1 문제 정의 및 고찰

본 프로젝트 문제에 대해 고찰하여 최적화해야 할 대상을 확인하고, 아래에서 이를 수리모형으로 바꾸어 분석을 시도한다.

1.1 문제 정의

본 문제는 프로젝트 수행 기간 중 작업들을 수행할 수 있는 노동자를 각 작업에 최적으로 분배했을 때 최단 수행 기간이 얼마인지를 구하는, 즉 **답이 정해져 있는** 문제이다. 이 문제에서 중요한 **제약 조건**은 다음과 같다. **(1)** 노동자는 10명이다. **(2)** 작업 간의 선후관계가 존재한다. **(3)** 작업량을 초과하더라도 그 당일에는 다음 후위 작업으로 넘어갈 수 없다. **(4)** 이 문제는 정수선형최적화 문제이다. **(5)** 한 작업에 노동자가 배치되면, 그 이후에는 그 작업이 끝날 때까지 노동자가 항상 한 명 이상은 배치되어야 한다. 이러한 제약 조건에서 몇 일만에 모든 프로젝트를 끝낼 수 있는지를 알아보는, 즉 작업 수행 일수라는 **목적 함수**를 최소화를 시켜야 하는 최적화 문제이다.

1.2 문제 고찰

우리가 사용하는 **XpressMP** Programme은 모든 제약식과 목적 함수가 선형으로 표현되어야만 해결할 수 있다. 따라서 우리가 해야 할 근본적인 작업은, 이 문제를 Linear Programming으로 모형화하는 것이다.이 문제를 Integer Optimisation Problem이면서 동시에 Binary Optimisation Problem으로 모형화할 수 있음을 뒤에서 보일 것이다. 또한, 제약식과 변수의 총 합이 5000을 넘을 수 없기 때문에, 이러한 문제를 해결할 수 있는 방법을 제약 조건의 모형화 부분에서 제시할 것이다.

2 모형 수립

현재의 상황을 그대로 선형으로 나타내는 것은 상당히 어려울 뿐더러, 목적함수를 제한된 변수만으로 표현하는 것 또한 어렵다. 이를 해결하는 과정에서 이진 변수가 도입된다. 여러 조건식을 모두 제약식으로 넣고 목적함수를 선형으로 만들기 위한 좋은 방법은 이진 변수를 활용하는 것이다. 현재의 문제는 아래의 Figure 1과 같다.

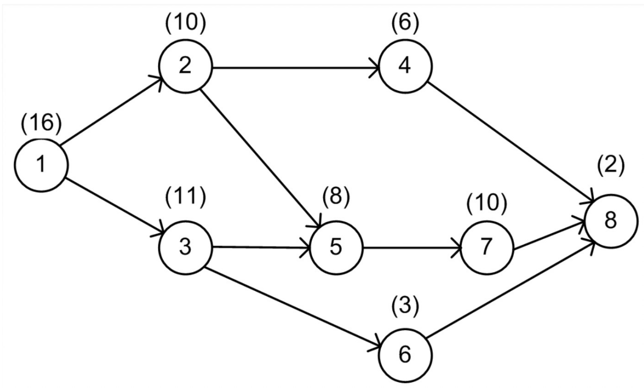

Figure 1: 주어진 문제 사항에 관한 그림

2.1 변수의 정의

먼저 결정 변수 $x_{ij}(i \in \{1, 2, ..., 8\}, j \in \{0, 1, ..., 33\})$을 정의하는데, 이는 i번째 프로젝트의 j번째 날에서의 노동자 투입 명수를 의미한다. 다음으로, 제약 조건과 목적 함수를 표현하기 위한 이진 변수 $y_{ij}, z_{ij}, c_{ij}(i \in \{1, 2, ..., 8\}, j \in \{0, 1, ..., 32\})$와 $a_j(j \in \{0, 1, ..., 32\})$, $b_{ij}(i \in \{1, 2, ..., 10\}, j \in \{0, 1, ..., 32\})$를 정의한다. 편의상, 각 작업에 필요한 작업량 상수를 $d_i(i \in \{1, 2, ..., 8\}$로 표현하자. 그리고, 혼합정수모형화에 필요한 충분히 큰 상수 $M = 999$를 정의하자.

2.2 제약 조건의 모형화

각각의 제약 조건에 대한 모형화 식을 작성하고, 그 이유를 설명하겠다. 이진 변수를 이용한 혼합정수모형화는 **경영과학 1**에서 이해한 것으로 가정한다.

(노동자 조건) 하루에 노동자는 10명 일한다.
$\sum_{i=1}^{8} x_{ij} = 10 \, (i \in \{1,2,...,8\}, j \in \{0,1,...,33\})$

(non-preemptive 조건) 어떤 작업이 시작되면, 끝날 때까지 최소 한 명의 노동자가 배정되어야 한다. $x_{iq} > 0$이면 $x_{i(q+1)} > 0$ 또는 $\sum_{j=0}^{q} x_{ij} \geq d_i \, (q \in \{0,1,...,32\})$. 뒤의 조건은 만약 다음 수가 0보다 크지 않다면 이는 모든 일이 종료되었음을 의미하는 조건이다. 이를 이진 변수를 도입하여 혼합정수모형으로 바꾸면 다음과 같다.
$x_{iq} \leq My_{iq}, 1 - x_{i(q+1)} \leq M(2 - y_{iq} - z_{iq}), d_i - \sum_{j=0}^{q} x_{ij} \leq Mz_{iq} \, (i \in \{1,2,...,7\}, q \in \{0,1,...,32\})$.

(완료 작업 조건) 완료된 작업에는 노동자를 투입하지 않는다. (단, 코드에서 이 조건을 무시해도 최적해를 구할 수 있는데 이는 후술한다.) $\sum_{k=0}^{q} x_{ik} \geq d_i$이면 $\sum_{k=q+1}^{33} x_{ik} \leq 0$. 뒤의 조건은 등호조건을 부등호를 이용하여 표현한 것이다. 이렇게 해야 혼합정수모형으로 변형하기 쉽기 때문이다. 이를 이진 변수를 도입하여 혼합정수모형으로 바꾸면 다음과 같다.
$\sum_{k=0}^{q} x_{ik} - (d_i - 1) \leq Mc_{iq}, \sum_{k=q+1}^{33} x_{ik} \leq M(1 - c_{iq}) \, (i \in \{1,2,...,7\}, q \in \{0,1,...,32\})$.

(작업의 선후 관계 조건) 유향 간선 집합 $\mathcal{N} = \{(1,2),(1,3),(2,4),(2,5),(3,5),(3,6),(4,8),(5,7),(6,8),(7,8)\}$을 생각하고, 이를 순서가 있는 자연수 집합 $\mathbb{Z}_{10} = \{1,2,...,10\}$에 대응하여 집합(전순서집합)으로 생각하자. 예컨대 $3 = (2,4)$이다. 이 때 $j = (i_1, i_2) \in \mathcal{N}$에 대해, $\sum_{k=0}^{q} x_{i_1 k} < d_i$이면 $\sum_{k=0}^{q} x_{i_2 k} \leq 0$이다. 이는 선위 작업이 마치지 않았으면 후위 작업은 작업된 양이 0이어야 한다는 의미이다. 이를 이진 변수를 도입하여 혼합정수모형으로 바꾸면 다음과 같다.
$d_i - \sum_{k=0}^{q} x_{i_1 k} \leq Mb_{j,q}, \sum_{k=0}^{q} x_{i_2 k} \leq M(1 - b_{jq}) \, (j \in \{1,2,...,10\}, q \in \{0,1,...,32\})$.

(목적함수) 목적함수는 기본적으로 작업 8을 해결하기까지 걸리는 시간이다. 하지만 이를 주어진 변수의 선형 결합으로 바로 나타내기는 어렵기 때문에, 이진 변수를 활용하고 또한 목적함수를 최소화시키는 것과 동치가 되는 수정된 목적함수를 사용한다. 그리고 그 목적함수를 최대화하는 문제를 해결하기로 하자. 작업 8에 도달하기 전까지 $x_{8j} = 0$이고, 작업 8에 도달한 직후부터 $x_{8j} > 0$이다. 작업 8에 도달하면 그때부터는 최대 가용 노동자 수인 10명을 모두 작업에 투자할 것이고, 따라서 $\frac{2}{10} = 0.2$보다 큰 최소의 정수인 1이 더해지는 값이 총 작업 일수라고 생각해도 무방하다. 한편 최대한 빠르게 작업을 해결하기 위해서 마지막 작업 8에서 소요되는 시간은 상수이므로 이는 최적화 문제에서 무시할 수 있고, 또한 시간을 최소화하는 대신 아래의 이진 변수의 합을 최대화하는 것으로 충분한데, 왜냐하면 이진 변수의 값 중 0이 적은 것이 소요되는 시간이 최소가 되는 것이고, 이 때 이진 변수의 합은 최대가 되기 때문이다. 즉 수정된 목적함수는 아래와 같이 이진 변수의 합으로 정의되며, 이를 최대화하는 문제는 원래 목적함수를 최소화하는 문제와 같다. 따라서 이 문제 또한 이진 변수를 도입하여 혼합정수모형으로 바꾸어 목적함수를 표현할 수 있으며 아래와 같다.
$a_q \leq Mx_{8q}, \frac{1}{M} x_{8q} \leq a_q \, (q \in \{0,1,...,33\})$, maximise $\sum_{q=0}^{33} a_q$.

(부연 설명) 33과 32라는 숫자는 변수의 개수가 5000개를 넘는 것을 막기 위해 임의적으로 설정된 숫자이다. 이렇게 임의적인 제약을 가해도 최적해가 제대로 구해질 수 있는 이유는 실제 해가 9일이기 때문으로, 33이라는 숫자보다 훨씬 작기 때문이다. 문제가 될 가능성은 33으로 제약했을 때 최소 작업 일수가 33으로 출력될 때뿐이다. 그 때만 숫자를 수정하면 된다.

2.3 정수 선형 최적화 모형으로의 표현

위의 모형을 아래와 같은 최적화 모형으로 표현하면 다음과 같다.

$$
\begin{aligned}
\max_{a_j, b_{ij}, c_{ij}, x_{ij}, y_{ij}, z_{ij}} \quad & \sum_{j=0}^{33} a_j \\
\text{s.t.} \quad & \sum_{i=1}^{8} x_{ij} = 10, \\
& (i \in \{1,2,...,8\}, j \in \{0,1,...,33\}), \\
& x_{iq} \le My_{iq}, 1 - x_{i(q+1)} \le M(2 - y_{iq} - z_{iq}), d_i - \sum_{j=0}^{q} x_{ij} \le Mz_{iq}, \\
& (i \in \{1,2,...,7\}, q \in \{0,1,...,32\}), \\
& \sum_{k=0}^{q} x_{ik} - (d_i - 1) \le Mc_{iq}, \sum_{k=q+1}^{33} x_{ik} \le M(1 - c_{iq}), \\
& (i \in \{1,2,...,7\}, q \in \{0,1,...,32\}), \\
& d_i - \sum_{k=0}^{q} x_{i_1 k} \le Mb_{j,q}, \sum_{k=0}^{q} x_{i_2 k} \le M(1 - b_{jq}), \\
& (j \in \{1,2,...,10\}, q \in \{0,1,...,32\}), \\
& a_q \le Mx_{8q}, \frac{1}{M} x_{8q} \le a_q, \\
& (q \in \{0,1,...,33\}), \\
& a_j, b_{ij}, c_{ij}, y_{ij}, z_{ij} \in \{0,1\}, x_{ij} \in \mathbb{N} \cup \{0\}, \\
& M = 999
\end{aligned}
$$

그리고 위의 수정된 목적함수가 최대일 때 그 값이 I라고 하면, 실제 우리가 구하고 싶은 최소 작업 일수는 $34 - I + \lceil \frac{d_8}{10} \rceil$ (일)이다.

3 코드

위의 Mixed Integer Optimisation Problem을 **XpressMP** Programme으로 변환한 코드는 다음과 같았다.

```
!@encoding CP949

model manpower
  options noimplicit
  uses "mmxprs"
  ! uses "mminsight" ! uncomment this line for an Xpress Insight model

  declarations
    ! 66 is the supremum, but to find the value,
    ! reduce it to 33. (assuming)
    ! if the assumed optimal value is not equal to 33, it is the real optimal value.
    x : array(1..8, 0..33) of mpvar
    y : array(1..8, 0..32) of mpvar
    z : array(1..8, 0..32) of mpvar
    a : array(0..33) of mpvar
    b : array(1..10, 0..32) of mpvar
    c : array(1..8, 0..32) of mpvar
    M : integer
    day : integer
    d : array(1..8) of integer

    Objective:linctr
  end-declarations

  initialisations from "Data/or1_project.txt"
    d
  end-initialisations

  ! sufficiently large integer
  M := 999

  !integer optimisation
  forall(p in 1..8) do
    forall(q in 0..33) do
      x(p,q) is_integer
      if q <=32 then
        y(p,q) is_binary
        z(p,q) is_binary
        c(p,q) is_binary
      end-if
    end-do
  end-do

  forall(q in 0..33) do
    a(q) is_binary
  end-do

  !1->2, 1->3, 2->4, 2->5, 3->5, 3->6,
  !4->8, 5->7, 6->8, 7->8 ; hierarchy
  forall(p in 1..10) do
    forall(q in 0..32) do
```

```
      b(p,q) is_binary
    end-do
  end-do

  !all-round working worker
  forall(q in 0..33) do
    sum(p in 1..8) x(p,q) = 10
  end-do

  !#1 condition
  forall(q in 0..32) do
    x(1,q) <= M*y(1,q)
    1-x(1,q+1) <= M*(2-y(1,q)-z(1,q))
    d(1)-sum(k in 0..q) x(1,k) <= M*z(1,q)
    sum(k in 0..q) x(1,k) - d(1) + 1 <= M*c(1,q)
    sum(k in (q+1)..33) x(1,k) <= M*(1-c(1,q))
  end-do

  !#2 condition
  forall(q in 0..32) do
    x(2,q) <= M*y(2,q)
    1-x(2,q+1) <= M*(2-y(2,q)-z(2,q))
    d(2)-sum(k in 0..q) x(2,k) <= M*z(2,q)
    sum(k in 0..q) x(2,k) - d(2) + 1 <= M*c(2,q)
    sum(k in (q+1)..33) x(2,k) <= M*(1-c(2,q))
  end-do

  !#3 condition
  forall(q in 0..32) do
    x(3,q) <= M*y(3,q)
    1-x(3,q+1) <= M*(2-y(3,q)-z(3,q))
    d(3)-sum(k in 0..q) x(3,k) <= M*z(3,q)
    sum(k in 0..q) x(3,k) - d(3) + 1 <= M*c(3,q)
    sum(k in (q+1)..33) x(3,k) <= M*(1-c(3,q))
  end-do

  !#4 condition
  forall(q in 0..32) do
    x(4,q) <= M*y(4,q)
    1-x(4,q+1) <= M*(2-y(4,q)-z(4,q))
    d(4)-sum(k in 0..q) x(4,k) <= M*z(4,q)
    sum(k in 0..q) x(4,k) - d(4) + 1 <= M*c(4,q)
    sum(k in (q+1)..33) x(4,k) <= M*(1-c(4,q))
  end-do

  !#5 condition
  forall(q in 0..32) do
    x(5,q) <= M*y(5,q)
    1-x(5,q+1) <= M*(2-y(5,q)-z(5,q))
    d(5)-sum(k in 0..q) x(5,k) <= M*z(5,q)
    sum(k in 0..q) x(5,k) - d(5) + 1 <= M*c(5,q)
    sum(k in (q+1)..33) x(5,k) <= M*(1-c(5,q))
  end-do

  !#6 condition
```

```
  forall(q in 0..32) do
    x(6,q) <= M*y(6,q)
    1-x(6,q+1) <= M*(2-y(6,q)-z(6,q))
    d(6)-sum(k in 0..q) x(6,k) <= M*z(6,q)
    sum(k in 0..q) x(6,k) - d(6) + 1 <= M*c(6,q)
    sum(k in (q+1)..33) x(6,k) <= M*(1-c(6,q))
  end-do

  !#7 condition
  forall(q in 0..32) do
    x(7,q) <= M*y(7,q)
    1-x(7,q+1) <= M*(2-y(7,q)-z(7,q))
    d(7)-sum(k in 0..q) x(7,k) <= M*z(7,q)
    sum(k in 0..q) x(7,k) - d(7) + 1 <= M*c(7,q)
    sum(k in (q+1)..33) x(7,k) <= M*(1-c(7,q))
  end-do

  !#8 condition
  forall(q in 0..32) do
    x(8,q) <= M*y(8,q)
    1-x(8,q+1) <= M*(2-y(8,q)-z(8,q))
    d(8)-sum(k in 0..q) x(8,k) <= M*z(8,q)
  end-do

  forall(q in 0..33) do
    a(q) <= M*x(8,q)
    (1/M)*x(8,q) <= a(q)
  end-do

  !1->2,3 condition
  forall(q in 0..32) do
    d(1)-sum(k in 0..q) x(1,k) <= M*b(1,q)
    d(1)-sum(k in 0..q) x(1,k) <= M*b(2,q)
    sum(k in 0..q+1) x(2,k) <= M*(1-b(1,q))
    sum(k in 0..q+1) x(3,k) <= M*(1-b(2,q))
  end-do

  !2->4,5 condition
  forall(q in 0..32) do
    d(2)-sum(k in 0..q) x(2,k) <= M*b(3,q)
    d(2)-sum(k in 0..q) x(2,k) <= M*b(4,q)
    sum(k in 0..q+1) x(4,k) <= M*(1-b(3,q))
    sum(k in 0..q+1) x(5,k) <= M*(1-b(4,q))
  end-do

  !3->5,6 condition
  forall(q in 0..32) do
    d(3)-sum(k in 0..q) x(3,k) <= M*b(5,q)
    d(3)-sum(k in 0..q) x(3,k) <= M*b(6,q)
    sum(k in 0..q+1) x(5,k) <= M*(1-b(5,q))
    sum(k in 0..q+1) x(6,k) <= M*(1-b(6,q))
  end-do

  !4->8 condition
  forall(q in 0..32) do
```

```
    d(4)-sum(k in 0..q) x(4,k) <= M*b(7,q)
    sum(k in 0..q+1) x(8,k) <= M*(1-b(7,q))
  end-do

  !5->7 condition
  forall(q in 0..32) do
    d(5)-(sum(k in 0..q) x(5,k)) <= M*b(8,q)
    sum(k in 0..q+1) x(7,k) <= M*(1-b(8,q))
  end-do

  !6->8 condition
  forall(q in 0..32) do
    d(6)-sum(k in 0..q) x(6,k) <= M*b(9,q)
    sum(k in 0..(q+1)) x(8,k) <= M*(1-b(9,q))
  end-do

  !7->8 condition
  forall(q in 0..32) do
    d(7)-sum(k in 0..q) x(7,k) <= M*b(10,q)
    sum(k in 0..(q+1)) x(8,k) <= M*(1-b(10,q))
  end-do

  Objective := sum(q in 0..33) a(q)

  maximise(Objective)

  writeln("Minimal value of manpower dist problem : ", 34-getobjval+ceil(d(8)/10), " days") ! 9 days

  writeln("")

  forall(p in 1..8) do
    forall(q in 0..8) do
      write(getsol(x(p,q))," ")
    end-do
    writeln("")
  end-do
end-model
```

위의 코드를 실행시켰을 때 출력된 결과는 아래와 같다. 최적화 문제의 편의를 위해 목적함수가 날짜 그 자체가 아니도록 설정했기 때문에, 위의 코드에서와 같이 구한 변수를 사용하여 값을 계산해줄 때 실제 최소 작업 일수가 출력된다.

```
FICO Xpress Mosel 64-bit v5.8.0, FICO Xpress v8.13.0
(c) Copyright Fair Isaac Corporation 2001-2021. All rights reserved
Compiling Untitled1.mos to out/Untitled1.bim with -g
Running model
Minimal value of manpower dist problem : 9 days

10 10 0 0 0 0 0 0 0
0 0 3 1 6 0 0 0 0
0 0 7 9 0 0 0 0 0
0 0 0 0 0 2 10 0 0
0 0 0 0 0 8 0 0 0
0 0 0 0 4 0 0 0 0
```

```
0 0 0 0 0 0 0 10 0
0 0 0 0 0 0 0 0 10

Process exited with code: 0
```

4 결과 분석 및 제언

XpressMP Programme을 가동한 결과 **최소 작업 일수는 9일이다.** 최소 작업 일수는 유일한 값을 가지지만, 이때 가능한 실제 작업 공정은 유일하지 않다. 또, 출력된 최소 작업 일수에서의 인력 분배는 언뜻 보기에 상당히 비효율적으로 보인다. 하지만 이는 **완료 작업 조건**에 의해 생긴 문제이며, 일을 먼저 하면 가속하여 다음 일을 할 수 없기 때문에 발생한다. 또한, 반드시 노동자 모두가 매일 어떠한 작업에라도 배정되어야 하기 때문에, 이에 맞추어 실제 작업 필요량을 훨씬 상회하는 작업량을 수행하기도 한다.

문제에서의 가정과 같이 모든 사람의 일률이 동일한 일률을 가지는 것과 달리, 현실에서는 한 사람이 같은 노동을 반복할 경우 효율성이 증가되는 것이 일반적이다. 이러한 효율성 함수를 계단형, 혹은 연속형으로 상정하여 문제를 해결할 수 있을 것이다. 또한, 현실에서는 노동의 종류에 따라 여러 사람이 같은 노동에 참여할 수록 노동의 효율성이 극대화되거나 오히려 극소화되는 경우도 생각해볼 수 있을 것이다. 이러한 문제 역시 최적화 이론을 통하여 해결할 수 있다.

이렇게 최적화 이론은 응용 수학의 눈부신 결과로서 공리속에서만 존재하지 않고 현실에 직접적으로 적용되어 많은 문제를 해결하는 데에 큰 기여를 한다.