

다목적 최적화를 위한 공생 진화알고리즘*

신경석** · 김여근***†

A Symbiotic Evolutionary Algorithm for Multi-objective Optimization*

Kyoung Seok Shin** · Yeo Keun Kim***

■ Abstract ■

In this paper, we present a symbiotic evolutionary algorithm for multi-objective optimization. The goal in multi-objective evolutionary algorithms (MOEAs) is to find a set of well-distributed solutions close to the true Pareto optimal solutions. Most of the existing MOEAs operate one population that consists of individuals representing the entire solution to the problem. The proposed algorithm has a two-leveled structure. The structure is intended to improve the capability of searching diverse and good solutions. At the lower level there exist several populations, each of which represents a partial solution to the entire problem, and at the upper level there is one population whose individuals represent the entire solutions to the problem. The parallel search with partial solutions at the lower level and the integrated search with entire solutions at the upper level are carried out simultaneously. The performance of the proposed algorithm is compared with those of the existing algorithms in terms of convergence and diversity. The optimization problems with continuous variables and discrete variables are used as test-bed problems. The experimental results confirm the effectiveness of the proposed algorithm.

Keyword : Multi-objective Optimization, Multi-objective Evolutionary Algorithm, Symbiotic EA, Pareto Optimal Solutions, Non-dominated Solutions

논문접수일 : 2006년 09월 25일 논문게재확정일 : 2007년 01월 12일

* 이 논문은 2005년도 전남대학교 학술연구비 지원에 의하여 연구되었음.

** 전남대학교 산학협력단

*** 전남대학교 산업공학과

† 교신저자

1. 서 론

현실 세계의 문제들은 흔히 상충되는 여러 목적들을 갖는다. 이러한 문제의 해결을 다루는 문제를 다목적 최적화(multi-objective optimization)라 부른다. 다목적 최적화 문제의 해결을 위한 대표적 접근방법 중 하나는 사전에 의사결정자의 선호정보를 파악하여 가장 선호하는 하나의 해를 제시하는 방법이고 다른 하나는 의사 결정자의 사전 선호정보 없이 여러 좋은 대안해를 구하여 그 중에서 의사결정자가 선택하도록 하는 방법이다. 전자는 목적들에 가중치를 두거나 효용함수를 사용하는 방법에 해당되고, 후자는 다양한 파레토(Pareto) 최적해(또는 '비지배해'라고도 함)를 구하는 것이 된다. 본 연구에서는 의사결정자에게 다양한 해를 제공하는 후자의 경우를 다룬다. 여기서 파레토 최적해(비지배해)란 모든 목적들에 대해 이 해보다 더 우수한 해가 존재하지 않는 해를 의미한다. 앞으로 용어의 혼동을 피하기 위해 파레토 최적해는 문제가 갖는 이론적인 실제 파레토 최적해를, 비지배해는 구해진 여러 해들 중에서 지배되지 않는 해를 나타내기로 한다.

다목적 문제에서 여러 비지배해를 구하기 위해 하나의 비지배해를 구하는 알고리즘을 반복 수행하는 방법을 사용할 수 있지만, 이 방법은 흔히 해 공간에 넓게 분포되는 다양한 해를 제공하지 못한다[2].

진화알고리즘은 자연의 진화과정을 모방한 확률적 탐색기법으로 여러 개체로 구성된 모집단을 운용하면서 해공간을 탐색한다. 진화알고리즘에서 모집단의 운용은 한 번의 알고리즘 실행으로 여러 비지배해를 구할 수 있게 한다. 이러한 이유로 진화알고리즘은 다목적 최적화문제의 해결에 매우 효과적인 기법으로 인식되고 있으며 이에 관한 많은 연구가 이루어져 왔다[21, 22]. 다목적 문제의 해결을 위한 진화알고리즘을 다목적 진화알고리즘(Multi-Objective Evolutionary Algorithm : MOEA)이라 부른다.

MOEA의 목표는 파레토 최적해 또는 이들 해에 가까운 해들을 다양하게 찾는 것이다. 따라서 MOEA

연구의 초점은 두 가지로, 그 하나는 (1) 구하는 여러 해가 파레토 최적해로의 수렴이고, 다른 하나는 (2) 이들 해들이 잘 분포되도록 하는 것이다[3]. 이는 좋은 대안들을 다양하게 의사결정자에게 제시하는 의미이다. MOEA에 관한 최초의 알고리즘은 VEGA(Vector Evaluated Genetic Algorithm)[19]이다. 이 방법은 각 목적 별로 부분모집단을 운영하여 각 목적 별(부분모집단별) 적응도 평가와 선택(selection), 유전연산을 반복하면서 비지배해를 찾아 나간다. 하지만 이 방법은 구한 해가 다양하지 못하고 각 목적의 극점에 편중된다는 단점을 갖는다. 이러한 단점을 보완하기 위해, Goldberg[8]는 적소(niche)의 개념을 이용하여 주위 개체의 밀집도가 높은 개체는 적응도를 낮게, 주위 밀집도가 낮은 개체는 적응도를 높게 배분하는 방법을 제안하였다 이 개념은 MOGA(Multi-Objective Genetic Algorithm) [7], NSGA(Nondominated Sorting Genetic Algorithm)[20], NPGA(Niched Pareto Genetic Algorithm)[9] 등의 여러 알고리즘에 적용되었다. 이들 연구는 적소 개념에 의해 다양한 모집단을 구하고 유지하는 데 중점을 두고 있다.

한편, 최근에는 엘리티즘(elitism) 전략을 이용하여, 다양한 해와 함께 파레토 최적해로의 수렴속도 향상에 관한 여러 연구가 행하여 졌다. MOEA에서 '엘리티즘 전략'이란 지금까지 발견된 비지배해를 보관하고 운영하는 전략을 의미한다. 이 전략을 채용한 알고리즘으로는 PAES(Pareto-Archive Evolution Strategy)[14], SPEA(Strength Pareto Evolutionary Algorithm)[24], NSGA-II[4], SPEA2[25] 등이 있다. PAES는 진화전략(Evolution Strategy : ES)을 응용한 방법으로, 하나의 부모 개체로 자손을 생산한 후, 생산된 자손을 현재까지 구한 비지배해 집합과의 지배관계 및 밀집거리를 비교하면서 비지배해 집합을 갱신하고, 또한 부모개체와 비교하여 자손을 생산할 부모개체를 갱신한다. 이 과정을 통해 수렴성과 다양성을 동시에 추구하는 방법이다. SPEA는 현재까지 발견된 비지배해들을 외부 모집단(external population)에 유지하여 적응도 평가에 참

여시키는 엘리티즘 전략을 사용하였다. 후에 SPEA2에서 외부 모집단 운영방법과 적응도 평가 방법을 개선하였다. NSGA-II는 NSGA의 많은 계산량에 관한 단점을 보완하고 엘리티즘을 위한 전략으로 $(\mu+\lambda)$ -ES(Evolution Strategy)에서의 선택전략을 사용하였다. $(\mu+\lambda)$ -ES는 μ 개의 개체를 갖는 모집단에서 λ 개의 자손을 생산하여, 모집단과 생산된 자손 개체 즉, $(\mu+\lambda)$ 개에서 가장 좋은 μ 개의 개체를 선택하여 다음 세대 모집단을 구성하는 알고리즘이다. NSGA-II에서는 개체의 파레토 순위를 우선적으로 고려하고 순위가 동일하면 밀집도(crowding distance)를 계산하여 밀집도가 낮은 개체가 자손 생산을 위해 선택되거나, 다음 세대 모집단을 구성하는 개체로 선택된다. 이 외에 Laumanns et al. [15]과 Deb et al.[5]은 ϵ -dominance 개념을 채용한 MOEA를 제안하였다. ϵ -dominance 개념은 각 목적 k 에 대해 적절한 ϵ_k 를 설정하고 ϵ_k 보다 큰 차이를 갖는 개체를 모집단 개체로 허용함으로써 모집단의 다양성 향상을 유도하였다.

기존 MOEA에 관한 연구는 모두 전체문제를 하나의 개체로 표현하여 이들 개체로 이루어진 하나의 모집단을 운영하고 있다. 본 연구에서는 공생 진화알고리즘 기반의 새로운 MOEA를 제안한다. 공생 진화알고리즘은 공생관계를 갖는 서로 다른 종들이 상호작용하고 상호 적응하며 공진화하는 과정을 모방한 확률적 탐색기법으로 복잡도가 높은 최적화문제의 해결에 효율적이라고 알려져 있다[11, 18]. 여기서 복잡하고 동적인 문제를 여러 부분문제로 분해하고, 이들 부분문제를 하나의 종으로 본다. 제안하는 알고리즘은 문제의 해를 하나의 개체로 표현하는 기존 MOEA와는 달리 전체문제를 부분문제로 분해한 후, 부분문제들로 이루어진 여러 부분모집단을 운영하여 다양한 비지배해를 효율적으로 탐색할 수 있도록 한다. 그리고 전체문제의 해를 개체로 표현하는 모집단을 또한 운영하여 비지배해로의 수렴성을 높이고자 한다. 여러 부분모집단의 공생진화과정을 계층 1로 하고, 전체해(entire solution)의 개체로 이루어진 하나의 모집단의 진화 과

정을 계층 2로 한다. 계층 1에서 얻은 좋은 해의 정보가 계층 2로 전달될 수 있는 구조로 설계된다. 따라서 본 연구에서 제안한 알고리즘을 2계층 다목적 공생 진화알고리즘(Two-leveled Multi-objective Symbiotic EA ; TMSEA)이라 부르기로 한다.

실험을 통해 제안하는 TMSEA의 성능을 기존 MOEA의 성능과 비교 분석한다. 실험문제로는 연속변수와 이산변수를 갖는 최적화문제를 사용한다. 성능 분석은 파레토 최적해로의 수렴성과 구한 해의 다양성 측면에서 이루어 진다.

2. TMSEA : 2계층 다목적 공생 진화알고리즘

2.1 개념과 특징

본 연구에서 제안하는 TMSEA의 개념과 그 특징은 다음과 같다.

첫째, TMSEA는 [그림 1]과 같이 2계층 구조로 이루어 진다. 계층 1에서는 부분문제를 나타내는 모집단들로 분리된 종들이 공생하는 과정을 모방한 공생 진화알고리즘이 수행된다. 이들 모집단 개체는 다른 모집단 개체와 짝을 이루어 적응도(fitness)를 부여 받고, 이 적응도에 기초하여 진화한다. 여기서 짝을 이루는 개체들을 공생자(symbionts)라 부른다. 부분문제 간의 공생진화는 병렬탐색을 강화하여 넓은 해공간의 효율적 탐색에 기여할 것으로 본다.

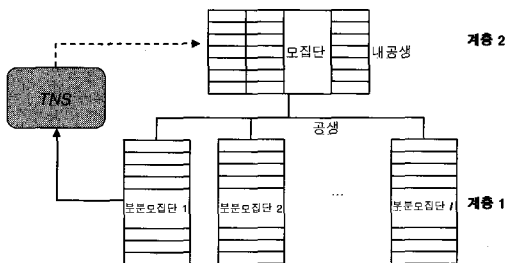
둘째, 계층 2는 전체해를 표현한 개체들로 구성된 하나의 모집단을 운영한다. 계층 2는 계층 1과 상호 작용한다. 계층 1의 진화과정에서 얻은 비지배해를 임시 비지배해 저장소인 TNS에 저장된다. 계층 1의 모든 부분모집단 진화가 끝나면 TNS에서는 계층 1의 한 세대가 진화하는 동안 발견된 비지배개체결합이 저장된다. 저장된 비지배개체결합은 계층 2의 진화 시 내공생자의 역할을 한다. 즉, 계층 2의 진화 시 계층 1에서 적응도가 높은 개체들이 결합된 상태로 상위 모집단으로 이동, 상위 모집단 진

화에 이용된다. 이 과정을 통해 계층 1에서 공생자의 잦은 변동으로 인해 저해될 수 있는 탐색방향의 유지를 통해 진화 속도의 향상을 유도할 수 있다.

셋째, 계층 1과 계층 2의 상호작용과 계층 2의 진화는 내공생진화(endosymbiotic evolution) 과정을 모방하고 있다. 내공생진화는 큰 숙주 원핵생물에 작은 원핵생물이 들어가 분리된 형태의 공생이 아니라 결합된 형태로 내공생하면서 진핵생물로의 진화를 의미한다[13]. 따라서 계층 1에 있는 개체를 공생자로 보고, 계층 2에 있는 개체는 이들이 결합한 상태로 생존하는 내공생자(endosymbionts)로 본다. 계층 1에서의 분리된 형태의 공생 진화는 병렬 탐색을 강화하는 반면, 계층 2에서 결합된 형태의 내공생 진화는 유망한 해공간의 집중탐색에 기여할 수 있다.

넷째, 계층 1과 계층 2의 각 모집단은 $(\mu+\lambda)$ -ES에서 사용되는 선택 전략을 사용한다. 즉 부분모집단은 생산된 자손과 부모와의 합집합($(\mu+\lambda)$ 개의 개체 중에서)에서 적응도를 기준으로 가장 좋은 μ 개의 개체를 선택하여 다음 세대 모집단을 구성하게 된다. 이 방법은 엘리티즘(elitism) 전략의 하나로 부모세대에서의 우수한 개체가 사라지는 것을 방지하여, 우수한 형질이 다음세대에 전파되어 수렴속도를 향상시키고자 하기 위함이다.

다섯째, 각 모집단에서 개체 적응도는 공생자 선택전략에 의해 계산된 목적함수 벡터에 대해 비지배 순위(rank)와 밀집거리를 고려하여 평가된다. 이 방법은 NSGA-III[4]에서 사용된 개념을 본 알고리즘에 맞게 채용한 방법으로 2.3절에서 설명한다.



[그림 1] TMSEA의 구조

2.2 TMSEA의 절차

TMSEA의 절차는 다음과 같다. 먼저 사용되는 기호를 정의한다.

$P_t[h, q]$: 세대 t 에서 계층 $h(=1, 2)$ 의 q 번째 부모 모집단

$Q_t[h, q]$: 세대 t 에서 계층 $h(=1, 2)$ 의 q 번째 자손 모집단

TNS : 임시 비지배해 보관소

단계 1. (초기화)

단계 1.1 : $t=0$, $TNS=\phi$ 로 둔다.

단계 1.2 : 각 부분문제를 위한 부분 모집단 $P_t[1, q]$ ($q=1, 2, \dots, l$)와 상위 모집단 $P_t[2, 1]$ 를 임의로 생성한다.

단계 1.3 : 계층 1의 각 부분 모집단($P_t[1, q]$, $q=1, 2, \dots, l$)에 대해, 상대 모집단($P_t[1, s]$, $s=1, 2, \dots, l, s \neq q$)에서 임의로 선택한 개체와 짝지어 목적함수 벡터를 계산한다.

단계 1.4 : 각 부분 모집단($P_t[1, q]$, $q=1, 2, \dots, l$)에 있는 개체의 적응도를 구해진 목적함수 벡터를 이용하여 평가한다.

단계 1.5 : 모집단 $P_t[2, 1]$ 에 있는 개체의 적응도를 평가한다. 그리고 $t=1$ 로 둔다.

단계 2. (계층 1의 진화) $q=1, 2, \dots, l$ 에 대해

단계 2.1 : $P_t[1, q]$ 로부터 이진토너먼트 선택과 유전연산을 통해 자손 모집단 $Q_t[1, q]$ 를 만든다.

단계 2.2 : $Q_t[1, q]$ 에 있는 각 개체는 모집단 $P_t[1, s]$ ($s=1, 2, \dots, l, s \neq q$)에서 비지배순위가 1인 개체를 임의로 선택, 짝을 지어 목적함수 벡터를 계산한다.

단계 2.3 : $Q_t[1, q]$ 의 비지배해 개체결합을 TNS에 복사한다.

단계 2.4 : $P_t[1, q] \cup Q_t[1, q]$ 에서 적응도를 평가하고 적응도가 높은 순으로 개체를 선택

하여 다음세대 모집단 $P_{t+1}[1, g]$ 를 구성한다. 즉, $(\mu+\lambda)$ -ES에서 사용되는 선택 전략을 사용한다.

단계 3. (계층 2의 진화)

단계 3.1 : $P_t[2, 1]$ 로부터 이진토너먼트 선택과 유전연산을 통해 자손을 생산하여 $Q_t[2, 1]$ 를 만든다.

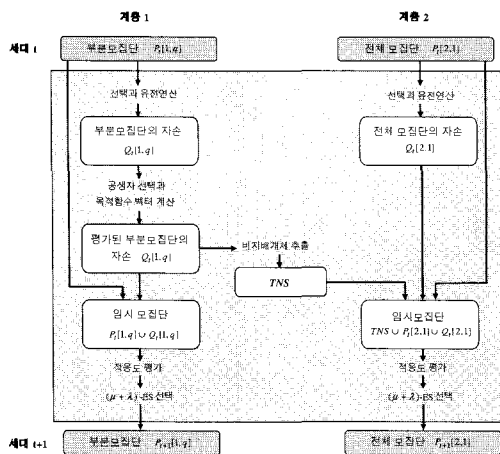
단계 3.2 : 합집합 $P_t[2, 1] \cup Q_t[2, 1] \cup TNS$ 의 임시모집단을 만들고, 임시모집단의 개체 적응도를 계산한다.

단계 3.3 : 임시모집단에서 적응도가 높은 순으로 개체를 선택하여 다음 세대 모집단 $P_{t+1}[2, 1]$ 을 구성한다. 즉, $(\mu+\lambda)$ -ES에서 사용되는 선택 전략을 사용한다.

단계 4. (종료조건)

종료조건을 만족하면 끝낸다. 이때 $P_t[2, 1]$ 에 있는 비지배해가 구하는 해가 된다. 그렇지 않으면 $t \leftarrow t+1$, $TNS = \emptyset$ 로 두고 단계 2로 간다.

[그림 2]는 TMSEA의 개념적 절차를 보여주고 있다.



[그림 2] TMSEA의 개념도

TMSEA의 특징을 MOEA가 추구하는 수렴성과 다양성의 측면에서 다음과 같이 정리할 수 있다.

2.2.1 파레토 최적해로의 수렴성

일반적으로 MOEA 성능은 적응도 평가 외에 우수 개체의 운용(엘리티즘) 여부에 많은 영향을 받는 것으로 알려져 있다[24]. 특히 엘리티즘의 사용은 파레토 최적해로의 수렴성에 많은 영향을 미친다. TMSEA는 부분문제로 이루어진 모집단, 전체문제로 구성된 모집단, 그리고 매 세대마다 부분모집단의 평가과정에서 발견된 비지배해로 이루어진 임시 비지배해 보관소(TNS)를 운영하면서 다양한 엘리티즘을 구현한다. 첫째, 계층 1의 각 부분모집단 개체는 목적함수 값을 계산하기 위해 상대 부분모집단에서 공생자를 선택한다. 이 때 공생자는 상대 부분모집단에 있는 비지배 개체 중에서 선택한다(단계 2.2). 둘째, 부분문제 모집단의 적응도 평가 시 발견된 비지배 개체는 비지배해 보관소에 보관되어(단계 2.3) 전체문제 모집단의 적응도 평가에 참여함으로써 계층 2와 상호작용한다(단계 3.2). 셋째, 각 부분모집단과 전체모집단의 진화는 $(\mu+\lambda)$ -ES의 선택전략을 사용하여 부모가 갖는 우수 개체를 다음 세대에 계속 유지될 수 있도록 한다(단계 2.4과 단계 3.3). 이러한 엘리티즘 전략의 사용은 공생 진화과정에서 각 모집단, 그리고 계층간의 우수 개체를 유지하고 이를 이용함으로써 파레토 최적해로의 수렴속도를 향상시킨다.

2.2.2 다양성 유지

TMSEA에서는 전체문제를 이루는 부분문제 각각을 하나의 종으로 간주하는 부분문제 모집단과 전체문제 모집단의 2개 종류로 이루어진 모집단을 운영한다. 이 때 부분문제 모집단의 개체는 다른 모집단들의 개체들과 짝을 지어 적응도를 부여 받고 각각 독립적으로 진화한다. 자신이 속한 환경에 따라 다른 적응도가 평가되어 진화하는 부분문제 모집단의 공생진화는 병렬탐색을 강화하여 복잡도가 높은 문제의 넓은 해공간 탐색에 유리하다. 또한, 계층 1의 부분문제 모집단에서 발견되는 우수한 개체결합이 계층 2의 전체문제 모집단과 상호작용함으로써 탐색강화와 함께 계층 2에 있는 모집단의

다양성 유지에 기여할 것으로 기대된다.

2.3 적응도 평가

계층 1에 있는 개체의 적응도 평가를 위해서는 먼저 다른 모집단에 있는 비지배해들을 공생자로 선택(절차에서 단계 2.2 참조)하여 목적함수 벡터를 계산한다. 이렇게 계산된 목적함수 벡터를 기준으로 적응도를 부여한다. 본 연구에서 개체 i 의 적응도는 다음의 평가함수 $eval_i$ 에 의해 계산되고, 평가함수 값이 낮을수록 높은 적응도를 갖는다.

$$eval_i = r(i) + \frac{1}{1+d(i)}$$

여기서 $r(i)$ 는 개체 i 의 비지배 순위(rank), $d(i)$ 는 개체 i 와 동일한 비지배 순위에 있는 다른 개체들과의 밀집거리를 각각 나타낸다. 이 방법은 개체의 비지배 순위를 우선적으로 고려하여 적응도를 부여하고, 같은 비지배 순위에 있는 개체는 밀집거리를 반영함으로써 서로 다른 적응도를 갖도록 한다. 반면 계층 2는 전체문제를 표현하는 개체로 구성되므로 이들 목적함수를 이용하여 계층 1에서와 동일한 방법으로 적응도를 평가한다.

본 연구에서 비지배순위를 구하는 방법은 Deb et al.[4]이 NSGA-II에서 제안한 방법을 사용한다. 이 방법은 자신을 지배하는 개체 수와 자신이 지배하는 개체집합을 이용하여 구하는 방법으로 계산시간이 상대적으로 적게 소요된다는 장점을 갖는다. 그리고 개체 i 의 밀집거리는 개체 i 와 각 목적 별로 가장 가까운 개체와의 거리 합이다. 이 때 각 목적 별로 끝에 위치한 개체는 무한대의 밀집거리를 갖는다고 본다. 비지배순위와 밀집거리를 구하는 절차는 <부록>을 참고할 수 있다.

3. 실험과 분석

본 연구에서 제안한 알고리즘의 성능을 평가하기 위하여 실험문제로는 연속변수(continuous varia-

bles)를 갖는 최적화 문제와 이산변수(discrete variables)를 갖는 조합 최적화문제를 사용하였다. TMSEA의 성능 분석을 위해 우수한 알고리즘으로 평가되는 NSGA-II와 SPEA2를 비교알고리즘으로 삼았다.

실험을 위한 모든 알고리즘은 JAVA 프로그래밍 언어로 구현되었으며, Intel Pentium 4 3.4Ghz CPU를 장착한 IBM계열 PC에서 수행되었다. 모집단 크기는 TMSEA의 경우 계층 2는 100, 계층 1의 부분 모집단 크기는 50개로 하였으며, 비교 알고리즘인 NSGA-II와 SPEA2의 모집단은 100개로 설정하였다. 유전연산과 관계된 파라미터는 TMSEA와 비교 알고리즘인 NSGA-II와 SPEA2에서 교차율 0.5, 돌연변이율 0.01로 모두 동일하게 사용하였다. 이러한 파라미터 값은 실험을 통해 얻은 결과이다. 종료조건은 모든 비교 알고리즘에서 공정한 평가를 위해 세대 수 보다는 재생산된 개체 수를 사용하였으며, 연속변수를 갖는 최적화문제의 경우에는 5,000개, 조합 최적화문제의 경우에는 8,000개의 재생산 개체 수로 두었다. 이러한 종료조건은 알고리즘의 성능을 판단하는 데 적절한 것으로 실험을 통해 검증하였다.

3.1 연속변수를 갖는 최적화 문제

실험(test-bed) 문제는 Zitzler et al.[23]가 제시한 2개 목적을 갖는 문제와 Deb et al.[6]의 3개 목적을 갖는 문제를 사용하였다. 이 문제들은 파레토 최적해가 알려져 있으며, 문제마다 각각의 특성을 갖는다. <표 1>은 실험에 사용된 문제이다.

각 문제의 파레토 최적해는 T1, T2, T3, T4의 경우 $g=1$ 에서, 그리고 DTLZ1, DTLZ2, DTLZ3는 $x_i^*=0.5$, $x_i^* \in X_M$ 에서 형성된다.

단일목적 문제와 달리 우수한 비지배해 집합을 다양하게 찾고자 하는 다목적 문제에서는 여러 척도를 사용하여 성능을 평가할 수 있다. 예를 들어 비지배해의 수렴속도, 구한 비지배해와 파레토 최적해(true Pareto optimal solution)간의 근사성, 구한 비지배해의 확산성 및 균일성 등이 척도로 사용

〈표 1〉 연속변수를 갖는 실험 문제

Problem	Function	변수범위	특성*
T1	Min. $f_1(x_1) = x_1$ Min. $f_2(\mathbf{x}) = g^*(1 - \sqrt{f_1/g})$ where $g = 1 + 9 \cdot (\sum_{j=2}^{30} x_j) / (30 - 1)$	$x_j \in [0, 1] \quad \forall j$	Convex
T2	Min. $f_1(x_1) = x_1$ Min. $f_2(\mathbf{x}) = g^*(1 - f_1/g)^2$ where $g = 1 + 9 \cdot (\sum_{j=2}^{30} x_j) / (30 - 1)$	$x_j \in [0, 1] \quad \forall j$	Non-Convex
T3	Min. $f_1(x_1) = x_1$ Min. $f_2(\mathbf{x}) = g^*(1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi \cdot f_1))$ where $g = 1 + 9 \cdot (\sum_{j=2}^{30} x_j) / (30 - 1)$	$x_j \in [0, 1] \quad \forall j$	Discrete
T4	Min. $f_1(x_1) = x_1$ Min. $f_2(\mathbf{x}) = g^*(1 - \sqrt{f_1/g})$ where $g = 1 + 10 \cdot (10 - 1) + \sum_{j=2}^{10} (x_j^2 - 10\cos(4\pi \cdot x_j))$	$x_1 \in [0, 1]$ $x_2, \dots, x_{10} \in [-5, 5]$	Multi-Modal
DTLZ1	Min. $f_1(\mathbf{x}) = 0.5 \cdot x_1 x_2 (1 + g(\mathbf{x}_M))$ Min. $f_2(\mathbf{x}) = 0.5 \cdot x_1 (1 - x_2) (1 + g(\mathbf{x}_M))$ Min. $f_3(\mathbf{x}) = 0.5 \cdot (1 - x_1) (1 + g(\mathbf{x}_M))$ where $g(\mathbf{x}_M) = 100 [\mathbf{x}_M + \sum_{x_j \in \mathbf{x}_M} \{(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))\}]$ $\mathbf{x}_M = \{x_3, \dots, x_7\}$, $ \mathbf{x}_M $ 은 원소 수	$x_j \in [0, 1] \quad \forall j$	Linear
DTLZ2	Min. $f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cdot \cos(x_1 \pi/2) \cdot \cos(x_2 \pi/2)$ Min. $f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cdot \cos(x_1 \pi/2) \cdot \sin(x_2 \pi/2)$ Min. $f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cdot \sin(x_1 \pi/2)$ where $g(\mathbf{x}_M) = \sum_{x_j \in \mathbf{x}_M} (x_j - 0.5)^2$, $\mathbf{x}_M = \{x_3, \dots, x_{12}\}$	$x_j \in [0, 1] \quad \forall j$	Spherical
DTLZ3	Min. $f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cdot \cos(x_1 \pi/2) \cdot \sin(x_2 \pi/2)$ Min. $f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cdot \cos(x_1 \pi/2) \cdot \sin(x_2 \pi/2)$ Min. $f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cdot \sin(x_1 \pi/2)$ where $g(\mathbf{x}_M) = 100 [\mathbf{x}_M + \sum_{x_j \in \mathbf{x}_M} \{(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))\}]$ $\mathbf{x}_M = \{x_3, \dots, x_{12}\}$, $ \mathbf{x}_M $ 은 원소 수	$x_j \in [0, 1] \quad \forall j$	Multi-Modal

주) * 특성 : 각 문제가 갖는 파레토 최적해 집합의 특성.

될 수 있다. 본 연구에서는 아래와 같은 Zitzler et al.[23]가 제시한 평가척도를 사용한다.

$$M_1 = \frac{1}{|\mathbf{U}|} \sum_{\mathbf{u} \in \mathbf{U}} \min\{\|\mathbf{u} - \mathbf{y}\| ; \mathbf{y} \in \mathbf{Y}_p\}$$

$$M_2 = \frac{1}{|\mathbf{U} - \mathbf{I}|} \sum_{\mathbf{u} \in \mathbf{U}} |\{\mathbf{v} \in \mathbf{U} ; \|\mathbf{u} - \mathbf{v}\| > \sigma^*\}|$$

$$M_3 = \sqrt{\sum_{k=1}^m \max\{(u_k - v_k)^2 ; \mathbf{u}, \mathbf{v} \in \mathbf{U}\}}$$

여기서 \mathbf{U} , \mathbf{Y}_p 는 각각 구한 비지배해와 파레토 최적해 집합을, m 은 목적의 수를 나타낸다. M_1 은 구해진 비지배해가 파레토 최적해와 근접 정도를 보여주는 지표로써 이 값이 작을수록 파레토 최적해에 근사함을 의미한다. M_2 는 구해진 비지배해 집합 중 주어진 σ^* 영역 외부에 위치한 비지배해들의 수로써 이 값이 클수록 비지배해 분포와 균일성이 더 우수하다. M_3 값은 구해진 비지배해의 영역으로,

만약 2개의 목적을 갖는 경우 양끝에 위치한 비지배해 간의 직선거리를 뜻한다. 따라서 M_3 이 클수록 다양한 해공간 탐색능력을 보여주므로 M_2 와 함께 다양성의 지표로 사용될 수 있다.

제안한 알고리즘 TMSEA와 비교 알고리즘 NSGA-II와 SPEA2에서 개체는 실수표현을 사용하였다. 교차연산자는 NSGA-II와 SPEA2의 모집단과 TMSEA의 계층 2의 전체문제 모집단에 대해서는 이산재결합(discrete recombination)교차 중 균등교차[1]를 사용하였다. 그리고 TMSEA에서 계층 1의 부분모집단은 각 변수의 수만큼 구성하였고, 이 계층의 부분문제 모집단에 대한 교차는 산술(arithmetical)교차[1]를 사용하였다. TMSEA에서 계층간 서로 다른 교차를 사용한 이유는 부분모집단 개체의 경우 한 개의 변수로 이루어져 있어 균등교차가 적절하지 않기 때문이다. 돌연변이는 TMSEA의 계층 1과 2를 포함하여 모든 비교 알고리즘에서 Gaussian 돌연변이를 적용하였다.

<표 2>는 실험문제에 대한 TMSEA와 NSGA-II, SPEA2 수행결과를 위에서 제시된 평가척도로써 계량화한 값을 보여준다. M_2 를 위한 σ^* 는 0.15로 설정하였다. 제시된 값은 알고리즘별로 문제당 10회 반복실험 결과의 평균값이다.

<표 2>에서 보듯이 수렴성을 나타내는 M_1 의 경우 DTLZ2를 제외한 모든 문제에서 제안한 알고리즘이 우수한 성능을 보였다. 이는 타 알고리즘에 비

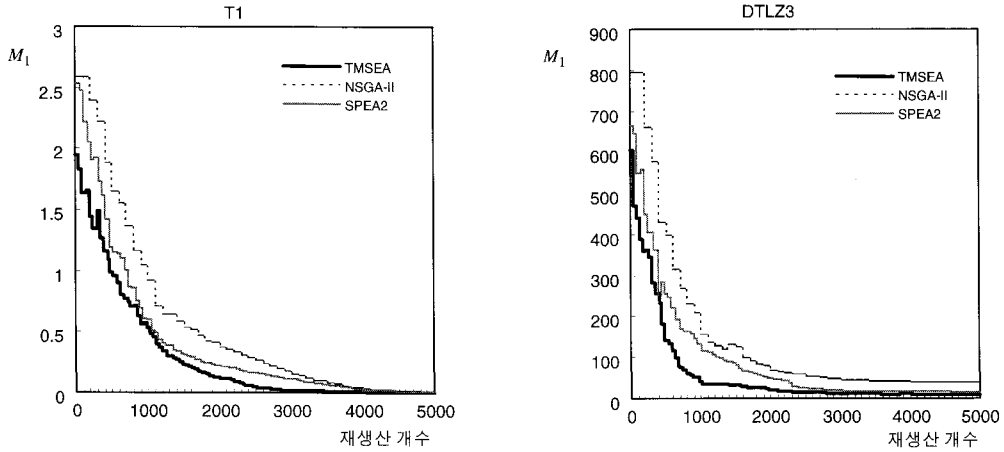
해 TMSEA가 파레토 최적해에 보다 근접한 해를 찾고 있음을 뜻한다. 구한 비지배해의 다양성의 척도인 M_2 와 M_3 의 관점에서 보면, TMSEA는 NSGA-II와 SPEA2와 비교할 때, 문제에 따라 다소 차이가 있으나 대체로 더 나쁜 결과를 보이지는 않았다.

M_1 값이 알고리즘간에 현저한 차이가 있으면 M_2 와 M_3 의 비교는 큰 의미가 없다. 넓은 지역에 걸쳐 다양한 해들을 균일하게 찾았다(M_2 , M_3 값이 크다) 할지라도 M_1 값이 큰 값을 갖는다면 이는 다른 알고리즘에 의해 찾아진 비지배해들에 의해 대부분 지배당함을 뜻하기 때문이다. 예를 들어 DTLZ1 문제에서 M_2 와 M_3 값은 NSGA-II가 TMSEA보다 큰 값을 가져 다양성은 우수하지만 M_1 값은 TMSEA가 0.650, NSGA-II는 1.550으로 보다 작다(TMSEA가 보다 우수하다). 이 경우 NSGA-II에 의해 구해진 비지배해는 TMSEA에 의해 구해진 비지배해에 의해 많은 부분 지배당하게 된다. 그리고 문제 DTLZ2에 있어서 TMSEA가 M_1 과 M_2 척도에 있어서 NSGA-II보다는 약간 좋지 않으나 SPEA2보다는 좋았다. 이 문제에 있어서는 성능의 큰 차이가 없는 것으로 판단된다.

[그림 3]은 문제 T1과 DTLZ3에 대해 알고리즘이 수행되는 동안 M_1 값의 변화를 보여주고 있다. 그림에서 알 수 있듯이 제안한 알고리즘이 탐색이 진행되는 동안 다른 알고리즘에 비해 빠른 수렴속도를 보임을 알 수 있다.

<표 2> 알고리즘의 성능 비교 : 연속변수의 실험문제

문제	M_1			M_2			M_3		
	TMSEA	NSGA-II	SPEA2	TMSEA	NSGA-II	SPEA2	TMSEA	NSGA-II	SPEA2
T1	0.004	0.067	0.032	26.017	13.830	29.676	1.411	1.402	1.385
T2	0.004	0.054	0.043	24.939	12.294	24.447	1.361	0.983	1.161
T3	0.017	0.043	0.019	22.071	11.340	17.163	1.964	2.005	1.950
T4	9.498	10.304	9.849	21.227	9.920	14.331	3.272	3.110	3.181
DTLZ1	0.650	1.550	1.610	40.573	47.155	43.928	1.785	4.917	4.114
DTLZ2	0.817	0.796	0.865	46.991	47.605	46.731	1.763	1.825	1.895
DTLZ3	7.127	14.219	12.465	33.674	28.415	42.949	34.199	26.476	23.289



[그림 3] T1과 DTLZ3의 수렴속도 비교

3.2 이산변수를 갖는 최적화 문제

이산변수를 갖는 최적화문제의 실험을 위해 다목적 0-1 배낭문제(knapsack problem)를 사용한다. 이 문제는 MOEA의 실험문제로 자주 사용되는 문제[24, 25]로, 각각의 이득과 비용을 갖는 n 개의 품목 중에서 어떤 품목을 선택하는 것이 각 배낭무게의 한계 내에서 최대의 이익을 얻겠는가 하는 문제이다. 여기서 고려하는 배낭의 수에 따라 목적의 수를 달리할 수 있다. 이는 다음과 같이 모형화된다.

$$\begin{aligned}
 & \text{Max. } f_1(\mathbf{x}) = \sum_{j=1}^n p_{1j} x_j \\
 & \text{Max. } f_2(\mathbf{x}) = \sum_{j=1}^n p_{2j} x_j \\
 & \dots \quad \dots \\
 & \text{Max. } f_m(\mathbf{x}) = \sum_{j=1}^n p_{mj} x_j \\
 & \text{s.t. } \sum_{j=1}^n w_{kj} x_j \leq c_k, \quad k=1, 2, \dots, m \\
 & x_j \in \{0, 1\}, \quad j=1, 2, \dots, m
 \end{aligned}$$

여기서 p_{kj} , w_{kj} 는 품목 j 가 k 번째 배낭에 선택될 때 이득과 무게를 각각 나타내고, c_k 는 배낭이 허용하는 무게의 한계치이다.

본 연구에서는 품목 수를 250, 500, 700개에 대해

각각 배낭(목적) 수를 2개와 3개로 하여 총 6개의 문제를 만들었다. 각 문제에 대한 p_{kj} , w_{kj} 그리고 c_k 는 Zitzler and Thiele[24]에서와 동일한 방식으로 설정하였다. 즉, p_{kj} , w_{kj} 는 범위 [10, 100]에서 임의로 생성하고, $c_k = 0.5 \sum_{j=1}^n w_{kj}$ 으로 주었다.

TMSEA에서 계층 1의 부분모집단은 각 문제를 10개의 부분문제로 분해하여 구성하였다. 그리고 모든 비교알고리즘에서 개체 표현, 유전연산 및 초기모집단 생성 방법들은 동일하게 사용하였다. 개체표현은 이진(binary)표현으로 인자 위치는 품목 번호, 인자 값은 해당 품목의 선택여부를 나타낸다. 유전연산은 이진표현에서 일반적으로 사용되는 이점교차(two point crossover)와 비트(bit) 돌연변이를 각각 사용하였다. 임의로 초기모집단을 생성한 후, 그리고 유전연산 후의 개체는 제약을 위반한 비가능일 수 있다. 이러한 비가능 개체에 대해서는 제약을 만족할 때까지 최대 이득/무게 비율이 가장 적은 품목을 제거하여 가능해를 유지하였다. 즉, 각 품목 j 에 대해 $\max_k (p_{kj}/w_{kj})$ 을 계산한 후, 이 값이 적은 품목 순으로 제약을 만족할 때까지 해당 품목을 제거한다.

<표 3>은 함수최적화 문제에서 사용된 동일한 척도로 계산한 결과를 보여준다. 각 값은 10회 반복

〈표 3〉 알고리즘의 성능 비교 1 : 조합최적화 문제

Prob.	M_1			M_2			M_3		
	TMSEA	NSGA-II	SPEA2	TMSEA	NSGA-II	SPEA2	TMSEA	NSGA-II	SPEA2
250, 2-obj.	180.47	163.81	589.05	11.82	7.10	5.57	1140.61	659.18	1019.49
500, 2-obj.	567.17	598.66	1470.26	9.58	5.77	4.35	1407.49	1017.40	885.64
750, 2-obj.	988.21	1043.59	2552.19	9.73	4.68	3.63	1923.36	1112.70	1205.11
250, 3-obj.	195.73	231.14	620.74	41.42	26.48	21.24	1519.96	1010.26	1373.85
500, 3-obj.	603.56	602.34	1404.59	45.29	19.14	11.70	2379.63	1380.17	1765.47
750, 3-obj.	1207.26	1142.48	2475.70	38.49	17.33	14.95	3028.63	1646.48	2261.86

〈표 4〉 알고리즘의 성능 비교 2 : 조합최적화 문제

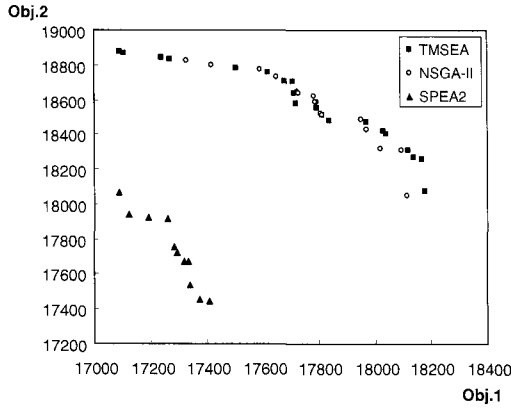
Prob.	Algorithms			TMSEA vs NSGA-II			TMSEA vs SPEA2		
	TMSEA N_T	NSGA-II N_N	SPEA2 N_S	N_{TN}	N_T/TN	N_N/TN	N_{TS}	N_T/TS	N_S/TS
250, 2-obj.	24.10	14.50	11.20	23.60	14.60	9.00	24.10	24.10	0.00
500, 2-obj.	19.30	11.70	8.80	21.00	13.30	7.70	19.30	19.30	0.00
750, 2-obj.	19.50	9.40	7.30	17.20	13.90	3.30	19.50	19.50	0.00
250, 2-obj.	82.90	53.00	42.50	89.50	68.90	20.60	82.90	82.90	0.00
500, 3-obj.	90.60	38.30	23.40	85.70	61.20	24.50	90.60	90.60	0.00
750, 3-obj.	77.00	34.70	29.90	57.40	32.40	25.00	77.00	77.00	0.00

실험 결과의 평균 값이다. 함수최적화 문제와 달리 배낭문제에서는 파레토 최적해가 알려져 있지 않으므로, M_1 을 계산하기 위한 집합 Y_p 는 각 문제에 대해 충분한 시간 동안 제안한 알고리즘과 비교 알고리즘을 수행한 후 구해진 비지배해 집합을 사용하였다. M_2 에서 σ^* 는 20.0으로 두었다.

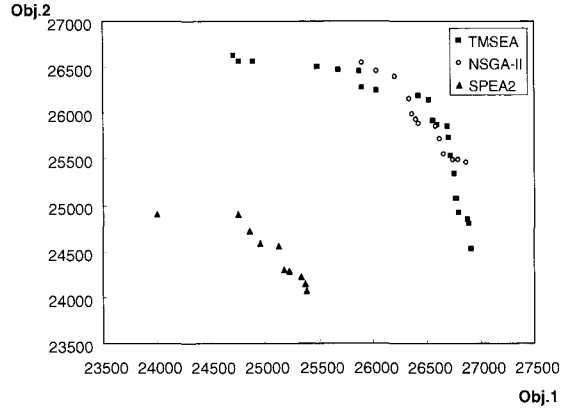
해의 질과 수렴속도를 보여주는 M_1 값은 TMSEA와 NSGA-II가 서로 비슷한 성능을 보였고 SPEA2의 경우가 가장 좋지 않았다. 다양성의 측면을 보여주는 척도인 M_2 와 M_3 에서는 TMSEA가 월등히 우수하였다. 이 결과는 TMSEA가 SPEA2보다는 모든 면에서 우수하고, NSGA-II와 수렴속도는 유사하지만 TMSEA에 의해 구해진 비지배해가 NSGA-II의 비지배해에 비해 보다 넓은 영역에 걸쳐 분포하고 있음을 의미한다. 따라서 TMSEA는 NSGA-II와 비교하여 질적으로 유사한 해들을 보다 다양하게 찾고 있다고 볼 수 있다.

이러한 관계를 보여주기 위해 Hyun et al.[10]이 제

시한 척도를 사용하여 분석해 보자. 설명을 위해 알고리즘 A, B에 의해 얻은 비지배해 집합을 각각 Set_A , Set_B 라 하자. 각 집합의 크기를 각각 $N_A(=|Set_A|)$, $N_B(=|Set_B|)$ 라고 하면, N_A , N_B 는 비지배해 집합의 개체 수를 뜻하므로 알고리즘 A와 B의 양적 지표가 된다. 하지만 이 N_A , N_B 의 지표는 단순한 개체의 숫자이므로 지배관계, 즉 질적인 비교에 대해서는 어떤 정보도 주지 못한다. 따라서 질적 비교를 위해 Set_A , Set_B 의 합집합에서 새로이 추출한 비지배해 집합 Set_{AB} 를 만들고 그 크기를 $N_{AB}(=|Set_{AB}|)$ 라고 하자. Set_{AB} 에서 Set_A 에 속한 개체 수, Set_B 에 속한 개체 수를 각각 $N_{A/AB}(=|Set_A \cap Set_{AB}|)$, $N_{B/AB}(=|Set_B \cap Set_{AB}|)$ 라고 놓으면 $N_{A/AB}$, $N_{B/AB}$ 값은 알고리즘 A, B에 의해 수행된 최종 비지배해의 질적 비교척도로 사용될 수 있다. $N_{A/AB} > N_{B/AB}$ 은 알고리즘 A에 의해 얻어진 비지배해들이 알고리즘 B보다 질적으로 우수하거나 또는 더 많은 비지배해를 구했음을 의미한다. 〈표 4〉는 이 척도에 의한



(a) 다목적 배낭문제 : 품목 수 = 500



(b) 다목적 배낭문제 : 품목 수 = 750

[그림 4] 비지배해 집합의 비교

분석 결과를 보여 주고 있다. 표에서는 TMSEA, NSGA-II, 그리고 SPEA2를 간단히 T , N , S 로 각각 표시한다.

<표 4>로부터 모든 문제에 대해 TMSEA는 양적 측면(N_T)과 질적 측면($N_{T/TN}$, $N_{N/TN}$)에서 모두 비교 알고리즘보다 우수한 성능을 보임을 알 수 있다. 특히 <표 3>에서 명확하지 않은 NSGA-II와의 성능 차이를 <표 4>는 보다 분명하게 보여준다. 즉, TMSEA는 NSGA-II에 비해 보다 많은 수의 비지배해를 찾고 있으며($N_T > N_N$), 질적인 측면에서도 보다 좋은 결과를 보이고 있다($N_{T/TN} > N_{N/TN}$). 이 결과로 TMSEA는 NSGA-II가 탐색하지 못하는 영역을 좀 더 효율적으로 탐색할 수 있다고 판단할 수 있다. [그림 4]는 2개 목적을 갖는 품목 500, 750의 문제에 대해 구해진 비지배해 집합을 표시한 것이다. 위에서 언급한 대로 TMSEA과 NSGA-II는 해의 질적인 측면에서는 유사하지만 다양성에 있어서 TMSEA가 보다 우수함을 보여준다.

컴퓨터 계산시간은 NSGA-II가 가장 적었고 TMSEA가 가장 많은 시간을 요구하였다. 대표적으로 연속변수를 갖는 실험문제 DTLZ3의 경우 TMSEA, NSGA-II, SPEA2 순으로 16.3초, 1.5초, 5.9초가 각각 소요되었고 조합최적화 문제 750, 3-obj.의 경우 각각 22.7초, 3.7초, 10.1초의 계산시간을 보였다. TMSEA은

부분모집단을 운영하므로 각 부분모집단에서 개체의 공생자 선택과 적응도 평가로 인해 비교 알고리즘들 보다 상대적으로 많은 계산시간이 소요된 것으로 보인다.

계층 1의 부분문제 모집단을 구성할 때 몇 개로 분해할 것인가 하는 분해도(degree of decomposition) 결정은 알고리즘 성능에 영향을 줄 수 있다 [17]. 본 연구에서는 문제의 복잡도를 고려하여 적절한 수로 고정시켰다. 분해도의 결정은 문제가 갖는 특성에 따라 달라질 수 있고 알고리즘의 구조(예로 계층의 수) 역시 달라질 수 있어 보다 세밀한 분석이 요구되기 때문이다. 분해의 정도와 계층의 수에 따른 성능 분석은 추후 연구주제가 될 수 있다.

실험결과, TMSEA는 연속변수를 갖는 최적화 문제에서는 해의 수렴성이 우수하였고 조합최적화 문제에서는 상대적으로 다양성이 우수하였다. 이는 함수최적화 문제의 경우 별도의 제약이 없어 비교적 균일한 해공간을 갖는 이유로 다양성보다는 수렴성에 효율적인 반면, 제약으로 인해 복잡한 해공간을 갖는 조합최적화 문제에서는 다양성에 있어 그 우수성이 무각된 것으로 판단된다. 또한 비교알고리즘인 NSGA-II와 SPEA2의 결과는 비록 다른 비교척도를 사용하였지만 수렴성, 다양성, 계산시간 등이 Khare et al.[12]과 Zitzler et al.[25]의 결과와

유사하였다.

4. 요약 및 결론

본 연구에서는 다목적 최적화문제에서 잘 분포된 파레토 최적해 또는 근사 파레토 최적해를 탐색하기 위하여 2계층 구조를 갖는 새로운 공생 진화알고리즘을 제안하였다. 기존 알고리즘이 전체문제를 한 개체로 표현한 모집단을 운영하는 반면 TMSEA에서는 전체문제를 분해한 부분문제를 표현한 여러 부분모집단과 함께 전체해를 표현한 개체들로 구성된 모집단을 동시에 운영한다. 이러한 구조는 다양한 해공간의 탐색을 유도하고 엘리티즘을 강화하여 우수한 비지배해를 다양하게 찾을 수 있도록 한다.

TMSEA의 성능 평가를 위해 연속변수를 갖는 함수최적화 문제와 이산변수를 갖는 조합최적화 문제를 실험문제로 사용하여 기존의 타 알고리즘과 비교 분석하였다. 실험 결과, TMSEA는 비교 알고리즘에 비해 해의 수렴성과 다양성 측면에서 우수하고 안정된 성능을 보였다. 특히, TMSEA는 연속변수의 최적화 문제에서는 해의 수렴성이, 조합 최적화문제에서는 해의 다양성이 우수하였다. 이러한 실험결과는 제안한 알고리즘이 복잡도가 높은 다목적 문제해결에 유용한 도구로 사용될 수 있음을 보여준다.

참고 문헌

- [1] 김여근, 윤복식, 이상복, 「메타휴리스틱」, 영지문화사, 1997.
- [2] Coello, C.A.C., "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems*, Vol.1, No.3(1999), pp.269-308.
- [3] Deb, K., "Multi-objective Genetic Algorithms : Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, Vol.7, No.3(1999), pp.205-230.
- [4] Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : NSGA-II," In M.S. et al.(Ed.), *Parallel Problem Solving from Nature - PPSN VI*, Berlin, Springer, (2000), pp.849-858.
- [5] Deb, K., M. Mohan, and S. Mishra, "Evaluating the ϵ -Dominance Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions," *Evolutionary Computation*, Vol.13, No.4(2005), pp. 501-525.
- [6] Deb, K., L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," TIK-Report No.112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, July, 2001.
- [7] Fonseca, C.M. and P.J. Fleming, "Genetic algorithm for multiobjective optimization, formulation, discussion and generalization," In Forrest, S. (ed.) *Genetic Algorithms : Proceeding of the Fifth International Conference*, 416-423. Morgan Kaufmann, San Mateo, CA. 1993.
- [8] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Massachusetts. 1989.
- [9] Horn, J., N. Nafpliotis, and D.E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," *IEEE international Conference on Evolutionary Computation*, Vol.1(1994), pp.82-87.
- [10] Hyun, C.J., Y.H. Kim, and Y.K. Kim, "A genetic algorithm for multiple objective sequencing problems in mixed model assem-

- bly," *Computers & Operations Research*, Vol.25(1998), pp.657-690.
- [11] Kim, J.Y., Y. Kim, and Y.K. Kim, "An endosymbiotic evolutionary algorithm for optimization," *Applied Intelligence*, Vol.15(2001), pp.117-130.
- [12] Khare, V., X. Yao, and K. Deb, "Performance Scaling of Multi-objective Evolutionary Algorithms," *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, Springer. Lecture Notes in Computer Science. Vol. 2632, Faro, Portugal, (April 2003), pp.376-390.
- [13] Kim, Y.K., J.Y. Kim, and Y. Kim, "An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines," *European Journal of Operational Research*, Vol.168, No.3(2006), pp.838-852.
- [14] Knowles, J.D. and D.W. Corne, "The Pareto archived evolution strategy : A new baseline algorithm for multi-objective optimization," *IEEE International Conference on Evolutionary Computation*, (1999), pp.98-105.
- [15] Laumanns, M., L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multi-objective optimization," *Evolutionary Computation*, Vol. 10, No.3(2002), pp.263-282.
- [16] Michalewics, Z., "Genetic Algorithms+Data Structures=Evolution Programs," Second, Extended Edition, Springer-Verlag, 1992.
- [17] Muhlenbein, H. and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm I. Continuous parameter optimization," *Evolutionary Computation*, Vol.1 No.2(1993), pp.25-49.
- [18] Potter, M.A., "The design and analysis of a computational model of cooperative co-evolution," Ph.D. dissertation, George Mason University, 1997.
- [19] Schaffer, J.D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," In *Genetic Algorithms and their Applications : Proceedings of the First International Conference on Genetic Algorithms*, pp.93-100, Lawrence Erlbaum, 1985.
- [20] Srinivas, N. and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, Vol.2, No.3(1985), pp.221-248.
- [21] Tan, K.C., T.H. Lee and E.F. Khor, "Evolutionary Algorithms for Multi-Objective Optimization : Performance Assessments and Comparison," *Artificial Intelligence Review*, Vol.17(2002), pp.253-290.
- [22] Veldhuizen, D.A.V. and G.B. Lamont, "Multiobjective evolutionary algorithms : Analyzing the state-of-the-art," *Evolutionary Computation*, Vol.8, No.2(2000), pp.125-147.
- [23] Zitzler, E., K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms : Empirical results," *Evolutionary Computation*, Vol.8, No.2(2000), pp.173-195.
- [24] Zitzler, E. and L. Thiele, "Multiobjective evolutionary algorithms : A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, Vol.3, No.4(1999), pp.257-271.
- [25] Zitzler, E., M. Laumanns, and L. Thiele, "SPEA2 : Improving the Strength Pareto evolutionary Algorithm," Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May, 2001.

부록

A1. 개체 집합 Q 에서 개체 i 의 비지배 순위 $r(i)$ 를 구하는 절차

fast-nondominated-sorting ($Q, r(i)$)

단계 1: $l=1, F_l=\phi$ 로 둔다.

단계 2: 개체 $p(p \in Q)$ 를 선택하고, $S_p=\phi, n(p)=0$ 으로 둔다.

단계 2.1: 각 개체 $q(q \in Q, q \neq p)$ 에 대해, $p > q$ 이면 $S_p \leftarrow S_p \cup \{q\}$, $p < q$ 이면

$n(p) \leftarrow n(p)+1$ 로 한다.

단계 2.2: $n(p)=0$ 이면 $r(p)=l$ 로 두고 $F_l \leftarrow F_l \cup \{p\}$, $Q \leftarrow Q - \{p\}$ 로 한다.

단계 2.3: $Q \neq \phi$ 이면 단계 2로 간다.

단계 3: 집합 $H=\phi$ 로 둔다.

단계 3.1: 집합 F_l 에서 임의로 개체 j 를 선택한다.

단계 3.2: S_j 에 있는 각 개체 k 에 대해 $n(k) \leftarrow n(k)-1$ 로 한다. 만약 $n(k)=0$ 이면

$H \leftarrow H \cup \{k\}$, $r(k)=l+1$ 로 한다.

단계 3.3: F_l 에서 개체 j 를 제거한다. $F_l \neq \phi$ 이면 단계 3.1로 간다.

단계 3.4: $l \leftarrow l+1, F_l=H$ 로 둔다.

단계 3.5: $F_l=\phi$ 이면 끝내고, 그렇지 않으면 단계 3으로 간다.

여기서, $p > q$ 는 개체 p 가 개체 q 를 지배함을 의미한다. 단계 1과 단계 2는 각 개체들의 비지배관계를 판단하고 최상위 지배순위($l=1$)를 구하는 과정이다. $n(p)$ 는 개체 p 를 지배하는 개체 수이고, S_p 는 개체 p 가 지배하는 개체들의 집합이다. $n(p)=0$ 인 개체 p 는 자신을 지배하는 개체 수가 0이므로 비지배해이다. 따라서 단계 2에서 집합 $F_l(l=1)$ 은 Q 의 개체들 중 비지배해들로 이루어진 개체집합이다.

단계 3은 위 단계에서 구해진 $n(p)$ 와 S_p 를 이용하여 차례로 비지배 순위를 결정하는 과정이다. 현재의 비지배 순위 l 에 속하는 개체 $j(j \in F_l)$ 가 지배하는 개체 집합은 S_j 이다. 따라서 S_j 내 각 개체 $k(k \in S_j)$ 의 $n(k)$ 값을 1씩 감소시켜 $n(k)=0$ 이 되면, 개체 k 는 현재 비지배해 외에 자신을 지배하는 개체가 없으므로 현재의 순위보다 한 단계 낮은 비지배 순위($l-1$)를 갖게 된다. 이러한 과정을 반복하여 개체의 비지배순위를 차례로 결정한다.

A2. 동일한 비지배순위를 갖는 개체 집합 S 에서 개체 i 의 밀집거리 $d(i)$ 를 구하는 절차

crowding-distance-assignment ($S, d(i)$)

$f_m(i)$: 개체 i 의 m 번째 목적의 함수 값

$m[j]$: m 번째 목적의 함수 값을 기준으로 정렬한 후의 j 번째 개체

L : 고려하는 목적의 수

단계 1: $m=1, d(i)=0, i=1, 2, \dots, |S|$ 로 둔다.

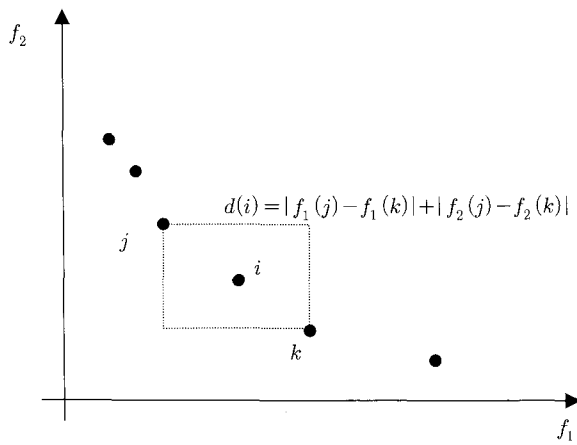
단계 2 : m 번째 목적의 함수값을 기준으로 오름차순으로 정렬한다.

$$d(m[1]) = d(m[|S|]) = \infty \text{로 둔다.}$$

단계 3 : $j=2, 3, \dots, |S|-1$ 에 대해 $d(m[j]) \leftarrow \{f_m(m[j+1]) - f_m(m[j-1])\}$ 을 계산한다.

단계 4 : $m=L$ 이면 끝내고, 그렇지 않으면 $m \leftarrow m+1$ 로 두고 단계 2로 간다.

개체 i 의 밀집거리 $d(i)$ 는 개체 i 와 동일한 비지배 순위를 갖는 개체 중에서 i 와 각 목적별로 거리가 가장 가까운 개체들과의 거리를 합한 값이다. 예를 들어 2개의 최소화 목적을 갖는 경우, 밀집거리는 [그림 5]와 같다.



[그림 5] 밀집거리