

인공 뉴럴 네트워크에 의한 FMS 일정관리

FMS scheduling through artificial neural network

양 정 문*
Jung-Mun Yang

문 기 주*
Geeju Moon

김 정 자*
Jung-Ja Kim

Abstract

Recently, neural network is recognized as a new approach to solve jobshop scheduling problems in manufacturing system. Scheduling problem is known to be a difficult combinational explosive problem with domain-dependence variations in general. In addition, the needs to achieve a good performance in flexible manufacturing system increase the dimensions of decision complexity. Therefore, mathematical approach to solve realistic problems could be failed to find optimal or optimal-trending. In this paper a technique with neural network for jobs grouping by job-attributes and Gaussian machine network for generating to near-optimal sequence is presented.

1. 서 론

최근에 현실 세계의 문제 해결을 위한 방법으로 인공지능이 많은 관심을 보이고 있다. 이에 따라 과거에 현장에서 해결할 수 없었던 문제에 대한 즉각적인 조치를 취하는데 많은 성과를 거두고 있다. 실질적으로 생산 제품에 대한 무수히 많은 설계 가능성과 일정 대안이 존재한다. 따라서 제조 시스템에서의 의사 결정은 매우 복잡한 양상을 띄고 있다. 그러므로 동적 특성을 갖고 있는 일정 관리 문제에서 최적 조합을 선택하기란 그렇게 쉬운 일이 아니다.

기업간 혹은 국가간의 경쟁이 치열해짐에 따라 우위를 점유하기 위해서는 새로운 제조 방법 도입의 필요성은 증대되고 있다. 이러한 신 제조 방법을 도입함으로써 과거 생산 방법에 비해 더욱 의사 결정의 복잡성은 더욱 증대될 수 있다. 하지만 이러한 시스템은 제조 시에 방대한 양의 정보를 조정하고 관리하기 위한 빠르고 세련된 의사 결정 시스템의 개발은 필연적이다. 유연성(Flexibility)을 갖고 있는 제조 시스템의 도입과 실현은 지능화된 의사결정수단을 필요로 하고 있다. 이는 완전 자동화 혹은 CIM(computer integrated manufacturing)를 실현하기 위해서는 여러 의사결정요소 간의 지적 시스템을 온라인으로 통합하는 것이 필요하다. 이 같은 통합은 단시간 내에 엄청난 양의 데이터 수집을 가능하며 시간 종속적인 추론(time dependence reasoning)과 적절한 제어를 할 수 있다.

일정관리문제는 생산 설비, 기술적 문제, 생산 요구량 및 제품 품질과 시간 등의 제약 조건 하의 생산 요구를 만족하는 가능한 작업처리순서를 찾는 것이다. 일정관리를 위한 일반적인 접근방법은 최적화, 제약기초분석(constraint-based analysis)[4], 발견적 작업배정기법, CDS(constraint-directed search)[8, 9, 10], 계층적 방법(hierarchical method)[9, 3], 분산적 방법(distributed method)[3]으로 분류할 수 있으며 인공 지능적 접근방법으로는 규칙기초방법(rule based method)[5, 11, 13], 대화적 방법(interactive method), TR(temporal reasoning)[3], 뉴럴 네트워크[3, 5, 6, 7, 12, 19] 등이 있다.

* 동아대학교 산업공학과

특히 뉴럴 네트워크는 Hopfield 네트워크[3], Hopfield 유사 네트워크[5, 6], SA 방법(Simulated Annealing Approach)[19]으로 나눌 수 있다. 이들 기법은 개별적으로 적용할 수 있다. 하지만 보다 효율적인 접근 방법으로는 몇 가지 방법을 혼용하여 사용함으로써 각 접근법의 한계성을 극복하려는 노력을 진행하고 있다.

제조일정문제는 목적함수가 여러 가지 국소 최소치를 갖는 최소화 문제로 분류할 수 있다. SA(Simulated annealing) 방법은 문제 크기가 작은 경우에 한정적으로 해결할 수 있으며, SA 알고리즘에서 시스템의 활동률(the rate of activity) 혹은 시스템의 에너지 수준이 국소 혹은 전역 최소화(global minimum)에 도달할 때까지 온도를 서서히 감소시킨다. 적절한 온도조절방법(annealing schedule)을 선택할 경우에는 jobshop 문제의 최적해 혹은 준 최적해를 찾을 수 있다[3, 5, 6, 7, 12, 19]. 뉴럴 네트워크에 전문가 시스템을 사용하면 뉴럴 네트워크는 패턴을 식별하고, 정의된 패턴에 따라 점화(fire)된다. 일정관리 시스템은 시간에 따라서 의사결정하는 문제이다. 그러므로 인공지능을 기반으로 하는 일정관리를 실현하기 위해서는 시간에 의한 추론과 그 즉시, 제조 환경을 이해하는 능력이 필요하다. 따라서 정적인 패턴 인식이 아니라 사건이 발생하는 순서에 따라 즉각 대응하는 동적인 패턴 인식이 필요하다.

시간 종속적 의사결정능력은 생산계획과 일정계획, 특히 FMS 혹은 CIM 환경을 사용하는 시스템에서 반드시 필요하다. 이를 위해서는 불확실성과 동적 행동을 다룰 수 있는 능력이 필요하다. 따라서 본 논문에서는 이러한 불확실성과 동적 양태에 적용할 수 있는 가우스 머신(Gaussian Machine) 네트워크를 이용하여 jobshop 작업배정문제를 해결하고자 한다.

2. 뉴럴 네트워크를 이용한 일정관리 시스템

2.1 시스템 구조

본 논문에서 사용한 시스템 구조를 살펴보자. figure 1에 보는 바와 같이 지식 베이스와 데이터 베이스를 가진 통합 시스템의 구조로 나눌 수 있다. 지식 시스템 구조는 사건발생순서에 종속적인 작업의 최적순서를 결정하는 가우스 머신 뉴럴 네트워크와 작업을 분류하는 SOFM(Self-organizing feature map)[14] 네트워크로 구성되어 있다.

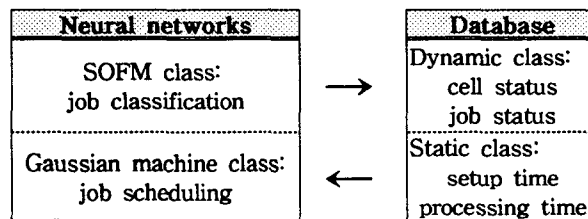


Figure 1. System structure for FMS scheduler system

2.2 적용한 뉴럴 네트워크의 구조

2.2.1 SOFM(Self-organizing feature map)

이 모형에서 사용되는 네트워크의 구조는 다음과 같다. i 는 입력 노드 수로서 작업특성을 나타내는 입력 벡터 변수의 수가 입력된다. 단, $i = 1, 2, \dots, n$ 이다. 또 j 는 출력 노드 수로서 이 문제에서는 작업특성의 그룹이다. 단, $j = 1, 2, \dots, m$ 이다. 또 x_i 는 i 번째 입력 노드를 표현하고 y_j 는 j 번째 출력 노드를 나타낸다. 그리고 각 노드 i 에서 j 까지 연결강도는 w_{ij} 로 나타낸다. 따라서 입력 벡터 $X=(x_1, x_2, \dots, x_n)$ 이고 출력 노드 j 에 대한 연결강도는 $W_j=(w_{1j}, w_{2j}, \dots, w_{nj})$ 이다. 각 출력 노드 y_j 에 대해 어떤 근접 j^* 를 출력 노드 j 와 가장 가까운 노드로 정의한다. 또 y_j 에 근접한 출력 노드의 집합을 근접치(neighbourhood)라 한다. 근접치는 Kohonen[14]이 정의한 것같이 출력 노드 i 에 연결된 출력 노드 j 의 수로 정의할 수 있다. SOFM의 탐색절차는 다음과 같다. 먼저 출력 노드의 선행배열을 초기화하고, 시

행착오 법으로 만족 해를 찾는다. 어떤 시간에 입력되는 입력 벡터는 하나이고 입력 노드와 출력 노드의 연결강도는 입력 벡터의 확률 밀도 함수에 근사하게 가중치의 확률 밀도 함수와 같이 적합하게 변경시킨다. 따라서 입력 벡터는 입력 벡터 변수이고 출력 벡터는 각 작업특성의 그룹이다. 여기서 시간에 따라 감소하는 적합변수(adaptation parameter) $v(t)$ 은 연결강도의 변화량을 조절한다. 그러면 X 가 시간에 따라 변한다면, $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ 로 나타낼 수 있다. 이 절차를 단계적으로 나타내면 다음과 같다.

단계 1. 입력 노드, 출력 노드, 가중치를 초기화한다.

단계 2. 새로운 벡터 $X(t)$ 를 입력한다.

단계 3. 각 출력 노드에 대하여 거리를 계산한다.

$$d_j = \sum_{i=1}^N [x_i(t) - w_{ij}(t)]^2$$

단계 4. d_j 값을 최소로 하는 출력노드 j 를 찾는다. 그리고 각 근접 가중치를 다음과 같이 변경한다.

$$w_{ij}(t+1) = w_{ij}(t) + v(t) [x_i(t) - w_{ij}(t)]$$

단계 5. $v(t)$ 를 변경한다. 단, $0 < v(t) < 1$ 이다.

단계 6. 입력 벡터 값이 없으면 중지하고, 그렇지 않으면 2단계로 간다.

2.2.2 가우스 머신(Gaussian Machine)

가우스 머신[20]은 확률적인 양태를 보이며 등급화된 반응(graded response)을 하는 뉴런 모형이다. 이 모형의 뉴런 입력 부에 첨가되는 랜덤 노이즈(random noise)가 정규 분포를 하기 때문에 가우스 머신이라 부른다. 이 가우스 머신의 활동규칙(activation rule)을 살펴보면 각 뉴런은 입력연결에 따라 다른 뉴런으로부터 출력값 o_j 을 받는다. 이 값을 NEU_i 로 나타낸다. 단 i 와 j 은 $1, 2, \dots, N$ 이다. NEU_i 에서 NEU_j 로 상호연결된 가중치를 w_{ij} 로 정의한다. 또한 이 뉴런은 입력 편의(input bias)를 가지고 있다. 그러므로 NEU_i 의 순입력 net_i 는 다음과 같다.

$$net_i = \sum_{j=1}^N w_{ij} o_j + \theta_i + \varepsilon \quad (1)$$

여기서 ε 은 랜덤 노이즈에 의해 발생하는 오차항이다.

뉴런 NEU_i 는 순입력 net_i 에 의해 활성화되며 NEU_i 의 활동치 a_i 는 다음과 같은 미분 방정식에 의해 변한다.

$$\frac{\Delta a_i}{\Delta t} = -\frac{a_i}{\tau} + net_i \quad (2)$$

여기서 τ 는 뉴런의 시간상수(time constant)이다. 단, Δt 는 0과 1사이의 범위이다.

입력 함수는 다음과 같은 함수에 의해 결정된다.

$$o_j = f(a_j) = \frac{1}{2} (\tanh \frac{a_j}{a_0} + 1) \quad (3)$$

여기서 a_0 를 참조 활성화 수준(reference activation level)이며 첫 번째 시스템 변수이다. 만일 a_0 가 0에 근접하면 이 함수는 McCulloch-Pitts 모형과 같은 함수가 된다.

가우스 머신에서 주의할 점은 입력 함수가 항상 랜덤 노이즈에 의해 영향을 받는다는 것이다. 노이즈 ε 은 평균이 0이고 분산이 σ^2 인 정규분포를 한다. 편차 σ 는 온도인 변수 T 에 종속적이며 다음과 같이 정의한다.

$$\sigma = kT \quad (4)$$

k는 상수로서 $k = \sqrt{\frac{8}{x}}$ 이며 T는 두번째 시스템 변수이다.

가우스 머신의 중요한 부분은 대수적으로 나타내는 에너지 함수이다. 네트워크의 안정성은 이 에너지 함수에 의존한다. 가우스 머신의 에너지 함수는 다음과 같다.

$$\begin{aligned}
 E = & \frac{A}{2} \sum_{X=1}^n (\sum_{i=1}^n o_{Xi} - \gamma)^2 + \frac{B}{2} \sum_{i=1}^n (\sum_{X=1}^n o_{Xi} - \gamma)^2 \\
 & + \frac{(A+B)Q}{4} \sum_{X=1}^n \sum_{Y=1}^n \sum_{i=1}^n \frac{d_{XY}}{d_{average}} \frac{o_{X,i} o_{Y,i-1} + o_{X,i} o_{Y,i+1}}{2} \\
 & + \frac{(A+B)}{2} \sum_{X=1}^n \sum_{i=1}^n o_{Xi} (1 - o_{Xi}) \quad (5)
 \end{aligned}$$

여기서 A, B, C, Q는 양의 상수이며 o_{Xi} 는 i번째 위치에 있는 작업 X를 나타낸다. 식(5)에서 첫 번째 항은 각 작업이 한번에 한 위치에만 있다는 것을 의미하며 두 번째 항은 어떤 한 위치에 오직 하나의 작업만이 일정 잡힘을 나타낸다. 셋째 항은 모든 작업을 처리하는데 소비하는 최소 총 준비시간을 나타낸다. 넷째 항은 주 사선제거 항(principal diagonal elimination term)이라고 하며 이 항에 의해 각 뉴런의 출력이 이전 값을 갖게 된다. 가우스 머신의 에너지 함수에 새로이 도입된 상수인자 $\gamma(\geq 1)$ 는 각 뉴런의 양의 편이(positive bias)를 증가시키기 때문에 장려 인자(encouragement factor)라고 한다.

가우스 머신 네트워크의 중요한 조절은 뾰족화(sharpening)와 열처리(annealing)이다. 뾰족화 조절은 참조 활동화 수준 a_0 의 값을 서서히 감소시키는 것이다. 일반적으로 쌍곡선 함수를 이용한다.

$$a_0 = \frac{A_0}{1 + \frac{t}{\tau_{a_0}}} \quad (6)$$

여기서 A_0 은 a_0 의 초기 값이며 τ_{a_0} 는 뾰족화 조절의 시간 상수이다.

가우스 머신에서 열처리 조절은 최적화에 사용된다. 이 열처리 조절은 뾰족화 조절과 같이 매우 중요하다. 열처리 조절도 뾰족화 조절과 같이 쌍곡선 함수를 이용한다.

$$T = \frac{T_0}{1 + \frac{t}{\tau_T}} \quad (7)$$

여기서 T_0 은 초기 온도, τ_T 는 뾰족화 조절의 시간 상수이다.

가우스 머신의 최적화 단계를 나타내면 다음과 같다.

- 단계 1. A_0 , T_0 와 τ_{a_0} 를 결정한다. 임의의 구조를 생성하고 그 구조의 에너지를 계산한다. $i=1$ 로 둔다.
- 단계 2. 또 다른 새로운 구조를 생성한다.
- 단계 3. 현재 구조와 새로운 구조의 에너지 차인 ΔE 를 계산한다.
 $\Delta E = New - Old$
- 단계 4. 만일 ΔE 가 0보다 적으면 단계 5로 가고 아니면 단계 6으로 간다.
- 단계 5. 일확확률변수 R를 생성하고 $R < P(\Delta E) = \exp(\Delta E / \Delta t)$ 이면 단계 6으로 가고 아니면 새로운 구조를 기각하고 단계 7로 간다.
- 단계 6. 현재 구조와 새로운 구조와 교체한다.
- 단계 7. (6)식과 (7)식과 같이 조절하고 (2)식과 (3)식으로 갱신한다. 만일 안정화되었으면 $i=i+1$ 로 두고 단계 2로 가고, $i=n$ 이면 단계 9로 간다.
- 단계 9. 종료한다.

2.3 뉴럴 네트워크용 입력 벡터의 생성

본 논문에서는 Dagli[3]와 같은 가설 하에 입력 벡터를 생성한다.

- (1) 제품은 n개의 그룹으로 나눌 수 있다.
- (2) 다음 작업이 동일한 그룹이면 작업 중인 머신을 교체할 필요가 없다.
- (3) 입력 벡터용 변수는 다음과 같다.

Machine Status : M_i , i = FMS에서 사용되는 머신 형태
 M_i = X-Y 좌표 상에서 운행되는 머신
 00 = m/c 1
 01 = m/c 2
 10 = m/c 3
 11 = m/c 4

Job status : J_j , j = 일정관리의 주요인

납기 : J_1 $\begin{cases} 0 = \text{긴급} \\ 1 = \text{보통} \end{cases}$ 양 : J_2 $\begin{cases} 0 = \text{적음(100개이하)} \\ 1 = \text{많음(100개이상)} \end{cases}$ 색상수: J_3 $\begin{cases} 00 = 3\text{회이하} \\ 01 = 3\text{회이상 } 5\text{회이하} \\ 10 = 5\text{회이상 } 10\text{회이하} \end{cases}$

(4) 우선 순위 설정

우선 순위는 다음과 같이 결정한다고 가정한다. 첫 번째, 납기가 긴급한 모든 작업을 우선 그룹에 배정하고, 전체 일정에서 최우선 작업이 된다. 이 로트 다음에 긴급하지 않은 작업이 따라온다.

3. 뉴럴 네트워크의 학습 자료

본 논문에서 제안한 입력 벡터 변수를 기준으로 학습 자료를 수집한다. 하지만 고려해야 할 다른 요인으로서는 완성품에 대한 재고가 있다. 본 논문에서는 완성품에 대한 초기 재고는 무시한다. 일반적으로 재고 유지비와 준비 비용간에 상충 관계가 존재하나 본 사례 연구에서는 재고 유지비용보다 준비 비용이 크다고 가정한다. 학습 자료는 SOFM 네트워크 이용하며 입력 벡터는 작업 특성에 따라 분류한다. 작업 분류의 척도로서 가중치를 사용한다. 이 가중치 구조는 table 2와 같다.

뉴럴 네트워크 구조는 입력, 출력 층으로 구성되어 있다. 입력 층에는 5개의 뉴런, 출력 층에 8개의 뉴런이 있다. 작업 분류를 위해 가중치 구조를 설정하고 기계 상태와 작업 상태를 나타내는 입력 벡터를 기초로 시간 요인이 계산하고 작업 클래스는 시간 요인을 기초로 한다. 예를 들면 $M_i M_j J_1 J_2 J_3$ 이 001001의 형태인 입력 벡터이면 긴급을 요하지 않고 소수량, 여러 색상인 작업을 나타낸다. 총시간 요인은 각 상태에 해당되는 가중치의 합으로 계산한다. 따라서 주어진 벡터의 시간 가중치는 10, 5, 40, 15와 10의 합인 80이다.

네트워크 학습 자료는 table 3과 같다.

4. 결론

뉴럴 네트워크의 자료 출력은 시간을 나타낸다. 일정 최적화에 사용한 가우스 머신 네트워크는 최적 혹은 준 최적 작업 순서를 구성한다. 이 작업 순서는 에너지 함수의 변수 A, B, C, D 그리고 Q를 사용하여 구한다. 하지만 때로는 계속 반복하여도 지역 최소치를 벗어나지 못하기 때문에 네트워크가 항상 최적 해에 수렴하지는 않는다.

본 논문에서 적용한 모형은 A=1, B=1, Q=1.15일 때 가우스 네트워크의 결과가 B-B 알고리즘(Branch and Bound Algorithm)으로 풀 정수 계획 법의 결과와도 동일한 결과임이 검증되었다. 본 논문에서 나타난 일반적인 구조는 다른 유형의 일정 관리 문제에 맞게 수정할 수 있으며 또 다른 형태의 학습 방법이나 발전적 방법을 이용하여 다양하게 적용할 수 있으리라 생각된다.

Table 1. Job characteristics

Job	Due Dates	Quadrant	Quantity	Colors
1	1, 7, 14, 28	00	50	1
2	7, 28	00	40	8
3	1, 14, 28	00	500	2
4	4, 28	00	700	7
5	1, 7, 14, 28	01	75	1
6	4,	01	30	8
7	1, 7	01	600	2
8	7, 14, 28	01	500	6
9	8	10	90	4
10	7, 14, 28	10	85	7
11	2	10	200	3
12	4	10	800	6
13	6, 24	11	50	2
14	1, 7, 26	11	75	7
15	2, 28	11	400	3
16	3, 12, 24	11	350	7

Table 2. Weighting scheme for job classification

	M ₁		M ₂		J ₁		J ₂		J ₃	
Value	0	1	0	1	0	1	0	1	0	1
Factor	10	0	5	20	0	40	15	5	5	10

Table 3. Training data for the networks

Job Number	J ₁	M	M	J ₂	J ₃	Time Factor	Classification
1	0	0	0	0	0	35	Class 1
2	0	0	0	0	1	40	Class 1
3	0	0	0	1	0	25	Class 1
4	0	0	0	1	1	30	Class 1
5	0	0	1	0	0	40	Class 2
6	0	0	1	0	1	55	Class 2
7	0	0	1	1	0	40	Class 2
8	0	0	1	1	1	45	Class 2
9	0	1	0	0	0	25	Class 3
10	0	1	0	0	1	30	Class 3
11	0	1	0	1	0	15	Class 3
12	0	1	0	1	1	20	Class 3
13	0	1	1	0	0	40	Class 4
14	0	1	1	0	1	45	Class 4
15	0	1	1	1	0	30	Class 4
16	0	1	1	1	1	35	Class 4
17	1	0	0	0	0	75	Class 5
18	1	0	0	0	1	80	Class 5
19	1	0	0	1	0	65	Class 5
20	1	0	0	1	1	70	Class 5
21	1	0	1	0	0	80	Class 6
22	1	0	1	0	1	95	Class 6
23	1	0	1	1	0	80	Class 6
24	1	0	1	1	1	95	Class 6
25	1	1	0	0	0	65	Class 7
26	1	1	0	0	1	70	Class 7
27	1	1	0	1	0	55	Class 7
28	1	1	0	1	1	60	Class 7
29	1	1	1	0	0	60	Class 8
30	1	1	1	0	1	80	Class 8
31	1	1	1	1	0	85	Class 8
32	1	1	1	1	1	75	Class 8

참 고 자 료

- [1] Blackstone, J. H., Phillips, P. T., and Yogg, G. L., "A state-of-the art survey of dispatching rules for manufacturing job-shop operation", International Journal of Production Research, Vol. 29, pp. 27-45, 1982.
- [2] Bruno , G., Elia, A and Laface, P., "A rule-based system to schedule production", IEEE Computer, Vol. 19, No. 7, pp. 32-40, 1986.
- [3] Dagli, C. H., Lammers, S., Vellanki, M., "Intelligent scheduling in manufacturing using neural network", Expert System, pp. 4-10, Spring, 1991.
- [4] Erscher, J., "Finding some essential characteristics of the feasible solution for a scheduling program", Operation Research, Vol. 20, pp. 27-45, 1982.
- [5] Foo. Y. P., and Takefuji. Y., "Integer linear programming neural networks for job-shop scheduling", Proceedings of the second IEEE International Conference on Neural Networks, pp. 341-348, 1988.
- [6] _____, "Stochastic neural networks for solving job-shop scheduling - Part1 : problem representation", Proceedings of the second IEEE International Conference on Neural Networks, pp. 275-282, 1988.
- [7] _____, "Stochastic neural networks for solving job-shop scheduling - Part2 : architecture and simulations", Proceedings of the second IEEE International Conference on Neural Networks, pp. 283-290, 1988.
- [8] Fox, M. S., Constraint-Directed Search: A case study of job-shop scheduling, Doctoral Thesis, Carnegie-Mellon University, 1983.
- [9] Fox, M. S., Allen, B. and Strohm, G., "Job-shop scheduling: an investigation in constraint-directed reasoning", Proc. 2nd AAAI, Vol. 1, pp. 155-158, 1982.
- [10] Fox, M. S., Smith, S. F., "ISIS - a knowledge-based systems for factory scheduling", Expert Systems, Vol. 1, No. 1, pp. 25-49, 1984.
- [11] Godin, V. B., "Interactive scheduling; historical survey and state of art", AIIE Transactions, Vol. 10, No. 3, pp. 331-337, 1978.
- [12] Hopfield. J., J. and Tank. D., W. "Neural computation of decisions in optimization problems", Biological Cybernetics 52, pp. 141-152, 1985.
- [13] Jae Kyu Lee, Min Soo Suh, "PARMS; a domain-specific knowledge-based parallel machine scheduling system", Expert Systems, Vol. 15, No. 3, pp. 198-214, 1988.
- [14] Kohonen, T., Self-organization and associative memory, Springer-Verlag, Berlin, 1984.
- [15] O'Commor, D. E., "Using expert systems to manage change and complexity in manufacturing In W. Reitman(ed)", Artificial Intelligence Applications for Business, pp. 149-157, 1984.

- [16] Orciuch, E. and Frost, J., "ISA: Intelligence Scheduling Assistant", Proc. 1st AI Applications, pp. 314-320, 1980.
- [17] Smith, S. F., Ow, P. S., "The use of multiple problem decompositions in time constraint planning tasks", Proc. 9th IJCAI, 1985.
- [18] Steffen, M. S. and Greene, T. J., "Automation the scheduling of parallel processors using AI methods", Proc. Industrial Engineering, 1986.
- [19] Vakharia. A., J. and Chang, Y. "A simulated annealing approach to scheduling a manufacturing cell", Proceedings of the ORSA/TIMS Meeting, 1988.
- [20] Yutaka Akiyama, Akira Yamashira, Masahiro, Yuchiro Anzai and Hideo Aiso, "The Gaussian machine: A stochastic neural network for solving assignment problems", Journal of neural network computing, pp. 43-51, winter 1991.