

하부 구성품의 신뢰도 특성을 고려한 복합 시스템의 최적 예방정비 주기 산출

이연호[†] · 이익도 · 이동우 · 손기홍

LIG Nex1, ILS 연구센터

Optimal Preventive Maintenance Period in Complex Systems in Considering Components Reliability Characteristic

Youn-ho Lee · Ik-Do Lee · Dong-Woo Lee · Ki-Hong Sohn

ILS(Integrated Logistics Support) R&D Lab, LIG Nex1, Seongnam, 463-400, Korea

Generally, the life-cycle cost of complex systems composed of several sub systems or equipments such as train, aircraft, weapon systems is spent much more during operation and maintenance phase than development phase. The maintenance cost for maintaining the availability and extending the life span of systems comprise a large proportion of systems operation cost. The cycle of preventive maintenance affects operation and maintenance cost a lot. In this study, we introduce a way minimizing life-cycle cost of systems by calculating more reliable preventive maintenance period than the results of previous study, using systems reliability data considered the reliability and failure effect ratio of sub-systems or components. We can solve the preventive maintenance period problem, known as NP-Hard, as quick as possible by using modified genetic algorithm than using other models introduced in previous study.

Keywords: Preventive Maintenance Period, Periodic Maintenance Period, Failure Effect Rate, Genetic Algorithm(GA), Complex System

1. 서론

시스템의 총 수명주기 비용은 시스템의 개발 및 생산 단계에 비해 운용/유지 단계에서 더 많이 발생하게 된다. 또한 운용/유지 단계에 소요 비용 중 큰 비중을 차지하는 요소 중에 하나가 시스템의 정상적인 가동을 위한 정비활동에 소요되는 비용이다. 이러한 현상은 시스템이 고가이거나 구성이 복잡할수록 더욱 두드러지게 나타난다. 그 예가 바로 군 무기체계 또는 철도, 항공기 산업이라고 할 수 있다. 이러한 유형의 시스템들은 일정 수준 이상의 가용도 유지가 상당히 중요한 요소이고, 이를 위해 수명주기 동안 수행되는 유지보수 활동 역시 막대한 비용이 소요된다. 이러한 문제점들을 해결하기 위한 방안 중

하나가 지속적인 예방정비를 통해 시스템의 품질을 항상 일정 수준 이상으로 유지하여 항상 가용한 상태로 만드는 것이다 (Ben-Daya and Duffuaa, 1995). 하지만 시스템의 신뢰도, 비용 그리고 예방정비 주기는 서로 밀접한 관계에 있다. 따라서 예방정비를 수시로 하게 될 경우 시스템은 높은 신뢰도를 유지하지만 그에 상응하는 높은 비용을 필요로 하게 되고, 반대로 예방정비 주기가 길어지면 비용은 감소하지만, 신뢰도 역시 감소하게 된다. 이와 같이 서로 상충관계에 있는 비용과 시스템 신뢰도를 고려한 최적 예방정비 주기 산출 문제는 이전부터 많은 연구가 진행되어 왔다.

예방정비는 그 효과에 따라 크게 두 가지로 분류할 수 있다. 첫 번째는 일반적인 예방정비로 시간이 지나 노후화된 시스템

이 논문은 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(과제번호: R05-2003-000-10445-0).

[†] 연락저자 : 이연호 연구원, 463-400 경기도 성남시 판교로 333, Tel : 031-8026-4938, Fax : 031-8026-7096, E-mail : younho.lee@lignex1.com
2011년 6월 26일 접수; 2011년 10월 20일 게재 확정.

을 일정 시점 이전의 상태로 회복시켜 주는 것이다. 이와 같은 예방정비는 시스템의 일부 구성품에 대해 수리 또는 교환을 수행하였을 경우로 해석할 수 있다. 두 번째는 Overhaul 개념의 정비이다. 일반적으로 Overhaul 형태의 정비 개념은, 완전분해/검사/수리/개조를 통해 시스템을 새 것과 같게 만드는 정비를 의미한다.

최적 예방정비 주기 선정에 대한 연구는 상기 두 개념의 연장선상에서 수행되어왔다(Lie and Chun, 1995). 하지만 과거 연구들은 시스템을 운용하는데 필수적인 구성품과 부가적인 구성품의 신뢰도를 동일하게 고려하였다. 이는 시스템 구성품들의 중요도를 고려하지 않았다는 부분에서 현실과 거리가 있다. 결국 보다 현실적이고 합리적인 예방정비 주기를 산출하기 위해서는 시스템을 구성하고 있는 구성품들의 고장률과 구성품들이 시스템에 미치는 영향을 함께 고려하여 예방정비 주기를 산출해야 한다. 하지만 시스템의 하부 구성품을 고려하여 예방정비 정책을 수립하기 위해서는 고려해야 할 경우의 수가 기하급수적으로 증가하게 되고, 그에 따라 문제의 계산 복잡도 역시 기하급수적으로 증가하게 된다. 이미 많은 연구가 진행되고 있는 산업재 기계장비 혹은 시스템의 주기 정비 문제(PMP : Periodic Maintenance Problem)는 NP-hard 문제로 알려져 있다(Bar-Noy *et al.*, 2002). 예방정비 대상 시스템의 특성에 따라 고려해야 할 요소들이 다르겠지만 앞서 언급한 요소들을 고려하여 최적 예방정비 주기를 산출해야 할 경우 이에 상응하는 계산 복잡도가 요구된다.

따라서 본 논문에서는 구성품의 신뢰도 뿐만 아니라 고장영향비(Failure effect ratio)를 고려한 시스템의 최적 예방정비 주기를 산출하기 위해 휴리스틱 기법 중 가장 일반적인 기법으로 알려진 유전자 알고리즘을 이용하였다. 유전자 알고리즘은 적자생존이라는 자연현상을 모델링한 탐색방법으로 생명공학뿐만 아니라 다양한 분야의 학문에 적용되어왔다(Godlberg, 1989; Beasley *et al.*, 1992). 또한 수많은 구성품으로 이루어진 복합 시스템의 경우 유전자 알고리즘을 통해 보다 빠른 시간 안에 해를 도출할 수 있으며 이는 군 무기체계 및 철도 산업과 같이 수많은 구성품으로 이루어진 시스템의 예방정비 주기를 산출하는데 있어서 효과적이다(Wang *et al.*, 1996; Marseguerra and Zio, 2000).

본 논문에서는 일반적인 유전자 알고리즘을 변형하여 최적 예방정비 주기 산출 모델에 적용하였다. 변형된 유전자 알고리즘은 일반적인 유전자 알고리즘의 교배(Crossover)와 변이(Mutation) 단계 외에 추가적으로 의도적인 유전자 변형을 수행한다. 변형된 유전자 알고리즘은 이진 유전자형으로 알고리즘을 진행함으로써 세 개의 유전자형으로 알고리즘을 진행한 경우 보다 빠른 시간 안에 세 개의 유전자형으로 구성된 근사 최적해를 도출할 수 있다. 즉, 세 개의 유전자형으로 구성된 해를 이진 유전자형으로 구성된 알고리즘의 진행 속도 및 계산 복잡도로 산출할 수 있다.

2. 관련 연구

정비활동은 사용 횟수와 수명에 따라 낡거나 약화되는 시스템의 가동률을 유지하는데 있어서 매우 중요한 업무이다. 또한 정비 업무 중 예방정비 정책 결정에 대한 문제는 서론에서 언급한 바와 같이 비용과 시스템 가동률이라는 모든 산업의 주요 관심사항을 동시에 고려해야 하는 복합적이며 민감한 문제이다. 따라서 각 산업에서는 예방정비 정책에 대한 많은 연구가 활발히 진행되어 왔다.

먼저 Chaudhuri and Sahu(1977)에서 예방정비 수행 후 시스템 신뢰도가 새 것과 같아지는 것이 아니라 일정수준만큼 이전 시점으로 회복, 즉 시스템의 나이가 감소되는 정비의 불완전성(Imperfect maintenance)을 소개하였으며, 그 이후 정비의 불완전성에 대한 연구는 꾸준히 지속되어왔다. 그 중 대표적인 것이 Nakagawa(1988)의 Sequential 모델로 불완전 예방정비로 인해 시간이 지남에 따라 정비 주기가 점차 짧아지며, Improvement factor를 사용하여 비용 모델을 제시하고 최적 정비 주기를 계산하였다. 또한 Nakagawa *et al.*(1983)에서는 예방정비 기간 중 고장과정은 비균질 포아송 과정(NHPP)을 따른다는 것을 증명하고, 각 고장에 대해서 최소 고장수리를 수행하는 최적의 예방정비 주기를 산출하였다.

Jun *et al.*(1998)에서는 Nakagawa *et al.*(1983)의 개념을 이용하여 각 예방정비 간 고장에 대한 최소 고장수리와 불완전 예방정비가 복합적으로 발생하는 시스템에 대한 최적 예방정비 주기를 산출하였으며, Ji(1999)에서는 Jun *et al.*(1998)의 연구를 이용하여 군 무기체계의 최적 Overhaul 주기를 산출하였다.

하지만 앞선 연구들은 비용분석만을 통한 최적 정비 주기 T 를 산출하였으며, 시스템의 가용도를 고려하지 못한 한계점이 있다. 이와 같은 연구의 한계는 시스템의 하부 구성품을 고려하지 않은 신뢰도 분석을 통해 비용분석을 수행함으로써 정비 주기 산출의 계산은 용이하지만 현실과는 거리가 있는 연구라고 할 수 있다. 게다가 하부 구성품의 신뢰도를 고려하지 않을 경우 최소수리비용 즉, 고장 수리비용이 구성품 개별로 발생하는 것이 아니라 장비 또는 시스템 전체에 대한 비용으로 발생하기 때문에 정확한 정비주기 산출이 어렵다는 문제점을 가지고 있다.

하부 구성품의 신뢰도 분석을 통한 예방정비 주기를 산출한 연구는 Tsai *et al.*(2001)에서 최초로 수행되었다. Tsai *et al.*(2001)에서는 앞서 언급한 고장 수리 및 불완전 예방정비 그리고 예방 교환(preventive replacement)을 적용하여 최적 정비 주기를 산출하였으며 정비 주기 산출을 위해 휴리스틱 기법의 하나인 유전자 알고리즘을 이용하였다.

Bar-Noy *et al.*(2002)에서 예방정비 주기 문제(PMP)가 NP-hard라는 것이 증명되었으며, 이 문제의 하한과 9/8-근사 알고리즘이 제안되었다. 또한 시스템의 복잡도 및 제약사항이 많다면 그에 비례해 예방정비 주기 산출 문제의 계산 복잡도 역시 기하급수적으로 증가한다는 사실이 알려지면서 이와 같

은 문제 대한 한계를 극복하기 위해 휴리스틱 기법을 적용한 연구들이 많이 진행되었다.

Seo and Seo(2003)와 Negnevitsky and Kelareva(1999)에서는 유전자 알고리즘을 이용한 발전소의 최적 정비 주기 산출 모델을 제안하고 있다. Negnevitsky and Kelareva(1999)에서는 예방정비 간 발생하는 발전소의 유휴를 최소화하여 플랜트의 가용도를 최대화하는 정비 주기를 산출하였으며, Seo and Seo (2003)에서는 발전소의 정비 주기를 산출하는데 있어 복합보전비용을 이용한 적합도 함수를 통해 최적의 예방 정비 주기를 산출하였다.

Foong *et al.*(2008)에서는 개미 알고리즘(ACO : Ant Colony Optimization)을 이용한 정비주기 최적화 모델을 실제 수력 발전소에 적용하여 그 결과가 다른 전통적인 방법에 비해 효과적이라는 것을 밝혔다.

이와 같이 많은 연구들이 정비 주기 최적화 문제에 접근하였지만 여전히 많은 한계를 가지고 있다. 먼저 하부 구성품을 고려한 정비 주기 문제는 2001년 처음으로 제안되었으며 이후 연구들은 정비주기 산출 문제의 계산 복잡도가 NP-hard 임이 증명되면서 추가적인 요소를 고려하여 보다 정확한 정비주기를 산출하기보다는 근사 최적화의 방법론에 대한 연구가 수행되었다. 또한 기존 정비주기 산출문제의 연구 대상이 산업 기계 설비 등에 집중되어 시스템의 내적 특성(구성품의 특성 및 시스템에 대한 영향, 구성품 간의 영향 등) 보다는 외적 특성(시스템의 구성 및 배치 등)에 집중되었다. 따라서 정확한 정비주기 산출을 위한 시스템 내적 특성을 고려한 연구는 하부 구성품 중 몇몇 주요 구성품에 대해서만 선택적으로 고려하는 등 제한적인 연구가 수행되었다. 따라서 기존 연구들에 비해 불필요한 정비소요 및 비용을 감소하기 위해서는 보다 정확하고 현실적인 정비 주기 산출이 필요하다. 이러한 문제를 해결하기 위해 본 논문에서는 시스템 신뢰도 계산 시 모든 하부 구성품의 신뢰도를 고려함과 동시에 하부 구성품의 고장 영향비를 이용하였고, 이렇게 도출된 시스템 신뢰도를 기반으로 시스템의 가용도를 일정 수준 유지하는 것을 목표로 하여 정비주기를 산출하였다. 또한 신뢰도 계산 시 하부 구성품의 신뢰도와 고장영향비까지 고려함으로써 복잡해진 정비주기 산출 문제를 빠른 속도로 계산하기 위해 변형된 유전자 알고리즘을 적용하였다.

3. Modeling

본 장에서는 시스템의 수명주기 동안 수행되는 예방정비 시 소요되는 비용을 산출하는 모델과 이를 이용하여 최적 예방정비 주기를 산출하는 유전자 알고리즘을 제안하고자 한다. 제 3.1절에서는 비용 모델에 적용된 기본가정을 기술하고, 제 3.2절에서는 예방정비 시 소요되는 비용 모델에 대해 상세히 설명하였으며, 제 3.3절에서는 유전자 알고리즘의 각 단계별

역할과 비용 모델의 적용에 대해 설명한다.

3.1 가정

시스템의 수명주기 간 발생하는 모든 고장에 대해서는 최소 수리를 수행한다고 가정한다. 최소 수리란 가동 중인 시스템에 고장이 발생하였을 때 해당 시스템의 재가동에 필요한 최소한의 비용이 소모되는 수리를 의미한다. 따라서 최소 수리 전 후 시스템의 노후화 과정은 변함이 없다(Nakagawa *et al.*, 1983). 이에 반해 예방정비는 시스템의 고장과 무관하게 일정 시점에 시스템을 점검하는 것으로 시스템의 노후화 과정에 영향을 미치는 작업을 수행한다. 노후화 과정에 미치는 영향의 정도는 예방정비의 종류에 따라 다르며 예방정비의 종류는 일반 예방정비와 Overhaul 정비로 나눌 수 있다. 일반 예방정비를 수행할 경우 시스템을 완전히 새것으로 만들지는 못하지만 일정 시점 이전의 상태로 회복시키며, Overhaul을 수행할 경우 시스템의 상태가 처음 구매하였을 때와 같은 새 것이 된다.

3.2 비용 모델

본 절에서는 각 예방정비 시 소요되는 비용 모델을 제안하고자 한다. 각 예방정비 시점에서 소요되는 비용은 예방정비 간 발생한 고장 수리의 비용인 사후 비용과 예방정비에 소요되는 예방정비 비용으로 구성되어 있다. 고장 수리의 경우 정비의 효과가 시스템의 신뢰도에 영향을 미치지 않지만 예방정비는 시스템의 신뢰도를 개선시킨다. 또한 지속적으로 유지하고자 하는 시스템의 목표 성능과 예방정비 종류에 따라 각각 다른 비용 모델을 적용한다.

3.2.1 고장률 분포

본 논문에서는 시스템의 고장 발생이 와이불 분포(Weibull Distribution)를 따른다고 가정한다. 개개의 부품에 대한 고장률 분포는 지수분포를 따른다고 알려져 있지만, 고장률이 상이한 여러 부품이 결합되면 이들 간 상호작용으로 인하여 각 부품들의 고장률이 변하기 때문이다. 이미 많은 관련 연구에서 와이불 분포를 이용하여 기계 수명과 관련된 수학적 모델링에 적용해왔다. 앞서 언급하였던 것과 같이 예방정비 간 고장 수리가 발생한다면 고장시점 t 까지의 시간 분포 $F(t)$ 와 이의 확률밀도함수 $f(t)$, 그리고 신뢰도 함수 $R(t)$ 는 다음 식들과 같이 나타낼 수 있다.

$$F(t) = 1 - e^{-(\lambda t)^\alpha} \quad (1)$$

$$f(t) = \alpha \lambda (\lambda t)^{\alpha-1} e^{-(\lambda t)^\alpha} \quad (2)$$

$$R(t) = e^{-(\lambda t)^\alpha} \quad (3)$$

단, α = 형상모수, λ = 척도모수이다. 이때, 고장 시점 t 에서의 고장률 함수 $r(t)$ 는 다음과 같이 정의된다.

$$r(t) = \frac{f(t)}{1 - F(t)} = \alpha\lambda(\lambda t)^{\alpha-1} \quad (4)$$

고장률 함수는 형상모수(α)에 따라 증가 혹은 감소하는 형상을 가진다. $\alpha < 1$ 이면 감소형 고장률 분포, $\alpha = 1$ 이면 지수 분포, $\alpha > 1$ 이면 증가형 고장률 분포가 된다. 본 비용 모델에서는 형상 모수(α)는 시간에 따라 고장률이 증가하는 노화 영향을 적용하기 위하여 $\alpha = 2$ 로 적용하였으며, 척도모수(λ)는 최초 시스템의 고장률을 적용하였다. 예방정비 간 고장 과정은 비균질 포아송 과정(Non-Homo geneous Poisson Process : NHPP)을 따르는 것으로 알려져 있다(Nakagawa *et al.*, 1983). 즉, 고장 발생 전과 후의 시스템 노후화 과정에는 변화가 없으며 증가형 고장률 분포의 경우 시간이 지남에 따라 고장 발생 횟수는 증가하게 된다. 또한 $R(t)$ 는 신뢰도 함수로 매 예방정비 시점에서 각 구성품의 신뢰도를 산출한다.

3.2.2 기본 비용모델

i 시점에서의 예방정비 소요 비용은 이전 예방정비 시점인 $i-1$ 부터 i 시점까지의 고장 발생에 대한 고장 수리비용과 예방정비 비용의 합으로 계산된다.

$$C_i = C_p + C_r \cdot E[N(t)] \quad (5)$$

C_i 는 i 번째 예방정비 시점에서의 소요 비용이며, C_p , C_r 은 각각 예방정비 비용과 고장 수리비용이다. $E[N(t)]$ 는 t 시간 동안의 평균 고장 발생 횟수이다. 즉, i 시점에서의 총 소요 비용은 예방정비 비용, C_p 와 고장정비 비용, $C_r \cdot E[N(t)]$ 의 합으로 나타낼 수 있다.

여기서 시간 간격 $t_{i-1} < t \leq t_i$ 에서의 $E[N(t)]$ 는 NHPP의 결과에 따라 다음과 같이 나타낼 수 있다(Nakagawa *et al.*, 1983).

$$E[N(t)] = \int_{t_{i-1}}^{t_i} r(t) dt \quad (6)$$

따라서 정비시점 i 에서의 총 소요 비용은 다음과 같이 계산할 수 있다.

$$C_i = C_p + C_r \cdot \int_{t_{i-1}}^{t_i} r(t) dt \quad (7)$$

이때 C_p 는 일반 예방정비 비용과 Overhaul 비용으로 나눌 수 있으며, 해당 예방정비 시점에서 수행하는 정비 종류에 따라 적용한다.

3.2.3 신뢰도 개선

가정에서 설명한 바와 같이 예방정비는 시스템의 신뢰도를 일정 시점 이전의 상태로 개선시킨다. 본 논문에서는 이와 같은 예방정비의 개념을 비용 모델에 반영하기 위해 예방정비

수행으로 인한 신뢰도 개선 효과를 시스템의 운용 시작시점을 변경시키는 방법으로 모델을 설계하였다. 시스템 운용 시작시점이란, 시스템이 가동을 시작한 시점을 말한다. 즉, 시스템 가동 후 임의의 시점이 시스템 운용 시작시점과 차이가 클수록 시스템은 노후화 되었다는 것을 의미하며 시스템의 신뢰도가 낮다는 것을 나타낸다. 따라서 예방정비 수행 후 개선되는 신뢰도만큼 시스템운용 시작시점을 예방정비 수행 시점 가까이로 이동시킴으로써 시스템의 신뢰도 개선 효과를 발생시킬 수 있다. 이때 시스템의 신뢰도 개선, 즉 시스템운용 시작시점이 이동하는 정도를 개선률 y 로 표현하였으며 y 는 0과 1사이의 값을 가진다. y 가 1에 가까울수록 신뢰도 개선 효과가 높아서 시스템운용 시작시점이 예방정비 시점에 근접하게 되며, 0에 가까울수록 신뢰도 개선 효과가 낮으므로 시스템운용 시작시점이 이전 예방정비 직후의 시스템운용 시작시점에 근접하게 된다.

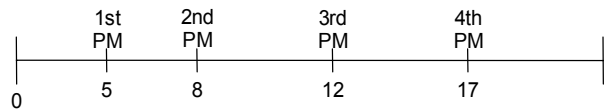


Figure 1. 수명주기 간 예방정비 시점

<Figure 1>과 같이 시점 0에서 시스템이 운용되기 시작하고, 시점 5, 8, 12, 17에서 예방정비가 발생한다고 가정하자. 또한 임의의 예방정비 시점을 t_i , t_i 에 대한 시스템운용 시작시점을 t_s^i , 시스템의 총 운용시간을 t 라고 가정하면, t_i 에서 시스템의 총 운용시간 t 는 $t_i - t_s^i$ 로 나타낼 수 있다. 앞서 설명한 바와 같이 만일 첫 번째 예방정비 수행 후 신뢰도 개선이 없다면 2nd PM에서의 시스템운용 시작시점은 $t_s^2 = t_s^1 = 0$ 이 된다. 반면, $0 < y_1 < 1$ 일 경우, 개선되지 못하는 신뢰도를 고려하여 현재 예방정비 시점인 t_1 에서 $(t_1 - t_0) \cdot (1 - y_1)$ 를 제외한 시점이 두 번째 예방정비(2nd PM)에서의 시스템운용 시작시점 t_s^2 이 된다.

$$\begin{aligned} t_s^2 &= t_1 - (t_1 - t_0) \cdot (1 - y_1) \\ &= 5 - (5 - 0) \cdot (1 - y_1) \\ &= 5y_1 \end{aligned}$$

두 번째 예방정비 수행 후 $y_2 = 0$ 일 경우 $t_s^3 = t_s^2$ 가 되며, $0 < y_2 < 1$ 일 경우 현재 예방정비 시점인 t_2 에서 첫 번째 예방정비 및 현재 예방정비 수행 시 개선되지 못하는 신뢰도를 모두 고려하여 제외한 시점이 세 번째 예방정비(3rd PM)에서의 시스템운용 시작시점 t_s^3 이 된다.

$$t_s^3 = t_2 - (t_1 - t_0) \cdot (1 - y_1) - (t_2 - t_1) \cdot (1 - y_2)$$

이와 같은 방법으로 n 번째 예방정비 수행 후 $n+1$ 번째 예방정비에서의 시스템운용 시작시점의 계산식은 $0 \leq y_n < 1$

일 경우 이전까지의 예방정비로 개선되지 못한 신뢰도를 모두 고려하여 다음과 같이 나타낼 수 있다.

$$t_s^{n+1} = t_n - \{(t_1 - t_0) \cdot (1 - y_1) + \dots + (t_n - t_{n-1}) \cdot (1 - y_n)\}$$

즉, 시점 t_i 에서 적용된 시스템 운용 시작시점이 t_s^i 이고 예방정비 수행 후 개선률 y_i 만큼 신뢰도가 회복되면 다음 예방정비 시점인 t_{i+1} 에서 적용될 새로운 시스템운용 시작시점인 t_s^{i+1} 는

$$t_s^{i+1} = t_i - D_i$$

$$D_i = D_{i-1} + (t_i - t_{i-1}) \cdot (1 - y_i) \quad (8)$$

으로 나타낼 수 있으며, 이때 D_i 는 i 번째 예방정비까지 개선되지 못하는 총 신뢰도이다.

반면 Overhaul의 경우에는 앞서 언급한 개념과 마찬가지로 시스템이 새것과 같아지므로 $y = 1, D_{i-1} = 0$ 의 값을 가지며 $i+1$ 에서 시스템 운용 시작시점은 다음과 같다.

$$t_s^{i+1} = t_i \quad (9)$$

3.2.4 목표 신뢰도(RM) 고려

3.2.4.1 예방정비 분류

본 논문에서는 예방정비를 일반 예방정비와 Overhaul로 분류하였다. 두 예방정비를 분류하는 방법은 일반 예방정비 수행을 통해 회복된 시스템의 신뢰도와 목표 신뢰도와의 차이로 분류한다. 앞서 언급한 신뢰도 개선을 따르면 시스템 또는 시스템의 신뢰도는 일반 예방정비 업무를 수행할수록 개선되지 않는 신뢰도의 누적으로 예방정비 수행 직후의 신뢰도가 점점 낮아진다. 결국 일반 예방정비 직후의 시스템 신뢰도가 목표 신뢰도에 근접하게 되면, 목표 신뢰도를 만족하기 위해 다음 예방정비 시점이 가까워진다. 이와 같은 상황이 반복되면 짧은 시간 안에 무수히 많은 일반 예방정비를 수행하게 되고, 그에 따른 비용 또한 증가한다. 따라서 본 논문에서는 Overhaul 기준선을 1과 목표 신뢰도 차이의 10%를 목표 신뢰도에 더한 값으로 설정하였으며, 예방정비를 통해 개선률 y_i 만큼 회복된 시스템의 신뢰도가 Overhaul 기준선에 미치지 못하면 Overhaul을 수행하도록 하였다.

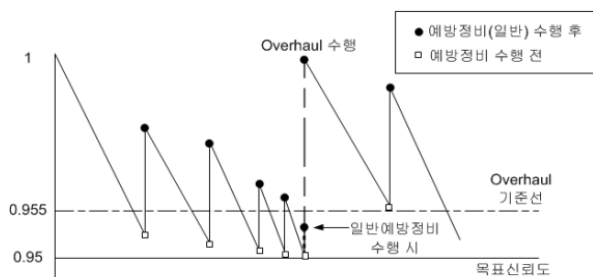


Figure 2. 시간에 따른 신뢰도 변화와 예방정비 분류

3.2.4.2 패널티 부여

패널티는 시스템의 신뢰도를 일정 수준 이상으로 유지하기 위해 설정하는 값으로써 목표로 하는 신뢰도에 대한 허용범위에 따라 값이 달라진다. 패널티 부여 기준은 예방정비 시점에서 시스템의 신뢰도와 목표 신뢰도 값을 비교하여 시스템 신뢰도가 목표 신뢰도 보다 작을 경우 부여된다. 시스템의 신뢰도가 목표 신뢰도보다 높은 경우($R(i) > RM$) 패널티를 적용하지 않은 기본 비용모델인 식 (7)을 각 구성품에 대해 적용하여 총 비용(C_i)을 산출한다.

$$C_i = C_p + C_r \cdot E[N(t)] \quad (10)$$

$$= C_p + C_r \cdot \int_{t_{i-1}}^{t_i} r(t) dt$$

$$= C_p + \sum_k \left(C_{r_k} \cdot \int_{t_{i-1}}^t r_k(t) dt \right)$$

$$\forall k \in N$$

이 때, C_p 는 일반예방정비 비용이 적용되며, C_{r_k} 은 구성품 k 에 대한 고장 수리비용, $r_k(i)$ 는 i 시점에 구성품 k 의 고장률 함수이며, N 은 구성품 집합이다.

반면, $R(i) \leq RM$ 인 경우 앞서 총 비용(C_i)에 패널티를 추가한다. 패널티는 임의의 시점에서 시스템의 신뢰도와 목표 신뢰도의 차이에 비례하여 적용되며 이 때 패널티 값은 아래와 같다.

$$\frac{RM - R(i)}{App} \cdot Penalty \quad (11)$$

이 때, RM 은 목표 신뢰도이며, $R(i)$ 는 i 번째 정비시점에서 시스템의 신뢰도, App 는 허용 범위, $Penalty$ 는 패널티 비용이다. 패널티는 허용 범위인 App 와 패널티 비용인 $Penalty$ 를 통해 시스템의 신뢰도와 목표 신뢰도의 차이에 대한 허용 수준을 조절할 수 있다. 즉, 목표 신뢰도의 만족에 강한 요구가 있다면 App 를 낮게 또는 $Penalty$ 를 높게 설정하고 반대일 경우 App 는 높게 또는 $Penalty$ 를 낮게 설정하면 된다. 따라서 패널티를 부여한 경우의 총 비용은 식 (10)과 식 (11)을 이용하여 아래와 같이 나타낼 수 있다.

$$C_i = C_p + \sum_k \left(C_{r_k} \cdot \int_{t_{i-1}}^{t_i} r_k(i) di \right) \quad (12)$$

$$+ \frac{RM - R(i)}{App} \cdot Penalty, \forall k \in N$$

3.2.5 구성품 고장영향비 고려

본 논문에서는 각 예방정비 시점에서 구성품의 고장영향비를 이용하여 시스템 신뢰도를 산출하고, 해당 시점에서의 시스템의 신뢰도가 목표 신뢰도 보다 높은 값을 유지하고 있는

지 판단한다. 시스템의 신뢰도 계산 시 하부 구성품의 고장이 시스템 운용에 미치는 고장영향에 따라 해당 구성품의 신뢰도를 시스템 신뢰도에 반영한다. 이를 통해 보다 현실적인 시스템 신뢰도 산출이 가능하며 모든 구성품이 동일한 비율로 시스템의 신뢰도에 반영되는 기존 연구에 비해 불필요한 정비 소요가 감소하여 비용 최소화가 가능하다. 목표 신뢰도 만족 여부를 판단하기 위해서는 시스템의 신뢰도 $R(i)$ 를 산출해야 한다. $R(i)$ 는 식 (3)과 시스템운용시간을 이용하여 다음과 같이 나타낼 수 있다.

$$R(i) = e^{-(\lambda_s t)^{\alpha}} = e^{-\{\lambda_s (t_i - t'_i)\}^{\alpha}} \quad (13)$$

하지만 식 (13)의 $R(i)$ 에 적용된 고장률 λ_s 는 시스템 하부 구성품의 고장률을 모두 합한 고장률이다. 실제 시스템의 운용 개념으로 $R(i)$ 를 산출할 경우, 하부 구성품의 고장이 시스템의 운용에 미치는 영향을 고려하여 시스템의 신뢰도를 분석하여야 한다. 따라서 하부 구성품의 고장영향을 고려한 시스템 고장 λ_s^* 는 다음과 같이 쓸 수 있다.

$$\lambda_s^* = \sum_k (\lambda_k \cdot \alpha_k) \quad (14)$$

이때, λ_k 는 구성품 k 의 고장률, α_k 는 구성품 k 의 고장영향비이다. 구성품의 고장영향비는 해당 구성품의 고장이 시스템을 운용함에 있어 미치는 영향에 따라 0과 1사이의 값을 적용하며, 1에 가까울수록 구성품의 이상이 시스템 운용에 미치는 영향이 크다는 것을 의미한다. 이 값은 시스템 및 구성품의 특성, 기능 그리고 사용자의 우선순위 등에 따라 적용 기준과 가중치를 다르게 적용할 수 있다. 본 논문에서는 적용 기준을 <Table 1>과 같이 “시스템 운용 가능/일부 가능/저하/영향 없음”, 4개로 분류하였으며 각각에 대한 가중치를 1, 0.8, 0.4, 0.1로 부여하였다.

Table 1. 고장영향비

값	설명
$\alpha = 1$	구성품 고장으로 시스템 운용 불가
$\alpha = 0.8$	구성품 고장으로 시스템 일부 운용
$\alpha = 0.4$	구성품 고장으로 시스템 운용 저하
$\alpha = 0.1$	구성품 고장으로 시스템 운용 영향 없음

예를 들어, 시스템 S 의 구성품 집합 $N = \{1, 2, 3, 4\}$ 이 존재한다고 가정하고, 구성품의 고장률과 고장영향비 집합이 각각 $\lambda_k = \{0.6, 0.9, 0.8, 0.5\}$, $\alpha_k = \{0.8, 0.4, 1, 0.8\}$ 라고 한다면 시스템 S 의 고장률은

$$\lambda_s = 0.6 \times 0.8 + 0.9 \times 0.4 + 0.8 \times 1 + 0.5 \times 0.8 = 2.04$$

가 된다. 만일 시스템 S 의 고장률을 구성품의 고장영향비를

고려하지 않으면 $\lambda_s = 2.8$ 이다. 이와 같은 차이가 발생하는 이유는 시스템 신뢰도 산출 시 하부 구성품의 고장률이 시스템 운용에 미치는 정도를 구성품 고장률에 고려하였기 때문이다.

이와 같이 방법으로 시스템의 신뢰도 $R(i)$ 를 산출하여 패널티 부여 여부를 결정하고, 총 비용을 산출한다.

3.3 유전자 알고리즘

유전자 알고리즘은 탐험과 탐사의 균형이 잘 이루어진 알고리즘으로 넓은 범위의 할당 문제에서 효율적인 해를 도출하는데 적합하다. 유전자 알고리즘은 각 세대의 잠재적인 해 집합을 검색하여 서로 다른 개체를 선별/교배하여 다음 세대의 개체를 생성한다. 이와 같은 방법으로 알고리즘이 국부 최적(local optimum)에 빠지는 것을 방지하고 전역 최적(global optimum)에 근사한 해를 도출하는데 도와준다. 또한 유전자 알고리즘에서 탐험과 탐사의 조절은 단순한 파라메타 값의 변경을 통해 수행할 수 있기 때문에 원하는 형태의 알고리즘으로 수정이 용이하다. 본 논문에서 적용하고자 하는 유전자 알고리즘은 일반적인 유전자 알고리즘의 변형 형태이다. 변형 유전자 알고리즘은 일반적인 유전자 알고리즘과 달리 교배(Crossover)나 변이(Mutation) 단계 외에 적합도를 산출하는 단계에서 추가적인 개체의 유전자형 변이를 발생시킬 수 있다. 이와 같은 추가적인 유전자형 변이를 통해 유전자형 구성이 $\{0, 1\}$ 에서 $\{0, 1, 2\}$ 로 확장되며, 확장된 유전자형 구성으로 해를 도출한다. 하지만 적합도 산출 단계가 끝나면 확장된 유전자형은 다시 기존의 이전 유전자형으로 복귀하며 알고리즘의 다음 단계를 수행한다. 이로 인해 유전자형이 3종류인 유전자 알고리즘에 비해 빠른 시간 안에 원하는 해를 얻을 수 있다.

3.3.1 유전자형 표현

일반적으로 유전자 알고리즘은 각각의 가능 해를 표현하는 개체(individual)들의 조합으로 전개되며 개체는 염색체(chromosome)의 조합으로 구성되어 있다. 본 논문에서는 하나의 개체가 시스템의 수명과 동일한 크기의 염색체로 이루어져 있으며, 유전자 알고리즘의 각 단계를 수행하는 유전자형은 이진수(0, 1)로 표현한다. 즉, i 번째 유전자형이 1일 경우, 해당 시스템은 i 년에 예방정비를 수행한다는 의미이며, 다음으로 j 번째 유전자형이 1로 나타났다면 j 번째 예방정비에서는 예방정비의 시간 간격 $j-i$ 년을 이용하여 j 번째 예방정비에 소요되는 비용을 계산한다.

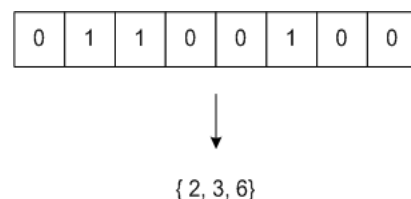


Figure 3. 유전자형과 예방정비 시점

또한, 단 한 번의 예방정비를 수행하지 않는 경우를 제외하기 위해서 염색체의 마지막 염색체는 반드시 1이 되도록 설정하여 마지막 시점에서의 시스템 상태에 대한 소요 비용을 산출한다.

3.3.2 적합도 함수(Fitness Function)

유전 알고리즘에서 사용하는 적합도 함수는 본 논문의 목적 함수인 비용을 최소화 하기 위해 앞서 설명된 총 수명주기 간 소요비용 C_i 의 역수를 적용하며 다음과 같이 나타낼 수 있다.

$$Fitness = 1 / \sum_i C_i, \quad \forall i \in T \quad (15)$$

여기서 총 소요비용인 C_i 를 계산하기 위해서는 앞선 비용 모델을 참조하여 예방정비 종류와 패널티 부여 여부를 결정해야 한다. 각 개체가 가지고 있는 염색체 중 유전자형이 “1”인 염색체에 대해 예방정비 종류를 판단한다. i 번째 정비시점이 일반 예방정비로 결정될 경우 유전자형은 그대로 “1”로 유지되고, $i+1$ 번째에 적용될 개선률 y_{i+1} 과 개선되지 못하는 신뢰도 D_i 를 계산한다. 반면 i 번째 정비시점이 Overhaul로 결정될 경우 유전자형은 “2”로 변경되고, $y_{i+1} = 1$, $D_i = 0$ 으로 설정된다. 도출되는 해는 변형된 유전자형을 포함하며 다음 단계로 진행될 때에는 기존 유전자형인 “1”로 변경한 후 진행된다.

3.3.3 교배(Crossover)

교배 단계에서는 보다 우수한 개체의 염색체를 다음 세대로 상속하기 위해 룰렛 휠 선택(Roulette Wheel Selection)을 통해 선별된 개체들만을 교배에 이용한다. 룰렛 휠 선택을 이용하여 총 개체의 절반을 선별하고, 이들 중 가장 높은 적합도를 가지는 개체가 나머지 선별된 개체들과 교배를 수행한다. 교배는 부모 개체를 3등분하여 이를 자식 개체가 나눠가지는 2점 교차를 수행한다. 이를 통해 자식 개체가 부모를 닮지만 부모와는 다른 새로운 개체를 생성한다. 또한 우수 개체 보존을 위해 최고 적합도를 가진 개체는 다음 세대에 다시 나타나도록 한다.

3.3.4 변이(Mutation)

각 개체에 대해 변이가 발생할 확률은 1%이며, 변이가 발생할 개체는 각 염색체 중 무작위로 하나를 선택하여 유전자형을 0에서 1, 또는 1에서 0으로 변경한다.

유전자 알고리즘 프로세스는 <Figure 4>와 같으며 각 세대의 해집합은 적합도 함수를 계산한 후에 산출된다. 알고리즘을 진행하면서 산출되는 최적해는 이전까지의 최적해와 비교하여 보다 높은 적합도를 가지는 해가 최종 해로 저장된다.

4. 실험 및 분석

본 장에서는 제안 모델과 두 비교 모델을 비교하여 제안 모델

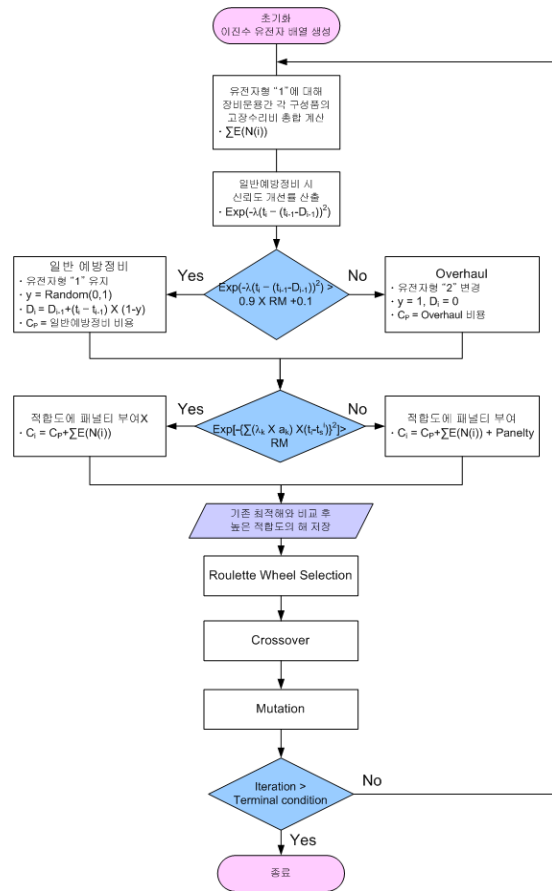


Figure 4. 유전자 알고리즘 프로세스

의 성능을 확인하고자 한다.

4.1 비교 모델

실험에서 사용한 비교 모델은 시스템 구성품의 고장률만 고려하고 고장영향비는 고려하지 않은 모델(이하 Component라고 칭함)이다. 비교 모델은 예방정비 주기 산출시 하부 구성품을 고려할 경우 계산 복잡도의 증가라는 한계를 극복하기 위해 휴리스틱 기법과 함께 최근 많이 연구되고 있는 모델이다. 실험에서는 Component 모델과 제안 모델을 동일한 조건에서 비교하기 위해 목표 신뢰도를 고려할 수 있도록 구현하였다.

4.2 실험

실험에는 실제 군 무기체계인 XX 체계의 특정 장치에 대한 분석 데이터를 이용하였다.

제안 모델 및 비교 모델은 MATLAB을 이용하여 구현하였으며 실험은 모두 동일한 성능의 컴퓨터상에서 수행하였다. 실험에 사용된 설정 값들은 <Table 2>과 같다. 본 논문에서는 <Table 2>의 설정 값들을 이용하여 소요 비용, 평균 예방정비 횟수 그리고 예방정비 직전의 시스템 신뢰도의 평균을 산출하여 상호 비교하였다.

Table 2. 초기 설정값

구분	설명	값
Life	시스템 수명 주기	25년
No_Individual	세대당 개체 수	500
RM	목표 신뢰도	0.9
Iteration	반복 횟수	500
Panelty	패널티 비용	10억 원
P_C	일반 예방정비 비용	10만 원
P_O	Overhaul 비용	3천만 원
Improve_rate	신뢰도 개선률	0.75
APP	신뢰도 허용 범위	0.001
AOR	연간 운용시간	100시간

먼저 제안하는 모델의 신뢰도 변화 그래프는 <Figure 5>와 같다. <Figure 5>를 보면 각 예방정비 시점에서 신뢰도 그래프가 수직 상승하는 것을 확인할 수 있으며 이 부분이 바로 예방정비를 통해 시스템의 신뢰도가 개선되는 것이다.

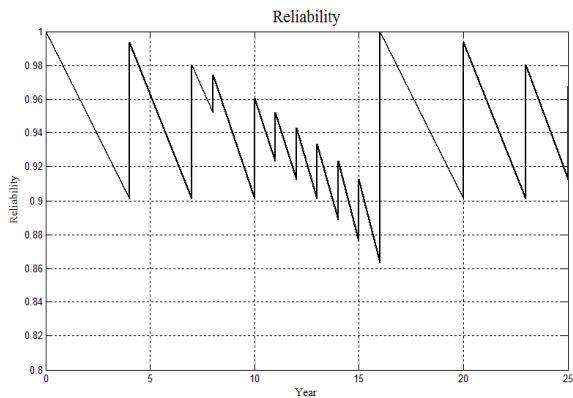


Figure 5. 제안 모델 적용에 따른 신뢰도 변화

<Table 3>에는 실험 결과를 예방정비 시점과 소요비용으로 정리하였다. Component 모델의 경우 소요 비용을 두 종류의 값으로 나타내었다. 상단에 있는 값은 패널티가 포함된 비용, 하단은 패널티가 포함되지 않은 순수소요비용이다.

실험 결과 제안 모델이 Component 모델에 비해 적은 비용을 필요로 하는 것으로 나타났다. 이는 Component 모델이 제안 모델에 비해 더 많은 예방정비를 수행하였기 때문이다.

Table 3. 각 모델에 따른 예방정비 주기 결과

Model	Solution(PM time)																									Cos(₩) ($\times 10^7$)
	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	
Component	0	1	0	1	1	1	1	1	1	2	0	1	0	1	1	1	1	1	1	2	0	1	0	1	1	17,318.3 7.299
Proposed	0	0	0	1	0	0	1	1	0	1	1	1	1	1	2	0	0	0	1	0	0	1	0	1	7,159.8 4.249	

<Figure 6>에는 제안 모델과 비교 모델을 활용하여 시스템을 10년 운용 시 신뢰도 변화가 나타나있다. <Figure 6>을 보면 제안 모델이 비교 모델에 비해 그래프의 기울기 즉, 신뢰도 감소율이 낮은 것을 알 수 있다. 이는 제안 모델이 구성품의 고장영향비를 고려하여 시스템 신뢰도를 산출하면서 나타난 결과로 볼 수 있다.

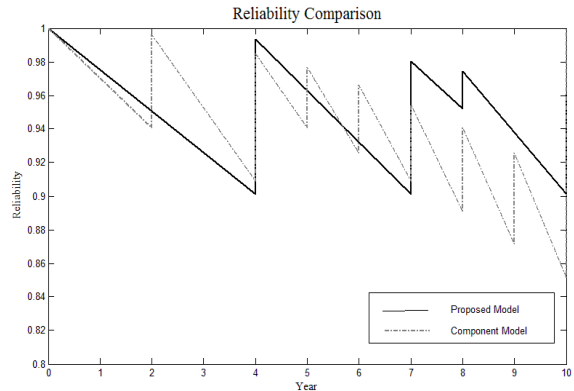


Figure 6. 각 모델별 시간에 따른 신뢰도 변화 비교

알고리즘 진행 동안의 비용 변화는 <Figure 7>에 나타나있다. 제안 모델의 경우 Iteration이 10회 이전에 최적해가 결정되었다. Component 모델은 시작 지점과 약 50회 지점에서 크게 감소한 이후로 비용 변화가 없다. Iteration을 500회로 수행하였는데 결과를 100회 까지만 제시한 이유는 50회부터 500회까지는 두 모델 모두 비용 변화가 없기 때문이다.

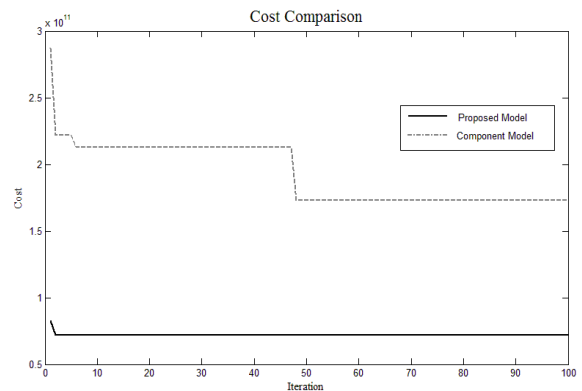


Figure 7. 모델별 Iteration에 따른 비용 변화 비교

<Figure 8>은 제안 모델과 두 비교 모델의 예방정비 직전의 신뢰도 비교 그래프이다. 예방정비 직전의 신뢰도란, 예방정비 시점에서 정비 수행 전의 시스템 신뢰도로 총 수명주기 동안 신뢰도가 가장 낮은 부분들로 볼 수 있다.

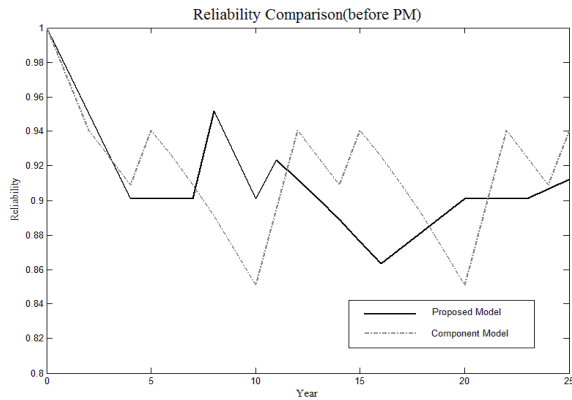


Figure 8. 각 모델별 시간에 따른 신뢰도 변화 비교

<Figure 8>을 보면 제안 모델의 신뢰도 변화폭이 Component 모델에 비해 더 작은 것을 알 수 있으며 목표 신뢰도(90%)에 보다 근접한 것을 알 수 있다.

<Table 4>를 보면 예방정비 수행 전 시스템의 신뢰도 값을 보다 상세히 확인할 수 있다.

Table 4. 각 모델별 예방정비 직전의 신뢰도

	Component		Proposed	
	PM time	Reliability	PM time	Reliability
S o l u t i o n	2	0.941	4	0.901
	4	0.909	7	0.901
	5	0.941	8	0.952
	6	0.926	10	0.901
	7	0.909	11	0.923
	8	0.891	12	0.912
	9	0.872	13	0.901
	10	0.851	14	0.889
	12	0.941	15	0.876
	14	0.909	16	0.863
	15	0.941	20	0.901
	16	0.926	23	0.901
	17	0.909	25	0.912
	18	0.891		
	19	0.872		
	20	0.851		
	22	0.941		
	24	0.909		
	25	0.941		
	Ave		0.913	

<Table 4>에서 Component 모델과 제안 모델의 평균 예방정비 횟수를 비교해보면 Component 모델은 19회, 제안 모델은 13회를 수행한 것을 알 수 있다. 제안 모델은 앞서 <Figure 6>에서 설명한 바와 같이 Component 모델에 비해 신뢰도가 완만히 떨어지기 때문에 보다 긴 정비주기를 가지게 되었고 그로 인해서 정비 횟수로도 목표 신뢰도를 만족할 수 있었다. 게다가 정비 횟수가 적다는 것은 그만큼 정비업무 수행에 따른 부수적인 비용이나 자원의 손실이 적다는 것을 의미한다. 따라서 정비주기 산출시 제안 모델을 적용할 경우 <Table 3>에 제시된 제안 모델과 Component 모델의 비용 차이보다 더 큰 경제적 이득을 기대할 수 있다.

5. 결론

본 논문에서는 군 무기체계, 항공, 철도와 같은 복합 시스템에 대한 최적 예방정비 주기를 산출하기 위해 시스템 구성품의 신뢰도 및 고장영향을 고려하여 정비주기를 산출하는 모델을 제안하였다. 구성품 및 구성품의 고장영향은 매 정비시점에서 목표 가용도의 만족 여부를 판단할 때 적용되었으며 제안한 모델의 성능을 확인하기 위해 비교 모델에 대해 동일한 설정 값을 적용하여 실험을 수행하였다. 제안 모델은 수명주기 동안 비교 모델에 비해 적은 유지비용으로 보다 안정적인 신뢰도를 유지하였다. 하지만 가장 큰 이점은 예방정비 수행 횟수이다. 실험 결과를 확인해 보면 제안 모델이 Component 모델에 비해 약 40~50%가 적은 예방정비를 수행하는 것으로 나타났다. 본 논문에서는 시스템 수명주기 간 소요자원으로 고장 수리비 및 예방정비 비용만을 고려하였지만 현실에서는 비용뿐만 아니라 시간, 인원 그리고 여러 부수지원 비용 등이 고려된다. 따라서 예방정비 횟수가 적을수록 정비 업무에 소요되는 전체적인 자원이 감소하며 제안 모델이 비교 모델에 비해 보다 효율적인 정비 업무를 수행하는 것으로 해석할 수 있다.

앞으로의 연구에는 구성품의 고장영향비 뿐만 아니라 구성품들 간 상호작용을 고려한 예방정비 주기를 산출할 필요가 있다. 하나의 시스템은 유기적 관계를 맺고 있는 구성품들간의 상호작용을 통해 자신의 임무를 수행하게 된다. 따라서 각 구성품의 고장영향을 개별적으로만 취급할 것이 아니라, 기능적 또는 물리적 연관관계를 파악하여 시스템에 미치는 영향을 고려해야 할 것이다.

참고문헌

Bar-Noy, A., Bhatia, R., Naor, J-S., and Schieber, B. (2002), Minimizing service and operation costs of periodic scheduling, *Mathematics of Operations Research*, 27(3), 518-544.

Beasley, D., Bull, D., and Martin, R. (1993), *An overview of genetic algorithm: part 1, fundamentals*, University Computing, 15(2), 58-69.

- Beasley, D., Bull, D., and Martin, R. (1993), *An overview of genetic algorithm; part 2, research topics*, University Computing, **15**(4), 170-181.
- Ben-Daya, M. and Duffuaa, S.-O. (1995), Maintenance and quality : the missing link, *Journal of Quality in Maintenance Engineering*, **1**(1), 20-26.
- Chaudhuri, D. and Sahu, K.-C. (1977), Preventive maintenance intervals for optimal reliability of deteriorating system, *IEEE Transactions on Reliability*, **26**(5), 371-372.
- Foong, W.-K., Simpson, A.-R. Maier, H.-R., and Stolp, S. (2008), Ant colony optimization for power plant maintenance scheduling optimization-a five-station hydropower system, *Annals of Operations Research*, **159**(1), 433-450.
- Goldberg, D.-E. (1989), *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, Boston, Massachusetts.
- Ji, G.-Y. (1999), *A Study on the Overhaul Cycle Determination with Minimum Maintenance Cost for New System*, Dept. of Industrial Engineering Graduate School of Engineering, Changwon, Kyungnam, Korea, 1999.
- Jun, C.-H., Park, K.-S., Hwang, W.-E., and Ju, M.-B. (1998), Estimation of Failure Rates and Maintenance Interval under Periodic Maintenance Policies with Minimal Repair at Failure-A Case of K Steelworks, *Journal of the Korean Institute of Plant Engineering*, **3**(1), 217-228.
- Lie, C.-H. and Chun, Y.-H. (1986), An algorithm for preventive maintenance policy, *IEEE Transactions on Reliability*, **35**(1), 71-75.
- Marseguerra, M. and Zio, E. (2000), Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation, *Reliability Engineering System Safety*, **68**(1), 69-83.
- Nakagawa, T. and Kowada, M. (1983), Analysis of a System with Minimal Repair and Its Application to Replacement Policy, *European Journal of Operational Research*, **12**(2), 176-182.
- Nakagawa (1988), Sequential imperfect preventive maintenance policies, *IEEE Transactions on Reliability*, **37**(3), 295-298.
- Negnevitsky, M. and Kelareva, G. (1999), Genetic Algorithms for maintenance scheduling in power systems, *Proc. 6th Int. Conf. on Neural Information*, **2**, 447-452.
- Seo, K.-K. and Seo, J.-H. (2003), Decision-making Method of Optimum Inspection Interval for Plant Maintenance by Genetic Algorithms, *Journal of the Society of Korea Industrial and Systems Engineering*, **26**(2), 1-8.
- Tsai, Y.-T., Wang, K.-S. and Teng, H.-Y. (2001), Optimizing preventive maintenance for mechanical components using genetic algorithms, *Reliability Engineering and System Safety*, **74**(1), 89-97.
- Wang, K.-S., Chang, W.-H., Tsai, Y.-T., and Hsu, F.-S. (1996), Using Genetic Algorithm Planning Preventive Replacement of Components in a System, *Proc. 13th national Conf. of the Chinese Society of Mechanical Engineers*, Taiwan : National Taiwan University, 271-278.