

불균등한 시설배치문제를 위한 Coral Reef Optimization 방법론 적용

정동진, 채준재[†]

한국항공대학교 항공교통물류학부 물류전공

Applicability of Coral Reef Optimization to Unequal area facility layout problem

Dongjin Jeong, Junjae Chae

School of Air Transport, Transportation and Logistics, Korea Aerospace University

Unequal area facility layout problem is an optimization problem that we place given departments on given floor space to minimize total material handling movement or maximize its efficiency. We study to find its applicability of Coral Reef Optimization to unequal area facility layout problem. Coral reef optimization is relatively new and there are not many researches applying this method to layout design. Specifically, a slicing tree structure is used to place rectangular shape blocks which represent functional departments in manufacturing system and the coral reef optimization is improving its fitness which is represented by total material movement based on the given configuration. The instances from previous literature are used to experiment and the results were compared to optimal solutions. To see the applicability, we only take small problems less than 20 blocks and result shows that the coral reef optimization properly work with unequal area facility layout problem.

Keywords: Unequal area facility layout problem, Coral reef optimization, Facility design, Meta-heuristic, Optimization

논문접수일 : 2021.10.12.

심사완료일 : 2022.06.22.

게재확정일 : 2022.06.24.

[†] Corresponding Author: jchae@kau.ac.kr

1. 서론

1.1 소개

시설배치문제(FLP: Facility Layout Problem)는 제약 조건을 만족하면서 그 효과를 최대화 하거나 비용이 최소화 되도록 시설을 배치하는 문제이다. Tompkins et al.(2010)은 좋은 배치는 회사의 총 제조물류 비용을 20% ~ 50%까지 줄일 수 있다고 했다. 이러한 이유로 FLP는 제조업에서 물류 비용을 줄이기 위해 고려해야 할 매우 중요한 문제이다(Kouvelis et al., 1992). FLP의 적용에는 여러가지가 있는데, 그 중에서 우리가 다룰 문제는 불균등한 면적의 시설 배치 문제(UA-FLP : Unequal Area Facility Layout Problem)이다(Drira et al., 2007; Hosseini-Nasab et al., 2018; Anjos and Vieira, 2017). 이는 일반적인 배치설계 문제에 많이 통용되는 형태이고 실제 산업현장과 유사하게 사용할 수 있는 문제 유형이다. UA-FLP 는 Armour and Buffa(1963)에 의해 처음 소개되었는데, 전체 직사각형 공간에 주어진 직사각형 모양의 시설을 배치하는 것인데, 이때 목표는 전체 물류이송거리의 합의 최소화다. 주요 제약조건으로는 첫째, 시설들 간의 중복(혹은 겹침)이 허용되지 않으며 둘째, 주어진 공간 안에 모든 시설이 배치되어야 한다. 일반적으로 UA-FLP의 목표는 배치가 이루어진 시설들 사이에서의 총 물류이송비용의 최소화 이다(Gonsalves and Resende, 2015).

1.2 문헌연구

최근 UA-FLP 해결을 위한 다양한 접근법이 소개 되었는데 Komarudin and Wong(2010)은 결정론적 절차와 휴리스틱/메타휴리스틱 방법의 해결법을 소개했고 Meller et al.(1998)은 Branch and Bound 알고리즘을 통해 비순환 그래프로 문제를 해결했다. Montreuil(1991) 과 Konak et al.(2006) 은 혼합 정수 계획법을 UA-FLP에 적용했고, Meller et al.(1998)은 Montreuil의 방법을 수정하여 보다 큰 규모의 문제를 해결했다. Sherali et al.(2003)과 Castillo et al.(2005)은 혼합정수계획법에서 유동적인 블록의 면적에 대한 선형화를 통해 오류를 제어할 수 있는 방법을 제안했다.

Saraswat et al.(2015)와 Purnomo and Wiwoho(2016)는 목표를 달성하기 위해 Sherali et al.(2003)의 방법을 사용했다. Chae and Regan(2016)은 최대 13개 블록에 대한 최적배치를 도출했다. 이들은 가로세로가 모두 정해진 시설과 그렇지 않은 시설을 동시에 고려했다. 일반적으로 메타 휴리스틱 방법은 매우 큰 사례에서 결정론적 방법보다 성능이 좋다. 이것이 휴리스틱 및 메타 휴리스틱 접근법이 이 문제를 해결하는 데 자주 사용되는 이유다. Tam(1992)은 최선의 해결책을 찾기 위해 LOGIC 이라고 불리는 Simulated Annealing 접근법을 소개했다. 그 후에 Scholz et al.(2009) 과 Kulturel-Konak(2012)은 Tabu Search를 제안했다. 다수의 연구자들은 유전 알고리즘(GA : Genetic Algorithm)을 채택했다. 이 방식으로 Tate and Smith(1995)는 실현 가능한 해만 보기 위해 페널티 함수를 포함하는 GA를 제안했다(때때로 값이 좋은 해의 경우 제약조건을 만족하지 못한다.). Azadivar and Wang(2000) 은 Slicing Tree Structure(STS)를 사용한 GA를 통해 UA-FLP를 해결했다. Wu and Appleton(2002) 과 Gomez et al.(2003)은 시설이 배치될 공간 내부의 복도를 고려해 문제를 해결했다. Enea et al.(2005)는 특히 가로 세로 비율(Aspect ratio, α) 제약조건을 고려하여 UA-FLP에 GA를 사용했다. 또한 Aiello et al.(2006)는 UA-FLP를 다루기 위해 GA와 Electre 알고리즘의 조합을 사용했으며, Liu and Meller(2007)는 혼합 정수 계획법과 GA를 결합한 방법을 고안했다. 최근에는 Palomo-Romero et al.(2017)가 아일랜드 모델에 기초한 GA를 사용해 다양성을 향상시켜 더 좋은 해를 만들었다. 그 외에도 다양한 메타휴리스틱이 사용되었는데, 개미 군집 최적화 (Komarudin and Wong, 2010; Wong and Komarudin, 2010; Kulturel-Konak, 2011; Liu and Liu, 2019), 인공 면역 체계 (Ulutas and Kulturel-Konak, 2012), Biased random-key GA (Gonsalves and Resende, 2015) 등이 그 사례

다. 마지막으로, Kang and Chae(2017)는 Geem et al(2001)이 제안한 Harmony Search 방법을 채택하여 UA-FLP를 해결했다. 그들은 해의 다양성을 위해 Shayan and Chittilappilly(2004)의 STS 표현을 개량했다.

위에서 언급한 알고리즘으로 UA-FLP의 해를 나타낼 때 보통 두 가지의 표현 중 하나로 나타낸다. 그것은 STS와 Flexible Bay Structure(FBS)다. <Figure 1> STS에서는 공간을 불규칙하게 나눌 수 있으며(Kang and Chae, 2017; Komarudin and Wong, 2010; Scholz et al., 2009; Shayan and Chittilappilly, 2004), FBS에서는 Bay라고 불리는 직사각형 공간에만 시설을 배치한다(Kulturl-Konak and Konak, 2011). Coral Reef Optimization을 사용한 선행연구(Garcia-Hernandezatal, 2019)는 FBS 방식을 사용했는데 이는 STS와 비교했을 때 배치 다양성이 다소 떨어지는 단점이 있다. 따라서 이번 연구에서는 Kang and Chae(2017)의 개량된 STS를 적용해 UA-FLP를 해결한다.

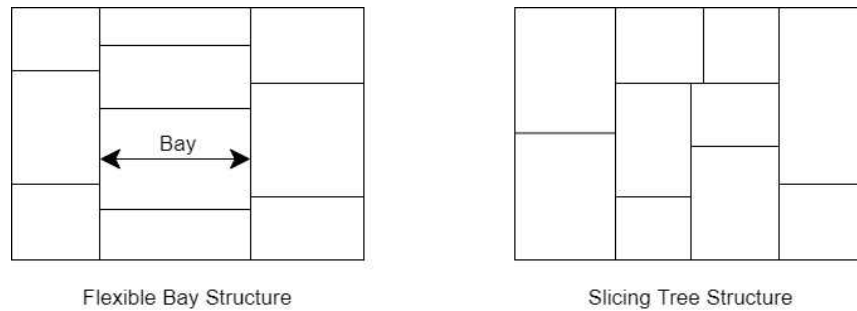


Figure 1. Flexible Bay Structure and Slicing Tree Structure

본 연구에서는 진화 기반 알고리즘인 산호초 최적화(CRO : Coral Reef Optimization) 방법(Salcedo-Sanz et al., 2014)의 적용성을 STS를 바탕으로 실험했다. CRO는 산호초에서 발생하는 생식, 착상, 복제 그리고 퇴화를 모방하는 진화형 알고리즘이다. CRO는 진화 알고리즘과 Simulated Annealing(Salcedo-Sanz et al., 2014a)을 조합한 것과 유사한 알고리즘으로 통신(Salcedo-Sanz et al., 2014d)(Salcedo-Sanz et al., 2014b), 에너지(Salcedo-Sanz et al., 2016)(Salcedo-Sanz et al., 2014c), 구조공학(Salcedo-Sanz et al., 2017)(Camacho-Gómez et al., 2018) 혹은 응용의료(Bermejo et al., 2018) (Yan et al., 2019)와 같은 다양한 분야에서 사용된다.

최근 CRO는 군집화(Medeiros, Xavier and Canuto, 2015), 인공신경망(Yang and Zhang, 2016), 시계열 분석(Duran-Rosal et al., 2018) 또는 자원 할당 문제 등 그 적용범위가 확대되고 있다. 이러한 문제들에서 CRO는 우수한 성능을 보여준다(Salcedo-Sanz, 2017). 우리 연구는 다양한 분야에서 그 효과를 보이고 있는 CRO를 UA-FLP에 적용하고자 한다. 이는 성능 자체의 평가보다도 최적해가 알려진 작은 규모의 문제들을 알려진 방법론대로 적용할 경우 적절히 해결하는지 확인하는 것이 일차적인 목표이다. FBS는 정해진 Bay 형태 안에서 배치를 하기 때문에 실질적인 최적해를 찾기에 한계가 있다. STS를 사용하게 되면 그 제약을 제거함으로써 인해 더 큰 수준의 다양성을 갖게 되어 최적 수준에 가깝게 된다. 이는 최적해를 찾기까지 더 많은 경우의 수가 요구되고 적절히 구현하기에 매우 어려운 조건이 된다. 이전 연구를 살펴보면 FBS를 이용한 UA-FLP는 CRO로 적용된 적이 있지만(Garcia-Hernandezetal, 2019), STS에 적용된 경우는 아직까지 없기 때문에 이 알고리즘을 통한 최적해로의 접근의 결과를 통해 이 알고리즘의 성능을 논할 수 있을 것이다.

2. 본론

2.1. UA-FLP

UA-FLP를 위한 수리모형을 통해 각 시설의 위치를 결정할 수 있다. UA-FLP는 주어진 직사각형 공간에 직사각형 모양의 시설을 배치하는 문제로, 이때 각 시설은 겹치지 않아야하며 각 시설의 중심좌표를 이용해 총 물류이송거리를 최소화하는 것을 목표로 한다. 이때 각 시설의 크기는 각각 다르다는 것을 가정한다. 문제를 해결하기전 각 시설의 넓이와 가로세로비(Aspect ratio, α)의 최댓값 그리고 시설이 들어가는 공간의 가로, 세로는 주어진다. 각 시설마다 그 값이 다른데, 그 이유는 시설마다 면적이 서로 다르기 때문이다. 이 문제의 목적함수는 각 시설간 총 물류이송거리의 최소화이며 정의된 변수와 매개변수는 다음과 같다. 이때 시설의 형태는 가로, 세로의 길이가 그 정해진 비율(Aspect ratio)안에서 움직일 수 있고 이후로는 편의를 위해 블록으로 명명하기로 한다.

Parameters and variables

n	각 블록의 수
L^s	주어진 공간의 x 축, y 축 길이
l_i^s	블록 i 의 x 축 혹은 y 축 길이
lb_i^s, ub_i^s	블록 i 의 x 축 혹은 y 축 길이의 하한, 상한값
f_{ij}	블록 i 와 j 사이의 물동량
d_{ij}^s	블록 i 와 j 사이의 x 축 혹은 y 축 거리
c_i^s	블록 i 의 중심의 x 좌표 혹은 y 좌표
z_{ij}^x	$\begin{cases} 1, \text{블록 } i \text{가 블록 } j \text{의 왼쪽에 있을 경우} \\ 0, \text{그 반대의 경우} \end{cases}$
z_{ij}^y	$\begin{cases} 1, \text{블록 } i \text{가 블록 } j \text{의 아래에 있을 경우} \\ 0, \text{그 반대의 경우} \end{cases}$

$$\text{Minimize} \quad \sum_i \sum_{j(i \neq j)} f_{ij} (d_{ij}^x + d_{ij}^y) \quad (1)$$

$$\text{Subject to} \quad a_i = l_i^x \times l_i^y, \forall i \quad (2)$$

$$d_{ij}^s = |c_i^s - c_j^s|, \forall i, j \forall s \quad (3)$$

$$lb_i^s \leq l_i^s \leq ub_i^s, \forall i, s \quad (4)$$

$$\sum_{s=x}^y (z_{ij}^s + z_{ji}^s) = 1, \forall i, j > i \quad (5)$$

$$c_i^s + \frac{l_i^s}{2} \leq c_j^s - \frac{l_j^s}{2} + L^s(1 - z_{ij}^s) \quad \forall i, j \neq i \quad \forall s \quad (6)$$

$$\frac{l_i^s}{2} \leq c_i^s \leq L^s - \frac{l_i^s}{2} \quad \forall i, s \quad (7)$$

$$z_{ij}^s \in \{0, 1\} \quad \forall i, j \neq i, s \quad (8)$$

불균등한 시설배치문제의 목적함수는 (1)에 적힌 것과 같이 총 물류이송거리(물동량 * 직각거리)의 최소화이다. 이때 (2)~(8)의 제약조건을 만족해야하는데, 블록 i 의 넓이는 (2)에 적힌 것과 같이 각 축 길이의 곱으로 나타내며, 각 블록간 거리는 (3)과 같이 직각거리이다. (4)는 각 블록은 길이의 상한값과 하한값을 가진다는 의미이며, (5)~(8)은 겹침방지에 대한 제약조건이다.

2.2. 방법론 소개

2.2.1 Layout representation

문제를 표현하기 위해 STS를 사용한다(Komarudin and Wong, 2010). STS는 주어진 공간을 수직, 수평적으로 연속해서 나눠 나눈 공간에 블록을 배치하는 방법이다. 이런 연속적인 분할 과정은 세가지 표현으로 구분한다. 각각 Cut point, Cut code, Department sequence이다. n 개의 블록을 배치하는 UA-FLP를 가정할 때 총 $(3n-2)$ 개의 벡터값이 필요하다.

Department sequence는 블록들의 배열 순서를 뜻하며 블록의 수인 n 개의 벡터로 표현된다. Cut point는 이 순서를 따라 블록을 자른다는 뜻이며, 블록 사이사이를 자르는 것이므로 총 $(n-1)$ 개의 벡터가 필요하다. 예를 들어 종이 한 장이 있다고 생각해보자. 종이를 반으로 자르면 2장이 된다. 다시 말해 종이를 자를수록 수량이 많아지고 그 분할된 종이를 속에 블록을 넣는 것이다. 따라서 n 개의 종이를 만들기 위해서는 $(n-1)$ 번 종이를 잘라야한다. Cut code는 어떤 방식으로 자를 것인지를 뜻한다. 기존에는 자르는 방법이 수직과 수평 두가지 뿐이었다(Tam, 1992a). 그러나 단순히 수직과 수평 이외에 역수직, 역수평이 추가되었고 우리 연구도 이 방식을 채택한다(Kang and Chae, 2017). 4가지 방식을 사용하는 이유는 2가지로는 해의 다양성이 떨어지기 때문이다. <Figure 2>는 각 표현을 예시를 통해 설명한 것이다.

Cut point는 Cut이 발생할 순서를 의미하고 Cut code는 0, 1, 2, 3 순서대로 수평, 수직, 역수평, 역수직을 뜻하는데 수평의 경우 자르는 위치를 기준으로 우측의 블록을 아래에 배치하고 좌측의 블록을 위에 배치한다. <Figure 2>를 보면 블록 1과 2, 3, 4, 5 사이에 Cut이 처음으로 발생하는데 그때 자르는 방식은 수평(0)이다. 그래서 1을 위로 배치하고 2, 3, 4, 5를 하단에 배치했다. 한번 잘린 블록은 배치에 영향을 줄 수 없으므로 블록 1의 위치는 가장 상단에 고정된다. 그 후로 블록 2와 3, 4, 5 사이에 수직(1) Cut이 발생하는데 수직으로 자를 경우 단순히 일렬로 둔다고 생각하면 된다. 다만 앞에서 좌측 블록의 위치가 고정됐던 것처럼 수직도 좌측의 블록의 위치가 확정된다. 마찬가지로 남은 블록에 대해 같은 과정을 반복하면 <Figure 2> 좌측 하단의 배치가 나온다. Cut code 2, 3은 0, 1의 반대개념인데 기존에 위에 배치했던 블록을 아래로 배치하고 기존에 일렬로 배치했던 것을 자르는 지점을 기준으로 좌측과 우측의 위치를 바꾸면 된다.

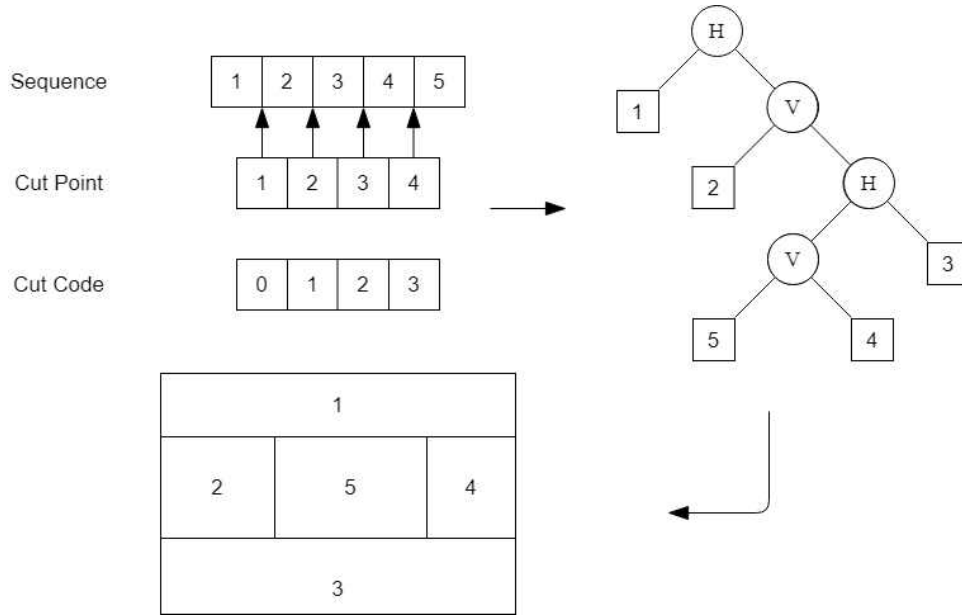


Figure 2. Slicing Tree Structure

2.2.2 Coral Reef Optimization

산호초 최적화 알고리즘(CRO)은 Salcedo-Sanz(2014a)에 의해 제안되었다. 산호초의 진화와 생태계에서 일어나는 과정을 모방하는 Bio-inspired algorithm이다. 이 알고리즘은 $N \times M$ 의 사이즈를 가진 직사각형 격자(암초, reef) 위에서 진행된다(<Figure 3>의 격자와 동일). CRO는 Population-based algorithm이고 그 알고리즘이 진행되는 공간이 <Figure 3>의 격자이다. 암초위의 다양한 장소에서 여러 산호가 자라나는 것과 같이 격자 위의 셀 하나하나가 각각의 산호를 가리킨다. (i, j) 에 위치한 각 사각형은 산호 $\eta(i, j)$ 를 나타낸다. $(i \leq N, j \leq M)$ 각각의 산호는 하나의 해, 하나의 배치를 뜻한다. 일단 암초를 정의하고 나면, 다음의 단계를 거친다.

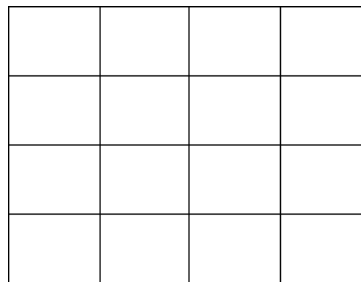


Figure 3. Grid(4X4)

1) 초기화

CRO 알고리즘의 시작은 암초에 있는 초기 산호의 수를 결정하는 것이다. 암초에서 비어있는 사각형과 사용중인 사각형의 비율을 결정하는 매개변수는 $\rho_0 (0 < \rho_0 \leq 1)$ 이다. 이를 이용해 초기 산호의 수 Initial Coral $s = N \times M \times \rho_0$ 가 결정된다. 산호의 수가 정해지면 암초 위에 랜덤한 배치를 갖는 해가 무작위로 자리를 잡는다. <Figure 4>의 경우 $\rho_0=0.6$ 을 가정했을 때 16개의 자리 중 10개가 초기해의 수로 지정되었고 각 초기해의 위치 또한 무작위로 결정된 것을 보여준다. 암초가 초기화되면 산호초는 번식하며 정지할 때까지 반복한다. 정지조건은 지정 횟수를 채우거나 일정 횟수동안 개선이 없으면 멈추는 것으로 한다. 앞

에서도 언급했듯이 각각의 산호는 UA-FLP 문제의 해를 뜻한다.

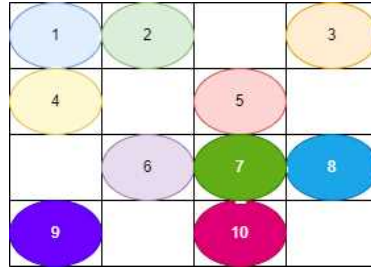


Figure 4. Initialization($\rho_0=0.6$)

2) 진화과정

번식 단계는 유성 생식(유성생식은 다시 내부, 외부 생식으로 나뉜다.)과 무성 생식으로 나뉘게 된다. 이 과정을 통해 현재 암초에 있는 산호들로부터 새로운 자식 산호를 만들어낸다. 유성 생식과 무성 생식 사이에는 착상단계가 존재하는데, 새로 만들어진 모든 산호가 암초에 들어갈 수 있지않고 그 중 일부만 착상된다. 마지막으로 퇴화단계를 통해 암초에서 가장 약한 산호를 제거한다. 유전 알고리즘과 비슷하게 건강하지 않은 산호는 없애고 건강한 산호를 통해 더 좋은 해를 생성하는 방식이다.

3) 외부 유성생식(Broadcast spawning)

이 과정은 두 단계로 이루어진다. 첫째, 암초에 존재하는 다수의 산호중 임의로 생식에 참여할 산호를 결정한다. 그 비율은 매개변수 F_b 로 결정하며 그 양은 (존재하는 산호의 수 \times 매개변수 F_b)다. 여기서 선택되지 않은 산호들(존재하는 산호의 수 $\times (1-F_b)$)은 나중에 내부 유성생식에 사용된다. 둘째, 선택된 산호중 임의로 두개씩 쌍을 짓는다. 각각의 쌍은 교차과정을 통해 자녀를 만든다. 우리는 여기서 Partially Matched Crossover(PMX)(Goldberg and robert, 1985), One point crossover(Holland, 1992)를 앞장에서 설명한 세가지 벡터집합에 대해 각각 사용했다. 즉, 세가지 벡터집합에 대해 각각 Crossover한 새로운 세가지 벡터집합이 형성되고 그 집합이 새로운 배치를 만들게 된다. 자녀가 생성되면 내부 유성생식이 끝날 때까지 따로 보관한다. <Figure 5>는 임의의 두 산호가 PMX Crossover를 사용해 자녀를 생성한 뒤 따로 보관하는 과정을 나타낸다.

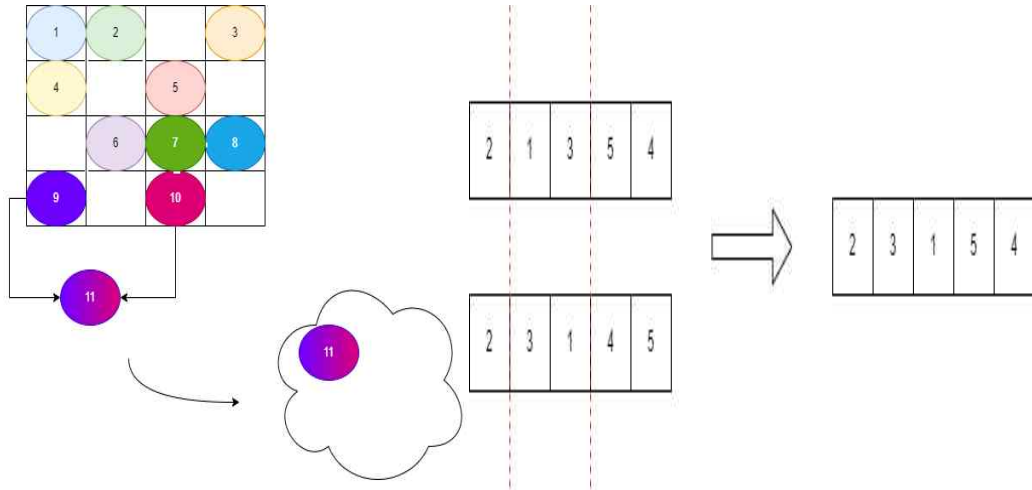


Figure 5. Broadcast spawning

4) 내부 유성생식(Brooding)

외부 유성생식에서 사용하고 남은 산호(존재하는 산호의 수 $\times (1-F_b)$)는 개별 산호에 대해 돌연변이 과정을 거친다. 돌연변이 과정은 Cut Point, Cut Code, Depart Sequence에 대해 Swap, Inversion을 사용한다. 유성생식이 끝나고 나면 암초에 착상하기 위한 준비를 한다. <Figure 6>은 외부 유성생식에 참여하지 않은 산호가 Swap 돌연변이를 일으켜 새로운 산호를 만들어 이전 단계의 자녀가 위치하는 곳에 함께 보관되는 모습을 표현했다.

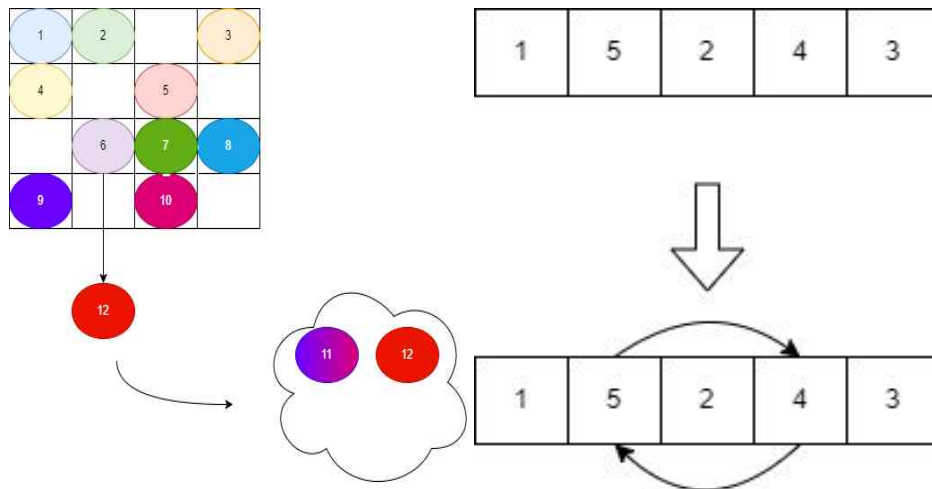


Figure 6. Brooding

5) 착상(Larvae setting)

앞의 모든 유성생식을 통해 만들어진 자식 산호들은 암초에 착상을 해야 살아갈 수 있는데, 새로운 산호들은 기존 암초의 임의의 위치에 다가간다. 만약 그 자리가 빈 자리일 경우 아무 조건 없이 착상되며 만약 그 위치에 이미 다른 산호가 존재한다면, 그 산호와 목적함수 값(물류이송거리)을 비교하여 새로운 산호의 값이 더 좋을 경우(더 낮은 물류이송거리를 가질 경우) 기존의 산호를 밀어내고 자리를 차지한다. 그러나 반대의 경우 몇 번의 과정을 반복한다(일반적으로 3회(Salcedo-Sanz et al., 2013)). 모든 암초위 자리를 한

번씩 돌아다닌 이후에도 착상하지 못한 자식 산호는 제거된다. <Figure 7>은 그 과정을 그림으로 표현했다. 초록색 산호는 기존에 있던 산호를 밀어내고 자리를 차지했으며 보라색 산호는 빈 자리를 찾아 착상했다. 빨간색 산호는 빈자리를 찾지 못하고 제거됐다.

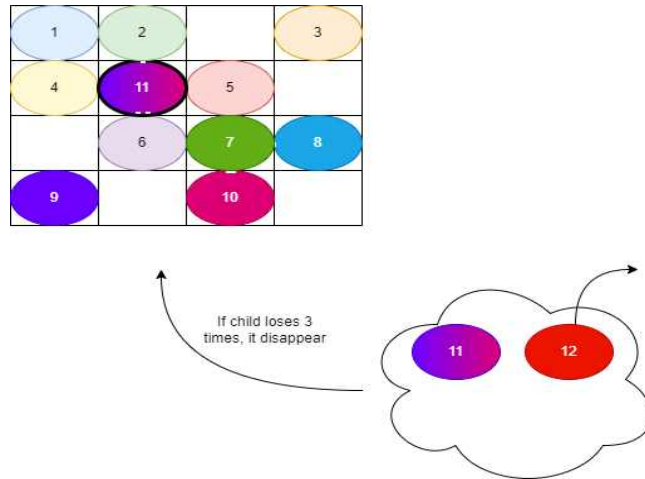


Figure 7. Larvae setting

6) 무성생식(Budding)

앞의 모든 과정을 끝내고 나면 암초는 부모 산호와 자식 산호, 둘연변이 산호가 섞여있는 형태다. 여기서 더욱 건강한 산호들만 다음 세대로 보내기 위해 산호는 자가복제를 한다. 모든 산호의 성능을 평가하여 목적함수 값에 따라 순위를 정하고 일정 비율(F_a)만큼 가장 건강한 산호를 복제한다. 그 후에 착상과정을 거쳐 암초에 들어오게 된다. 유성생식과의 차이점은 앞에서 생성된 자녀 산호는 자리를 찾지못해 죽을수도 있는 반면 복제된 산호는 가장 건강하기 때문에 무조건 착상이 된다. <Figure 8>는 가장 좋은 성능을 하진 하늘색 산호가 일부 복제되어 암초에 착상하는 과정을 보여준다.

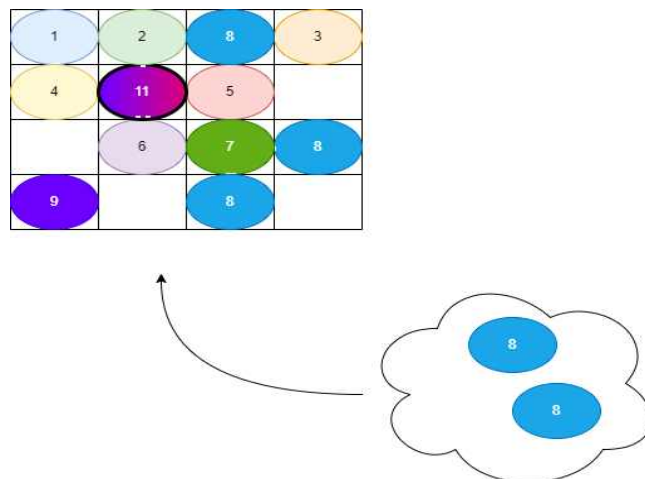


Figure 8. Budding

7) 퇴화(Depredation)

여기까지 알고리즘이 수행되면 암초에 존재하는 가장 나쁜 목적함수 값을 지닌 산호들은 매우 낮은 확률(P_d)로 따라 암초에서 제거된다. (P_d 는 0.1로 고정) 이때 제거될 산호를 고르는 방식은 무성생식과 마찬가지로 산호들을 목적함수 값에 따라 순위를 매긴 뒤 가장 나쁜 값을 가진 산호부터 순서대로 선택된다. 제거될 산호의 수는 매개변수 F_d 에 의해 결정된다(제거될 산호의 수 = 존재하는 산호의 수 $\times F_d$). <Figure 9>는 살구색 산호가 제거되는 모습을 보여준다. 제거되고 난 뒤의 빈공간은 다음 세대의 새로운 자식 산호를 위해 쓰인다. 정리하면 다양성을 위해 안좋은 해를 가진 산호도 다음 세대로 가져가지만 일정 확률에 따라 그런 산호들을 제거할 때도 있다. <Figure 10>은 지금까지의 내용을 정리한 Coral reef optimization의 플로우차트이다. CRO의 Psuedo-code는 Algorithm1과 같다.

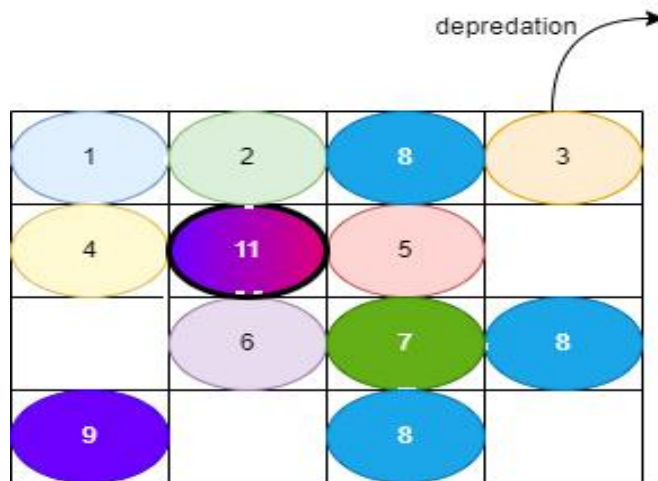


Figure 9. Depredation

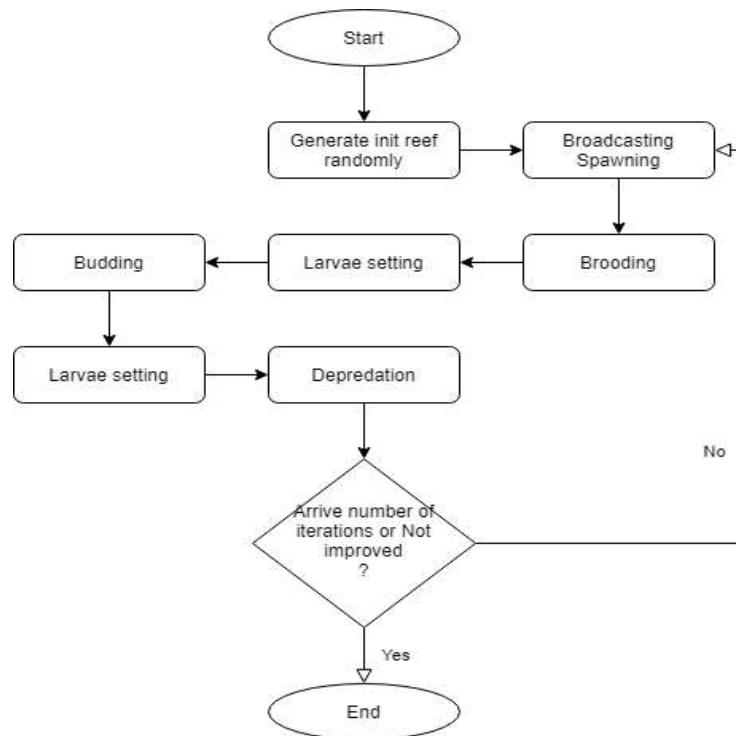


Figure 10. Flowchart of CRO

Algorithm 1 Psuedo-code of CRO

procedure CRO($N, M, iter_{max}, \rho_0, F_b, F_a, F_d, P_d$)**Initialize** set corals on coral reef with random position that equals to $\rho_0 \times N \times M$.

Evaluate initial population and select the one with the best fitness value.

while $iter \leq iter_{max}$ **or** not_update ≤ 1000 **do** generate new corals fraction F_b by **broadcast spawning** generate new corals fraction $1-F_b$ by **brooding** **larvae setting** generate best corals fraction F_a by **asexual reproduction** **depredation** of F_d worst reef corals with probability P_d **for** each coral i **do** **if** $OFV_i > OFV_{best}$ **then**

not_update = not_update + 1

else

not_update = 0

end for**end while****return** best feasible solution**end procedure**

2.3. 실험결과

이번 장에서는 UA-FLP를 STS 방식으로 해결한 기존의 연구들과 CRO의 결과를 비교한다. 코드는 JAVA로 작성되었으며, Intel Core i5 CPU(1.6 Ghz)와 8GB의 RAM, 그리고 Window 10 환경에서 실험하였다. 총 10개의 데이터셋을 이용했으며 이에 대한 정보는 <Table 1>에 정리되어 있다. 이 10개의 데이터셋은 UA-FLP 문제를 해결하기 위해 많은 연구자들이 실험을 하였는데 이 과정에서 주로 사용되어 온 데이터로 비교 가능한 정보를 제공하기에 이를 활용했다.

각각 이름, 블록 수, 공간 너비, 공간 길이, 가로세로 비율관계(Aspect ratio, α)이며 예외로 l_{min} 은 각 블록의 최소 길이를 뜻한다. 위의 문제들은 모두 블록들의 넓이 합계가 전체 면적과 동일하거나 거의 비슷한 문제이다.

본 연구는 CRO를 통해 각 문제의 물류이송거리를 최소로 만드는 배치를 찾고자 한다. 물류이송거리는 블록간 물동량 * 블록간 직각거리로 계산되는데 동일한 공간에 어떻게 배치하느냐에 따라 그 값이 크게 달라진다. 이때 목적함수를 총 물류이송거리로 하고 그 값이 낮을수록 더 좋은 배치라고 평가한다.

CRO는 매개변수의 값이 결과에 상당한 영향을 미친다. 따라서 적절한 매개변수를 결정해야하는데, 본 연구에서는 통계프로그램 R의 irace 패키지를 이용하여 그 값을 결정했다. 우리 문제의 매개변수는 총 4개(ρ_0, F_b, F_a, F_d)이고 그 값은 각각 (0.75, 0.53, 0.69, 0.54)이며 <Table 2>에 정리되어있다. Grid Size는 각각 3번에 걸쳐 시행했는데 (4X4, 7X7, 10X10)가 그 값이다. Grid의 Size를 나눠 실험한 이유는 CRO는 Population-based algorithm이기 때문에 Pop의 크기에 따라서도 그 결과에 영향을 미치기 때문이다. 실험은 각 문제에 대해 50번씩 진행되었으며 1번의 실험당 총 50000번의 iteration을 진행하되 1000번 이상 Best Solution의 변화가 없으면 중단했다. Grid의 사이즈에 따른 실험 결과는 <Table 3>, <Table 4>, <Table 5>에 정리되어 있다.

Table 1. Problem specification

Problem name	# of Fac.	W * H	Aspect ratio	Reference
fO7	7	8.54×13.00	$\alpha_{\max} = 5$	(Meller et al., 1998)
fO8	8	11.31×13.00	$\alpha_{\max} = 5$	(Meller et al., 1998)
fO9	9	12.00×13.00	$\alpha_{\max} = 5$	(Meller et al., 1998)
O7	7	8.54×13.00	$\alpha_{\max} = 4$	(Meller et al., 1998)
O8	8	11.31×13.00	$\alpha_{\max} = 4$	(Meller et al., 1998)
O9	9	12.00×13.00	$\alpha_{\max} = 4$	(Meller et al., 1998)
vC10s	10	25.00×51.00	$l_{\min} = 5$	(Van Camp et al., 1991)
vC10a	10	25.00×51.00	$\alpha_{\max} = 5$	(Van Camp et al., 1991)
Nug12	12	3.00×4.00	$\alpha_{\max} = 5$	(Nugent et al., 1968)
AB20	20	3.00×2.00	$\alpha_{\max} = 4$	(Armour and Buffa, 1963)

Table 2. Parameter of CRO from R irace

<i>Parameters</i>	<i>Range</i>	<i>Best Value</i>	<i>Definition</i>
ρ_0	[0.1, 0.9]	0.75	Initial ratio of coral reef
F_b	[0.1, 0.9]	0.53	Fraction of broadcast spawning
F_a	[0.1, 0.9]	0.69	Fraction of asexual-reproduction
F_d	[0.1, 0.9]	0.54	Fraction of depredation

Table 3. Result of Gird(4X4)

Problem name	Best OFV	Mean OFV	Avg Process time(sec)
fO7	17.75	17.81	0.4
fO8	22.39	23.56	0.6
fO9	23.46	24.75	0.9
O7	131.69	132.89	0.2
O8	243.14	247.20	0.2
O9	239.06	247.13	0.2
vC10s	21684.5	22968.23	0.3
vC10a	19468.48	20446.90	0.3
Nug12	257.5	261.54	0.4
AB20	5333.21	6082.73	1.3

* Bold numbers are same as best known OFV

Table 4. Result of Grid(7X7)

Problem name	Best OFV	Mean OFV	Avg Process time(sec)
fO7	17.75	17.81	0.6
fO8	22.39	23.33	0.4
fO9	23.46	24.65	0.3
O7	131.69	132.89	0.2
O8	243.14	247.48	0.2
O9	239.06	244.93	0.3
vC10s	20915.71	22816.91	0.3
vC10a	18522.79	20410.44	0.3
Nug12	257.5	262.86	0.3
AB20	5533.8	6021.19	1.3

* Bold numbers are same as best known OFV

Table 5. Result of Grid(10X10)

Problem name	Best OFV	Mean OFV	Avg Process time(sec)
fO7	17.75	17.83	0.7
fO8	22.39	23.53	0.3
fO9	23.46	24.60	0.3
O7	131.69	132.90	0.4
O8	243.14	247.09	0.3
O9	239.06	246.76	0.3
vC10s	22126.68	22906.28	0.3
vC10a	18522.79	20488.19	0.3
Nug12	257.5	264.21	0.4
AB20	5505.25	6010.15	1.4

* Bold numbers are same as best known OFV

2.3.1. 문제 해결 : 블록의 넓이 총합 = 전체 공간 넓이

O7, O8, O9, fO9, vC10a, vC10s, Nug12, AB20은 블록의 총넓이 합이 전체 공간의 넓이와 정확하게 일치한다. 즉, 주어진 문제에 빈 공간이 없다는 뜻이다. fO7~8 은 약간의 빈 공간이 존재하지만 목적함수에 큰 영향을 줄 정도는 아니고 그 값이 미비하기 때문에 별도의 더미변수는 사용하지 않았다.

vC10 문제는 조건에 따라 vC10a 와 vC10s 로 나뉜다. vC10a는 aspect ratio 조건이 존재하며 반대로 vC10s는 모든 블록에 대해 최소길이(l_{min}) 조건이 존재한다. AB20문제는 Armour and Buffa(1963)가 사용한 비대칭 flow matrix를 갖고 있다. 나머지 모든 문제는 대칭 flow matrix를 사용한다.

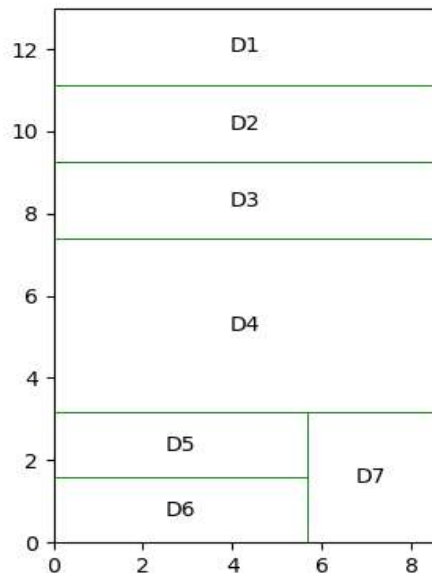
알고리즘의 적용성을 확인하기 위해 최적값과의 비교를 기본으로 한다. 다만 최적값이 알려져 있지 않은 경우 이전에 유사한 연구를 한 결과와의 비교를 통해 적용성을 평가하였다. 비교군은 3개의 그룹으로 나뉜다.(수학적 방법, 메타휴리스틱, 하이브리드) 수학적 방법의 경우 Castillo et al.(2005)가 쓰였고 메타휴리스틱의 경우 Goncalves and Resende(2015) 와 Komarudin and Wong(2010), Kulturel-Konak and Konak(2011b), Scholz et al.(2009)가 사용되었다. 마지막으로 하이브리드 방법은 Chang and Ku(2013)이 사용되었다. 결과는 <Table 6>에 정리되어있다.

보통 10개 미만의 블록을 배치하는 문제를 low size problem이라 한다. CRO는 다른 알고리즘과 마찬가지로 low size problem은 큰 무리없이 해결한다. 그러나 몇몇 문제의 경우 최적값에 도달하지 못했다.(O9) 30개 이하의 medium size problem에서는 vC10s와 AB20에서는 좋은 값을 도출하지 못했다. 그러나 전부 현재 가장 잘 알려진 값과 차이가 10% 미만이고 그간의 연구들보다 좋은 값인 경우도 있기 때문에 의미가 없다고 할 수는 없다. Grid size 차이에 따른 실험 결과를 <Table 2>, <Table 3>, <Table 4>에 정리했는데 그 결과를 살펴보면 사이즈가 클수록 평균 목적함수 값이 좋아진다. 일반적으로 Population-based algorithm은 Pop의 크기가 커질수록 한 세대에 해가 가질 수 있는 다양성이 커져서 더 좋은 결과를 보인다. 실제로 AB20의 실험결과를 살펴보면 Grid의 크기가 클수록 평균 OFV값이 낮아졌다. 큰 사이즈의 문제일수록 평균값이 차이나는 것으로 보아 추후 연구에서 Large size problem을 다룰 시 Grid의 크기를 키워야할 것이다. 또한 우리 연구의 가장 큰 장점은 그 수행시간에 있다. 상대적으로 열악한 실험환경에서 AB20 문제의 경우 Kang and Chae(2017)는 44.51초가 걸렸으며 같은 알고리즘을 사용한 연구인 Salcedo-Sanz et al.(2019)는 202초가 걸렸다(문제는 동일하지만 해결방식은 다름. 이전 연구는 FBS, 우리 연구는 STS를 사용함). 그에 반해 <Table 2>, <Table 3>, <Table 4>를 보면 위 방법으로는 평균 1.3초가 걸린다. 해의 성능이 시간에 비례하는 휴리스틱 방법의 특성상 위 결과는 의미가 있다. large size problem의 경우 그 시간이 더욱 단축될 것이다. 몇몇 문제에서 좋은 값이 나오지 못하는 이유는 빠르게 수렴해서이다. 따라서 추후 연구는 수렴을 천천히 하며 다양성을 늘리는 과정이 관건이 될 것이다. 각 문제의 평면 배치도는 <Figure 11> ~ <Figure 15> 에 나타나있다.

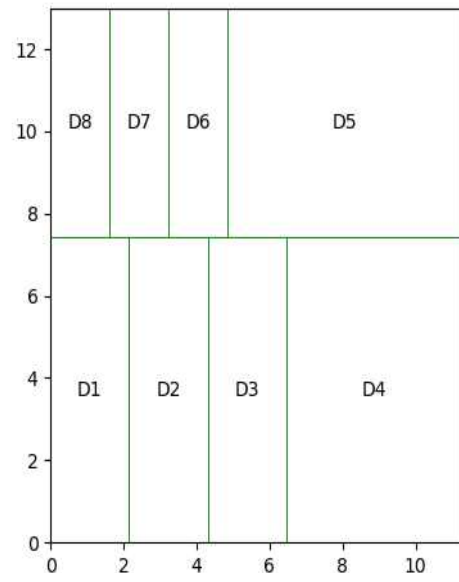
Table 6. Summary of the result with our approach

Research	Castillo et al., 2005	Scholz et al., 2009	Komarudin and Wong, 2010	Kulturel-Konak and Konak, 2010	Goncalves and Resnede, 2015	Chang and Ku, 2013	Kang and Chae, 2017	This approach 2021
Methodology	MILP	Tabu-Search	Ant Colony Optimization	Probabilistic Tabu Search	Biased random-key GA	QCP model based-Improved Harmony Search	Improved Harmony Search	CRO
Problem name	Best OFV							
f07	17.75						17.75	17.75
f08	22.31						22.39	22.39
f09	23.46						23.46	23.46
O7	131.58	132.00	131.68		131.56		131.69	131.69
O8	242.93	243.16	243.12		242.92	243.18	243.13	243.13
O9	236.14	239.07	236.14		236.57	236.14	236.14	239.06
vC10s	21,297.98	19,994.10	19,967.60	22,899.65	19,951.17	20,770.89	19,973.66	20915.71
vC10a				20,142.13		19,465.77	18,522.79	18522.79
Nug12				275.50			275.50	275.50
AB20		5225.96	4972.56	5336.36	5021.17	5151.86	4959.11	5333.21

* Bold numbers are same as best known OFV

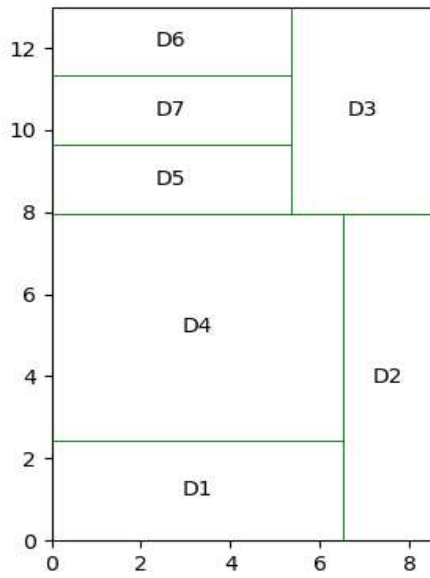


(a) fO7(OFV = 17.75)

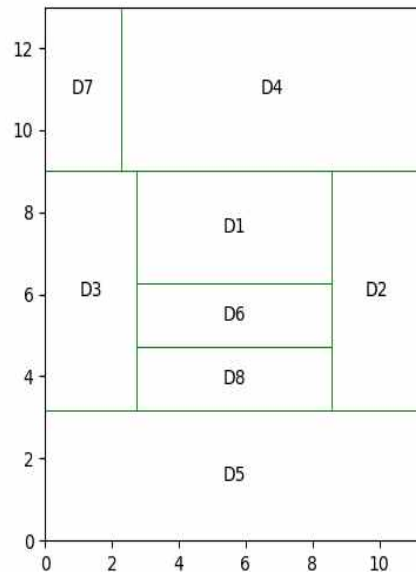


(b) fO8(OFV = 22.39)

Figure 11. Layout Configuration generated by CRO

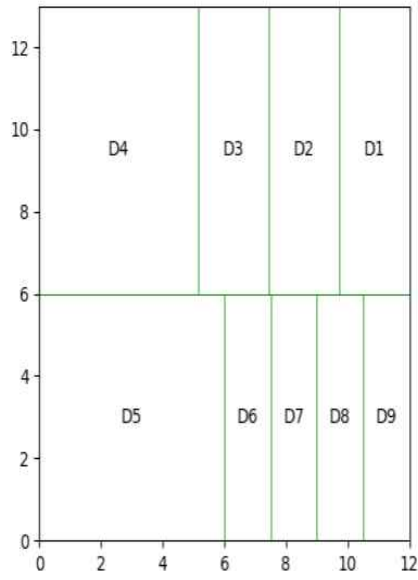


(a) O7(OFV = 131.69)

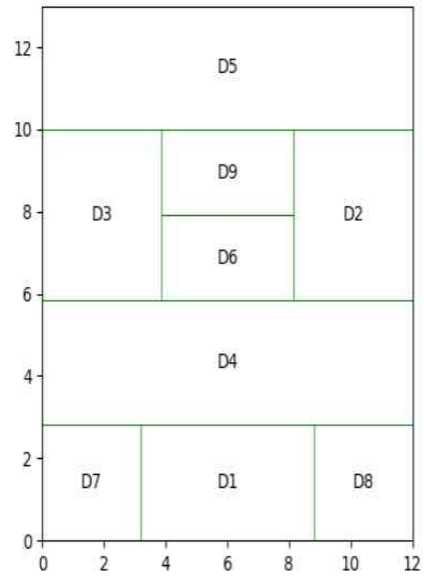


(b) O8(OFV = 243.13)

Figure 12. Layout Configuration generated by CRO

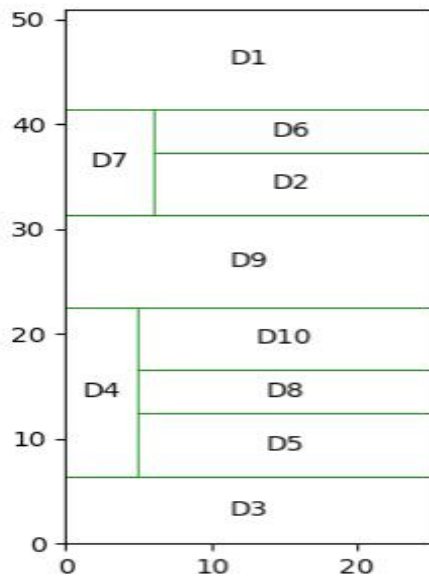


(a) fO9(OFV = 23.46)

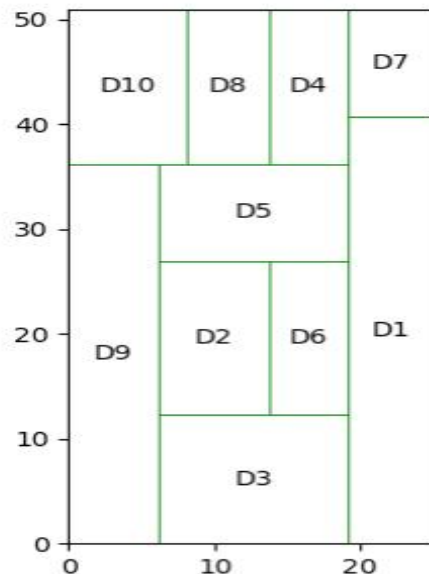


(b) O9(OFV = 239.06)

Figure 13. Layout Configuration generated by CRO

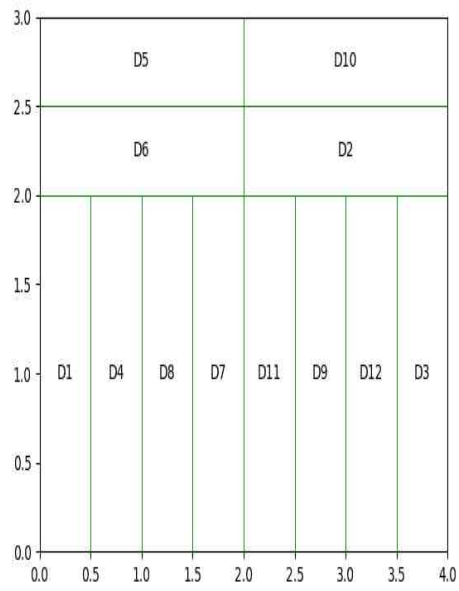


(a) vC10a(OFV = 18522.79)

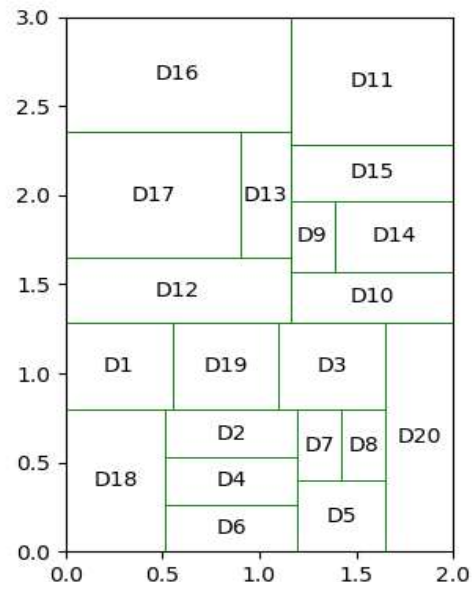


(b) vC10s(OFV = 20915.71)

Figure 14. Layout Configuration generated by CRO



(a) Nug12(OFV = 257.50)



(b) AB20(OFV = 5333.21)

Figure 15. Layout Configuration generated by CRO

3. 결론

본 연구는 STS를 기본으로 하여 산호초 최적화(CRO)를 UA-FLP에 적용 가능한지에 대한 평가를 수행했다. 제안된 CRO는 참고자료에서 추출한 10개의 데이터에 적용되었으며 그 성능은 1. 최적해가 알려져 있는 경우 이에 대한 비교, 2. 최적해가 알려져 있지 않은 경우 최근 연구자의 결과와 비교를 통해 분석되었다. CRO 알고리즘은 실험한 문제의 대부분 좋은 해를 도출했고(10개 중 7개의 최적해) 알려진 최선의 해에 도달하지 못하는 경우도 있었지만 비교적 빠르게 그 해를 찾는 것을 확인했다. 현재의 CRO는 초기 설계자의 기본 형태를 그대로 적용하여 UA-FLP에 활용 가능한지를 분석한 것인데 몇몇 문제를 잘 해결하는 것으로 보아 적용이 가능한 것으로 보인다. 다만 수행 시간에 관한 문제가 남아있다. 수행 시간이 빠르다는 것은 수렴도와 연관이 있다. 사이즈가 작은 문제의 경우 많은 iteration을 거치지 않더라도 최적해가 나올 수 있는 반면, 사이즈가 큰 문제에서는 충분한 횟수의 iteration이 진행되기 전에 수렴해 지역 해에 빠져 알고리즘이 종료된다. 따라서 이후에는 알고리즘 자체의 변화를 통해 UA-FLP에 맞는 형태로 변화시켜 수렴 속도를 줄이는 방향으로 진행해야 한다. 그 이후에는 변형된 형태의 CRO(Modified CRO) 연구로 더 큰 문제에도 적용 가능하도록 하는 것이 추후의 연구 방향이 될 것이다.

참고문헌

- Aiello, G., Enea, M., & Galante, G. (2006). A multi-objective approach to facility layout problem by genetic search algorithm and Electre method. *Robotics and Computer-Integrated Manufacturing*, 22(5-6), 447-455.
- Anjos, M. F., & Vieira, M. V. C. (2017). Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. In *European Journal of Operational Research* (Vol. 261, Issue 1).
- Bermejo, E., Chica, M., Damas, S., Salcedo-Sanz, S., & Cordón, O. (2018). Coral Reef Optimization with substrate layers for medical Image Registration. *Swarm and Evolutionary Computation*, 42.
- Gordon C, & Elwood, A. ! (1963). A Heuristic Algorithm and simulation approach to relative location of facilities - Combinatorial Difficulty of the Problem. *Management Science*, 9(2).
- Azadivar, F., & Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17).
- Camacho-Gómez, C., Wang, X., Pereira, E., Díaz, I. M., & Salcedo-Sanz, S. (2018). Active vibration control design using the Coral Reefs Optimization with Substrate Layer algorithm. *Engineering Structures*, 157.
- Castillo, I., & Westerlund, T. (2005). An ε -accurate model for optimal unequal-area block layout design. *Computers and Operations Research*, 32(3).
- Chae, J., & Regan, A. C. (2016). Layout design problems with heterogeneous area constraints. *Computers and Industrial Engineering*, 102.
- Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2).
- Durán-Rosal, A. M., Gutiérrez, P. A., Salcedo-Sanz, S., & Hervás-Martínez, C. (2018). A statistically-driven Coral Reef Optimization algorithm for optimal size reduction of time series. *Applied Soft Computing Journal*, 63.
- Enea, M., Galante, G., & Panascia, E. (2005). The facility layout problem approached using a fuzzy model and a genetic search. *Journal of Intelligent Manufacturing*, 16(3).
- Garcia-Hernandez, L., Arauzo-Azofra, A., Salas-Morera, L., Pierreval, H., & Corchado, E. (2013). Recycling Plants Layout Design by Means of an Interactive Genetic Algorithm. *Intelligent Automation and Soft Computing*, 19(3).
- Garcia-Hernandez, L., Salas-Morera, L., Garcia-Hernandez, J. A., Salcedo-Sanz, S., & Valente de Oliveira, J. (2019). Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. *Expert Systems with Applications*, 138.
- Geem, Z. W., Kim, J. H., & Loganathan, G. v. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76(2).
- Goldberg, D. E., & Lingle, R. (1998). Alleles, loci. and the traveling salesman problem. In *Evolutionary Computation: The Fossil Record*.

- Gómez, A., Fernández, Q. I., de la Fuente García, D., & García, P. J. (2003). Using genetic algorithms to resolve layout problems in facilities where there are aisles. *International Journal of Production Economics*, 84(3).
- Gonçalves, J. F., & Resende, M. G. C. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research*, 246(1).
- Hosseini-Nasab, H., Fereidouni, S., Fatemi Ghomi, S. M. T., & Fakhrzad, M. B. (2018). Classification of facility layout problems: a review study. *International Journal of Advanced Manufacturing Technology*, 94(1-4).
- John, H. (1992). *Holland, Adaptation in natural and artificial systems*. Ann Arbor MI University of Michigan Press, Ann Arbor.
- Kang, S., & Chae, J. (2017). Harmony search for the layout design of an unequal area facility. *Expert Systems with Applications*, 79.
- Komarudin, & Wong, K. Y. (2010). Applying Ant System for solving Unequal Area Facility Layout Problems. *European Journal of Operational Research*, 202(3).
- Konak, A., Kulturel-Konak, S., Norman, B. A., & Smith, A. E. (2006). A new mixed integer programming formulation for facility layout design using flexible bays. *Operations Research Letters*, 34(6).
- Kouvelis, P., Kurawarwala, A. A., & Gutiérrez, G. J. (1992). Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63(2).
- Kulturel-Konak, S., & Konak, A. (2011). Unequal area flexible bay facility layout using ant colony optimisation. *International Journal of Production Research*, 49(7).
- Kulturel-Konak, S., & Konak, A. (2011). A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Engineering Optimization*, 43(12).
- Liu, J., & Liu, J. (2019). Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Applied Soft Computing Journal*, 74.
- Liu, Q., & Meller, R. D. (2007). A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. *IIE Transactions (Institute of Industrial Engineers)*, 39(4).
- Medeiros, I. G., Xavier, J. C., & Canuto, A. M. P. (2015). Applying the Coral Reefs Optimization algorithm to clustering problems. *Proceedings of the International Joint Conference on Neural Networks*, 2015-September.
- Meller, R. D., Narayanan, V., & Vance, P. H. (1998). Optimal facility layout design. *Operations Research Letters*, 23(3-5).
- Montreuil, B. (1991). *A Modelling Framework for Integrating Layout Design and flow Network Design*.
- Paes, F. G., Pessoa, A. A., & Vidal, T. (2017). A hybrid genetic algorithm with decomposition phases for the Unequal Area Facility Layout Problem. *European Journal of Operational Research*, 256(3).
- Palomo-Romero, J. M., Salas-Morera, L., & García-Hernández, L. (2017). An island model genetic

- algorithm for unequal area facility layout problems. *Expert Systems with Applications*, 68.
- Purnomo, M. R. A., & Wiwoho, Y. S. (2016). Multi-objective Mixed Integer Programming approach for facility layout design by considering closeness ratings, material handling, and re-layout cost. *IOP Conference Series: Materials Science and Engineering*, 105(1).
- Salcedo-Sanz, S., del Ser, J., Landa-Torres, I., Gil-López, S., & Portilla-Figueras, J. A. (2014). The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *Scientific World Journal*, 2014.
- Salcedo-Sanz, S., Camacho-Gómez, C., Magdaleno, A., Pereira, E., & Lorenzana, A. (2017). Structures vibration control via Tuned Mass Dampers using a co-evolution Coral Reefs Optimization algorithm. *Journal of Sound and Vibration*, 393.
- Saraswat, A., Venkatadri, U., & Castillo, I. (2015). A framework for multi-objective facility layout design. *Computers and Industrial Engineering*, 90.
- Scholz, D., Petrick, A., & Domschke, W. (2009). STaTS: A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research*, 197(1).
- Shayan, E., & Chittilappilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure. *International Journal of Production Research*, 42(19).
- Sherali, H. D., Fraticelli, B. M. P., & Meller, R. D. (2003). Enhanced model formulations for optimal facility layout. *Operations Research*, 51(4).
- Sikaroudi, A. M. E., & Shahanaghi, K. (2016). Facility layout by collision detection and force exertion heuristics. *Journal of Manufacturing Systems*, 41.
- Tam, K. Y. (1992). A simulated annealing algorithm for allocating space to manufacturing cells. *International Journal of Production Research*, 30(1).
- Tate, D. M., & Smith, A. E. (1995). Unequal-area facility layout by genetic search. *IIE Transactions (Institute of Industrial Engineers)*, 27(4).
- Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2011). *Facilities planning - 4th edition*. *International Journal of Production Research*, 49(24).
- Ulutas, B. H., & Kulturel-Konak, S. (2013). Assessing hypermutation operators of a clonal selection algorithm for the unequal area facility layout problem. *Engineering Optimization*, 45(3).
- Wong, K. Y., & Komarudin. (2010). Solving facility layout problems using Flexible Bay Structure representation and Ant System algorithm. *Expert Systems with Applications*, 37(7).
- Wu, Y., & Appleton, E. (2002). The optimisation of block layout and aisle structure by a genetic algorithm. *Computers and Industrial Engineering*, 41(4).
- Yan, C., Ma, J., Luo, H., & Patel, A. (2019). Hybrid binary Coral Reefs Optimization algorithm with Simulated Annealing for Feature Selection in high-dimensional biomedical datasets. *Chemometrics and Intelligent Laboratory Systems*, 184.
- Yang, Z., Zhang, T., & Zhang, D. (2016). A novel algorithm with differential evolution and coral reef optimization for extreme learning machine training. *Cognitive Neurodynamics*, 10(1).