

역물류 네트워크 모델의 최적화를 위한 협력적 공진화 알고리즘

†한 용 호*

A Cooperative Coevolutionary Algorithm for Optimizing a Reverse Logistics Network Model

†Yong Ho Han*

■ Abstract ■

We consider a reverse logistics network design problem for recycling. The problem consists of three stages of transportation. In the first stage products are transported from retrieval centers to disassembly centers. In the second stage disassembled modules are transported from disassembly centers to processing centers. Finally, in the third stage modules are transported from either processing centers or a supplier to a manufacturer, a recycling site, or a disposal site. The objective is to design a network which minimizes the total transportation cost. We design a cooperative coevolutionary algorithm to solve the problem. First, the problem is decomposed into three subproblems each of which corresponds to a stage of transportation. For subproblems 1 and 2, a population of chromosomes is constructed. Each chromosome in the population is coded as a permutation of integers and an algorithm which decodes a chromosome is suggested. For subproblem 3, an heuristic algorithm is utilized. Then, a performance evaluation procedure is suggested which combines the chromosomes from each of two populations and the heuristic algorithm for subproblem 3. An experiment was carried out using test problems. The experiments showed that the cooperative coevolutionary algorithm generally tends to show better performances than the previous genetic algorithm as the problem size gets larger.

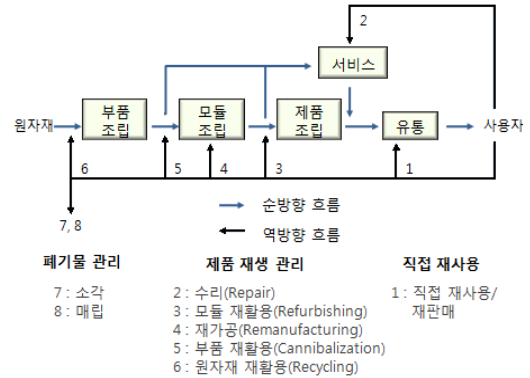
Keywords : Cooperative Coevolutionary Algorithm, Genetic Algorithm, Reverse Logistics Network Design

1. 서 론

오늘날 제조업체는 엄격한 법률의 제정, 사회적 책임, 기업 이미지, 환경에 대한 소비자의 인식 및 경제적 이익 등으로 인하여 보다 환경친화적인 제품을 공급해야 할 뿐만 아니라, 제품 수명이 다한 뒤에는 사용된 제품을 회수하는 역물류의 노력도 더 기울여야 하는 실정이다. Rogers and Tibbon-Lembke[25]에 의하면 역물류란 사용된 제품의 가치를 회복시키거나 적절히 처분할 목적으로 원자재, 재공품, 완제품 및 관련 정보를 소비 지점에서부터 본래 지점까지 효율적으로 그리고 낮은 비용으로 흐름을 계획하고, 실행하고, 통제하는 과정으로 정의된다. 특히 고가 제품의 경우 사용된 제품의 재생은 폐기보다 경제적으로 더욱 유리하기 때문에 재생 방법이 많이 사용된다. 사용된 제품들을 여러 채널을 통하여 회수하기 위하여 네트워크를 운용해야 한다. 만약 제품이 효율적으로 회수되지 않는다면 기업들은 더 많은 비용을 초래하여 신제품의 원가를 상승시키게 된다. 따라서 제품의 회수를 위한 네트워크는 효율적이며 경제적으로 운용되어야 한다.

Thierry et al.[27]에 의하면 역물류를 통한 제품의 회수 패턴은 그 목적에 따라 직접 재사용, 재생 및 폐기의 세 가지 유형으로 분류된다. 특히 재생 유형은 <그림 1>에서처럼 수리(repair), 모듈재활용(refurbishing), 재가공(remanufacturing), 일부재활용(cannibalization) 및 원자재 재활용(recycling)의 5가지로 분류될 수 있다. 수리는 사용된 제품을 사용가능한 정도로 되돌리는 것을 의미한다. 수리된 제품의 품질은 일반적으로 새 제품의 품질보다 상대적으로 떨어진다. 모듈재활용은 사용된 제품을 특정한 품질수준까지 올리는 것을 의미한다. 사용된 제품을 모듈들로 분해한 후, 중요한 모듈들을 검사하고 수리하거나 교체한다. 품질 표준은 새 제품보다 덜 엄격하다. 재가공은 사용한 제품을 새 제품과 같은 정도의 엄격한 품질 표준으로 올리는 것을 의미한다. 사용된 제품은 완전히 분해되고 모든 모듈과 부품들을 철저히 검사한다. 앞의 세 가

지 방법에서는 사용된 제품의 상당부분이 재사용된다. 일부재활용은 사용된 제품으로부터 재사용가능한 부품들의 일부만을 회수하는 것이다. 리사이클링에서는 제품과 컴포넌트의 특성 및 기능이 상실되고, 다만 제품과 컴포넌트로부터 나오는 원료를 재사용하는 것이다.



<그림 1> 통합 공급 사슬(Thierry et al.[27])

Fleischmann et al.[6]은 선행 연구에서 제기된 여러 가지 역물류에 대한 수리적 모형들을 크게 유통계획, 재고통제 및 생산계획의 세 분야로 분류할 수 있음을 보였다. Mutha and Pokharel[19]은 30여 편의 역물류 네트워크 설계 문제에 대한 선행 연구를 검토한 결과 모형 개발을 필요로 하는 상황이 각각 다 독특함을 발견하였으며, 시설의 위치와 용량의 결정을 통한 역물류 비용의 절감이 중요함을 보였으며, 그리고 회수된 제품의 재가공 및 중고시장에서의 판매도 중요한 요소임을 보였다. 일부 연구에서는 사용된 제품은 여러 모듈들을 포함한다는 사실도 고려하였다. 디자인 측면에서는 제조업체들은 최종 조립을 위한 단계들을 줄일 뿐만 아니라 사용된 제품의 보다 빠른 분해와 수리를 촉진하기 위하여 그들의 제품을 점차 모듈화하고 있다. 따라서 모듈화는 사용된 제품으로부터 나오는 사용가능한 모듈의 폐기를 방지하는 데 도움이 된다. 예를 들어, Kusumastuti et al.[14]와 Franke et al.[7]은 재가공을 위한 회수 제품의 모듈식 아키텍처를 고려하였다. Pokharel and Mutha[22]는 역물류

에 대한 기존의 연구들을 역물류 시스템의 입력, 처리, 구조, 출력의 네 그룹으로 나누어 검토한 결과, 역물류 네트워크의 구축에 관하여 다양한 유형의 수리적 모형이 이미 제시되어 있음을 보여주었다.

이 대부분의 모형들은 혼합정수계획법으로 모델링되었다(Chandra and Grabis[3]). CPLEX나 GAMS (General Algebraic Modeling System)와 같은 소프트웨어를 사용함으로써 이 모형들에 대한 최적해 또는 근사해를 구하였다. 예를 들어, Jayaraman et al.[11]은 CPLEX를, Kim et al.[12]은 OPL Studio를, 그리고 Mutha and Pokharel[19]과 Srivastava [26]는 GAMS를 사용하였다. Pati et al.[21]은 목표 계획법 모형을 만들어 LINDO-32를 사용하여 해를 구하였다.

그러나 문제의 복잡성(complexity)과 많은 수의 변수 및 제약 조건으로 인하여 현실적인 규모의 문제는 물론이며 상당히 적은 규모의 문제에 대해서도 역물류 모형을 풀기위한 기존의 MIP 도구의 사용은 제한된다(Jayaraman et al.[11]). 따라서 휴리스틱 해법이 이 문제를 해결하는 데 유용한 접근방법이 될 수 있다. 예를 들어, Lu and Bostel[17]은 라그랑지 완화(Lagrangian relaxation) 기법에 기반을 둔 휴리스틱 알고리즘을 제시하였다.

최근 들어, 유전 알고리즘(genetic algorithm)은 최적화 문제에 대하여 참신한 접근방법으로서 기본적인 네트워크, 물류 네트워크, 통신 네트워크, 생산 스케줄링, 조립 라인 밸런싱 등의 최적화 문제를 해결하는 데 사용되었다(Gen et al.[9]). 역물류 네트워크 문제에 대해서도 유전 알고리즘(genetic algorithm)이 성공적으로 적용되었다(Lin et al.[16], Min et al.[18], Ko et al.[13]). 특히 Gen et al.[8]은 2단계 수송문제에 대하여 각 단계별로 우선순위(priority) 기반의 염색체 표현방법을 사용하고, 수송문제의 고유한 특성을 이용하여 주어진 염색체의 값으로부터 그에 대응하는 단계별 해(수송계획)의 생성 방법을 제시한 후, 유전 알고리즘을 단계별로 순차적으로 적용함으로써 전체 해를 구하는 방법을 제시하였다. Lee et al.[15]은 Gen et al.[8]에서 제시

된 유전 알고리즘을 기반으로 하여 역물류 문제에 대하여 근사해를 구해주는 알고리즘을 제시하였다.

최근 진화 알고리즘 분야에서는 유전 알고리즘 이외에 협력적 공진화 알고리즘(cooperative coevolution algorithm)의 사용이 늘어나고 있다. 협력적 공진화 알고리즘이란 자연세계에서 공생관계를 갖는 서로 다른 종들이 상호작용하고 상호적응하며 공생진화하는 과정을 모방한 일종의 확률적 탐색 기법으로, 상호의존적인 여러 부분문제들로 구성된, 복잡도가 높은 최적화 문제를 해결하는 데 효율적인 것으로 알려져 있다(Potter[24]). 이 알고리즘에서는 대상 문제를 보다 작은 몇 개의 부분문제로 분할하고, 이 부분문제들을 독립적으로 푼 후, 이 부분문제들을 상호결합함으로써 본래 문제의 해를 구한다(Potter and DeJong[23]). 부분문제 사이의 상호작용은 부분문제에 대응하는 모집단내 각 개체(individual)를 상호협력적으로 평가할 때에만 이루어진다. 협력적 공진화 알고리즘은 다목표 의사결정(신경석, 김여근[1]), 컨테이너 작업 스케줄링(Park et al.[20]), 2단계 수송 네트워크 설계(한용호[2]) 등의 문제에 성공적으로 적용된 바 있다. 그러나 역물류 네트워크의 설계 문제에 대하여 협력적 공진화 알고리즘을 적용한 연구는 아직 수행된 바 없다.

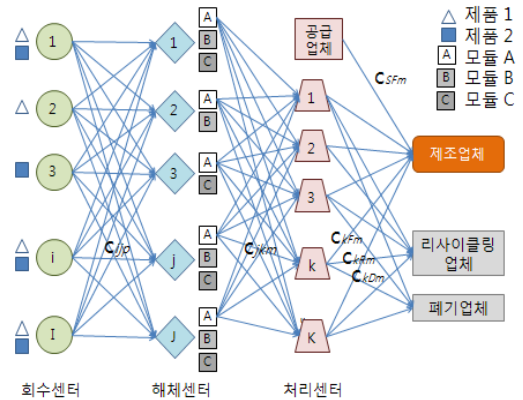
따라서 본 연구에서는 역물류 네트워크 설계 문제를 설정하고, 이 문제에 대하여 협력적 공진화 알고리즘에 기반한 해법을 제시하고, 다양한 테스트 문제를 대상으로 기존의 유전 알고리즘 기반 해법과의 비교를 통해 본 해법의 성능을 보이고자 한다. 제 2장에서는 역물류 네트워크 설계 문제의 수리 모형을 제시한다. 제 3장에서는 본 문제에 대한 협력적 공진화 알고리즘의 세부 내용을 설계한다. 제 4장에서는 테스트 문제들을 만들어 모집단의 크기와 생성 세대수를 변화시켜 가면서 본 연구에서 제시하는 협력적 공진화 알고리즘과 기존의 유전 알고리즘을 반복 실행시키고 그 결과를 분석한다. 마지막으로 제 5장에서 결론을 맺는다.

2. 역물류 네트워크 설계 모형

본 연구에서는 다양한 형태의 모형 가운데 비교적 일반적인 형태의 역물류 네트워크 설계 모형을 설정하고, 이 모형에 대하여 협력적 공진화 알고리즘의 설계 내용을 소개한다. 본 문제에서 대상으로 하는 제품은 제조업체에서 제조되어 도매업체와 소매업체의 순방향 네트워크를 거쳐 최종소비자에게 전달된다. 반면 이 제품의 역물류는 순방향 네트워크와는 다른 별도의 네트워크를 거쳐 원래의 제조업체로 회수된다. 본 문제의 역물류 네트워크에서는 <그림 2>에 나타난 것처럼 복수 개의 회수센터, 해체센터 및 처리센터와 하나씩의 제조업체, 공급업체, 리사이클링업체 및 폐기업체가 공급 사슬을 형성한다. 먼저 각 회수센터는 사용된 제품을 사용자들로부터 회수한다. 회수된 각 제품은 각 회수센터로부터 지정된 하나의 해체센터로 수송된다. 이것은 1단계 수송에 해당한다. 각 해체센터에서는 수송되어 온 각 제품이 제품유형별로 BOM(bill of material) 정보에 따라 여러 유형의 모듈들로 해체된다. 해체된 각 모듈은 모듈 유형별로 각 해체센터로부터 지정된 하나의 처리센터로 수송된다. 이것은 2단계 수송에 해당한다. 각 처리센터에서는 수송되어 온 모듈들을 재처리한다. 재처리된 모듈들은 제조업체의 수요를 충족시키기 위하여 우선적으로 제조업체에 할당된다. 만약 모든 처리센터에서의 특정 모듈 유형의 공급량의 합이 제조업체에서의 수요량보다 작은 경우 그 부족량은 외부의 공급업체로부터 조달받는다. 반대로 특정 모듈 유형의 공급량의 합이 그 제조업체에서의 수요량을 초과할 경우 그 초과량은 저장능력의 범위 안에서 리사이클링업체 및 폐기업체에 순차적으로 할당된다. 이것은 3단계 수송에 해당한다.

본 문제는 이러한 역물류 공급 사슬의 환경 속에서 각 센터 및 업체들의 제품(또는 모듈)의 공급량, 수요량 및 저장능력에 관한 제약조건들을 만족하면서 총 수송비용을 최소화시키는 역물류 네트

워크를 설계하는 것이다. 즉, 각 센터 또는 업체 간의 수송량과 사용할 해체센터 및 처리센터들을 결정하는 것이다.



<그림 2> 역물류 공급 사슬 모형

본 문제의 수리적 모형을 구축하기 위하여 다음과 같은 가정, 인덱스, 파라미터 및 결정변수를 사용한다.

가정 :

- 1) 각 센터 및 업체에서의 제품 또는 모듈의 재고량은 고려하지 않는다.
- 2) 제조업체에서의 모듈에 대한 수요량은 사전에 주어진다.
- 3) 회수센터, 분해센터 및 처리센터의 공급량 및 저장 능력은 사전에 주어진다.
- 4) 회수된 제품은 회수센터와 분해센터 사이에서는 제품 형태로 수송되고, 분해센터와 처리센터 사이에는 모듈 형태로 수송된다.
- 5) 회수되는 모듈의 유형 및 수량은 회수된 제품의 BOM 정보에 따라 결정된다. 상이한 유형의 제품들에서 동일한 유형의 모듈이 회수될 수 있다.
- 6) 각 회수센터, 분해센터 및 처리센터는 다양한 유형의 제품 또는 모듈을 취급할 수 있다.
- 7) 제품에서 회수된 모듈은 모두 재사용이 가능한 정도의 품질 수준을 가지고 있다.

인덱스 :

- I : 회수센터의 집합
- J : 해체센터의 집합
- K : 처리센터의 집합
- P : 제품 유형의 집합
- M : 모듈 유형의 집합
- i : 회수센터($i \in I$)
- j : 분해센터($j \in J$)
- k : 처리센터($k \in K$)
- p : 제품($p \in P$)
- m : 모듈($m \in M$)
- F : 제조업체
- R : 리사이클링업체
- D : 폐기업체
- S : 공급업체

- x_{jkm} : 해체센터 j 로부터 처리센터 k 로의 모듈 m 의 수송량
- x_{kFm} : 처리센터 k 로부터 제조업체 F 로의 모듈 m 의 수송량
- x_{kRm} : 처리센터 k 로부터 리사이클링업체 R 로의 모듈 m 의 수송량
- x_{kDm} : 처리센터 k 로부터 폐기업체 D 로의 모듈 m 의 수송량
- y_{sFm} : 공급업체 S 로부터 제조업체 F 로의 모듈 m 의 수송량
- z_j : 0-1 변수(해체센터 j 의 개설 시에는 1의 값을, 그렇지 않은 경우에는 0의 값을 취함)
- w_k : 0-1 변수(처리센터 k 의 개설 시에는 1의 값을, 그렇지 않은 경우에는 0의 값을 취함)

파라미터 :

- a_{ip} : 회수센터 i 의 제품 p 의 공급능력
- b_{jm} : 해체센터 j 의 모듈 m 의 저장능력
- u_{km} : 처리센터 k 의 모듈 m 의 저장능력
- u_{Rm} : 리사이클링업체 R 의 모듈 m 의 저장능력
- d_m : 제조업체 F 의 모듈 m 에 대한 수요량
- c_{ijp} : 회수센터 i 로부터 해체센터 j 로의 제품 p 의 단위당 수송비용
- c_{jkm} : 해체센터 j 로부터 처리센터 k 로의 모듈 m 의 단위당 수송비용
- c_{kFm} : 처리센터 k 로부터 제조업체 F 로의 모듈 m 의 단위당 수송비용
- c_{kRm} : 처리센터 k 로부터 리사이클링업체 R 로의 모듈 m 의 단위당 수송비용
- c_{kDm} : 처리센터 k 로부터 폐기업체 D 로의 모듈 m 의 단위당 수송비용
- J_{\max} : 개설이 허용되는 해체센터의 최대수
- K_{\max} : 개설이 허용되는 처리센터의 최대수

이 문제의 수리적 모형은 다음과 같이 주어진다.

$$\begin{aligned} \text{Minimize } Z_1 = & \sum_{i \in I} \sum_{j \in J} \sum_{p \in P} c_{ijp} x_{ijp} + \sum_{j \in J} \sum_{p \in P} c_{jp} z_{jp} \\ & + \sum_{j \in J} \sum_{k \in K} \sum_{m \in M} c_{jkm} x_{jkm} + \sum_{k \in K} \sum_{m \in M} c_{km} w_{km} \\ & + \sum_{k \in K} \sum_{m \in M} c_{kFm} x_{kFm} + \sum_{k \in K} \sum_{m \in M} c_{kRm} x_{kRm} \\ & + \sum_{k \in K} \sum_{m \in M} c_{kDm} x_{kDm} + \sum_{m \in M} c_{sFm} y_{sFm} \end{aligned} \quad (0)$$

$$\text{subject to } \sum_{j \in J} x_{ijp} \leq a_{ip} \quad \forall i, p \quad (1)$$

$$\sum_{k \in K} x_{jkm} \leq b_{jm} z_j \quad \forall j, m \quad (2)$$

$$\sum_{k \in K} (x_{kFm} + x_{kRm} + x_{kDm}) \leq u_{km} w_k, \quad \forall m \quad (3)$$

$$y_{sFm} + \sum_{k \in K} x_{kFm} \geq d_m \quad \forall m \quad (4)$$

$$\sum_{j \in J} z_j \leq J_{\max} \quad (5)$$

$$\sum_{k \in K} w_k \leq K_{\max} \quad (6)$$

$$x_{ijp}, x_{jkm}, x_{kFm}, x_{kRm}, x_{kDm}, y_{sFm} \geq 0, \quad \forall i, j, k, p, m \quad (7)$$

$$z_j = \{0, 1\} \quad \forall j \quad (8)$$

$$w_k = \{0, 1\} \quad \forall k \quad (9)$$

결정변수 :

- x_{ijp} : 회수센터 i 로부터 해체센터 j 로의 제품 p 의 수송량

여기서 목적함수를 나타내는 식 (0)은 사용된 제품의 회수로부터 시작하여 재사용가능한 모듈이

될 때까지 발생하는 세 단계별 수송비용 및 해체센터와 처리센터의 개설비용(고정비)의 합을 의미한다. 식 (1)~식 (3)은 각각 회수센터, 해체센터 및 처리센터의 공급량 또는 저장능력의 제한을 의미한다. 식 (4)는 처리센터 및 공급업체에서의 각 유형별 모듈 공급량은 제조업체의 수요를 충족시켜야 함을 의미한다. 식 (5)와 식 (6)은 각각 개설이 허용된 해체센터 및 처리센터 수의 상한을 나타낸다. 식 (7)은 결정변수들의 비음조건을 나타낸다. 식 (8)과 식 (9)는 해당 결정변수들의 이진 제약을 나타낸다.

3. 협력적 공진화 알고리즘의 설계

협력적 공진화 알고리즘은 복잡도가 높은 문제를 2개 이상의 부분문제로 분할하여 전체적으로

- 단계 1 : 각 부분문제에 대응하는 염색체들의 초기 모집단 P_1 과 P_2 를 생성한다.
- 단계 2 : 초기 모집단 P_1 과 P_2 에 대하여 적합도 평가 시에 필요한 초기 협업자 c_1 과 c_2 를 정한다.
- 단계 3 : 종료 조건이 만족될 때까지 다음 단계들(3.1-3.3)을 반복한다.
- 3.1 모집단 P_1 의 각 개체 i 에 대하여
- 3.1.1 개체 i 와 협업자 c_2 를 조합하여 전체 문제에 대한 하나의 해 S_1 을 생성한다.
- 3.1.2 해 S_1 을 평가함으로써 개체 i 의 적합도를 구한다.
- 3.2 모집단 P_2 의 각 개체 i 에 대하여
- 3.2.1 개체 i 와 협업자 c_1 을 조합하여 전체 문제에 대한 하나의 해 S_2 를 생성한다.
- 3.2.2 해 S_2 을 평가함으로써 개체 i 의 적합도를 구한다.
- 3.3 모집단 P_1 에서 적합도가 가장 높은 개체를 c_1 로 둔다.
모집단 P_2 에서 적합도가 가장 높은 개체를 c_2 로 둔다.
- 3.4 유전 연산자를 적용하여 모집단 P_1 과 P_2 의 개체들을 갱신한다.
- 단계 4 : 지금까지 생성된 해 가운데 가장 좋은 해를 선택한다.

<그림 3> 협력적 공진화 알고리즘의 기본적 수행 절차

해의 탐색 공간의 범위를 줄임으로써 해의 탐색 효율을 높이고자 하는 해의 탐색기법이다. 자연계에서 다양한 종들 사이에 상호작용하는 매커니즘이 존재하듯이, 복잡도가 높은 문제를 다루는 협력적 공진화 알고리즘에서도 성격이 다른 여러 부분문제들 사이에 유기적으로 상호작용 할 수 있도록 하는 모집단의 구조와 진화 매커니즘이 요구된다 (Potter[24]). 여기서는 이러한 협력적 공진화 알고리즘의 여러 세부적인 사항들을 설계한다.

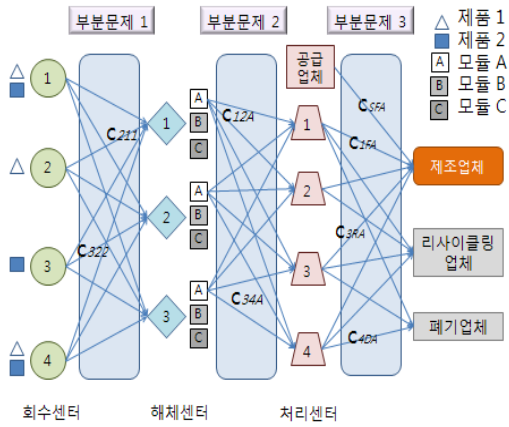
3.1 협력적 공진화 알고리즘의 기본적 수행절차

두 개의 부분 문제로 분할되는 문제의 경우 협력적 공진화 알고리즘의 기본적 수행 절차를 <그림 3>과 같이 표현할 수 있다. 지금부터는 이 절차에 따라 본 문제에 대하여 부분문제의 생성, 부분문제에 대한 해의 생성, 염색체의 적합도 평가, 그리고 모집단의 세대교체를 위해 적용할 유전 연산자 등을 설계한다.

3.2 부분문제의 생성

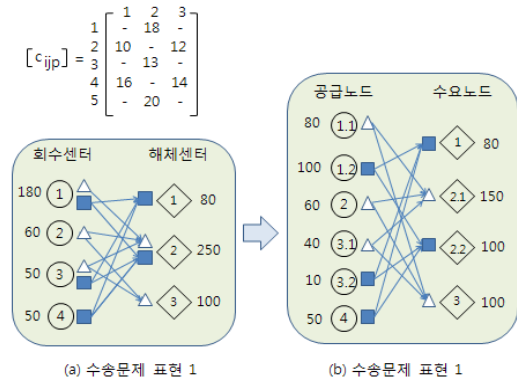
3.2.1 부분문제로의 분할

우선 본 문제를 세 개의 부분문제로 분할한다. 예를 들어, 4개의 회수센터, 3개의 해체센터, 4개의 처리센터 및 하나씩의 제조업체, 리사이클링업체, 폐기업체, 공급업체로 구성되어 있는 공급사슬 문제를 생각해 보자. 이 문제를 수송단계에 따라 <그림 4>와 같이 3개의 부분문제(부분문제 1, 2 및 3)로 분할할 수 있다. 부분문제 1은 각 회수센터로부터 각 분해센터로 각 유형별 제품들(제품 1과 제품 2)의 수송량을 결정하는 것이다. 해체센터에서 각 제품이 해체되어 여러 유형의 모듈이 생성된다. 부분문제 2는 해체된 모듈들을 각 해체센터로부터 각 처리센터로 할당하는 것이다. 그리고 부분문제 3은 각 처리센터 및 공급업체로부터 제조업체, 리사이클링업체 및 폐기업체로 이 우선순위에 따라 모듈들을 순차적으로 할당하는 것이다.



〈그림 4〉 역물류 문제의 부분문제로의 분할

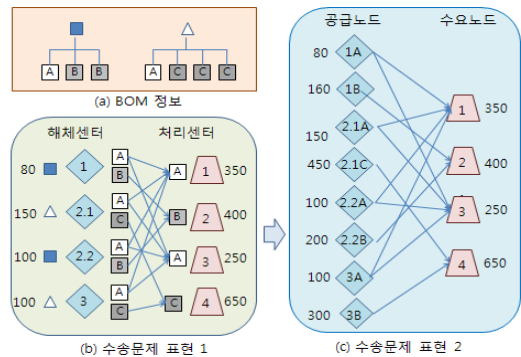
1) 부분문제 1 : 하나의 회수센터나 해체센터는 일반적으로 여러 유형의 제품을 취급한다. 이러한 상황을 기본적인 수송 모형으로 나타내기 위하여 〈그림 4〉에서의 회수센터나 해체센터를 나타내는 각 노드를 〈그림 5〉(b)와 같이 그 곳에서 취급하는 제품 유형의 개수에 해당하는 수의 공급 노드 또는 수요노드들로 분할하여 표현한다. 〈그림 5〉(a)에서는 두 유형의 제품을 기호(△과 ■)로 표현하였다. 회수센터 1과 3은 두 유형의 제품(△과 ■)을, 그리고 회수센터 2와 4는 각각 한 유형의 제품만을 공급할 수 있음을 나타낸다. 따라서 〈그림 5〉(b)에서 회수센터 1은 공급노드 1.1과 1.2의 두 개의 노드로, 그리고 회수센터 3은 두 개의 공급노드



〈그림 5〉 부분문제 1의 생성

3.1과 3.2로 분할하여 표현하였다. 〈그림 5〉(a)에서 해체센터 2는 두 유형의 제품을 다룰 수 있기 때문에, 이것을 〈그림 5〉(b)에서는 두 개의 수요노드 2.1과 2.2로 분할하여 표현하였다.

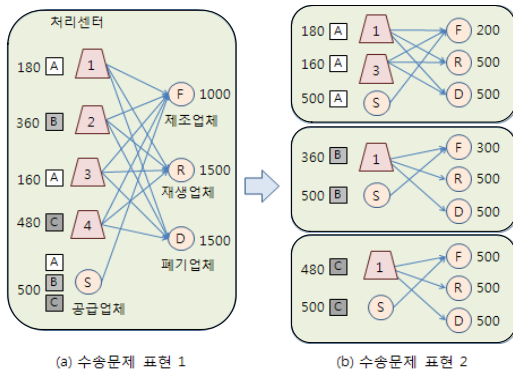
2) 부분문제 2 : 여기서는 각 해체센터로부터 각 처리센터로의 모듈의 수송량을 결정해야 한다. 각 해체센터에서의 모듈 공급량은 보유한 제품의 수량과 BOM 정보를 바탕으로 산출된다. 예를 들어, 〈그림 6〉(a)의 BOM 정보로부터 제품 ■는 1개의 모듈 A와 2개의 모듈 B로 구성되고, 제품 △는 1개의 모듈 A와 3개의 모듈 C로 구성됨을 알 수 있다. 따라서 〈그림 6〉(b)의 해체센터 1의 경우 80개의 제품 ■로부터 80개의 모듈 A와 160개의 모듈 B가 생성되어, 해체센터 1의 모듈 공급능력이 결정된다. 〈그림 5〉에서처럼, 여기에서도 하나의 해체센터나 처리센터를 그 센터에서 취급하는 모듈 유형의 개수만큼의 노드들로 분할하여 표현한다. 따라서 〈그림 6〉(b)의 해체센터 1은 〈그림 6〉(c)에서 두 개의 공급노드 1A와 1B로 표현되고, 해체센터 2.1은 두 개의 공급노드 2.1A와 2.1C로 표현된다. 나머지 센터들도 같은 방식으로 표현된다.



〈그림 6〉 부분문제 2의 생성

3) 부분문제 3 : 〈그림 7〉(a)에서 공급노드는 처리센터들과 하나의 공급업체를 나타낸다. 각 처리센터의 모듈 공급능력은 부분문제 2의 할당 결과에 따라 정해지며, 공급업체의 모듈 공급능력은 사전에 주어진다. 처리센터 또는 공급업체가 여러 유

형의 모듈을 취급한다고 하면 부분문제 1과 2에서 처럼 처리센터 또는 공급업체를 모듈 유형의 수에 해당하는 공급 노드들로 분할하여 표현할 수 있다. <그림 7>(a)에서 각 처리센터는 A, B, C 모듈 중 한 유형의 모듈만을 취급하며, 공급업체는 모든 모듈을 취급하는 것을 나타낸다. <그림 7>(a)의 세 유형의 모듈 할당 문제를 <그림 7>(b)에서처럼 각 모듈별로 할당문제를 분할하여 표현할 수 있다. 본 부분문제의 해는 협력적 공진화 알고리즘 대신, 각 모듈별로 Lee et al.[15]에 의해 제시된 휴리스틱 알고리즘을 적용하여 해를 구하기로 한다. 이 알고리즘의 내용은 뒤에서 소개한다.



<그림 7> 부분문제 3의 생성

3.2.2 부분문제에 대한 염색체 표현

부분문제 3의 해는 휴리스틱 기법에 의해 구하기 때문에 염색체 표현방법을 필요로 하지 않는다. 따라서 부분문제 1과 2에 대해서만 그에 대응하는 모집단내 각 염색체의 표현방법을 정할 필요가 있다. 각 염색체의 값을 기반으로 하158여 그에 대응하는 해를 구하기 때문에, 염색체의 표현방법은 최적화 알고리즘의 성능에 절대적인 영향을 미친다. 부분문제 1과 2의 모집단내 각 개체에 대하여 Gen et al.[8]에 의하여 제시된 우선순위 기반의 표현방법을 채택하여 사용키로 한다.

하나의 염색체는 여러 유전인자(gene)들로 구성된다. 각 유전인자는 해당 염색체 내부에서의 상대적 위치 및 그 유전인자가 취하는 값이라는 두 가

지 요소를 지닌다. 부분문제 1의 경우 첫째, 각 유전인자의 위치는 <그림 5>(b)에 있는(공급노드 또는 수요노드 중) 하나의 노드를 의미하도록 정한다. 따라서 하나의 염색체내 유전인자의 개수는 공급노드의 수(m)와 수요노드의 수의 합(n)에 해당한다. 둘째, 각 유전인자의 값은 1부터 $m+n$ 까지 정수들의 임의의 하나의 순열(permutation)로부터 정해진다. 이 값들은 그 염색체 내 여러 유전인자들 사이의 상대적 우선순위로 사용된다. 부분문제 2의 경우에도 부분문제 1과 똑같은 방법으로 각 유전인자의 수와 값을 정하도록 한다. 예를 들어, <그림 8>에서 6개의 공급노드와 4개의 수요노드로 구성된 부분문제 1에 대하여 모집단 내 하나의 염색체를 1부터 10까지의 정수로 된 하나의 순열 형태로 표현하였다. 그리고 8개의 공급노드와 4개의 수요노드로 구성된 부분문제 2에 대하여 모집단 내 하나의 염색체를 1부터 12까지의 정수로 된 하나의 순열 형태로 표현하였다.



<그림 8> 부분문제 1과 2에 대한 염색체 표현

3.3 부분문제에 대한 해의 생성

앞의 세 가지 부분문제 중 부분문제 1과 2는 공급노드 및 수요노드의 개수와 노드의 의미만 다를 뿐 사실상 같은 유형의 문제로서 같은 해법을 적용할 수 있다. 여기서는 먼저 부분문제 1이나 2에서 하나의 염색체 값이 주어졌을 때 이 값에 대응하는 부분해를 구할 수 있는 알고리즘을 소개한다. 부분문제 3은 수요 노드들 사이에 우선순위가 존재한

다는 면에서 부분문제 1이나 2와는 문제의 성격이 다소 틀린다. 이 부분문제 3에 대하여 채택한 휴리스틱 해법은 그 다음에 소개한다.

3.3.1 부분문제 1에 대한 해의 생성

부분문제 1의 염색체 모집단이 주어졌을 때 그 하나의 염색체에 대응하는 해를 구할 수 있는 알고리즘이 필요하다. Gen et al.[8]의 우선순위 기반 알고리즘을 기반으로 하여, <그림 9>와 같은 알고리즘을 만들어 사용키로 한다.

<p>입력물</p> <p>a_i : 공급노드 i에서의 해당 제품의 공급능력 $\forall i \in I \times P$</p> <p>b_j : 수요노드 j에서의 해당 제품의 수용능력 $\forall j \in J \times P$</p> <p>c_{ij} : 공급노드 i로부터 수요노드 j로의 해당 제품의 단위당 수송비 $\forall i \in I \times P, \forall j \in J \times P$</p> <p>$v_1(i+j)$: 염색체의 값 $\forall i \in I \times P, \forall j \in J \times P$</p> <p>출력물</p> <p>$x_{ij}$: 공급노드 i로부터 수요노드 j로의 제품 수송량 $\forall i \in I \times P, \forall j \in J \times P$</p> <p>절차</p> <p>단계 1 : $x_{ij} \leftarrow 0 \forall i \in I \times P, \forall j \in J \times P$</p> <p>단계 2 : $l \leftarrow \max \{ I P + J P \}$; 하나의 노드를 선택한다.</p> <p>단계 3 : $l \leq I P$이면, $i^* \leftarrow l$; 하나의 공급노드를 선택한다. $j^* \leftarrow \min \{ j v_1(j) \neq 0, j \in J \times P \}$; 최저비용의 수요노드를 선택한다. 그렇지 않으면 ($l > I P$), $j^* \leftarrow l$; 하나의 수요노드를 선택한다. $i^* \leftarrow \min \{ i v_1(i) \neq 0, i \in I \times P \}$; 최저비용의 공급노드를 선택한다.</p> <p>단계 4 : $x_{i^*j^*} \leftarrow \min \{ a_{i^*}, b_{j^*} \}$; 할당량을 정한다. 공급노드(i^*) 및 수요노드(j^*)의 할당 가능량을 다음과 같이 수정한다. $a_{i^*} = a_{i^*} - x_{i^*j^*}, b_{j^*} = b_{j^*} - x_{i^*j^*}$</p> <p>단계 5 : $a_{i^*} = 0$이면, $v_1(i^*) = 0$ $b_{j^*} = 0$이면, $v_1(I P + j^*) = 0$</p> <p>단계 6 : $v_1(i) = 0 \forall i \in I \times P$ 이면, x_{ij}를 구한다. 그렇지 않으면, 단계 1로 돌아간다.</p>

<그림 9> 부분문제 1의 염색체로부터 해의 생성 알고리즘

이 알고리즘에 의해 지정된 하나의 공급노드와 하나의 수요노드 사이에 수송량 할당을 의미하는 아크(arc)가 순차적으로 하나씩 추가됨으로써, 결과적으로 하나의 1단계 수송 네트워크가 완성된다.

3.3.2 부분문제 2에 대한 해의 생성

부분문제 2의 염색체 모집단이 주어졌을 때, 하나의 염색체에 대응하는 해를 구할 수 있는 알고리즘의 내용은 기본적으로 <그림 9>의 알고리즘과 같다. 다만 수송대상이 제품대신 모듈이며, 알고리

<p>입력물</p> <p>b_j : 공급노드 j에서의 해당 모듈의 공급능력 $\forall j \in J \times M$</p> <p>u_k : 수요노드 k에서의 해당 모듈의 수용능력 $\forall k \in K \times M$</p> <p>c_{jk} : 공급노드 j로부터 처리센터 k로의 해당 모듈의 단위당 수송비용 $\forall j \in J \times M, \forall k \in K \times M$</p> <p>$v_2(j+k)$: 염색체 값 $\forall j \in J \times M, \forall k \in K \times M$</p> <p>출력물</p> <p>$x_{jk}$: 공급노드 j로부터 수요노드 k로의 모듈 수송량 $\forall j \in J \times M, \forall k \in K \times M$</p> <p>절차</p> <p>단계 1 : $x_{jk} \leftarrow 0, \forall j \in J, \forall k \in K$</p> <p>단계 2 : $l \leftarrow \max \{ J M + K M \}$; 하나의 노드를 선택한다.</p> <p>단계 3 : $l \leq J M$이면, $j^* \leftarrow l$; 최소비용을 지닌 하나의 해체센터를 선택한다. $k^* \leftarrow \min \{ k v_2(k) \neq 0, k \in K \times M \}$; 최저비용의 공급노드를 선택한다. 그렇지 않으면 ($l > J M$), $k^* \leftarrow l$; 하나의 수요노드를 선택한다. $j^* \leftarrow \min \{ j v_2(j) \neq 0, j \in J \times M \}$; 최저비용의 공급노드를 선택한다.</p> <p>단계 4 : $x_{j^*k^*} \leftarrow \min \{ a_{j^*}, \mu_{k^*} \}$; 할당량을 정한다. 공급지(i^*) 및 수요지(j^*)의 할당 가능량을 다음과 같이 수정한다. $b_{j^*} = b_{j^*} - x_{j^*k^*}, \mu_{k^*} = \mu_{k^*} - x_{j^*k^*}$</p> <p>단계 5 : $b_{j^*} = 0$이면, $v_2(j^*) = 0$ $\mu_{k^*} = 0$이면, $v_2(J M + j^*) = 0$</p> <p>단계 6 : $v_2(j) = 0 \forall j \in J \times M$ 이면, x_{jk}를 구한다. 그렇지 않으면, 단계 1로 돌아간다.</p>

<그림 10> 부분문제 2의 염색체로부터 해의 생성 알고리즘

즘의 수행에 필요한 공급노드들의 공급량이 부분 문제 1의 결과에 의존한다는 면에서 차이가 있다. 이 점들을 감안하여 부분문제 2에 대한 해의 생성 알고리즘을 <그림 10>과 같이 나타낼 수 있다.

3.3.3 부분문제 3에 대한 해의 생성

부분문제 3에 대한 해를 구하기 위하여 Lee et al.[15]이 제시한 <그림 11>의 휴리스틱 알고리즘을 사용하기로 한다. 각 모듈 유형별로 이 알고리즘을 한 번씩 적용함으로써 부분문제 3의 해를 구할 수 있다.

입력물

- u_{km} : 처리센터 k 에서의 모듈 m 의 공급능력
 $\forall k \in K$
- u_R : 리사이클링업체 R 에서의 각 모듈의 수용능력
- d_m : 제조업체 F 에서의 모듈 m 의 수요량
- c_{kFm} : 처리센터 k 로부터 제조업체 F 로의 모듈 m 의 단위당 수송비용
 $\forall i \in I, \forall j \in J, \forall m \in M$
- c_{kRm} : 처리센터 k 로부터 리사이클링업체 R 로의 모듈 m 의 단위당 수송비용
- c_{kDm} : 처리센터 k 로부터 폐기업체 D 로의 모듈 m 의 단위당 수송비용
- c_{SFm} : 공급업체 S 로부터 제조업체 F 로의 모듈 m 의 단위당 수송비용
- x_{jkm} : 해체센터 j 로부터 처리센터 k 로의 모듈 m 의 수송량

출력물

- x_{kFm} : 처리센터 k 로부터 제조업체 F 로의 모듈 m 의 수송량
- x_{kRm} : 처리센터 k 로부터 리사이클링업체 R 로의 모듈 m 의 수송량
- x_{kDm} : 처리센터 k 로부터 폐기업체 D 로의 모듈 m 의 수송량
- y_{SFm} : 공급업체 S 로부터 제조업체 F 로의 모듈 m 의 수송량

절차

단계 1 : 모듈 유형별로 각 해체센터로부터 각 처리센터로의 총수송량 X_m 을 구한다.

$$X_m = \sum_{j \in J} \sum_{k \in K} x_{jkm} \quad \forall m$$

단계 2 : 각 모듈 유형별로 다음 조건에 따라 경우 1, 2, 3 중 하나를 진행한다.

경우 1 : $d_m > X_m$ 이면 단계 3으로 간다.

경우 2 : $d_m = X_m$ 이면 단계 4로 간다.

경우 3 : $d_m < X_m$ 이면 단계 5로 간다.

단계 3 : 각 처리센터로부터 나오는 모든 모듈(X_m)을 제조업체 F 에게 공급한다.

제조업체의 부족량($d_m - X_m$)은 공급업체에서 새 모듈로 공급받는다. 즉,

$$y_{sFm} = d_m - X_m$$

단계 4 : 각 처리센터로부터 나오는 모든 모듈(X_m)을 제조업체 F 에게 공급한다.

단계 5 : 수송비용이 가장 낮은 처리센터로부터 시작하여 순차적으로 모두 X_m 개의 모듈을 제조업체 F 에 우선적으로 할당한다.

그 후 나머지 ($X_m - d_m$)개의 모듈이 리사이클링업체의 수용능력 안에 있으면, 그 모듈들을 모두 리사이클링업체에 할당한다. 그렇지 않으면, u_R 개의 모듈은 우선적으로 리사이클링업체에 할당하고, 나머지 ($X_m - d_m - u_R$)개의 모듈은 모두 폐기업체에 할당한다.

단계 6 : x_{kFm} , x_{kRm} , x_{kDm} 및 y_{sFm} 의 값을 구한다.

<그림 11> 부분문제 3에 대한 해의 생성 알고리즘 (Lee et al.[15])

3.3.4 부분문제들 사이의 상호관계

지금까지 부분문제 1, 2, 3의 해를 생성할 수 있는 알고리즘을 각각 <그림 9>, <그림 10> 및 <그림 11>과 같이 소개하였다. 이 각각의 알고리즘이 실행되기 위해서는 다른 부분문제의 실행결과로부터 나오는 데이터를 입력 데이터로 사용해야하는 상호의존성을 지니고 있다. 먼저 부분문제 1에 대한 알고리즘(<그림 9>)의 실행에 필요한 입력 데이터는 사전에 모두 주어진다. 이 알고리즘이 실행되면, 각 회수센터로부터 각 해체센터로의 제품 유형별 수송량(x_{ijp})을 출력물로 얻을 수 있다. 이 출력물을 기반으로 하여 부분문제 2에 대한 알고리즘(<그림 10>)의 실행에 필요한 입력 데이터 즉, 각 해체센터의 모듈 유형별 공급량(b_{jm})의 값을 계산할 수 있다. 부분문제 2에 대한 알고리즘이 실행되면, 각 해체센터로부터 처리센터로의 모듈 유형별 수송량(x_{jkm})을 얻을 수 있다. 이 값은 부분문제 3에 대한 알고리즘(<그림 11>)의 실행에 필요한 입력 데이터로 사용된다. 이와 같이 세 가지 부분문제를 부분문제 1, 2, 3의 순으로 순차적으로

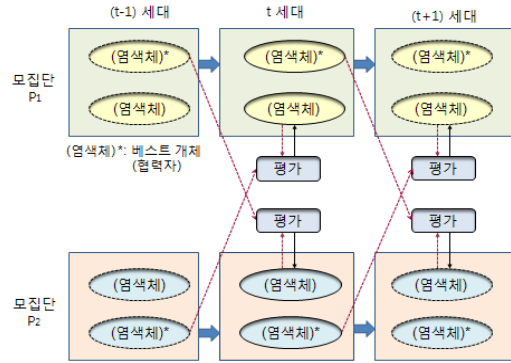
그 해를 구함으로써, 다음 부분문제에 대한 입력 데이터를 제공하고 결과적으로 전체 문제에 대한 해를 구할 수 있게 된다.

3.4 적합도 평가

이제 부분문제 1, 2, 3의 해들을 조합하여 전체 문제의 해를 도출하고, 부분문제 1과 2의 탐색체 모집단내 각 개체의 적합도를 평가해야 한다. 이를 위해 먼저 부분문제 1과 2의 탐색체의 조합에 바탕을 둔 적합도 평가의 원리를 설명하고, 사용할 적합도 함수의 설정 및 제약조건의 처리 방법을 소개하고, 마지막으로 각 개체에 대한 적합도 평가 알고리즘을 제시한다.

3.4.1 적합도 평가 원리

두 개의 모집단으로 구성되는 협력적 공진화 알고리즘에서 하나의 모집단내 개체가 상대 모집단내 개체와 결합될 때 비로소 전체 문제에 대한 적합도를 구할 수 있다. 일반적으로 적합도에는 상대 모집단과의 상호작용의 효과도 포함된다. <그림 12>에서 t 세대에서 모집단 P_1 내 각 개체를($t-1$) 세대에서 모집단 P_2 안에서 가장 좋은 적합도를 보였던 개체(베스트 개체)와 조합함으로써, 모집단 P_1 내 각 개체의 적합도를 계산한다. 그리고 t 세대에서 가장 좋은 적합도를 지니는 모집단 P_1 내 개체(베스트 개체)는 ($t+1$) 세대에서 모집단 P_2 내 개체를 평가할 때 모집단 P_1 의 협력자(collaborator) (c_1)로 사용하기 위하여 그 값을 보관된다. 마찬가지로, t 세대에서 모집단 P_2 의 각 개체를($t-1$) 세대에서 모집단 P_1 안에서 가장 좋은 적합도를 보였던 개체(베스트 개체)와 조합함으로써, 모집단 P_2 내 각 개체의 적합도를 계산한다. 그리고 t 세대에서 가장 좋은 적합도를 지니는 모집단 P_2 내 개체(베스트 개체)는 ($t+1$) 세대에서 모집단 P_1 내 개체를 평가할 때 모집단 P_2 의 협력자(c_2)로 사용하기 위하여 그 값을 보관된다.



<그림 12> 개체들의 조합에 의한 적합도 평가 원리

3.4.2 적합도 함수의 설정

적합도 함수는 앞에서 소개한 수리모형의 목적 함수 Z_1 을 이용하여 만들 수 있다.

$$\begin{aligned}
 Z_1 = & \sum_{i \in I} \sum_{j \in J_p \subseteq P} c_{ijp} x_{ijp} + \sum_{j \in J_p \subseteq P} c_{jp} z_{jp} \\
 & + \sum_{j \in J} \sum_{k \in K_m \subseteq M} c_{jkm} x_{jkm} + \sum_{k \in K_m \subseteq M} c_{km} w_{km} \\
 & + \sum_{k \in K_m \subseteq M} c_{kFm} x_{kFm} + \sum_{k \in K_m \subseteq M} c_{kRm} x_{kRm} \\
 & + \sum_{k \in K_m \subseteq M} c_{kDm} x_{kDm} + \sum_{m \in M} c_{SFm} y_{SFm}
 \end{aligned}$$

목적함수 Z_1 의 첫째 항과 둘째 항의 값은 부분 문제 1의 해로부터 구해지고, 셋째 항과 넷째 항의 값은 부분문제 2의 해로부터 구해지고, 마지막 네 항의 값은 부분문제 3의 해로부터 구해진다. 부분 문제 3의 해는 협력적 공진화 알고리즘이 아니라 휴리스틱 알고리즘을 사용해서 구해지기 때문에, 부분문제 3과 관련된 항들이 적합도 함수 안에 포함되면 휴리스틱 알고리즘의 영향으로 인하여 부분문제 1과 2에 대한 해의 탐색 효율이 저하될 수 있다. 따라서 부분문제 3과 관련되는 항들은 적합도 함수에서 배제시키고 적합도 함수를 다음과 같이 설정한다.

$$\begin{aligned}
 Z_2 = & \sum_{i \in I} \sum_{j \in J_p \subseteq P} c_{ijp} x_{ijp} + \sum_{j \in J_p \subseteq P} c_{jp} z_{jp} \\
 & + \sum_{j \in J} \sum_{k \in K_m \subseteq M} c_{jkm} w_{jkm} + \sum_{k \in K_m \subseteq M} c_{km} w_{km}
 \end{aligned}$$

3.4.3 제약 조건의 처리

부분문제 1, 2, 3에 대한 해의 도출 알고리즘은 수리모형의 제약조건들 중 다음의 제약조건 (5)와 식 (6)을 제외한 나머지 제약조건들의 충족을 보장한다.

$$\sum_{j \in J} z_j \leq J_{\max} \quad (5)$$

$$\sum_{k \in K} w_k \leq K_{\max} \quad (6)$$

제약조건 (5)와 식 (6)도 충족시키기 위하여 Goldberg[10]의 페널티 방법(penalty method)을 사용하여 적합도 함수 *Fitness*를 다음과 같이 조정한다. 여기서 M 은 매우 큰 임의의 수를 나타낸다.

$$\begin{aligned} Fitness = & Z_2 + M \cdot \text{Max} \left(\sum_{j \in J} z_j - J_{\max}, 0 \right) \\ & + M \cdot \text{Max} \left(\sum_{k \in K} w_k - K_{\max}, 0 \right) \end{aligned}$$

따라서 만일 모집단 P_1 또는 P_2 의 어떤 개체가 제약조건 (5) 또는 식 (6)을 만족하지 못하면, 적합도 평가 시에 그 개체의 적합도에 큰 페널티를 부여하여 세대변천 과정에서 그 개체가 자연적으로 도태되도록 유도한다.

3.4.4 적합도 평가 알고리즘

이제까지의 설명을 바탕으로 두 모집단 P_1 및 P_2 내 각 개체에 대한 적합도 평가 알고리즘을 <그림 13>과 같이 정리할 수 있다.

단계 1 : 모집단 P_1 (또는 P_2) 내 각 평가대상 개체에 대하여, 모집단 P_2 (또는 P_1)의 이전 세대의 베스트 개체를 협력자 c_2 (또는 c_1)로 사용하여 단계 2~단계 10을 실행한다(초기 세대의 협력자로는 상대 모집단내 임의의 개체를 사용한다).

단계 2 : $z_j \leftarrow 0 \forall j \in J, w_k \leftarrow 0 \forall k \in K$; 사용할

해체센터 및 처리센터의 수의 초기화

단계 3 : <그림 9>의 알고리즘을 실행하여 부분문제 1의 해를 구한다. 제품 유형별 수송량 (x_{ijp})과 1단계 수송비용 C_1 을 구한다.

단계 4 : $\sum_{i \in I} \sum_{p \in P} x_{ijp} \neq 0$ 이면, $z_j \leftarrow 1 \forall j \in J$;

개설될 해체센터의 수의 파악

단계 5 : x_{ijp} 의 값으로부터 부분문제 2의 각 공급 노드의 모듈 공급량(b_j) 값을 계산한다.

단계 6 : 모듈 공급능력(b_j) 값을 입력물로 사용하여 <그림 10>의 알고리즘을 실행하고, 부분문제 2의 해를 구한다. 부품 유형별 수송량 (x_{jkm})과 2단계 수송비용 C_2 를 구한다.

단계 7 : $\sum_{j \in J} \sum_{m \in M} x_{jkm} \neq 0$ 이면, $w_k \leftarrow 1 \forall k \in K$;

개설될 처리센터 수의 파악

단계 8 : 각 모듈 m 에 대하여, x_{jkm} 값을 입력물로 사용하여 <그림 11>의 알고리즘을 실행하고, 부분문제 3의 해 $x_{kFm}, x_{kRm}, x_{kDm}, y_{SFm}$ 와 3단계 수송비용 C_3 를 구한다.

단계 9 : 평가대상 개체의 목적함수 값 $Z_1 (= C_1 + C_2 + C_3)$ 및 적합도 함수 *Fitness*의 값을 계산한다.

단계 10 : 모집단 P_1 (또는 P_2)내 평가 개체들 가운데 최소의 *Fitness* 값을 가진 개체를 베스트 개체로 선택한다. 다음 세대에서 모집단 P_2 (또는 P_1)내 개체들의 적합도를 평가할 때 이 개체를 협력자 c_1 (또는 c_2)으로 사용하기 위하여 저장한다.

단계 11 : 최소의 Z_1 값을 지닌 평가대상 개체들을 두 모집단 P_1 과 P_2 에서 구한다.

<그림 13> 모집단 P_1 과 P_2 내 개체의 적합도 평가 알고리즘

3.5 모집단의 세대교체를 위한 연산

유전 연산의 유형에는 선택(selection), 교배(crossover) 및 돌연변이(mutation) 연산이 있다. 모집단의 세대교체가 이루어질 때 두 모집단 P_1 및 P_2 는 선택 연산을 통하여 축적된 정보를 이용하여 탐색하고, 교배 및 돌연변이 연산을 통하여 새로운 해의 영역을 탐색한다. 협력적 공진화 알고리즘에서 두 모집단 P_1 및 P_2 는 각각 독립적으로 유전 연산을 적용하여 독립적으로 세대교체를 이루어간다. 본 연구에서 적용한 구체적인 연산자를 소개한다.

3.5.1 선택 연산

선택 연산자로는 크게 랜덤(random), 비례(proportional), 토너먼트(tournament) 및 랭크 기반(rank-

nk-based) 선택 등을 들 수 있다(Engelbrecht[5]). 본 연구에서는 이 중에서 엘리티스트(elitist) 전략을 수반한 이진 토너먼트(binary tournament) 선택 연산(De Jong[4])을 두 모집단 P_1 및 P_2 각각에 사용한다. 이진 토너먼트 선택이란 균등 확률 분포(uniform distribution)를 사용하여 선택 풀에서 2개의 개체를 선택하여, 그 중에서 적합도가 더 높은 개체를 토너먼트의 승자로 지정하여 다음 세대의 개체로 선택하는 방식이다. 엘리티스트 전략이란 선택 과정을 통하여 가장 우수한 염색체가 선택되지 않더라도 새로운 세대에서도 그 염색체를 보존하는 하는 방식으로, 일반적으로 선택 연산의 보조 수단으로 사용된다.

3.5.2 교배 연산

본 문제의 염색체는 순열 형태로 표현된다. 순열 형태의 염색체에 적용할 수 있는 교배 연산으로는 partially mapping crossover(PMX), order crossover(OX), 및 position-based crossover(PX) 등의 연산자를 들 수 있다. PMX와 OX 연산자는 기존 two-cutpoint 연산의 변형이며, PX 연산자는 uniform 연산의 변형이다. 본 연구에서는 두 모집단 모두 범용성이 높은 order crossover(OX) 연산자를 사용한다.

3.5.3 돌연변이 연산

돌연변이 연산은 조기 수렴(premature convergence)(Goldberg[10])을 방지하고 새로운 해 공간을 탐색하기 위하여 일반적으로 염색체 내 유전인자의 값을 수정함으로써 이루어진다. 돌연변이 연산으로는 insert 연산자 및 swap 연산자가 주로 사용된다. insert 연산자에서는 하나의 디지털(digit)을 무작위로 선택한 후, 그것을 무작위로 선택된 새로운 위치에 삽입한다. swap 연산자에서는 두 디지털을 무작위로 선택한 후, 그들의 위치를 상호 교환한다. 본 연구에서는 두 모집단 P_1 및 P_2 에 각각 swap 연산자를 사용한다.

4. 실험

4.1 실험 설정

본 연구에서 제안한 협력적 공진화 알고리즘의 성능을 확인하기 위하여 Lee et al.[15]이 제시한 하이브리드 유전 알고리즘의 성능과 비교하는 방법을 사용한다. 두 알고리즘을 마이크로소프트사의 비주얼 스튜디오 2009 환경에서 비주얼 C++ 언어를 사용하여 객체지향 방식으로 각각 프로그래밍 하였다. 이 두 알고리즘에서 세대 변천시 돌연변이 비율은 모두 $1/(\text{모집단 내 개체수})$ 로 설정하였다.

모두 6개의 테스트 문제를 만들었다. <표 1>에 나타낸 바와 같이 회수센터, 해체센터 및 처리센터의 수를 각각 3에서부터 10, 3에서 10, 그리고 5에서 10까지 변동시켰다. BOM 정보로서 사용되는 제품유형 및 모듈유형의 수는 각각 최대 3으로 제한하였다. 테스트 문제의 데이터는 모두 무작위로 생성되었다. 1단계, 2단계 및 3단계 수송비용은 모두 1과 20사이의 균등 분포(uniform distribution)로부터 생성되었다. 회수센터에서의 제품 공급량, 해체센터 및 유통센터의 수용 능력은 모형의 제약 조건을 감안하여 적절한 평균값을 지니는 균등 분포로부터 설정되었다. 이 문제의 복잡도는 NP-hard 이기 때문에, 규모가 커짐에 따라 최적해를 구하는 것은 불가능하다. 각각의 테스트 문제에 대하여 모집단의 크기와 생성 세대수를 변화시켜 3가지의 시나리오를 설정하고, 각 시나리오에 대하여 두 알고리즘을 각각 10번씩 반복 실행하여 해를 구하였다.

<표 1> 테스트 문제의 크기

문제	회수 센터 수	해체 센터 수	처리 센터 수	제품 유형 수	모듈 유형 수
1	3	3	5	2	3
2	3	4	5	3	2
3	4	5	6	2	3
4	5	6	7	3	3
5	7	7	7	2	3
6	10	10	10	2	3

4.2 실험 결과

두 알고리즘의 성능을 상호 비교하기 위하여 각 알고리즘의 염색체 모집단내 각 평가개체들 가운데 최소 수송비용을 지닌 개체를 비교 기준으로 정하였다. 여기서 수송비용이란 1단계, 2단계 및 3단계 수송비용의 합을 의미한다. 6개의 테스트 문제를 3개의 시나리오별로 협력적 공진화 알고리즘과 유전 알고리즘을 각각 10회씩 반복 실행시켰다. 실험 결과를 <표 2>와 같이 요약할 수 있다. 이 표는 두 알고리즘에 의한 10회 반복 실험 결과를 목적함수 값(즉, 최소 수송비용)의 최소값, 평균값 및 최대값을 기준으로 나타낸 것이다.

문제 1의 경우 최소값, 평균값 및 최대값의 세

기준에서 모두 유전 알고리즘이 협력적 공진화 알고리즘보다 상대적으로 더 좋은 결과를 보였다. 반면, 문제 2와 3의 경우에는 6개의 시나리오 모두에서 두 알고리즘이 비슷한 성능을 보였다. 평균값과 최대값의 기준에서는 협력적 공진화 알고리즘이 상대적으로 더 우수하며, 최소값 기준으로는 유전 알고리즘이 더 우수한 결과를 보였다. 문제의 크기가 상대적으로 더 큰 문제 4, 5, 6의 경우에는 평균값의 기준으로 세 시나리오에서 모두 협력적 공진화 알고리즘이 유전 알고리즘에 비해 모두 더 좋은 결과를 보였으며, 최소값 및 최대값의 기준에서도 협력적 공진화 알고리즘이 절대적으로 더 좋은 결과를 보였다. 전체적으로 문제 1의 경우를 제외하면 공진화 알고리즘이 유전 알고리즘보다 더 나은

<표 2> 협력적 공진화 알고리즘과 유전 알고리즘의 실험 결과

문제	모집단 크기	생성 세대수	최소값		평균값		최대값	
			공진화	유전	공진화	유전	공진화	유전
1	30	300	14395	14395	15790	14621	18062	15207
	50	500	14437	14395	14764	14488	16164	14771
	100	1,000	14437	14395	15260	14424	17848	14437
2	30	300	24165	24165	24167	24167	24183	24181
	50	500	24165	24165	24165	24167	24165	24183
	100	1,000	24165	24165	24165	24167	24165	24183
3	40	400	21411	21344	22447	22160	23442	24500
	80	800	21344	21344	22214	22039	22547	22897
	120	1200	21344	21344	22010	22119	22497	23422
4	75	750	36350	36337	36722	39694	37325	45110
	100	1,000	36142	36484	37240	38587	41127	41679
	150	1,500	36069	35967	36562	39044	36920	40314
5	80	1,000	27422	28250	28273	29148	31413	29701
	100	2,000	27422	28166	28273	28999	28714	30170
	150	3,000	27495	28722	28310	29539	28872	30180
6	100	2,000	29777	30213	31418	32461	33624	35054
	150	3,000	29682	30053	31833	33177	34778	34731
	200	5,000	29694	29963	30610	33140	32857	35948

주) 1) ‘공진화’는 본 연구에서 제안하는 협력적 공진화 알고리즘에 의해 생성된 비용을 나타내며, ‘유전’은 유전 알고리즘에 의해 생성된 비용을 나타낸다.

2) 표시는 두 알고리즘에 의해 생성된 해 가운데 더 좋은 해를 나타낸다.

해를 생성함을 알 수 있다.

사용된 각 문제의 입력 데이터를 살펴본 결과, 문제 1의 경우 1단계 및 2단계 수송비용의 합이 3단계 수송비용보다 작아서 비현실적으로 설정된 반면, 나머지 문제 2, 3, 4, 5, 6의 경우 1단계 및 2단계 수송비용의 합이 3단계 수송비용보다 크게끔 현실적으로 설정되어 있었다. 협력적 공진화 알고리즘의 적합도는 1단계 및 2단계 수송비용의 합을 기준으로 평가된다. 즉, 이 과정에서 휴리스틱 알고리즘으로 구한 3단계 수송문제의 해는 이 적합도 평가에서 제외된다. 따라서 문제 1의 경우 적합도 평가에서 제외된 3단계 수송비용이 1단계 및 2단계의 수송비용의 합보다 더 크기 때문에 공진화 알고리즘의 탐색 효율이 크게 떨어져 유전 알고리즘보다 더 좋지 못한 결과를 보인 것으로 판단된다.

각 문제의 시나리오별로 모집단의 크기와 생성세대수가 증가함에도 불구하고 현저한 해의 개선이 보이지 않는다. 이러한 현상도 3단계 수송문제의 해가 적합도 평가에서 배제됨으로 인해 발생한 것으로 보인다.

협력적 공진화 알고리즘은 초기 세대부터 해를 생성하고, 세대변천에 따라 더 우수한 해를 탐색해내는 특성을 가지고 있기 때문에, 현실세계의 큰 문제를 다루고자 할 때에는 부분문제의 각 모집단의 크기 및 생성세대수를 본 실험에서보다 충분히 더 크게 설정함으로써 만족스러운 해를 도출할 수 있을 것이다.

5. 결 론

본 연구에서는 사용된 제품의 재활용을 위한 역물류 네트워크 설계 문제를 다루었다. 역물류 네트워크 설계 문제의 해를 구하기 위해 그 동안 혼합정수 계획법이 주로 사용되었으며, 휴리스틱 알고리즘도 가끔 제시되었다. 최근에는 유전 알고리즘이 성공적으로 적용되었다. 본 연구에서는 이 문제에 대하여 새로운 접근방법으로서 협업적 공진화 알고리즘의 사용을 제안하고, 이 알고리즘의 설계

내용을 소개하였다. 본 연구의 역물류 문제는 다수의 회수센터로부터 해체센터들로의 1단계 수송문제, 다수의 해체센터들로부터 처리센터들로의 2단계 수송문제, 다수의 처리센터들과 하나의 공급업체로부터 제조업체, 리사이클링업체 및 폐기업체로의 3단계 수송 문제로 구성되어 있다. 1단계는 제품 형태로, 그리고 2단계 및 3단계는 제품이 분해된 모듈 형태로 수송된다. 각 센터는 여러 유형의 제품 또는 모듈을 취급한다. 먼저 1, 2, 3단계 수송문제를 각각 부분문제 1, 2, 3으로 분할하였다. 부분문제 1과 2에 대응하는 모집단의 각 개체(염색체)를 정수의 순열 형식으로 표현하고, 또한 이 염색체로부터 각 부분문제의 해를 구할 수 있는 알고리즘을 제시하였다. 그리고 부분문제 3에 대한 해를 구하기 위하여 별도의 휴리스틱 알고리즘을 사용하였다. 부분문제 1과 2에 대응하는 각 모집단의 하나의 개체로부터 해를 생성하는 알고리즘과 부분문제 3의 휴리스틱 알고리즘을 조합하여 각 평가 개체의 적합도 평가 방법을 정하였다. 일부 제약조건들이 세대변천 과정을 통해 만족되도록 유도하기 위하여 적합도 함수에 페널티를 부가하는 방법을 사용하였다. 모집단내 개체의 선택, 교배 및 돌연변이 연산에 대한 연산자를 지정하였다. 이렇게 설계된 협력적 공진화 알고리즘의 성능을 파악하기 위하여 테스트 문제들을 만들고 반복적인 실험을 통해 기존의 유전 알고리즘의 성능과 비교해보았다. 그 결과, 전체적으로 공진화 알고리즘이 유전 알고리즘보다 더 나은 해를 생성하였다.

본 연구의 협력적 공진화 알고리즘은 역물류 네트워크 설계의 구축에 관한 다양한 수리적 모형 중 하나에 대해서만 설계되었지만, 이 알고리즘의 내용은 그 외의 수리적 모형에도 여전히 적용될 수 있을 것으로 기대된다. 또한 본 연구에서는 하나의 부분문제의 해를 구하는데 휴리스틱 알고리즘을 사용함으로써 협력적 공진화 알고리즘의 탐색 효율이 다소 떨어지는 문제점을 발견하였다. 향후 이 문제점의 해결을 위한 연구가 수행되어야 할 것이다.

참 고 문 헌

- [1] 신경석, 김여근, “다목적 최적화를 위한 공생 진화알고리즘”, 『한국경영과학회지』, 제32권, 제1호(2007), pp.77-91.
- [2] 한용호, “2단계 수송문제에 대한 협력적 공진화 알고리즘 기반의 혁신적 해법”, 『산업혁신 연구』, 제25권, 제4호(2009), 경성대학교 산업개발연구소, pp.227-251.
- [3] Chandra, C. and J. Grabis, *Supply Chain Configuration : Concepts, Solutions and Applications*, Springer, 2007.
- [4] De Jong, K.A., *Evolutionary Computation : A Unified Approach*, MIT Press, 2006.
- [5] Engelbrecht, A.P., *Computational Intelligence*, Second Edition, Wiley, 2007.
- [6] Fleischmann, Moritz, Jacqueline M. Bloemhof-Ruwaard, Rommert Dekker, Erwin van der Laan, Jo A.E.E. van Nunen and Luk N. Van Wassenhove, “Quantitative Models for Reverse Logistics: A Review,” *European Journal of Operational Research*, Vol.103(1997), pp.1-17.
- [7] Franke, C., B. Basdere, M. Ciupek, and S. Seliger, “Remanufacturing of Mobile Phones—Capacity, Program and Facility Adaptation planning,” *Omega*, Vol.34(2006), pp.562-574.
- [8] Gen, Mitsuo, F. Altiparmak, and L. Lin, “A Genetic Algorithm for Two-stage Transportation Problem Using Priority-based Encoding,” *OR Spectrum*, Vol.28(2006), pp.337-354.
- [9] Gen, Mitsuo, Runwei Cheng, and Lin Lin, *Network Models and Optimization : Multi-objective Genetic Algorithm Approach*, Springer, 2008.
- [10] Goldberg, David E., *Genetic Algorithm : in Search, Optimization and machine Learning*, Addison Wesley, 1989.
- [11] Jayaraman, V., R.A. Patterson, and E. Rolland, “The Design of Reverse Distribution Networks : Models and Solution Procedures,” *European Journal of Operational Research*, Vol.150(2003), pp.128-149.
- [12] Kim, Kibum, Iksoo Song, Juyong Kim, and Bongju Jeong, “Supply Planning Model for Remanufacturing System in Reverse Logistics Environment,” *Computers and Industrial Engineering*, Vol.51(2006), pp.279-287.
- [13] Ko, Hyun Jeung and Gerald W. Evans, “A Genetic Algorithm-based Heuristics for The Dynamic Integrated Forward/Reverse Logistics Network for 3PLs,” *Computers and Operations Research*, Vol.34(2007), pp.346-366.
- [14] Kusumasturi, R., R. Piplani, and G. Lim, “An Approach to Design Reverse Logistics Networks for Product Recovery,” Proceedings of IEEE International Engineering Management Conference, Singapore, (2004), pp. 1239-1243.
- [15] Lee, J., Mituo Gen, and Kyong-Gu Rhee, “Network Model and Optimization of Reverse Logistics by Hybrid Genetic Algorithm,” *Computers and Industrial Engineering*, Vol. 56(2009), pp.951-964.
- [16] Lin, Lin, Mituo Gen, Xiaoguang Wang, “Integrated Multistage Logistics Network Design by Using Hybrid Evolutionary Algorithm,” *Computers and Industrial Engineering*, Vol. 56(2009), pp.854-873.
- [17] Lu, Zhiqiang and Nathalie Bostel, “A Facility Location Model for Logistics Systems Including Reverse Flows : The Case of Remanufacturing Activities,” *Computers and Operational Research*, Vol.34(2007), pp.299-323.

- [18] Min, Hokey, Hyun Jeung Ko, Chang Seong Ko, "A Genetic Algorithm Approach to Developing the Multi-echelon Reverse Logistics Network for Product Returns," *Omega*, Vol.34(2006), pp.56-69.
- [19] Mutha, Akshay, and S. Pokharel, "Strategic Network Design for Reverse Logistics and Remanufacturing Using New and Old Product Modules," *Computers and Industrial Engineering*, Vol.56(2009), pp.334-346.
- [20] Park, K., T. Park, and K.R. Ryu, "Planning for Remarshaling in An Automated Container Terminal using Cooperative Coevolutionary Algorithms," Proceedings of the 24th Annual ACM Symposium on Applied Computing 2009(ACM SAC 2009), Honolulu, Hawaii, (2009) pp.1098-1105.
- [21] Pati, Rupesh Kumar, Prem Vrat and Pradeep Kumar, "A Goal Programming Model for Paper Recycling System," *Omega*, Vol.36(2008), pp.405-417.
- [22] Pokharel, Shaligram and Akshay Mutha, "Perspectives in Reverse Logistics," *Resources, Conservation and Recycling*, Vol.53(2009), pp.175-182.
- [23] Potter, M.A. and K.A. De Jong, "A Cooperative Coevolutionary Approach to Function Optimization," PPSN III : Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature. London, UK : Springer-Verlag, (1994), pp.249-257.
- [24] Potter, M. A., *The Design and Analysis of a Computational Model of Cooperative Coevolution*, Ph. D. Dissertation, George Mason University, 1997.
- [25] Rogers, Dale S. and Ronald S. Tibbon-Lembke, *Going Backwards : Reverse Logistics Trends and Practices*, Center for Logistics Management, University of Nevada, Reno, Reverse Logistics Council, 1998.
- [26] Srivastava, Samir K., "Network Design for Reverse Logistics," *Omega*, Vol.36(2008), pp. 535-548.
- [27] Thierry, Martijn, Marc Salomon, Jo Van Nunen and Luk Van Wassenhove, "Strategic Issues in Product Recovery Management," *California Management Review*, Vol.37, No. 2(1995), pp.114-135.