

# 공급사슬 네트워크 설계를 위한 협력적 공진화 알고리즘에서 집단들간 상호작용방식에 관한 연구\*

한 용 호<sup>†</sup>

부산외국어대학교 e-비즈니스학과

## A Study on Interaction Modes among Populations in Cooperative Coevolutionary Algorithm for Supply Chain Network Design

Yongho Han<sup>†</sup>

Department of e-Business, Busan University of Foreign Studies

### ■ Abstract ■

Cooperative coevolutionary algorithm (CCEA) has proven to be a very powerful means of solving optimization problems through problem decomposition. CCEA implies the use of several populations, each population having the aim of finding a partial solution for a component of the considered problem. Populations evolve separately and they interact only when individuals are evaluated. Interactions are made to obtain complete solutions by combining partial solutions, or collaborators, from each of the populations. In this respect, we can think of various interaction modes. The goal of this research is to develop a CCEA for a supply chain network design (SCND) problem and identify which interaction mode gives the best performance for this problem. We present general design principle of CCEA for the SCND problem, which require several co-evolving populations. We classify these populations into two groups and classify the collaborator selection scheme into two types, the random-based one and the best fitness-based one. By combining both two groups of population and two types of collaborator selection schemes, we consider four possible interaction modes. We also consider two modes of updating populations, the sequential mode and the parallel mode. Therefore, by combining both four possible interaction modes and two modes of updating populations, we investigate seven possible solution algorithms. Experiments for each of these solution algorithms are conducted on a few test problems. The results show that the mode of the best fitness-based collaborator applied to both groups of populations combined with the sequential update mode outperforms the other modes for all the test problems.

Keywords : Cooperative Coevolutionary Algorithm, Genetic Algorithm, Supply Chain Network Design, Population, Collaborator, Interaction

논문접수일 : 2014년 07월 04일    논문게재확정일 : 2014년 11월 18일

논문수정일 : 1차(2014년 08월 19일), 2차(2014년 10월 11일)

\* 이 논문은 2014년도 부산외국어대학교 학술연구조성비에 의해 연구되었음.

† 교신저자, yhhan@bufs.ac.kr

## 1. 서 론

전통적인 유전 알고리즘(genetic algorithm : 이하 GA로 칭함)은 최적화 문제에 대한 하나의 해법으로서 결정변수의 수가 늘어남에 따라 탐색체의 길이도 이에 대응하여 늘어나서 그 결과 제한된 시간 내에 생성되는 해의 질이 저하되거나 적절한 해를 구하기까지 시간이 더 많이 소요된다는 문제점을 지니고 있다. 최근 들어, 공진화 알고리즘(coevolutionary algorithm)은 최적화 문제의 근사해를 구할 수 있는 매우 강력한 수단으로 입증되어 왔다. 공진화 알고리즘은 특히 GA가 좋은 성능을 보이지 못하거나 GA가 적용되기 어려운 문제들의 해를 구하는 데 사용될 수 있다. 전통적인 GA는 동질적인 하나의 탐색체 집단만을 사용한다. 여기서 각 탐색체는 문제에 대한 하나의 (전체)해를 표현한다. 반면, 공진화 알고리즘은 여러 개의 탐색체 집단을 사용한다. 각 집단의 탐색체는 하나의 표준적인 GA를 사용하여 전체해의 특정 부분을 표현한다. 각 집단을 대표하는 하나씩의 개체들이 결합되면 이에 대응하는 하나의 전체해가 생성된다. 이들 집단들은 서로 경쟁하거나 또는 협력하면서 진화해 나갈 수 있다.

그 중 협력적 공진화 알고리즘(cooperative co-evolutionary algorithm; 이하 CCEA로 칭함)은 최적화 문제에 대한 근사 해법으로서 Potter and De Jong[14]에 의해 처음으로 소개되었다. 세대 교체시 각 집단은 다른 집단들을 대표하는 개체들과 협력적 상호작용을 통해 진화해 나간다. CCEA에서의 탐색 공간의 분할에 의한 성능 향상은 다양한 함수의 최적화 문제에서 이미 입증된 바 있다[7, 8]. CCEA는 컨테이너 로딩계획[13], job shop 일정계획[17], 재고통제 시스템 설계[10] 등의 최적화 문제에 적용된 바 있다. 공급사슬 네트워크 설계(supply chain network design; 이하 SCND로 칭함) 분야에서 CCEA는 2단계 수송문제[2], 역물류 네트워크 설계[1], 그리고 다단계 공급사슬 네트워크 설계 문제에 부분문제들로의 분할 방법이나 집단내 탐색체의

표현방법 등을 달리하면서 적용된 바 있다[3-5].

특히, 공급사슬 네트워크 분야에서 가장 높은 수준의 복잡도를 지닌 페쇄루프 SCND 문제에 CCEA가 적용된 바 있다[6]. 이 연구에서는 CCEA의 적용을 통해 대상 문제에 대한 근사해를 구하기 위하여 CCEA에 필요한 일반적인 여러 속성들의 설계 내용을 구체적으로 제시하는 데 초점을 맞추었다. CCEA의 GA와의 가장 큰 차이점은 CCEA는 복수개의 탐색체 집단을 취하여 그 집단들 사이의 상호작용을 통해 해를 개선시켜 나간다는 점이다. 여기서 상호작용이란 평가 대상 집단의 각 개체들을 다른 각 집단들을 대표하는 개체(즉, 협력자)를 결합함으로써 대상 문제에 대한 하나의 완전한 해를 얻고 또한 평가대상 개체의 적합도를 구하는 과정을 의미한다. 이런 점에서 CCEA는 GA가 더욱 고도화된 형태로 볼 수 있다. CCEA의 기반에 해당하는 GA를 설계하는 데 필요한 여러 속성들에 대해서는 이미 많은 연구가 수행된 바 있으며, 이 연구결과들은 CCEA의 설계시에도 그대로 적용될 수 있다. 반면 GA에는 없는 반면 CCEA에는 존재하는 가장 큰 특징은 탐색체 집단들간의 상호작용이다. 그러나, SCND 문제에 대하여 집단들간 상호작용에 대한 연구는 아직까지 찾아볼 수 없다.

따라서 본 연구에서는 SCND 문제를 대상으로 하여 한용호[6]에서 설정한 CCEA의 설계속성들을 기반으로 하여 CCEA 안에서 탐색체 집단들간의 다양한 상호작용 방식들이 해의 성능에 미치는 영향을 더욱 심도있게 파악하고자 한다. SCND 문제는 순방향 물류 네트워크 설계 문제와 역방향 물류 네트워크 설계 문제가 결합된 형태로서 이 두 가지 각각의 설계 문제보다 더 많은 수의 결정변수들을 수반하기 때문에 CCEA가 GA보다 더 효과적인 해법이 될 것으로 기대된다. 본 SCND 문제는 그 특성상 여러 개의 탐색체 집단으로 쉽게 분할될 수 있기 때문에 집단간 상호작용의 성과를 파악하고자 하는 연구 목적에 잘 부합된다.

본 연구에서는 먼저, SCND 문제에서 탐색체의 인코딩 및 디코딩 방식의 유사성에 따라 전체 탐색

체 집단들을 크게 두 그룹으로 분류하고, 또한 각 집단별 협력자 선택방식으로 베스트 적합도(fitness) 기반 및 무작위 기반이라는 두 가지 방식을 고려한다. 따라서 두 그룹의 집단과 두 가지의 협력자 선택방식의 조합에 따라 모두 네 가지의 집단간 상호작용 방식을 고려한다. 다음으로, 각 염색체 집단의 세대 교체가 이루어짐에 됨에 따라 그 집단의 협력자도 업데이트되어야 한다. 그 업데이트 시점을 기준으로 병행적 업데이트와 순차적 업데이트의 두 방식을 고려한다. 따라서 집단들간 상호작용 방식과 협력자의 업데이트 방식의 조합에 따라 모두 7가지 방식의 해법을 설정한다. 테스트 문제들에 대한 실험을 통해 각 방식에 의한 해들을 구하고 이 해들을 바탕으로 각 해법의 성능을 비교 및 분석하고 그 결과를 소개한다.

제 2장에서는 CCEA에서 일반적으로 사용되는 설계 속성들을 소개하고 각 속성별로 선택할 수 있는 옵션들을 소개한다. 제 3장에서는 CCEA의 적용 대상이 되는 SCND 문제 및 그 수리모형을 소개한다. 제 4장에서는 SCND 문제에 대하여 CCEA의 각 설계 속성별로 하나의 옵션을 지정한다. 제 5장에서는 집단간 상호작용 방식의 대안들 및 협력자 선택방식의 대안들을 설정하고, 이 두 가지 요소의 조합으로 생성되는 각 해법을 테스트 문제들에 적용하는 실험을 수행한다. 실험에서 얻어진 해를 바탕으로 각 해법의 성능을 비교 및 분석하고 그 결과를 소개한다. 마지막으로 제 6장에서 결론을 맺는다.

## 2. CCEA의 설계 속성들

### 2.1 CCEA 개요

CCEA의 패러다임을 실현하기 위하여 먼저 주어진 문제를 여러 개의 부분문제들로 분해해야 한다. 각 부분문제에 대응하는 부분해를 나타내기 위하여 염색체의 표현방법을 정의하고 그 속에 존재할 개체의 수를 정함으로써 각 집단이 형성된다. 각 집단은 최적화 대상문제의 지정된 부분문제의 최

적화를 담당한다. 매 세대가 끝날 때마다 각 집단은 유전연산을 통해 그 속의 개체들을 새로이 생성한다. 또한 그 집단의 각 개체를 다른 나머지 집단들의 베스트 개체들과 결합시킴으로써 그 개체의 적합도를 구한다. 그 결과 그 집단의 베스트 개체가 구해진다. 알고리즘의 종료조건이 만족되면 각 집단의 베스트 개체들을 결합함으로써 하나의 전체해가 생성된다. 이러한 CCEA의 전체적인 수행 절차를 다음의 유사 코드로 표현할 수 있다[9].

### CCEA의 유사 코드

```

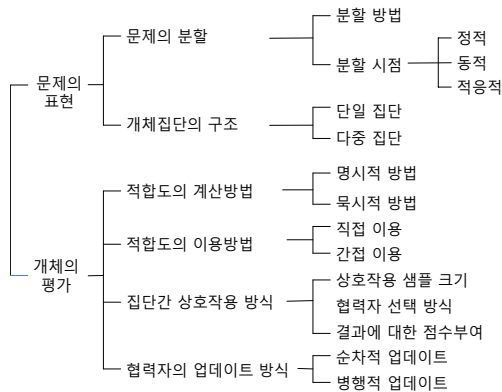
t = 0
for each  $i \in [1, J]$  do //  $i$ 번째 집단에 대하여
     $Pop_i(t)$ 을 초기화시킨다.
     $Pop_i(t)$ 을 평가한다.
end for
migrateBestSols( $P_i^0$ )  $\forall i \in [1, J]$  // 베스트 개체를
협력자로 지정
while 종료조건 = false do
    t = t + 1
    for each  $i \in [1, J]$  do
         $Pop_i(t-1)$ 으로부터 집단  $Pop_i(t)$ 을 선택한다.
         $Pop_i(t)$ 에 유전연산들을 적용한다.
         $Pop_i(t)$ 을 평가한다.
    end for
end while
migrateBestSols( $P_i^0$ )  $\forall i \in [1, J]$ 
solution = assembleBestSols( $P_i^t$ ) // 각 집단의 협
력자들을 결합하여 해를 구한다.

```

### 2.2 CCEA 설계속성의 유형

CCEA는 그 내부에 여러 유형의 설계속성들을 포함한다. 하나의 CCEA가 실행되기 위해서는 각 유형의 설계속성에 할당 가능한 여러 옵션들 중에서 하나가 선택되어야 한다. Wiegand[16]는 협력적 및 경쟁적 공진화 알고리즘을 모두 포함한 공진화 알고리즘에 대하여 그 설계속성들의 분류들을

제시한 바 있다. 이 분류틀을 기반으로 하여 CCEA만의 설계속성 분류틀을 <그림 1>과 같이 수정할 수 있다. 이 그림에서 CCEA의 설계속성은 최상위 수준에서 개체의 ‘표현’ 및 개체의 ‘평가’라는 두 가지 유형으로 분류된다. ‘표현’ 유형은 CCEA가 문제의 해를 구하기 위하여 문제를 어떻게 표현하느냐에 관련된 속성들을 포함하며, ‘평가’ 유형은 개체의 적합도를 어떻게 구하느냐에 관련된 속성들을 포함한다. 두 유형의 설계속성들을 차례로 소개한다.



<그림 1> CCEA 설계속성의 분류틀

### 2.2.1 문제의 표현에 관련된 설계속성들

문제의 표현에 관련된 설계속성은 크게 1) 문제의 분할 및 2) 개체집단의 구조의 두 가지로 구성된다.

1) **문제의 분할** : 이 속성은 다시 분할 방법과 분할 시점이라는 두 개의 속성으로 나누어진다.

- **분할 방법** : 이 속성은 대상 문제를 두 개 이상의 하위 부분문제들로 분할하는 방법을 가리킨다. 분할 방법은 일반적으로 문제의 성격에 따라 고유하게 결정된다. 부분문제들로 분할된 구조는 대상 문제의 자연적 구조와 일치할 수도 있고 그렇지 않을 수도 있다. 즉, ‘분할 방법’에 의한 문제분할과 ‘자연적’ 문제분할사이에 차이가 생길 수도 있다.

- **분할 시점** : 이 속성의 옵션으로는 1) 설계자에 의해 사전에 결정되는 정적(static) 옵션, 2) 설계자에 의해 미리 부여된 어떤 룰에 따라 런타임 중에 조정되는 동적(dynamic) 옵션, 그리고 3) 런타임 중에 시스템에 의해 자체적으로 조정되는 적응적(adaptive) 옵션 등이 존재한다.

2) **개체집단의 구조** : 대상 문제의 분할 방법에 따라 두 개 이상의 하위 부분문제들이 생성된다. 각 부분문제에 대응하여 하나씩의 집단이 생성된다. 부분문제들이 2개, 3개 생성됨에 따라 개체집단들도 2집단 구조, 3집단 구조 등을 취하게 된다.

### 2.2.2 개체의 평가에 관련된 설계속성들

집단내 개체의 평가와 관련된 속성들로는 1) 적합도의 계산방법, 2) 적합도의 이용방법, 3) 집단간 상호작용 방식 및 4) 협력자의 업데이트 방식이 존재한다.

1) **적합도의 계산방법** : 이 속성은 각 개체에 대한 적합도를 계산하는 방식을 가리키는 것으로, 다음과 같이 명시적 방법 또는 암묵적 방법의 두 옵션이 존재한다.

- **명시적 방법** : 주어진 최적화 문제의 목적함수를 기반으로 각 개체의 적합도를 계산할 수 있다. 대부분의 CCEA는 이 명시적 방법을 사용한다. 개체의 적합도는 알고리즘의 실행 전에 명시적으로 정의된다.

- **묵시적 방법** : 다중 집단에서 적합도 할당이 하나의 집단에서는 명시적으로, 그리고 다른 집단에서는 묵시적으로 이루어지는 경우도 있다. 예를 들어, 주어진 문제가 하나의 상위 집단과 여러 개의 하위 집단들로 분할된 경우 하위 집단들내 개체는 자신의 상위 집단 내 개체의 성과에 대한 기여도를 통해 그 적합도를 간접적으로 할당받을 수 있다.

2) **적합도의 이용방법** : 적합도의 값은 기본적으로 해 공간에서 진화 탐색의 방향에 영향을 주는

내부 수단으로 사용된다. 예를 들어, 비례선택 (proportional selection) 연산의 경우 하나의 적합도 값이 산출되면, 이 값을 선택연산에 직접 이용할 수도 있으며, 또는 이 값을 바탕으로 어떤 주관적인 값을 생성한 후 이 값을 확률적인 선택연산의 과정에서 간접으로 이용할 수도 있다.

**3) 집단간 상호작용 방식 :** 하나의 개체가 평가될 때 완전한 해를 형성하기 위하여 모든 다른 집단별로 협력자(collaborator)들이 선택된다. 적합도 평가를 하는 동안 발생하는 집단간 상호작용은 다음의 세 가지 속성을 지닌다.

- 상호작용 샘플 크기 : 적합도 평가에 사용되는, 각 집단별 협력자의 수를 가리킨다. 먼저, 하나의 개체를 평가하기 위하여 그 개체를 각 집단별로 모든 잠재적 협력자와 일일이 상호작용 시키는 방식을 생각해 볼 수 있다. 이 방법은 가장 좋은 결과를 낳는 반면, 가장 복잡하며 많은 계산량을 수반한다. 반대로, 평가대상 개체를 각 집단별로 하나씩의 협력자와 한 번만 상호작용 시키는 방식을 생각해 볼 수 있다. 이 방식은 가장 간단한 방식에 해당한다. 위의 극단적인 두 방식사이에 평가대상 개체를 각 집단의 모든 잠재적 협력자 중 그 일부와 상호작용을 시키는 여러 방법들을 생각해 볼 수 있다.
- 협력자 선택 방식 : 협력자 선택 시 편중(bias)의 정도를 의미한다. 무작위 선택 방식과 적합도 기반의 선택 방식이 존재한다.
- 상호작용 결과에 대한 점수부여 : 상호작용에 의해 생성된 복수 개의 목적함수 값으로부터 하나의 적합도를 할당하는 방식을 가리킨다.

**4) 협력자의 업데이트 방식 :** 다중 집단을 사용함에 따라 관련 용어를 다음과 같이 보다 명확히 정의할 필요가 있다. CCEA에서 하나의 집단에서 부모의 선택, 자손의 생성, 평가 및 생존 선택의 단계들을 거치는데 소요되는 시간을 세대 (generation)라고 정의한다. 그리고 모든 집단에서 위 단계들을 완전히 거치는 데 소요되는 시간을 라운드(round)라고 정의한다. 각 집단으로

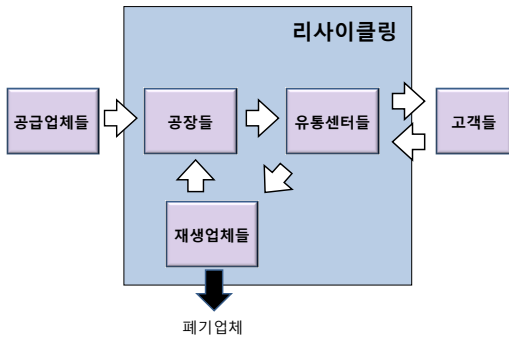
부터 하나의 개체를 선택하여 이것을 그 집단의 협력자로 사용하는 데 있어, 각 집단의 현 상태에서부터 하나의 개체를 선택할 것인지, 아니면 이전 라운드의 집단으로부터 하나의 개체를 선택할 것인지를 정해야 한다. 이것은 각 집단들이 처리되고 업데이트되는 타이밍 상의 문제로서 다음의 두 가지 옵션이 존재한다.

- 순차적 업데이트 : 각 집단은 다른 모든 집단들의 현 상태에서부터 협력자들을 선택한다. 각 집단은 라운드 로빈(round-robin) 방식으로 처리된다(뒤에서 소개될 <그림 5>는 이 방식의 예를 보여준다). 그 결과, 어떤 라운드에서 먼저 처리되고 업데이트된 집단은 나중에 처리되는 집단에게 영향을 미칠 수 있다.
- 병행적 업데이트 : 매 라운드가 끝나는 시점에 각 집단들이 동시에 병행으로 처리된다. 각 집단의 협력자들은 다른 모든 집단들의 이전 세대로부터 선택된다.

### 3. SCND 문제

본 연구에서 다루고자 하는 공급사슬 네트워크의 형태는 <그림 2>와 같이 표현될 수 있다. 문제의 복잡성을 줄이기 위해 이 공급사슬은 한 가지 종류의 제품만을 대상으로 한다. 그리고 완제품은 하나의 부품만을 포함한다고 가정한다. 부품 또는 제품의 순방향 물류는 공급업체로부터 시작하여, 공장, 유통센터를 거쳐 고객에게 이른다. 순방향 물류는 고객의 수요에 의해서 시작되는 풀(pull) 시스템으로 가동된다. 반면, 역방향 물류는 고객의 폐제품이 유통센터(수거센터의 기능을 병행), 재생업체를 거쳐 재생부품으로 재생되어 공장으로 되돌아간다. 역방향 물류는 푸시(push) 시스템으로 가동된다. 순방향 및 역방향 물류의 양은 각각 고객의 수요량 및 폐제품 회수율에 의해 결정된다. 공급업체는 새부품을, 그리고 재생업체는 재생부품을 공장에 공급한다. 새부품과 재생부품은 사용 성능상의 차이가 없는 것으로 가정한다. 각 공장은 순방

향 물류와 역방향 물류가 만나는 지점이 된다. 이로 인해 공급사슬의 폐쇄루프가 형성된다. 각 유통센터는 순방향 물류에 대해서는 유통센터로서의 기능을, 그리고 역방향 물류에 대해서는 수거센터로서의 기능을 수행한다. 따라서 각 유통센터의 저장능력은 이 두 기능에 의해 공유된다. 각 고객들의 폐제품 중 일정한 비율만이 수거센터(유통센터)로 수거된 후, 모두 재생업체로 운송된다. 재생업체는 폐제품을 분류, 검사하여 그 중 일정 비율의 양은 폐기하고, 나머지는 재생작업을 거쳐 공장에 재생부품으로 공급된다.



〈그림 2〉 공급사슬 형태[6]

본 문제는 이러한 환경 하에서 관련 총비용을 최소화시키기 위하여 1) 공장, 유통센터 및 재생업체별 후보군 중에서 실제로 운영될 업체들을 선택하고, 2) 순방향의 각 계층(공급업체-공장, 공장-유통센터, 유통센터-고객) 및 역방향의 각 계층(고객-유통센터, 유통센터-재생업체, 재생센터-공장)에서 각 공급지와 각 수요지사이의 수송량을 결정하는 것이다.

폐쇄루프 SCND 문제에 대한 수리 모형은 다음과 같이 정식화된다.

**파라미터 :**

- $a_i$  : 공급업체  $i$ 의 부품공급능력( $i=1, 2, \dots, I$ )
- $b_j$  : 공장  $j$ 의 제품생산능력( $j=1, 2, \dots, J$ )

- $Sc_k$  : 유통센터  $k$ 의 제품저장능력( $k=1, 2, \dots, K$ )
- $pd_k$  : 유통센터  $k$ 의 제품저장능력 중 역물류에 대한 할당비율
- $pc_l$  : 고객  $l$ 로부터의 폐제품의 수거비용 ( $l=1, 2, \dots, L$ )
- $pl_m$  : 재생업체  $m$ 에서의 폐제품의 폐기비용 ( $m=1, 2, \dots, M$ )
- $d_l$  : 고객  $l$ 의 수요량
- $e_m$  : 재생업체  $m$ 의 폐제품 처리능력
- $s_{ij}$  : 공급업체  $i$ 로부터의 공장  $j$ 에서의 부품단위당 운반비용
- $t_{jk}$  : 공장  $j$ 로부터 유통센터  $k$ 로의 제품단위당 운반비용
- $u_{kl}$  : 유통센터  $k$ 로부터 고객  $l$ 로의 제품단위당 운반비용
- $w_{mj}$  : 재생업체  $m$ 으로부터 제조업체  $j$ 로의 부품 단위당 운반비용
- $Ru_{lk}$  : 고객  $l$ 로부터 유통센터  $k$ 로의 폐제품단위당 수거비용
- $f_j$  : 공장  $j$ 의 운영비용(고정비)
- $g_k$  : 유통센터  $k$ 의 운영비용(고정비)
- $h_m$  : 재생업체  $m$ 의 운영비용(고정비)
- $\psi$  : 폐제품 단위당 처분비용(고정비)
- $P$  : 운영할 수 있는 공장 수의 상한
- $Q$  : 운영할 수 있는 유통센터 수의 상한
- $W$  : 운영할 수 있는 재생업체 수의 상한
- $C_p$  : 공장들간 이용율 편차의 상한

**결정변수 :**

- $x_{ij}$  : 공급업체  $i$ 로부터 공장  $j$ 로의 신부품 수송량
- $y_{jk}$  : 공장  $j$ 로부터 유통센터  $k$ 로의 제품 수송량
- $z_{kl}$  : 유통센터  $k$ 로부터 고객  $l$ 로의 제품 수송량
- $Rz_{lk}$  : 고객  $l$ 로부터 유통센터  $k$ 로의 폐제품 수송량
- $o_{km}$  : 유통센터  $k$ 로부터 재생센터  $m$ 으로의 폐제품 수송량
- $Rd_{mj}$  : 재생업체  $m$ 으로부터 공장  $j$ 로의 재생부품 수송량

$\alpha_j$  : 0-1 변수(공장  $j$ 의 가동시 1의 값을, 그렇지 않은 경우 0의 값을 취함)

$\beta_k$  : 0-1 변수(유통센터  $k$ 의 개설시 1의 값을, 그렇지 않은 경우 0의 값을 취함)

$\gamma_m$  : 0-1 변수(재생업체  $m$ 의 개설시 1의 값을, 그렇지 않은 경우 0의 값을 취함)

이 문제의 수리 모형은 다음과 같다.

$$\begin{aligned} \text{Minimize } Z = & \sum_i \sum_j s_{ij} x_{ij} + \sum_j \sum_k t_{jk} y_{jk} \\ & + \sum_k \sum_l u_{kl} z_{kl} + \sum_k \sum_m v_{km} o_{km} \\ & + \sum_m \sum_n w_{mj} Rl_{mj} + \sum_l \sum_k Rl_{lk} Rz_{lk} + \sum_j f_j \alpha_j \\ & + \sum_k g_k \beta_k + \sum_m h_m \gamma_m + \psi \sum_m pl_m \sum_k o_{km} \end{aligned} \quad (1)$$

subject to

$$\sum_j x_{ij} \leq a_i, \forall i \quad (2)$$

$$\sum_k y_{jk} \leq b_j \alpha_j, \forall j \quad (3)$$

$$\sum_l z_{kl} + \sum_m o_{km} \leq Sc_k \beta_k, \forall k \quad (4)$$

$$\sum_m o_{km} \leq pd_k Sc_k \beta_k, \forall k \quad (5)$$

$$\sum_k z_{kl} \geq d_l, \forall l \quad (6)$$

$$\sum_k o_{km} \leq e_m \gamma_m, \forall m \quad (7)$$

$$\sum_j \alpha_j \leq P \quad (8)$$

$$\sum_k \beta_k \leq Q \quad (9)$$

$$\sum_m \gamma_m \leq W \quad (10)$$

$$\sum_i x_{ij} + \sum_m Rl_{mj} = \sum_k y_{jk}, \forall j \quad (11)$$

$$\sum_j y_{jk} = \sum_l z_{kl}, \forall k \quad (12)$$

$$\sum_k Rz_{lk} = pc_l \sum_k z_{kl}, \forall l \quad (13)$$

$$\sum_l Rz_{lk} = \sum_m o_{km}, \forall k \quad (14)$$

$$\sum_k o_{km} = \sum_j Rl_{mj} + pl_m \sum_k o_{km}, \forall m \quad (15)$$

$$\sqrt{\sum_j \left( \sum_k y_{jk} / b_j - \sum_j \sum_k y_{jk} / \sum_j b_j \right)^2} \leq C_p \quad (16)$$

$$\alpha_j, \beta_k, \gamma_m \in 0, 1, \forall j, k, m \quad (17)$$

$$\begin{aligned} x_{ij}, y_{jk}, z_{kl}, o_{km}, Rl_{mj}, Rz_{lk} \in N \cup \{0\} \quad (18) \\ \forall i, j, k, l, m \end{aligned}$$

목적함수 (1)은 제품, 부품 및 폐제품의 수송비와 각종 시설의 운용에 필요한 고정비의 합을 최소화한다. 제약식들은 크게 두 유형으로 분류된다. 첫번째 유형은 제약식 (2)~식 (10)으로 각 시설능력의 제약을 나타낸다. 두 번째 유형은 제약식 (11)~식 (15)로 네트워크 내 각 경유노드에서의 흐름보전의 법칙을 나타낸다. 제약식 (2)와 제약식 (3)은 각각 순방향 물류에서 공급업체와 공장의 공급능력의 제약을 나타낸다. 제약식 (4)는 순방향 물류와 역방향 물류에서 동시에 이용할 수 있는 유통센터 능력의 제약을 나타낸다. 제약식 (5)는 유통센터 능력 중 역방향 물류에 대한 할당분의 제약을 나타낸다. 제약식 (6)은 유통센터들의 공급량이 고객 수요를 충족시켜야 함을 나타낸다. 제약식 (7)은 재생업체로의 수송량의 합은 재생업체의 능력을 초과할 수 없음을 나타낸다. 제약식 (8)~제약식 (10)은 각각 가동될 공장, 유통센터, 재생업체의 수는 사전에 주어진 수( $P, Q, R$ )를 초과할 수 없음을 나타낸다. 제약식 (11)은 공급업체들과 재생업체들로부터 각 공장으로서의 유입량은 그 공장으로부터 유통센터들의 유출량과 같아야 함을 나타낸다. 제약식 (12)는 공장들로부터 각 유통센터로의 유입량은 그 유통센터로부터 고객들의 유출량과 같아야 함을 나타낸다. 제약식 (13)은 고객별 제품 수요량과 그 고객으로부터의 폐제품 수거량 사이의 관계를 나타낸다. 제약식 (14)는 고객들로부터 각 유통센터로의 유입량은 그 유통센터로부터 재생업체들로의 유출량과 같아야 함을 나타낸다. 제약식 (15)는 유통센터들로부터 각 재생업체로의 유입량은 그 재생업체로부터 공장들과 폐기업체들로의 유출량의 합과 같아야 함을 나타낸다. 그 외, 제약식 (16)은 공장들간 이용을 편치는 주어진 상한을 초과할 수 없음을 나타낸다. 제약식 (17)은 이진변수를, 제약식 (18)은 비음 정수 조건을 각각 나타낸

다. 이 모형에서 비율을 나타내는 변수( $pd_k, pc_i$ , 및  $p'_m$ )가 사용되는 모든 항은 정수값을 취하는 것으로 가정한다. 이 변수들과 곱해지는 양은 충분히 크다고 가정할 수 있기 때문이다. 이 문제는 NP-hard로 알려져 있다[15]. 따라서 최적해 대신 우수한 근사해를 구할 수 있는 효율적인 알고리즘을 필요로 한다. 본 연구에서는 이 문제에 대한 근사해법으로 CCEA의 적용을 전체로 한다.

### 4. CCEA의 설계

이제 제 2절에서 소개한 CCEA 설계 속성의 분류들에 따라 제 3절의 SCND 문제에 대하여 CCEA의 각 속성들의 값을 지정한다.

#### 4.1 문제의 표현에 관련된 설계 속성들

##### 4.1.1 문제의 분할

- **분할 방법** : SCND 문제는 <그림 3>에서와 같이 모두 8개의 부분문제들로 분할된다. 부분문제 1은 공급업체(재생업체 포함)-공장간, 부분문제 2는 공장-유통센터간, 그리고 부분문제 3

은 유통센터-고객간, 부분문제 4는 고객-유통센터간, 그리고 부분문제 5는 유통센터-재생업체간 수송문제를 나타낸다. 부분문제 6, 7 및 8은 각각 가용한 후보업체들 중에서 실제로 가동될 공장, 유통센터 및 재생업체를 선택하는 문제를 나타낸다. 이러한 분할 방법은 대상 문제 내부의 자연적 구조와 일치한다.

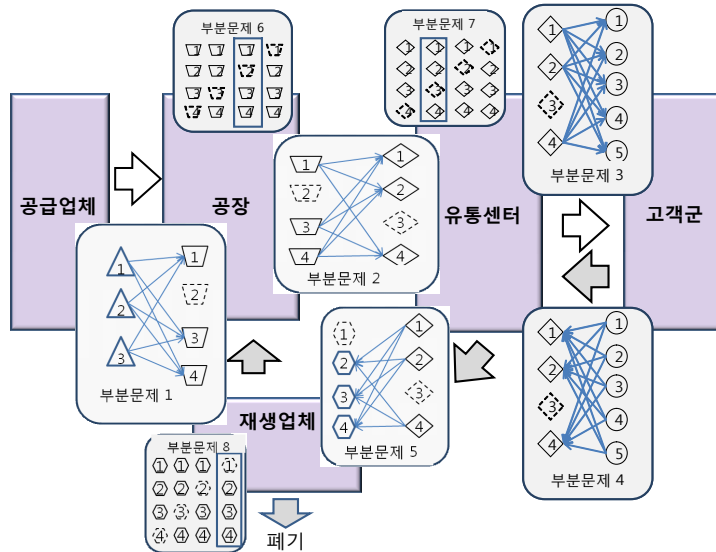
- **분할 시점** : 분할 시점은 설계자에 의해 사전에 결정되는 정적(static) 옵션을 선택한다.

#### 4.1.2 개체집단의 구조

SCND 문제에 대한 8개의 부분문제 각각에 대하여 <그림 3>에서와 같이 모두 8개의 검색체 집단들이 구성된다. 집단 1-5의 각각에 대하여 Gen et al.[11]이 제시한 검색체 표현방법과 해독 알고리즘을 공통적으로 적용함으로써 그 집단에 대응하는 해(즉, 전체 문제에 대해서는 부분해에 해당함)를 구하고자 한다.

##### (a) 집단별 검색체의 표현

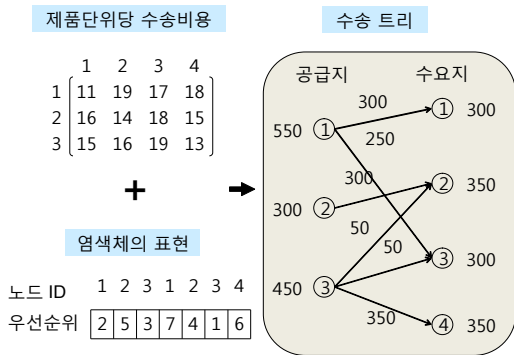
<그림 4>는 3개의 공급지와 4개의 수요지를 지닌 하나의 수송 문제에서 Gen et al.의 방법을 통해



<그림 3> SCND 문제의 8개 부분문제들로의 분할



하나의 실행가능해 즉, 수송 네트워크가 생성되는 예를 나타낸다. 먼저, 공급지의 수와 수요지의 합에 해당하는 수 즉, 7개의 유전인자들로 구성되는 하나의 염색체가 정의된다. 그 염색체의 각 유전인자들의 값은 1부터 7(= 공급지의 수 + 수요지의 수)까지 정수들의 임의의 하나의 순열(permutation)로부터 할당된다. 각 유전인자의 값은 공급지 및 수요지를 연결하는 아크(arc)를 추가하는 과정에서 그에 대응하는 노드가 선택되는 상대적 우선순위를 의미한다. <그림 4>에는 순열 (2, 5, 3, 7, 4, 1, 6)이 할당된 모습을 보여준다. 그 결과, 공급지 노드 1에는 정수 2가, 그리고 수요지 노드 1에는 정수 7이, 그리고 수요지 노드 4에는 정수 6이 할당되었다.



<그림 4> 수송 네트워크의 생성 예

그 다음, 제품 단위당 수송비용 및 각 노드의 용량이 입력 데이터로 주어지면 염색체 해석시 우선순위의 순으로 공급지 또는 수요지 노드를 하나씩 선택해가면서 이 노드와 연결된 최소 비용을 이루는 하나의 (수요지 또는 공급지)노드를 찾아 이 두 노드(공급지 및 수요지)를 연결하는 아크(arc)를 추가해간다. <그림 4>에서 우선순위 값이 가장 높은 수요지 노드 1이 가장 먼저 선택되어 (공급지 1-수요지 1)사이의 아크에 300단위가 할당된 것을 볼 수 있다. 모든 노드들이 우선순위에 따라 순차적으로 선택되고 나면 <그림 4>의 우편에 있는 수송 네트워크(즉, 각 공급지와 수요지간의 수송할당

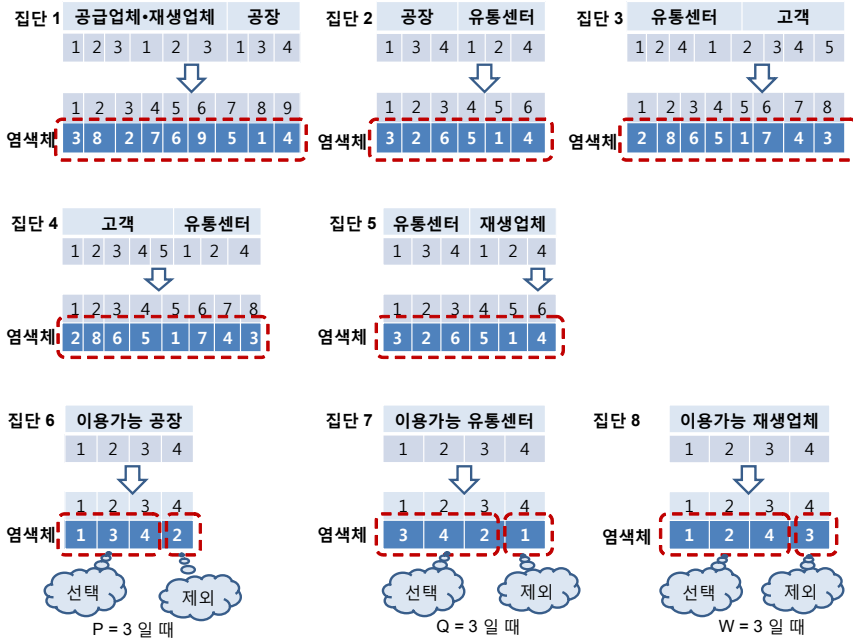
량) 즉, 실행가능해가 구해진다.

<그림 5>에서 집단 2는 공장 {1, 3, 4}가 공급지 집합을 구성하고, 유통센터 {1, 2, 4}가 수요지 집합을 구성하는 하나의 수송문제에 대응한다. Gen et al.[11]의 방법에 따라 집단 2의 각 염색체의 유전인자들을 1부터 6(= 공급지의 수+수요지의 수)까지 정수들의 임의의 하나의 순열 (3, 2, 6, 5, 1, 4)로써 표현되어 있다. 집단 3, 4, 5에 대해서도 공급자 집합과 수요자 집합의 대상만 다를 뿐 집단 2와 똑같은 방식으로 그 염색체를 표현할 수 있다. 다만 집단 1의 경우 공급자 집합은 공급업체들과 재생업체들의 합으로 구성됨을 주목해야 한다. 이처럼 집단 1-5에 속하는 각 염색체는 그 유전인자들을 하나의 정수 순열로써 구성한다.

다음으로, <그림 5>의 집단 6은 이용가능한 4개의 후보공장 가운데 실제로 가동할 공장을 최대 3개까지 선택하는 하나의 문제에 대응한다. 각 염색체의 유전인자들은 1부터 4(= 이용가능한 후보공장의 수)까지 정수들의 임의의 하나의 순열로써 표현된다. 예를 들어, <그림 5>의 집단 6에서는 하나의 순열 (1, 3, 4, 2)가 지정되어 있다. 이 염색체로부터 이에 대응하는 하나의 부분해(선택된 공장들)를 다음과 같이 매우 간단히 생성할 수 있다. 선택할 수 있는 공장의 수는 최대 3이기 때문에(즉,  $P=3$ ) 이 염색체의 좌측 유전인자로부터 3개의 유전인자의 값들을 취하고 그 값에 대응하는 공장을 선택한다. <그림 5>의 집단 6의 경우 유전인자 (1, 3, 4)가 이에 해당하기 때문에 공장 (1, 3, 4)를 선택한다. (그 오른쪽에 있는) 나머지 유전인자의 값에 대응하는 공장(<그림 5>의 집단 6의 경우 공장 2)은 선택에서 제외됨을 의미한다. 집단 7, 집단 8에 대해서도 집단 6과 똑같은 방식으로 그 염색체를 표현할 수 있다.

(b) 염색체 표현과 제약조건식의 충족

각 시설능력의 제약을 나타내는 제약식 (2)~식 (7)은 각각의 공급지들의 공급능력 또는 수요지들의 수요량에 해당하는 파라미터들을 Gen et al.[11]



〈그림 5〉 8개 집단별 검색체의 표현

알고리즘의 입력 데이터로 사용함으로써 모두 충족될 수 있다. 제약식 (8)~제약식 (10)은 집단 6, 7, 8에서의 표현 및 해석방법에 의해 직접적으로 충족된다. 네트워크내 각 경유노드에서의 흐름보전의 법칙을 나타내는 제약식 (11)~제약식 (15)는 각 집단의 개체를 순차적으로 해석하는 과정에서 먼저 해석되는 집단의 산출을 나중에 해석되는 집단의 입력으로 사용함으로써 모두 충족될 수 있다. 마지막으로 제약식 (16)은 현재의 검색체 표현에 의해 충족된다는 보장이 없다. 따라서 이 제약식에 대하여 Goldberg(1989)의 penalty 방법을 사용키로 한다. 이 방법에서 제약식 (16)을 만족하지 못하는 해는 세대변천시 도태되도록 유도하기 위하여 목적함수  $Z^*$ 를 다음과 같이 재정의한다.

$$Z^* = Z + M \times \max\left(\sqrt{\sum_j \left(\sum_k y_{jk}/b_j - \sum_j \sum_k y_{jk}/\sum_j b_j\right)^2} - C_p, 0\right)$$

여기서  $Z$ 는 수리모형의 목적함수인 식 (1)을 그

리고  $M$ 은 임의의 매우 큰 수를 나타낸다. 이제 수리모형 상의 모든 제약식들이 충족될 수 있다.

(c) 집단들간의 상호의존관계

이제 각 집단별로 하나씩의 검색체가 주어졌을 때 이 검색체들을 결합해서 본래 문제에 대한 하나의 (전체)해를 구하는 방법이 필요하다. 각 집단은 자신의 출력정보를 다른 집단에 대한 입력정보로 제공하기도 하고, 다른 집단에서의 출력정보를 자신의 입력정보로 받기도 한다. 따라서 집단들간 검색체를 해석하는 순서가 정해져야 한다. 먼저, 집단 6~집단 8은 각각 운영될 공장, 유통센터 및 재생업체들을 선택하는 문제와 대응되기 때문에, 다른 부분문제들로부터 입력정보를 필요로 하지 않는다. 그리고 제품의 순방향 물류(공급업체-공장-유통센터-고객)는 full 시스템에 해당하기 때문에 먼저 집단 3(유통센터-고객간 수송량 결정)의 해를 구한 후, 집단 3으로부터의 산출정보(각 유통센터의 공급량)를 부분문제 2에 대한 입력정보로 사용하여 집단 2

(공장-유통센터간 수송량 결정)의 해를 구해야 한다. 한편, 집단 3의 해를 구하기 위해서는 집단 7의 해(즉, 운용될 물류센터들의 결정)를 입력정보로 이용해야 한다. 따라서 기호  $a \ll b$ 는 집단 a가 집단 b보다 먼저 그 해를 구해야 한다는 순서를 의미한다고 정의할 때, 순서  $7 \ll 3 \ll 2$ 가 만족되어야 한다. 그리고 집단 2의 해를 구하기 위해서는 집단 6의 해(즉, 운용될 공장들의 결정)를 입력정보로 이용해야 한다. 따라서 순서  $6 \ll 2$ 가 만족되어야 한다. 또한, 폐제품의 역방향 물류(고객-유통센터-제생업체)는 push 시스템에 해당한다. 따라서 집단 3의 해를 입력정보로 사용하여 집단 4(고객-유통센터간 수송량 결정)의 해를 구한 후, 집단 5(유통센터-제생업체간 수송량 결정)의 해를 구할 수 있다. 따라서 순서  $3 \ll 4 \ll 5$ 를 만족해야 한다. 한편, 집단 5의 해를 구하기 위해서는 집단 7의 해(즉, 운용될 제생업체들의 결정)를 입력정보로 이용해야 한다. 따라서 순서  $7 \ll 5$ 를 만족해야 한다. 마지막으로, 집단 1에서는 공장에 대한 부품의 공급지가 공급업체와 제생업체의 두 그룹으로 구성됨으로써 순방향 물류와 역방향 물류의 폐쇄루프가 형성된다. 따라서 집단 1은 나머지 집단들이 모든 해석된 후 마지막으로 해석되어야 한다. 따라서 전체해를 구하기 위하여 위에서 언급한 부분적 순서들을 모두 만족하는 하나의 전체 순서를 정해야 한다. 본 연구에서는 그 중 순서  $7 \ll 3 \ll 6 \ll 2 \ll 4 \ll 8 \ll 5 \ll 1$ 를 사용한다.

#### 4.2 개체의 평가에 관련된 설계 속성들

- 1) **적합도의 계산방법** : 앞에서 재정의한 목적함수  $Z$ 를 각 염색체를 평가하기 위한 적합도 함수로 사용한다. 이러한 적합도 계산방법은 알고리즘의 실행 전에 명백히 정의되기 때문에 명시적(explicit) 방법에 해당한다.
- 2) **적합도의 이용방법** : 산출된 각 개체의 적합도는 두 가지 용도로 사용된다. 먼저, 각 개체의

적합도는 선택 연산시에 이용된다. 즉, 선택 연산으로서 이진 토너먼트(binary tournament) 연산을 적용하여 대상 집단에서 임의로 두 개의 개체를 선택한 후, 그 중 더 좋은 적합도를 지닌 개체를 그 집단의 다음 세대의 개체로 선택할 때 사용된다. 또한, 각 개체의 적합도는 한 집단 내에서 그 집단을 대표하는 협력자를 정하는 기준으로 사용된다.

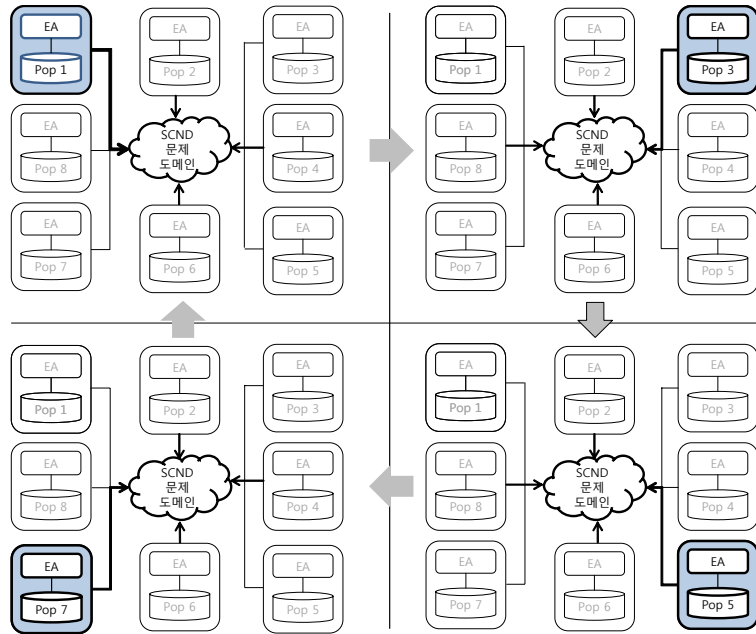
3) **집단들간 상호작용 방식** : 이와 관련된 속성별로 다음과 같은 옵션들을 적용한다.

- 상호작용 샘플 크기 : 평가대상 집단 내 각 개체는 나머지 모든 집단들로부터 각각 하나씩의 개체(협력자)와 결합(상호작용)한다. 따라서 이에 대응하는 하나의 해를 구할 수 있다.
- 협력자 선택 방식 : 적합도 기반의 선택 방식을 채택한다.
- 상호작용 결과에 대한 점수부여 : 상호작용 샘플 크기가 1이기 때문에 산출된 하나의 값을 가공하지 않고 바로 적합도로 사용한다.

4) **협력자의 업데이트 방식** : 논문의 목적상 순차적 업데이트 방식과 병행적 업데이트 방식을 모두 채택하여 실험기로 한다. 특히, 순차적 업데이트 방식에서는 <그림 6>에서와 같이 각 집단이 라운드 로빈 방식으로 하나씩 평가된다. 하나의 집단이 자신의 평가 순서가 되면 그 집단내 각 개체는 나머지 모든 집단들로부터의 협력자들과 결합하여 자신의 적합도를 구하게 된다. 이 그림에서는 8개의 집단 가운데 집단 1, 3, 5, 7의 평가 모습만을 나타내고 있다.

#### 4.3 유전 연산

각 집단내 염색체의 세대교체는 다른 집단과는 독립적으로 선택(selection), 교배(crossover) 및 돌연변이(mutation) 연산을 통해 이루어진다. 8개 집단의 염색체는 모두 정수의 순열 형태로 표현되어 있기 때문에 모든 집단에 같은 연산을 공통적으로 적용한다. 선택 연산으로는 엘리트리스트(elitist) 전



〈그림 6〉 다중 집단에서의 라운드로빈 방식의 집단별 평가

락을 수반한 이진 토너먼트 연산, 교배 연산으로는 OX(order crossover) 연산, 그리고 돌연변이 연산으로는 swap 연산을 적용한다.

### 5. 집단들간 상호작용 방식들의 성능 비교

이제 SCND 문제에 대하여 네 가지 유형의 집단들간 상호작용방식 및 두 가지 유형의 협력자 업데이트 방식을 설정하고 이 두 가지 요소들의 조합으로 만들어지는 해법들을 테스트 문제들에 적용하는 실험을 실시하고 그 결과를 비교 분석한다.

#### 5.1 집단들간 상호작용방식의 설정

본 SCND 문제에 대하여 제 4절에서 8개의 염색체 집단들을 정의하였다. 이 염색체 집단들은 그 성격상 두 유형으로 분류될 수 있다.(〈그림 3〉 참고) 첫 번째 유형(집단유형 1)은 공급사슬 내 특정 계층에서의 공급지들과 수요지들간의 수송문제에

대응하는 것으로서, 집단 {1, 2, 3, 4, 5}가 여기에 속한다. 두 번째 유형(집단유형 2)은 업체들의 후보군들 중에서 실제로 가동될 업체를 선택하는 문제에 대응하는 것으로서, 집단 {6, 7, 8}이 여기에 해당한다.

한편, 각 집단유형별로 협력자를 선택할 때 베스트 적합도 기반과 무작위 기반이라는 두 가지 방식을 생각해 볼 수 있다. 따라서 두 가지 유형의 집단과 두 가지 유형의 협력자선택 방식의 조합에 따라 만들어지는 4가지의 집단들간 상호작용 방식을 〈그림 7〉과 같이 나타낼 수 있다.

- 방식 1 : 집단유형 1과 집단유형 2에 속하는 각 집단의 내부에서 모두 무작위 방식으로 하나의 개체를 선택하여 이것을 그 집단의 협력자로 사용한다. 이 방식은 대상 문제에 대하여 하나의 염색체를 형성하고 이 염색체를 몇 개의 부분으로 분할한 후 각 부분마다 전통적인 GA를 독립적으로 적용하고 진화시켜나가는 방식이다.
- 방식 2 : 집단유형 1에 속하는 각 집단 안에서는 하나의 개체를 무작위로 선택하고, 집단유형 2

		집단유형 2의 협력자 선택방식	
		무작위	베스트 적합도
집단유형 1의 협력자 선택방식	무작위	상호작용 방식 1	상호작용 방식 2
	베스트 적합도	상호작용 방식 3	상호작용 방식 4

〈그림 7〉 4가지 유형의 집단들간 상호작용 방식

〈표 1〉 테스트 문제들의 파라미터

문제	공급 업체 수	공장 수	공장 수 상한	유통 센터 수	유통 센터 수 상한	재생 업체 수	재생 업체 수 상한	고객 수	폐제품 수거 비율( $pc_f$ )	폐제품 폐기 비율( $pl_m$ )	$C_p$
1	3	4	3	4	3	4	3	5	0.2	0.2	0.3
2	4	5	4	5	4	5	4	10	0.2	0.2	0.3
3	6	8	5	8	5	8	5	15	0.2	0.2	0.3
4	8	10	7	10	7	10	7	20	0.2	0.2	0.3
5	10	20	14	20	14	20	14	40	0.2	0.2	0.4
6	15	30	20	30	20	30	20	50	0.2	0.2	0.4

에 속하는 각 집단 안에서는 하나의 베스트 개체를 찾아서 이것들을 각각 그 집단의 협력자로 사용하는 방식이다. 이것은 GA와 CCEA가 혼합된 방식에 해당한다.

- 방식 3 : 집단유형 1에 속하는 각 집단 안에서는 하나의 베스트 개체를 탐색하고, 집단유형 2에 속하는 각 집단 안에서는 하나의 개체를 무작위로 선택하여 이것들을 각각 그 집단의 협력자로 사용하는 방식이다. 이것은 GA와 CCEA가 혼합된 또 다른 방식에 해당한다.
- 방식 4 : 집단유형 1과 집단유형 2에 속하는 각 집단에서 모두 그 내부에서 하나의 베스트 개체를 탐색하여 이것들을 각각 그 집단의 협력자로 사용하는 방식이다. 이 방식은 전형적인 CCEA에 해당한다.

다음으로, 세대교체에 따라 각 집단이 그 협력자를 업데이트하는 방식이 생성해의 성능에 미치는 영향을 파악하고자 한다. 따라서 제 2절에서 설명한 바 있는 다음의 2가지 옵션을 고려대상으로 설정한다.

- 순차적 업데이트
- 병행적 업데이트

4가지의 집단들간 상호작용방식과 2가지의 협력자 업데이트 유형의 조합에 따라 모두 8가지의 해법을 생각할 수 있다. 그런데 상호작용방식 1에서는 사실상 하나의 염색체 집단으로 구성되기 때문에 협력자의 업데이트 개념이 존재하지 않아 2가지 해법으로 대응되지 않는다. 따라서 모두 7가지의 해법을 고려할 수 있다. 이제 각 해법별로 여러 테스트 문제들에 대한 실행을 통해 그 성능을 파악하고자 한다.

## 5.2 실험 계획

실험에 필요한 알고리즘들은 모두 비주얼 C++ 언어를 사용하여 객체지향 방식으로 코딩한다. 각 알고리즘을 실행하기 위하여 6개의 테스트 문제들을 준비한다. 테스트 문제의 크기와 관련된 파라미터들의 값은 <표 1>과 같다. 모든 테스트 문제에 4가지의 각 방식을 적용해서 실행할 때 8개의 염색체 집단들이 모두 같은 집단 크기를 가지도록 설정

한다. 테스트 문제별 집단 크기( $n$ )는 <표 2>에 나타나 있다.

<표 2> 문제별 각 방식의 집단크기와 생성 세대 수

문제	집단 크기( $n$ )	생성 세대 수			
		방식 1	방식 2	방식 3	방식 4
1	40	800	200	133	100
2	50	1,200	300	200	150
3	70	1,600	400	267	200
4	100	2,400	600	400	300
5	150	4,000	1,000	667	500
6	180	8,000	2,000	1,333	1,000

CCEA나 GA의 실행시간은 모든 생성세대에 걸쳐 평가될 염색체의 수에 절대적으로 의존한다. 따라서 4가지 방식별로 평가될 염색체의 수가 모두 같아지도록 각 해법에서의 생성세대 수를 정하도록 한다. 방식 1부터 방식 4까지의 네 가지 방식에서 한 세대당 평가되는 개체수의 비율은  $n:4n:6n:8n$  이기 때문에 비율(1:1/4:1/6:1/8)에 따라 각 해법에서의 생성 세대수를 정한다. 그 결과 각 방식과 테스트 문제의 조합에 따라 얻어진 생성세대 수가 <표 2>에 나타나 있다.

이상 설계된 각 해법들을 테스트 문제별로 5회씩 반복 실행한다. 각 해법의 실행과정에서 염색체의 유전인자에 값을 부여할 때 난수가 사용된다. 이 난수의 생성에 사용되는 시드(seed) 값을 변경시킴으로써 각 해법들을 반복 실행한다.

### 5.3 실험 결과

위 실험계획에 따라 7가지의 해법에 의해 생성된 해에 대응하는 목적함수 값들을 <표 3>에 나타내었다. 각 해법별로 5회 반복실험에서 얻어진 5개의 목적함수 값을 포함하여 그 평균 및 최소값도 나타내었다. 각 해법의 성능을 비교하기 위하여 <표 3>에서의 테스트 문제별로 각 해법에 의해 얻어진 목적함수 값들의 평균( $a$ )을, 방식 4에 의해 얻어진 목적함수 값들의 평균( $b$ )으로 나눈 값, 즉 비율( $a/b$ )

을 <표 4>에 나타내었다. 마찬가지로 방식으로 테스트 문제별로 각 해법에 의해 얻어진 목적함수 값들의 최소값( $c$ )을, 방식 4에 의해 얻어진 목적함수 값들의 최소값( $d$ )으로 나눈 값, 즉 비율 ( $c/d$ )도 <표 4>에 나타내었다. 이 표로부터 다음과 같은 결과들을 관찰할 수 있다.

1. 순차적 업데이트 방식과 상호작용방식 4를 적용하는 해법이 목적함수 값의 평균과 최소값 면에서 모든 7가지 해법들 가운데 가장 좋은 성능을 보였다. 그리고 병행적 업데이트 방식과 상호작용방식 4를 적용하는 해법이, 비율 ( $a/b$ )이 1.00~1.21, 그리고 비율( $c/d$ )이 1.00~1.23사이의 값을 보임으로써 두 번째로 좋은 성능을 보였다.
2. 순차적 업데이트 하에서의 방식 2와 3은 상호간에 비율 ( $a/b$ )의 값이 0.00~0.06, 그리고 비율 ( $c/d$ )의 값이 0.01~0.05사이의 근소한 차이만을 보임으로써 이 두 방식은 서로 비슷한 성능을 가진다고 판단된다. 그리고 병행적 업데이트 하에서의 방식 2와 방식 3은 상호간에 비율( $a/b$ )의 값이 0.00~0.06, 그리고 비율( $c/d$ )의 값이 0.00~0.13사이의 차이만을 보임으로써 이 두 방식도 비슷한 성능을 지닌다고 판단된다.
3. 순차적 업데이트 하에서의 방식 2와 병행적 업데이트 하에서의 방식 2를 비교해보면 상호간에 비율 ( $a/b$ )의 값이 0.00~0.03, 그리고 비율( $c/d$ )의 값이 0.01~0.04사이의 차이만을 보임으로써 이 두 해법은 매우 비슷한 성능을 보였다. 그리고 순차적 업데이트 하에서의 방식 3과 병행적 업데이트 하에서의 방식 3을 비교해보면 상호간에 목적함수 값의 비율 ( $a/b$ )의 값이 0.00~0.07, 그리고 비율( $c/d$ )의 값이 0.01~0.13사이의 차이만을 보임으로써 이 두 방식은 대체로 비슷한 성능을 보인다고 판단된다.
4. GA 기법에 해당하는 해법 1은 비율 ( $a/b$ )의 값과 비율( $c/d$ )의 값이 각각 1.12~1.35 및 1.13~1.38사이의 값을 가짐으로써 7가지 방식 중에서 가장 좋지 못한 성능을 보였다.

〈표 3〉 7가지 해법으로 생성된 해의 목적함수 값

		순차적 업데이트				병행적 업데이트		
		방식 1	방식 2	방식 3	방식 4	방식 2	방식 3	방식 4
문제 1	1	5727	5951	5603	4769	6155	5711	5605
	2	5870	6369	6285	6011	7370	4956	6645
	3	5877	5804	6004	5028	6200	5391	6163
	4	5885	6215	6200	5607	6140	5993	6095
	5	6098	6475	6198	4890	5647	6877	7292
	평균	5891	6163	6058	5261	6302	5786	6360
	최소값	5727	5804	5603	4769	5647	4956	5605
문제 2	1	6421	6115	5799	5486	5898	5925	5486
	2	6181	6061	5773	5486	5991	5861	5486
	3	6251	5744	5941	5571	5766	5944	5486
	4	6431	5773	5717	5486	6084	6161	6018
	5	6301	6150	5789	6018	6082	5852	5597
	평균	6317	5969	5838	5609	5964	5949	5615
	최소값	6181	5744	5773	5486	5766	5852	5486
문제 3	1	12625	12432	10749	8668	11564	12924	10950
	2	12009	11240	10990	9496	12444	10800	9287
	3	11902	11708	11551	9299	11549	11518	11268
	4	12006	12200	11381	11744	10949	12453	11446
	5	12131	10577	12011	9055	12332	12024	9466
	평균	12131	11631	11097	9154	11768	11747	10483
	최소값	11902	10577	10749	8668	10949	10800	9287
문제 4	1	15714	16064	15615	13474	15182	16581	15502
	2	15310	16341	16428	15630	16148	13945	15717
	3	15981	15347	15755	12076	16251	16489	14383
	4	15865	15610	15420	13482	15998	14784	16033
	5	16602	16598	17088	11678	16474	16098	14521
	평균	15894	15992	16061	13268	16011	15579	15231
	최소값	15310	15347	15420	12076	15182	13945	14383
문제 5	1	30011	27569	28581	22388	27225	28456	23107
	2	28758	28368	28171	21587	26686	28567	23427
	3	29113	27750	28579	21797	27991	29010	23278
	4	29407	27213	28281	21497	27247	27727	23671
	5	29985	27267	28201	22093	27645	28445	24135
	평균	29455	27633	28363	21872	27359	28441	23524
	최소값	28758	27213	28171	21493	26686	27727	23107
문제 6	1	42571	42057	40020	33189	41319	41166	36405
	2	42663	41514	41321	34460	42192	42003	37159
	3	42466	40356	41604	34371	40237	40694	38581
	4	42768	42219	41880	30756	42241	41869	36374
	5	43283	41893	40556	32572	41144	41329	36600
	평균	42751	41608	41076	33070	41427	41412	37024
	최소값	42466	40356	40020	30756	40237	40694	36374

〈표 4〉 네 가지 방식으로 생성된 목적함수 값의 비율

		순차적 업데이트				병행적 업데이트		
		방식 1	방식 2	방식 3	방식 4	방식 2	방식 3	방식 4
문제 1	평균	1.12	1.17	1.15	1	1.20	1.10	1.21
	최소값	<b>1.20</b>	<b>1.22</b>	<b>1.17</b>	<b>1</b>	<b>1.18</b>	<b>1.04</b>	<b>1.18</b>
문제 2	평균	1.13	1.06	1.03	1	1.06	1.06	1.00
	최소값	<b>1.13</b>	<b>1.05</b>	<b>1.04</b>	<b>1</b>	<b>1.05</b>	<b>1.07</b>	<b>1.00</b>
문제 3	평균	1.26	1.21	1.15	1	1.22	1.22	1.09
	최소값	<b>1.37</b>	<b>1.22</b>	<b>1.24</b>	<b>1</b>	<b>1.26</b>	<b>1.25</b>	<b>1.07</b>
문제 4	평균	1.20	1.21	1.21	1	1.21	1.17	1.15
	최소값	<b>1.31</b>	<b>1.31</b>	<b>1.32</b>	<b>1</b>	<b>1.30</b>	<b>1.19</b>	<b>1.23</b>
문제 5	평균	1.35	1.26	1.30	1	1.25	1.30	1.08
	최소값	<b>1.34</b>	<b>1.27</b>	<b>1.31</b>	<b>1</b>	<b>1.24</b>	<b>1.29</b>	<b>1.07</b>
문제 6	평균	1.29	1.26	1.24	1	1.25	1.25	1.12
	최소값	<b>1.38</b>	<b>1.31</b>	<b>1.30</b>	<b>1</b>	<b>1.31</b>	<b>1.32</b>	<b>1.18</b>
전체	평균	1.12~1.35	1.06~1.26	1.03~1.30	1	1.06~1.25	1.06~1.30	1.00~1.21
	최소값	<b>1.13~1.38</b>	<b>1.05~1.31</b>	<b>1.04~1.32</b>	<b>1</b>	<b>1.05~1.31</b>	<b>1.04~1.32</b>	<b>1.00~1.23</b>

SCND 문제에 대한 이상과 같은 실험결과로부터 다음과 같은 사실을 확인할 수 있다. 첫째, SCND 문제에 CCEA 해법을 적용할 시 모든 집단에서 베스트 적합도 기반의 협력자를 적용함과 동시에 협력자를 순차적으로 업데이트하는 옵션을 채택하는 해법이 가장 좋은 성능을 보인다. 둘째, (모든 집단에서 베스트 기반의 협력자를 적용하는) 상호작용 방식 4의 적용이 협력자의 순차적 업데이트 방식의 적용보다 해법의 성능에 훨씬 더 큰 영향을 미친다. 셋째, 일부 집단에는 베스트 적합도 기반의 상호작용방식을 적용하고 나머지 집단에는 무작위 기반의 상호작용 방식을 적용하는 해법들(방식 2와 방식 3)은 각각 GA와 CCEA의 혼합 형태의 하나로서 협력자의 업데이트 방식에 관계없이 모두 비슷한 성능을 보인다. 이 해법들은 순차적 업데이트 방식이 적용된 CCEA에 비해 그 성능이 상대적으로 떨어진다. 넷째, 전통적인 GA 즉, 상호작용 방식 1의 적용은 가장 좋지 못한 해를 생성함으로써 여러 부분문제들로 구성된 형태를 지닌 SCND 문제에 GA를 적용하는 데 한계가 있음을 보여준다.

## 6. 결 론

CCEA는 최적화 문제의 근사해를 구할 수 있는 강력한 수단으로 입증되어 왔다. CCEA는 특히 GA가 좋은 성능을 보이지 못하거나 GA가 적용되기 어려운 문제들의 해를 구하는 데 사용될 수 있다. CCEA에서는 문제의 일부분에 대한 해를 탐색하기 위해 두 개 이상의 집단을 사용한다. 각 집단은 독립적으로 진화하며, 개체들이 평가될 때만 상호작용이 일어난다. 염색체 집단들간의 상호작용은 GA에는 존재하지 않고 CCEA에만 존재하는 가장 큰 특징이다. 본 연구에서는 SCND 문제를 대상으로 하여 CCEA 안에서 염색체 집단들간의 다양한 상호작용 방식들이 해의 성능에 미치는 영향을 파악하고자 하였다.

먼저, CCEA에 필요한 여러 설계속성들을 분류틀로 표현하였다. 이 분류틀 속에서 CCEA의 모든 설계속성들을 최상위 수준에서 개체의 ‘표현’ 및 개체의 ‘평가’라는 두 유형으로 분류하고, 각 유형별 하위 설계속성들에 적용할 수 있는 여러 옵션들을 소개하였다. 그리고 SCND 문제와 이에 대응하



는 수리모형을 소개한 후 이 문제에 대하여 CCEA를 적용하는 데 필요한 모든 속성들을 다음과 같이 설계하였다.

주어진 SCND 문제에 대하여 모두 8개의 염색체 집단들을 설계하였다. 이 집단들은 그 성격에 따라 크게 두 가지의 유형으로 분류할 수 있으며, 각 집단 유형에는 베스트 적합도 기반과 무작위 선택의 두 가지의 협력자 선택 방식이 존재한다. 이에 따라 모두 네 가지의 집단간 상호작용 방식을 설정하였다. 한편 세대교체시 협력자를 업데이트 할 때 병행적 업데이트와 순차적 업데이트의 두 가지 방식을 설정하였다. 네 가지의 집단들간 상호작용 방식과 두 가지의 협력자 업데이트 방식의 조합에 따라 모두 7가지의 해법을 설정하였다. 이 해법들에 대한 성능을 파악하기 위하여 테스트 문제들을 대상으로 실험을 실시하였다.

실험결과로부터 다음과 같은 사실을 확인하였다. SCND 문제에 대하여 CCEA 해법을 적용할 시 모든 집단에서 베스트 적합도 기반의 협력자를 사용함과 동시에 협력자를 순차적으로 업데이트하는 옵션을 채택할 때 가장 좋은 성능을 보였다. 일부 집단에는 베스트 적합도 기반의 상호작용 방식을 적용하고 나머지 집단에는 무작위 기반의 상호작용 방식을 적용하는 해법들은 순차적 업데이트 방식이 적용된 CCEA에 비해 그 성능이 크게 떨어졌다. 전통적인 GA는 여러 부분문제들로 구성된 형태를 지닌 SCND 문제에 적용하는 데 한계가 있음을 보여 주었다.

본 연구에서는 CCEA의 설계시 집단들간의 상호작용 방식으로서 집단별로 단 하나씩의 협력자의 사용을 그 전제로 하였으나 향후 각 집단별로 여러 개의 협력자를 활용하는 방법도 연구할 필요가 있을 것이다.

## 참 고 문 헌

- [1] 한용호, “역물류 네트워크 모델의 최적화를 위한 협력적 공진화 알고리즘”, 『경영과학』, 제27권, 제3호(2010), pp.15-31.
- [2] 한용호, “2단계 수송문제에 대한 협력적 공진화 알고리즘 기반의 혁신적 해법”, 『산업혁신연구』, 제25권, 제4호(2009), 경성대학교 산업개발연구소, pp.227-251.
- [3] 한용호, “협력적 공진화 알고리즘에 기반한 다단계 공급사슬 네트워크의 설계”, 『한국SCM학회지』, 제11권 제2호(2011), pp.87-96.
- [4] 한용호, “순열 표현 기반의 협력적 공진화 알고리즘을 사용한 다단계 공급사슬 네트워크의 설계”, 『경영과학』, 제29권 제2호(2012), pp.1-14.
- [5] 한용호, “5개 집단의 협력적 공진화 알고리즘을 사용한 다단계 공급사슬 네트워크 설계”, 『한국SCM학회지』, 제12권, 제2호(2012), pp.1-10.
- [6] 한용호, “다중 집단 기반의 협력적 공진화 알고리즘을 사용한 폐쇄루프 공급사슬 네트워크의 설계”, 『한국SCM학회지』, 제13권, 제1호(2013), pp.55-66.
- [7] Danoy, G., P. Bouvry, and T. Martins, *hLCGA : A Hybrid Competitive Coevolutionary Genetic Algorithm*, Proceedings of the 6<sup>th</sup> International Conference on Hybrid Intelligent Systems, (2006), pp.48-51.
- [8] Danoy, G., B. Dorrnsoro, and P. Bouvry, *Overcoming Partitioning in Large Ad Hoc Networks Using Genetic Algorithms*, GECO, (2009), pp.1347-54.
- [9] Dorrnsoro, B., G. Danoy, A.J. Nebro, and P. Bouvry, “Achieving Super-Linear Performance in Parallel Multi-Objective Evolutionary Algorithms by Means of Cooperative Coevolution,” *Computers and Operations Research*, Vol.40 No.6(2013), pp.1552-1563.
- [10] Eriksson, R. and B. Olsson, *Cooperative Coevolution in Inventory Control Optimisation*, Artificial Neural Nets and Genetic Algorithms : Proceedings of the International Conference, (1997), pp.583-587.

- [11] Gen, M., F. Altıparmak, and L. Lin, "A Genetic Algorithm for Two-Stage Transportation Problem Using Priority-Based Encoding," *OR Spectrum*, Vol.28(2006), pp.337-354.
- [12] Goldberg, D.E., *Genetic Algorithm : in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [13] Pimpawat, C. and N. Chaiyaratana, "Three-Dimensional Container Loading Using a Cooperative Co-evolutionary Genetic Algorithm," *Applied Artificial Intelligence*, Vol.18(2004), pp.581-601.
- [14] Potter, M. and K. De Jong, *A Cooperative Coevolutionary Approach to Function Optimization*, In : Parallel Problem Solving from Nature (PPSN III), Springer, (1994), pp.249-257.
- [15] Wang, H.F. and H.W. Hsu, "A Closed-Loop Logistics Model with a Spanning-Tree Based Genetic Algorithm," *Computers and Operations Research*, Vol.37(2010), pp.376-389.
- [16] Wiegand, R.P., *An Analysis of Cooperative Algorithms*, Dissertation for Ph.D. at George Mason University, 2003.
- [17] Xing, L.N., Y.W. Chen, and K.W. Yang, "Multi-population Interactive Coevolutionary Algorithm for Flexible Job Shop Scheduling Problem," *Computational Optimization and Applications*, Vol.48(2011), pp.139-155.