

## JIT(Just-In-Time)하의 작업스케줄링을 위한 인공신경망\*

An Artificial Neural Network for Job Scheduling under Just-In-Time Environment

김수영\*\*, 이문규\*\*\*, 윤찬균\*\*\*

Soo-Young Kim\*\*, Moon-Kyu Lee\*\*\*, and Chan-Kyun Yun\*\*\*

### Abstract

JIT manufacturing requires that materials and services should be done neither too early nor tardy. Therefore, related jobs have to be performed just-in-time. In this paper, we consider a problem of determining a schedule of jobs with a common due date. The objective is to minimize the squared deviation of complete times from the due date. An artificial neural network which is based on the mean field theory is proposed for the problem. Computational results obtained from computer simulations show that the network performs very well.

### 1. 서론

흔히 JIT(just-in-time) 생산방식의 주요 효과로 "최소의 공정재고"와 "필요에 의한 적기생산"을 든다. 국내외를 막론하고 극히 어려운 사업환경에 처해 있는 대다수 업체들이 이와 같은 효과를 기대하여 JIT 생산방식의 도입을 검토하였거나 시행 중에 있다. 물론,

JIT 생산방식이 우리나라 산업실정에 적합한 가에는 의문의 여지가 많으나, 추구하고자 하는 "적기생산"이라는 기본개념은 생산관리 측면에서 분명히 주요한 지표가 된다. 따라서, 앞으로 JIT 생산 시스템이나 이와 유사한 시스템의 운영에 적합한 생산관리 기술의 수요가 크게 증대될 것이 확실하다.

\* 이 논문은 1993년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

\*\* 포항공대 산업공학과

\*\*\* 계명대학교 산업공학과

이러한 생산관리 기술 중에서 가장 핵심분야로 대두되는 것은, 작업(또는 공정) 스케줄링 분야인데, 그 이유는 납기준수가 적기 생산의 선결요건이 되기 때문이다. 특히, JIT 하에서의 스케줄링은 주어진 납기에 맞추기 위하여 목적함수가 이른시간(earliness time)과 지연시간(tardiness time)을 함께 최소화해야 한다는 면에서 기존의 스케줄링 문제들과는 크게 다른 특성을 갖는다.

본 연구에서는, 특히 납기가 공통으로 주어진 작업들의 스케줄링 문제를 다룬다.

이 분야의 기존의 연구는, 설정된 목적함수를 최소로 하는 최적납기를  $d^*$ 라 할 때, 주어진 납기가 그 보다 작거나 큰 경우에 따라 각기 제한문제(restricted problem)와 비제한문제(unrestricted problem)의 두 부분으로 나뉘어진다. 대체적으로 비제한문제가 제한문제보다 풀기가 비교적 용이하다. 따라서, 기존의 연구는 비제한문제에 집중되어 있는데, Bagchi 등[3]은 비제한문제가 완료시간의 분산을 최소화하는 완료시간 분산문제(completion time variance problem)로 변형시킬 수 있음을 보였고, 同문제에 대하여 Vani & Raghavachari[14]는 발견적(heuristic)절차를 제시하였다. 또한, 원래의 비제한문제를 위한 열거식절차(enumerative procedure)가 같은 해에 Bagchi[2]에 의하여 제안되었고, 그 후 Zheng 등[19]이 일반 일차식이나 이차식을 망라한 일반 목적함수를 갖는 문제의 최적해를 보장하는 분지한계법(branch and bound algorithm)을 개발한 바있다.

한편, 제한문제에 대한 연구는 상당히 부

진한 편이다. Bagchi 등[3]이 작업 개시시점에서 지연시간이 없다(즉, zero start time (ZST))는 가정하에 납기가 짧은 문제의 최적해를 구할 수 있는 분지해법(branching procedure)을 제시하였고, 그 후 De 등[4]이 ZST를 가정하지 않은 상태에서 Bagchi 등의 방법을 보완하여 同문제의 최적해를 보장하는 수정된 분지해법을 발표하였다.

인공신경망을 이용한 스케줄링 분야의 연구는 1985년 Hopfield & Tank[10]가 제시한 에너지함수의 개념을 도입한 모형이 순회판매원문제(traveling salesman problem(TSP))를 풀기 위한 도구로서 가능성이 보여진 후, 여러가지 조합최적화문제(combinatorial optimization problem)에의 응용 관련 연구들이 수행되어 왔다. 특히, 작업스케줄링 분야의 신경망 응용연구는 최근에 상당수가 발표되었다.

Foo & Takefuji[6]는 정수선형계획(integer linear programming(ILP))신경망을 응용하여, 주어진 작업들의 작업개시시간들의 합을 최소화하는 스케줄링 문제를 다루었다. Van Hulle[13]는 Foo & Takefuji의 ILP신경망이 때때로 주어진 작업의 제약조건(예를 들어, 선후관계)을 만족하지 않는 해가 나오는 것을 보완하도록 신경망을 확장하여 목적계획 신경망(goal programming network)으로 정형화한 모형을 제시한 바 있다. Vaithyanathan & Ignizio[11]는 stochastic 신경망을 사용하여, 자원제약하에서의 작업스케줄링 문제를 복수의 多次元 Knapsack 문제로 변환하여 해를 구할 수 있음을 보였다. Arizono 등[1]은

작업들의 흐름시간(flow time)의 합을 최소화하기 위한 스케줄링문제에 stochastic 신경망을 적용한 방법을 제시하였다. Yih 등[16]은 로봇의 즉시처리(real-time)스케줄링을 위하여 準-마코프(semi-Makov)의사결정 모형과 역전파(back propagation)신경망모형을 조합한 해법을 제시하였다. 최근에는 통신성능을 최대화하도록 하는 채널할당(channel assignment) 스케줄링에 평균장이론(mean field theory)을 적용한 연구결과[15]가 발표된 바 있다.

이상과 같이 스케줄링 분야의 신경망응용 연구가 다수 수행되었으나, 본 연구의 대상문제인 공통납기하의 작업 스케줄링에 관한 기존의 연구는 그 중요성에도 불구하고 아직까지 발견되지 않는다. 따라서, 본 연구에서는 전자 chip화 할 경우 엄청난 계산속도가 예상되는 Hopfield 신경망모형을 기초로 한 평균장 해법의 적용 가능성을 검토하고자 한다.

## 2. 문제의 정형화

본 연구에서 다루고자 하는 문제는 다음과 같이 정의 된다 :

“각  $n$ 개의 작업에 대한 공정소요시간  $P_j$ ,  $j=1, \dots, n$  와 공통납기  $d$ 가 주어졌을 때, 납기에 대비한 이른시간과 지연시간의 제곱의 합을 최소화 하는 작업순서와 작업완료시간을 결정한다.”

JIT 환경하의 스케줄링 문제에서는 여러가

지 목적함수를 생각하여 볼 수 있으나, 본 연구에서는

$$\text{Min } Z = \sum_{j=1}^n (C_j - d)^2 \quad (1)$$

로 정의되는 목적함수를 대상으로 한다(단,  $C_j$ 는 작업의 완료시간을 의미함). 그 이유는, 가장 일반적으로 사용되는 목적함수가

$$\text{Min } Z = \sum_{j=1}^n [\alpha_j(d - C_j)^2 + \beta_j(d - C_j)^2]$$

의 형태를 취하는데, 이 경우 이른시간과 지연시간에 대한 벌과금 (penalty) 즉,  $\alpha_j$ 와  $\beta_j$ 를 실제적으로 정확히 정의하기가 어렵고, 또한 구한 解도 작업완료시간과 공통납기 사이에 상당히 큰 차이가 발생할 우려가 많다는 단점이 지적되기 때문이다. 따라서, 기존의 여러 연구결과에서도 지적한 바와 같이, 형태 (1)의 목적함수가 보다 더 현실성이 있다고 본다.

이 문제는  $d$ 값의 크기에 따라 앞서 언급한 바와 같이 제한문제와 비제한문제로 구분되는데, 본 연구에서는 ZST를 가정으로 한 제한문제를 다룬다. 참고로, 비제한문제는 본 연구결과를 기초로 하여 향후 추가연구과제로 다루려 한다. 이와 같은 제한문제의 최적해는 다음의 두 property를 만족한다[5]:

Property 1. 임의의 두 작업사이에 지연시간(idle time)이 존재하지 않는다.

Property 2. 최적스케줄의 공정소요시간은

V형태를 취한다. 즉, 최소공정 시간의 작업을 전후로 하여 앞에서는 소요시간이 큰 작업순서로, 뒤에서는 소요시간이 작은 작업순서로 정렬된다.

결정변수로서

$$Z_{xi} = \begin{cases} 1 & \text{작업 } x \text{ 스케줄의 } i\text{번째에 위치할 경우} \\ 0 & \text{기타} \end{cases}$$

로 정의하면, 대상문제는 다음과 같이 비선형 0-1정수계획으로 정형화된다:

$$\text{Min } \sum_{i=1}^n (\sum_{j \leq i} \sum_{x=1}^n P_x Z_{xj} - d)^2 \quad (2)$$

s.t.

$$\sum_{x=1}^n Z_{xi} = 1 \quad \forall i \quad (3)$$

$$\sum_{i=1}^n Z_{xi} = 1 \quad \forall x \quad (4)$$

$$Z_{xi} = 0 \text{ 또는 } 1 \quad \forall x, i$$

이 문제를 기존의 비선형 정수계획 해법을 적용하여 직접 풀기에는 시간적으로나 경제적으로 비효율적이다. 물론 문제 크기가 아주 작은 경우에는 Bagchi 등[3] 이나 De 등[4]이 제시한 분지해법을 이용할 수가 있으나, 문제 크기가 커지면 그 또한 곤란하다. 여기서는 이것을 극복할 수 있는 하나의 해법으로 신경망모형을 고려하는 것이다.

### 3. Hopfield 신경망모형

1984년 Hopfield[9]는 기존의 그의 신경망모형[8]이 다음의 에너지함수를 최소화할 수 있음을 보였다.

$$E = -(1/2) \sum_{i=1}^m \sum_{j=1}^m T_{ij} V_i V_j - \sum_{i=1}^m V_i I_i \quad (5)$$

여기서  $m$  = 신경망의 뉴런수

$T_{ij}$  = 뉴런  $i$ 와 뉴런  $j$ 와의 연결상수

$V_i$  = 뉴런  $i$ 의 출력값

$I_i$  = 뉴런  $i$ 의 bias

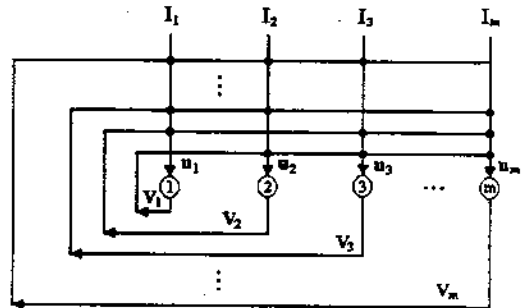


그림 1. Hopfield 신경망의 구조

즉,  $T_{ij}$ 가 대칭이면( $T_{ij} = T_{ji}$ ), 뉴런  $i$ 의 입력값  $U_i$ 의 미분방정식으로 표현되는 전류동적방정식(circuit dynamic equation)

$$\frac{du_i}{dt} = -u_i + \sum_{j=1}^m T_{ij} V_j + I_i \quad (6)$$

에 따라 진화를 거치면, 최종적으로  $E$ 의 지역최소점(local minimum point)으로 수렴하게 된다는 것이다. Hopfield 신경망의 처리 유닛(processing unit) 또는 뉴런들은 그림 1에서 보는 바와 같이 서로 완전연결(fully

connected) 되어 있는데,  $u_i$ 와  $V_j$ 의 관계는  $V_j$ 가 마다입력값  $u_i$ 를 활성화함수(activation function)  $g(\cdot)$  통하여 전이시킨 것으로서,  $V_j = g(u_j)$ 로 나타내진다. 일반적으로 사용되어지는  $g$ 함수는 시그모이드(sigmoid)함수나  $\tanh(\cdot)$ 을 사용한다. Hopfield & Tank[10]는 1년 뒤에 위 성질을 이용하여 Hopfield 모형을 TSP에 응용한 과정과 그 결과를 발표하였다. 이 때 사용된 에너지함수는, 제약식을 만족시키지 못하는 것에 대한 벌과금(penalty)과 원 문제의 순수 목적함수의 합으로서 표현되었다.

#### 4. 평균장을 이용한 신경망 알고리즘

대상이 되는 최적화문제를 Hopfield 모형으로 변환시키기 위해서는 우선 문제의 결정변수를 해당되는 뉴런으로 정의하고, 제약식을 벌과금 형태로 원래의 목적함수 (2)에 추가시켜 에너지함수화 하여야 한다. 따라서 2절에서 정형화된 문제의 결정변수,  $Z_{xi}$ 를 뉴런  $x_i$ 의 출력값  $V_{xi}$ 로 대응시키고(총 뉴런의 갯수  $m = n^2$ ), 제약식 (3) 와 (4)을 비용함수화 하면 에너지함수는

$$E = \frac{A}{2} \sum_i^n (\sum_x^n V_{xi} - 1)^2 + \frac{B}{2} \sum_x^n (\sum_i^n V_{xi} - 1)^2 + C \sum_{i=1}^n (\sum_{j \leq i}^n \sum_{x=1}^n P_{xZ_{xi}} - d)^2 \quad (7)$$

이 된다(A, B, C는 각 항의 상대적인 가중치 상수). 여기서 Hopfield 모형을 직접 적용하려면 (5)과 (7)의 에너지함수를 서로 비교하여서 연결상수  $T_{ij}$ 와 bias  $I_i$ 를 구한 다음, 동적방정식 (6)을 풀어서 신경망의 해를 얻게

된다. 이러한 Hopfield 기본모형은 자체적으로 몇가지 문제점을 안고 있다. 그 중의 하나는 에너지함수 (7)식에서 각항의 계수 A, B와 C의 값을 적절하게 설정하지 못하면 최적해를 구할 수 없게 된다는 점과, 또 하나는 가능해(feasible solution)를 생성해 내는 계수값의 영역이 매우 한정되어 있고 지역최소점에 빠지는 경우가 빈번히 발생한다는 점 등이다. 이를 극복하기 위하여 많은 연구가 수행되어 왔는데 가장 효과적인 것 중의 하나가 Van den Bout & Miller[12]가 제시한 방법으로 계수들을 정규화하고 동시에 어닐링(annealing)과 평균장이론을 조합하여 확장 적용한 것이다. 그 결과 지역 최소점에 수렴되는 현상이 현저히 줄었고, 가능해를 생성하는 계수의 범위가 매우 커지게 되어 문제 적용에 다양성을 제공하였다. 따라서, 여기서는 이들의 연구결과를 우리의 스케줄링 문제에 적용하여 보고, 그 결과로부터 얻어진 관찰사항을 검토하여 보기로 한다.

##### 4.1 평균장 이론

자성체(magnetic material)는 간단하게 말하면 원자마그넷(atomic magnet)의 집합으로 구성되어 있다고 할 수 있는데, 원자마그넷은 외부(external)와 내부(internal)로부터 받는 磁場(magnetic field)의 크기에 따라 자력의 방향을 갖게된다. 위와 아래의 두 방향만을 갖는 자성체를 가정하여 보면 시스템의 변화 구조가 Hopfield 신경망 모형과 상당히 유사하다.  $W_{ij}$ 를 원자 마그넷  $j$ 가 원자마그넷  $i$ 에 미치는 영향의 세기이고,  $S_i$ 는 원자마그넷  $i$ 의 자력의 방향이며  $h_i^{ext}$ 는 외부에서 받는 자장의 크기라고 정의하면, 원자마그넷  $i$ 에 마

치는 자장의 크기는

$$h_i = \sum_j W_{ij} S_j + h^{ext}$$

의 식으로 표현되고, 이에 해당되는 에너지는

$$H = -(1/2) \sum_i \sum_j W_{ij} S_i S_j - h^{ext} \sum_i S_i$$

이다[7]. 즉,  $h_i = -(\partial H / \partial S_i)$ 가 성립된다.

이러한 시스템의 변화에 영향을 주는 또 하나의 요인은 온도이다. 각 원자마그네틱은  $h_i$  값에 따라 자력의 방향을 유지하려고 하나 온도가 높으면 그것을 방해하는 힘이 강하게 된다. 결국, 어떤 원자마그네틱의 磁性은 시시각각 변하는 주위의 다른 원자마그네틱으로부터 받는 자장과 온도의 두 요소에 따라 결정된다. 따라서, 이러한 변화구조를 수리적으로 정확하게 풀기가 어려우므로, 다른 원자마그네틱의 자성을 그 평균값으로 고정시킨 상태에서 임의의 원자마그네틱의 자성을 조정하는 절차를 통하여 해결하고자 하는 것이 평균장이론이다. 그러므로, 평균장은  $S_j$ 의 평균을  $\langle S_j \rangle$ 라 할 때

$$\langle h_i \rangle = \sum_j W_{ij} \langle S_j \rangle + h^{ext}$$

으로 표현된다. 여기서  $\langle S_j \rangle$ 는 정의된 활성화 함수에 따라 달라질 수 있으나, 다음과 같이 통상, 온도 T를 모로 한 Boltzmann 분포를 따른다:

$$\langle S_j \rangle \propto \exp(\langle h_j \rangle / T).$$

여기서 우리 문제를 이 이론에 적용하여 보자. 우선 (7)에 정의된 에너지 함수의 항목 중에서 제약식에 해당되는 첫번째 항은 다음과 같이 다시 표현될 수 있다:

$$E_1 = (\lambda/2) \sum_{i=1}^n \sum_{x=1}^n \sum_{y \neq x}^n V_{xi} V_{yi}$$

여기서,  $\lambda$ 는 원래의 목적함수 (2)에 대비한 제약식에 주어지는 가중치계수이다.  $E_1$ 은 두개의 작업이 동일한 스케줄순서에 할당되는 것에 대한 벌과금으로서 제약식(3)에 상응된다. 마찬가지로, 제약식 (4)에 상응하는 에너지함수를 정의할 수 있으나, 이 경우는 신경망 알고리즘에서 이 제약식을 만족하도록 기술적으로 처리할 수 있으므로 생략한다.

다음으로, 본 문제의 목적함수를 정리하면

$$\begin{aligned} E_2 &= \sum_{i=1}^n (\sum_{j \leq i} \sum_{x=1}^n P_x V_{xi} - d)^2 \\ &= \sum_{i=1}^n (\sum_{j \leq i} \sum_{x=1}^n P_x V_{xi})^2 - 2d \sum_{i=1}^n \sum_{j \leq i} \sum_{x=1}^n P_x V_{xi} + nd^2 \\ &= \sum_{x=1}^n \sum_{i=1}^n \sum_{y=1}^n \sum_{j=1}^n N_{ij} P_x P_y V_{xi} V_{yj} - 2d \sum_{x=1}^n \sum_{i=1}^n (n-i+1) P_x V_{xi} + nd^2 \end{aligned}$$

$$\text{단, } N_{ij} = \begin{cases} n-i+1 & i \geq j \text{ 일 경우,} \\ n-j+1 & \text{기타.} \end{cases}$$

따라서,  $E_2$ 에서 상수항을 제외하고  $E_1$ 과 합하면 에너지함수는

$$E = (\lambda/2) \sum_{i=1}^n \sum_{x=1}^n \sum_{y \neq x}^n V_{xi} V_{yi} +$$

$$\sum_{x=1}^n \sum_{i=1}^n \sum_{y=1}^n \sum_{j=1}^n N_{ij} P_x P_y V_{xi} V_{yj}$$

$$- 2d \sum_{x=1}^n \sum_{i=1}^n (n-i+1) P_x V_{xi}$$

가 된다. 여기서 위의 평균장이론을 적용하기 위해서는

$$\langle S_j \rangle = V_j$$

$$\langle h_i \rangle = E_{xi} = -\lambda \sum_{y \neq x} V_{yi} - 2 \sum_{y=1}^n \sum_{j=1}^n N_{ij} P_x P_y V_{yj} + 2d(n-i+1) P_x$$

를 각각 대입하여 생각하면 된다. 그리고, 제약식 (4)을 만족시키기 위하여,  $V_{xi}$ 를 작업  $x$ 가 스케줄 위치  $i$ 에 배정될 확률로 해석하여 매 iteration마다

$$V_{xi} = \exp(E_{xi}/T) / \sum_{j=1}^n \exp(E_{xj}/T) \quad (8)$$

로 수정한다. 다음 절에서는 이를 활용하여 온도  $T$ 가 고정되었을 때 해를 구하는 과정을 소개한다.

#### 4.2 고정온도 해법 (Fixed-Temperature Procedure(FTP))

유니트  $xi$ 의 출력값은 (8)식에 표현된 바와 같이 평균장  $E_{xi}$ 와 온도  $T$ 에 영향을 받는다. 온도  $T$ 가 커지면  $V_{xi}$ 값이  $E_{xi}$ 에 별 영향을 받지 않고 모든  $i$ 에 대하여 거의 같아지게 된다. 결국 비가능해(infeasible solution)로 신경망이 수렴될 확률이 증대되는 것이다. 온도가 감소하면 평균장이 큰  $V_{xi}$ 의 값이 다른 것에 비하여 상대적으로 커지게 되어, 작업  $x$ 가 특정  $i$ 에 배정될 가능성이 상승하여 가능

해가 증가하게 된다. 따라서, 높은 온도에서부터 점차 내려가면 특정 온도부터는 가능해가 발생하기 시작하게 되는데, 이것을 臨界 온도(critical temperature,  $T_c$ )라 부른다. 주어진 문제에 따라 임계온도를 알 수 있으면, 그 온도에서 고정시킨채로 신경망의 수렴해를 구하면 되지만, 그것을 미리 알 수 없기 때문에 실제로는 시행착오(trial and error)에 의하여 적정온도를 선택하게 된다. 온도가  $T$ 로 주어졌을 때 수렴해를 구하는 절차를 요약하면 다음과 같다:

#### FTP(T):

$T \leftarrow T_0$  (임의로 주어진 온도);

행렬  $V = (V_{xi})_{n \times n}$ 의 초기화;

DO {

작업  $x$ 를 임의로 선택;

모든  $i$ 에 대하여 평균장  $E_{xi}$ 계산:

$V_{xi} = \exp(E_{xi}/T) / \sum_{j=1}^n \exp(E_{xj}/T)$ 로 수정;

}

WHILE  $|| V_{new} - V_{old} || > \epsilon$ .

여기서  $\epsilon$ 는 매우 작은 실수이고  $|| \cdot ||$ 는 norm을 의미한다.

#### 4.3 평균장 어닐링(Mean-Field Annealing (MFA)) 해법

앞에서 지적한 바와 같이 FTP를 이용하기 위해서는 온도가 적절히 주어져야 한다. 이러한 문제점을 극복하기 위해서 개발된 것이 어닐링 기법으로서 1985년 Hopfield & Tank [10]에 의하여 최초로 제안된 후 여러 학자들[15, 17, 18]에 의하여 그 성능과 타당성이 검증되어 왔는데, 대표적인 것이 Van den

Bout & Miller[12]의 TSP 응용연구이다. 초기온도를  $T_0$ 로 주어졌을 때의 MFA해법을  $MFA(T_0)$ 라 하면 그 절차는 아래와 같이 요약된다.

MFA( $T_0$ ):

$T_{new} = T_{old} = T_0$  (초기 설정온도)로 초기화;

행렬  $V$  초기화;

DO {

    FTP( $T_{new}$ ) 수행;

$T_{new} = \alpha T_{old}$ ;

}

UNTIL  $\|V\|^2/n \approx 1$ .

위 해법에서 온도수정에 사용되는 상수  $\alpha$ 는  $0 < \alpha < 1$  사이의 값으로서 MFA의 속도를 조절하는 기능을 갖는다.

## 5. 컴퓨터 실험결과

### 5.1 예비 실험

앞절에서 기술된 2가지 해법을 스케줄링 문제에 적용한 후, 그 성능을 평가하기 위하여 총 작업수가 10개( $n=10$ )인 서로 다른 유형의 두 문제를 만들었다. 표 1에서 보는 바와 같이 문제 A는 작업공정소요시간의 분포가 심하게 양쪽으로 치우쳐져 있고, 문제 B는 그 반대로 일정한 크기로 증가하는 값을 갖는다. 이렇게 극단적인 문제유형을 선정한 이유는, 앞절에서 기술된 해법들의 문제에 따른 성능차이를 알아내기 위함이다. 또한 여러 유형의 문제에 적용할 수 있는 확장성을 확보하기 위하여 공정소요시간을 0과 1사이

의 값으로 정규화하였다. 주어진 스케줄링 문제의 최적해가 언제든지 ZST을 갖기 위해서는 다음의 조건을 만족해야 한다: 공정소요시간을  $P_1 \leq P_2 \leq \dots \leq P_n$ 의 올림차순으로 정렬하였을 때,

$$d \leq \underline{d} = (1/n) \sum_{x=1}^n (n-x+1)P_x$$

이 조건은 (i)식의 목적함수를 초기지연시간  $w$ 를 포함하여 일반화하면

$$Z = \sum_{j=1}^n (C_j + w - d)^2$$

와 같이 되는데, 여기서  $z$ 를 결정변수  $w$ 로 편미분하여 최적  $w$ 를 구한 다음 그 값이 0보다 크지 않은 조건으로부터 얻은 것이다.

#### 5.1.1 FTP 적용 결과

$\lambda = 5, 10, 20, 50, 100, 200, 1000$ 의 각 경우에 대하여, 시행착오방법으로 임계온도를 구한 후 각 임계온도에서 FTP를 적용한 결과가 표 2에 주어져 있다. 이 표의 수치들은  $(\lambda, T)$ 조합의 각 경우에서 20개의 해를 구하여 평균한 값들이다. 같은  $(\lambda, T)$ 로 주어졌다 하더라도, 난수(random number)가 요구되는 임의적인 풀이과정이 존재하기 때문에 얻어진 해가 달라 질 수 있는 것이다.

표 2에서 보는 바와 같이  $\lambda$ 가 고정된 경우 임계온도를 초과하면 가능해가 나오지 않고, 임계온도 이하에서 가능해가 나타나지만 임계온도로부터 멀어질수록 열등한 결과를 초래함을 알 수 있다. 표 1의 결과로부터 추가로 얻어진 관찰결과를 요약하면 다음과 같다:



표 1. 평가 대상 문제

| 문제 | 공정소요시간         |                |                |                |                |                |                |                |                |                 |     |          |                |
|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----|----------|----------------|
|    | P <sub>1</sub> | P <sub>2</sub> | P <sub>3</sub> | P <sub>4</sub> | P <sub>5</sub> | P <sub>6</sub> | P <sub>7</sub> | P <sub>8</sub> | P <sub>9</sub> | P <sub>10</sub> | d   | <u>d</u> | z <sup>*</sup> |
| A  | 0.05           | 0.10           | 0.15           | 0.20           | 0.25           | 0.80           | 0.85           | 0.90           | 0.95           | 1.0             | 1.8 | 1.85     | 21.59          |
| B  | 0.1            | 0.2            | 0.3            | 0.4            | 0.5            | 0.6            | 0.7            | 0.8            | 0.9            | 1.0             | 2.0 | 2.20     | 23.88          |

@ 최적해에서의 목적함수 (1)의 값.

- 1)  $\lambda$ 가 증가할수록  $T_c$ 는 커진다.
- 2)  $\lambda$ 가 증가함에 따라  $T_c/\lambda$ 의 비율은 감소하다가 증가하는 경향이 있다.
- 3) 가능해를 생성해 내는  $\lambda$ 의 폭이 매우 크다.
- 4) 임계온도 근처에서는 최적해 대비 편차가 1%이내인 좋은 해를 얻을 수 있다.
- 5) 전체적으로 볼 때 문제유형에 따른 성능상의 차이는 없다. 그러나, 임계온도에서의 수렴해 중에서 최적해의 숫자는 문제 B가 문제 A에 비하여 월등히 많다. 이것으로 보아, 공정소요시간의 분포가 편중되어 있는 경우는, 수렴해가 최적해에 매우 근접되어 있기는 하나 최적해 자체로는 수렴하기가 어렵다는 것을 알 수 있다.

5.1.2 MFA해법 적용 결과

앞절의 관찰결과로부터 우리는 주어진 문제에 대한 임계온도를 정확하게 알기만 하면 FTP를 적용하여 우수한 해를 구할 수 있다고 판단할 수 있다. 그러나, 문제를 풀어 보기 전에 정확한 임계온도를 분석적으로 추정하기가 어려우므로, FTP를 이용하기 위해서는 시행착오법에 의존할 수 밖에 없다. 이러한 단점은 MFA를 사용하면 어느 정도 극복할 수 있다.

그림 2는  $\lambda = 50$ 일 때  $T_0 = 150$ 으로 문

제 A에 MFA해법을 적용하여 얻은 결과로서, 온도가 감소함에 따른 에너지의 변화를 극명하게 보여주고 있다. 이 경우의 임계온도는 9.0(표 1. 참조)인데, 이 온도 근처에 이르러 에너지가 급격히 떨어짐을 알 수 있다. 따라서, 이 성질을 이용하여, 문제가 주어지면 MFA해법을 우선 적용하여 임계온도를 대략 추정한 후 FTP를 임계온도 근처에서 반복 적용하던지 아니면, 추정치에 근접된  $T_0$ 값을 사용하여 MFA해법을 재적용하는 방법들을 고려할 수 있을 것이다.

그림 3은, 같은 결과를 가지고  $E_1$ 과  $E_2$ 의 총에너지 E에 대한 비율로 표시하여 온도에 따른 변화를 알 수 있도록 표현한 것이다. 여기서는 임계온도 9.0근처에서부터 온도가 감소함에 따라 순수 목적함수인  $E_2$ 값의 비율이 점차 상승하여 가능해로 수렴된 상태( $T=2.0$ )에서는 100%에 도달함을 보여주고 있다. 표 3에서는 초기온도( $T_0$ )가 다르게 주어지는 경우에 MFA해법을 적용한 결과를 FTP의 것과 비교하고 있다. 이 표의 수치는 각 경우에서 얻어진 20개의 평균값을 의미한다. 이로부터 얻어진 관찰결과는 다음과 같이 요약된다:

- 1) FTP와 MFA해법을 성능면에서 비교하여 보면,  $\lambda$ 의 값이 매우 크거나 작은 경우를 제외하면 차이가 없다. 따라서,  $\lambda$ 의 값이 적

표 2. ( $\lambda$ , T)의 각 조합에 대한 FTP 적용 결과

a. 문제 A

| $\lambda \setminus T$ | 0.6   | 1.7   | 3.6   | 9.0   | 16.8  | 30.6  | 128.5 | $N_{opt}^{\oplus}$ | TC / $\lambda$ |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|--------------------|----------------|
| 1000                  | N.A.  | 36.49 | 34.79 | 34.92 | 34.91 | 34.98 | 21.82 | 0                  | 7.78           |
| 200                   | 33.72 | 32.22 | 29.92 | 26.93 | 25.42 | 21.80 |       | 0                  | 6.54           |
| 100                   | 33.42 | 30.84 | 27.51 | 24.14 | 21.83 |       |       | 0                  | 5.95           |
| 50                    | 31.68 | 27.95 | 24.99 | 21.94 |       |       |       | 0                  | 5.56           |
| 20                    | 27.94 | 24.34 | 22.38 |       |       |       |       | 0                  | 5.56           |
| 10                    | 24.58 | 22.42 |       |       |       |       |       | 0                  | 5.88           |
| 5                     | 23.05 |       |       |       |       |       |       | 0                  | 8.33           |

b. 문제 B

| $\lambda \setminus T$ | 0.6   | 1.6   | 3.7   | 8.9   | 16.7  | 30.3  | 128.7 | $N_{opt}^{\oplus}$ | TC / $\lambda$ |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|--------------------|----------------|
| 1000                  | N.A.  | 37.99 | 37.25 | 36.15 | 36.54 | 34.08 | 24.18 | 5                  | 7.77           |
| 200                   | 36.07 | 34.55 | 34.37 | 30.38 | 27.44 | 23.97 |       | 11                 | 6.60           |
| 100                   | 35.80 | 33.87 | 30.80 | 27.23 | 23.96 |       |       | 12                 | 5.99           |
| 50                    | 33.42 | 32.32 | 27.69 | 24.02 |       |       |       | 12                 | 5.62           |
| 20                    | 30.75 | 27.25 | 24.02 |       |       |       |       | 7                  | 5.41           |
| 10                    | 27.36 | 25.42 |       |       |       |       |       | 0                  | 6.25           |
| 5                     | 25.40 |       |       |       |       |       |       | 0                  | 8.33           |

@ 최적해가 얻어진 횟수

당한(?) 수준이라고 판단될 경우에는 MFA 해법을 사용하는 것이 시행착오에 의하여 임계온도를 찾는 노력과 시간을 절약할 수 있다.

2) 초기온도가 낮을수록 MFA해법의 결과가 우수하다. 그러므로, 가능한  $T_c$ 에 가까운 초기온도를 사용하는 것이 바람직하다.

5.2 본 실험

앞 절에서는 FTP와 MFA의 해법의 특성을 알아 보기 위하여 규모가 아주 작은 경우, 대표적인 두 문제에 대한 실험을 수행하였다. 여기서는 일반적인 문제해결에 사용될 수 있는 MFA의 해법이 문제 크기에 따라 어떠한

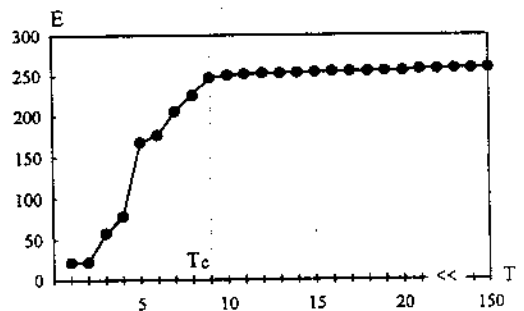


그림 2. 어닐링 온도에 따른 에너지의 변화 ( $\lambda=50$  이고  $T_0=150$  일 때)

성능을 나타내는지를 파악하기 위하여, 다양한 문제를 임의로 생성하여 MFA해법을 적용하여 풀어 보았다.

문제의 크기 n은 10, 15, 20, 25 의 네가지

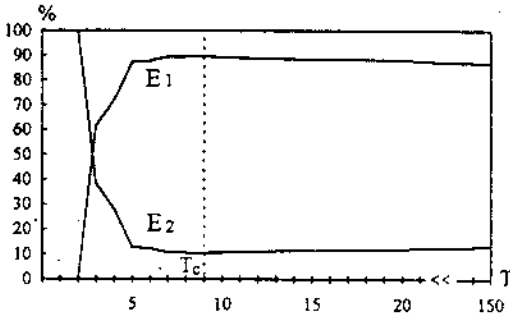


그림 3. 어닐링 온도에 따른 E<sub>1</sub>과 E<sub>2</sub>의 비율 변화  
(λ=50 이고 T<sub>0</sub>=150 일 때)

공통납기는 각각 다음의 확률분포를 따르는 것으로 가정하였다:

$$P_x \sim U(0, 1), P_x \neq 0,$$

$$d \sim U(0, \underline{d}).$$

여기서 U(a,b)는 확률변수의 하한과 상한이 각각 a, b인 均等분포(uniform distribution)를 가리킨다. 특히, 납기를  $\underline{d}$  이하로 설정한 이유는 어떠한 경우에도 제약문제가 되도록 하기 위함이다.

경우를 고려하였다. 각 경우에 대하여 25 문제식을 임의로 생성하였는데 공정소요시간과

이와 같이 하여 생성된 문제를, 편의상 λ

표 3. FTP(T<sub>c</sub>)와 MFA(T<sub>0</sub>)를 적용한 결과

a. 문제 A

| λ              | FTP(T <sub>c</sub> ) | T <sub>0</sub> = 150 | MFA(T <sub>0</sub> ) |       |
|----------------|----------------------|----------------------|----------------------|-------|
|                |                      |                      | 300                  | 450   |
| 1000           | 21.82*(128.5)        | 23.05                | 23.10                | 25.86 |
| 200            | 21.80*(30.6)         | 22.19                | 23.02                | 24.22 |
| 100            | 21.83(16.8)          | 21.77*               | 21.84                | 22.94 |
| 50             | 21.94*(9.0)          | 21.96                | 21.99                | 22.71 |
| 20             | 22.38(3.6)           | 21.75*               | 21.81                | 21.85 |
| 10             | 22.42*(1.7)          | 22.56                | 23.38                | 26.45 |
| 5 <sup>ⓐ</sup> | 23.05(0.6)           | -                    | -                    | -     |

b. 문제 B

| λ              | FTP(T <sub>c</sub> ) | T <sub>0</sub> = 150 | MFA(T <sub>0</sub> ) |       |
|----------------|----------------------|----------------------|----------------------|-------|
|                |                      |                      | 300                  | 450   |
| 1000           | 24.18*(128.7)        | 24.68                | 26.40                | 26.96 |
| 200            | 23.97*(30.3)         | 24.54                | 24.97                | 25.35 |
| 100            | 23.96*(16.7)         | 24.13                | 25.36                | 26.34 |
| 50             | 24.02(8.9)           | 23.98*               | 24.07                | 24.65 |
| 20             | 24.02*(3.7)          | 24.08                | 24.11                | 24.26 |
| 10             | 25.42*(1.6)          | 25.78                | 26.08                | 27.92 |
| 5 <sup>ⓐ</sup> | 25.40(0.6)           | -                    | -                    | -     |

\* 표의 수치는 각 행에서 최소값을 나타냄.

ⓐ 이 경우는 해가 구해지지 않았음.

= 250,  $T_0 = 40$  으로 놓고 MFA에 적용하여 풀었고(각 문제당 10개의 해를 구하여 평균값을 취함) 그 성능을 평가하기 위하여 Bagchi 등[3]이 제시한 분지해법을 이용하여 구한 최적해와 비교하였다. 최적해법과 MFA 해법 모두 Turbo-C를 이용 프로그래밍하였고 사용된 컴퓨터는 IBM-PC 80486DX2/66MHz이다. 표 4에서는 각 문제 크기에 대하여 생성된 25문제의 결과를 평균한 목적함수의 값, 오차율 및 해를 구하는데 소요된 CPU 시간을 요약하였다. 여기서 오차율은 다음과 같이 정의 된다:

오차율=

$$\frac{100\%(\text{MFA 해의 목적함수의 값} - \text{최적해의 목적함수의 값})}{\text{최적해의 목적함수의 값}}$$

평균 오차율은 당연히  $n$ 이 증가함에 따라 커지게 되나  $n=25$ 인 경우에도 3.62%로서 상당히 우수한 결과를 나타내었다.  $n>25$ 인 경우에는 현재의 80586급이하의 컴퓨터에서 최적해 계산시간이 너무 많이 걸리기 때문에 고려할 수 없었다.

일반적으로, Hopfield 기반의 신경망들은 고속의 병렬처리가 가능한 전자 chip化를 전제로 하기 때문에 소요시간을 비교하는 것은 무의미하다(전자 chip을 이용하면 극히 짧은 시간내에 해를 구할 수 있음). 그러나, 여기서는 chip이 개발되기 전에 에뮬레이터(emu-late)하는 측면에서 소프트웨어를 이용할 수도 있기 때문에 최적해법과의 시간상의 차이를 비교하여 보았다. MFA 해법이 수렴해를 얻기 위하여 다수의 반복작업을 수행하기 때문에 소규모( $n \leq 15$ ) 문제에서는 최적해 보

다 시간이 많이 걸리나 문제 규모가 커질수록 분지해법에 비하여 월등함을 알 수 있다.  $n=25$ 인 경우에는 소요시간이 분지해법 대비 1.62%에 불과하였다. 이러한 경향은  $n$ 이 증가할수록 현저해질 것으로 보이기 때문에 신경망 에뮬레이터로서 소프트웨어가 사용될 수도 있음을 보여주는 것으로 해석된다.

JIT 제조시스템에서는 작업 스케줄링에 있어서 특히 즉시성이 요구되기 때문에 시간상의 제약을 많이 받게 되는데, 이와 같이 빠른 시간내에 우수한 해를 제공하는 인공지능망을 사용하므로써 이러한 제약을 상당부분 해소할 수 있을 것이다.

지금까지의 컴퓨터실험 결과로부터 본 연구에서 제시된 JIT하의 작업 스케줄링을 위한 MFA해법이 비교적 규모가 큰 문제에 적용하였을 때에도 만족할만한 성능을 보인다는 것을 확인할 수 있었다. 특히, 위 실험이 해법에서 사용되는 두 모수  $\lambda$ 와  $T_0$ 를 최적화하지 않은 상태에서 얻은 결과이므로 앞으로 추가연구를 통하여 두 모수의 최적화가 이루어질 수 있다면 상당한 개선이 가능할 것으로 기대된다.

## 6. 결론

본 연구에서는 공통납기를 갖는 작업스케줄링 문제를 비선형 0-1정수계획으로 정형화한 후, Hopfield 신경망모형으로의 변환과정을 검토하였다. 다음, Hopfield 모형보다 실제적인 평균장이론을 이용하여 정형화된 작업스케줄링 문제를 해결하기 위한 FTP와 MFA 해법을 적용하여 그 성능을 평가하고 얻어진 해의 특성을 관찰하였다. 그 결과, 이 해법들

표 4. 문제의 크기에 따른 MFA해법의 적용 결과

|              | n         |             |              |                |
|--------------|-----------|-------------|--------------|----------------|
|              | 10        | 15          | 20           | 25             |
| 평균 목적함수 값*   | 35.3/34.7 | 125.8/123.6 | 307.1/299.9  | 567.2/546.9    |
| 평균 오차율(%)    | 1.73      | 2.07        | 2.57         | 3.62           |
| 평균CPU 시간(초)* | 2.81/0.16 | 13.46/3.25  | 40.03/140.70 | 86.77/5,346.58 |

\* 표내의 수치 a/b에서 a,b는 각각 MFA와 최적해법의 수치를 의미함.

의 성능이 상당히 우수한 것으로 나타나, 본 스케줄링 문제에 효과적으로 적용될 수 있음을 보여 주었다.

#### 참 고 문 헌

- [1] Arizono, I., Yamamoto, A., and Ohta, H. "Scheduling for Minimizing Total Actual Flow Time by Neural Networks", *International Journal of Production Research*, 30 (3), 503-511, 1992.
- [2] Bagchi, U., Chang, Y., and Sullivan, R. "Minimizing Absolute and Squared Deviations of Completion Times with Different Earliness and Tardiness Penalties and a Common Due Date", *Naval Research Logistics Quarterly*, 34, 739-751, 1987.
- [3] Bagchi, U., Sullivan, R., and Chang, Y. "Minimizing Mean Absolute Deviation of Completion Times About a Common Due Date", *Naval Research Logistics Quarterly*, 33, 227-240, 1986.
- [4] De, P., Ghosh, J., and Wells, c., "A Note on the Minimization of Mean Squared Deviation of Completion Times About a Common Due Date", *Management Science*, 35, 1143-1147.
- [5] Eilon, S. and Chowdhury, I. G., "Minimizing Waiting Time Variance in the Single Machine Problem", *Management Science*, 23 (6), 567-575, 1977.
- [6] Foo, Y. P. S. and Takefuji, Y. "Stochastic Neural Network for Solving Job-Shop Scheduling: Part 1 and 2", *Proceedings of IEEE International Conference on Neural Networks*, San Diego, II.275-II.290, 1988.
- [7] Hertz, J., Krogh, A., and Palmer, R. G., *Introduction to the Theory of Neural Computation*, Santa Fe Institute, Addison-Wesley Publishing Company, 1991
- [8] Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Natl. Acad. Sci. USA*, 79, 2554-2558, 1982.
- [9] Hopfield, J. J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons", *Proc. Natl. Acad. Sci. USA*, 81, 3088-3092, 1984.
- [10] Hopfield, J. J. and Tank, D. W. "Neural Computation of Decisions in Optimiza-

- tion Problems”, *Biological Cybernetics*, 52, 141-152, 1985.
- [11] Vaithyanathan, S. and Ignizio, J. P. A “Stochastic Neural Network for Resource Constrained Scheduling”, *Computers and Operations Research*, 19 (3 & 4), 241-254, 1992.
- [12] Van den Bout, D. E. and Miller, T. K. “Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing”, *Biological Cybernetics*, 62, 129-139, 1989.
- [13] Van Hulle, M. M., “A Goal Programming Network for Mixed Integer Linear Programming: A Case Study for the Job-Shop Scheduling Problem”, *International Journal of Neural Systems*, 2 (3), 201-209, 1991.
- [14] Vani, V. and Raghavachari M., “Deterministic and Random Single Machine Scheduling With Variance Minimization”, *Operations Research*, 35 (1), 111-120, 1987.
- [15] Wang, G. and Ansari, N., “A Neural Network Approach to Broadcast Scheduling in Multi-hop Radio Networks”, *Proceedings of IEEE International Conference on Neural Networks*, Orlando, 4699-4703, 1994.
- [16] Yih, Y., Liang, T.-P., and Moskowitz, H., “Robot Scheduling in a circuit Board Production Line: A Hybrid OR/ANN Approach”, *IIE Transactions*, 25 (2), 26-33, 1993.
- [17] Yu, C. S. and Lee, W. D., “Parallel Mean Field Annealing Neural Network for Solving Traveling Salesman Problem”, *Proceedings of Intl. Joint Conference on Neural Networks*, Baltimore, 532-536 (IV), 1992
- [18] Zhang, Z. Z., Ansari, N., Hou, E., and Yi, P.-K., “Multiprocessor Scheduling by Mean Field Theory”, *Proceedings of Intl. Joint Conference on Neural Networks*, Baltimore, 582-587(IV), 1992.
- [19] Zheng, W.-X., Nagasawa, H., and Nishiyama, N., “Single Machine Scheduling for Minimizing Total Cost with Identical, Asymmetrical Earliness and Tardiness Penalties”, *International Journal of Production Research*, 31 (7), 1611-1620, 1993.