트레일러와 트렉터를 사용하는 운송문제 분석

Analysis of transportation problems with trailers and tractors

한윤택*, 장수영

포항공과대학교 산업경영공학과

*bluerain@postech.ac.kr

Abstract

This paper considers an interesting transportation problem where trailers and tractors are involved in moving material. We identified a class of combinatorial optimization problems for minimizing the number of tractors and trailers required to accommodate the transportation needs. Then, we show that the fundamental problem is NP-hard and analyze its properties to develop efficient heuristic to handle the problem effectively.

1. 서론

이 논문에서는 이론적으로 흥미롭고 중요하지만 비교적 덜 알려진 트레일러와 트렉터가 함께 물건을 운반하는 운송문제(TTP – Trailer and Tractor Problem)에 대해서 다룬다. 간단히 설명하면, 트레일러는 운송되는 물건들을 직접 싣고, 트렉터는 트레일러를 필요한 위치로 운송한다. 이 때 목적은 주어진 물건들을 각각의 정해진 시작위치에서 도착위치로 운송하는 최소 비용의 운송계획 작성이며 비용은 사용되는 트레일러와 트렉터의 수로 표현된다.

여기에서 일반적인 TTP문제를 여러 세부 문제로 분류하는 몇 가지 요건을 소개하고 또한 앞으로 다룰 문제 상황에서 이 요건들에 대한 전제를 언급한다. 첫 번째는 트레일러에 실을 수 있는 물건의 내용에 관한 요건이다. 일반적으로는 트레일러는 각각의 제한된 적재능력을 가지고, 실을 수 있는 물건의 양은 이에 따라 결정된다. 하지만 본 논문에서 모든 트레일러는 같은 적재능력을 가지고 또한 트레일러에 싣게 되는 물건들은 사전에 적재능력에 맞추어서 하나의 로트(Lot)로 묶여있다고 가정한다. 물건들을 로트로 묶는 문제는 물건의 종류에 따른 제약조건에 의해 달라진다. 일례로 철강코일 제품을 로트로 묶는 문제에 대해서는 [2]의 연구를 참고할 수 있다. 트레일러는 하나의 로트만을 싣게되고 특정 출발위치서 로트를 싣게 되면 다른 물건은 실을 수 없고 대응되는 도착위치로 이동되어야 한다. 따라서 전체적으로 주어진 개수의 로트를 각 해당 요청지역 사이에 운반하는 문제가 된다. 두 번째 논의 사항은 각 로트를 해당 위치에서 트레일러에 싣거나 내리는 시간 즉 작업시간에 관해서다. 로트를 트레일러에 싣고 내리는 일에는 각각 기구가 사용되고 각 위치에서 이러한 기구도 일정에 따라 사용됨으로 로트의 작업시간이 정해지면 임의로 조정할 수 없다. 일반적인 경우에는 로트의 작업시간이 주어지지 않은 상

태에서 일정을 계획하게 되고 이 경우 크레인들의 일정도 함께 계획된다. 하지만 때론 로트의 작업시간이 정해진 상태에서 운송 일정 계획을 해야 하는 상황도 있다. 이 경우 하나의 결정변수-작업시간은 이미 정해졌으므로 보다 제한적인 문제로 볼 수 있다.

우리는 두 가지의 문제에 대해 분석하게 된다. 두 문제 모두 트레일러에 싣는 물건은 하나의 로트로 묶여있지만, 첫 번째 문제에서는 출발 지역과 도착지역에서 로트의 작업시간이 정해지지 않은 경우를 다루고(TTP-NFT Trailer and Tractor Problem with Not Fixed Time), 두 번째로 다룰 문제는 로트의 작업시간이 정해진 경우(TTP-FT Trailer and Tractor Problem with Fixed Time)를 다룬다. 우리는 TTP-FT문제가 TTP-NFT문제의 제한적 문제라는 점을 알고 있다.

1.1 연구의 동기

이 연구의 계기는 POSCO의 철강 코일 제품 해송출하 계획에서 얻었다. 철강 코일 제품들은 운송 계획을 세우기 전에 로트로 묶여지고 여러 위치의 창고에 보관된다. 각 창고(warehouse)에 있는 로트에는 최종목적지에 따른 부두(dock)가 정해져 있다. 앞으로 논문에선 로트의 싣는 위치를 창고, 내리는 위치를 부두라 부른다. 트레일러는 창고에서 로트를 싣고(상차), 트렉터는 트레일러를 창고에서 정해진 부두로 운반한다. 부두에 도착하면 트레일러는 로트를 내리게 (하차) 된다. 이러한 작업을 수행하는 일정 계획이 기간별로 수립되어야 한다. 하지만 현재 관리자들이 임의로 결정해서 수행하고 있는 실정이고 또한 해송 출하는 출하 제품 중 가장 큰 부분을 차지하고 있기 때문에 이 문제에 대한 분석은 절실하다. 이러한 분석은 철강산업뿐만 아니라 커다란 부피와 용량을 가지는 제품들을 생산하는 다양한 산업 즉 트레일러와 트렉터가 이용되는 산업 등에도 적용될 수 있다.

1.2 기존 연구

다양한 운송 문제들이 기존에 연구 되었다. 이러한 문제들 중에서 몇 가지는 우리가 다루는 TTP와 유사한 점을 발견할 수 있다. 먼저 유사한 세 문제를 제시하고 이들과 우리 문제의 연관성과 차이점을 분석하겠다. 첫 번째는 Routing-scheduling 문제로, 각각의 일들이 운송 네트워크의 각 노드에 위치하고 기계들이 일들을 처리하기 위해서 네트워크상을 움직인다[5]. 모든 일을 주어진 시간 내에 처리하기 위한 최소 대수의 기계를 찾는 목적을 가진다. 두 번째로는 Pickup and Delivery Problem with Time Window (PDPTW)가 있다. 이는 Time Window가 있는 차량운행경로문제(VRPTW)의 보다 일반화된 형태로 각 물건의 운송 요청이 특정 출발지역에서 정해진 시간(Time Window) 사이에 수행해야 하는 싣기(Pickup)과 도착지역에서 마찬가지로 정해진 시간 사이에 일어나야 하는 배달(Delivery)로 이루어지며, 모든 운송 요청을 수행하기 위한 최소 비용 경로를 만드는 목적을 가진다. 이 때 운송 차량은 정해진 적재능력을 가지며, 또한 하나의 경로에 같은 물건의 싣기와 배달이 함께 포함되어야 하는 짝 제약 조건도 가진다[8]. 마지막으로 Dial-a-ride 문제를 들 수 있다. 이 문제는 온라인 일정 계획문제, 즉 계획을 세우는 일과 요청이 들어오는 일이 함께 일어나는 경우로 볼 수 있다. 이 문제에서 이용객은 원하는 출발과 도착 지역을 요구하고, 운송은 다른 이용객들도 함께 서비스하

는 차량에 의해서 이루어진다[6]. 이 때 목적은 여러 제약조건 하에서 모든 이용객의 요청을 수행하는 최소 비용 경로를 만드는데 있다.

위에 소개한 문제들은 모두 비용을 최소화 하기 위한 차량경로를 작성하고, 또한 각 지역에서 일 수행, 신기, 배달, 이용객탑승 등의 작업을 한다는 점이 TTP문제에서 트레일러와 트렉터 수를 줄이는 최적 운송계획을 작성하고, 각 위치에서 로트를 싣거나 내리는 작업(상/하차 작업)을 하는 점과 유사하다. 따라서 TTP문제에서 위에 문제에서 최소 비용 경로를 찾는 방법들이 도움이 되 보이지만, TTP문제는 이를 어렵게 하는 커다란 차이가 있다. 즉 위에 제시된 세 문제들은 모두 차량 하나만이 운송수단으로 이용되고, 이는 곳 트레일러와 트렉터가 합쳐진 형태로 볼 수 있다. 즉 우리 문제는 두 가지 운송수단의 경로 계획이 필요한 매우 다른 속성을 갖는다. 기존의 연구들에서 다루어지지 않은 부분으로 앞으로 많은 연구가 필요한 분야이다.

1.3 접근방법과 논문의 구성

다음 장에서는 우리가 다루기로 한 두 가지 문제 중 보다 일반적인 TTP-NFT에 대해 문제의 인스턴스로 이루어진 수학적 모델을 제시하고, 이 문제가 어려운 문제(NP-hard)임을 보인다. 이에 대한 접근 방법으로 로트의 상/하차 시간을 발견적 방법으로 정하면, 3장에서 다룰 문제 TTP-FT가 된다. 3장에서는 상/하차 시간이 정해진 문제 TTP-FT에 대해 다룬다. 여기에서 우리는 트레일러와 트렉터의 일정 계획을 순차적으로 수행한다. 이를 각각 트레일러 일정계획, 트렉터 일정계획으로 명하면, 트레일러 일정계획은 최소 개수 쇄 분해(MCCD)[4]을 통해 쉽게 풀 수 있음(in polynomial time)을 보인다. 트렉터 일정계획은 요건을 정함에 따라 트레일러 일정계획과 마찬가지 방법으로 쉽게 풀 수 는 경우와 어려운 문제(NP-hard)가 되는 경우로 나누어짐을 보인다. 마지막으로 4장에서는 본 논문의 의미와 결론을 제시하고 추후 필요한 연구에 대해서 정리함으로써 논문을 끝마친다.

2. 문제정의와 계산복잡도

2.1 문제정의

우리는 여기에서 TTP-NFT를 정의한다. 우선 운송계획 시 주어진 정보와 필요한 가정들을 언급하고, 문제 인스턴스를 통해 수학적 모델을 제시한다. 운송 계획을 수립할 때 아래 정보들은 주어진다.

1)각 로트의 창고(상차)위치와 부두(하자)위치
2)각 로트의 상차 소요시간, 내리는 하차시간

그리고 우리는 다음과 같은 사항을 가정한다

1)모든 트레일러와 트렉터의 일정 계획이 시작될 때와 끝날 때 위치는 한 곳(storage)이다..
2)트렉터가 트레일러와 결합하거나 분리하는 시간은 트렉터의 이동시간에 포함된다.

인스턴스(I^{TTP}) 와 표기법

- 운송 계획 수립 기간: T
- n개의 위치 집합 $P=\{s,p_1,...,p_i,...p_{n-1}\}$:
  s->트레일러와 트렉터의 보관장소
- 수행할 로트의 m개 집합 $L=\{l_1,\ldots,l_i,\ldots l_m\}$

$li=(ls_i,ld_i,w_i,us_i,ud_i,d_i)$ 순서대로, 상차 시작시간, 상차 소요시간, 창고 위치,하차 시작시간, 하차 소요시간, 부두 위치. $w_i,d_i \in P$

- nTI:사용가능한 트레일러 수
  nTT:사용가능한 트렉터 수

- 두 위치 p_h 와 p_i 사이의 거리는 $d(p_h,p_i)$
- 트렉터의 속력 v 는 다음과 같다.
  - a: 로트를 실은 트레일러를 실은 상태
  - b: 빈 트레일러를 실은 상태
  - c: 트렉터만 이동할 때
- $T(d(p_h,p_i),v)$: 두 위치 p_h 와 p_i 사이를 v의 속도로 이동할 때 소요시간

목적

운송계획 수립기간 내에 모든 로트의 운송 요청을 만족시키는 최소 트레일러와 트렉터 수를 구하라.

이 문제에 대한 자세한 증명은 여기서는 생략하고 어려운 문제(NP-hard)임을 보이는 방법에 대해 간략히 설명한다. 문제의 특정상황(special case)로 트레일러와 트렉터가 한 대씩 있는 경우를 가정하면, 두 운송기기는 항상 함께 움직인다. 따라서 하나의 차량으로 생각할 수 있으며 또한 로트를 창고에서 실은 후엔 반드시 대응하는 부두로 가야 함으로 각 로트에 대응하는 창고와 부두를 하나의 도시로 생각할 수 있다. 즉 TTP-NFT는 하나의 차량이 주어진 개수의 도시를 모두 주어진 시간 안에 여행할 수 있는지를 묻는 외판원문제(Traveling salesman problem)로 축소된다.

즉 TTP-NFT는 P=NP가 아니라면 다항시간(polynomial time)알고리즘이 존재하지 않는다. 따라서 우리는 우선 발견적 방법으로 로트의 상/하차 시간을 정해줌으로써 문제에 접근한다. 상/하차시간이 정해지면 이는 곧 TTP-FT가 되고, 3장에서 이에 대한 해결책이 다루어진다.

3. 아래문제 알고리즘과 계산복잡도

3.1 TTP-FT 정의와 풀이 방법

TTP-FT의 수학적 모델은 앞장에서 제시한 TTP-NFT에서 결정 변수였던 로트의 상차 시작시간과 하차 시작시간이 상수로 정해지고 나머지 부분은 동일한 모델이 된다. 본 논문에서는 TTP-FT의 풀이 방법으로 트레일러 일정 계획과 트렉터 일정계획을 순서대로 제시하겠다. 우선 트레일러가 어떤 로트들을 순서대로 실을지 결정을 하게 되고, 여기서 나오는 트레일러의 이동 경로가 트렉터가 수행해야 할 일의 집합이 되고, 이를 수행할 트렉터 일정계획이 세워진다

.

3.2 트레일러 일정 계획

로트의 싣고 내리는 일의 시작시간과 끝나는 시간을 이용하여 다음과 같은 트레일러-그래프(G-trailer)로 나타낼 수 있다.

G-trailer(N,A)

N: $n \in N$, 노드n은 다음과 같은 4가지 속성을 가지며 각 로트 에 대응하여 생성.

n_{lp}: 작업의 시작장소(w)

n_{up}: 작업의 끝나는 장소(d)

n_{st}: 작업의 시작시간(ls)

n_{et}: 작업의 끝나는 시간(us)

A: arc(i,j) 트렉터의 누 노드 i,j간 이동가능성

i에서 j로 arc 설정 $i_{et}+T(d(i_{up},j_{sp}),v) \leq j_{st}$ 일 때

i에서 j로 arc 없음 그외

위와 같이 그래프를 생성한 후 노드들을 작업의 시작시간 증가 순으로 번호를 준다. 여기에서 우리는 G-trailer 그래프는 무순환 그래프가 됨을 쉽게 확인 할 수 있고 이 특징으로 인하여 MCCD를 이용하여 쉽게 문제를 풀 수 있게 된다.

MCCD

→Simple 그래프에서 쇄 분해 G=(V,A)는 쇄 $C_1=(V_1,A_1)$, ….$Cr=(V_r,A_r)$ 의 집합으로 각각 노드 집합들은 서로 교집합이 없고 합집합은 V가 된다. 여기서 r은 분해된 쇄의 개수를 나타내며

MCCD문제는 최소개수의 쇄을 구하는 문제다. 일반적인 MCCD문제는 어려운 문제로 알려져 있지만 순환이 없는 그래프에서는 최대개수 이분매칭(MCBM)을 이용해 풀 수 있다. 자세한 풀이 방법은 [1]과[3]을 참고한다. 이와 같이 문제를 풀이하면 최소 개수의 쇄과 각 쇄에 속하는 노드의 집합을 알게 된다. 이 해는 곧 트레일러 일정 계획의 해에 대응되어, 쇄의 개수는 필요한 트레일러의 수, 쇄에 속하는 노드의 순서와 집합은 각 트레일러가 실어야 할 로트다. 결론적으로 트레일러 일정 계획은 다항시간 알고리즘을 가지는 쉬운 문제다.

3.3 트렉터 일정계획

트레일러 일정이 세워지면 이에 따라 트렉터가 수행해야 할 활동들이 정해진다. 활동들은 각 트레일러를 창고에서 첫 번째 수행할 로트의 창고로 운송하는 공급(feed) 활동, 로트를 창고에서 부두로 운송하는 버스(bus)활동, 트레일러가 현재 로트를 부두에 내려놓고 다른 로트를 싣기 위해 새로운 창고로 옮겨지는 재사용(reuse)활동, 그리고 트레일러를 해당 부두에 옮겨 마지막 로트를 내린 후 창고로 다시 트레일러를 운송하는 되돌림(return)활동으로 구성된다. 여기에서 두 가지 경우를 생각 할 수 있다. 첫 번째 경우는 상/하차 작업시간외에는 트레일러를 작업장에 위치시킬 수 없는 경우다. 즉 상/하차 작업장의 공간이 충분치 않아서 상차 작업이 끝나면 즉시 트렉터가 트레일러를 빼내어야 하고 또한 하차 작업이 시작되는 시점에 트레일러가 도착해야 한다. 두 번째 경우는 작업장의 공간이 충분해서 상/하차 시간에 상관없이 트레일러가 작업장에 위치할 수 있는 경우다. 두 경우에 트렉터 일정 계획은 쉬운 문제 혹은 어려운 문제로 나누어진다. 이를 설명하기 위해 트렉터-그래프(G-tractor) 문제로 나타낸다.

G-tractor(N,A)

N: $n \in N$, 모든 트렉터의 각 활동에 대해 대응하는 하나의 노드가 생성된다. 노드 n은 4가지 속성을 가진다.

n_{lp}: 활동의 시작장소

n_{up}: 활동의 끝나는 장소

n_{st}: 활동의 시작시간

n_{et}: 활동의 끝나는 시간

A: arc(i,j)

(i,j) arc 설정 $i_{et}+T(d(i_{up},j_{sp}),v) \leq j_{st}$ 일 때

(i,j) arc 없음 그외

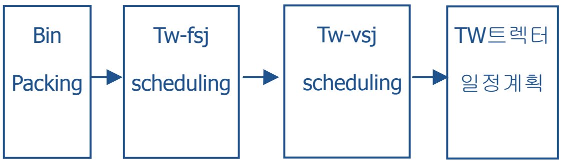
예를 들어서 로트 1이 시작위치 w에서 도착위치 d로 움직이는 버스활동에 대응해 하나의 노드가 정의되고 시작장소는 w, 도착장소는 d, 시작시간은 상차 작업이 끝나는 시간, 끝나는 시간은 하차 작업이 시작하는 시간이 된다.

앞에서 언급한 첫 번째 경우 즉 상/하차 작업시간에만 트레일러가 작업장에 위치할 수 있는 경우는 트레일러 일정 계획과 마찬가지로 MCCD를 이용해 다항시간에 풀 수 있는 쉬운 문제가 된다. 하지만 두 번째 경우, 즉 상/하차 작업시간에 상관없이 트레일러가 작업장에 위치할 수 있는 경우에는 위 그래프에서 노드가 가지는 시작시간과 끝나는 시간의 차이가 트렉터의 이동이 일어나야 하는 시간창(TW-time window)이 된다. 이로 인하여 어려운 문제가 됨을 증명한다.

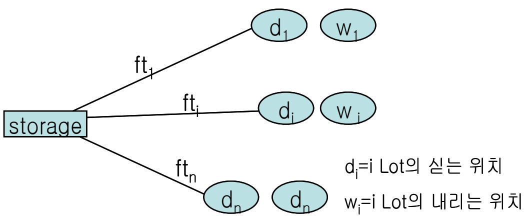
Theorem1. 시간창(TW)이 있는 트렉터 일정계획문제는 어려운문제(NP-hard)다.

Proof. 시간창 트렉터 일정계획문제의 정의는 Appendix1을 참고한다. 우리 문제의 결정버전(decision version)을 생각해보면 '일정 계

획 수립시간 T안에 모든 활동을 K대의 트렉터로 수행할 수 있는가’ 와 같다. 문제의 어려움을 증명하기 위해 NP-hard문제인 bin-packing에서부터 [그림1]과 같이 3단계로 변환을 하게 되는데 그 중 Time Window Variable Size Job(TWVSJ) scheduling이 NP-hard임을 Appendix에서 보이고, 이 문제에서 시간창트렉터 일정계획 문제의 결정버전으로의 변환을 본문에서 보인다.



[그림1] Theorem2의 문제 변환 과정



[그림2]TW트렉터 일정계획 Special case

변환은 다음과 같이 TWVSJ-scheduling 인스턴스를 특수사례 시간창 트렉터 일정계획문제 인스턴스([그림2]참조)로 바꾸어준다.

- 트렉터의 속도는 모든 경우 1이다.
- $d(w_i,d_i)=1$ for all i
- 사용가능한 트렉터 대수는 기계의 숫자와 같다. (nTT=n’)
- 공급과 버스활동들 (a_{m+i}, i=m+1~3m)을 만들어준다.
  - 공급활동(i=m+1~2m) $t_i=ft_i+1$ $es_i=0$ $le_i=M$ $sl_i=s$, $el_i=w_i$ $type_i$=공급
  - 버스활동(i=2m+1~3m) $t_i=1$ $es_i=0$ $le_i=M$ $sl_i=w_i$, $el_i=d_i$ $type_i$=버스
  - r을 모든 위치들 사이 거리 중 가장 큰 값으로 한다. $r=\max(d(p_i,p_j))$ for all i,j
  - $M=m\{(\sum fti+1)+r\}+m\{1+r\}+r$
- 각 일 j_i는 되돌림 활동 a_i에 대응한다.
  - $es_i = M+es_i$, $le_i = M+le_i$, $sl_i = p_i$, $el_i = p_{i+m}$, $t_i = fti$, $type_i$ = 되돌림
- 활동계획 수립기간: S+M

만약 하나의 트렉터가 다른 되돌림 활동을 수행한 후에 i활동을 수행하게 되면 창고에서 i활동의 시작위치로 오는 시간이 소요된다. 이 시간은 창고로 돌아가는 시간(ft_i)과 같고 이를 합하면 $2ft_i$가된다. 따라서 트렉터가 처음 수행하는 활동이 아닌 경우 2배의 운송시간이 소요된다. 임의로 만들어진 공급과 버스활동들이 하나의 트렉터로 수행가능 하도록 큰 상수시간(M)을 정해주었다. 위와 같이 변환해주면 시간창트렉터 일정계획은 TW-VSJ schedule이 k대로 될 때만 가능하며 반대 방향으로도 성립 함으로 NP-hardness가 증명된다.

4. 결론

본 논문에서는 기존과는 다른 트렉터와 트레일러 두 가지 운송도구가 사용되는 문제를 소개했다. 이 문제는 일반적으로 어려운 문제(NP-hard)임을 증명하였다. 하지만 로트의 상/하차 시간을 발견적 방법으로 정해주고 나면, 트레일러와 트렉터의 일정계획을 순차적으로 짜는 문제가 되고, 트레일러의 일정계획을 MCCD를 이용하여 다항시간 알고리즘을 제시했다. 또한 트렉터의 일정 계획은 경우에 따라 다항시간 내에 풀수 있는 알고리즘을 제시했으며 어려운 문제인 경우에는 이를 증명하였다.

필요한 연구로는 우선, 발견적 방법으로 상/하차 작업시간을 정한 후, 트레일러와 트렉터의 일정계획을 순차적으로 정한 해가

전체 최적해에 비해 얼마나 나쁠 수 있는지 (Worst-case analysis)를 제시해야 하겠다. 또한 논문의 서론에서 가정한 제약조건이 없는 보다 일반적인 TTP문제에서 분류되는 다양한 문제들의 다항시간 알고리즘의 존재 여부와 계산 복잡도에 관해 정리가 필요하다.

Reference

[1] 강동욱, 철강 코일제품 해송출하계획을 위한 그래프 모형, 석사학위논문, 2005

[2] 이정원, 철강 코일제품 수송 팔레트설계 최적화, 석사학위논문, 2004

[3] C.H. Papadimitriou and Steiglitz K. Combinatorial Optimization, Dover Publications: Mineeola, NY, 1988.

[4] G.Steiner, On the k-path partition of graphs, Theoretical Computer Science, 2147-2155, 2003

[5] I.Averbakh, O.Berman and I.Chernykh, The routing open-shop problem on a network: Complexity and Approximation, European Journal of Operational Research, In Press

[6] Jean-francois Cordueau, A branch and cut algorithm for dial a ride problem, submitted to Operations Research, 2003-9

[7] M.R.Garey, D.S.Johnson, Computers and Intractability: a guide to the theory of NP-completeness, Freeman, San Fransisco, 1979

[8] Y. Dumas, J,Desrosiers and F.Soumis, The pickup and delivery problem with time windows, Volume 54, Issue 1, 1991-10, 7-22

Appendix

1. 시간창트렉터 일정계획 문제 정의

인스턴스

- 일정 계획 수립기간: T
- 위치의 집합L={s,p_1,...,p_i,...p_n}
- 활동의 집합 A={a_1,…a_l}
- a_i = (sl_i,el_i,es_i,le_i,t_i,$type_i$), 시작위치, 끝나는 위치, 시작할 수 있는 가장 빠른 시간, 끝내야 하는 가장 늦은 시간, 수행시간,종류
  종류 = {공급, 버스, 재사용, 되돌림}
- 두 위치 p_h 와 p_i 사이의 거리는 d(p_h,p_i)로 표시한다.
- 트렉터의 속력 v 는 다음과 같다.
  - a: 로트를 싣은 트레일러를 실은 상태
  - b: 빈 트레일러를 싣은 상태
  - c: 트렉터만 이동할 때
- T(d(p_h,p_i),v): 두 위치 p_h 와 p_i 사이를 v의 속도로 이동할 때 소요시간

목적

모든 활동을 수행하기 위한 필요한 최소의 트렉터 대수를 구하라

2. Time Window Variable Size Job(TWVSJ)scheduling 의 NP-hardness

우선 Time Window Fixed Size Job (TWFSJ)이 NP-hard임을 보이고 이 문제를 통해 우리 문제가 NP-hard임을 보인다.

Lemma1. TWFSJ는 NP-hard이다.

인스턴스

- 데드라인 S, n개의 기계
- m개의 일 집합 J'={j'_1,j'_2,…j'_m} j'_i={es_i,le_i,t_i}
  → Time window(es_i,le_i) 수행시간 t_i

목적

데드라인 S안에 모든 일들을 n대의 기계로 수행할 수 있는가?

NP-hardness

NP-hard인 Bin-packing문제가 TW-FSJ의 special case임을 보인다. 즉 TW-FSJ문제의 모든 일에 time window를 (0,S)로 정하면 이는 곧 정해진 대수의 기계(Bin)에 다양한 크기의 일(Item)을 채워 넣는 Bin-packing문제가 된다.

Lemma2. TWVSJ는 NP-hard다

인스턴스

- 기계 n'대, 일 m' 개. 데드라인S'
- 일 집합 J = {j_1,j_2,…j_m} j_i=(es_i,le_i,t_i) 가능한 최초 시작시간, 가능한 최후 종료시간, 수행시간)
  일의 수행시간 t_i는 수행 할 기계에 앞서 수행된 일이 있는지 여부에 따라 달라진다.
  t_i = ft_i i번째 일을 해당 기계가 첫 번째수행할 때
  = $2ft_i$ 그 외 경우

목적

데드라인 S'내에 n대의 기계로 모든 일을 수행할 수 있는가?

NP-hardness

TW-FSJ scheduling이 다음과 같이 변환된다.

- 같은 개수의 일과 기계를 만들고, 데드라인의 2배 더하기 1로 정해준다. (n'=n, m'=m, S'=2S+1)
- 각각의 일 j_i=(es_i,le_i,t_i)로부터 대응하는 일 j'_i를 만든다. j'_i=(es_i,le_i,t_i)
- 또한 기계 대수만큼의 새로운 일을 만든다.
  j_i(i=m+1~m+n) where j_i=(0,1,10)

이와 같이 변환하면 TW-FSJ scheduling문제가 TW-VSJ scheduling문제의 special case가 됨을 확인할 수 있고 NP-hardness가 증명된다.