타부서치를 이용한 2차원 직사각 적재문제에 관한 연구
(Application of Tabu Search to the Two-Dimensional Bin Packing Problem)

이 상 헌

국방대학교 운영분석학과
우 122-875 서울시 은평구 수색동 205번지

Abstract

The 2 DBPP(Two-Dimensional Bin Packing Problem) is a problem of packing each item into a bin so that no two items overlap and the number of required bins is minimized under the set of rectangular items which may not be rotated and an unlimited number of identical rectangular bins. The 2 DBPP is strongly NP-hard and finds many practical applications in industry. In this paper we discuss a tabu search approach which includes tabu list, intensifying and diversification strategies. The HNFDH(Hybrid Next Fit Decreasing Height) algorithm is used as an internal algorithm. We find that use of the proper parameter and function such as maximum number of tabu list and space utilization function yields a good solution in a reduced time. We present a tabu search algorithm and its performance through extensive computational experiments.

1. 서 론

2차원 직사각 적재문제(2DBPP: Two- Dimensional Bin Packing Problem)는 물품/물류 관리에서 발생하는 일상적인 적재 활동을 기술하는 용어로써 작은 개체를 고정된 공간에 적재하는 행위를 포함한다. 본 문제는 규정된 크기의 유리나 금속 원판을 직사각 형태로 자르는 물품 절단과 동일한 의미로 사용된다. 절단된 조각의 형상은 원판 자체보다 크지 않은 무작위 크기의 직사각 형태로써, 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 문제로써 도심에 위치한 공장에서 효과적인 적재 계획을 수립하지 않는다면 저장시설 확충에 따른 부수비용이 필요할 것이며, 트럭 수송의 경우 불필요한 운송으로 인해 비용이 낭비될 것이다. 위 예와 같이 절단(cutting)과 적재(packing) 문제는 상이한 제약 상황 및 목적으로 서로 다른 산업 시설에서 흔히 볼 수 있다. 예를 들어 나무, 유리, 종이 산업은 일반적으로 표준화된 형상의 절단 작업이 요구되며 수송, 선박배치, 재직, 가죽산업 등에서는 비표준 및 무작위 형상의 적재작업이 요구된다.

이러한 유형의 문제는 1960년대 이후 지속적으로 연구되어 1차원 및 2차원 이상의 최적 알고리듬이 Gilmore & Gomory[6,7]에 의해 최초로 연구되었다. 보다 세밀한 연구로써 Golden[8]은 2차원 절단(cutting stock)문제에 대해 실험하였고, Hinxman[9]은 2차원 조각손실(trim loss) 및 배치(assortment) 문제에 대한 해법을 실 공정에 적용시켰으며, Rayward-Smith & Shing[12]에 이르러 '1차원 및 2차원 적재 문제'라는 용어가 사용되었다. 3차원 직사각 적재문제에 관한 연구는 Dowsland[3]가 시초로 Dowsland. K & Dowsland. W[4]에 의해 확장되었다. Whelan & Batchelor[14]는 불규칙한 개체에 대한 적재를 연구하였으며 Dowsland. K & Dowsland. W[5]에 의해 상대적으로 우수한 해법이 개발되었다.

위와 같은 기법들은 모두가 최적해를 구하기란 거의 불가능한 NP-complete문제로써 지역 최적해에 빠지는 단점이 있으므로 이를 극복하기 위해 메타휴리스틱 기법에 대한 연구가 이루어져 왔다.

메타휴리스틱 기법은 단순 휴리스틱으로 초기해를 구한 후 이를 개선하는 형태로 발전하였는데, 예를 들어 Smith[13]는 초기해를 슬라이드 알고리듬으로 구한 후 유전자 알고리듬을 이용하여 최적해를 구했으며 Liu & Teng[11]은 개체가 적재되는 순서를 조합형태로 표현하고 적재 위치를 Bottom-Left 알고리듬으로 구현하였다. 이러한 표현 기법은 교차연산자와 돌연변이 연산자의 효과적 활용에 근거한 유전자 알고리듬을 통하여 유용한 기법임이 밝혀졌다.

Kapmke[10]는 1차원 적재문제에 시뮬레이티드 어닐링(Simulated annealing)기법을 적용하였으며, Dowsland[3]에 의해 동일 개체 및 다른 크기의 개체를 팔렛트에 적재하는 실험으로 확장되었다. Dowsland는 2차원 무한 적재 문제(strip packing)를 다루었는데, 적재 가능해와 불가능해를 동시에 고려하므로, 탐색간 목적함수를 총 겹치는 영역과 수직 또는 수평 개체 이동에 대응하는 모든 해를 포함하는 이웃으로 선정하였다. 이를 통하여 현재해를 개선시키는 새로운 가능 해가 발견되면 상한기준 높이를 갱신하여 해를 발전시켰다.

타부서치에 관한 연구는 Blazewicz 등[2]이 최초로 시도하였다. Blazewicz 등은 단순한 적재 절차

로써 가능해를 생성한 후 타부서치를 이용하여 기존 레이아웃을 개선시키는 방법을 취하였는데, 개체 하나를 임의로 선택한 후 가능한 수많은 목표지점을 선별하여 가장 좋은 곳에 적재한다. 여기서 이동은 해당 개체가 다른 개체와 겹치지 않으면서 적재됨을 의미하며 최근에 이동한 개체가 타부리스트 원소가 된다. Blazewicz 등[2]의 이론적 연구와는 달리 Lodi 등[1]은 현실에서 사용가능한 직사각 적재 알고리듬을 개발하였다. Lodi 등은 두 가지 제한사항으로써 고정된 개체와 초기해로 생성된 레이아웃을 생성하였다. 위 가정에 적용된 타부서치의 중요한 특징은 특별한 적재문제에 독립적인 탐색 기법과 이웃을 적용하였다. 가장 우수한 해를 구하기 위한 절차로써 단순 휴리스틱 알고리듬으로 초기해를 구한 후 이동을 통하여 적재된 개체들을 다시 분배하는 과정을 취한다. 이동은 k 개의 특정 저장소에 적재된 부분 개체집합과 목표 저장소에 적재된 단일 개체의 교환을 통해 이루어진다. 알고리듬은 자동으로 지역 최적해로부터 벗어나기 위하여 탐색간 k 를 지속적으로 갱신한다. 따라서 높은 수의 k 는 보다 강력한 재결합을 의미하지만 동시에 이웃 탐색간 상당한 계산 시간을 요구한다. Lodi 등[1] 이 사용한 알고리듬은 Finite First Fit과 Finite Best Fit이며 초기해를 개선하기 위하여 열거된 저장소중 비울 수 있는 가능성이 가장 높은 저장소를 해에서 삭제시키기 위해 개체 및 저장소의 크기와 임의상수로 계산되는 채우기 함수(FF: Filling Function)를 활용하고, 타부서치의 비교 기준으로써 이웃수를 조절하는 최대 타부목록 개수를 모든 개체군에 동일하게 적용하였다.

본 논문에서는 2차원 직사각 적재문제의 최적해를 혼합우측 적재 알고리듬(HNFDH: Hybrid Next Fit Decreasing Height Algorithm)으로 보다 최적해에 근접한 좋은 초기해를 생성한 후 타부서치로 해를 개선하기 위하여 공간 활용도 함수를 사용하여 각 개체군에 적합한 최대 타부목록 개수를 동적으로 산정하고 강화 및 다양화전략을 구사함으로써 DBPP 문제에 대한 보다 효율적인 최적해 기법을 제시하고자 한다.

2. 문제의 모형화

2차원 적재문제는 여러 개체들의 적절한 조합을 고려하여 큰 저장소에 적재하는 최적화 문제로써 수많은 산업 시설에서 응용되었으며 단순히 작업공정의 한 부분으로 제한되지 않고 OR과 회계 분야 등을 통해 보다 추상적인 형태로 발전되었다. 적재문제는 유사한 문제의 다양성과 응용분야로 인해 동일한 분야이지만 다른 이름의 형태를 갖고 있다. 따라서 비록 이러한 문제들이 다른 응용분야에서 발견된다 할지라도 결국 같은 논리 구조를 갖게 된다. 분류구조는 크게 공간차원 및 비공간차원과 관련된 문제로 나눌 수 있다. 첫 번째 그룹은 3차원까지 정의되는 유클리디안공간 속에서 다발(bundle) 절단문제, 차량 로딩문제, 팔레트 로딩 문제 등과 같이 '절단과 적재' 또는 '로딩 문제'로 구성된다. 또 다른 그룹은 메모리 배치, 재정 회계, 동전 교환, 라인 밸런싱 등의 무게, 시간, 화폐 등과 같은 비 공간적 차원을 다루는 추상적인 '절단과 적재' 문제를 포함한다.

본 논문의 모형은 직사각 개체들을 동일한 형태의 저장소에 적재하고 각 저장소는 레벨(level) 형태를 이루는 2차원 직사각 적재문제로써 가정사항은 다음과 같다. 먼저 기하학적인 동질성을 보장하기 위해 모든 개체와 저장소는 직사각형태로 고정된 폭과 높이로 이루어져 있고 각 개체는 기타 개체 및 저장소의 모서리와 겹치지 않으며, 접합 공간은 0이다. 또한 수평으로 적재되고 회전하지 않으며 폭과 높이는 저장소의 폭과 높이보다 크지 않은 정수이다. 다음으로 레벨적재를 위한 가정사항으로써 각 레벨에서 최좌측 개체는 해당 레벨에서 높이가 가장 크고 단일 저장소의 최아래 레벨은 다른 레벨의 높이보다 크며 각 개체들은 높이에 따른 내림차순으로 정렬된 후 번호가 다시 부여된다.

2.1 변수 정의

개체를 r_i라 정의하고 y_i를 i번째 레벨에 적재되는 개체라고 하자. 여기서 r_i가 최좌측에 위치한다면 y_i는 1이며 그렇지 않은 경우 0값을 갖는다. 개체 i에 의해서 초기화되는 i번째 레벨의 잠재적인 개수를 n, 레벨 k에 의해서 초기화되는 k번째 저장소의 개수를 마찬가지로 n이라 가정할 경우 사용되는 변수는 다음과 같다.

y_i: $i \in J$ (J는 직사각 개체집합).

q_k: $k \in J$ 레벨 k가 저장소 k를 초기화하면 1, 그렇지 않을 경우 0.

H: 저장소의 높이

W: 저장소의 폭

h_i: 개체 i의 높이

w_i: 개체 i의 폭

2.2 수리모형

2.1에서 정의한 변수를 이용하여 수리모형을 구성하면 목적함수인 식 (1)과 제약식인 식 (2)부터 식 (6)과 같이 수식화 할 수 있다.

$$\text{Minimize} \quad \sum_{k=1}^{n} q_k \tag{1}$$

Subject to

$$\sum_{i=1}^{j-1} x_{ij} + y_j = 1, \ \ \forall j \tag{2}$$

$$\sum_{j=i+1}^{n} w_j x_{ij} \leqq (W - w_i) y_i, \quad (i = 1, \ldots, n-1) \tag{3}$$

$$\sum_{k=1}^{i-1} z_{ki} + q_i = y_i, \ \ \forall i \tag{4}$$

$$\sum_{i=k+1}^{n} h_i z_{ki} \leqq (H - h_k) q_k, \quad (k = 1, \ldots, n-1) \tag{5}$$

$$y_i,\ x_{ij},\ q_k,\ z_{ki} \in \{0,1\}, \ \ \forall i, j, k \tag{6}$$

위에서 x_{ij}와 z_{ki}는 각각 개체와 레벨을 나타내며

개체 j가 레벨 i에 적재될 때 x_{ij}는 1이고 그렇지 않을 경우 0이며, z_{ki}는 레벨 i가 저장소 k에 위치할 때 1이며 기타의 경우 0이다.

본 모형의 근본적인 목적은 개체 적재에 사용되는 총 저장소 개수(n)를 최소화하는 것이므로 식 (1)이 목적함수가 된다. 유효한 적재 상태를 구성하기 위한 제약식으로는 식 (2)~식 (6)이 사용되며, 식 (2)~식 (5)에서 i, j, k의 크기는 레벨 적재 가정사항에 의해 $j > i$와 $i > k$이 성립한다. 식 (2)는 각 개체가 정확히 한 번만 적재됨을 의미하며, 식 (4)는 각 레벨이 하나의 저장소에만 위치한다는 것을 뜻한다. 식 (3)과 식 (5)는 각각 사용된 레벨의 폭과 저장소의 높이를 제한한다.

3. 터부서치 알고리듬

타부서치의 일반적인 구성요소로는 '단기메모리와 장기메모리', '통제조건과 열망조건', '강화(intensifying)전략과 다양화(diversification) 전략' 등을 포함한다. 열망조건은 비록 타부목록에 들어있는 금지된 이동이라 할지라도 지금껏 발견된 최고해를 능가하는 이동이라면 타부목록에서 그 이동을 빼주어 다음 반복의 개선해로 채택되도록 한다. 강화전략은 단기메모리 함수를 사용하여 현재 상태보다 좋은 해를 탐색하도록 환경을 맞춘다. 이러한 전략은 제한된 영역에서 최상의 해를 적극적으로 탐색하는 데 초점을 두며, 다양화 전략은 장기메모리 함수를 이용하여 해 공간에서 방문하지 않았던 영역을 탐색하도록 한다. 단기메모리 함수는 타부목록과 열망조건으로 구현된다. 타부목록은 최근에 방문했던 곳의 이동을 기억하여 다시 탐색하지 않도록 한다. 이동은 일정기간 목록에 남아 있어서 보다 좋은 해를 적극적으로 탐색하도록 한다.

3.1 기본 알고리듬

적재문제에 타부서치를 적용하기 위한 기본 알고리듬의 Step 1은 단순 휴리스틱 절차이고 Step 2는 타부서치 절차를 나타낸다. 본 절차는 Finite First Fit으로 초기해를 구한 후 고정된 최대 타부목록 개수와 Filling Function으로 초기해를 개선한 Lodi 등[1]의 연구와 달리 적재되는 개체를 입력 받은 후, 이를 단순 휴리스틱 알고리듬으로 높이에 따라 내림차순으로 정렬하여 우측 적재를 실시하며, 타부서치기법으로써 개체군에 적합한 최대 타부목록 개수를 산정하고 공간 활용도 함수로써 최적해를 산출한다.

$K_{\max}$는 타부서치 기법에서 금지되는 이동 목록을 저장하는 타부목록(tabu list)의 최대개수를 의미하고 k는 강화전략간 갱신되는 타부목록 개수를 의미한다. 식 (7)의 하한값(lb: lower bound)은 모든 개체의 총 면적을 단일 저장소 면적으로 나누어 올림한 이상적인 최적해이다. 따라서 하한값은 어떠한 최종해 보다 반드시 작거나 같기 때문에 종료조건이 된다. 또한 다양화 전략의 $D_{\max}$는 허용되는 다양화 전략의 최대반복횟수이며 d는 갱신되는 다양화 전략의 횟수이다.

$$lb = \left\lceil \frac{\sum_{i=1}^{n}(w_i \times h_i)}{W \times H} \right\rceil \tag{7}$$

Step 1에서 적용된 단순 발견 알고리듬은 초기해 생성을 위해 이용하고 타부서치 구동시 매 단계마다 반복되며, 모든 개체를 정렬한 후 이용하는 오프라인 알고리듬이다. 계산시간은 $O(n\log n)$이며 특정 문제에 대해 이상적으로 사용할 수 있는 저장소 개수를 M이라 할 때 $2M$개 이하만을 사용한다.

3.2 공간 활용도 함수

타부서치를 반복할 때마다 식 (8)의 각 저장소의 활용도 함수(S_U)가 가장 낮은 저장소(wb: the weakest bin)를 선별하여 개체들을 다른 저장소에 적재시키면 wb를 기존해(nb)에서 삭제시킬 수 있다. 이때 구한 해($nb-1$)가 종료 조건($nb = lb$)을 만족하지 않는다면 각 저장소의 S_U값은 갱신된 상태에서 다음 타부서치의 비교 기준이 된다.

$$S_U = \frac{|W \times H - S_S|}{W \times H} \tag{8}$$

식 (9)의 k는 해당 저장소의 총 개체수이며, 여유 공간(S_S: slack area)을 계산한다.

$$S_S = W \times H - \sum_{i=1}^{k} w_i \times h_i \tag{9}$$

본 알고리듬의 종료 조건은 하한값에 도달한 경우, 제한시간이 초과된 경우, 그리고 최대 다양화 반복 횟수에 도달한 경우이다.

3.3 단기 메모리를 이용한 강화 전략

초기해가 하한값(lb)과 같지 않을 경우 최초로 강화전략을 취하는데, 이는 타부목록의 크기를 k라 할 때, 최초 타부서치가 끝난 뒤($k=1$) 얻어진 최적해가 하한값과 같지 않을 경우 이웃의 크기를 동적으로 변환($k = k+1$)시켜 최대 이웃 개수에 도달($k = K_{\max}$)할 때까지 반복한다.

3.4 장기 메모리를 이용한 다양화 전략

다양화 전략은 해공간에서 방문하지 않았던 새로운 영역을 탐색하기 위한 방법으로써 지역 최적해를 벗어나기 위해 사용되며 보다 넓은 해 영역을 탐색하도록 구현한다. 이의 구현은 더 이상 강화전략만으로 해를 개선시킬 수 없을 경우 총 저장소(nb) 중 가장 작은 공간 활용도 함수 값을 가진 저장소의 1/2을 해에서 제거한 후 다시 일련의 과정으로 적재 행위를 시행하여 그 동안 탐색하지 못한 다른 영역을 탐색하도록 한다.

4. 알고리듬 실행 및 분석

프로그램은 Visual C++6.0으로 작성하여, Pentium Ⅲ 930Mhz, 메모리 256M인 환경에서 구동했다. 총 21,000개의 개체에 대해 $K_{\max}$를 변화시키면서 최적해와 CPU 계산 시간을 측정했고, 기본 환경으로 다양화 전략을 2회까지 허용하였다.

본 실험은 Lodi 등[1]의 기존 연구(CODE: FORTRAN 77, HARDWARE: Silicon Graphics INDY R4000sc 100MHz)와 동등한 환경에서 테스트 할 수 없는 제한사항과 단순 발견 알고리듬으로써 'Finite First Fit/ Finite Best Fit' 알고리듬을 동일하게 구축할 수 없는 한계로 인하여, 초기해 생성을 위한 혼합 우측적재 알고리듬을 Lodi 등[1] 연구와 본 논문에 동일하게 적용하고 타부서치 수행간 취약 저장소를 선별한 후 최종 저장소 개수를 줄이는 방법으로써 공간 활용도 함수를 사용하여 CPU 계산시간을 향상시키며 각 개체군에 적합한 최대타부목록 개수를 산정하여 보다 효율적인 최적해를 유도하여 비교한다.

이에 대한 결과로 전반적으로 향상된 CPU 계산시간을 유도할 수 있었으며, 특히 개체수가 40 이하 일 때, 보다 우수한 해가 생성되었다. 각 Class별 발견된 우수한 해의 분포는 <표 1>와 같다.

<표 1> CPU 계산 시간

개체수 (Item)	Time (Lodi *et al.*)	Improved Time	Improved Ratio(%)
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

<표 1>에서 두 번째 열은 Filling Function으로 취약저장소를 선별한 경우의 CPU 계산시간이며 세 번째 열은 공간활용도 함수로 취약 저장소를 해에서 삭제시킨 경우이다. 개체수가 20~40인 경우 계산 시간의 개선 비율이 대폭적으로 증가한 이유는 $K_{\max}$가 2이기 때문이며, 개체수 60이상일 때 점차적으로 향상됨을 알 수 있다.

5. 결 론

본 연구에서는 2차원 직사각 적재문제로써 길로틴(guillotine) 형태를 보이는 레벨 적재(level packing) 알고리듬을 다루었으며 균등한 크기의 저장소에 무작위로 생성된 직사각개체들을 적재하여 총 사용되는 저장소 개수를 최소화 하였다. 본 논문에서 제시한 알고리듬은 보다 빠르고 만족할 만한 수준의 해를 얻을 수 있도록 단순 발견 알고리듬인 혼합우측 적재 알고리듬과 상위 수준의 발견적 알고리듬인 타부서치를 이용하여 2차원 적재 문제를 적용하였다. 본 연구에서 제시한 발견적 알고리듬은 특정 공정에 쓰이는 단순 발견 알고리듬으로 초기해를 산출하고 국부탐색능력이 우수한 타부서치를 이용하여 최고해를 산출함으로써 비교적 빠른 수렴 속도와 만족할 만한 좋은 해를 동시에 얻을 수 있었다. 또한 취약한 저장소 선정을 위해 공간활용도 함수를 적용함으로써 향상된 CPU 계산시간을 유도 할 수 있었다.

참고문헌

[1] Andrea Lodi, Silvano Martello and Daniele Vigo, "Approximation Algorithms for the Oriented Two-Dimensional Bin Packing Problem," *European Journal of Operational Research*, Vol.112(1999), pp.158-166.

[2] Blazewicz, J., Hawrylak, P. and Walkowiak, R., "Using a Tabu Search Approach for Solving the Two-Dimensional Irregular Cutting Problem," *Annals of Operations Research*, Vol.41(1993), pp.313-327.

[3] Dowsland, K., "Some Experiments with Simulated Annealing Techniques for Packing Problems," *Operational Research*, Vol.68(1993), pp.389-399.

[4] Dowsland, K. A. and Dowsland, W. B., "Packing Problems," *European Journal of Operations Research*, Vol.56(1992), pp.2-14.

[5] Dowsland, K. A. and Dowsland, W. B., "Solution Approaches to Irregular Nesting Problems," *European Journal of Operations Research*, Vol.84(1995), pp.506-521.

[6] Gilmore, P. C. and Gomory, R. E., "A Linear Programming Approach to the Cutting-Stock Problem Part Ⅰ," *Operational Research*, Vol.9(1961), pp.724-746.

[7] Gilmore, P. C. and Gomory, R. E., "A Linear Programming Approach to the Cutting-stock Problem Part Ⅱ," *Operational Research,* Vol.11(1963), pp.863-888.

[8] Golden, B., "Approaches to the Cutting Stock Problem," *AIIE Transactions*, Vol.8 (1976), pp.265-274.

[9] Hinxman, A. I., "The Trim Loss and Assortment Problems," *European Journal of Operational Research,* Vol.5(1980), pp.8-18.

[10] Kampke, T., "Simulated Annealing: Use of a New Tool in Bin-Packing," *Annals of Operations Research*, Vol.16(1988), pp.327-332.

[11] Liu, D. and Teng, H., "An Improved BL-Algorithm for Genetic Algorithm of the Orthogonal Packing of Rectangles," *European Journal of Operational Research*, Vol.112(1999), pp.413-419.

[12] Rayward-Smith, V. J. and Shing, M. T., "Bin Packing," *Bulletin of the Institute of Mathematics and Its Applications*, Vol.19 (1983), pp.142-146.

[13] Smith, D., "Bin-Packing with Adaptive Search," *In Grefenstette(ed), Proceedings of an International Conference on Genetic Algorithms and their Applications*, Lawrence Erlbaum, (1985), pp.202-206.

[14] Whelan, P. F. and Batchelor, B. G., "Automated Packing Systems: Review of Industrial Implementations," *SPIE*, *Machine Vision Architectures, Integration and Applications* 2064, (1993), pp.358-369.