제를 풀기 위한 방법으로 분지한계(branch and bound)법 등이 있으나 이것 역시 지수형의 계산량이 소요된다는 문제점이 있으며 기존의 연구들[4, 5, 6] 역시 이러한 문제를 가지고 있다.

본 연구에서는 신경망(neural nets)을 이용하여 기존의 방법보다 간단하게 문제를 표현하고 비교적 짧은 시간내에 조립순서를 생성할 수 있는 방법을 제시 한다. 또한 조립순서의 경제성을 고려하여 우

선순위를 위반하지 않는 가능해 중에서 비용요소를 최소로 하는 조립순서도 찾고자 한다. 이를 동시에 해결해 줄 수 있는 일체화된 시스템을 개발하였으며 C언어를 사용하여 시스템을 구축하였다. 개발된 시스템의 성능과 효율성을 평가하기 위해 자동차용 발전기 조립품에 대한 시뮬레이션을 386PC에서 실험하였고 그 결과를 제시하였다.

2. 기존 연구의 고찰

조립순서 생성을 위한 연구는 Bourjault[7]가 조립품의 조립선(liaison)모델을 제시하면서 시작되었다. 조립품을 부품들간의 조립선 그래프로 나타내고 이에 근거한 구조화된 질문에 사용자가 대답함으로써 주어진 조립품에 대한 모든 우선순위(precedence) 지식을 획득하는 절차를 제시하였다. 그러나 부품의 수가 늘어나면 부품들간의 조립작업(mating task)수도 따라서 증가하게 되므로 우선순위를 묻는 질문의 수가 가하급수적으로 늘어난다는 문제점을 안고 있다. De Fazio와 Whitney[4]는 각 조립작업별로 선행작업과 후행작업을 사용자로부터 대화식으로 입력받아 조립품의 우선순위를 생성하였다. 이들의 방법은 각 조립작업에 대해 두가지 형태(해당 작업에 선행되어야 할 작업과 후행되어야 할 작업)의 질문을 하여 조립품의 모든 우선순위를 생성하는 것으로 Bourjault의 방법을 개선하여 $2L^2$ (L 은 조립선의 수) 이상이던 질문의 수를 $2L$ 개로 감소시켰다.

Hommem de Mello와 Sanderson[5]은 모든 가능한 조립순서를 나타낼 수 있도록 조립작업을 AND/OR 그래프로 표현하였다. AND/OR 그래프는 비교적 간결하게 모든 가능한 조립순서를 나타낼 수 있으며 트리탐색 방법에 의하여 가능한 조립순서를 찾을 수 있다는 장점이 있다. 한편, Wolter[8]는 XAP/1(Experimental Assembly Planner 1)을 개발했는데 그는 여기서 부품들간의 기하학적인 제약을 ACG(Assembly Constraints Graph)로 표현하였다. 이는 AND/OR 그래프와 유사한 것으로서 상호 변환이 가능하다.

위의 연구들이 대부분 조립순서의 생성에만 초점을 맞춘 반면 조립공정 자체를 CAPP(Computer

Aided Process Planning)의 일부로 보아 좀더 광범위한 자동화를 시도하는 연구도 진행되어 왔다. 이들 연구들은 CSG(Constructive Solid Geometry)등과 같은 CAD데이터를 분석하여 조립에 필요한 정보를 추출하고자 하였다. Liu와 Popplesstone[9]은 Solid modeller로부터 조립될 부품들간의 결합특징(mating feature)을 추출해 내고자 하였다. 즉, PADL2가 제공하는 boundary 정보로부터 실린더(cylinder)나 구멍(hole)과 같은 간단한 결합특징을 PROLOG를 이용하여 추론하는 방법을 제시하였다. Tapadia와 Handerson[10]은 부품의 solid 데이터로부터 룰베이스(rule-base)를 이용하여 조립에 관련된 부품특징을 얻어내고자 하였다. 그 밖에도 조립을 위한 특징을 추출하기 위한 많은 연구들[11, 12, 13, 14]에 있어서 부품을 적절한 자료구조로 표현하고 이로부터 기하학적 특징들을 추출하려 하고 있으나 복잡한 부품은 인식하지 못하는 등 실제로 이용하기 위해서는 해결해야 할 문제들이 많이 남아있다.

TSP를 위한 신경망 응용에 관련된 많은 연구중 본 논문과 관련있는 연구를 살펴보면, Hopfield와 Tank[15]는 최적화 문제를 풀기위한 아날로그 수치적 네트워크(Analog computational network)의 일반적 구조를 발표하였다. 이 연구는 최적화 문제의 해를 구하는데 소요되는 시간의 단축을 꾀할 수 있고 종래의 방법들과는 전혀 다른 최적화 문제들을 다룰 수 있다는 가능성을 제시하였다. 이들이 다룬 최적화 문제는 TSP로 계산 결과 나온 해들이 전체적 최적해(global optimum)를 보장하지 못하고 모수에 지나치게 민감하다는 문제는 있으나 빠른 시간안에 좋은 해들을 얻어낼 수 있다는 장점이 있다. Chen[3]은 Hopfield와 Tank의 TSP 최적화 신경망의 에너지 함수에 우선순위를 반영하는 새로운 에너지 항을 추가시켜 이를 조립순서 결정 문제에 적용시켰다. 이 새로운 에너지 항은 특정한 요구조건을 만족시키는 방향으로 뉴런의 상태를 유도하는 것과 마찬가지로 네트워크의 상태가 우선순위를 위반하지 않도록 해준다.

기존 연구들을 고찰해 보면, 신경망을 이용한 방법을 제시한 연구[3]와 다른 방법을 이용하여 일체화시킨 연구[16]는 있으나 본 논문에서 시도하

는 신경망을 이용하여 조립순서를 자동으로 생성하는 일체화된 시스템을 개발한 연구는 없었던 것으로 보인다.

3. 조립순서의 결정

본 연구의 목적은 앞서 언급한 바와 같이 조립품의 우선순위로부터 조립순서를 자동생성하는 일체화된 시스템을 개발하는데 있다. 사용자로부터 조립구속조건에 따른 선 후행작업을 입력받아 조립품의 우선순위를 생성하고 이를 TSP 신경망모델에 반영하여 조립순서를 생성한다. 이때 생성되는 가능한 조립순서중 보다 나은 해를 구하기 위해 신경망 모델에 입력자료로 들어가는 비용행렬에 기존 연구 [3]와 틀리게 의미를 부여 한다. 이를 위해 Hopfield와 Tank의 신경망 모델에서의 TSP최적화를 수정하여 이용한다.

3.1 조립순서 결정의 문제점

일반적으로 n개의 부품으로 구성되어 있는 조립제품은 부품들간의 결합에 있어서 우선순위 관계(precedence relationship)를 가진다. 이러한 우선순위 관계는 조립품을 이루고 있는 각 부품들의 기하학적 형상에 의하여 조립순서에 제한을 받기 때문이다. 따라서 적절한 조립순서의 결정은 다음의 두가지 이유때문에 용이하지가 않다. 첫째, 가능한 조립순서의 조합(set)수는 부품 갯수가 적은 경우에도 많아질 수 있으며 부품의 수가 증가함에 따라 가능한 조립순서 조합 수는 지수적으로 증가하게 된다. 둘째, 조립품의 설계를 약간만 변경해도 가능한 조립순서의 선택은 크게 변화하기 때문이다 [17].

조립순서를 나타내는 그래프는 노드(node)를 조립할 작업들로 보고 각 노드를 연결하는 간선(edge)을 두 작업간의 비용으로 표시하여 작성할 수도 있다. 따라서 조립순서의 결정문제는 각 조립작업을 한번씩 방문하는 TSP와 성격은 비슷하나 일반적인 TSP로는 표현할 수 없다. 그 이유는 조립작업간에 우선순위 관계가 있기 때문이다. 즉, 일반 TSP에서 도시를 방문하는 판매원의 이동에 제한이 있다는 것이다. 예를 들면, "도시 A를 방

문하기 전에는 도시 B를 방문하지 못한다.”는 것이 이에 해당된다.

3.2 우선순위의 표현과 생성

일반적으로 기계조립품을 구성하고 있는 부품들은 무방향 그래프(undirected graph) $G=(P, L)$ 로 나타낼 수 있다. 이는 정점(vertex)의 집합 $P, P=\{p_i \mid p_i \in G, i=1, 2, \dots\}$ 와 이 정점들을 연결하는 간선의 집합 $L, L=\{l_i \mid l_i \in G, i=1, 2, \dots\}$

로 나타낼 수 있다. 각각의 정점은 각 부품들을 나타내고 각각의 간선 l_i 는 두 부품을 연결하는 조립선(liaison)을 나타낸다.

Bourjault[7]의 연구에서 예시된 볼펜을 보면, 그림 1은 볼펜의 조립품을 나타내며 그림 2는 조립품의 결합작업(mating operation)을 나타내는 조립선 그래프이다. 그림 2에 나타나 있는 조립품의 조립선 그래프와 조립작업에 필요한 지식에 의해 조립품의 우선순위 지식을 알아낼 수 있다.

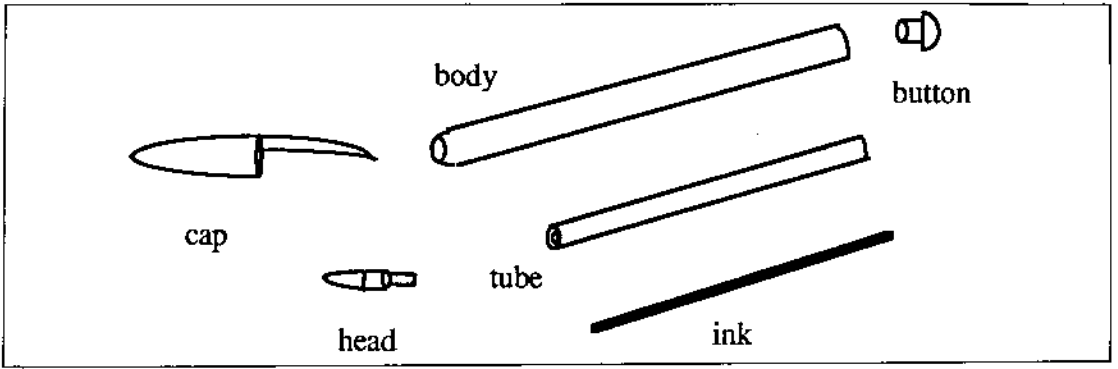


그림 1. 볼펜의 조립품

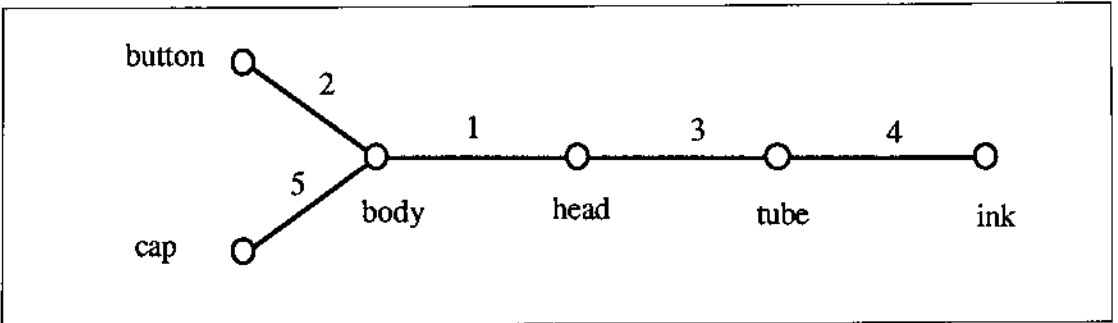


그림 2. 볼펜 조립품의 조립선 그래프

본 논문에서는 우선순위 생성에 De Fazio와 Whitney[4]의 질의응답 방법을 이용한다. 예를 들어 그림 1과 그림 2에서, 조립선 1(head와 body의 결합)은 조립선 5(cap과 body의 결합)보다 선행되어야 한다는 것, 조립선 3과 4(head, tube, ink의 결합)는 조립선 1과 2(head, body, button의 결합)에 선행되어야 한다는 것등이다. 모든 조립선에 대해 부품들 간의 관계를 조사하면 그 조립품에 있어서의 모든 우선순위지식을 얻어낼 수 있는데 볼펜조립품의 우선순위를 정리하면 다음

과 같다.

- 3→4
- 1→5
- 4→(1 & 2)

여기서 기호 ‘→’는 ‘반드시 선행되어야함’을 나타내며 기호 ‘&’는 논리 AND를 나타낸다. 4→(1 & 2)는 작업 4가 작업 1과 2에 앞서 이루어져야 함을 나타낸다. 만일 조립작업 4가 조립작업 1 또는 2에 선행되어야 한다면 이때는 우선순위를 4→

$$+B/2 \sum_{i=1}^N \sum_{X=1}^N \sum_{Y=1, Y \neq X}^N V_{xi} V_{yi}$$

$$+C/2 \left(\sum_{X=1}^N \sum_{i=1}^N V_{xi} - n \right)^2;$$

A, B, C는 상수 (3)

식 (3)에서 첫번째 항은, 하나의 행 X에서 2개의 열 i, j의 뉴런 출력값이 동시에 1이라면 에너지를 A만큼 크게 하는 항으로, 같은 도시를 두번 통과하는 것을 방지한다. 두번째 항은, 행렬의 하나의 열 i에서 X, Y 2개 뉴런의 출력값이 동시에 1이라면 에너지를 B만큼 크게 하는 항으로, 2개의 도시를 동시에 통과하는 논리적으로 모순된 경로를 배제하는 효과가 있다. 세번째 항은, 행렬의 모든 요소의 합이 n이 되지 않으면 에너지를 C에 비례해서 크게 하는 항으로, 통과하는 도시의 수가 전부 n개가 되도록 제약을 부여한다.

TSP의 두번째 요구조건은 가능한한 짧은 경로를 찾는 것이다. 이는 식 (3)의 에너지 함수에 다음의 항을 첨가하면 된다.

$$D/2 \sum_{X=1}^N \sum_{Y=1, Y \neq X}^N \sum_{i=1}^N d_{xy} V_{xi} (V_{y, i+1} + V_{y, i-1})$$

(4)

식 (4)에서 D는 상수이며 d_{xy} 는 두 도시 사이의 거리이다. 아랫첨자는 편의를 위하여 modulo, n으로 정의한다. 즉, V_{n+1} 는 V_1 와 같다. 따라서 식 (4)는 유효한(valid) 방문순서인 경우 전체의 방문거리 또는 비용을 나타낸다. 식 (4)를 통해 인접한 두 도시간의 거리를 에너지 함수에 반영함으로써 거리의 합이 적은 값을 갖도록 뉴런을 배치하는 역할을 담당한다.

뉴런간의 가중치(weight)는 다음과 같이 결정된다.

$$W_{xi, yi} = -A \delta_{xy}(1 - \delta_{ii}) - B \delta_{ii}(1 - \delta_{xy}) - C - D d_{xy}(\delta_{i, i+1} + \delta_{i, i-1})$$

(5)

여기서 $i=j$ 인 경우에는 $\delta_{ii}=1$ 이고 그렇지 않은 경우에는 $\delta_{ii}=0$ 이다. 식 (5)에서 첫번째 항은 그림 3에서와 같이 하나의 행에 1이 하나 이상 오는 것을 방지한다. 만일 그렇지 않다면 V_{xi} 와 V_{yi} 사이의 가중치를 A만큼 감소시킨다. 마찬가지로 두번

제 항은 하나의 열에 1이 하나 이상 나타나는 것을 방지한다. 세번째 항은 전체적 억제(global inhibition)를 나타내며 네번째 항은 두 도시의 거리에 비례하여 가중치를 감소시키는 항이다.

우선순위의 반영

각 조립작업을 방문해야 할 도시로 보면 조립순서의 결정문제는 TSP와 유사함을 알 수 있다. 그러나 조립순서 결정문제는 식 (3)과 식 (4)의 에너지 함수를 그대로 사용할 수는 없다. 왜냐하면 앞서 언급했듯이 조립순서의 결정에는 우선순위 제약이 있기 때문이다. 즉, TSP에서 외관원이 방문해야 할 도시중 다른 도시보다 먼저 방문해야 하는 도시가 있다면, 혹은 도시 A보다 B를 먼저 방문해야 한다는 등의 제약이 첨가되는 것이다.

이를 위해서는 우선순위 제약을 위반하지 않도록 에너지 함수를 수정하여 Hopfield와 Tank모형을 이용한 Chen의 방법 [3]을 사용한다. 식 (3)과 식 (4)의 에너지 항을 합한 에너지 함수를 E'이라고 하고, 새로운 에너지 항을 추구한 전체 에너지 함수를 E_{new} 라 하면 이는 다음과 같은 식이 된다.

$$E_{new} = E' + F/2 \sum_{i=1}^N \sum_{j=1, i < j}^N V_{xi} V_{yj}; X = \text{후행 작업}, Y = \text{선행작업}$$

(6)

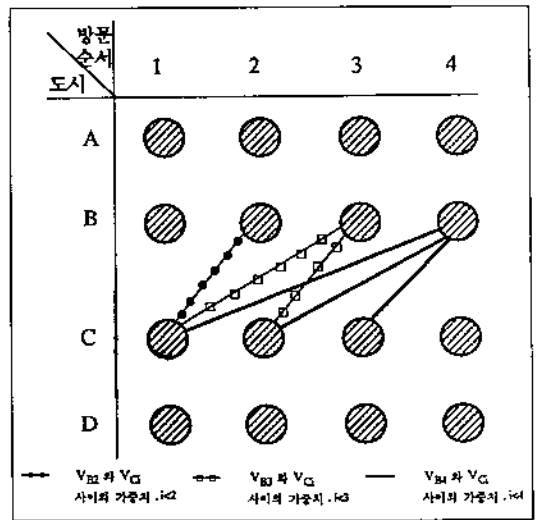


그림 4. 우선순위 B→C를 반영한 뉴런간의 가중치

여기서 후행작업과 선행작업은 각각 '→' 연산자의 오른쪽과 왼쪽의 조립작업에 대응되며, F는 양의 상수이다. 그림 4에서 보는 바와 같이 B→C의 우선순위를 Hopfield와 Tank 모델에 반영하기 위해서는 해당되는 가중치가 음이 되도록 한다. 즉, 두 번째와 세 번째 행에 있는 뉴런간의 가중치는 다음과 같이 결정된다.

$$T_{x_i, y_j}^{new} = T_{x_i, y_j} - F; X=C, Y=B \text{에 대해,}$$

그리고 $i < j$ 식 (6)에서 보면 $i < j$ 일때 V_{x_i} 와 V_{y_j} 가 동시에 1이 되는 경우에는 전체 에너지를 증가시키게 된다. 뉴런은 에너지를 감소시키는 방향으로 수렴하기 때문에 후행작업이 선행작업에 우선하는 경우를 배제하게 된다. 그림 4는 우선순위 B→C가 에너지 함수에 반영되어 선행작업 B가 i번째 위치한 경우 후행작업 C가 i번째 이후에 나타나도록 $j < i$ 일때 뉴런 V_{x_i} 와 V_{y_j} 사이에 연결된 가중치를 F만큼 감소시킨다. 그림 5는 선행작업 B가 두번째 위치한 경우 후행작업 C가 두번째 이후에 나타나도록 해가 형성되는 것을 보여준다.

	조립 순서	1	2	3	4
조립 작업	A				
	Y=B	0	1	0	0
	X=C	0	0	?	?
	D				

그림 5. 우선순위 B→C를 고려한 TSP해

3.2절에서도 언급했듯이 AND나 OR의 어떠한 조합도 결합법칙이나 배분법칙에 의해 이끌어낼 수 있다. 따라서 AND나 OR의 조합으로 구성된 복잡한 우선순위로 에너지 함수에 반영시킬 수가 있게 된다.

3.4 경제적 조립순서

조립순서에는 가능한(feasible) 조립순서와 경제적인 조립순서가 있다. 가능한 조립순서란 부품의 기하학적 제약을 만족하는 즉, 조립의 우선순위를

위반하지 않는 조립순서를 말한다. 그러나 가능한 조립순서라 해도 실용적인 측면에서 볼때 바람직하지 못할 수도 있다. 따라서 가능한 조립순서중에서 경제적인 조립순서를 찾는 것은 가능한 조립순서를 찾는 것만큼 중요한 일이다.

여기서 '경제적'이라는 의미는 반드시 비용만을 뜻하지는 않는다. 각 조립작업에 걸리는 시간이라든지 특정한 조립작업 순서를 따르기 위해 새로운 고정구의 요구, 부품의 접근 방향(orientation) 변경, 조립작업 진행시 조립품의 안정성 여부 등 생각해 볼 수 있는 기준은 여러가지가 있다. 그러나 이러한 척도들을 조립순서 결정에 반영하기 위해서는 사전에 이를 정량적으로 표시해야 하는 문제점이 있다. 따라서 이러한 비용요소들을 체계적으로 얻어내는 방법에 관한 연구도 이루어져야 할 것이다.

본 논문에서는 하나의 조립품안에 여러개의 동일 부품이 존재할 때 이들 동일 종류의 부품들이 연속해서 조립될 수 있는 비용요소를 고려해 보았다. 이 비용요소는 Hopfield와 Tank 모델에 입력자료로 들어가는 비용행렬에 포함되며 3.3절의 식 (4)에 의해 전체 에너지 함수에 반영된다. 조립작업이 연속하여 이루어지는 것이 바람직한 경우에는 비용을 0.5로, 나머지는 1.0으로 한다. 그림 6에 작업 B, C가 연속하는 것이 좋은 경우의 비용행렬을 예로 나타내었다.

	A	B	C	D
A	0.0	1.0	1.0	1.0
B	1.0	0.0	0.5	1.0
C	1.0	0.5	0.0	1.0
D	1.0	1.0	1.0	0.0

그림 6. 작업 B, C 연속 선후시의 비용행렬

4. 조립순서 자동생성 시스템

조립순서를 자동으로 생성하는 시스템은 C언어로 짜여진 3개의 모듈로 구성되어 있다. 전체의 시스템 구성도가 그림 7에 나타나 있다. 그림 7에서 보는 바와 같이 먼저 사용자는 질의응답 모듈을 통

해 조립품의 조립선 그래프를 바탕으로 선 후행작업을 입력하게 된다. 입력된 자료들은 우선순위에 해석 모듈을 통해 우선순위를 해석하게 되고 그 결과는 신경망을 이용한 조립순서 생성모듈에 넘겨진다. 이때 조립작업간의 비용요소인 비용행렬이 입력자료로 들어간다. 조립순서 생성모듈에서 비용행렬과 우선순위를 고려하여 적절한 조립순서가 자동 생성되어 진다. 각 모듈에 대한 자세한 설명은 다음과 같다.

질문형식에 해당하는 각 작업의 선행작업을 모두 입력받아 리스트(list)에 저장한 뒤 두번째 질문형식인 후행작업을 입력 받는다. 각 작업의 후행작업이 하나씩 입력될 때마다 이미 저장되어 있는 리스트의 변경여부를 확인하여 선행작업의 추가가 필요한 경우는 리스트를 갱신한다. 예를 들어 첫번째 질문형식에서 얻어진 우선순위가 $1 \leftarrow (5 \& 6 \& 7)$ 이고 두번째 질문형식에서 작업 8의 후행작업중에 1이 있다면 기존의 우선순위에 이를 추가한다. 위의 경우에는 $1 \leftarrow (5 \& 6 \& 7 \& 8)$ 이 된다. 조립작업에 대한 모든 질의 응답이 끝나면 생성된 우선순위는 화일로 저장되며 이는 우선순위 해석모듈의 입력자료도 이용된다.

4.1 선 후행작업 질의응답 모듈

우선순위의 생성은 기본적으로 3.2절에서 예로 보인것과 같은 방식으로 이루어진다. 먼저 첫번째

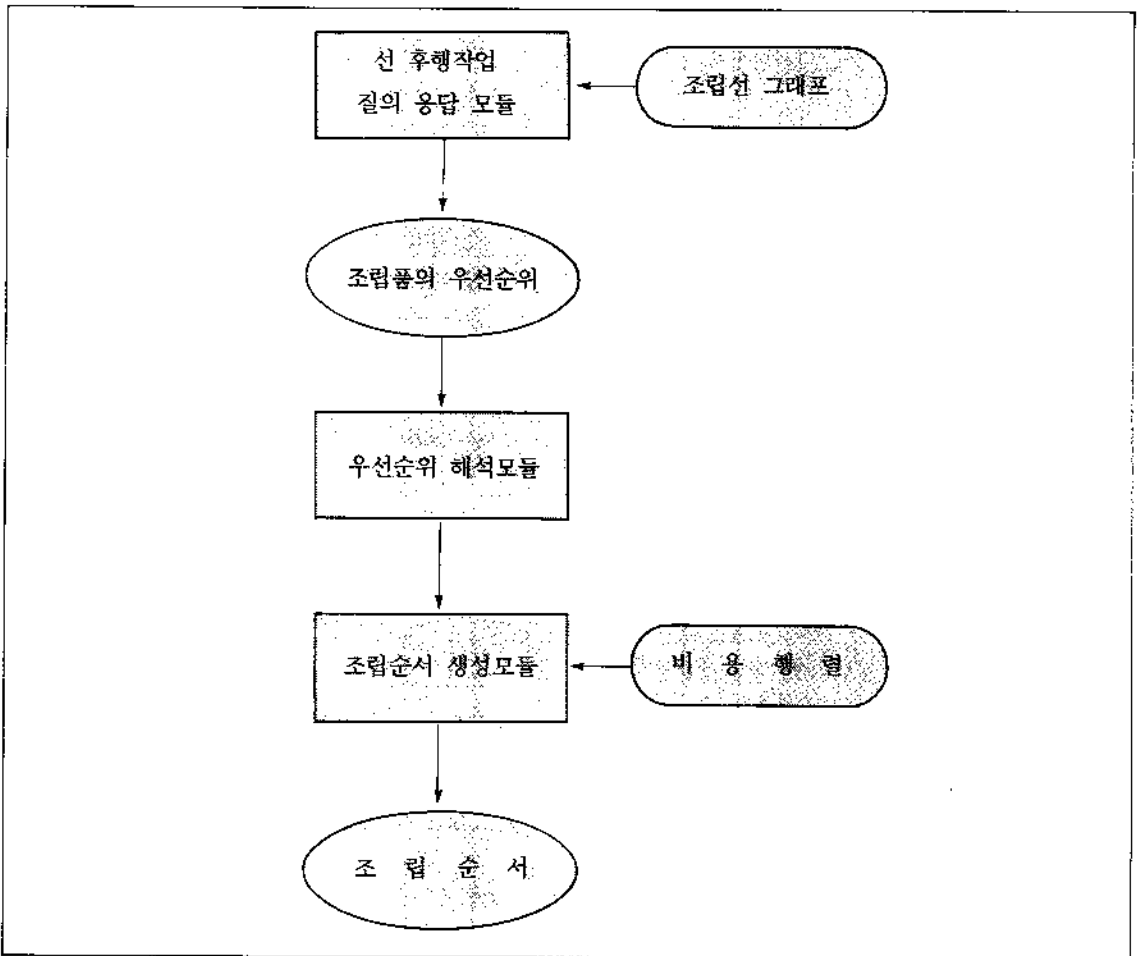


그림 7. 조립순서 자동생성 시스템 구성

4.2 우선순위 해석모듈

우선순위 해석모듈에서는 선 후행작업 질의응답 모듈에서 얻어진 우선 순위를 해석하여 신경망의 에너지함수에 반영한다. 우선순위 해석모듈의 코드 (code)가 그림 8에 나타나 있다. 그림 8은 선행작업들이 AND로 연결되어 있을때 이를 해석하여 조립순서 생성모듈의 변수에 대입하는 코드이나, 문번호 2에서 하나의 문자를 읽는다. 문번호 3에서는 읽어들인 문자가 개행(line feed)문자이면 다음 우선순위를 해석한다. 문번호 4에서는 읽어들인 문자가 숫자인지를 확인한다. 만일 숫자이면 이는 선행작업이므로 문번호 6~8을 실행하고 AFTER

가 0이면 후행작업이므로 문번호 11~13을 실행한다. 문번호 6~8은 읽어들인 숫자들을 공백이 읽혀질 때까지 계속 합쳐서(concatenate) 선행작업을 나타내는 2차원 배열 pred[][]에 저장한다. 문번호 11~13은 후행작업을 배열 succ[]에 저장하는 작업이다. 문번호 16~20은 문자 '<'가 읽혀졌을 때 다음에 나올 숫자들이 선행작업임을 나타내도록 flag AFTER를 1로 한다. 화일의 끝에 올 때까지 문번호 1~26의 프로그램을 수행한다. 표 1에 하나의 우선순위 화일(file)과 이를 그림 8의 모듈로 처리한 후의 결과를 나타내었다.

```

1      while(!feof(prec) {
2          TOKEN=fgetc(prec);
3          if(TOKEN=='\n') {AFTER=0;j=0;i++;;}
4          else if (TOKEN>='0' && TOKEN<='9') {
5              if (AFTER) {
6                  itoa(TOKEN-48, TOKEN_S, 10);
7                  strcat(pred_S, TOKEN_S);
8                  pred[i][j]=atoi(pred_S);
9              }
10             else {
11                 itoa(TOKEN-48, TOKEN_S, 10);
12                 strcat(succ_s, TOKEN_S);
13                 succ[i]=atoi(succ_s);
14             }
15         }
16         else if(TOKEN=='<') {
17             TOKEN=fgetc(prec);
18             AFTER=1;
19             *succ_s=NULL;
20         }
21         else if(TOKEN=='&')
22             j++;
23             *pred_s=NULL;
24         }
25         else if(TOKEN=='') *pred_s=NULL;
26     }

```

그림 8. 우선순위 해석모듈의 코드

표 1. 우선순위와 우선순위 해석모델에 의한 결과

우선순위	변수에 저장된 값
4←(7&8)	succ[0]=4 pred[0][]={7, 8}
8←7	succ[1]=8 pred[1][]={7}
9←(11&12&13&14)	succ[2]=9 pred[2][]={11, 12, 13, 14}

4.3 조립순서 생성모델

조립순서의 생성모델은 Hopfield와 Tank에너지 함수에 우선순위를 반영하는 새로운 에너지항을 추가한 프로그램이다. 조립순서 생성모델에서는 기하학적 제약을 위반하지 않는 가능해와 이들 가능해 중에서 작은 비용을 가지는 경제적인 해를 모두 산출한다. 모델의 전체기능을 그림 9에 pseudo코드로 나타내었다.

5. 실험 및 결과분석

개발된 시스템의 성능평가와 효율성을 평가해 보기 위해 자동차용 발전기에 대하여 조립순서를 생성해 보았다. 사용된 컴퓨터는 co-processor가 탑재된 20MHz급 386PC이다. 기존의 연구들이 대개 10개 내외의 부품들을 가진 조립품을 대상으로 하였으나 [3, 4, 5, 6, 7, 16] 본 논문에서는 18종류 33개의 부품으로 이루어진 자동차용 발전기 조립품을 대상으로 하였다. 그림 10에 전체 조립품의 구성도가 나타나 있으며 부품들 간의 조립작업을 나타낸 조립선 그래프가 그림 11에 나타나 있다. 각 작업에 대한 선 후행작업을 입력하고 우선순위 해석모델에서 그림 12와 같은 조립품의 모든 우선순위를 얻을 수 있었다. 그림 10에서 보는 바와 같이 nut 1, 2, 3, 4와 r_spacer 1, 2, 3, 4 등은 동일한 조립방향을 갖는 같은 종류의 부품이다. 이로 인한 비용요소를 그림 13의 비용행렬에 나타내었다.

표 2에는 생성된 조립순서를 나타내었으며 이들의 비용 그래프를 그림 14에 나타내었다. 총 300개의 조립순서를 생성시킨 결과 이중 7%인 21개가 가능한 조립순서였다. 소요 시간은 14시간 30분으로 하나의 순서생성에는 약 3분 정도, 하나의

가능한 조립순서 생성에는 약 41분 정도 시간이 걸렸다.

그림 14에서 보는 바와 같이 비용이 가장 적은 조립순서는 그림 13에서 비용을 줄일 수 있는 요소를 모두 포함한 것으로서 이때의 비용은 24.5이다. 이 경우에는 24.5인 조립순서는 나오지 않았으며 비용이 25.0인 조립순서가 1개, 25.5인 조립순서가 5개 생성되었다.

6. 결론 및 추후 연구방향

가능한 조립작업순서는 부품들간의 기하학적 형상에 의하여 제약을 받는다. 이러한 제약은 AND/OR 우선순위관계로 표현될 수 있고 이를 만족하는 조립순서 중에서 최적의 조립순서를 찾는 문제는 노드 방문순서의 제약이 있는 TSP로 볼 수 있다. 본 논문에서는 조립순서 결정에 뉴럴네트를 이용하여 가능한 조립순서를 자동생성할 수 있는 일체화된 시스템을 구현하였다. 또한 조립비용을 줄일 수 있는 요소를 비용행렬에 반영하여 가능한 조립순서 중에서 경제적인 조립순서를 얻고자 하였다.

자동차용 발전기 조립품에 대해 조립순서를 생성해본 결과 조립 비용의 최저값에 근접한 조립순서를 얻을 수 있었다. 따라서 개발된 시스템은 조립품의 구성이 복잡한 경우에도 비교적 짧은 시간내에 가능하고도 경제적인 조립순서를 효과적으로 생성하는 것으로 평가된다. 부품의 수가 증가하고 비용행렬이 복잡하여 최적해를 구할 수 없는 경우에는 여러개의 가능해를 생성하여 그중 비용이 최소인 조립순서를 선택하면 된다.

추후 연구방향은 다음과 같다.

- 1) 조립작업간의 비용 설정방법에 대한 연구가 계속되어 조립작업간의 비용요소를 체계적으로 신경망 모델에 반영할 수 있어야 한다.

```
1  load precedence();
2  read cost matrix();
3  for run=1 to N
4      initialize neuron's state();
5      do
6          each neuron -> sigmoid function
7          state'change = state'change + new neuron's state - old neuron's state
8          for all neurons
9              all energy_terms = 0
10             for each row has more than one '1'
11                 increase energy_termA
12             for each column has more than one '1'
13                 increase energy_termB
14             for
15                 increase energy_termC
16             for two cities are adjacent
17                 calculate energy_termD
18             for precedence relationship
19                 increase energy_termF
20             sum all energy_terms
21             change neuron's state
22     until state's change is very small
23     chech if sequence is valid
24         if valid
25             append sequence to file and goto 3
26         if not valid
27             skip and goto 3
```

그림 9. 조립순서 생성모듈의 pseudo코드

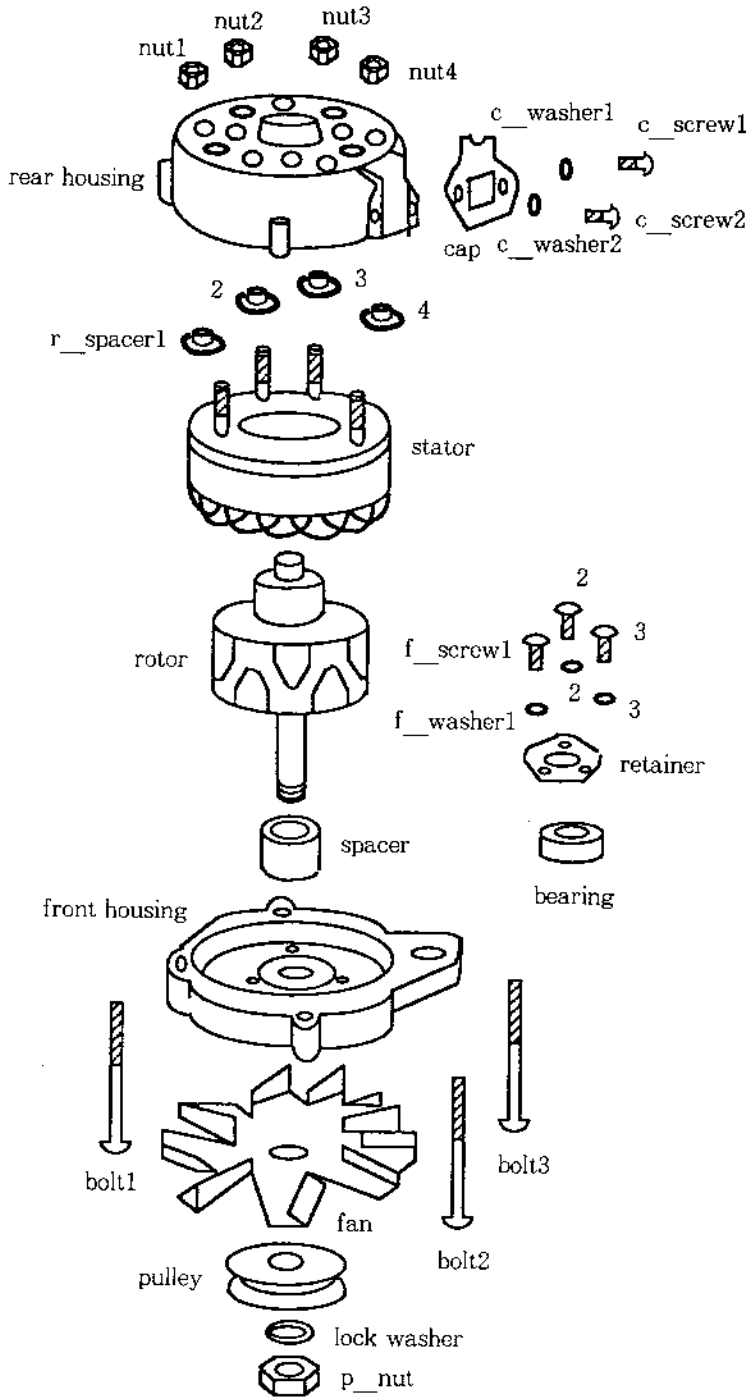


그림 10. 발전기 조립품의 구성도

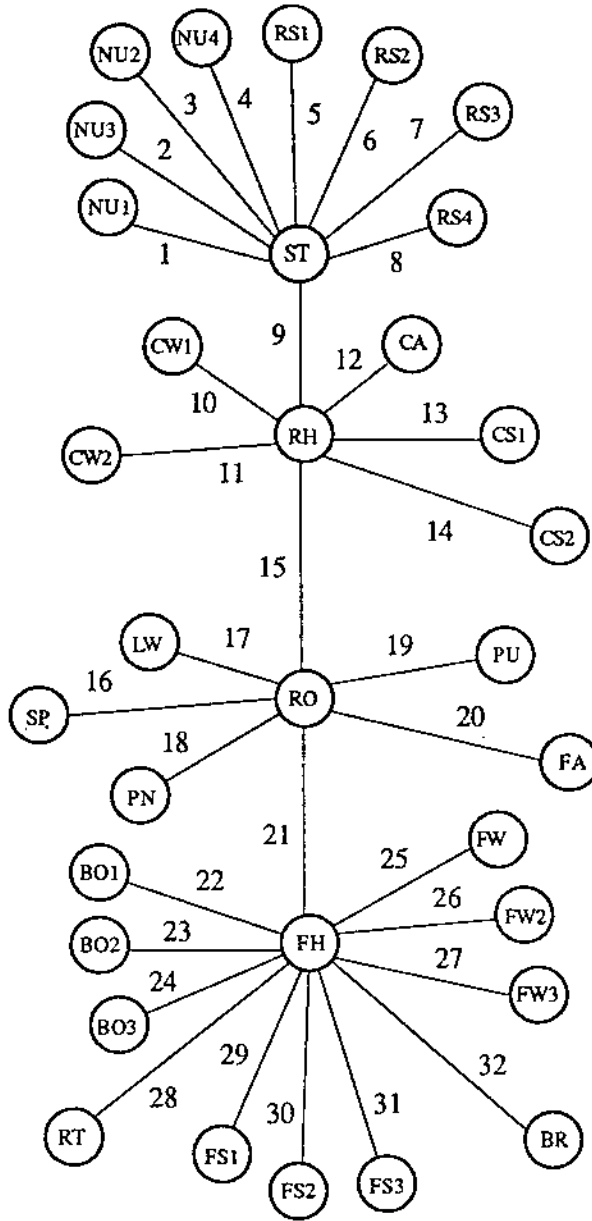


그림 11. 발전기 조립품의 조립선 그래프

- 1 <- (5 & 6 & 7 & 8 & 9)
 2 <- (5 & 6 & 7 & 8 & 9)
 3 <- (5 & 6 & 7 & 8 & 9)
 4 <- (5 & 6 & 7 & 8 & 9)
 9 <- (5 & 6 & 7 & 8)
 10 <- 12
 11 <- 12
 13 <- (10 & 11 & 12)
 14 <- (10 & 11 & 12)
 15 <- (5 & 6 & 7 & 8 & 9)
 16 <- 15
 17 <- (15 & 16 & 19 & 20 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 18 <- (15 & 16 & 17 & 19 & 20 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 19 <- (15 & 16 & 20 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 20 <- (15 & 16 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 21 <- (15 & 16)
 22 <- (5 & 6 & 7 & 8 & 9 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 23 <- (5 & 6 & 7 & 8 & 9 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 24 <- (5 & 6 & 7 & 8 & 9 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32)
 25 <- (28 & 32)
 26 <- (28 & 32)
 27 <- (28 & 32)
 28 <- 32
 29 <- (25 & 26 & 27 & 28 & 32)
 30 <- (25 & 26 & 27 & 28 & 32)
 31 <- (25 & 26 & 27 & 28 & 32)

그림 12. 발전기 조립품의 우선순위

조립 작업	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
1	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
2	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
3	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
9	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
10	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
17	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
18	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
19	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
22	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
23	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
28	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

그림 13. 발전기 조립작업의 비용행렬

표 2. 발전기의 가능한 조립순서

NO.	조립순서	비용
#1	12 5 6 8 32 7 28 10 13 25 29 26 30 11 14 27 31 16 9 4 2 3 1 21 15 20 19 17 22 24 18 23	25.5
#2	32 28 25 29 27 31 16 12 10 13 26 30 5 7 8 9 21 6 15 1 2 11 14 4 3 22 24 23 20 19 17 18	25.5
#3	32 28 25 29 16 26 5 7 27 30 6 31 8 12 10 13 9 21 1 11 2 3 14 4 15 20 19 17 18 22 23 24	28.0
#4	16 8 5 12 10 11 13 32 28 25 26 6 7 27 29 31 14 30 9 15 21 1 4 3 2 22 20 24 19 23 17 18	28.5
#5	6 5 16 7 8 32 28 27 31 12 10 25 29 26 30 11 14 13 9 21 1 4 15 20 2 3 19 17 18 23 22 24	26.0
#6	5 8 16 7 6 12 10 13 32 11 14 28 25 29 26 30 27 31 21 9 15 3 4 1 2 20 19 17 18 23 24	25.5
#7	12 5 8 7 6 16 9 32 28 10 11 25 29 26 30 1 2 14 13 3 27 31 4 15 21 20 19 17 18 22 24 23	26.5
#8	12 10 13 32 28 25 29 26 11 14 27 31 30 5 6 8 7 9 16 21 15 20 19 17 18 23 24 2 3 22 1 4	26.0
#9	16 8 5 32 28 25 26 30 27 6 7 29 31 12 11 14 9 3 1 2 10 13 4 21 15 20 19 17 18 22 23 24	26.5
#10	5 8 6 7 9 16 32 28 25 29 26 27 30 12 31 10 13 15 21 11 14 1 3 4 2 20 19 17 18 24 22 23	25.5
#11	16 32 28 27 31 26 30 5 6 25 29 7 8 12 10 13 11 14 9 21 2 3 4 1 15 20 22 24 19 23 17 18	25.5
#12	32 28 25 29 16 26 27 5 30 31 7 6 8 12 9 21 1 2 10 13 3 11 14 4 15 20 19 17 24 18 23 22	27.5
#13	12 10 14 5 6 11 13 16 7 8 9 32 28 25 29 27 31 15 1 2 26 30 4 3 21 22 23 24 20 19 17 18	26.5
#14	16 32 28 25 27 26 30 29 31 5 6 7 12 10 8 13 11 14 21 9 15 23 22 24 20 19 17 18 3 4 2 1	26.5
#15	5 6 16 32 28 25 26 12 10 11 13 14 27 7 8 29 30 31 9 15 21 4 3 2 22 1 23 24 20 19 17 18	28.5
#16	12 10 11 7 6 5 8 13 14 16 9 31 28 25 26 27 29 30 31 21 15 20 19 17 18 1 4 22 2 23 3 24	29.0
#17	12 11 14 16 32 28 26 27 25 29 30 31 5 7 10 13 6 8 9 3 4 15 21 22 2 1 23 24 20 19 17 18	27.0
#18	7 8 16 32 28 25 29 27 5 6 31 12 10 13 9 21 4 2 26 30 3 11 14 1 15 22 23 20 19 17 18 24	27.0
#19	16 5 12 10 13 32 28 8 6 7 25 29 26 30 27 11 1 2 14 31 9 21 15 3 4 22 23 24 20 19 17 18	26.5
#20	12 10 13 6 7 16 5 8 9 32 28 25 29 27 31 2 1 15 21 26 30 20 19 22 3 4 24 17 23 18 11 14	26.5
#21	32 28 12 10 13 25 29 26 11 14 27 31 30 5 6 8 7 9 16 21 15 20 19 17 18 23 24 2 3 22 14	26.0

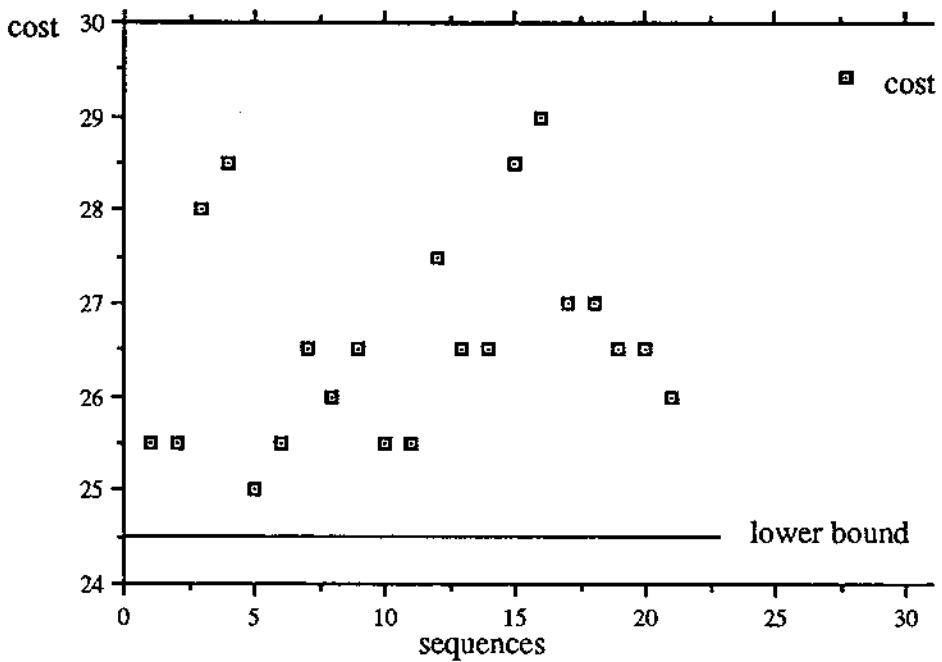


그림 14. 발전기의 조립순서 비용 그래프

- 2) 조립순서 생성시간을 좀 더 단축시킬 수 있는 방안을 모색한다. 이를 위해서는 알고리즘의 개선이나 신경망 하드웨어 이용을 고려해 볼 수 있을 것이다.
- 3) CAD 데이터 베이스와 조립작업의 지식 베이스로부터 우선순위를 자동추출할 수 있는 방법에 관한 연구가 요구된다.

참 고 문 헌

1. 정부영, 손경준, 이경휘(1991), 「조립계획 자동생성에 관한 연구」, IE-TR-91-05, Technical Report, 포항공과대학 산업공학과.
2. Sanderson, A.C., Homem de Mello, L.S., and Zhang, H. (1990), "Assembly Sequence Planning", *AAAI*, pp.62-81.
3. Chen, C.L. Philip (1990), "Neural Computation for Planning AND/OR Precedence-Constraint Robot Assembly Sequences", *IJCNN*, pp.127-142.
4. De Fazio, T.L. and Whitney, D.E. (1987), "Simplified Generation of All Mechanical Assembly Sequences", *IEEE J. of Robotics and Automation*, Vol.RA-3, No.6, pp.640-657.
5. Homem de Mello, L.S. and Sanderson, A.C. (1990), "AND/OR Graph Representation of Assembly Plans", *IEEE Trans. Robotics and Automation*, Vol.6, No.2, pp.188-199.
6. Homem de Mello, L.S. and Sanderson, A.C. (1991), "Representations of Mechanical Assembly Sequences", *IEEE Trans. Robotics and Automation*, Vol.7, No.2, pp.211-227.
7. Bourjault, A. (1984), "Contribution a une approche methodologique de l'assemblage automatise: Elaboration automatique des sequences operatoires," PhD. Dissertation, L'Universite de Franche-Comte.
8. Wolter, J.D. (1989), "On the Automatic Generation of Assembly Plans", *IEEE Intl. Conf. on Robotics and Automation*, pp. 62-68.

9. Liu, Y. and Popplestone, R.J. (1991), "Planning for Assembly from Solid Models", *IEEE Intl. Conf. on Robotics and Automation*, pp.222-227.
10. Tapadia, R.K. and Handerson, M.R. (1990), "Using a Feature-Based Model for Automatic Determination of Assembly Handling Codes", *Comput. and Graphics* Vol.14, No.2, pp.251-262.
11. Hoffman, R. (1989), "Automated Assembly in a CSG Domain", *IEEE Intl. Conf. on Robotics and Automation*, pp.210-215.
12. Huang, Y.F. and Lee, C.S.G.(1989), "Precedence Knowledge in Feature Mating Operation Assembly Planning", *IEEE Intl. Conf. on Robotics and Automation*, pp. 216-221.
13. Lee, S. and Shin, Y.G. (1990), "Assembly Planning Based on Geometric Reasoning", *Comput. and Graphics*, Vol.14, No.2, pp. 237-250.
14. Nnaji, B.O. and Liu, H.C.(1990), "Feature reasoning for automatic robotic assembly and machining in polyhedral representation", *Intl. J. Prod. Res.*, Vol.28, No.3, pp. 517-540.
15. Hopfield, J.J. and Tank, D.W.. (1985), "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics*, Vol.52, pp.141-152.
16. Baldwin, D.F., Abell, T.E., Lui, Man-Cheung Max, De Fazio, T.L., and Whitney, D.E. (1991), "An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Product", *IEEE Trans. Robotics and Automation*, Vol.7, No.1, pp.78-94.
17. 손경준, 정무영(1991), 「뉴럴 네트워크를 이용한 최적 조립순서의 결정」, 대한산업공학회 지 추계학술발표대회 논문집, pp.199-209.
18. Wasserman, Philip D. (1989), *Neural Computing*, pp.93-111.