

발견적 기법을 활용한 2차원 직사각 적재문제에 관한 연구

이정민/이상현**

1. 서론

2차원 직사각 적재문제(2DBPP: 2 dimensional bin packing problem)는 물품/물류 관리에서 일어나는 일상적인 적재 활동을 기술하는 용어로서 작은 개체(Items)를 고정된 공간(Bins or Objects)에 적재하는 행위를 포함한다. 본 문제는 규정된 크기의 유리나 금속 원판을 직사각 형태로 자르는 물품 절단(Stock Cutting)과 같은 의미로 사용된다. 절단된 조각의 형상은 원판 자체보다 크지 않은 무작위 크기의 직사각 형태인데, 여기서 제기되는 문제는 요구되는 조각 개수를 맞추기 위해서 표준 크기인 원판을 얼마나 최소로 사용하느냐는 것이다. 만일 효과적인 절단 계획이 수립되지 않는다면 낭비되는 조각이 많아져 결국 원 재료비 상승을 초래할 것이다. 또한 같은 문제로서 도심에 위치한 공장에서 효과적인 적재 계획을 수립하지 않는다면 저장시설 확장에 따른 부수비용이 필요할 것이며, 트럭 수송의 경우 불필요한 운송으로 인해 비용이 낭비될 것이다. 위 예와 같이 절단(Cutting)과 적재(Packing) 문제는 다른 제약 및 목적으로 서로 다른 산업 시설에서 흔히 볼 수 있다. 예를 들어 나무, 유리, 종이 산업은 일반적으로 표준화된 형상의 절단 작업이 요구되며 수송, 선박 배치, 재직, 가죽 산업 등에서는 비표준 및 무작위 형상의 개체 적재 작업이 요구된다(Dyckhoff, 1990).

이런 유형의 문제는 1960년대 이후 꾸준히 연구되어 1차원 및 2차원 이상의 최적 알고리즘이 Gilmore et al(1961, 1963, 1965)에 의해 최초로 연구되었다.

* 본 내용은 '03년도 한국국방경영분석학회 추계학술대회 발표내용을 정리한 것임.

** 국방대학교 운영분석과

Golden(1976)과 Hinxman (1980)은 2차원 절단 문제에 대한 해법을 실 공정에 적용시켰으며, Rayward-Smith and Shing (1983)에 의해 '1차원 및 2차원 적재 문제'라는 용어가 사용되기 시작했다. 위와 같은 기법들은 모두가 최적해를 구하기란 거의 불가능한 NP-Complete문제(Fowler et al, 1981)이므로 주로 메타휴리스틱 기법에 대한 연구가 많이 이루어지고 있다. 예를 들어 Smith(1985)는 유전자 알고리즘(GA: Genetic Algorithm)을 적재문제에 적용시켜 Jacobs (1996), Liu and Teng(1999)에 의해 완벽한 형태를 갖추게 된다. 또한 Kaprncke(1988)는 1차원 적재 문제에 금속 담금질 방법(SA: Simulated Annealing)을 적용하였고, Dowsland (1993)에 의해 동일 저장소 및 다른 크기의 저장소에 적재하는 실험으로 확장되었으며 Lodi et al(1999)은 타부서치를 이용하여 우수한 해를 구했다.

기존 연구들은 적재 문제에 있어서 최적화 해법에는 많은 한계가 있으며 전통적인 발견적 해법으로는 지역 최적해에 빠질 수 있다는 단점이 있다는 것을 알 수 있다. 또한 상위 수준의 발견적 해법 중 한 가지의 알고리즘으로만 수행된 일부 연구에서는 계산 시간이 과다하거나 최고해의 수준이 다소 미흡함을 알 수 있다. 따라서 발견적 기법을 변형한 형태가 연구되고 있는데, 이에 대한 대표적인 예로서, Jacobs(1996)는 단순 알고리즘과 유전자 알고리즘을 결합한 발견적 해법을 활용했으며, Burke et al(1999)은 타부서치와 금속담금질 기법을 결합하여 유전자 알고리즘 보다 나은 결과를 확인했다. 이러한 기법들은 모두 지역해를 벗어나거나 선 수렴(pre maturing) 등을 극복하기 위한 시도들이었다.

본 연구에서는 일반적으로 알려진 발견적 알고리즘과 상위 수준의 발견적 알고리즘을 조합함으로써 탐색 능력을 향상시키기 위하여 혼합 우측 적재 알고리즘(Hybrid Next Fit Decreasing Height Algorithm, Frenk et al, 1987)과 타부서치를 조합하여 2단계로 구성된 발견적 알고리즘을 제시하고자 한다.

2. 문제의 모형화

본 연구의 목적을 달성하기 위한 구성으로 제 2절에서는 2DBPP에 대한 수리모형을 정의하며 제 3절에서는 발견적 알고리즘의 구성에 대해서 설명한다. 제 4절에서는 실험을 통하여 결과를 검증하며, 끝으로 제 5절에서는 결론과 향후 연구과제를 고찰한다. 본 연구에서 구성하고자 하는 2DBPP 모형은 Gilmore et al(1963)이 제시한 열 생성(Column Generation) 기법을 적용한다. 이에 대한 기본사항으로서 직사각 형태인 각 개체는 회전 이동할 수 없으며, 균등한 크기의 직사각 저장소에 최대한 많은 개체를 저장한다. 이러한 2DBPP의 모형을 구성하는데 필요한 가정사항은 구체적으로 다음과 같다.

1. 모든 개체와 저장소는 직사각형태로서 고정된 폭과 높이로 이루어져 있다.
2. 각 개체는 기타 개체 및 저장소의 모서리와 겹치지 않으며, 접합 공간은 0이다.
3. 각 개체는 수평으로 적재되며 회전하지 않는다.
4. 각 개체의 폭과 높이는 저장소의 폭과 높이보다 크지 않은 정수이다.
5. 하나의 저장소에 적재가 완료되면 다음 저장소로 적재를 반복한다.

2.1 변수 정의 및 모형 구성

본 절에서는 위에서 정의한 모형과 가정사항을 수리적인 모형으로 구성하기 위해 필요한 변수들을 정의하고 모형을 수립하며 제약식에 대해 설명한다. 먼저 수리모형에서 사용되는 변수들은 다음과 같다.

x_j : 저장소(Bin), $j = 1, \dots, m$

r_i : 직사각개체, $i = 1, \dots, n$

H : x 의 높이

W : x 의 폭

h_i : r_i 의 높이

w_i : r_i 의 폭

a_{ij} : x_j 에 적재하는 r_i

A_j : a_{ij} 에 의해 생성된 2진 열 벡터(binary column vector) 행렬

이러한 변수들을 이용하여 수리모형을 구성하면 목적함수인 식 (1)과 제약식인 식 (2)~(6)이 구성된다.

$$\text{Minimize } \dots\dots\dots \sum_{j=1}^m x_j \quad (1)$$

Subject to $\dots\dots\dots$

$$\sum_{j=1}^m a_{ij} x_j = 1, \dots\dots\dots \forall i \quad (2)$$

$$\sum_{i=1}^n a_{ij} h_i \leq H, \dots\dots\dots \forall j \quad (3)$$

$$\sum_{i=1}^n a_{ij} w_i \leq W, \dots\dots\dots \forall j \quad (4)$$

$$x_j \in \{0, 1\}, \dots\dots\dots \forall j \quad (5)$$

$$a_{ij} \in \{0, 1\}, \dots \dots \dots \forall i, j \quad (6)$$

$$(i = 1, \dots, n, j = 1, \dots, m)$$

본 모형의 근본적인 목적은 저장소(m) 개수를 최소화 하는 것이므로 식 (1)이 목적함수가 된다. x_j 는 오로지 1개만 존재하므로 식 (2)가 성립된다. 여기서 a_{ij} 는 <Figure 1>과 같이 x_j 의 패턴을 결정짓는다.

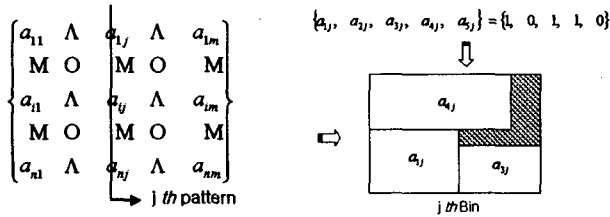


Figure 1. Column Generation Approach.

2차원 적재를 위해 하나의 x_j 에 포함되는 모든 개체는 W와 H를 초과할 수 없으므로 식 (3)~(4)가 만족되어야 한다. 또한 x_j 가 최종 해에 포함된다면 1이며 그렇지 않을 경우 0값을 갖는다(식 (5)). 마지막으로 x_j 에 r_i 가 적재될 때 $a_{ij} = 1$ 이며 그렇지 않을 경우 0이므로 식 (6)이 성립된다.

3. 발견적 알고리즘

본 절에서는 제 2절에서 구성한 수리적 모형의 목적함수인 최소 저장소 개수를 산출하기 위한 발견적 알고리즘을 제시한다. 발견적 알고리즘은 두 단계의 절차로 구성되며, 첫 번째 단계에서는 혼합 우측 적재 알고리즘을 이용하여 초기해를 산출하고, 두 번째 단계에서는 타부서치 기법을 이용하여 공간활용도 함수로 선정되는 저장소를 초기해에서 제거하여 총 저장소 개수를 줄이는 효과적인 알고리즘을 제시한다. 이러한 조합으로 알고리즘을 구성한 이유는 혼합 우측 적재 알고리즘은 단순 발견 알고리즘으로서 지역 최적해에 빠질 수 있는 위험이 있기 때문에 상위 수준의 발견적 알고리즘인 타부서치로 단순 발견 알고리즘을 제어한다면 보다 우수한 해를 생성할 수 있기 때문이다.

3.1 순서도

본 절에서는 2단계로 구성된 발견적 알고리즘의 전반적인 절차를 설명한 후 이를 순서도로 나타낸다.

먼저 알고리즘의 절차에서 사용된 용어 중 K_{max} 는 타부서치 기법에서 금지되는

이동 목록을 저장하는 타부목록(Tabu List)의 최대 개수를 의미하고 하한값(Lb: Lower Bound)은 식 (7)과 같이 모든 개체들의 총 면적을 저장소 면적으로 나누어 올림한 이상적인 최적해를 나타낸다. 따라서 하한값은 모든 최종해 보다 반드시 작거나 같기 때문에 종료 조건이 된다.

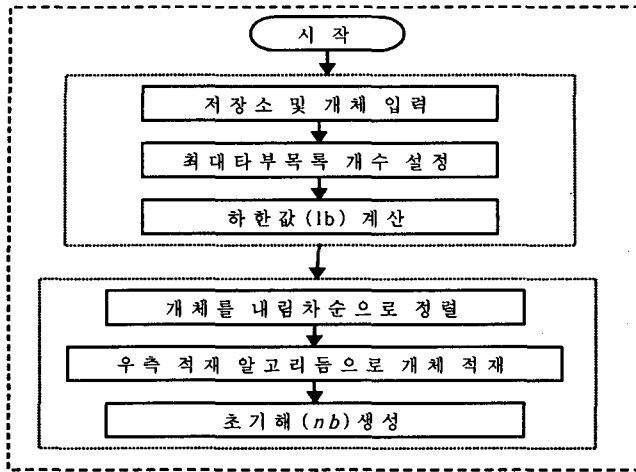
$$lb = \text{Int} \left\{ \frac{\sum_{i=1}^n w_i h_i}{W \times H} \right\}, \dots\dots\dots (7)$$

3.2 혼합 우측 적재 알고리즘(초기해 산출)

단순 발견 알고리즘은 초기해 생성을 위해 이용하며 타부서치 구동시 매 단계마다 반복된다. 본 연구에서는 그러한 단순 발견 알고리즘으로서 혼합 우측 적재 알고리즘(Hybrid Next Fit Decreasing Height Algorithm, Frenk et al, 1987)을 사용하였다. 이는 모든 개체를 정렬한 후 이용하는 오프라인(Off-line) 알고리즘이며, 특정 문제에 대해 이상적으로 사용할 수 있는 저장소 개수를 M이라 할 때 2M개 이상을 이용하지 않는데, 적재 행위 절차는 다음과 같다.

1. 모든 개체들을 높이에 따라 내림차순으로 정렬한다.
2. 첫 번째 개체(r_1)를 저장소(x_1)의 최 좌측 아래에 적재한다.
3. 나머지 개체를 r_1 의 우측에 적재한다. 우측 공간이 없다면 r_1 의 위쪽에 적재를 시도한다.
4. x_1 에 더 이상 적재할 수 있는 공간이 없다면 다른 저장소에 적재하는 작업을 반복한다.
5. 모든 작업이 끝나면 저장소 개수를 셴한다.

▶ 1단계: 단순 발견 알고리즘



▶ 2단계: 타부서치 알고리즘

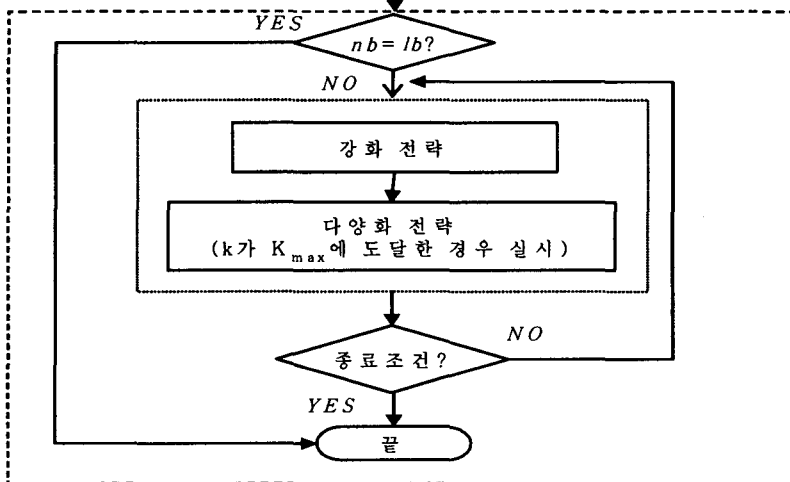


Figure 2. Flow Diagram of Metaheuristic Algorithm.

3.3 타부서치

본 연구에서는 타부목록의 크기를 2 또는 3으로 정하였다. 아울러 비록 타부목록에 들어있는 금지된 이동이라 할지라도 지금까지 발견된 최고해를 능가하는 이동이라면 타부목록에서 그 이동을 빼주고 다음 반복의 개선해로 채택하는 소위 열망(Aspiration)조건도 적용하였다. 타부서치에서 종료조건은 조금씩 다르게 설정할 수 있는데 2DBPP에서는 일반적으로 종료 시간에 이르거나 하한값에 도달한 경우에 종료하는 방식을 쓰게 된다. 이외에도 정해진 반복수까지 탐색을 진행하는 경우나 탐색이 허용되는 최고한도에 도달했을 때 또는 이웃이 없을 때까지 탐색을 계속하는 종료조건도 있다. 본 논문에서는 종료시간을 채우거나 하한값에 도달한 경우와

최대 다양화 반복 횟수에 도달한 경우를 종료조건으로 이용하였다.

강화 전략으로는 최초 타부서치가 끝난 뒤 그동안 얻어진 최고해를 기준으로 최대 이웃 개수에 도달할 때까지 타부서치를 작동하는데 이때 이웃의 크기를 동적으로 변환시키면서 충분히 좋은 해 근처를 집중적으로 조사하도록 유도하였다. 또 다양화 전략으로는 최대 이웃 개수에 도달해서도 최적해의 변화를 확인할 수 없을 경우 총 저장소 중 가장 작은 공간 활용도 함수 값을 가진 저장소 중 1/2을 해에서 제거한 후 다시 일련의 과정으로 적재 행위를 시행하여 그 동안 탐색하지 못한 다른 영역을 탐색하도록 유도하였다.

타부서치 기법은 강화 전략 및 다양화 전략을 통해 총 저장소 개수를 줄인다. 이때 저장소 목록 중 제거할 수 있는 저장소를 선택하도록 각 저장소마다 공간을 이용하고 있는 활용도 지수(Index, S_U)가 부여된다. 이를 공간 활용도 함수(Space Utilization Function)로서 계산하는데, S_U 는 각 저장소의 사용되는 비율을 나타내므로 그 값이 클수록 제거되기 어렵다는 것을 의미한다. 따라서 값이 가장 낮은 저장소를 선별하여 해당 저장소에 적재된 개체들을 다른 저장소로 옮겨 보다 좋은 해를 구할 수 있다.

즉 타부서치를 반복할 때마다 식 (9)의 S_U 가 가장 낮은 저장소(wb: the weakest bin)를 선별하고, 내부 개체들을 다른 저장소에 우측 적재 알고리즘으로 적재행위를 하여 wb를 기존 해에서 삭제시킴으로써 총 저장소 개수를 줄일 수 있다. 이때 구한 해가 종료 조건을 만족하지 않는다면 각 저장소의 S_U 값은 갱신된 상태에서 다음 타부서치의 비교 값이 된다.

$$S_S = W \times H - \sum_{i=1}^n w_i \times h_i \quad \dots\dots\dots (8)$$

$$S_U = \frac{|W \times H - S_S|}{W \times H} \quad \dots\dots\dots (9)$$

식 (9)의 k 는 해당 저장소에 적재된 개체의 총 수를 나타내며, S_S 는 여유공간(Slack Area)을 계산한다.

4. 알고리즘 실행 및 분석

본 절에서는 제 3절에서 설명한 이론을 기초로 하여 실험을 수행한다. 코딩은 Visual C++6.0이며, CPU는 Pentium III 930Mhz, 메모리는 256M인 환경에서 구동했다. 총 21,000개의 개체에 대해 K_{max} 를 변화시키면서 최적해와 CPU 계산

시간을 측정했으며, 기본 환경으로 다양화 전략을 2회까지만 허용, 타부길이(Tabu Tenure)는 20이다. 실험 결과는 K_{max} 가 1일 때 항상 나쁜 해가 생성되었고, 2인 경우 r 이 40이하에서 그리고 3인 경우는 r 이 60이상일 때 각각 우수한 해가 생성되었다. 4인 경우는 오히려 2와 3에 비해 해의 질이 나빠짐을 확인할 수 있었다. 마지막으로 CPU 계산 시간은 K_{max} 가 커질수록 지수적으로 증가했으며, 공간 활용도 합수를 통해 계산시간이 단축되었다.

4.1 실험 계획

본 실험을 위한 개체는 Visual C++ 6.0 seed를 적용하여 같은 기준으로 생성하였으며, 그 기준은 <Table 1>과 <Table 2>와 같다.

Class I ~ III은 Burkey and Wang(1987)에 의해 제시되었으며 각 저장소 크기의 범위 이내에서 무작위로 개체를 생성한다. Class IV ~ VII은 Lodi. et. al(1999)이 Class I ~ III을 확장시킨 개념으로써 <Table 2>의 형(Type) 기준과 같이 저장소의 폭과 높이 비율을 기준으로 하므로 보다 다양한 형태의 개체를 생성할 수 있다.

Table 1. Criterion of the Generation I

Class	저장소크기	각개체
I	$W_1 = H_1 = 10$	구간 [1,10]에서 무작위로 개체생성
II	$W_2 = H_2 = 40$	구간 [1,35]에서 무작위로 개체생성
III	$W_3 = H_3 = 100$	구간 [1,100]에서 무작위로 개체생성
IV	$W_4 = H_4 = 100$	Type 1 확률 70%, Type 2,3,4 확률 각각 10%
V	$W_5 = H_5 = 100$	Type 2 확률 70%, Type 1,3,4 확률 각각 10%
VI	$W_6 = H_6 = 100$	Type 3 확률 70%, Type 1,2,4 확률 각각 10%
VII	$W_7 = H_7 = 100$	Type 4 확률 70%, Type 1,2,3 확률 각각 10%

Table 2. Criterion of the Generation II

Type 1	w_3 는 $\left[\frac{2}{3}W, W\right]$, h_3 는 $\left[1, \frac{1}{2}H\right]$ 의 구간에서 무작위 개체생성
Type 2	w_3 는 $\left[1, \frac{1}{2}W\right]$, h_3 는 $\left[\frac{2}{3}H, H\right]$ 의 구간에서 무작위 개체생성
Type 3	w_3 는 $\left[\frac{1}{2}W, W\right]$, h_3 는 $\left[\frac{1}{2}H, H\right]$ 의 구간에서 무작위 개체생성
Type 4	w_3 는 $\left[1, \frac{1}{2}W\right]$, h_3 는 $\left[1, \frac{1}{2}H\right]$ 의 구간에서 무작위 개체생성

위에서 각 Class는 20, 40, 60, 80, 100개의 개체로 구성되며, 각각은 통계를 위해 10번의 작업이 반복된다. 즉 각 Class는 3000개의 개체로 구성된다. <Figure 3>은 <Table 1>과 <Table 2>의 기준에 의해 생성된 개체들의 분포를 보여준다.

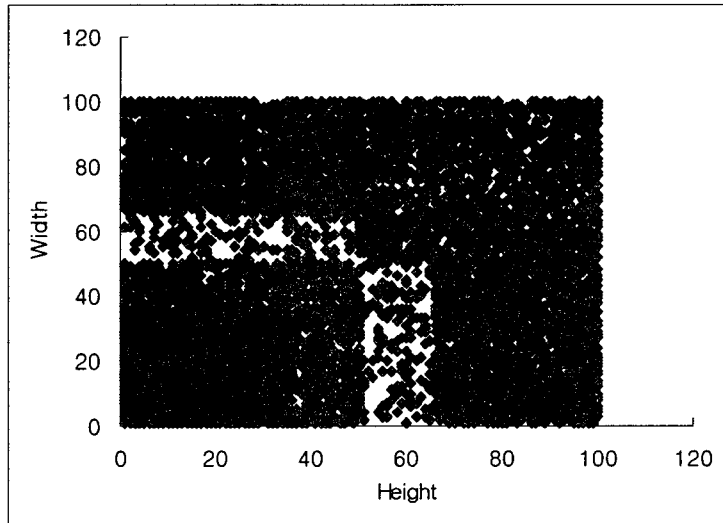
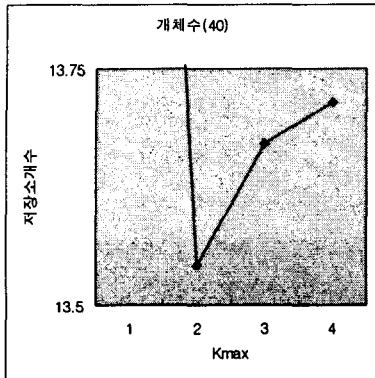


Figure 3. Item Distribution.

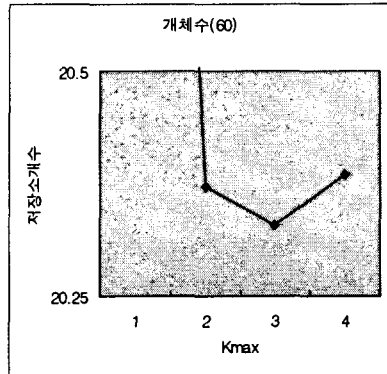
4.2 실험 결과

<Table 3>은 3절에서 제시한 알고리즘을 7개의 개체군 적용한 결과를 보이고 있다. table 안의 각 행은 개체군에 대한 세부 결과로서 각 열은 기준 하한값(lb), 단순 발견 알고리즘(HNFDHA)에 의한 초기 값, 기존 문제와 K_{max} 를 동적으로 변화시켰을 때의 해와 CPU Time을 각각 나타낸다.

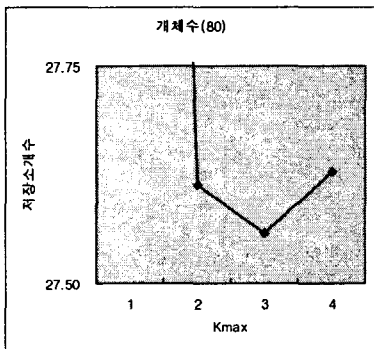
<Table 3>에서 알 수 있듯이 '제시된 해법'은 '기존의 해법'에 비해 전반적으로 CPU 계산 시간이 향상되었으며, 특히 개체수가 40 이하 일 때, 보다 우수한 해가 생성됨을 알 수 있다. 각 Class별 발견된 우수한 해의 분포는 <Table 4>와 같으며, 개체수에 대한 K_{max} 와 저장소 개수의 변화는 <Figure 4>와 같다.



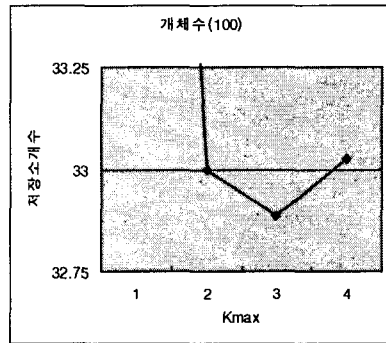
(a)



(b)



(c)



(d)

Figure 4. Relationship of the K_{max} and bins.

<Figure 4>는 K_{max} 가 1일 때 항상 나쁜 해가 생성되며, 4인 경우 역시 상대적으로 우수하지 못한 해가 생성됨을 보여준다. 이는 타부목록이 너무 작으면 해의 순환이 일어나 효율적인 탐색이 어려우며, 반대로 클 경우 너무많은 움직임이 금지되기 때문에 해의 질이 나빠지기 때문이다.

<Table 4>에서 개체수 20~40인 경우 계산 시간의 개선 비율이 대폭적으로 증가한 이유는 K_{max} 가 2이기 때문이며, 개체수 60이상일 때 점차적으로 CPU 계산 속도가 향상됨을 알 수 있다. <Figure 5>는 K_{max} 에 따라 소요되는 CPU 계산 시간을 나타내며, K_{max} 가 커질수록 개체수에 따라 계산시간이 지수적으로 증가함을 알 수 있다.

Table 3. Results of class I ~ VII

Class	개체수	lb (평균)	NFDH (평균)	기존해법		제시된해법	
				nb1(평균)	평균CPU시간(초)	nb2(평균)	평균CPU시간(초)
I	20	6.4	9.5	7.3	0.41	7.3	0.32
I	40	12.0	18.1	13.9	4.34	13.7	2.02
I	60	18.5	28.4	20.6	15.48	20.6	15.37
I	80	25.3	38.6	28.3	24.59	28.3	23.96
I	100	30.5	47.1	33	57.55	33	54.27
II	20	4.4	7.2	5.6	0.43	5.6	0.40
II	40	8.2	13.9	10	2.70	9.8	1.22
II	60	12.5	22	14.8	8.90	14.8	8.82
II	80	17.3	30.2	20.5	12.18	20.5	11.90
II	100	20.5	36.2	23.9	25.57	23.9	24.13
III	20	5.4	8.3	6.8	0.46	6.8	0.37
III	40	10.1	16.9	12.4	3.43	12.4	1.39
III	60	15.7	26.4	18.6	11.61	18.6	11.51
III	80	21.5	36.5	25.6	23.51	25.6	22.98
III	100	25.9	43.9	29.7	30.49	29.7	28.96
IV	20	4.7	6.5	5.8	0.35	5.8	0.30
IV	40	9.7	14.4	11.6	3.43	11.5	1.44
IV	60	14.0	21.3	16.5	10.38	16.5	10.27
IV	80	19.7	30.6	23.8	25.05	23.8	24.50
IV	100	23.8	36.2	28.1	37.25	28.1	35.15
V	20	4.8	6.9	5.9	0.49	5.9	0.42
V	40	9.6	14.6	11.7	4.66	11.6	1.88
V	60	14.1	20.9	17	21.64	17	21.48
V	80	19.5	29.3	23.1	42.54	23.1	41.34
V	100	24.1	35.8	28.6	95.94	28.6	91.70
VI	20	9.8	16	14.8	2.85	14.8	0.98
VI	40	18.0	31	27.8	18.25	27.7	3.84
VI	60	27.6	47.9	43.7	70.30	43.7	69.67
VI	80	37.1	62.9	57.6	178.41	57.6	173.51
VI	100	45.0	77.2	69.6	302.32	69.6	293.47
VII	20	3.8	5.4	4.6	0.29	4.6	0.27
VII	40	6.9	10.9	8.3	2.09	8.2	1.23
VII	60	9.4	14.9	11.1	5.87	11.1	5.81
VII	80	12.2	19	14	12.30	14	12.05
VII	100	15.3	23.6	17.3	11.53	17.3	11.03

※ 기존 해법은 Lodi et al, 1999이 제시한 발견적 기법임.

Table 4. Improved calculation time of CPU

개체수	기존 시간	개선된 시간	개선된 비율(%)
20	0.75	0.44	70.45
40	5.56	1.86	198.92
60	20.60	20.42	0.88
80	45.51	44.32	2.69
100	80.30	76.96	4.34

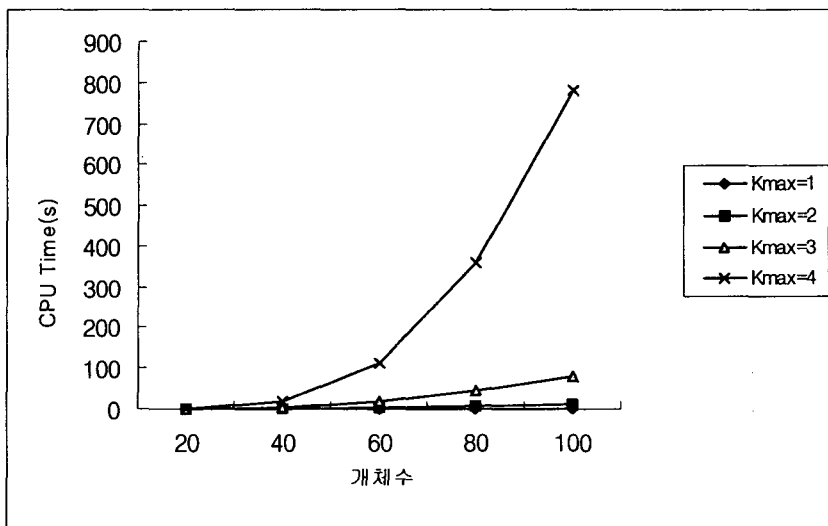


Figure 5. Relationship of the items and calculation time of CPU.

<Figure 5>를 통해 K_{max} 가 1~3인 경우 전반적으로 빠른 계산 속도를 확인 할 수 있었으나, 4인 경우에는 개체수가 60 이상일 때 상당한 과부하가 걸림을 알 수 있다. 본 실험에서는 Class VII인 경우 특히 정도가 심함을 확인할 수 있었다 (<Table 5>).

이상으로 본 연구가 제시하는 알고리즘을 적용하여 특정 최대타부목록 개수가 일정한 개체 수에서 보다 나은 해가 생성된다는 것과 CPU 계산 시간이 향상됨을 확인하였다. 본 연구는 기존 연구와 똑같은 환경에서 실험할 수 없는 한계로 인해 1차적으로 기존연구를 기본 알고리즘으로 수행하여 해와 CPU 계산 시간을 구한 후, 제안된 알고리즘의 해와 CPU 계산 시간을 분석하였으므로 객관적인 비교를 하고 할 수 있다.

Table 5. Result of class VII(mean value)

개체수	lb	NFDH	$K_{max}=1$	T1	$K_{max}=2$	T2	$K_{max}=3$	T3	$K_{max}=4$	T4
20	9.8	16	15	0.259	14.8	0.98	14.8	2.84	14.8	7.27
40	18.0	31	28.7	0.507	27.7	3.84	27.8	18.18	27.8	92.01
60	27.6	47.9	44.7	0.786	43.7	9.68	43.7	69.67	43.7	590.11
80	37.1	62.9	58.9	1.014	57.6	17.54	57.6	173.51	57.7	1718.51
100	45.0	77.2	70.6	1.101	69.6	29.38	69.6	293.47	69.8	3817.88

※ 각 수치는 전체 평균임.

5. 결론

무기체계의 획득환경은 변화되어 왔음에도 불구하고 미 군원 또는 모방생산으로 인해 전력화하는데 별 어려움이 없었던 과거의 경험을 위주로 획득환경을 판단해 온 우리 군은 1990년대 중반이후부터 선진국의 기술이전회피, 정보의 획득제한이 대두하기 시작하여 무기체계를 연구 개발하는데 많은 문제점이 발생하기 시작했다. 특히 우리 군은 거시적인 효과를 중시하는 분위기로 주장비 위주의 개발을 하여, 무기체계에 필요한 전력화 지원요소에 대한 개발 및 발전이 더디게 되어 무기체계 전력화시 운용상에서의 문제점이 발생하고 유지하는 측면에서도 많은 문제점을 야기하였다.

따라서 무기체계 전력화를 보장하기 위해서는 국방획득관리규정에 전력화 지원요소의 소요제기 및 시험평가 절차가 구체화되어야 한다. 또한 주장비 및 전력화 지원요소 관련 사항이 제대로 검토되려면 과학적 기법 즉, 전투실험 및 기술시험에 의한 업무절차 여건이 조성되어야 하는데, 전투실험기법에 대한 마인드의 변화, 전투실험기법 획득, 전투 실험 센터 설립 등을 위한 대책을 제시하였다.

본 연구에서는 제시한 무기체계 전력화 지원요소의 업무 절차에 대한 개선방안을 통해 효율적인 업무가 추진된다면 우리 군이 소요로 하는 무기체계를 획득함에 있어서 전력화 지원요소의 간과로 인해 발생하는 문제점 해소와 미래전에 부합되는 무기체계를 획득할 수 있는 계기가 될 것으로 기대된다.