

혼합 중복전략을 고려한 시스템 신뢰도 최적화 문제

김홍섭^{1*} · 전건욱²

¹공군 군수사령부 / ²국방대학교 운영분석학과

A Reliability Optimization Problem of System with Mixed Redundancy Strategies

Heung-Seob Kim¹ · Geonwook Jeon²

¹Logistics Management Wing, Air Force Logistics Command, Daegu, 701-799

²Department of Operations Research, Korea National Defense University, Seoul, 122-875

The reliability is defined as a probability that a system will operate properly for a specified period of time under the design operating conditions without failure and it has been considered as one of the major design parameters in the field of industries. Reliability-Redundancy Optimization Problem(RROP) involves selection of components with multiple choices and redundancy levels for maximizing system reliability with constraints such as cost, weight, etc. However, in practice both active and cold standby redundancies may be used within a particular system design. Therefore, a redundancy strategy(active, cold standby) for each subsystem in order to maximize system reliability is considered in this study. Due to the nature of RROP, i.e. NP-hard problem, A Parallel Particle Swarm Optimization(PPSO) algorithm is proposed to solve the mathematical programming model and it gives consistently better quality solutions than existing studies for benchmark problems.

Keyword: reliability optimization, redundancy strategy, cold standby, parallel particle swarm optimization(PPSO)

1. 서론

신뢰도(Reliability)란 임의 시스템이 주어진 운용환경 하에서 의도한 기간 동안 의도된 기능을 정상적으로 수행할 확률을 의미한다. 이러한 신뢰도의 중요성은 1986년 Chernobyl 원전사고, 우주왕복선 Challenger호의 폭발사고 및 2003년 Colombia호의 귀환 중 폭발사고 등의 대표적인 사례로써 이미 입증되었다. 오늘날에는 일상에서 사용되는 생활용품에서부터 각종 전자제품, 자동차, 선박, 항공기 등의 첨단 시스템에 이르기까지 기능과 구조가 복잡화, 정교화 되면서 고장 발생에 대한 우려도 커지고 있다. 반면 소비자들은 높은 품질과 신뢰도를 갖는 제품을 요구하고 있으며, 제품의 품질과 신뢰도는 기업의 경쟁

력이 되었다. 이에 따라, 신뢰도는 다양한 산업분야의 시스템 설계에 있어 중요한 척도로 고려되고 있다. 또한 시스템의 신뢰도 향상이 제조 또는 판매 이후에 수행될 경우, 리콜(Recall) 등에 의한 막대한 비용 지출과 기업 이미지의 실추가 수반되므로, 시스템의 신뢰도는 기획 또는 설계 초기단계부터 고려되는 것이 바람직하다.

현재까지 주로 연구되어온 신뢰도-중복 최적화 문제(RROP : Reliability-Redundancy Optimization Problem)는 각 서브시스템의 중복형태를 활성중복(Active Redundancy)만으로 고려하여, 비용 및 무게 등의 가용한 자원 내에서 시스템의 신뢰도를 최대화할 수 있는 최적의 부품을 선택하고 부품수를 결정하는 문제였다. 하지만 실제 산업현장에서는 활성중복과 대기중복(Stan-

*연락처 : 김홍섭, 701-799 대구광역시 동구 아양로 352 사서함 304-170 군수관리단 소요관리처, Fax : 02-309-6233,

E-mail : afrotc02@naver.com

투고일(2011년 11월 18일), 심사일(1차 : 2011년 12월 10일), 게재확정일(2011년 12월 10일).

db by Redundancy)을 혼용하여 시스템을 설계하게 된다.

따라서 본 연구에서는 Coit(2003)이 제안한 각 서브시스템의 최적 중복전략(Redundancy Strategy : Active or Standby)을 추가적으로 고려하여 시스템의 신뢰도를 최대화하는 문제를 다룬다. 활성중복은 서브시스템에 배치된 모든 부품이 작동하면서 부하(Load)를 공유하고 있는 구조이다. 대기중복은 Cold-Standby로 고려하였으며, 이는 서브시스템에 중복설계된 예비(Sencondary) 부품들은 작동 중인 주(Primary) 부품에 고장이 발생하기 전까지는 유휴(Powered-off) 상태로 대기하고 있으며, 주 부품에 고장발생시 대기 중인 예비 부품으로 전이(Switchover)하여 서브시스템의 기능을 유지하는 구조이다. 따라서 대기 중인 부품은 고장이 발생하지 않는다. 또한 고장감지기(Failure detector and Switch)는 항상 작동상태를 유지하면서 고장을 탐지하며, 신뢰도가 1.0 이하인 불완전 스위치(Imperfect S/W)로 고려하였다.

Chern(1992)은 RROP를 NP-hard인 조합최적화 문제(Combinatorial Optimization Problem)로 분류하였다. 이러한 조합최적화 문제의 해법으로는 다양한 최적해 해법(Optimal Algorithm)과 Heuristic 기법이 제안되었으나, 최적해 해법은 서브시스템 개수의 증가 등 문제가 확장되는 경우 계산시간이 많이 소요되어 적용이 곤란하고, Heuristic 기법은 해결하고자 하는 문제의 특성에 국한되어 개발되는 단점을 가지고 있다. 따라서 최근에는 다양한 문제에 공통적으로 적용이 가능한 메타 휴리스틱(Meta-Heuristic) 기법을 활용한 연구가 주로 수행되고 있다. 본 연구에서는 일부 Heuristic 기법이 포함된 병렬 개체군집최적화(PPSO : Parallel Particle Swarm Optimization) 알고리즘을 제안하였다.

제 2장에서는 중복전략별 서브시스템의 신뢰도 확률모형과 RROP의 수리모형을 설명하고, 제 3장에서는 RROP의 해법으로 제안하는 PPSO 알고리즘에 대해 설명한다. 제 4장에서는 수치실험 결과를 기존 연구와 비교하고, 혼합 중복전략을 고려했을 때의 신뢰도 향상 효과를 분석한다. 제 5장에서는 결론 및 향후 연구방향을 제시한다.

2. 시스템 신뢰도 및 수리모형

본 장에서는 병렬과 k -out-of- n 구조의 서브시스템에 대해 활성중복과 대기중복인 경우의 신뢰도 확률모형과 서브시스템들이 직렬로 연결된 <Figure 1>과 같은 시스템의 신뢰도를 최

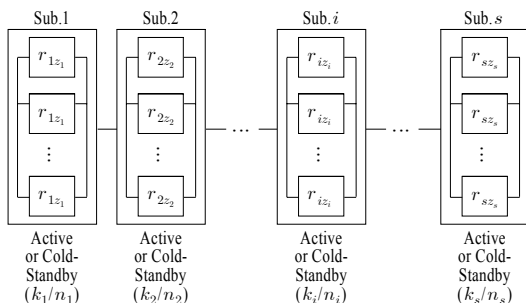


Figure 1. System Configuration for This Study

대화하기 위한 수리모형을 제시한다. 사용하는 표기(Notation)는 다음과 같다.

<Notation>

- i : 서브시스템의 Index($i = 1, 2, \dots, s$)
- s : 시스템을 구성하는 서브시스템의 수
- z_i : 서브시스템 i 에 선택된 부품의 Index ($z_i = 1, 2, \dots, q_i$)
- q_i : 서브시스템 i 의 대안 부품 수
- k_i : 서브시스템 i 의 최소 부품수
- n_i : 서브시스템 i 를 구성하는 부품수
- t : 시스템의 운용시간
- $R_s(t)$: 시스템의 신뢰도
- $R_{i,z_i}(t)$: 부품 z_i 로 구성된 서브시스템 i 의 신뢰도
- $r_{i,z_i}(t)$: 부품 z_i 의 신뢰도
- $r_{k_i,z_i}(t)$: 부품 z_i 를 직렬로 k_i 개 연결시의 신뢰도
- $\rho_i(t)$: 서브시스템 i 의 고장감지기 신뢰도
- n_{i,z_i}^c : 서브시스템 i 에 부품 z_i 를 선택할 때, 대기 중복이 추천되는 최대 부품수
- f_{i,z_i} : 부품 z_i 의 고장시간 분포 함수
- $f_{i,z_i}^{(x)}$: f_{i,z_i} 의 x 차 합성적(x -convolution)
- $m_{i,z_i}, \lambda_{i,z_i}$: 에랑(Erlang) 분포의 모수
- λ_{i,z_i} : 지수(Exp) 분포의 모수
- C, W : 시스템의 비용, 무개의 제약
- $n_{\max,i}$: 서브시스템 i 의 최대(제약) 부품 수
- c_{i,z_i}, w_{i,z_i} : 부품 z_i 의 비용, 무게
- $c_{s,i}, w_{s,i}$: 서브시스템 i 의 고장감지기 비용, 무게
- N, A, S : 중복전략의 Code (N : 중복없음, A : 활성중복, S : 대기중복)

2.1 병렬 서브시스템의 신뢰도

고장시간 분포가 Erlang($m_{i,z_i}, \lambda_{i,z_i}$)을 따르는 부품의 운용시간 t 에서의 신뢰도는 식 (1)과 같으며, 이러한 부품으로 구성된 활성중복 서브시스템의 운용시간 t 에서의 신뢰도는 동시에 작동을 시작한 n_i 개의 부품 중 1개 이상의 부품이 작동하고 있을 확률로 식 (2)와 같다.

$$r_{i,z_i}(t) = \exp(-\lambda_{i,z_i}t) \sum_{l=0}^{m_{i,z_i}-1} \frac{(\lambda_{i,z_i}t)^l}{l!} \quad (1)$$

$$R_{i,z_i}(t) = 1 - [1 - r_{i,z_i}(t)]^{n_i} \quad (2)$$

대기중복 서브시스템은 작동을 시작한 주 부품에 고장이 발생하면, 이를 감지하고 대기 중인 부품으로 부하를 전이하는

과정이 대기 부품의 수 $(n_i - 1)$ 만큼 반복되고, n_i 번째의 고장이 발생하면 서브시스템은 기능을 상실하게 된다. 고장 감지와 대기 부품으로의 전이에 성공할 확률은 고장감지기의 신뢰도 $\rho_i(t)$ 가 된다. Coit(2003)은 이러한 확률과정(Stochastic process)을 식 (3)과 같이 유도하고, 이에 대한 근사 신뢰도를 식 (4)와 같이 제시하였다.

$$R_{i_{z_i}}(t) = r_{i_{z_i}}(t) + \sum_{x=1}^{n_i-1} \int_0^t \rho_i(u) r_{i_{z_i}}(t-u) f_{i_{z_i}}^{(x)}(u) du \quad (3)$$

$$\widetilde{R}_{i_{z_i}}(t) = r_{i_{z_i}}(t) + \rho_i(t) \sum_{x=1}^{n_i-1} \int_0^t r_{i_{z_i}}(t-u) f_{i_{z_i}}^{(x)}(u) du \quad (4)$$

$(0, t]$ 동안 고장이 발생한 부품의 수를 $N_i(t)$ 로 정의하면, $N_i(t) < n_i$ 인 경우 $N_i(t)$ 는 Erlang 도착간격 시간을 갖는 재생과정이고, Erlang($m_{i_{z_i}}, \lambda_{i_{z_i}}$)는 i.i.d. Exp($\lambda_{i_{z_i}}$)의 m 차 합성적이다. 따라서 $N_{i,m}^*(t)$ 을 Poisson(λ) 과정이라 하면, $P[N_i(t) < x] = P[N_{i,m}^*(t) < mx]$ 가 되어 식 (5)가 도출된다.

$$\int_0^t r_{i_{z_i}}(t-u) f_{i_{z_i}}^{(x)}(u) du = \exp(-\lambda_{i_{z_i}} t) \sum_{l=m_{i_{z_i}}x}^{m_{i_{z_i}}(x+1)-1} \frac{(\lambda_{i_{z_i}} t)^l}{l!} \quad (5)$$

따라서 식 (5)를 식 (4)에 적용하고, Poisson($\lambda_{i_{z_i}}$)의 누적확률분포함수(c.d.f)를 이용하여 정리하면 대기중복 시의 서브시스템 신뢰도는 식 (6)과 같이 도출된다.

$$\begin{aligned} \widetilde{R}_{i_{z_i}}(t) = & [1 - \rho_i(t)] \left[\exp(-\lambda_{i_{z_i}} t) \sum_{l=0}^{m_{i_{z_i}}-1} \frac{(\lambda_{i_{z_i}} t)^l}{l!} \right] \\ & + \rho_i(t) \left[\exp(-\lambda_{i_{z_i}} t) \sum_{l=0}^{m_{i_{z_i}}n_i-1} \frac{(\lambda_{i_{z_i}} t)^l}{l!} \right] \end{aligned} \quad (6)$$

2.2 k-out-of-n 서브시스템의 신뢰도

고장시간 분포가 Exp($\lambda_{i_{z_i}}$)를 따르는 부품의 운용시간 t 에서의 신뢰도는 식 (7)과 같으며, 이러한 부품으로 구성된 활성중복 k-out-of-n 서브시스템의 운용시간 t 에서의 신뢰도는 동시에 작동을 시작한 n_i 개의 부품 중 k_i 개 이상의 부품이 작동하고 있을 확률로 식 (8)과 같다.

$$r_{i_{z_i}}(t) = \exp(-\lambda_{i_{z_i}} t) \quad (7)$$

$$R_{i_{z_i}}(t) = \sum_{k=k_i}^{n_i} \binom{n_i}{k} [r_{i_{z_i}}(t)]^k [1 - r_{i_{z_i}}(t)]^{n_i-k} \quad (8)$$

대기중복 서브시스템은 n_i 개의 부품 중 k_i 개가 작동을 시작하며, 작동 중인 부품에 고장이 발생하면 이를 감지하고 대기

중인 부품으로 부하를 전이한다. 이러한 과정은 대기 부품의 수 $(n_i - k_i)$ 만큼 반복되고, $(n_i - k_i + 1)$ 번째의 고장이 발생하면 서브시스템은 기능을 상실하게 된다. 이때, 고장감지기의 신뢰도 $\rho_i(t)$ 는 고장감지와 대기 부품으로의 전이에 성공할 확률이 된다.

또한, 서브시스템이 정상 작동하는 동안 항상 k_i 개의 부품이 작동 중에 있고, 부품의 고장시간 분포 Exp($\lambda_{i_{z_i}}$)의 무기억 속성(Memoryless property)으로 인해 작동 중인 k_i 개의 부품은 고장시간 분포가 Exp($k_i \lambda_{i_{z_i}}$)인 하나의 부품으로 고려할 수 있다. 즉 k-out-of-n 서브시스템은 신뢰도가 식 (9)와 같은 부품($n_i - k_i + 1$)개가 병렬로 연결된 것으로 고려할 수 있다. 따라서 신뢰도 확률모형은 식 (10)과 같이 도출된다.

$$r_{k_i, i_{z_i}}(t) = \exp(-k_i \lambda_{i_{z_i}} t) \quad (9)$$

$$\widetilde{R}_{i_{z_i}}(t) = r_{k_i, i_{z_i}}(t) + \rho_i(t) \sum_{x=1}^{n_i-k_i} \int_0^t r_{k_i, i_{z_i}}(t-u) f_{k_i, i_{z_i}}^{(x)}(u) du \quad (10)$$

$(0, t]$ 동안 고장이 발생한 부품의 수를 $N_i(t)$ 로 정의하면, $N_i(t) < n_i$ 인 경우, $N_i(t)$ 는 Exp($k_i \lambda_{i_{z_i}}$) 도착간격 시간을 갖는 재생과정으로 Poisson($k_i \lambda_{i_{z_i}}$) 과정이 되어 식 (11)이 도출되며, 결과적으로 부품의 고장시간 분포가 Exp($\lambda_{i_{z_i}}$)를 따르는 대기중복 k-out-of-n 서브시스템의 신뢰도 확률모형은 식 (12)와 같이 도출된다.

$$\int_0^t r_{k_i, i_{z_i}}(t-u) f_{k_i, i_{z_i}}^{(x)}(u) du = \exp(-k_i \lambda_{i_{z_i}} t) \frac{(k_i \lambda_{i_{z_i}} t)^x}{x!} \quad (11)$$

$$\begin{aligned} \widetilde{R}_{i_{z_i}}(t) = & [1 - \rho_i(t)] \exp(-k_i \lambda_{i_{z_i}} t) \\ & + \rho_i(t) \left[\exp(-k_i \lambda_{i_{z_i}} t) \sum_{x=0}^{n_i-k_i} \frac{(k_i \lambda_{i_{z_i}} t)^x}{x!} \right] \end{aligned} \quad (12)$$

2.3 최적 중복전략의 선택

최적 중복전략은 선택된 대안 부품(z_i)과 부품수(n_i)에서 서브시스템 i 의 신뢰도를 최대화하는 중복의 형태를 의미한다. 만약 고장감지기의 신뢰도가 1.0(Perfect S/W)이라면, 항상 대기(Cold-Standby) 중복이 최적 중복전략이 된다. 하지만 고장감지기의 신뢰도가 1.0미만(Imperfect S/W)인 경우는 <Figure 2>와 같이 특정 중복수(Number of Redundancies) 이내에서는 대기중복이, 그 이상에서는 활성중복이 최적 중복전략으로 추천된다. 따라서 대기중복이 최적 중복전략이 되는 최대 부품수를 임계 부품수($n_{i_{z_i}}^c$)로 정의하고, 식 (13)과 같은 대기중복시의 신뢰도 상한과 활성중복시의 신뢰도 하한을 Hoeffding(1963)의 부등식을 이용해 식 (14)와 같이 적용하여 식 (15)와 같은 임계 부품수의 상한을 도출하였다.

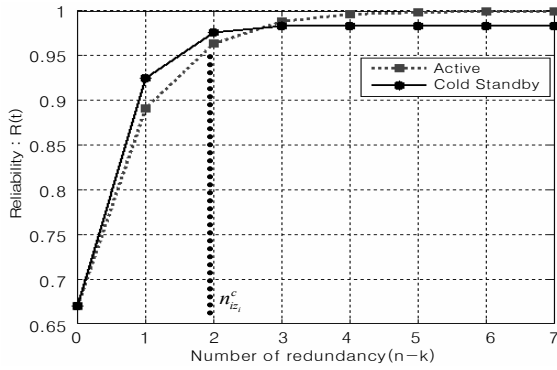


Figure 2. Reliability Variation According to Number of Redundancy Levels

$$\overline{R_{i,z_i,c}}(t) = [1 - \rho_i(t)] r_{k_i,z_i}(t) + \rho_i(t) \quad (13)$$

$$\overline{R_{i,z_i,a}} = 1 - \frac{1}{2} \exp \left[-2 \frac{(n_i r_{i,z_i}(t) - k_i + 1)^2}{n_i} \right] \quad (14)$$

$$\overline{n_{i,z_i}^c} = \left\lceil \left[\frac{\beta_{i,z_i} + \sqrt{\beta_{i,z_i}^2 + 4(k_i - 1)}}{2r_{i,z_i}} \right]^2 \right\rceil \quad (15)$$

$$\text{여기서, } \beta_{i,z_i} = \sqrt{-\frac{1}{2} \ln [2(1 - \overline{R_{i,z_i,c}})]}$$

수치실험 결과, 서브시스템의 최소 부품수 k_i 와 임계 부품수 상한의 평균 인근에서 실제 임계부품수를 관찰할 수 있었다. 따라서 초기 탐색구간을 $[k_i, \overline{n_{i,z_i}^c}]$ 로 설정하고, Bisection Method를 통해 실제 임계 부품수를 탐색하였으며, 임계 부품수 이내에서는 대기중복을 그 이상에서는 활성중복을 최적 중복전략으로 선택하였다.

2.4 수리계획 모형

수리계획 모형은 Misra and Shorma(1973)가 제안한 방법에 따라 비선형의 시스템 신뢰도 확률모형을 대수변환(Logarithm Transform)하고, 새로운 결정변수(y_{ijp})를 도입하여 이진정수계획(Binary Integer Programming) 모형으로 변환하였다.

$$y_{ijp} = \begin{cases} 1, & \text{if } n_i = p \text{ and } z_i = j \text{ for subsystem } i \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Max. } \ln \widetilde{R}_s = \sum_{i=1}^s \ln \widetilde{R}_i = \sum_{i=1}^s \sum_{j=1}^{q_i} \sum_{p=k_i}^{n_{\max,i}} \gamma_{ijp} y_{ijp} \quad (16)$$

Subject to

$$\sum_{i=1}^s \sum_{j=1}^{q_i} \sum_{p=k_i}^{n_{\max,i}} \alpha_{ijp} y_{ijp} \leq C \quad (17)$$

$$\sum_{i=1}^s \sum_{j=1}^{q_i} \sum_{p=k_i}^{n_{\max,i}} \beta_{ijp} y_{ijp} \leq W \quad (18)$$

$$\sum_{j=1}^{q_i} \sum_{p=k_i}^{n_{\max,i}} y_{ijp} = 1, \forall i \quad (19)$$

$$1 \leq z_i \leq q_i, z_i \in \text{Integer} \quad (20)$$

$$k_i \leq n_i \leq n_{\max,i}, n_i \in \text{Integer} \quad (21)$$

$$y_{ijp} \in \{0, 1\} \quad (22)$$

$$\text{For } p = k_i, i \in N \text{ and} \quad (23)$$

$$\cdot \gamma_{ijp} = \ln [Eq. (1)] \text{ or } \gamma_{ijp} = \ln [Eq. (9)]$$

$$\cdot \alpha_{ijp} = p c_{ij}, \beta_{ijp} = p w_{ij}$$

$$\text{For } k_i < p \leq n_{i,z_i}^c, i \in S \text{ and} \quad (24)$$

$$\cdot \gamma_{ijp} = \ln [Eq. (6)] \text{ or } \gamma_{ijp} = \ln [Eq. (12)]$$

$$\cdot \alpha_{ijp} = p c_{ij} + c_{s,i}, \beta_{ijp} = p w_{ij} + w_{s,i}$$

$$\text{For } n_{i,z_i}^c < p \leq n_{\max,i}, i \in A \text{ and} \quad (25)$$

$$\cdot \gamma_{ijp} = \ln [Eq. (2)] \text{ or } \gamma_{ijp} = \ln [Eq. (8)]$$

$$\cdot \alpha_{ijp} = p c_{ij}, \beta_{ijp} = p w_{ij}$$

수리모형의 목적함수 식 (16)은 대수변환된 시스템의 신뢰도를 최대화하는 것이며, 제약함수 식 (17)과 식 (18)은 시스템 구성을 위한 자원의 제약(Constraint)을 의미하는 것으로 시스템을 구성하는 부품들의 비용과 무게의 합이 가용한 자원을 초과할 수 없음을 의미한다. 식 (19)에서는 결정변수 y_{ijp} 가 선택된 부품과 부품수에서는 1이 되고, 이외의 모든 경우는 0이 되어 각 서브시스템에 선택할 수 있는 부품과 부품수의 조합이 유일함을 의미하고 있다. 식 (20)과 식 (21)은 각 서브시스템에 선택할 수 있는 부품 종류와 부품수의 범위를 지정하고 있으며, 식 (22)는 결정변수가 0-1(Binary) 변수임을 나타내고 있다.

식 (23)부터 식 (25)는 결정된 부품수에 따른 최적 중복전략에서의 서브시스템의 신뢰도와 소요되는 자원(비용, 무게)을 계산한다. 식 (23)은 중복이 없는 경우, 식 (24)는 대기중복이 선택된 경우로 고장감지기의 비용과 무게가 추가적으로 고려된다. 식 (25)는 활성중복이 선택되는 경우이다.

3. 병렬 개체군집최적화(PPSO) 알고리즘

Particle Swarm Optimization(PSO)는 Kennedy and Eberhart(1995)에 의해 제안된 진화형 계산 기법으로, 사회적 교호작용(Social interaction)에 의한 생체군집의 행동양식 시뮬레이션 모형에 대한 연구에서 출발하여 새나 물고기의 무리가 먹이를 찾아가는 과정에 착안한 확률적 최적화 기법이며, 모 집단 기반의 메타 휴리스틱(Population based Meta-heuristic) 기법이다. PSO의 기본 개념은 무리(Swarm)가 먹이를 찾아가는 과정에서 무리 전체가 정보를 공유한다는 가설과 무리 내부의 각 개체(Particle)가 자신의 경험이 무리 전체에 공유되어 있는 정보를 기초로 행동한다는 것이다. 즉 각 개체는 다차원 공간(Multi-dimensional space) 공간에서, 과거에 경험했던 자신의 최적 위치정보와 이웃의 경험 정보를 이용하여 자신의 위치정보를 갱신하면서 해

를 탐색하게 된다. 일반적으로 PSO는 연산의 효율성을 바탕으로 짧은 계산시간에 양질의 해를 제공하는 것으로 알려져 있다. 또한 파라미터의 조정을 통해 전역탐색(Global search)과 지역탐색(Local search)의 균형을 유지할 수 있으며, 다른 확률적 알고리즘들에 비해 안정적인 수렴 특징을 갖는다.

본 연구에서는 복수의 무리들이 광범위한 공간에 분산되어 독립적으로 해를 탐색하면서 발견한 양질의 위치정보를 무리간에 공유함으로써 협력하는 시스템을 묘사하여 <Figure 3>과 같은 PPSO(Parallel PSO) 알고리즘을 제안하였다. 이는 전역 최적해 탐색 확률을 증대하고, 임의 무리가 국부 최적해로 조기 수렴(Premature convergence)하는 것을 방지하고자 하였다.

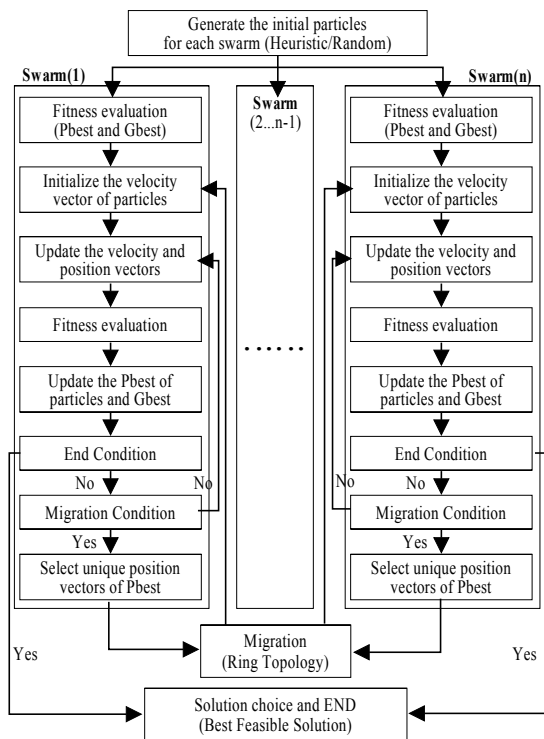


Figure 3. Diagram of PPSO Algorithm

3.1 개체의 위치벡터 표현

PPSO 알고리즘의 첫 번째 단계는 최적화 문제의 잠재해(Potential solution)를 개체의 위치벡터로 표현하는 것이다. 본 연구에서는 개체의 위치벡터를 각 서브시스템에 선택되는 대안 부품의 번호와 부품수를 정수 문자열(Integer string)의 형태로 <Figure 4>와 같이 표현하였으며, 각 서브시스템별 부품 번호(z_i)의 범위는 1부터 대안 부품수(q_i)까지, 부품수의 범위는 최

Sub'(i)	Component					# of Components				
	1	2	...	s-1	s	1	2	...	s-1	s
\vec{x}_i	$1 \leq z_i \leq q_i$					$k_i \leq n_i \leq n_{\max,i}$				

Figure 4. Particle Representation

소 작동 부품수(k_i)부터 최대 부품수($n_{\max,i}$)까지이다. 최적 중복전략은 제 2.3절의 내용과 같이 고장감지기와 선택된 부품의 신뢰도, 그리고 결정된 부품수에 대한 종속변수로 고려하였다.

따라서 위치벡터의 길이는 $2 \times$ 서브시스템 개수(s)가 되며, 위치벡터의 i 열과 $s+i$ 열은 각각 서브시스템 i 의 선택된 부품의 번호와 부품수에 대한 정보를 담고 있다.

3.2 초기 모집단 생성

시스템의 신뢰도는 가용한 자원(비용, 무게)을 충분히 활용하여 고신뢰도의 부품을 선택하고 보다 많은 부품수로 중복 설계할 때 향상될 수 있으므로, 최적해는 자원 제약의 경계지역에 존재할 확률이 높다. 따라서 초기 개체들을 실행가능영역(Feasible region) 내에서 생성하도록 하는 Heuristic 기법과 주로 자원의 제약을 초과하여 생성되는 임의생성 기법을 혼용하였다. 이는 <Figure 5>와 같은 개념에서 개체들이 실행가능영역의 내·외에 분포되어 개체들 간의 교호작용을 통해 자원 제약의 경계영역을 충분히 탐색하도록 하였으며, 개체의 비율은 Heuristic 기법 70%, 임의생성 기법 30%를 적용하였다.

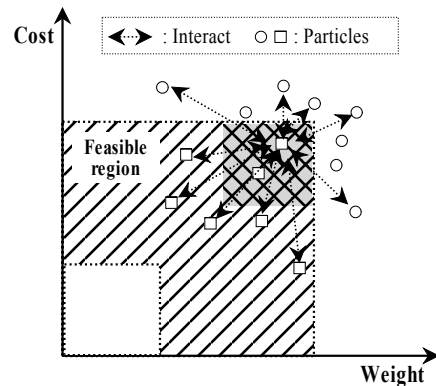


Figure 5. Concept of Searching Solution

(1) Heuristic 기법

Heuristic 기법은 가용한 자원을 각 서브시스템에 할당하고 서브시스템을 활성중복으로 고려했을 때, 할당된 자원으로 서브시스템의 신뢰도를 최대화하는 부품과 부품수의 조합을 초기 개체로 반영하였다.

첫 번째 단계는, Kim *et al.*(2004)이 제시한 부품의 상대적 기여도를 식 (26)과 같이 계산하고, 식 (27)을 통해 정규화하여 Roulette Wheel에서 해당 부품이 선택될 확률로 고려하였다. 즉 상대적으로 적은 자원이 소요되면서 신뢰도가 높은 부품이 선택될 확률이 높아진다. 이후 [0, 1] 구간의 Uniform 난수를 통해 각 서브시스템에 사용할 부품을 선택하고, 최소 부품수 배치에 소요되는 자원을 식 (28)과 같이 할당하였다.

$$d_{iz_i} = \frac{r_{iz_i}}{(c_{iz_i}/C) + (w_{iz_i}/W)} \quad (26)$$

$$p_{iz_i} = d_{iz_i} / \sum_{z_i=1}^{q_i} d_{iz_i} \quad (27)$$

$$c_{a,i}^1 = c_{iz_i} \times k_i, w_{a,i}^1 = w_{iz_i} \times k_i \quad (28)$$

두 번째 단계는, 첫 번째 단계에서 할당 후 자원의 잔여량을 각 서브시스템에 임의 할당하였다. 할당 방법은 [1, s]에서 정수형 Uniform 난수로 서브시스템을 선택하고, 비용의 잔여량을 c_r , 무게의 잔여량을 w_r 이라 할 때, 각각 [1, c_r], [1, w_r]에서 정수형 Uniform 난수로 할당할 자원량($c_{a,i}^2, w_{a,i}^2$)을 결정하였다. 이러한 방법으로 모든 자원의 잔여량이 소진될 때까지 반복하였다.

세 번째 단계는, 각 서브시스템의 구조를 활성중복 k-out-of-n으로 고려했을 때, 할당된 자원($c_{a,i}, w_{a,i}$)으로 서브시스템의 신뢰도를 최대화하는 부품과 부품수의 조합을 선택하였다. 세 번째 단계에 대한 의사코드(Pseudo code)는 <Figure 6>과 같다.

```

1 :  $c_{a,i} = \sum_{h=1}^2 c_{a,i}^h$  and  $w_{a,i} = \sum_{h=1}^2 w_{a,i}^h$ 
2 : For  $i = 1$  to  $i = s$ 
3 : For  $z_i = 1$  to  $z_i = q_i$ 
4 : Decide the number of redundancy
   ;  $n_{iz_i} = Mn(\lfloor c_{a,i}/c_{iz_i} \rfloor, \lfloor w_{a,i}/w_{iz_i} \rfloor)$ 
5 : Calculate subsystem reliability(i) by Eq.(8)
6 : End
7 : End
8 : Determine the component( $z_i$ ) for subsystem(i)
   ;  $z_i = \text{component index of } Max(R_{iz_i})$ 
    
```

Figure 6. Procedure for 3th Stage of Heuristic

(2) 임의생성 기법

임의생성 기법은 각 서브시스템에 사용할 부품 번호와 부품수를 각각 [1, q_i]와 [1, $n_{max,i}$]에서 정수형 Uniform 난수로 생성하였다. 임의생성 기법으로 생성된 개체들은 주로 자원의 제약을 초과하여 적합도(Fitness)는 좋지 않으나 Heuristic 기법으로 생성된 모집단에 비해 다양성이 우수하게 나타난다. 따라서 임의생성 기법은 Heuristic 기법으로 생성된 모집단의 부족한 다양성을 보완하고, 실행가능영역의 개체들과 교호작용을 통해 최적해가 존재할 확률이 높은 자원 제약의 경계영역으로 접근하면서 보다 광범위하게 해를 탐색하도록 하였다.

3.3 적합도 평가

적합도 평가는 개체가 갖고 있는 대안 부품의 번호와 부품수에 대한 정보를 통해 중복전략을 결정하고, 계산된 시스템 신뢰도와 자원 제약의 충족 여부에 따라 벌금(Penalty)을 부여하여 해의 품질을 평가하는 단계이다.

가용한 자원을 충분히 활용할수록 시스템의 신뢰도가 향상

될 수 있으므로, 자원의 경계영역에 최적해가 존재할 확률이 높다. 따라서 자원의 제약을 초과하더라도 제약의 인근에 위치한 개체들은 최적해에 근사한 정보를 많이 포함하고 있을 가능성이 높다. 따라서 본 연구에서는 자원 제약의 초과에 대한 허용치(NFT : Near Feasible Threshold)를 부여하여 유용한 정보를 포함한 개체들이 배제되는 것을 방지하고, 식 (29)와 같이 계산이 진행될수록 허용치를 감소시켜 제약을 엄격하게 적용하는 Coit and Smith(1996)가 제안한 적용형 벌금함수를 식 (30)과 같이 적용하였다.

<Notation>

- $R_{p,i}(\vec{x}_i)$: 개체 i 의 벌금함수 적용 후 적합도
- $R(\vec{x}_i)$: 개체 i 의 벌금함수 적용 전 신뢰도
- R_{all} : 무리 전체 $R(\vec{x})$ 중 최대 신뢰도
- R_{feas} : 실행가능 개체 $R(\vec{x})$ 중 최대 신뢰도
- $NFT_{t,j}$: t -iteration에서 자원 j 의 초과 허용치
- $NFT_{o,j}$: $t=0$ 에서 자원 j 의 초과 허용치
- θ_j : 자원 j 에 대한 파라미터($j = c, w$)
- ψ : 감쇄 상수
- $\Delta c_i, \Delta w_i$: 개체 i 의 비용/무게 제약 초과량

$$NFT_{t,j} = \frac{NFT_{o,j}}{1 + \psi t}, j = c, w \quad (29)$$

$$R_{p,i}(\vec{x}_i) = R(\vec{x}_i) - (R_{all} - R_{feas}) \left[\left(\frac{\Delta w_i}{NFT_w} \right)^{\theta_w} + \left(\frac{\Delta c_i}{NFT_c} \right)^{\theta_c} \right] \quad (30)$$

3.4 속도/위치벡터 갱신

각 개체는 탐색공간(Search space)를 날아다니며 자신이 경험했던 최적 위치정보(Pbest, \vec{p}_i)를 기억하고, 무리 전체의 최적 위치정보(Gbest, \vec{p}_g)를 공유한다. 따라서 각 개체들은 Pbest와 Gbest의 위치정보를 이용하여 자신의 속도와 위치를 변경하고, 탐색과정에서 지속적으로 Pbest와 Gbest의 정보를 갱신하면서 해를 탐색하게 된다.

PSO에서 해 탐색의 핵심인 개체의 운동방정식은 식 (31)과 같은 Shi and Eberhart(1998)의 모형을 사용하였으며, Inertia weight(w)는 [0.5, 1]에서의 Uniform 난수를 적용하였다. RROP는 정수계획 문제로 이산적인(Discrete) 해 공간을 갖게 되므로, 식 (32)와 같이 위치벡터를 반올림하여 정수화하였다. <Figure 7>은 개체의 속도벡터와 위치벡터의 갱신 과정을 보여준다.

$$\vec{v}_i(t+1) = \omega \vec{v}_i(t) + \left[\begin{aligned} & \vec{U}(0, \phi_1) \otimes (\vec{p}_i - \vec{x}_i(t)) \\ & + \vec{U}(0, \phi_2) \otimes (\vec{p}_g - \vec{x}_i(t)) \end{aligned} \right] \quad (31)$$

$$\vec{x}_i(t+1) = \text{round} \{ \vec{x}_i(t) + \vec{v}_i(t+1) \} \quad (32)$$

PSO에서 개체가 급격하게 발산(Explosion)하는 현상이 발생

할 수 있다는 것이 일반적으로 알려져 있다. 따라서 발산현상을 방지하기 위해 각 차원의 속도에 대한 상·하한($\pm V_{max}$)을 설정하였으며, 계산이 진행될수록 식 (33)과 같이 V_{max} 를 감소시켜 이동 속도를 통제함으로써 국부 영역에서는 해를 세밀하게 탐색하도록 하였다. 또한 위치벡터는 각 서브시스템의 대안 부품수(q_i)와 최대 중복수($n_{max,i}$) 제약을 초과하지 못하도록 제한하였다.

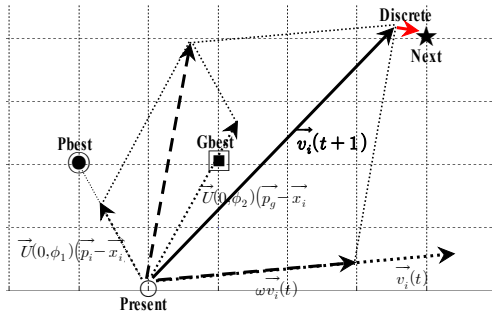


Figure 7. Updating Position and Velocity Vector

$$V_{max}(t+1) = (V_{max}(t) - 0.5) \exp\left[-\frac{0.1t}{T} V_{max}(t)\right] + 0.5 \quad (33)$$

3.5 중복 개체의 제거

PSO에서는 계산이 진행될수록 <Figure 8>과 같이 개체들이 무리의 최선해(Gbest)로 수렴하는 현상이 나타난다. 따라서 이러한 현상을 고려하여 이주(Migration)를 수행하기 전에 각 무리에서 위치벡터가 동일한 중복 개체들을 제거하여 무리가 갖고 있는 다양성은 유지하면서 무리의 개체 수를 감소시켜 계산시간을 단축하였다.

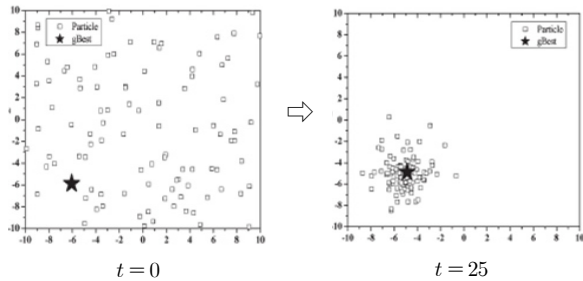


Figure 8. Convergence Characteristic of Particles

3.6 이주(Migration)

이주는 이웃하는 무리 간에 주기적으로 적합도가 우수한 개체들을 교환하여 양질의 정보를 공유하는 과정이다. 무리들이 분산된 공간에서 독립적으로 해를 탐색한 후 정보를 교환함으로써 최적해 탐색 확률을 높이고, 임의의 무리가 국부 최적해로 조기 수렴하는 문제를 방지하고자 하였다. 이주 구조는 <Figure

9>와 같은 원형 이웃구조(Ring Topology)를 적용하였으며, 이 주시킬 개체는 Pbest의 적합도가 높은 $round\{이주율 \times 개체수\}$ 까지 선별하였다.

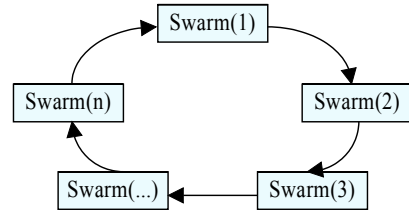


Figure 9. Ring Topology

4. 수치실험 및 분석

신뢰도 확률모형의 반복계산을 피하기 위해 각 대안 부품의 부품수별 최적 중복전략에서 서브시스템의 신뢰도를 계산하고, <Figure 10>과 같은 3차원 행렬로 구성하여 배낭문제(Knapsack Problem)로 고려하였다. 실험에 적용한 PPSO와 벌금함수의 파라미터는 <Table 1>과 같다.

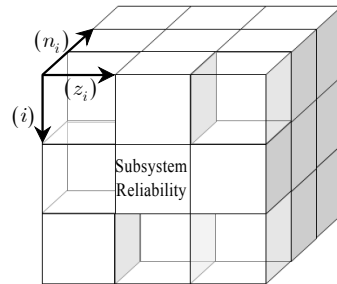


Figure 10. Subsystem Reliability Matrix

Table 1. Parameters for PPSO

PSO		Parallel		Fitness	
Para.	Value	Para.	Value	Para.	Value
Pop_Size	200	Swarm	5	ψ	1
Iteration	50	Mig_Freq	5	θ_c	2
ϕ_1/ϕ_2	2/2	Mig_Frac	0.3	θ_w	1

4.1 직렬-병렬 시스템(예제 1)

본 예제는 Coit(2003)이 제안한 문제로 시스템은 14개의 서브시스템으로 구성되어 있으며, 각 서브시스템의 대안 부품의 종류는 3~4개이고, 최대 부품수는 6개이다. 따라서 해 탐색공간은 3.45×10^{22} 개를 초과한다. 자원의 제약은 비용을 130으로 고정하고, 무게는 159부터 191까지 1씩 증가시켜 실험하였다. 또한 고장감지기의 신뢰도는 0.99를 적용하였다. 실험결과 는 <Table 2>와 같다.

Table 2. The Experimental Results for Example 1

No.	W	GA	MA	PPSO (This Study)	
				Best	%MPI
1	159	0.9641	<u>0.9691</u>	0.983799	47.57
2	160	0.9629	<u>0.9688</u>	0.984059	48.91
3	161	0.9636	<u>0.9663</u>	0.984517	54.06
4	162	0.9564	<u>0.9643</u>	0.984777	57.36
5	163	0.9675	<u>0.9698</u>	0.985407	51.68
6	164	0.9619	<u>0.9678</u>	0.985542	55.10
7	165	0.9454	<u>0.9632</u>	0.986126	62.30
8	166	0.9647	<u>0.9670</u>	0.986261	58.37
9	167	0.9614	<u>0.9639</u>	0.986732	63.25
10	168	0.9669	<u>0.9687</u>	0.986914	58.19
11	169	<u>0.9602</u>	0.9591	0.987185	67.80
12	170	0.9705	<u>0.9719</u>	0.987520	55.59
13	171	0.9639	<u>0.9647</u>	0.987791	65.41
14	172	0.9608	<u>0.9685</u>	0.987974	61.82
15	173	0.9717	<u>0.9767</u>	0.988258	49.60
16	174	<u>0.9707</u>	0.9676	0.988580	61.02
17	175	0.9681	<u>0.9700</u>	0.988712	62.37
18	176	0.9703	<u>0.9708</u>	0.989047	62.49
19	177	0.9738	<u>0.9758</u>	0.989319	55.86
20	178	<u>0.9734</u>	0.9661	0.989501	60.53
21	179	0.9756	<u>0.9839</u>	0.989687	35.94
22	180	0.9834	<u>0.9839</u>	0.990109	38.57
23	181	0.9741	<u>0.9831</u>	0.990109	41.47
24	182	0.9797	<u>0.9841</u>	0.990477	40.11
25	183	0.9737	<u>0.9823</u>	0.990602	46.91
26	184	0.9804	<u>0.9866</u>	0.990750	30.97
27	185	<u>0.9844</u>	0.9841	0.990999	42.30
28	186	0.9821	<u>0.9839</u>	0.991185	45.25
29	187	0.9815	<u>0.9853</u>	0.991243	40.43
30	188	0.9771	<u>0.9874</u>	0.991493	32.48
31	189	<u>0.9861</u>	0.9847	0.991605	39.60
32	190	0.9863	<u>0.9870</u>	0.991768	36.68
33	191	0.9856	<u>0.9865</u>	0.991989	40.66
Avg. CPU Time			526 Sec.	1.24 Sec.	

Note) Best of existing studies/Superior solution.

$$\%MPI = \{(Best-Existing_Best)/(1-Existing_Best)\} \times 100.$$

실험결과, 33종의 모든 문제에서 Tavakkoli-Moghaddam *et al.*(2008)의 GA(Genetic Algorithm)와 Safari *et al.*(2010)의 MA(Memetic Algorithm) 보다 우수한 해를 탐색하였으며, 계산시간 또한 대폭 단축되었다. 이는 중복전략을 종속변수로 고려하여 위치벡터의 길이가 기존 연구 대비 2/3로 감소됨에 따라 탐색공간이 (1/2)¹⁴배 축소되고, 계산량을 감소시킨 효과이며, 특히 배낭문제로 고려함으로써 신뢰도 확률모형의 반복 계산을 제거한 것이 계산시간 단축에 주요했던 것으로 분석된다.

4.2 직렬-k-out-of-n 시스템(예제 2)

본 예제는 본 연구에서 제안하는 문제이며, 실험 데이터는 Coit and Liu(2000)의 데이터를 활용하였다. 시스템은 14개의 k

-out-of-n 서브시스템으로 구성되어 있으며, 서브시스템별 대안 부품수는 3~4개, 최소 부품수(k)는 1~3개이며, 최대 부품수는 모두 6개이다. 따라서 해 탐색공간은 2.739×10^{21} 을 초과한다. 고장감지기의 신뢰도는 0.99를 적용하였다. 실험결과는 <Table 3>과 같다.

Table 3. The Experimental Results for Example 2

No.	W	PPSO Algorithm (20 runs)				
		Max.			Min. Reliability	S.D.
		C	W	Reliability		
1	159	110	159	0.330194	0.311610	0.0093
2	160	111	160	0.337370	0.323358	0.0068
3	161	112	161	0.353645	0.336864	0.0045
4	162	111	162	0.357521	0.331972	0.0077
5	163	113	163	0.386660	0.335544	0.0101
6	164	113	164	0.390898	0.368897	0.0098
7	165	114	165	0.399393	0.377054	0.0079
8	166	115	166	0.414451	0.385249	0.0089
9	167	115	167	0.418993	0.363094	0.0171
10	168	116	168	0.428099	0.423173	0.0020
11	169	118	169	0.445572	0.414177	0.0073
12	170	118	170	0.450455	0.437021	0.0058
13	171	119	171	0.460245	0.429676	0.0072
14	172	120	172	0.477597	0.465196	0.0051
15	173	120	173	0.482832	0.470153	0.0064
16	174	121	174	0.493325	0.493325	-
17	175	123	175	0.498246	0.488792	0.0021
18	176	124	176	0.517031	0.503606	0.0063
19	177	124	177	0.522697	0.508806	0.0047
20	178	125	178	0.534057	0.534057	-
21	179	125	178	0.535471	0.535471	-
22	180	127	180	0.549099	0.544954	0.0017
23	181	128	181	0.558055	0.553104	0.0024
24	182	128	182	0.567182	0.567182	-
25	183	128	183	0.579682	0.576432	0.0007
26	184	128	184	0.586035	0.585579	0.0002
27	185	129	185	0.598771	0.588194	0.0054
28	186	124	186	0.602073	0.597528	0.0011
29	187	125	187	0.611893	0.597528	0.0062
30	188	125	188	0.621900	0.621900	-
31	189	126	189	0.632043	0.632043	-
32	190	125	190	0.634590	0.634590	-
33	191	128	191	0.649845	0.644940	0.0015
Avg. CPU Time		0.81 Sec. (S.D : 0.132)				

4.3 혼합 중복전략의 효과 분석

동일한 자원(비용, 무게)으로 활성중복만을 고려한 경우와 혼합 중복전략을 고려하여 최적화된 시스템의 신뢰도를 비교하여 효과를 분석하였으며, 신뢰도의 향상 효과는 {(Mixed-Active)/Active}×100(%)로 계산하였다.

직렬-병렬 시스템은 활성중복만을 고려하여 PPSO로 추가 실험한 결과와 <Table 2>를 비교한 결과는 <Figure 11>과 같

다. 시스템 신뢰도는 평균 1.5% 향상되었으며, 무게의 제약이 적어질수록 많이 향상되어 무게 제약이 159일 때 최대 3.1% 향상되는 것으로 나타났다.

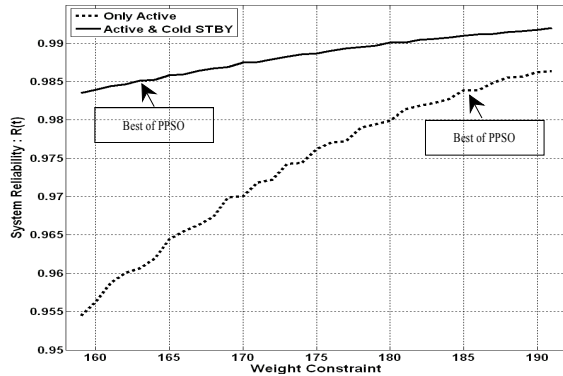


Figure 11. Improvements in Series-Parallel System

직렬- k -out-of- n 시스템은 Bae(2008)가 CPLEX를 통해 제시한 활성중복에서의 최적해와 <Table 3>의 PPSO 최선해를 비교한 결과는 <Figure 12>와 같다. 시스템 신뢰도는 평균 9.3%, 무게 제약이 185일 때 최대 10.3% 향상되었다.

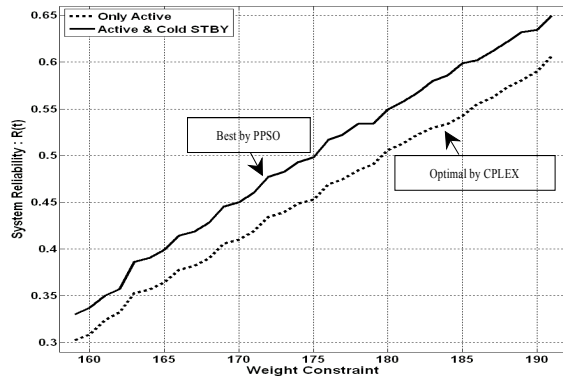


Figure 12. Improvements in Series- k -out-of- n System

결론적으로, 혼합 중복전략을 고려하는 경우 비용, 무게의 자원 제약으로 설계할 수 있는 중복수에 제한이 따를 때 신뢰도 향상 효과가 높으며, 적은 자원으로도 목표하는 신뢰도를 충족시킬 수 있게 된다. 따라서 이는 적정 신뢰도를 보장하면서 보다 소형화/경량화된 산업 시스템 설계에 유용하게 적용될 수 있다.

5. 결론 및 향후 연구방향

현재까지 주로 연구된 RROP는 중복전략으로 활성중복만을 고려함으로써 실제 산업현장의 시스템 설계에 적용이 제한적이었다. 따라서 본 연구에서는 현실 시스템에 주로 적용되고 있는 활성중복과 대기중복을 대상으로 최적의 중복전략을 선택

하는 것을 추가적으로 고려하는 문제를 다루었으며, 서브시스템의 구조를 Coit(2003)보다 일반화된 구조(k -out-of- n)로 확장함으로써 보다 다양한 현실 시스템에 적용이 가능하도록 하였다. 또한 활성중복만을 고려했을 때 보다 혼합 중복전략을 고려했을 때 시스템의 신뢰도를 향상시킬 수 있음을 확인하였으며, 이는 최근의 소형화/경량화 추세에 있는 제품들의 설계에 유용하게 적용될 수 있을 것이다.

본 연구에서는 RROP의 해법으로, 복수의 무리들이 광범위한 공간에 분산되어 독립적으로 해를 찾아가면서 발견한 양질의 정보를 무리 간에 공유는 협력 체계를 묘사한 PPSO 알고리즘을 제안하였다. Coit(2003)의 Benchmark 문제에 대한 수치실험 결과, 기존 연구들 보다 짧은 계산시간 내에 우수한 해를 제공함으로써 PPSO 알고리즘의 우수성을 검증하였다.

향후 연구방향으로는 부품의 고장시간 분포를 PH-type 분포 등 다양한 분포로 고려하고, 분석적 해석에 제한이 따르는 경우 확률모형의 수치해법(Numerical analysis)과 최적화 알고리즘(Meta-Heuristic 등)을 연계하여 산업현장의 다양한 현실 문제들을 해결할 수 있는 연구가 필요할 것으로 판단된다.

참고문헌

Bae, C.- O. (2008), A Simulated Annealing Algorithm for Reliability Optimization Problems with Component Choices in Series Systems, *Dong-Eui University (Thesis for the Degree of Ph. D. in Engineering)*.

Chern, M.-S. (1992), On the Complexity of reliability redundancy allocation in a series system, *Operations Research Letters*, 11(5), 309-315.

Coit, D. W. (2003), Maximization of system reliability with a choice of redundancy strategies, *IEEE Transactions*, 35, 535-543.

Coit, D. W. and Liu, J. (2000), System Reliability Optimization with k -out-of- n Subsystems, *International Journal of Reliability, Quality and Safety Engineering*, 7(2).

Coit, D. W. and Smith, E. (1996), Adaptive Penalty Methods for Genetic Optimization of Constrained Combinatorial Problems, *INFORMS Journal on Computing*, 8(2).

Hoeffding, W. (1963), Probability inequalities for sums of bounded random variables, *Journal of the American Statistical Association*, 58(301), 13-30.

Kennedy, J. and Eberhart, R. C. (1995), Particle swarm optimization, *Proceedings of IEEE international Conference on Neural Networks*, 1942-1948.

Kim, J. Y. and Kim, K. S. (2004), Optimum redundancy design for maximum system reliability : A genetic algorithm approach, *Journal of Korean Society for Quality Management*, 32(4), 125-139.

Misra, K. B. and Sharma, U. (1973), Reliability optimization of a system by zero-one programming, *Microelectronics and Reliability*, 12(3), 229-233.

Safari, J. and Tavakkoli-Moghaddam, R. (2010), A redundancy allocation problem with the choice of redundancy strategies by a memetic algorithm, *J. Ind. Eng. Int.*, 6(11).

Shi, Y. and Eberhart, R. C. (1998), A modified particle swarm optimizer, *In Proceedings of the IEEE international conference on evolutionary computation*, 69-73.

Tavakkoli-Moghaddam, R., Safari, J., and Sassani, F. (2008), Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm, *Reliability Engineering and System Safety*, 93(4), 550-556.

**김홍섭**

한국항공대학교 공학사
 국방대학교 운영분석학과 석사
 현재 : 공군 군수사령부 군수관리단
 관심분야 : 신뢰도 공학, 일정계획, 메타
 휴리스틱, 수요예측

**전건욱**

공군사관학교 전자공학과 학사
 고려대학교 산업공학과 석사
 University of Louisville 산업공학박사
 현재 : 국방대학교 운영분석학과 교수
 관심분야 : 최적화기법 응용, 일정계획,
 신뢰도, 셀형 제조시스템