

컨테이너의 최적 적하계획을 위한 PPGA

김길태* · 조석제** · 진강규** · 김시화***

*(주)도탈소프트뱅크, **한국해양대학교 기계·정보공학부, ***한국해양대학교 해사수송과학부

PPGA for the Optimal Load Planning of Containers

Kil-Tae Kim* · Seok-Jae Cho** · Gang-Gyoo Jin** · Si-Hwa Kim***

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

**Division of IT, Korea Maritime University, Busan 606-791, Korea

**Division of Maritime Transportation Science, Korea Maritime University, Busan 606-791, Korea

요약 : 컨테이너 적하계획은 항만 장비운용의 효율성을 결정짓는 주요 요인 중 하나이다. 적하계획 문제에 유전알고리즘을 응용할 때 처리 작업 수가 많게 되면, 기존의 방법으로 많은 시간이 걸린다. 이 문제를 해결하기 위하여 본 논문에서는 이주모델과 링구조 기반의 의사병렬 유전알고리즘을 개발하고 컨테이너의 최적 적하순서를 결정하는 문제에 적용하여 그 유효성을 밝힌다.

핵심어 : 의사병렬 유전알고리즘, 이주모델, 링구조, 컨테이너 적하계획

Abstract : The container load planning is one of key factors for efficient operations of handling equipments at container ports. When the number of containers are large, finding a good solution using the conventional genetic algorithm is very time consuming. To obtain a good solution with considerably small effort, in this paper a pseudo-parallel genetic algorithm(PPGA) based on both the migration model and the ring topology is developed. The performance of the PPGA is demonstrated through a test problem of determining the optimal loading sequence of the containers.

Key words : pseudo-parallel genetic algorithm, migration model, ring topology, container load planing

1. 서론

최근 국제간의 수출입 화물의 증가로 컨테이너 물동량을 신속, 정확하게 처리해주는 양적하 서비스가 컨테이너 터미널의 경쟁요소로 부각되면서 항만 간에 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 갖추 수 있는 방안으로는 터미널의 운영방법을 개선하거나 제한된 터미널 장비를 효율적으로 사용하는 것이라 할 수 있다. 이와 관련된 방법으로 수출 컨테이너의 양적하순서 결정, 컨테이너 터미널 장비의 할당, 수출장치의 배정 계획, 컨테이너 적하작업 시 재취급의 최소화 등이 모색되어 왔다.

Gifford(1981)는 컨테이너 하역 작업부두에서 컨테이너선의 선적계획을 위한 학습적 절차를 제안하였고, 컨테이너선의 적하계획을 위한 방법론을 제안하면서 Cho(1982)는 Integer programming 기법을 적용하고 있으며, Kim et

al.(1997)은 유전알고리즘을 적용하는 문제를 검토하였고, 이 연구는 Kim(1998)에 의해 더욱 체계적으로 다루어 졌다. 한편 유사문제로 Peterkofsky and Daganzo(1990)는 그레이의 스케줄링 문제를, Kozan and Preston(1999)은 컨테이너 이송의 스케줄링 문제를, Chan and Ima(1996)는 다수 선박의 선적 배정 문제에 유전알고리즘 기반의 경험적 방법을 적용하였고, Nishimura et al.(2001)은 선박의 동적 선적할당 문제를 다루었다.

이들은 전형적인 NP 문제로서 처리될 변수(작업 수)가 많은 곳에 기존의 유전알고리즘(Genetic algorithm: GA)을 적용하면 GA는 지역해 수렴문제를 극복하기 위해 대집단을 요구하게 되고 이는 연산부담을 과중시키는 요인이 된다. 이러한 문제는 다수의 프로세서가 각자 소집단의 GA를 독립적으로 실행하고 주기적으로 이들 간에 정보를 교환함으로써 개선이 가능하다(Chipperfield, 1996). 따라서 본 연구에서는 중규모 이상의 조합최적화 문제 해결에 적합한 이주모델과 링구조 기반의 병렬 유전알고리즘(Parallel genetic algorithm: PGA)을 개발한다. PGA는 다수개의 직렬 GA가 결합되고 각 GA의 지역집단은 고립시간 동안 독립적으로 진화하다가 서

* 대표저자 : 김길태(회원), ktkim@tsb.co.kr 051)630-7080
** sjcho@mail.hhu.ac.kr 051)410-4344
** gjjin@mail.hhu.ac.kr 051)410-4341
*** 정희원, shalom@mail.hhu.ac.kr 051)410-4237

로 간에 이주자를 교환하는 구조를 갖는다. 각 직렬 GA는 재생산, 교배, 돌연변이 연산자와 엘리트 전략을 채용한다.

PGA는 처리속도를 극대화하기 위해 일반적으로 여러 개의 프로세서로 구현되지만, 하드웨어상의 제약이 불가피할 때는 단일 프로세서에서 가상적으로 구현할 수 있으므로 본 연구에서도 의사병렬 유전알고리즘(Pseudo-parallel genetic algorithm: PPGA)를 구현하고 컨테이너 적하계획 문제에 응용한다. 특히 컨테이너 선적작업 시 적하 순서를 적절히 정하는 것은 장비운용의 효율성을 높이는데 결정적인 역할을 하게 되므로 개발한 PPGA를 적용하고 그 유효성을 검증한다.

2. 컨테이너 적하계획을 위한 PPGA

이전에서는 조합최적화 문제에 기존의 GA를 사용할 때 발생하는 연산부담 문제를 개선하기 위하여 이주모델과 링구조 기반의 PPGA를 개발한다.

2.1 염색체의 표현 및 집단의 초기화

GA를 이용할 때 초기 개체집단이 구성되는데 개체들은 잠정적인 해를 의미하며 유전자의 역할을 반영하도록 염색체(Chromosome)로 표현된다. 컨테이너 적하계획의 목적은 최적의 작업순서를 찾는 것이므로 염색체의 유전자들은 특정 컨테이너를 의미하게 된다. 따라서 PPGA에서 다루는 염색체는 각 작업순서와 컨테이너 그룹(목적지와 크기가 같은 컨테이너)과 연관된 부스트링들의 조합으로 구성된다. Table 1은 그룹기반 염색체의 한 예로서, 'JED 20' 그룹은 Jeddah 행 20' 컨테이너 5개, 'HAM 20' 그룹은 Hamburg 행 20' 컨테이너 7개, 'HKG 40' 그룹은 Hongkong 행 40' 컨테이너 3개로 구성되어 있다. 초기집단을 이루는 염색체들은 같은 그룹내의 유전자를 가지고 무작위로 초기화된다.

Table 1 A chromosome

s	4	5	6	13	14	15	1	2	3	11	12	7	8	9	10
Yard bay	2		1			3			5		4				
Group	JED 20		HAM 20			HKG 40			JED 20		HAM 20				

2.2 적합도 평가

유전 연산자를 통해 새로운 집단이 완성될 때마다 좋은 개체와 더 나은 개체를 구별하기 위하여 적합도(Fitness)가 평가되는데, 일반적으로 적합도함수는 목적함수의 적절한 사상을 통해 얻어진다. PPGA에서도 적합도함수가 최대화 문제로 기술되고 음의 값을 갖지 않도록 다음 사상 식을 이용한다.

$$f_i(k) = -F_i(k) + \gamma \tag{1}$$

여기서 $F_i(k)$ 와 $f_i(k)$ 는 각각 i 번째 염색체의 목적함수와 적합도함수이고, k 는 세대, γ 는 $f_i(k) \geq 0, k \in [0, \infty)$ 의 관계를 만족하는 상수로서 적절한 탐색압이 유지되도록 매 세대마다 스케일링 윈도우 기법(Scaling window scheme)으로 변경된다 (Grefenstette, 1986). 이 기법은 과거의 다수 세대 집단에서 발견되는 목적함수의 최소값으로 γ 를 변경해준다.

2.3 PPGA 유전 연산자와 전략

PPGA는 재생산, 교배, 돌연변이를 기본 유전 연산자로 채용한다. 재생산은 '적자생존'의 현상을 모방한 연산자로서 적합도를 기반으로 개체들을 선택하고 교배급원을 형성해주는 역할을 해주며, PPGA는 교체를 수반한 잔여확률표본선택을 이용한다. 교배는 자연계의 유성생식을 모방하여 개체 간에 정보교환을 해주기 위한 연산자인데, 교배 시 비적법성(Illegitimacy) 문제를 해결하기 위해 PPGA는 PMX를 이용한다. 돌연변이는 준최적해나 사점(Dead corner)으로부터 벗어나게 하고 탐색영역을 확대해주기 위한 연산자로서 PPGA에서는 교체돌연변이와 역치가 이용된다. 특히 돌연변이 시 자손 염색체의 적법성이 유지되도록 같은 그룹 내에서 유전자들이 선택되고 교체된다.

기본 유전 연산자 외에 PPGA에는 엘리트 전략(Elitist strategy)이 채용된다. 이 전략은 집단 내에서 가장 강한 개체가 다음 세대에서 생존하는 것을 보장해 준다.

2.4 이주모델 구조의 병렬화

앞서 언급하였듯이 GA가 지역해 수렴문제를 극복하기 위해 큰 집단을 취하게 되면 연산부담이 급격히 증가하게 된다. 이 문제는 다수의 프로세서가 각자 소집단의 직렬 GA를 독립적으로 실행하고 주기적으로 이들 간에 정보를 교환함으로써 성능개선이 가능하다(Chipperfield, 1996). GA의 병렬화는 전역모델, 이주모델 또는 확산모델 방법으로 달성될 수 있으며 PPGA에서는 이주모델(Migration model)을 채용한다. 이주모델은 집단을 수개의 지역집단(Local population)으로 나누고, 각 집단은 직렬 GA 제어 아래 독립적으로 다른 진화과정을 겪다가 좋은 유전자가 증식되는 것을 돕기 위하여 지역집단간에 시시각각으로 개체들을 이주 교환한다. 이 모델은 각 지역집단에 하나의 프로세서가 할당되고 각 집단은 독립적으로 병렬적으로 정해진 세대, 즉 고립시간(Isolation time) 동안 돌아간다. 각 프로세서는 서로 다른 초기 집단으로부터 시작하기 때문에 유전적 부동은 이 집단들을 다른 방향으로 몰고 갈 수 있다.

2.5 이주방법

PPGA가 소규모의 지역집단을 다루게 되므로 인해 발생하는 유전적 부동현상을 막기 위해 집단 간에 유전자 교류가 충분히 일어나게 해줄 필요가 있다. 이주자를 서로 주고받는

방법으로는 Fig. 1과 같은 링구조를 채택하였다.

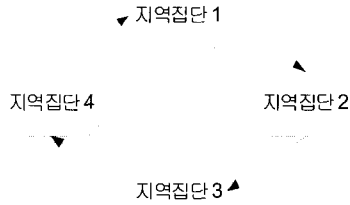


Fig. 1 Example of ring topology

이 구조는 어느 한 방향으로 인접 집단끼리 개체들을 전달하게 된다. 이 때 일어나는 주요 관점으로는 지역집단의 수, 각 집단의 크기, 이주률, 고립시간, 이주할 개체와 교체될 개체들의 선택 등이 있다(진강규, 2000).

2.6 PPGA의 구현

PGA는 처리속도를 극대화하기 위해 여러 개의 프로세서로 구현되지만, 본 연구와 같이 단일 CPU가 탑재된 PC를 이용할 때는 PPGA로 구현할 수밖에 없다. PPGA는 이주모형을 기반으로 하므로 그 구조는 다수개의 직렬 GA가 결합되고 각 GA의 지역집단은 고립시간 동안 독립적으로 진화하다가 서로 간에 이주자를 교환하게 된다. 이주자의 교환은 주기적으로 일어난다. Fig. 2는 PPGA의 구조를 보여준다.

의 Yard-bay에서의 위치는 장치도(Yard map)에 주어진다.

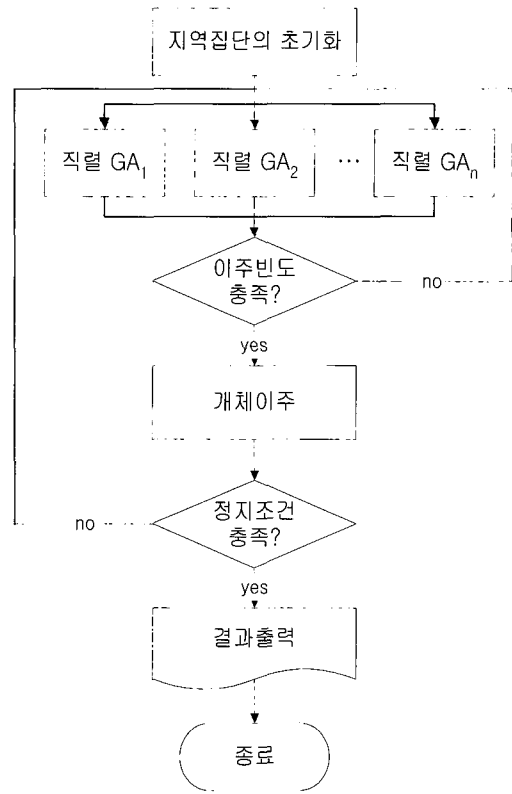


Fig. 2 Structure of a PPGA

3. 컨테이너 적하계획의 정식화

이절에서는 개발한 PPGA를 컨테이너의 적하계획에 응용하고 최적화 도구로서의 유효성을 검증하도록 적하계획 문제를 정식화한다.

3.1 컨테이너의 양적하

컨테이너를 양적하할 때 야드 또는 선박에서 발생하는 재취급을 줄이기 위해서는, 사전에 치밀한 작업계획을 세우는 것이 필요하다. 플래너는 양적하 컨테이너의 수량, 접안 선석의 상황, 계획 기간 중 투입 가능한 키 크레인(QC) 상황을 고려하여 투입할 QC 대수와 작업시간 등을 정하고 작업계획을 작성하게 된다. 작업 계획에는 선적하거나 양하할 컨테이너의 번호, 슬롯(Slot)의 위치, 컨테이너의 작업 순서 등이 나타나 있다. Table 2는 간단한 수출 컨테이너 선적 작업계획의 예이며, Fig. 3은 컨테이너들이 적하될 적하도면의 예를 보여주고 있다.

Table 2에 의하면 QC는 먼저 'JED 20' 그룹의 컨테이너 3개를 Ship bay 49/50에, 'HAM 20' 그룹의 컨테이너 3개를 Ship bay 51에, 'HKG 40' 그룹의 컨테이너 3개를 Ship bay 53에 적재하는 등의 순서로 총 15개를 취급하게 된다. 이들

Table 2 Loading job schedule for a QC

Job sequence	1	2	3	4	5
Container group	JED 20	HAM 20	HKG 40	JED 20	HAM 20
Number of containers	3	3	3	2	4
Loading map location	Bay 49/50	Bay 51	Bay 53	Bay 55/56	Bay 57

3.2 문제의 정식화

수출 컨테이너의 선적을 위한 QC의 작업 계획은 장치장의 스트래들 캐리어(SC)가 컨테이너를 야드 트랙터/트레일러(YT)에 실는 과정에서 야드 내 장치장 사이를 이동하는 총 이동거리에 영향을 주므로, 이를 최소화하여 작업 능률을 높이는 것은 대단히 중요하다. 여기서는 QC의 작업 계획이 주어져 있을 때, 장치장의 SC의 이동 거리를 최소화하면서 QC의 작업 계획을 만족시키는 적하계획 문제를 정식화한다. 이 문제의 정식화는 앞서 수행된 연구들(Gifford, 1981; Kim, 1997; 1998)과 유사성을 가지며, 1) 적하계획은 작업 계획상의 QC의 작업순서 조건을 만족시켜야 하고, 2) 장치장에서 집어 올린 각 그룹 별 총 컨테이너 수는 작업 계획상의 각 그룹 별 선적요구 컨테이너 수와 일치하여야 한다 라는 두

가지 제약 조건을 고려하면 다음과 같다.

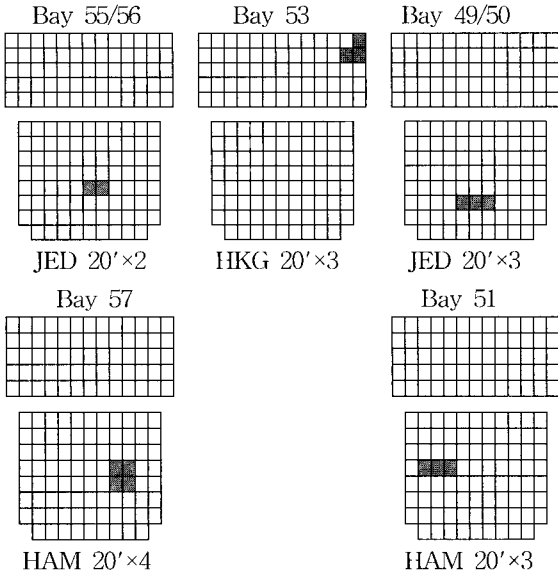


Fig. 3 Loading map of a containership

기호(Index):

- $t = 1, 2, \dots, m$: QC의 작업순번
- i (또는 j) = $1, 2, \dots, n$: Yard bay 번호
- $k = 1, 2, \dots, l$: 컨테이너 그룹 번호
- k_t : 작업순서 t 동안 집어 올려야 할 컨테이너 그룹의 번호

자료(Data):

- B : 장치장의 yard-bay 번호 집합
- G : 컨테이너 그룹의 번호 집합
- $S(k)$: k 컨테이너 그룹의 작업이 필요한 작업순번의 집합
- $B(k)$: k 컨테이너 그룹의 컨테이너들이 쌓여있는 yard-bay 번호의 집합
- $A(B) = \{(i,j) | i,j \in B\}$
- C_{kj} : Yard bay j 에 쌓여있는 k 그룹 컨테이너의 수
- L_t : 작업순서 t 동안 선적되는 컨테이너의 수
- D_{ij} : Yard bay i 와 j 사이의 SC의 이동거리

의사결정변수(Decision variable):

- x_j^t : SC가 작업순서 t 동안 yard-bay j 에서 집어 올린 컨테이너 수
- $y_{ij}^t = \begin{cases} 1, & \text{SC가 작업순서 } t \text{를 마친 후 yard-bay } i \text{와 } j \text{사이를 움직일 경우} \\ 0, & \text{그렇지 않을 경우} \end{cases}$
- $z_{ij}^t = \begin{cases} 1, & \text{SC가 Job sequence } t \text{ 동안 yard-bay } i \text{와 } j \text{사이를 움직일 경우} \\ 0, & \text{그렇지 않을 경우,} \end{cases}$

정식화(Formulation):

$$\text{Min } F = \sum_{t=0}^{m-1} \sum_{(i,j) \in A(B)} D_{ij} y_{ij}^t + \sum_{t=1}^m \sum_{(i,j) \in A(B)} D_{ij} z_{ij}^t \quad (2)$$

S.t.

$$\sum_{j \in B(k_t)} x_j^t = L_t, \quad t \in [1, m] \quad (3a)$$

$$\sum_{t \in S(k)} x_j^t = C_{kj}, \quad k \in G, j \in B \quad (3b)$$

$$x_j^t \leq M(\sum_{i \in B} z_{ij}^t + \sum_{i \in B} y_{ij}^{t-1}), \quad j \in B(k_t), t \in [2, m] \quad (4a)$$

$$x_j^t \leq M(\sum_{i \in B} z_{ij}^t + \sum_{i \in B} y_{si}^0), \quad j \in B(k_1), t = 1 \quad (4b)$$

$$\sum_{i \in B} y_{si}^0 = 1 \quad (5a)$$

$$\sum_{j \in B} y_{jt}^m = 1 \quad (5b)$$

$$\sum_{i \in B} (y_{ji}^{t-1} + z_{ji}^t) - \sum_{j \in B} (y_{ij}^t + z_{ij}^t) = 0 \quad (6)$$

$$\sum_{(i,j) \in A(B)} z_{ij}^t = |A| - 1, \quad t \in [1, m] \quad (7)$$

$$y_{si}^0, y_{jt}^m \in \{0, 1\}, \quad i, j \in B$$

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

식 (2)의 목적함수 F 는 장치장의 SC의 이동 거리를 의미하고, 제약조건 식 (2)는 적하계획이 작업 계획상의 QC의 작업순서 조건을 만족시켜야 하고, 장치장에서 집어 올린 각 그룹 별 총 컨테이너 수가 작업 계획상의 각 그룹 별 선적요구 컨테이너 수와 일치하여야 한다는 것을 의미한다. 그리고 식 (3)은 장치장의 SC가 yard-bay를 방문할 때 컨테이너를 집어 올릴 수 있음을 나타내고, 식 (4)는 각각 작업의 착수와 종료를 보장한다. 식 (5)는 적하작업의 일관된 흐름을 보장하고, 식 (6)은 순환 고립 해를 제거해 준다.

4. 시뮬레이션 및 결과검토

이절에서는 PPGA의 유효성을 검증하기 위해 컨테이너 터미널에서 다루는 유사 문제에 적용하고 그 탐색성능을 새로운 직렬 GA(이하 Serial genetic algorithm: SGA라 함)와 비교한다.

4.1 시뮬레이션

시뮬레이션에 사용된 작업계획은 Table 3과 같다. 전체 처리 물량을 감안하여 두 대의 QC에 적절히 배분하였다.

두 GA의 성능비교는 De Jong의 오프라인 성능(Offline performance)을 계량함으로써 달성된다(De Jong, 1975). 오프라인 성능은 초기 세대부터 T세대까지 최적성능의 평균값으로 정의된다.

$$u^* = \sum_{k=1}^T F^*(k) / T \tag{8}$$

$$\text{단, } F^*(k) = \text{best}_{\substack{1 \leq i \leq N \\ 1 \leq j \leq k}} \hat{F}_i(j) \tag{9}$$

여기서 u^* 는 오프라인 성능, $F^*(k)$ 는 k세대까지의 집단 중에서 발견된 최적의 목적함수 값, $\hat{F}_i(j)$ 는 j번째 세대에서 i번째 개체의 목적함수 값, N은 집단의 크기, best는 ‘minimum’을 의미하는 연산자이다.

4.2 결과검토

GA는 확률적 탐색도구로서 성능은 무작위로 구성되는 초기집단에도 직접 영향을 받게 된다. PPGA의 경우도 예외일 수 없으므로 초기집단의 영향을 줄이기 위하여 독립적인 씨드(Seed)로 프로그램을 10번 돌려 오프라인 성능의 평균값과 표준편차를 구하였고, 그 결과를 Table 5에 요약하였다. SGA의 경우도 같은 표에 요약하였다. 두 GA에서 다루는 총 개체 수가 같음에도 불구하고 PPGA가 SGA에 비해 성능이 좋고 또 더 빨리 해를 찾는 것을 확인할 수 있었다.

Table 5 Comparison of offline performance

	Offline performance	
	QC ₁ /SC ₁	QC ₂ /SC ₂
PPGA	796.765 ± 1.779	562.897 ± 1.092
SGA	808.065 ± 4.799	565.707 ± 1.671

Fig. 4와 Fig. 5는 두 GA가 해를 탐색해가는 과정을 보여주고 있다.

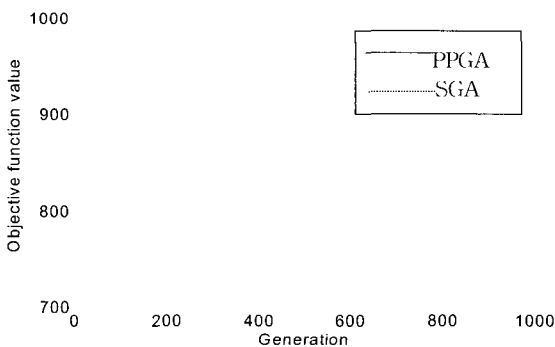


Fig. 4 Search processes of PPGA and SGA(QC₁/SC₁)

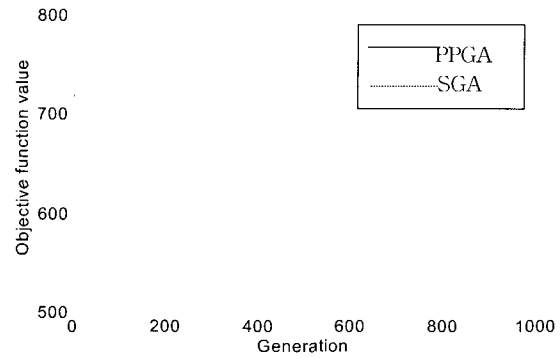


Fig. 5 Search processes of PPGA and SGA(QC₂/SC₂)

QC₁/SC₁ 작업의 경우 SGA는 790세대 가까이에서 해를 찾는데 반해 PPGA는 약 100세대에서 해를 찾았고, QC₂/SC₂의 경우에는 SGA가 560세대 부근에서 찾는데 비해 PPGA는 약 40세대에서 해를 찾는 것을 확인할 수 있다. 결과적으로 시뮬레이션에서 얻은 QC₁/SC₁와 QC₂/SC₂의 최적 작업순서는 다음과 같고 이 때의 목적함수 값은 각각 792와 560이었다.

QC₁/SC₁의 최적 작업순서:

29-28-31-32-30-34-21-22-23-24-25-33-19-18-20-17-16-15-5-8-6-7-26-27-9-10-12-11-14-13-3-1-4-2

QC₂/SC₂의 최적 작업순서:

12-10-11-23-22-21-20-17-19-18-16-9-8-7-6-4-5-31-30-29-26-25-24-28-27-14-15-13-1-2-3

5. 결론

항만의 경쟁력과 생산성을 높이는 방안으로 수출 컨테이너의 양적하순서 결정, 재취급의 최소화 등이 연구되어 왔는데, 컨테이너 선적 시 적하 순서를 적절히 정하는 문제는 터미널 장비운용의 효율성에 크게 영향을 준다. 이들은 대표적인 NP-문제로 다루는 변수가 많아지면 전통적인 GA로는 해결이 어려우므로 본 연구에서는 이주모델과 링구조 기반의 PGA를 개발하고 이를 단일 CPU 컴퓨터상에서 PPGA로 구현하였다. 개발한 PPGA를 컨테이너 취급장비의 총 이동거리를 최소화하는 문제를 통해 시뮬레이션을 실시한 결과 PPGA의 성능이 SGA보다 우수함을 확인할 수 있었다.

참고 문헌

[1] 진강규(2000), “유전알고리즘과 그 응용”, 교우사.
 [2] Chan, W. T. and Imai, A. (1996), “The berth allocation problem: heuristic method using genetic algorithm”, Proc. of the 1st JSPS-NUS Seminar on Integrated Engineering, pp. 109-114.

- [3] Chipperfield, A. J., Fleming, P. J. and Zomaya, A. Y. (1996), "Parallel Genetic Algorithms, Parallel & Distributed Computing Handbook", McGraw-Hill.
- [4] Cho, D. W. (1982), "Development of a methodology for containership load planning", PhD Thesis, Oregon State University.
- [5] De Jong, K. A.(1975), "An analysis of the behavior of a class of genetic adaptive systems", Doctoral Dissertation, The University of Michigan, Ann Arbor, Michigan.
- [6] Gifford, L. A. (1981), "Containership load planning heuristic for a transtainer based container port", MSc Thesis, Oregon State University.
- [7] Grefenstette, J. J.(1986), "Optimization of control parameters for genetic algorithms, IEEE Trans", Syst., Man, Cybern., Vol. SMC-16, No.1, pp. 122-128.
- [8] Kim, K. H., Kim, K. Y. and Ko, C. S.(1997), "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.
- [9] Kim, K. Y. (1998), Mathematical models and search techniques for containership load planning", PhD Thesis, Pusan National University.
- [10] Kozan, E. and Preston, P.(1999), "Genetic algorithm to schedule container transfers at multimodal terminals", Int. Trans. in Operation Research, Vol. 6, pp. 311-329.
- [11] Nishmura, E., Imai, A. and Papadimitriou, S.(2001), "Berth allocation planning in the public berth system by genetic algorithms", European J. of Operation Research, Vol. 131, pp. 282-292.
- [12] Peterkofsky, R. I. and Daganzo, C. F.(1990), "A branch-and-bound solution method for the crane scheduling problem", Transportation Research, Vol. 24B, pp. 159-172.

원고접수일 : 2003년 12월 13일

원고채택일 : 2004년 6월 28일