

# SCGA를 이용한 컨테이너의 최적적하 계획

김길태\* · 김도웅\* · 권봉재\* · 유성호\*\* · 진강규\*\*\*

\*한국해양대학교 대학원, \*\*한국해양대학교 기계·정보공학부

## Optimal Container Load Planning Using a SCGA

G. T. Kim\* · D. U. Kim\* · B. J. Kwon\* · S. H. Yoo\* · G. G. Jin\*\*

\*Graduate School, Korea University, Busan 606-791, Korea

\*\*Division of Mechanical and Information Eng., Korea Maritime University, Busan 606-791, Korea

**ABSTRACT** : The container load planning is one of key factors for efficient operations of handling equipments at container ports. This paper presents the problem of determining the optimal loading sequence of the containers which a straddle crane (S/C) picks up for minimizing its total travel distance. To obtain a good solution with considerably small effort, a heuristic search procedure based on a symbolic genetic algorithm is developed. The performance of the proposed algorithm is demonstrated through a test problem.

**KEY WORDS** : container load planning, genetic algorithm, containership

### 1. 서 론

최근 국제간의 수출입 화물의 증가로 컨테이너 물동량을 신속, 정확하게 처리해주는 양적하 서비스가 터미널의 경쟁요소로 부각되면서 항만간에 경쟁이 더욱 심화되고 있다. 물류비용을 절감하고, 서비스 질을 높여 항만의 경쟁력과 생산성을 높이는 방안으로 첨단 설비를 확충하고 신설하는 문제를 생각할 수 있겠지만, 이는 대규모의 예산을 필요로 하므로 현실적으로 쉽지 않다. 이러한 점을 고려해 볼 때 소규모 비용으로 단기간에 경쟁력을 갖출 수 있는 방안으로는 터미널의 운영방법을 개선하거나 제한된 터미널 장비를 효율적으로 사용하는 것이라 할 수 있다. 이와 관련된 방법으로 수출 컨테이너의 양적하 순서 결정[1-4], 컨테이너 터미널 장비의 할당[5], 수출장치의 배정 계획[6,7], 선석배정(Berth allocation) 계획[8,9], 컨테이너 적하작업 시 재취급의 최소화[10] 등이 모색되어 왔다.

Gifford[1]는 컨테이너 하역 작업부두에서 컨테이너선의 선적 계획을 위한 학습적 절차를 제안하였고, 컨테이너선의 적하 계획을 위한 방법론을 제안하면서 Cho[2]는 Integer programming 기법을 적용하고 있으며, Kim 등[3]은 유전알고리즘을 적용하는 문제를 검토하였고, 이 연구는 Kim[4]에 의해 더욱 체계적으로 다루어 졌다. 한편 Peterkofsky와 Daganzo[5]는 그레인의 스케줄링 문제를 다루었고, Chan과 Imai[8]는 다수 선박의 선석배정 문제에 유전알고리즘을 적용하였고, Nishimura 등[9]은 선박의 동적 선석할당 문제를 다루었고, Watanabe[10]

는 컨테이너의 장치장에서 재취급 횟수를 추정하는 간단한 방법을 제시하였다.

특히 터미널에서 컨테이너 선적작업 시 적하 순서를 적절히 정하는 문제는 장비운용의 효율성에 크게 영향을 준다. 본 연구에서는 자동 컨테이너 최적 위치 결정 시스템 및 자동 적하 계획시스템 개발의 일환으로, 컨테이너 취급장비의 총 이동거리를 최소화하는 방법으로 최적 적하계획을 결정한다. 이는 대표적인 NP 문제로 비록 잠정적인 해의 수가 유한한 곳에서 해를 찾게 되지만 다루는 변수의 수가 많아지면 기존의 방법으로는 해결하기가 어렵다. 최근 NP 문제들을 푸는 도구로서 유전알고리즘을 이용하는 연구가 점점 확대되고 있으므로 본 연구에서도 기호코딩 유전알고리즘을 개발하고 이를 이용한다. 개발된 프로그램을 컨테이너 터미널에서 다루어지는 것과 유사한 문제에 적용하고 그 유효성을 검증한다.

### 2. 컨테이너 적하계획의 정형화

#### 2.1 컨테이너의 양적하

컨테이너선의 양적하 작업에는 갠트리 크레인(Gantry crane: G/C), 스트래들 캐리어(Straddle carrier: S/C), 트랜스퍼 크레인(Transfer crane: T/C), 야드 트랙터/트레일러(Yard tractor/trailer: Y/T) 등과 같은 장비들이 이용되고 있다. G/C는 선박에 근접하여 양적하 작업을 하도록 설계된 장비이고, Y/T는 작업용 컨테이너 운반트럭이고, S/C와 T/C는 야드 내에서 컨테이너를 다단적하는 장비로서 Y/T로부터 컨테이너를

받아 쌓고 또 Y/T에 실어주는 역할을 한다. Fig. 1은 컨테이너 전용 하역장비[11]를 보여주고 있다.

선박에 컨테이너를 양적하할 때 야드 또는 선박에서 발생하는 재취급을 줄이기 위해서는, 사전에 치밀한 작업 계획을 세우는 것이 필요하다. 플래너는 양적하 컨테이너의 수량, 접안 선석의 상황, 계획 기간 중 투입가능한 G/C 상황 등을 고려하여 투입할 G/C 대수와 작업시간 등을 정하고 작업계획을 작성하게 된다. 이 때 투입한 G/C 별로 양적하 물량을 균등하게 나누고 서로간에 간섭이 적도록 계획하는 것이 필요하다. 작업 스케줄상에는 Ship-bay에 선적되고 양하되어 질 컨테이너의 번호, 슬롯(slot)의 위치, 컨테이너의 작업 순서 등이 나타나 있다. Fig. 2와 Fig. 3은 컨테이너선의 구조와 컨테이너들이 적하될 적하도면을 보여주고 있으며, Table 1은 간단한 작업 계획표이다.

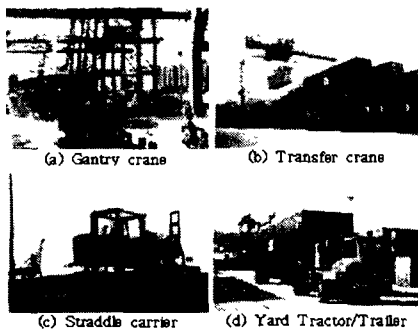


Fig. 1 Container handling systems

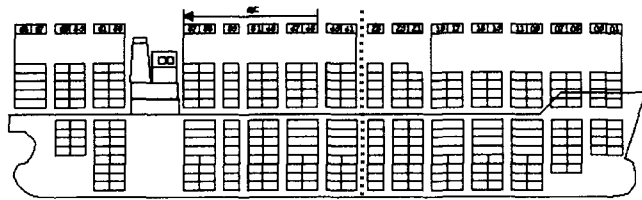


Fig. 2 Structure of a containership

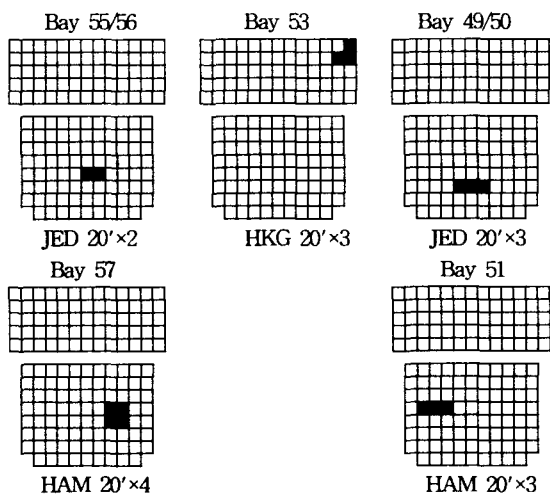


Fig. 3 Loading map of a containership

Table 1 Loading job schedule

Job sequence	1	2	3	4	5
Group	JED 20	HAM 20	HKG 40	JED 20	HAM 20
Containers in storage	3	3	3	2	4
Location	Bay 49/50	Bay 51	Bay 53	Bay 55/56	Bay 57

Table 1의 작업 순서에 의하면 먼저 “JED 20” 그룹의 컨테이너 3개를 Bay 49/50에, “HAM 20” 그룹의 컨테이너 3개를 Bay 51에, “HKG 40” 그룹의 컨테이너 3개를 Bay 53 순서로 총 15개를 취급하게 된다. 따라서 컨테이너 하역장치의 첫 번째 경유지는 “JED 20” 그룹의 컨테이너 3개를 선적하기 위해 장치장을 방문하는 순서와 일치한다. Fig. 4는 적하작업을 도식적으로 보여준다. 작업 순서에 따라 수출 컨테이너를 컨테이너선에 적하할 경우에는 S/C나 T/C로 컨테이너를 집어 올려 Yard-bay의 끝 부분으로 옮겨 Y/T에 실는다. 그러면 Y/T는 이를 G/C까지 운송하고, G/C는 컨테이너를 들어올려 적하하게 된다. 본 연구에서는 컨테이너 하역장비로 G/C, S/C, Y/T가 사용되는 것으로 가정한다.

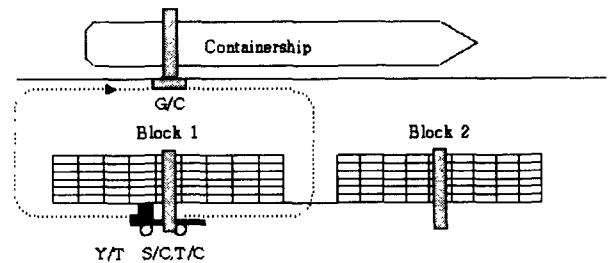


Fig. 4 Container loading using S/C or T/C

2.2 문제의 정형화

컨테이너의 적하순서는 S/C가 컨테이너를 Y/T에 실는 과정에서 발생하는 야드 내 장치장 사이를 이동하는 총 이동거리에 영향을 주므로, 이를 최소화하여 작업 능력을 높이는 것은 대단히 중요하다. 여기서는 S/C의 이동 거리를 최소화하도록 컨테이너의 적하 순서 문제를 정형화하고 해를 얻는다. 이때 같은 장치장(같은 번호) 내에서 이동거리는 처리 순서와는 관계없이 일정하므로, 장치장 사이를 움직인 거리만을 고려한다. 다루는 문제는 앞서 수행한 연구[1-4]와 유사하며 1) 적하계획은 작업계획표에 대응하는 G/C의 조건을 만족시키고, 2) 각 장치장에서 선택되는 각 그룹의 전체 컨테이너 수가 대응되는 그룹의 초기 컨테이너 수와 일치한다라는 두 가지 제약 조건을 고려하여 정형화한 결과를 요약 정리하면 다음과 같다[4].

주어진 문제에서 사용될 기호정의는 다음과 같다.

- m: S/C가 전체 작업을 구성하는 Subtour의 수
- n: Yard-bay 수
- l: 컨테이너 그룹 수
- B: Yard-bay 번호로 구성되는 집합, B= {1,2,3,...,n}
- G: 컨테이너 그룹 번호로 구성되는 집합, G= {1,2,3,...,l}

$C_{hj}$ : Yard-bay  $j$ 에 쌓여있는 그룹  $h$ 의 초기 컨테이너의 수  
 $S(h)$ : 컨테이너 그룹이  $h$ 인 Subtour의 집합  
 $r_t$ : Subtour  $t$  동안 적하되는 컨테이너의 수  
 $d_{ij}$ : Yard-bay  $i$ 와  $j$  사이의 이동 거리  
 $A(B) = \{(i,j) \mid i,j \in B\}$   
 $S, T$ : 시작과 종료 노드  
 $t$ : Subtour 번호

$$y_{ij}^t = \begin{cases} 1, \text{ 만약 } S/C \text{가 subtour } t \text{를 완성한 후 yard-bay } i \text{와 } j \\ \text{사이를 움직이면} \\ 0, \text{ 아니면} \end{cases}$$

$$z_{ij}^t = \begin{cases} 1, \text{ 만약 } S/C \text{가 subtour } t \text{ 동안 yard-bay } i \text{와 } j \\ \text{사이를 움직이면} \\ 0, \text{ 아니면} \end{cases}$$

$x_j^t$ : Subtour  $t$  동안 Yard-bay  $j$ 에서 적하되는 컨테이너 수

따라서 컨테이너 적하 문제의 목적함수는 식 (1)로 표시되고

$$F = \sum_{t=0}^{m-1} \sum_{(i,j) \in A(B)} d_{ij} y_{ij}^t + \sum_{t=1}^m \sum_{(i,j) \in A(B)} d_{ij} z_{ij}^t \quad (1)$$

다음과 같은 제약성 최적화 문제로 정형화된다.

Minimize  $F$

제약조건:

$$\sum_{j \in B} y_{sj}^0 = 1$$

$$-\sum_{i \in B} y_{it}^m = -1$$

$$\sum_{j \in B} (y_{ji}^{t-1} + z_{ji}^t) - \sum_{j \in B} (y_{ij}^t + z_{ij}^t) = 0, i \in B, t \in [0, m]$$

$$\sum_{(i,j) \in A(w)} z_{ij}^t = |w| - 1, \forall w \subseteq B, t \in [1, m]$$

$$x_j^t \leq M \left( \sum_{i \in B} z_{ij}^t + \sum_{i \in B} y_{ij}^{t-1} \right), j \in B, t \in [2, m]$$

$$x_j^1 \leq M \left( \sum_{i \in B} z_{ij}^1 + \sum_{i \in B} y_{sj}^0 \right), j \in B, t = 1$$

$$\sum_{i \in B} x_j^t = r_t, t \in [1, m]$$

$$\sum_{t \in S(h)} x_j^t = c_{hj}, h \in G, j \in B$$

$$y_{ij}^t \in \{0, 1\}, i, j \in B, t \in [1, m-1]$$

$$y_{sj}^0 \in \{0, 1\}, j \in B$$

$$y_{it}^m \in \{0, 1\}, i \in B, t = m$$

$$z_{ij}^t \in \{0, 1\}, i, j \in B, t \in [1, m]$$

$$x_j^t \geq 0, j \in B, t \in [1, m]$$

단,  $|w|$ 는 집합  $w$ 의 원소크기,  $M$ 은 충분히 큰 수이다.

위에서도 짐작할 수 있듯이 컨테이너의 적하계획은 비선형 조합 최적화 문제가 되어 전통적인 방법으로는 해결하기 어려우므로 GA 프로그램을 개발하고 이용한다.

### 3. 컨테이너 적하계획을 위한 유전알고리즘

유전알고리즘(Genetic algorithm: GA)는 매우 복잡한 최적화 문제를 다룰 수 있는 메커니즘을 제공함으로써 지난 10년간 신호처리, 시스템 식별 및 제어, 신경회로망의 설계 및 학습, VLSI 설계, 단백질 분자의 구조 결정, TSP 및 작업 스케줄링 등 여러 분야에 응용되어 왔다[11,14]. GA를 이용할 때 기본적으로 요구되는 전제는 초기집단을 구성하는 것이다. 집단은 잠정적인 해를 의미하는 다수의 개체로 형성되고 이들은 유전자의 역할을 반영하도록 흔히 자연 파라미터 형태로 표현된다. 본 연구에서 GA의 역할은 최적화 문제의 해를 얻는 것이므로 기호코딩을 채용하는 기호코딩 유전알고리즘(Symbolic-coded genetic algorithm: SCGA)을 사용한다.

#### 3.1 염색체의 표현 및 초기집단의 생성

GA는 해공간 상의 파라미터를 부공간 상의 염색체로 변환하여 사용해 왔고 여기에는 이진코딩이 주로 이용되어 왔다. 그러나 조합 최적화 문제에 이진코딩을 적용하면 기존의 일점 교배나 단순돌연변이는 적법하지 못한 해를 만들게 되어, 그 대안으로 기호코딩이 제안되어 왔다. SCGA는 기호코딩을 채용하므로 염색체의 유전자들은 특정 컨테이너를 의미하는 문자로 구성된다. 그러나 기존의 코딩법과의 차이점은 염색체가 각 작업순서와 컨테이너 그룹(목적지와 크기가 같은 컨테이너)과 관련있는 부스트링들의 조합으로 구성된다는 것이다. Fig. 5는 이를 보여주고 있다.

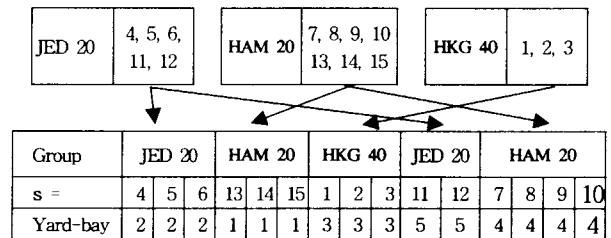


Fig. 5 Group-based chromosome

탐색공간 상의 점들을 표현하는 방법이 결정되면 다음은 초기 개체집단의 생성이 요구된다. 집단은 개체들이 진화해 가는 도중에 총체적 기억장소로서의 역할을 담당하게 되고, 많은 세

대를 거치는 동안에 좋은 해의 유전자를 보유하게 된다. k세대에서의 집단 P(k)는 염색체로 특정지워지는 N개 개체들의 집합으로 정의된다. 본 연구에서 초기집단 P(0)은 무작위법으로 초기화 하도록 한다. 앞에서 잠시 언급하였듯이 염색체는 해당 그룹의 컨테이너를 의미하는 부스트링의 조합으로 구성되기 때문에, 각 부스트링 내의 유전자들은 해당 그룹에 속한 컨테이너 번호 중에서 무작위로 선택하여 초기화한다.

**3.2 유전 연산자**

**(1) 재생산(Reproduction)**

재생산은 적자생존(Survival of the fittest)의 현상을 모방한 연산자로서 적합도값을 기반으로 집단 내의 개체들을 선택하여 교배급원을 형성한다. 재생산을 알고리즘형태로 구현한 것으로는 룰렛휠선택, 순위에 기초한 선택, 토너먼트선택, 교체를 수반한 잔여확률표본선택 등이 있으며, SCGA는 교체를 수반한 잔여확률표본선택을 사용한다. 이 연산자를 통해 집단 내 각 염색체는 기대치의 정수값 만큼은 우선 배정받고, 나머지 숫자만큼은 소수점 이하의 크기를 가지고 확률적으로 경쟁하게 된다.

**(2) 교배(Crossover)**

교배는 자연계의 유성생식을 알고리즘 형태로 흉내낸 것으로서 탐색공간 상의 새로운 점을 찾기 위해 부모 염색체 쌍을 임의로 선택하고, 교배점 이후의 유전자들을 서로 교환 결합함으로써 자손을 생성한다. 이러한 연산은 교배된 자손의 수가 어버이 집단의 크기와 같을 때까지 반복된다. 조합 최적화 문제에 적용할 수 있는 연산자로는 부분사상교배(PMX), 순서교배(OX), 순환교배(CX), 위치기반교배 등[16]이 있다. 비록 이들이 비적법성(Illegitimacy) 문제를 복구해주는 특별한 메커니즘을 가질지라도, 그룹기반의 염색체 표현을 채용하고 있는 본 연구의 특성상 PMX 외는 비적법성 문제가 계속 발생될 수 있으므로 SCGA는 PMX를 이용한다. PMX는 먼저 임의로 선택된 두 절단점 내의 부스트링(교차영역)을 서로 교환하고, 다음 교차영역으로부터 얻어지는 사상관계에 따라 나머지 유전자를 정한다. 이 연산자는 Fig. 6과 같이 4단계를 통해 이행된다.

**단계 1:** 두 절단점이 부모 염색체 내에서 임의로 선택된다.

	A		B			C			A		B				
s <sub>1</sub> (k)=	5	4	6	9	7	15	2	1	3	12	11	14	8	13	10
s <sub>2</sub> (k)=	6	12										7	13	10	14

**단계 2:** 절단점 사이의 부스트링을 서로 교환한다.

s <sub>1</sub> (k)=	x	x										x	x	x	x
s <sub>2</sub> (k)=	x	x	6	9	7	15	2	1	3	12	11	x	x	x	x

**단계 3:** 교차영역으로부터 사상관계를 얻는다.

$$6 \leftrightarrow 11 \leftrightarrow 4, 8 \leftrightarrow 9 \leftrightarrow 15 \leftrightarrow 7, 1 \leftrightarrow 2, 3 \leftrightarrow 3, 5 \leftrightarrow 12$$

**단계 4:** 나머지는 사상관계에 따라 자손을 만든다. 먼저 자손 s<sub>1</sub>(k)의 1번 위치에 12가 온 것은 5↔12의 관계 때문이다. 2번과 13번 위치가 6, 7으로 교체된 것은 각각 4↔11↔6, 8↔9↔15↔7의 관계 때문이다. 나머지 위치는 어떤 사상관계도 갖고 있지 않기 때문에 자신이 된다. 자손 s<sub>2</sub>(k)도 비슷한 방법으로 만들어진다.

	A		B			C			A		B				
s <sub>1</sub> (k)=	12	6										14	7	13	10
s <sub>2</sub> (k)=	4	5	6	9	7	15	2	1	3	12	11	8	13	10	14

Fig. 6 PMX

**(3) 돌연변이(Mutation)**

모의진화 초기에 소수의 초우량 개체가 집단을 지배함으로써 유전적 다양성이 결핍되거나, 모든 염색체의 특정 위치가 같은 유전자로 고정되게 되면 준최적해(Suboptimal solution)나 사점(Dead corner)에 빠지게 되는 요인이 된다. 이러한 원치 않는 해로부터 벗어나고 탐색영역을 확대해주기 위한 한 메커니즘이 돌연변이이다. SCGA에는 두 가지 돌연변이 연산자가 사용된다.

**가. 교체돌연변이(Swapping mutation)**

교체돌연변이는 염색체 내에서 두 유전자를 임의로 취하고 이들을 서로 교환한다. 이 때 염색체의 적법성이 유지되도록 유전자들은 같은 그룹 내에서 선택된다. Fig. 7은 교체돌연변이의 예를 보여주고 있다.

	A		B			C			A		B				
s <sub>1</sub> (k)=	12	6	11		15	9	1	2	3	5	4	14	7	13	10
s <sub>1</sub> (k)=	12	6	11	7	15	9	1	2	3	5	4	14		13	10

Fig. 7 Swapping mutation

**나. 역치(Inversion)**

역치는 염색체안에서 두 역치점을 임의로 선택하고, 선택된 부스트링을 역순으로 재배치시킨다. 비적법성 문제가 일어나지 않도록 역치점은 그룹 내에서 선택되어 진다. Fig. 8은 역치의 한 예를 보여준다.

	A		B			C			A		B				
s <sub>1</sub> (k)=	12		7	15	9	1	2	3			14	8	13	10	
s <sub>1</sub> (k)=	12	4	5	7	15	9	1	2	3	11	6	14	8	13	10

Fig. 8 Inversion

**3.3 적합도 평가**

유전 연산자를 거쳐 새로운 집단이 완성될 때마다 개체들의 적합도(Fitness)가 평가되는데 이는 대개 목적함수로부터 계산

된다. 환경에 대한 생물의 적합도는 장점을 의미하고 크면 클수록 더 많은 보상을 받듯이, SCGA에서도 적합도가 큰 개체가 더 많은 보상을 받을 수 있도록 적합도 함수는 최대화 문제 형태로 기술되어야 하고 음의 값을 가져서는 안 된다. 이 때문에 적합도를 계산하기 위해서는 적절한 변환이 필요하다.

$$f(\mathbf{s}_i(k)) = F_i + \gamma \quad (3)$$

여기서  $F_i$ 는  $i$ 번째 염색체의 목적함수이고,  $\gamma$ 는  $\forall k \in [0, \infty)$ 에 대해  $f(\mathbf{s}_i(k)) \geq 0$ 의 관계를 보장하도록 결정되는 상수이다.

### 3.4 기타 전략

#### (1) 스케일링 윈도우(Scaling window)

만일 세대 초기에 집단에 초우량 개체가 출현하게 되면 재생산 연산자는 이 개체를 여러 번 복제하게 되고 그 결과 유전적 다양성이 결핍되어 지역 최적점으로 수렴하게 된다. 이와 반대로 세대 후기에 개체들이 한 점 주위로 군집되면 좋은 개체와 더 나은 구분하는 것이 어렵게 된다. 따라서 초기에는 개체간의 상대적인 적합도의 영향을 줄여주고, 후기에는 크게 해 줄 필요가 있다. 자주 응용되는 방법 중의 하나가 스케일링 윈도우[14]이다. 이 방법은 과거의 다수 세대 집단에서 발견되는 목적함수의 최소값으로 식 (3)의  $\gamma$ 를 계속적으로 변경해줌으로써 달성될 수 있는데 사용된 집단의 수를 스케일링 윈도우  $W_s$ 라 한다.

#### (2) 엘리트 전략(Elitist strategy)

모의진화 과정에서 새 유전 연산자의 확률적 속성 때문에 한 세대의 최적 개체가 다음 세대에서 살아남지 못하고 소멸되는 경우가 발생할 수 있다. 최적 개체의 소멸은 좋은 특성을 가지는 유전자를 잃게 되어 탐색이 침체되는 원인을 제공한다. 엘리트 전략은 집단 내에서 가장 강한 개체가 다음 세대로 변경되지 않고 전달되는 것을 보장하기 위해 고안된 것이다. 엘리트 전략은 만약 이전 세대의 최적 개체를 저장하고 있다가 일련의 모의진화가 수행된 후 현 세대에서 소멸된 것이 확인되면, 저장된 최적 개체를 현 세대에서 가장 약한 개체 또는 임의로 선택된 개체와 교환하게 된다.

## 4. 시뮬레이션 및 결과검토

컨테이너 터미널에서 다루어지는 것과 유사한 문제에 SCGA를 적용하고 그 유효성을 검증한다. 전체 적하 물량을 감안하여 두 대의 G/C에 할당된 작업계획은 Table 2에 보여주고 있다. S/C와 G/C 각각 한 대는 같은 조를 이루어 독립적으로 작업하게 된다. Table 3은 야드 내의 컨테이너 장치도를 보여주고 있다. S/C<sub>1</sub>의 작업 계획은 먼저 1번 블록에서 "HAM 20" 그룹의 컨테이너 6개를, 다음 같은 블록에서 "SOU 40" 그룹의 컨테이너 5개를 처리하는 순서로, 총 36개의 컨테이너를 옮기는 것이고, S/C<sub>2</sub>의 작업계획은 2번 블록에서 "JED 20" 그룹의 컨테이너 6개를 처리하는 것을 시작으로 총 31개의 컨테이너

를 옮기는 것이다. 편의상 이 문제에서 블록간의 거리는 100m, Yard-bay 간의 거리는 4m로 간주하였다. S/C<sub>1</sub>의 초기 위치는 1번 블록, 1번 Yard-bay 이고, S/C<sub>2</sub>의 초기 위치는 2번 블록, 1번 Yard-bay이며 작업을 완료하면 다시 초기 위치로 되돌아오는 것으로 간주하였다.

Table 2 Job schedule

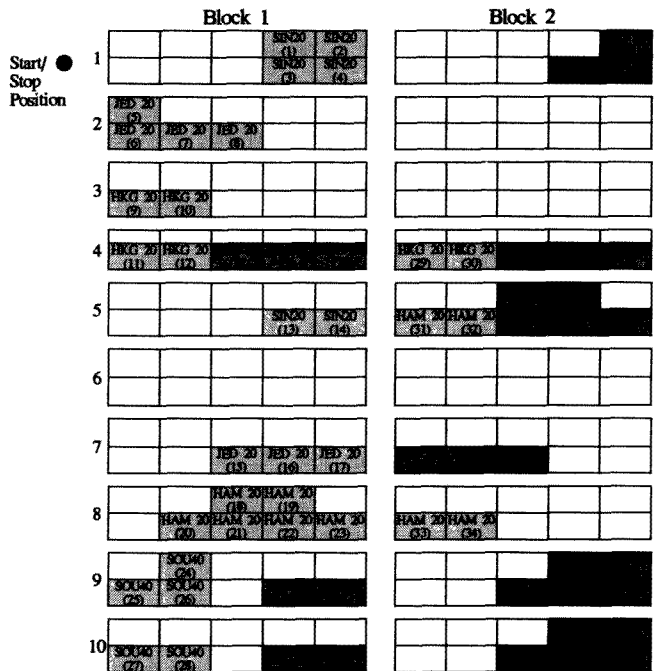
Sequence	1	2	3	4	5	6
G/C <sub>1</sub> Group	HAM 20	SOU 40	HAM 20	JED 20	HKG 20	SIN 20
S/C <sub>1</sub> Containers	6	5	4	7	6	6

Sequence	1	2	3	4	5
G/C <sub>2</sub> Group	JED 20	HAM 20	HAM 40	HKG 40	HKG 20
S/C <sub>2</sub> Containers	6	5	4	10	6

주어진 문제에서 염색체를 표현하면서 작업순서와 컨테이너 그룹을 기반으로 하였고, 특정 컨테이너를 의미하는 문자 번호를 유전자로 하였다. 염색체의 부스트링은 해당 그룹에 속하는 컨테이너로부터 무작위적으로 선정하여 구성하였다.

Table 3 Yard map



적절한 염색체로부터 적합도를 계산하기 위해서는 Table 2, Table 3을 참조하여 먼저 식 (1)의 목적함수를 계산하는 과정이 필요하다.

SCGA의 제어 파라미터로 집단의 크기  $N=20$ , 교배확률  $P_c=0.9$ , 돌연변이확률  $P_m=0.02$ , 스케일링 윈도우  $W_s=1$ 을 사용하였고, 시뮬레이션을 실시한 결과는 Fig. 9와 Fig. 10에 보여주고 있다. 결과적으로 G/C<sub>1</sub>, S/C<sub>1</sub>와 G/C<sub>2</sub>, S/C<sub>2</sub>의 최적 작업순서는 각각 Table 4, Table 5와 같고, 목적함수 값은 각각 496, 488이었다.

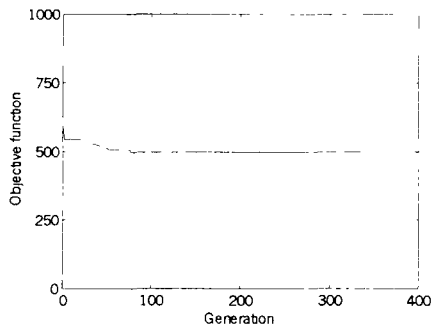


Fig. 9 Search process of SCGA(G/C<sub>1</sub>, S/C<sub>1</sub>)

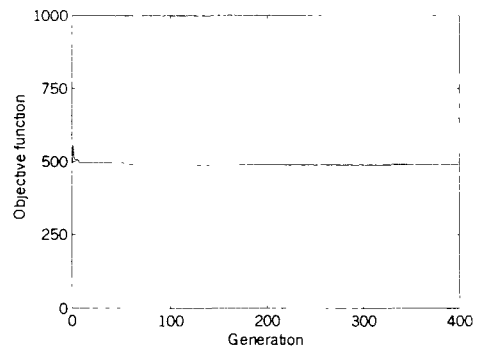


Fig. 10 Search process of SCGA(G/C<sub>2</sub>, S/C<sub>2</sub>)

Table 4 Optimal job sequence for G/C<sub>1</sub> and S/C<sub>1</sub>

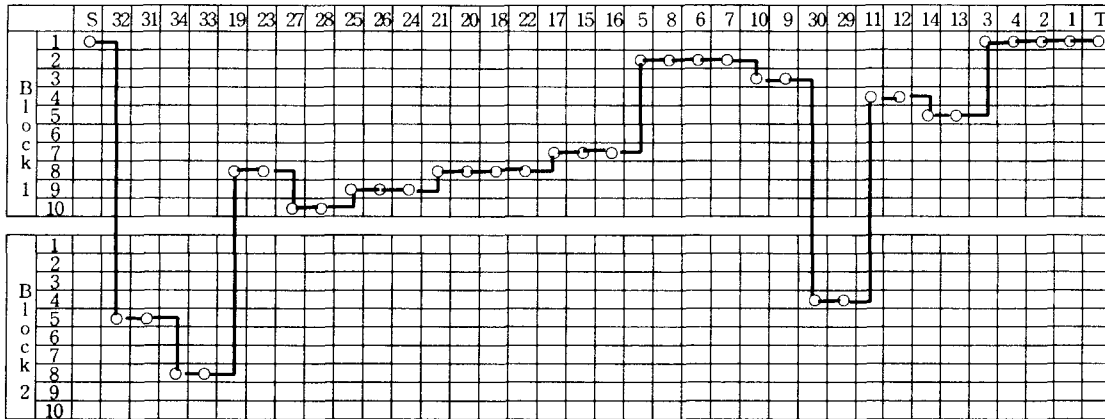
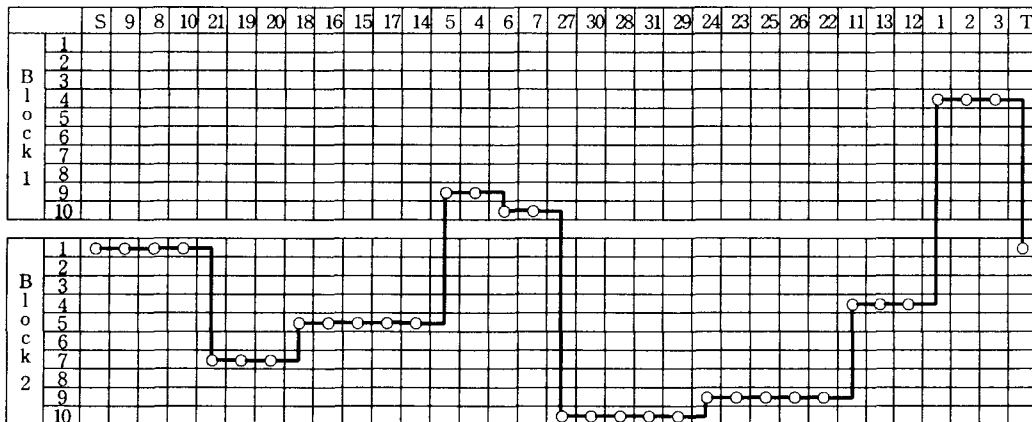


Table 5 Optimal job sequence for G/C<sub>2</sub> and S/C<sub>2</sub>



## 5. 결 론

항만의 경쟁력과 생산성을 높이는 방안으로 수출 컨테이너의 양적하순서 결정, 수출장치장의 배정 계획, 선석 배정계획, 컨테이너 터미널 장비의 할당, 컨테이너 적하작업 시 재취급의 최소화 등이 연구되어 왔는데, 특히 컨테이너 선적 시 적하순서를 적절히 정하는 문제는 터미널 장비운용의 효율성에 크게 영향을 준다.

따라서 본 연구에서는 컨테이너 취급장비의 총 이동거리를 최소화하는 문제를 다루었다. 이는 대표적인 NP-문제로 다루는 변수가 많아지면 해의 수가 지수함수적으로 증가하여 전통적인 방법으로는 해결하기가 어려우므로, 유전알고리즘 프로그램을 개발하고 이용하였다. 프로그램은 염색체의 복호화, 적합도 평가, 집단의 초기화, 유전 연산자 등 기능별로 구분하여 구현하였고, 확장성과 호환성이 유지되도록 함수 형태로 작성하였다. 예제 문제를 통해 시뮬레이션을 실시한 결과 만족한 결과를 얻을 수 있었다.

## 후 기

이 연구는 (주)토탈소프트뱅크의 지원에 의해 수행되었음.

## 참 고 문 헌

- [1] L. A. Gifford, Containership load planning heuristic for a transtainer based container port, MSc Thesis, Oregon State University, 1981
- [2] D. W. Cho, Development of a methodology for containership load planning, PhD Thesis, Oregon State University, 1982
- [3] K. H. Kim, K. Y. Kim and C. S. Ko, "Load scheduling using a genetic algorithm in port container terminals," J. of the Korean Institute of Industrial Engineers, Vol. 23, No. 4, pp. 546-660, 1997
- [4] K. Y. Kim, "Mathematical models and search techniques for containership load planning," PhD Thesis, Pusan National University, 1998
- [5] R. I. Peterkofsky and C. F. Daganzo, "A branch-and-bound solution method for the crane scheduling problem", Transportation Research, Vol. 24B, pp. 159-172, 1990
- [6] E. Kozan and P. Preston, "Genetic algorithm to schedule container transfers at multimodal terminals", Int. Trans. in Operation Research, Vol. 6, pp. 311-329, 1999
- [7] M. Taleb-Ibrahimi, B. De castilho and C. F. Daganzo, "Storage space vs. handling work in container terminals", Transportation Research, Vol. 27B, pp. 13-32, 1993
- [8] W. T. Chan and A. Imai, "The berth allocation problem: heuristic method using genetic algorithm, Proc. of the 1st JSPS-NUS Seminar on Integrated Engineering, pp. 109-114, 1996
- [9] E. Nishmura, A. Imai and S. Papadimitriou, "Berth allocation planning in the public berth system by genetic algorithms", European J. of Operation Research, Vol. 131, pp. 282-292, 2001
- [10] I. Watanabe, "Characteristics and analysis method of efficiencies of container terminal: An approach to the optimal loading/unloading method," Container Age, pp. 36-47, 1991
- [11] <http://www.logistics21.com/>
- [12] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," IEEE Trans. Syst., Man, Cybern., Vol. SMC-16, No.1, pp. 122-128, 1986
- [13] M. Gen and R. Cheng, Genetic algorithms and engineering design, John Wiley & Sons, Inc., N. Y., 1997
- [14] 진강규, 유전알고리즘과 그 응용, 교우사, 2000

원고접수일 : 2002년 0월 0일

원고채택일 : 2002년 0월 0일