

적응형 의사결정 트리와 최단 경로법을 이용한 기계 진단 및 보전 정책 수립

백 준 겐*

Machine Diagnosis and Maintenance Policy Generation
Using Adaptive Decision Tree and Shortest Path Problem

Jun-Geol Baek*

■ Abstract ■

CBM (Condition-Based Maintenance) has increasingly drawn attention in industry because of its many benefits. CBM problem is characterized as a state-dependent scheduling model that demands simultaneous maintenance actions, each for an attribute that influences on machine condition. This problem is very hard to solve within conventional Markov decision process framework. In this paper, we present an intelligent machine maintenance scheduler, for which a new incremental decision tree learning method as evolutionary system identification model and shortest path problem as schedule generation model are developed. Although our approach does not guarantee an optimal scheduling policy in mathematical viewpoint, we verified through simulation based experiment that the intelligent scheduler is capable of providing good scheduling policy that can be used in practice.

Keyword : CBM, ADT (Adaptive Decision Tree), shortest path problem, maintenance schedule

1. 서 론

제조 시스템에서 예기치 않은 기계 고장은 시스

템의 생산 능력과 제품의 품질을 저하시키는 원인이 되므로 기계 상태를 정확히 진단(diagnosis)하고 이에 따른 효율적인 보전(maintenance)을 수

논문접수일 : 2001년 8월 7일

논문게재확정일 : 2002년 3월 8일

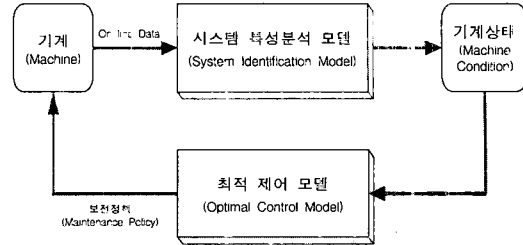
* 인덕대학 산업시스템경영과

행함으로써 기계의 신뢰성(reliability)을 향상시키는 일은 매우 중요하다.[5] 기계 보전에 있어서 과거에는 미리 정해진 주기를 바탕으로 예방 보전(preventive maintenance)을 수행하였던 것에 반해 오늘날의 생산 현장에서는 실시간으로 수집되는 데이터를 기반으로 기계 상태를 진단하고 기계의 이상 징후가 나타난 경우에 보전을 수행하는 상태기반 보전(condition-based maintenance)으로의 변화가 진행되고 있다. 상태기반 보전은 기계의 이상 징후가 나타난 경우 보전을 수행하기 때문에 불필요한 보전 행위를 줄일 수 있고 이로 인해 적은 비용으로 효율적인 보전을 수행할 수 있다는 장점을 지닌다.[13]

상태기반 보전 모델은 기계 상태 데이터가 실시간으로 수집되어 데이터베이스에 저장되는 환경에서 성공적으로 구축될 수 있으며, 동적(dynamic)으로 변화하는 기계 상태를 기반으로 보전 계획을 수립한다는 측면에서 순차적 의사결정 문제(sequential decision making problem)로 표현될 수 있다.

일반적으로 순차적 의사결정 문제를 풀기 위해서는 실시간으로 수집되는 데이터를 기반으로 시스템의 특성을 분석하는 방법에 대한 연구와 분석된 시스템의 특성을 바탕으로 효율적인 제어 정책(control policy)을 수립하는 방법에 대한 연구가 필요하다.[4] 따라서 상태기반 보전 모델은 [그림 1]과 같은 구조를 지닌 순차적 의사결정 문제로 정의될 수 있으며, 문제의 풀이를 위해서는 시스템 특성 분석 모델(system identification model)을 통해 기계 상태를 추론하고, 추론된 기계 상태를 기반으로 최적 제어 모델(optimal control model)을 통해 효율적인 보전 정책을 수립할 수 있는 방법에 대한 연구가 수행되어야 한다.

기계 상태의 추론을 위해 사용되는 데이터는 기계의 특성을 나타내는 속성(attribute) 집합(A)과 속성 값에 따른 기계 상태(c)로 표현되며, 속성 집합이 n 개의 속성으로 이루어져 있다고 가정하면 $A = \{a_1, a_2, \dots, a_n\}$ 로 표현될 수 있다. 따라



[그림 1] 상태기반 보전 모델

서 기계로부터 수집되는 기계 상태 데이터는 다음과 같은 형태로 정의될 수 있다.

$$E(A, c) = \{a_1, a_2, \dots, a_n, c\}$$

각각의 속성($a_i, i = 1, 2, \dots, n$)은 표현하는 값의 특성에 따라 연속형 속성(continuous attribute)과 이산형 속성(discrete attribute)으로 분류될 수 있다. 예를 들면 연속형 속성은 수치형(numerical) 값을 갖는 온도나 속도 등을 정의한 속성을 의미하고, 이산형 속성은 공정 번호 등과 같이 미리 정의된 이산형(discrete) 값 중에서 하나를 갖는 속성을 의미한다.

기계의 속성이 이산형 값을 갖는 경우 속성 값의 범위(range)는 식 (1)과 같이 정의될 수 있다. 식 (1)에서 r_i 는 이산형 속성 a_i 가 가질 수 있는 값의 개수를 의미한다.

$$\text{range}(a_i) = \{a_i^1, a_i^2, \dots, a_i^{r_i}\} \quad (1)$$

연속형 값을 갖는 속성에 대한 속성 값의 범위는 다음과 같이 정의될 수 있다.

$$\text{range}(a_i) \in R$$

또한 속성 값에 따른 기계 상태를 나타내는 c 는 이산형 집합 C 에서 하나의 원소 값을 가질 수 있으며, C 가 k 개의 원소로 이루어진 집합이라고 가정하면 다음과 같이 정의될 수 있다.

$$c \in C = \{C_1, C_2, \dots, C_k\}$$

따라서 기계 상태 데이터는 n 개의 속성 값과 그에 따른 기계 상태 값을 나타내는 $(n+1)$ 튜플 (tuple)로 표현될 수 있으며, 기계 상태 데이터 집합 E 에 속한 데이터 $e_t (t=1, 2, \dots, |E|)$ 는 $\{a_{1t}, \dots, a_{it}, \dots, a_{nt}, c_t\}$ 로 표현된다. 각각의 데이터(e_t)에서 속성 값에 따른 기계 상태 c_t 는 반드시 집합 C 의 원소 중 하나에 할당되어야 하는데 이러한 기계 상태 c_t 를 결과 클래스(class)라고 정의하고 다음과 같이 표현한다.

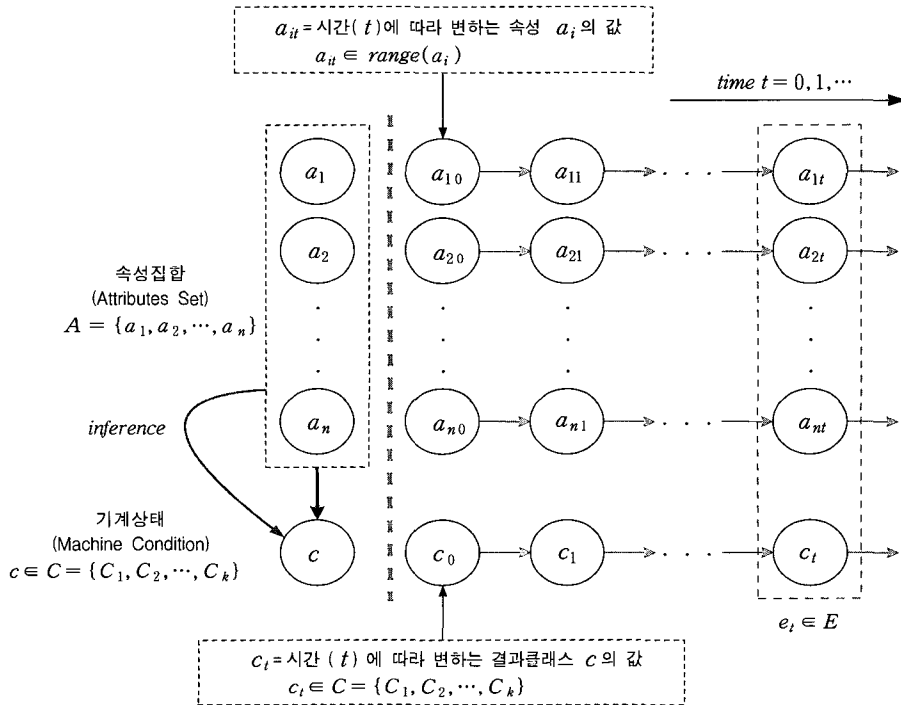
$$c_t = \text{Class}(e_t) \in C \quad \text{for each } e_t \in E$$

실시간으로 수집되는 기계 상태 데이터는 [그림 2]와 같이 기계의 특성을 나타내는 속성($a_i, i=1, 2, \dots, n$) 값이 시간의 흐름에 따라 변화하고, 속성 값의 변화에 따라 기계 상태(c)도 변화하는 이중 확률 프로세스 모델(doubly stochastic process model)로 표현될 수 있으며 다음과 같은 특징을

지닌다.

- 기계 상태 데이터에서 기계의 특성을 나타내는 속성은 이산형 속성과 연속형 속성이 혼재되어 있다.
- 기계 상태를 나타내는 결과 클래스의 집합은 2개 이상의 원소를 가질 수 있다.
- 기계의 속성 값과 결과 클래스는 시간의 흐름에 따라 변화한다.
- 기계의 속성 값과 결과 클래스는 상관 관계를 갖는다.
- 기계 상태 데이터는 실시간으로 수집되고 시간이 흐름에 따라 누적된다.

순차적 의사결정 문제로 표현되는 상태기반 보전 문제를 풀기 위해 일반적으로 사용하는 방법은 마코프 의사결정 프로세스(Markov decision process) 모델을 이용하는 방법이다[2, 3, 6]. 마코프 의사결정 프로세스 모델을 이용한 방법은 전이 확률

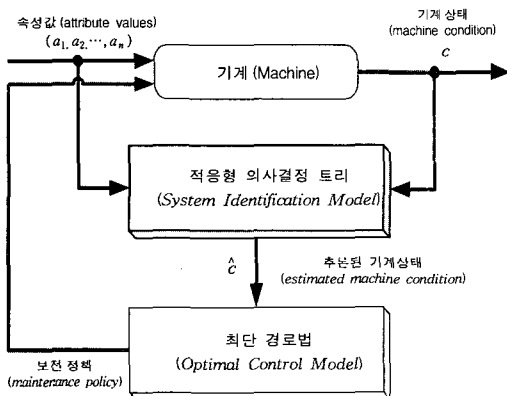


[그림 2] 기계 상태 데이터 모델

행렬(transition probability matrix)을 통해 시간의 흐름에 따라 변화하는 기계 상태를 추론하고 이를 기반으로 최적의 보전 정책을 수립하는 방법을 제시하고 있다. 그러나 마코프 의사결정 프로세스 모델은 기계 상태의 변화를 나타내는 전이 확률 행렬이 미리 주어져 있다는 가정과 보전을 수행하였을 때 변화하는 기계 상태 값이 확률적으로 정의될 수 있다는 가정을 지니고 있기 때문에 완전한 의미의 상태기반 보전 모델이 수립될 수 없다는 단점을 지닌다. 최근 들어 기계 학습(machine learning) 기법을 이용하여 기계 상태 데이터를 기반으로 전이 확률 행렬을 추론하는 연구가 진행되고 있으나 기계 상태 데이터가 위에서 기술한 바와 같이 매우 복잡한 구조를 지니고 있기 때문에 상태기반 보전 문제에 적용하기에는 어려움이 따른다[13, 14].

따라서 본 연구에서는 순차적 의사결정 문제로 모델링되는 상태기반 보전 문제의 해법을 찾기 위해 적응형 의사결정 트리(adaptive decision tree)와 최단 경로법(shortest path problem)을 이용한 모델을 제시하고자 한다.

[그림 3]은 적응형 의사결정 트리와 최단 경로법을 이용하여 구축한 상태기반 보전 정책 수립 모델을 나타낸 그림으로써, 적응형 의사결정 트리가 기계 상태를 추론하는 시스템 특성 분석 모델(system identification model)로 사용되었음을 나타내며 추



[그림 3] 상태기반 보전 정책 수립 모델

론된 기계 상태를 기반으로 효율적인 보전 정책을 수립하기 위한 최적 제어 모델(optimal control model)로써 최단 경로법이 사용되었음을 나타내고 있다.

2. 문제 정의

기계 상태 데이터는 [그림 2]에서 나타낸 바와 같이 기계의 특성을 나타내는 속성 집합 $A = \{a_1, a_2, \dots, a_n\}$ 과 속성 값에 따른 기계 상태(c)로 표현되며, t 시점에서 수집되는 기계 상태 데이터(e_t)의 i 번째 속성 값은 a_{it} 로 표현되고 그 때의 기계 상태는 c_t 로 표현되어질 수 있다. 또한 속성 a_i 가 연속형 값을 갖는 속성이라면 속성 값의 범위는 $l(a_i) \leq a_i \leq u(a_i)$ 로 표현되어질 수 있다. 단, $l(a_i)$ 는 속성 a_i 가 가질 수 있는 최소값을 의미하고 $u(a_i)$ 는 속성 a_i 가 가질 수 있는 최대값을 의미한다.

보전 작업(maintenance action)은 각각의 속성과 연관되어 행해지게 되며 속성 a_i 와 관련된 보전 작업은 $x_i \equiv x(a_i)$ 로 표현될 수 있다. 따라서 보전 정책(maintenance policy)을 수립한다는 것은 식 (2)와 같이 각각의 속성 값에 따른 보전 작업을 정의하는 것과 같다.

$$\pi_i = \text{domain}(a_i) \rightarrow \text{domain}(x_i) \quad (2)$$

보전 작업에는 기계 고장의 원인이 되는 구성 요소를 복구(repair)하거나 교체(replacement)하는 작업뿐만 아니라 기계 상태를 정상으로 되돌리기 위해서 연속형 속성의 값을 조정하는 행위까지 포함된다. 따라서 연속형 속성 a_i 에 대한 보전 작업은 $x_i \in \text{domain}(a_i)$ 로 정의될 수 있다. 보전 정책의 수립은 식 (2)에서 정의한 속성별 보전 작업을 모든 속성에 대해 정의하는 것으로써 보전 작업 벡터(vector) $X = \{x_1, x_2, \dots, x_n\}$ 를 정의하기 위해 식 (3)과 같은 정책을 수립하는 것과 같다.

$$\pi = \text{domain}(A) \rightarrow \text{domain}(X) \quad (3)$$

t 시점에서의 기계 상태 데이터(e_t)를 기반으로 보전 작업을 수행하기 위해서는 각각의 속성 값(a_{it})에 따른 보전 작업(x_{it})으로 인해 비용이 발생하게 되는데 이러한 보전 비용(maintenance cost)은 식 (4)와 같이 정의될 수 있다. 식 (4)에서 $\sigma(a_{it}, x_{it})$ 는 속성 a_i 의 값이 a_{it} 일 때 보전 작업 x_{it} 를 수행함으로써 발생하는 비용을 나타내는 함수(function)를 의미한다.

$$mc_{it} = \sigma(a_{it}, x_{it}) \quad (4)$$

따라서 t 시점에서의 보전 작업으로 인해 발생하는 총 비용(total cost)은 식 (5)와 같이 정의될 수 있다.

$$MC_t = (mc_{1t}, mc_{2t}, \dots, mc_{nt}) \quad (5)$$

본 연구의 대상이 되는 기계 상태 데이터 e_t 는 시간($t = 0, 1, 2, \dots$)의 흐름에 따라 변화하기 때문에 변화하는 기계 상태에 따라 행해지는 보전 정책(π)으로 인해 발생하는 누적 보전 비용(cumulative maintenance cost)은 식 (6)과 같이 정의될 수 있다.

$$V^\pi(e_0) = \sum_{t=0}^L MC_t \quad (6)$$

식 (6)에서 e_0 는 초기($t = 0$)의 기계 상태를 의미하고 L 은 대상 기계의 생명 주기(lifetime)를 의미한다. 따라서 최적의 보전 정책을 수립한다는 것은 누적 보전 비용을 최소화시킬 수 있는 보전 정책(π^*)을 찾아내는 문제로 정의할 수 있으며 이는 식 (7)과 같이 정의할 수 있다.

$$\pi^* = \operatorname{argmin}_\pi V^\pi(e_0) \quad (7)$$

그러나 [그림 2]에서 제시한 바와 같은 실시간으로 수집되는 기계 상태 데이터를 기반으로 식 (7)

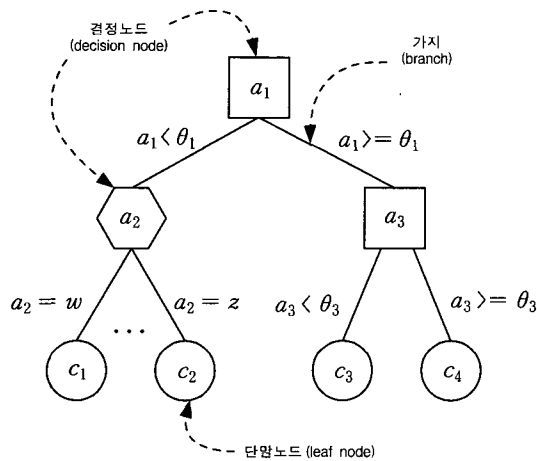
을 만족시키는 최적의 보전 정책을 수학적으로 찾아낸다는 것은 사실상 불가능하기 때문에 본 연구에서는 적응형 의사결정 트리를 이용하여 실시간으로 수집되는 데이터를 기반으로 기계 상태를 추론하고, 추론된 기계 상태가 기계 고장과 같은 이상 징후를 나타내는 경우 효율적인 보전 정책을 수립하기 위한 방법으로 최단 경로법을 이용한 상태 기반 보전 정책 수립 모델을 제시하고자 한다.

3. 적응형 의사결정 트리

본 연구에서는 실시간으로 수집되는 기계 데이터로부터 기계 상태를 추론하기 위한 시스템 특성 분석 모델로써 백준걸[1]이 제시한 적응형 의사결정 트리인 ADT(Adaptive Decision Tree)를 사용한다. 기계 데이터의 분석을 통해 기계 상태 정보를 추론하는 것은 기계 고장 진단 시스템 구축을 위해 매우 중요한 연구 과제이다. 이런 문제는 기존에 신경망(neural network), 퍼지 이론(fuzzy logic), 사례 기반 추론(case-based reasoning) 등 여러 가지 방법을 이용하여 연구되어 왔으나, 의사결정 트리는 이러한 방법들과는 달리 기계 고장의 원인을 규칙(rule)으로 표현할 수 있기 때문에 사용자가 기계 고장 원인을 쉽게 이해할 수 있다는 장점을 지닌다.

의사결정 트리는 데이터를 구성하는 속성과 클래스와의 연관 관계를 규명하기 위해 데이터 집합(data set)을 부분 집합(subset)으로 분할하고, 분할된 부분 집합의 특성을 규명하는데 사용되는 방법론이다. 의사결정 트리 문제는 속성 값과 결과 클래스 값의 직교 곱(cartesian product)으로 표현되는 데이터 공간에서 데이터 집합 E 가 주어졌을 때 이 데이터 집합을 각 속성 값에 따라 부분 집합으로 분할하는 것이며, 이 때 분할 기준은 각 부분 집합에 포함된 모든 데이터가 최대한 같은 결과 클래스 값을 갖도록 하는 것이다. 의사결정 트리는 [그림 4]에서 보듯이 그래프의 일종으로 단말 노드(leaf node)를 제외한 중간 노드(이하 결정 노드

(decision node)라 함.)는 분할 기준이 되는 속성을 의미하고, 단말 노드는 상위 결정 노드 패스(path)에 의해 분류된 결과 클래스들의 집합을 의미한다. 가지(branch)는 결정 노드의 분할 기준이 되는 속성의 분할점(splitting point)이 된다.



범례 (Legend)

- : 연속형 속성(Continuous-valued attribute)
- ⬡ : 이산형 속성(discrete-valued attribute)
- : 결과클래스(Class)
- θ : 분할점(splitting point)

[그림 4] 의사결정 트리

의사결정 트리의 구축 방법은 데이터가 증가할 때 기존에 구축된 트리를 어떻게 처리하는 지에 따라 비증분형(non-incremental) 방법과 증분형(incremental) 방법으로 분류할 수 있다. 비증분형 의사결정 트리 구축 방법은 고정된 데이터 집합으로부터 하나의 트리를 만드는 것으로 구현이 쉬운 반면 새로운 데이터가 추가될 경우 이 변화를 반영하기 위해 처음부터 다시 트리를 구축해야 하는 문제점을 지닌다. 비증분형 의사결정 트리의 구축에 관련된 대표적인 연구로는 C4.5[8]를 들 수 있다. C4.5는 ID3[7] 알고리즘을 확장한 것으로 결정 노드에서 분할 기준이 되는 최적 검사 속성을 선택하기 위해서 속성 선택 척도(metric)로 정보 획득량(information gain)을 사용한다. 이 척도는 정보 이

론(information theory)의 엔트로피(entropy) 개념을 사용한 것으로써 엔트로피 값은 데이터의 분할이 잘 될수록 작은 값을 갖게 된다. 따라서 검사 속성 후보 중에서 엔트로피 값이 최소가 되는 속성을 해당 결정 노드의 검사 속성으로 선택하게 된다. 그러나 C4.5는 고정된 데이터에 대해 정보 이론에 근거한 효율적인 의사결정 트리를 구축할 수 있으나 비증분형 의사결정 트리 구축 방법론이 갖는 문제점을 그대로 지니고 있기 때문에 본 연구에서와 같이 실시간으로 수집되는 기계 상태 데이터가 주어지는 경우 변화하는 기계 특성을 정확히 반영한 기계 고장을 예측하기 위해서는 데이터가 증가할 때마다 새롭게 의사결정 트리를 구축해야 하는 문제점을 지닌다.

따라서 위에서 언급한 비증분형 의사결정 트리의 문제점을 해결하기 위한 방법으로 데이터가 증가하더라도 기존에 구축된 의사결정 트리를 버리지 않고 데이터가 증가될 때마다 기존의 트리를 수정해 나가는 증분형 의사결정 트리의 구축에 관한 연구가 진행되고 있다. 증분형 의사결정 트리에 관련된 대표적인 연구로는 ID5R[11, 12]을 들 수 있다. ID5R은 ID4[9]와 ID5[10]와 같은 증분 의사결정 트리 구축 알고리즘을 기반으로 개발된 방법론으로서 새로운 데이터가 추가될 때마다 엔트로피를 나타내는 E-score를 척도로 사용하여 기존의 트리를 재구성하는 방법을 제시하고 있다. 또한 ID5R은 풀-업(pull-up) 알고리즘을 이용하여 데이터가 추가될 때마다 속성별 E-score를 갱신하고 이를 기준으로 기존에 구축된 의사결정 트리를 수정함으로써 추가로 입력되는 데이터의 특성을 반영한 의사결정 트리를 구축할 수 있도록 하였다. ID5R은 추가되는 데이터에 따라 E-score를 갱신하고 트리를 재구성하므로 비증분형 트리 구축 방법보다는 효율적인 트리를 구축할 수 있지만 수집되는 데이터에 아무런 제약을 두지 않고 그대로 받아들이기 때문에 너무나 잦은 트리의 재구성으로 인해 계산 부하가 증가할 수 있다는 단점을 지닌다.

본 연구의 대상이 되는 기계 데이터는 [그림 2]에서와 같이 시간의 흐름에 따라 기계의 속성 값과 그에 따른 기계 상태 값이 변화하고 변화된 데이터가 누적되어 저장되는 특징을 지니므로 이러한 기계 상태 데이터베이스를 기반으로 학습을 수행하기 위해서는 의사결정 트리가 실시간으로 수집되는 데이터에 따라 재구성되는 충분한 의사결정 트리 구조를 지녀야 한다. 그러나 기존의 충분한 의사결정 트리는 새로운 데이터의 입력에 따라 기존의 트리가 너무나 빈번하게 변경될 수 있다는 단점을 지니기 때문에 본 연구에서는 백준걸[1]이 제시한 적응형 의사결정 트리를 시스템 특성 분석 모델로 사용하고자 한다. 백준걸[1]이 제시한 적응형 의사결정 트리는 새로운 데이터가 추가로 입력됨에 따라 기존에 구축된 의사결정 트리를 확장 및 변경함으로써 시간에 따라 변화하는 기계의 특성을 반영한 의사결정 트리를 구축할 수 있도록 하였으며, 너무나 빈번한 트리의 재구성을 방지하기 위해 의사결정 트리의 적합도 검사와 노이즈 데이터 검사를 통해 불필요한 트리의 갱신을 방지하고 효율적인 트리의 확장 및 변경이 가능하도록 하였다. 백준걸[1]이 제시한 적응형 의사결정 트리 구축 절차는 다음과 같이 요약할 수 있으며, 각 단계별 자세한 내용은 백준걸[1]의 논문을 참조하길 바란다.

단계 1. 초기 의사결정 트리 구축

검사 속성(test attribute) 선택 척도인 SDM (Sum of Dominance Metric)을 이용하여 이미 수집된 데이터 집합에 대한 초기 의사결정 트리를 구축한다.

단계 2. 군(Batch) 데이터 구성

너무 잦은 트리의 갱신을 방지하고 오류 데이터의 검사가 용이하도록 실시간으로 수집되는 데이터를 일정 크기의 군으로 형성한다.

단계 3. 의사결정 트리 갱신 여부 판단

- 단계 3.1 데이터 군의 적합도 검사
입력된 데이터 군이 현재의 트리에 얼마나 잘 맞

는지를 검사한다. 일정 허용 오차 내에서 데이터 군이 적합하다는 판단이 서면 <단계 2>로 가고(현재의 트리를 보존), 그렇지 않으면 <단계 3.2>로 간다.

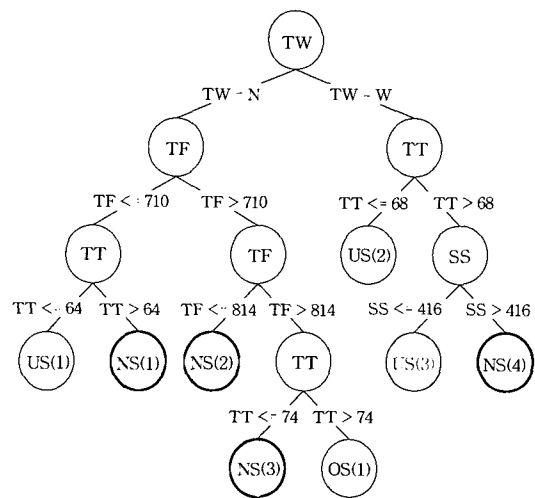
• 단계 3.2 노이즈(noise) 데이터 검사

데이터 군에 노이즈 데이터가 섞여있는지를 검사한다. 노이즈 데이터가 존재하면 이를 제거하고 <단계 4>로 간다.

단계 4. 의사결정 트리의 갱신(변경 및 확장)

SDM 척도를 이용하여 트리를 변경하거나 확장한다.

백준걸[1]이 제시한 적응형 의사결정 트리 구축 방법을 이용하여 실시간으로 수집되는 기계 상태 데이터를 기반으로 기계 속성과 기계 상태와의 상관 관계를 규명하기 위한 사례로써 본 연구에서는 태핑 작업(tapping process)을 수행하는 기계 T에 대한 데이터를 기반으로 의사결정 트리를 구축하였다. 기계 T는 공정 변수에 따라 태핑 작업의 결과가 변화하는 특징을 지니게 되는데 태핑 작업의 결과는 태핑된 구멍(hole)의 크기가 너무 크거나 작은 경우 문제가 발생하기 때문에 적절한 구멍의 크기를 유지할 수 있도록 하는 일이 중요하다. 본 연구에서는 태핑 작업에 영향을 미치는 공정 변수



[그림 5] 적응형 의사결정 트리

<표 1> 기계 이력 데이터 형식

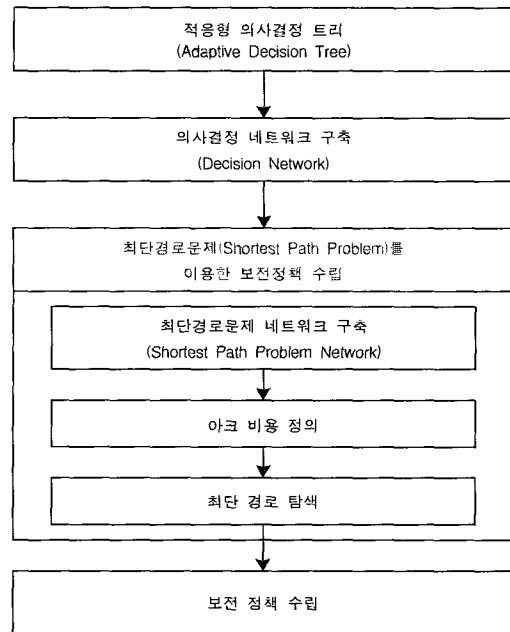
	Name	Description	Type	Domain
attributes	SS	Spindle Speed	continuous	200~600 (RPM)
	TT	Tapping Torque	continuous	50~80 (NM)
	TF	Thrust Force	continuous	600~900 (W)
	LF	Lateral Force	continuous	400~800 (W)
	TW	status of Tap Wearing	discrete	N(normal), W(worn)
class	HS	Hole Size	discrete	US, NS, OS

를 <표 1>과 같은 속성으로 정의하고, 태핑된 구멍의 크기에 따라 결과 클래스(US(Under Size), NS(Normal Size), OS(Over Size))를 정의한 데이터를 기반으로 태핑 작업의 이상 상황을 진단할 수 있는 적응형 의사결정 트리를 구축하였으며 구축된 의사결정 트리는 [그림 5]와 같은 형태를 지닌다. [그림 5]에서 단말 노드(leaf node)의 괄호 안의 숫자는 노드의 구별을 위해 편의상 번호를 부여한 것으로써 예를 들어 US(1)은 결과 클래스가 US인 첫 번째 단말 노드를 의미한다.

4. 최단 경로법을 이용한 보전 정책 수립

본 연구에서 시스템 특성 분석 모델로 사용한 적응형 의사결정 트리는 실시간으로 수집되는 기계 상태 데이터를 기반으로 기계 속성과 기계 상태와의 상관 관계를 규명한 것으로써 의사결정 트리를 통해 기계의 속성 값에 따른 기계 상태를 추론할 수 있도록 하였다. 본 절에서는 적응형 의사결정 트리를 통해 추론된 기계 상태를 기반으로 기계의 이상 상황을 진단하고, 이상 상황이 진단되었을 경우 이상 상황이 발생한 원인 분석 및 효율적인 보전 정책을 수립하기 위한 방법으로 [그림 6]과 같은 보전 정책 수립 모델을 제시한다.

[그림 5]는 <표 1>에서 기술한 기계 상태 데이터를 기반으로 구축한 적응형 의사결정 트리를 나타내는 그림으로써 결과 클래스에 영향을 미치는 중요 속성이 {TW, TF, TT, SS}임을 알 수 있



[그림 6] 보전 정책 수립 모델

고 기계의 속성 값에 따라 4가지 유형의 이상 상황이 발생할 수 있음을 나타내고 있다. 예를 들어 속성 값이 {360, 80, 720, 516, W}인 경우 결과 클래스는 이상 상황인 US(3)가 되며, 이 때 이상 상황 발생의 원인이 되는 속성은 {TW, TT, SS}임을 알 수 있다. 따라서 본 연구에서는 [그림 5]와 같은 의사결정 트리를 이용하여 기계의 속성 값에 따른 기계 상태를 추론하고, 기계의 이상 상황이 진단되었을 경우 [그림 6]과 같은 절차를 통해 효율적인 보전 정책을 수립하는 방법을 제시하고자 한다.

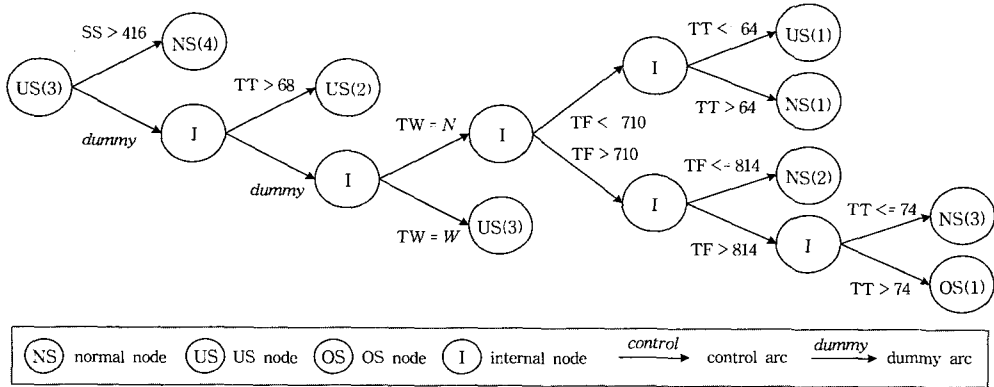
4.1 의사결정 네트워크 구축

적응형 의사결정 트리를 통해 진단된 기계 이상 상황에 대한 효율적인 보전 정책을 수립하기 위해 본 연구에서는 구축된 의사결정 트리를 바탕으로 의사결정 네트워크(decision network)를 구축한 후 이를 바탕으로 최단 경로법을 이용하여 효율적인 보전 정책을 수립하는 방법을 제시하고자 한다. 의사결정 네트워크는 구축된 의사결정 트리를 통해 기계의 이상 상황이 진단되었을 경우 이상상황(고장 노드)으로부터 전환 가능한 정상상황(정상 노드)으로의 경로를 나타낸 네트워크를 의미한다. 예를 들어 [그림 7]은 [그림 5]와 같은 의사결정 트

리를 통해 기계 상태가 $US(3)$ 라는 이상상황으로 진단되었을 경우 기계 상태를 정상상황($NS(1)$, $NS(2)$, $NS(3)$, $NS(4)$)로 전환하기 위해 필요한 속성 값의 조정을 나타낸 의사결정 네트워크 표현한 그림이다. [그림 7]에서 시작 노드는 기계 상태가 이상 상황인 $US(3)$ 로 진단되었음을 나타내며, 이상 상황 노드를 정상 노드로 전환하기 위해 필요한 속성 값의 조정을 아크(arc)를 통해 표현한 것이다. 예를 들어 $US(3)$ 노드를 정상 노드인 $NS(4)$ 로 전환하기 위해서는 속성 SS 의 값을 " $SS > 416$ "으로 조정해야 함을 나타낸다. 네트워크의 중간에 위치한 내부 노드(internal node)는 의

〈표 2〉 의사결정 네트워크 구축 알고리즘

의사결정 네트워크 구축 알고리즘 : Construct_Network(FN, T)
<p>FN : 진단된 이상 상황 노드 (diagnosed fault node) T : 의사결정 트리 (decision tree) $Decision_Net$: 의사결정 네트워크 (decision network) $Node$: 의사결정 네트워크를 구성하는 노드의 리스트 (pointer to a list of nodes) Arc : 의사결정 네트워크를 구성하는 아크의 리스트 (pointer to a list of arcs)</p> <p>Begin Algorithm /* 진단된 이상 상황 노드가 의사결정 네트워크의 시작 노드로 사용됨 */ $Current_Node_Up = Current_Node_Dn = FN$ /* 의사결정 트리에서 진단된 이상 상황 노드의 깊이 */ $Current_D = DepthOfNode(FN)$ /* 의사결정 트리에서 부모(Parent) 노드의 탐색을 통해 의사결정 네트워크에서 사용될 새로운 노드와 아크를 추가. 이 작업을 의사결정 트리의 루트 노드까지 반복 */ WHILE($Current_D > 0$) { $Current_Node_Up = ParentNode(Current_Node_Up)$ $Node \leftarrow CreateNode(Current_Node_Up)$ $Arc \leftarrow CreateArc(Current_Node_Up)$ $Current_D \leftarrow DepthOfNode(Current_Node_Up)$ /* 루트 노드로부터 가능한 정상 상황 노드로의 Path를 탐색하여 의사결정 네트워크에서 사용될 노드와 아크를 추가. 이 작업을 단말 노드까지 반복 */ WHILE ($Current_Node_Dn \neq LEAF_NODE$) { $Current_Node_Dn = ChildNode(Current_Node_Dn)$ $Node \leftarrow CreateNode(Current_Node_Dn)$ $Arc \leftarrow CreateArc(Current_Node_Dn)$ } } /* 위에서 추가된 노드와 아크를 이용하여 의사결정 네트워크를 구축 */ ConstructDecNet ($Node, Arc, Decision_Net$) Return $Decision_Net$ End</p>



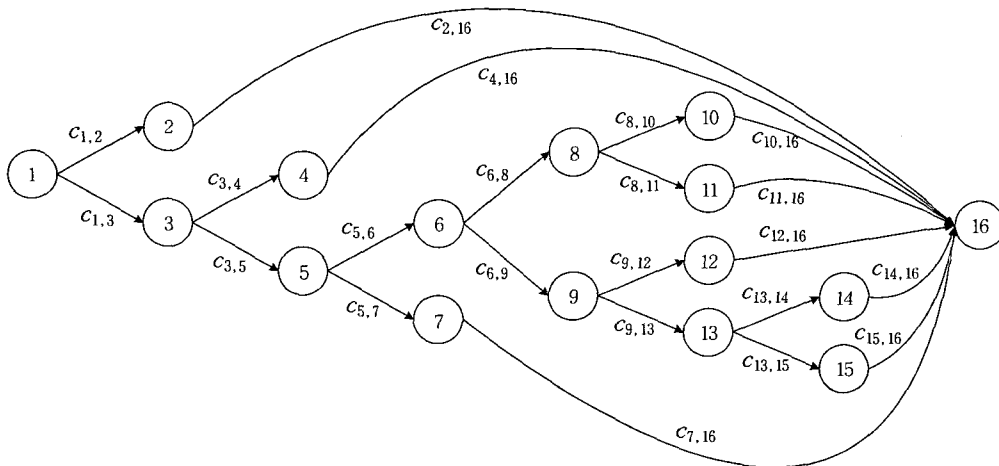
[그림 7] 단말 노드 US(3)에 대한 의사결정 네트워크

사결정 트리로부터 의사결정 네트워크를 구축하기 위해 사용한 가상 노드(dummy node)를 나타내고, *dummy*로 표시된 아크는 내부 노드와의 연결을 위해 임시로 사용된 가상 아크(dummy arc)로써 단순히 노드간의 연결을 의미할 뿐 어떠한 속성 값의 조정도 수행되지 않음을 나타낸다. 의사결정 트리를 기반으로 의사결정 네트워크를 구축하는 절차는 <표 2>과 같다.

4.2 최단 경로 문제 네트워크

본 연구에서는 [그림 7]에서 제시한 의사결정 네

트워크를 통해 여러 가지 가능한 정상 노드로의 전환 중 가장 효율적인 보전 정책을 수립하기 위해 최단 경로법(shortest path problem)을 이용한 방법을 제시하고자 한다. 이를 위해서는 우선적으로 [그림 7]에서 제시한 의사결정 네트워크를 [그림 8]과 같은 최단 경로 문제 네트워크로 변환하는 작업이 필요하다. [그림 8]은 최단 경로 문제로의 변환을 위해 작성된 네트워크로써 표기상의 편의를 위해 노드를 숫자(1, 2, ..., 16)로 표기하였고, 노드 16은 최단 경로 문제로의 변환을 위해 사용된 가상의 목적점(destination)을 나타낸다. 따라서 [그림 8]에서 제시한 네트워크의 시작점(source)인 노드 1에



[그림 8] 최단 경로 문제 네트워크

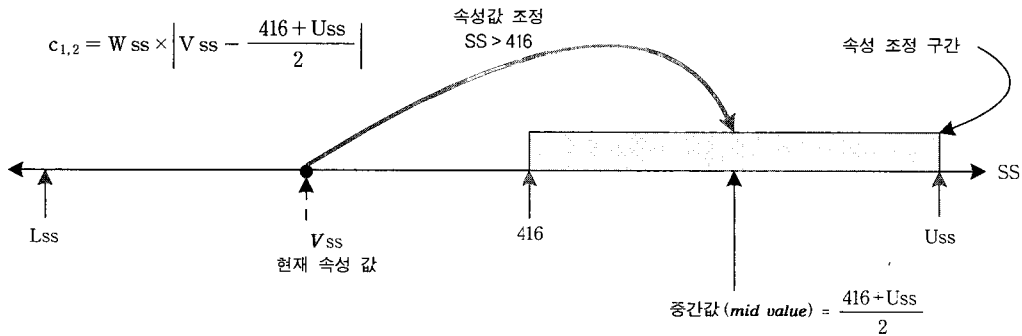
서 목적적인 노드 16까지의 최단 경로를 찾는 것은 진단된 기계 이상 상황에 대한 보전 정책을 수립하는 것과 같은 의미로 해석될 수 있다. 또한 [그림 8]의 아크에 표현된 $c_{i,j}$ 는 노드 i 에서 노드 j 로의 변환을 위해 필요한 속성 값의 조정을 위해 소요되는 비용을 나타낸 것으로써 최단 경로 문제에서 노드 사이의 거리 개념으로 사용될 수 있으며 속성의 특성과 조정되는 값의 크기에 따라 달라질 수 있다. [그림 7]에서 사용된 가상 아크는 네트워크의 구성을 위해 임의로 사용된 아크이므로 최단 경로 문제를 위한 네트워크에서 아크 비용이 0으로 정의되어 진다. 즉, $c_{1,3} = c_{3,5} = 0$ 이 된다. 또한 노드 16은 최단 경로 문제를 정의하기 위해 임의로 사용된 가상 노드이므로 이와 연결된 아크도 가상 아크가 되고 아크 비용은 0으로 정의되어진다. 즉, $c_{2,16} = c_{4,16} = c_{7,16} = c_{10,16} = c_{11,16} = c_{12,16} = c_{14,16} = c_{15,16} = 0$ 이 된다. 최단 경로 문제를 통한 보전 정책 수립의 목적은 최소 비용으로 기계의 이상 상황을 정상 상황으로 변환하기 위한 것이므로 기계 상태를 이상 상황으로 유도하는 아크는 최단 경로로 선택되지 않도록 해야한다. 예를 들면 [그림 8]에서 아크 (3→4), (5→7), (8→10), (13→15)는 기계 상태를 정상이 아닌 $US(2)$, $US(3)$, $US(1)$, $OS(1)$ 으로 유도하는 것이므로 이러한 아크는 최단 경로로 선택되어지지 않도록 하여야 한다. 따라서 본 연구에서는 기

계 상태를 고장으로 유도하는 아크의 비용을 ∞ 로 정의함으로써 해당 아크가 최단 경로로 선택되지 않도록 한다.

정상 노드로의 전환을 위해 속성 값의 조정이 실제로 수행되는 아크의 비용은 속성의 특성과 정책에 따라 다르게 정의되어질 수 있으며 본 연구에서는 속성의 특성을 이산형 속성과 연속형 속성으로 구분하여 아크 비용을 정의한다.

이산형 속성의 경우 속성이 가질 수 있는 값이 유한하기 때문에 정상 노드로의 전환을 위해 필요한 속성 값에 따라 아크 비용이 정의될 수 있다. 예를 들어 아크 (5→6)에 대한 비용은 이산형 속성 TW 의 값을 “ W ”에서 “ N ”으로 조정하기 위한 비용으로써 속성의 현재 값(W)을 다른 값(N)으로 대체하기 위한 비용으로 정의할 수 있다. 따라서 이산형 속성의 아크 비용은 이산형 속성이 가질 수 있는 속성 값에 따른 대체 비용(replacement cost)을 정의함으로써 아크 비용을 정의할 수 있다. 이산형 속성 TW 가 가질 수 있는 속성 값이 $\{N, W\}$ 로 정의되고 속성 값에 따른 대체 비용이 $RCost(N \rightarrow W)$, $RCost(W \rightarrow N)$ 로 정의되는 경우 아크 (5→6)에 대한 비용 $c_{5,6}$ 는 $RCost(W \rightarrow N)$ 로 정의될 수 있다.

연속형 속성의 경우는 아크 비용이 속성 값의 조정 정책에 따라 다르게 정의될 수 있는데 본 연구에서는 중간 값(mid-value) 방식을 이용한 속성 값 조정 정책을 제시한다. 중간 값 방식의 속성 값



[그림 9] 속성 값 조정 정책 (중간 값 방식)

조정 정책은 아크에 정의된 연속형 속성의 속성 값을 조정하기 위해 현재의 속성 값을 속성 조정 구간의 중간 값으로 조정하는 것으로 [그림 9]와 같이 설명할 수 있다.

[그림 9]는 노드 1에서 노드 2로의 변환을 위한 속성 값의 조정을 위해 중간 값 방식을 적용한 상황을 나타낸 그림이다. 아크 (1→2)의 속성 값 조정은 “ $SS > 416$ ”으로써 속성 SS 의 값을 416보다 크게 조정하는 것을 나타내고 있으며, 중간 값 방식을 이용한 속성 값의 조정은 현재의 속성 값 v_{SS} 를 속성 조정 구간의 중간 값인 $\frac{416 + U_{SS}}{2}$ 로 조정하는 것을 나타낸다. 중간 값의 표현식에서 사용된 U_{SS} 는 속성 SS 의 상한 값으로서 속성 값의 조정이 $SS > 416$ 로 정의되는 경우 조정 구간의 정의를 위해 사용되어진다. 만약 속성 값의 조정이 $SS \leq 416$ 로 정의되는 경우에는 속성 SS 의 하한 값인 L_{SS} 가 중간 값의 정의를 위해 사용되며, 이때 중간 값은 $\frac{416 + L_{SS}}{2}$ 로 정의되어진다. [그림 9]는 아크 (1→2)의 속성 조정 비용($c_{1,2}$)이 중간 값 방식에 의해 $w_{SS} \times \left| v_{SS} - \frac{416 + U_{SS}}{2} \right|$ 로 정의되어짐을 나타내고 있다. w_{SS} 는 속성 SS 의 값을 조정하기 위해 소요되는 단위 비용을 나타낸 것으로서 속성의 특성에 따라 달라질 수 있으며 이로 인해 아크의 비용도 달라질 수 있다.

4.3 아크 비용 정의

연속형 속성에 대한 아크 조정 비용은 속성별 단위 조정 비용(w_{SS} , w_{TT} , w_{TF})을 어떻게 정의하느냐에 따라 다르게 정의될 수 있으며, 본 연구에서 다음과 같은 아크 비용 정의 정책을 제시한다.

● 정책 1

[그림 8]에서 제시한 네트워크의 모든 아크 비용을 1로 정의한다. 정책 1은 속성 값의 조정 비용을

고려하지 않고, 단지 이상 상황 노드에서 가장 가까운 위치에 있는 정상 상황 노드를 선택할 수 있도록 하는 정책이다. 따라서 정책 2에 의해 아크 비용이 정의되면 시작점(이상 상황 노드)에서 가장 가까운 위치에 있는 정상 노드로의 경로가 최단 경로로 정의되며 이에 따른 보전 정책이 수립된다.

● 정책 2

아크 비용에서 정의한 속성별 단위 조정 비용을 1로 정의한다. ($w_{SS} = w_{TT} = w_{TF} = 1$) 따라서 정책 2에 의해 정의된 아크 조정 비용은 현재의 속성 값과 조정 구간의 중간 값과의 차이에 의해 정의된다.

● 정책 3

속성별 단위 조정 비용을 속성의 특성에 따라 정의하는 정책으로써 본 연구에서는 속성별 단위 조정 비용을 속성이 가질 수 있는 값의 범위에 따라 정의한다. 이는 속성 값의 범위가 좁은 속성이 속성 값의 범위가 넓은 속성에 비해 보전 대상 속성으로 선택되는 확률이 상대적으로 높아지는 오류를 줄이기 위한 속성 값 정규화(normalization) 방법으로서 속성 값의 범위가 넓은 경우 단위 조정 비용을 낮게 정의하고 속성 값의 범위가 좁은 경우 단위 조정 비용을 높게 정의함으로써 속성 값의 범위에 편향(biased)되지 않는 속성 조정 비용을 정의할 수 있도록 하였다. 정책 3에서 제시하는 정규화된 단위 조정 비용은 연속형 속성에 대해 적용되는 특성으로 식 (8)과 같이 정의할 수 있다. 단, 식 (8)에서 $range(a_i)$ 는 속성 a_i 가 가질 수 있는 속성 값의 범위를 의미한다.

$$w_{a_i} = \left(\frac{range(a_i)}{\sum_{i=1}^n range(a_i)} \right)^{-1}, \quad i = 1, \dots, n \quad (8)$$

본 연구에서는 <표 1>에서 제시한 바와 같이 3개의 연속형 속성 $\{SS, TT, TF\}$ 이 존재하고,

이들이 갖는 속성 값의 범위가 $range(SS) = 400$, $range(TT) = 30$, $range(TF) = 300$ 이므로 정규화된 단위 조정 비용은 다음과 같이 정의될 수 있다.

$$w_{SS} = \left(\frac{range(SS)}{range(SS) + range(TT) + range(TF)} \right)^{-1} = \left(\frac{400}{400 + 30 + 300} \right)^{-1} = 1.8$$

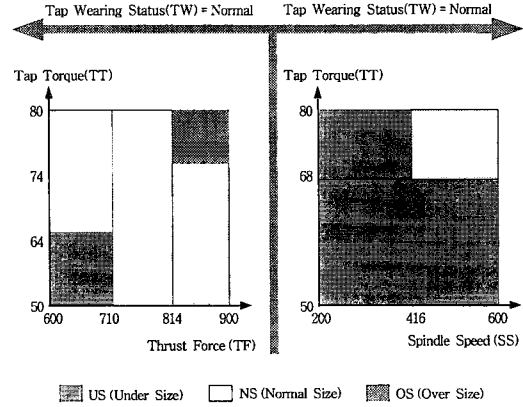
$$w_{TT} = \left(\frac{range(TT)}{range(SS) + range(TT) + range(TF)} \right)^1 = \left(\frac{30}{400 + 30 + 300} \right)^{-1} = 24.3$$

$$w_{TF} = \left(\frac{range(TF)}{range(SS) + range(TT) + range(TF)} \right)^1 = \left(\frac{300}{400 + 30 + 300} \right)^{-1} = 2.4$$

● 정책 4

본 연구의 대상이 되는 기계 데이터의 결과 클래스 값은 중요 속성인 $\{SS, TT, TF, TW\}$ 에 의해 결정되어지고, 이들 속성 값과 결과 클래스와의 관계는 [그림 10]과 같이 표현되어진다. [그림 10]은 속성 값에 따른 결과 클래스를 나타낸 그림으로써 결과 클래스의 값이 이산형 속성인 TW 에 의해 가장 많은 영향을 받으며 TW 의 값이 “N”으로 유지되었을 때 결과 클래스가 정상일 확률이 가장 높음을 알 수 있다. 또한 연속형 속성 중에서는 속성 TT 와 TF 의 값을 조정하였을 때 기계 상태가 정상으로 유지될 확률이 속성 SS 를 조정하는 것에 비해 상대적으로 높다는 것을 알 수 있다.

따라서 정책 4에서는 각각의 속성 값에 대해 기계 상태가 정상으로 유지될 확률을 계산하여 이를 아크 조정 비용 정의에 반영하도록 한다. 즉, 기계 상태가 정상으로 유지될 확률이 높은 속성의 조정



[그림 10] 기계 속성과 결과 클래스와의 관계

비용을 상대적으로 낮게 정의함으로써 해당 속성이 보전 대상 속성으로 선택될 확률이 높아지도록 하는 정책으로써 속성별 모수(parameter)를 식 (9)와 같이 정의하고 이를 아크 비용 정의에 반영하도록 한다.

$$r_{a_i} = \sum_{k=1}^{N(a_i)} \left(\frac{I_k(a_i)}{range(a_i)} \times P_{I_k(a_i)}(NS) \right) \quad (9)$$

식 (9)는 각각의 속성 $(a_i, i=1, \dots, n)$ 에 대한 속성 값의 범위가 $range(a_i)$ 이고, 의사결정 트리에 의해 분류된 속성 값의 구간이 $I_k(a_i)$ ($k=1, \dots, N(a_i)$)일 때 정상 유지 확률 모수를 정의한 것이다. 단, 식 (9)에서 $N(a_i)$ 는 의사결정 트리에 의해 분류된 속성 a_i 의 구간 수를 의미하고, $P_{I_k(a_i)}(NS)$ 는 구간 $I_k(a_i)$ 에서 기계 상태가 정상(NS)일 확률을 의미한다.

예를 들면, 속성 TF 에 대한 정상 유지 확률 모수 r_{TF} 는 의사결정 트리를 통해 분류되어진 3개의 구간 “ $600 \leq TF \leq 710$ ”, “ $710 < TF \leq 814$ ”, “ $814 < TF \leq 900$ ”과 각각의 구간에 대해 계산되어진 기계 상태 정상 확률 $P_{I_1(TF)}(NS) = 0.32$, $P_{I_2(TF)}(NS) = 0.54$, $P_{I_3(TF)}(NS) = 0.52$ 를 통해 다음과 같이 계산되어 질 수 있다.

〈표 3〉 정책별 아크 비용 정의

Arc ($i \rightarrow j$)	Cost for each arc ($i \rightarrow j$): $C_{i,j}$			
	정책 1	정책 2 $w_{SS} = w_{TT} = w_{TF} = 1$	정책 3 $w_{SS} = 1.8, w_{TT} = 24.3,$ $w_{TF} = 2.4$	정책 4 $w_{SS} = 1.8, w_{TT} = 24.3,$ $w_{TF} = 2.4$ $r_{SS} = 0.18, r_{TT} = 0.43,$ $r_{TF} = 0.45, r_{TW} = 0.49$
1 → 2	1	$w_{SS} \times \left v_{SS} - \frac{(416+600)}{2} \right $	$w_{SS} \times \left v_{SS} - \frac{(416+600)}{2} \right $	$\frac{w_{SS}}{r_{SS}} \times \left v_{SS} - \frac{(416+600)}{2} \right $
1 → 3	1	0	0	0
2 → 16	1	0	0	0
3 → 4	1	∞	∞	∞
3 → 5	1	0	0	0
4 → 16	1	0	0	0
5 → 6	1	$RCost(W \rightarrow N)$	$RCost(W \rightarrow N)$	$\frac{1}{r_{TW}} \times RCost(W \rightarrow N)$
5 → 7	1	∞	∞	∞
6 → 8	1	$w_{TF} \times \left v_{TF} - \frac{(710+600)}{2} \right $	$w_{TF} \times \left v_{TF} - \frac{(710+600)}{2} \right $	$\frac{w_{TF}}{r_{TF}} \times \left v_{TF} - \frac{(710+600)}{2} \right $
6 → 9	1	$w_{TF} \times \left v_{TF} - \frac{(710+900)}{2} \right $	$w_{TF} \times \left v_{TF} - \frac{(710+900)}{2} \right $	$\frac{w_{TF}}{r_{TF}} \times \left v_{TF} - \frac{(710+900)}{2} \right $
7 → 16	1	0	0	0
8 → 10	1	∞	∞	∞
8 → 11	1	$w_{TT} \times \left v_{TT} - \frac{(64+80)}{2} \right $	$w_{TT} \times \left v_{TT} - \frac{(64+80)}{2} \right $	$\frac{w_{TT}}{r_{TT}} \times \left v_{TT} - \frac{(64+80)}{2} \right $
9 → 12	1	$w_{TF} \times \left v_{TF} - \frac{(710+814)}{2} \right $	$w_{TF} \times \left v_{TF} - \frac{(710+814)}{2} \right $	$\frac{w_{TF}}{r_{TF}} \times \left v_{TF} - \frac{(710+814)}{2} \right $
9 → 13	1	$w_{TF} \times \left v_{TF} - \frac{(814+900)}{2} \right $	$w_{TF} \times \left v_{TF} - \frac{(814+900)}{2} \right $	$\frac{w_{TF}}{r_{TF}} \times \left v_{TF} - \frac{(814+900)}{2} \right $
10 → 16	1	0	0	0
11 → 16	1	0	0	0
12 → 16	1	0	0	0
13 → 14	1	$w_{TT} \times \left v_{TT} - \frac{(74+50)}{2} \right $	$w_{TT} \times \left v_{TT} - \frac{(74+50)}{2} \right $	$\frac{w_{TT}}{r_{TT}} \times \left v_{TT} - \frac{(74+50)}{2} \right $
13 → 15	1	∞	∞	∞
14 → 16	1	0	0	0
15 → 16	1	0	0	0

$$r_{TF} = \left(\frac{110}{300} \times 0.32\right) + \left(\frac{104}{300} \times 0.54\right) + \left(\frac{86}{300} \times 0.52\right) = 0.45$$

본 사례 연구에서 사용되는 나머지 중요 속성인 {SS, TT, TW}에 대한 정상 유지 확률 모수는 다음과 같이 정의될 수 있다.

$$r_{SS} = \left(\frac{216}{400} \times 0\right) + \left(\frac{184}{400} \times 0.4\right) = 0.18$$

$$r_{TT} = \left(\frac{14}{30} \times 0.29\right) + \left(\frac{4}{30} \times 0.52\right) + \left(\frac{6}{30} \times 0.62\right) + \left(\frac{6}{30} \times 0.54\right) = 0.43$$

$$r_{TW} = \left(\frac{1}{2} \times 0.78\right) + \left(\frac{1}{2} \times 0.19\right) = 0.49$$

위에서 제시한 4가지 아크 비용 정의 정책에 의해 정의된 아크 비용은 <표 3>과 같이 정리할 수 있다.

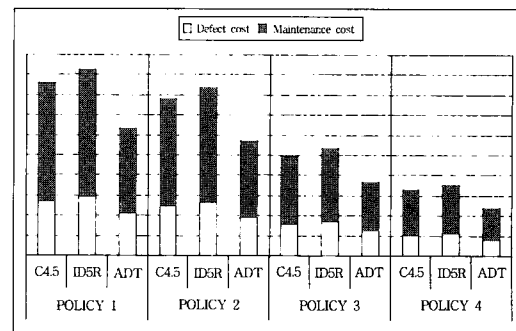
따라서 [그림 8]과 같은 최단 경로 문제 네트워크가 주어지고 네트워크를 구성하는 아크의 비용이 <표 3>과 같이 정의되는 경우, 적응형 의사결정 트리를 통해 진단된 이상상황 노드로부터 최소의 비용으로 전환 가능한 정상 노드를 최단 경로법을 통해 찾아낼 수 있으며 최단 경로법을 통해 탐색된 정상 노드로의 경로는 진단된 이상 상황에 대한 최소 비용 보전 정책을 의미한다.

4.4 비용 정의 정책 비교

본 연구에서는 앞에서 기술한 4가지 비용 정의 정책을 시뮬레이션을 통해 비교함으로써 대상 기계의 특성과 상황에 맞는 효율적인 보전 정책을 수립할 수 있도록 하였다. 시뮬레이션을 통한 비용 정의 정책의 비교는 기계 가공이 진행되는 동안 기계의 속성 값과 기계 상태가 변화하고 이에 따른 이상 상황이 발생하였을 때, 의사결정 네트워크를 통해 최단 경로를 찾아내고 이를 바탕으로 수립된

보전 정책에 의해 이상 상황에 대한 보전을 수행한 결과를 비교함으로써 어떤 정책에 의해 아크 비용을 정의하는 것이 가장 효율적인가를 평가해 볼 수 있다.

[그림 11]은 각각의 비용 정의 정책에 따라 수립된 보전 정책에 의해 기계의 이상 상황에 대한 보전을 수행하였을 때 발생하는 총 비용(total cost)을 비교한 결과로써 총 비용은 이상 상황 발생으로 인해 야기되는 손실 비용(defect cost)과 보전을 수행하기 위해 소요되는 보전 비용(maintenance cost)의 합으로 정의된다. 손실 비용은 발생된 이상 상황의 유형에 따라 다르게 정의될 수 있으며 본 예제에서와 같이 2가지 유형의 이상상황 {US, OS}가 존재하는 경우 각각의 유형에 따른 손실 비용이 정의되어야 한다. 또한 보전 비용은 발생된 이상 상황의 유형과 비용 정의 정책에 따라 달라질 수 있기 때문에 발생된 이상 상황 유형과 비용 정의 정책을 고려한 보전 비용이 정의되어야 한다.



[그림 11] 비용 정의 정책 비교

[그림 11]은 본 연구에서 사용한 시스템 특성 분석 모델인 적응형 의사결정 트리와 기존의 의사결정 트리 구축 방법론인 C4.5, ID5R에 의해 구축된 의사결정 트리를 기반으로 각각의 비용 정의 정책에 의해 아크 조정 비용을 정의하였을 때 발생하는 총 비용을 비교한 결과이다. 실험 결과 본 연구에서 제시한 적응형 의사결정 트리가 기존의 의사결정 트리 구축 방법론에 비해 4가지 정책 모두에 대

해 가장 적은 총 비용이 소요되는 것으로 판명되었다. 또한 4가지 비용 정의 정책 중에서는 정책 4에 의해 아크 조정 비용을 정의하고 이에 따라 보전 정책을 수립하는 것이 가장 적은 총 비용이 소요됨을 알 수 있다. 이는 정책 4에서 아크 조정 비용을 정의할 때 속성별 특성에 따른 단위 조정 비용을 고려하고 정상 상태 유지 확률이 높은 속성이 보전 대상 속성으로 선택될 수 있도록 함으로써 다른 속성에 비해 기계 상태가 정상을 유지할 확률을 높였기 때문이다.

5. 결 론

본 연구에서는 실시간으로 수집되는 데이터를 기반으로 기계 상태를 추론하고 추론된 기계 상태가 기계 고장과 같은 이상 상황 징후를 나타낸 경우 효율적인 보전 정책을 수립하기 위한 방법으로 적응형 의사결정 트리와 최단 경로법을 이용한 상태기반 보전 모델을 제시하였다. 실시간으로 수집되는 기계 상태 데이터에는 시간의 흐름에 따라 변화하는 기계의 특성이 반영되어 있으므로 이러한 특성을 반영한 기계 고장 정보를 추론하기 위해서는 의사결정 트리가 실시간으로 수집되는 데이터에 따라 재구성되어야 한다. 따라서 본 연구에서는 백준걸[1]이 제시한 적응형 의사결정 트리 구축 방법을 이용하여 새로운 데이터의 입력에 따른 효율적인 의사결정 트리 구축을 수행할 수 있도록 하였다. 적응형 의사결정 트리 구축 방법은 기존의 증분 의사결정 트리 구축 방법이 지니는 문제점인 새로운 데이터의 입력에 따라 너무나 빈번하게 트리가 변경된다는 단점을 극복하기 위해 의사결정 트리의 적합도 검사와 노이즈 데이터 검사를 통해 불필요한 트리의 갱신을 방지함으로써 의사결정 트리 구축의 효율성을 높일 수 있다는 장점을 지닌다.

본 연구에서 사용한 적응형 의사결정 트리는 실시간으로 수집되는 기계상태 데이터를 기반으로 기계의 속성과 기계 상태와의 연관 관계를 규명할

수 있는 도구로 사용될 수 있으며 이를 통해 기계 고장과 같은 이상 상황 정보를 추론할 수 있다. 기계 상태 추론 결과 이상 상황이 발생하였을 경우 효율적인 보전을 위해서는 이상 상황이 발생한 원인에 대한 보전 작업이 수행되어야 하는데 본 연구에서는 이를 위해 의사결정 네트워크와 최단 경로법을 이용한 효율적인 보전 정책 수립 방법을 제시하였다.

참 고 문 헌

- [1] 백준걸, 김강호, 김성식, 김창욱, "실시간 기계 상태 데이터베이스에서 데이터 마이닝을 위한 적응형 의사결정 트리 알고리즘", 「대한산업공학회지」, 제26권, 제2호(2000), pp.171-182.
- [2] Boutilier, C., R. Dearden, and M. Goldszmidt, "Stochastic dynamic programming with factored representation," *Artificial Intelligence*, Vol.121, No.1-2(2000), pp.49-107.
- [3] Hernandez-Lerma, O., *Adaptive Markov Control Process*, Springer-Verlag, 1989.
- [4] Kumar, P.R. and P. Varaiya, *Stochastic Systems : Estimation, Identification and Adaptive Control*, Prentice Hall, 1986.
- [5] Lee, J., "Measurement of machine performance degradation using neural network," *Computers in Industry*, Vol.30, No.3(1996), pp.193-209.
- [6] Puterman, M.L., *Markov Decision Process : Discrete Stochastic Dynamic Programming*, John Wiley and Sons, 1994.
- [7] Quinlan, J.R., "Induction of decision trees," *Machine Learning*, Vol.1, No.1(1986), pp. 81-106.
- [8] Quinlan, J.R., *C4.5 : Programs for machine learning*, Morgan Kaufmann, 1993.
- [9] Schlimmer, J.C. and D. Fisher, "A case study of incremental concept induction," *Proceedings of the Fifth International Con-*

- ference on Artificial Intelligence*, Los Altos, California(1986), pp.496-501.
- [10] Utgoff, P.E., "ID5 : An incremental ID3," *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, MI(1988), pp.107-120.
- [11] Utgoff, P.E., "Incremental induction of decision trees," *Machine Learning*, Vol.4(1989), pp.161-186.
- [12] Utgoff, P.E., "An Improved Algorithm for Incremental Induction of Decision Trees," *Proceedings of the Eleventh International Conference on Machine Learning*(1994).
- [13] Williams, J.H., A. Davies, and P.R. Drake, *Condition-based Maintenance and Machine diagnostics*, Chapman and Hall, 1994.
- [14] Witten, I.H., and E. Frank, *Data mining : practical machine learning tools and techniques with JAVA implementations*, Morgan Kaufmann, 2000.