번호 :   /

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

#1. 노드 D 와 G 간의 최단거리경로와 최단거리 구하기

1) 목적함수 : 경로를 지난 횟수 * 경로의 거리 의 합이 최소가 되어야 한다.
2) 제약식 : G 에서는 1번만 나가고, D 에는 한번만 들어온다.
G와 D를 제외하고는, 각 노드에 들어온 횟수는 나간 횟수와 동일하다.
각 경로를 지난 횟수는 정수다.

풀이과정 :
린도를 사용하여 풀어보자.

<untitled>

```
min 6AI+2AG+21AF+13AE+3AC+9BC+21BD+16CI+3CA+24CE+9CB+21DB
+11DE+14DF+24EC+4EI+13EA+11ED+16FD+21FA+6FH+20GI+2GA+21HI+6HF+20IG+6IA+4IE+16IC
s.t.
GI+GA = 1
BD+ED+FD = 1
GI+AI+EI+CI -IG-IC-IE-IA = 0
AI+AE+AF+AG -IA-EA-FA-GA = 0
HI+HF -IH-FH = 0
FH+FD+FA -AF-HF-DF = 0
CI+CA+CE+CB -IC-AC-EC-BC = 0
EC+EI+EA+ED -CE-IE-AE-DE = 0
BC+BD -CB-DB = 0
end
gin AI
gin AG
gin AF
gin AE
gin AC
gin BC
gin BD
gin CI
gin CA
gin CE
gin CB
gin DB
gin DE
gin DF
gin EC
gin EI
gin EA
gin ED
gin FD
gin FA
gin FH
gin GI
gin GA
gin HI
gin HF
gin IG
gin IA
gin IE
gin IC
```

OBJECTIVE FUNCTION VALUE

1) 23.00000

VARIABLE	VALUE	REDUCED COST
AI	1.000000	6.000000
AG	0.000000	2.000000
AF	0.000000	21.000000
AE	0.000000	13.000000
AC	0.000000	3.000000
BC	0.000000	9.000000
BD	0.000000	21.000000
CI	0.000000	16.000000
CA	0.000000	3.000000
CE	0.000000	24.000000
CB	0.000000	9.000000
DB	0.000000	21.000000
DE	0.000000	11.000000
DF	0.000000	14.000000
EC	0.000000	24.000000
EI	0.000000	4.000000
EA	0.000000	13.000000
ED	1.000000	11.000000
FD	0.000000	16.000000
FA	0.000000	21.000000
FH	0.000000	6.000000
GI	0.000000	20.000000
GA	1.000000	2.000000
HI	0.000000	21.000000
HF	0.000000	6.000000
IG	0.000000	20.000000
IA	0.000000	6.000000
IE	1.000000	4.000000
IC	0.000000	16.000000
IH	0.000000	0.000000

답:
최단경로는, G-> A-> I-> E-> D 이다.
이때의 최단거리는 23 이다.

번호 :   /

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

#2 선형계획 문제 풀기

제시된 목적함수를 해석하면, sin^2(i) + cos(i) 의 값은 x_i의 계수라고 말할 수 있다.

(i 에 따라서만 변하고, 그 값은 조절 불가능하기 때문)

풀이과정 :

1) 계수를 계산한 후 obj(목적함수 계수벡터) 벡터를 만든다.
2) 소수를 뽑는다.
3) cons(제약식 계수 벡터)를 만든 후 값을 채워넣는다.
   * 1:198 행은 $x_i + x_{i+1} + x_{i+2} \le 2.758$ 에 관한 계수이다.
   * 199 행 ~ 은 $x_i \le 1.56\,(i\ is\ 소수)$ 에 관한 계수이다.
4) dir(제약식 부등호 벡터) 에 “<=” 를 채워넣는다.
5) rhs(제약식 우변상수 벡터) 에 2.758을 198개, 1.56을 46개 채워넣는다.

최종적으로 lpsolve pakage를 이용하여 값을 구한다.

```r
obj <- rep(0, 200) # 목적함수 계수 벡터 c
for (i in 1: 200) {
  obj[i] <- sin(i)^2 + cos(i) }

# 소수 뽑기
p <- primes(200)
p_count <- length(p)

cons <- matrix(rep(0, 200*(198+p_count)), ncol = 200) # 제약식 계수벡터
for (i in 1:198){
  cons[i, i]<- 1
  cons[i, i+1]<- 1
  cons[i, i+2]<- 1
}
for(i in 1:p_count){
  cons[198+i, p[i]] <- 1
}
dir <- rep("<=", 198+p_count)   # 제약식 부등호 벡터
rhs <- c(rep(2.758, 198),rep(1.56, p_count))  # 제약식 우변상수 벡터 b

# 문제를 풀어보자
lp("max", obj, cons, dir, rhs)
LP1 <- lp("max", obj, cons, dir, rhs)

LP1$objval # 목적함수를 알려줌
LP1$solution # 최적해를 알려줌
```

답 :

```
> LP1$objval # 목적함수를 알려줌
[1] 166.1084
> LP1$solution # 최적해를 알려줌
  [1] 2.758 0.000 0.000 0.000 1.560 0.000 1.198 1.560 0.000 0.000 1.560 1.198 0.000 1.560 0.000 0.000 1.560 1.198
 [19] 0.000 1.560 1.198 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758
 [37] 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000
 [55] 2.758 0.000 0.000 2.758 0.000 0.000 1.560 1.198 0.000 1.560 1.198 0.000 1.198 1.560 0.000 1.198 1.560 0.000
 [73] 0.000 2.758 0.000 0.000 2.758 0.000 0.000 1.560 1.198 0.000 1.560 0.000 0.000 1.560 1.198 0.000 1.560 1.198
 [91] 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758
[109] 0.000 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 1.560 1.198 0.000
[127] 1.560 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 1.198 1.560 0.000 1.198 1.560 0.000 0.000 2.758 0.000
[145] 0.000 2.758 0.000 0.000 1.560 1.198 0.000 1.560 1.198 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758
[163] 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 1.198
[181] 1.560 0.000 1.198 1.560 0.000 0.000 2.758 0.000 0.000 2.758 0.000 0.000 1.560 1.198 0.000 1.560 1.198 0.000
[199] 0.000 2.758
```

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

#3 인력운영의 최적방안 제시

풀이과정 :

x_i 를 i 조에 배치된 인원의 수,

y_j 를 j 시간대에 서비스를 받지 못한 고객의 수 라고 하자.

1) 발생하는 총 비용은 정규직과 파트타임 출납원의 임금 + 제 시간대에 서비스를 받지 못한 고객의 손실비용 합 이므로
이를 최소화 시키는 목적함수는
min 680000x1 + 340000x2 + 425000x3 + 255000x4 + 255000x5+ 65000y1 + 65000y2 + 65000y3 + 65000y4 + 65000y5 + 65000y6 + 65000y7 + 65000y8 + 65000y9 이다.

2) 제약식
* 창구는 총 6개가 존재하므로, 각 시간대에 배치된 인원의 수는 6을 넘을 수 없다.
각 시간대에 배치된 인원의 수를 표로 정리하면 다음과 같다.
시간대 1 : x1 + x2
시간대 2 : x1 + x2
시간대 3 : x1 + x2 + x4
시간대 4 : x2 + x4
시간대 5 : x1 + x3 + x4
시간대 6 : x1 + x3
시간대 7 : x1 + x3 + x5
시간대 8 : x1 + x3 + x5
시간대 9 : x1 + x3 + x5
* 점심시간대를 제외하고 정규직은 1명 이상 배치되어야 하므로, x1 >= 1
* y_j 는 시간대에 서비스를 받지 못한 고객의 수 이다.
예시로 시간대 1에서, x1은 20명 x2 는 25 명을 상대할 수 있으므로, 은행에서 상대할 수 있는 고객의 합은 20x1 + 25x2 이다.
위 고객의 합이 50을 넘지 않는다면, 50 – (20x1 + 25x2) 가 y1 이 된다.
이를 제약식 형태로 쓰면 20x1 + 25x2 + y1 >= 50 이다. (최소화 문제이므로, 20x1 + 25x2 가 50을 넘는다면 y1 은 자동으로 0이 부여)
모든 시간대에 동일하게 적용하면
20x1 + 25x2 + y1 >= 50
20x1 + 25x2 + y2 >= 75
20x1 + 25x2 + 25x4 + y3 >= 150
25x2 + 25x4 + y4 >= 160
20x1 + 25x3 + 25x4 + y5 >= 150
20x1 + 25x3 + y6 >= 75
20x1 + 25x3 + 25x5 + y7 >= 58
20x1 + 25x3 + 25x5 + y8 >= 93
20x1 + 25x3 + 25x5 + y9 >= 88

```
<untitled>
min 680000x1 + 340000x2 + 425000x3 + 255000x4 + 255000x5+ 65000y1 + 65000y2 + 65000y3 + 65000y4 + 65000y5 + 65000y6 + 65000y7 + 65000y8 + 65000y9
s.t.
x1 + x2 <= 6
x1 + x2 + x4 <= 6
x2 + x4 <= 6
x1 + x3 + x4 <= 6
x1 + x3 <= 6
x1 + x3 + x5 <= 6
x1 + x3 + x5 <= 6
x1 + x3 + x5 <= 6
x1 >= 1
20x1 + 25x2 + y1 >= 50
20x1 + 25x2 + y2 >= 75
20x1 + 25x2 + 25x4 + y3 >= 150
25x2 + 25x4 + y4 >= 160
20x1 + 25x3 + 25x4 + y5 >= 150
20x1 + 25x3 + y6 >= 75
20x1 + 25x3 + 25x5 + y7 >= 58
20x1 + 25x3 + 25x5 + y8 >= 93
20x1 + 25x3 + 25x5 + y9 >= 88
end
gin x1
gin x2
gin x3
gin x4
gin x5
gin y1
gin y2
gin y3
gin y4
gin y5
gin y6
gin y7
gin y8
gin y9
```

답 :
발생하는 임금과, 고객의 손실을 최소로 하기 위해 하루당 필요한 금액은 6,410,000 원이다.
근무조 1은 1명, 근무조 2는 3명, 근무조 3은 3명, 근무조 4는 2명, 근무조 5에는 0명이 배치되어야 한다.
시간대 3에서 5명, 시간대 4에서 35명, 시간대 5에서 5명이 제때 서비스를 받지 못하고 은행을 떠난다.

```
       OBJECTIVE FUNCTION VALUE

       1)       6410000.

VARIABLE        VALUE
       X1         1.000000
       X2         3.000000
       X3         3.000000
       X4         2.000000
       X5         0.000000
       Y1         0.000000
       Y2         0.000000
       Y3         5.000000
       Y4        35.000000
       Y5         5.000000
       Y6         0.000000
       Y7         0.000000
       Y8         0.000000
       Y9         0.000000
```

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

#4 예측모델의 파라미터 추정

1. 목적함수

목적함수를 z = max($|d_i|$) 라고 하자. 이때 구해야 하는 것은 min z 이다.

2. 제약식

첫째로, $d_i = y_i - (w_0 + w_1 X_{1i} + w_2 X_{2i} + w_3 \log X_{1i} X_{2i})$ 이므로, $d_i + (w_0 + w_1 X_{1i} + w_2 X_{2i} + w_3 \log X_{1i} X_{2i}) = y_i$ 로 바꾸어 표현할 수 있다.
i의 값은 0~ 20 이므로, 발생되는 d_i의 총 개수는 20개 이다.
둘째로, z = max($|d_i|$) 이므로 이를 선형적으로 나타내기 위해
z >= $|d_i|$, 즉 z >= d_i 또는 z >= -d_i 로 나타낼 수 있다.
위 경우를 cons 행렬로 만든다.
행렬의 0~20 번째 열은 $d_i (i = 1, 2, \ldots 20)$ 의 계수를 나타낸다.
행렬의 21 ~ 24 번째 열은 $w_j (j = 0, 1, 2, 3)$ 의 계수를 나타낸다.
행렬의 25 번째 열은 z의 계수를 나타낸다.

위 목적함수, 제약식 조건을 바탕으로 obj, cons, dir, rhs 행렬을 제작하고 R의 선형계획법을 이용하여 w를 추정한다.
풀이과정 :

```r
x1 <- c(19.49, 19.16, 16.26, 8.45, 12.77, 1.12, 7.41, 13.27, 15.94, 9.53, 9.49, 15.49, 12.37, 14.35, 5.49, 18.18, 10.13, 10.35, 15.66, 2.80)
x2 <- c(3.69, 11.95, 8.73, 7.23, 18.37, 18.33, 13.94, 1.30, 18.56, 10.81, 18.43, 10.07, 1.28, 6.27, 5.96, 9.69, 13.35, 7.02, 16.36, 12.44)
y <- c(95.93, 139.59, 115.55, 80.52, 138.83, 88.26, 103.74, 53.37, 148.89, 96.17, 134.32, 112.62, 51.72, 98.51, 57.07, 118.92, 110.61, 79.00, 144.15, 79.09)

cons <- matrix(rep(0, 25*(20+20+20)), ncol = 25)
for (i in 1:20){
  cons[i, i] <- 1 # di
  cons[i, 21]<- 1 # w0
  cons[i, 22]<- x1[i] # w1
  cons[i, 23]<- x2[i] # w2
  cons[i, 24]<- log(x1[i]*x2[i]) #w3
}
for (i in 21:40){
  cons[i, i-20] <- 1
  cons[i, 25]<- -1
}
for (i in 41:60){
  cons[i, i - 40] <- -1
  cons[i, 25]<- -1
}

obj <- c(rep(0, 24), 1) # 목적함수 계수 벡터 c
dir <- c(rep("=", 20), rep("<=", 40))  # 제약식 부등호 벡터
rhs <- c(y, rep(0, 40))  # 제약식 우변상수 벡터 b

# 문제를 풀어보자
lp("min", obj, cons, dir, rhs)
LP1 <- lp("min", obj, cons, dir, rhs)
LP1$solution[21:24]
```

답 :
최종적으로, z = max($|d_i|$) 의 값은 16.76182 이다.
이때의 예측 모델 파라미터는,
$w_0 = 0, w_1 = 3.077866, w_2 = 4.219369, w_3 = 2.472561$ 이다.

```
Success: the objective function is 16.76182
> LP1 <- lp("min", obj, cons, dir, rhs)
> LP1$solution[21:24]
[1] 0.000000 3.077866 4.219369 2.472561
```

강원대학교 시험 답안지

			감독교수확인	이 두 호
2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

\# 5. 비선형 최적화 문제

제시된 식은 다음과 같다.

$\min f(x,y) = x^2 - xy + 2y^2 - 2x + e^{x+y}$

(1) 해석함수에서, 그래디언트 벡터를 구한 후 극점을 대입했을 때 0이 되면 극점이다.
  헤시안 행렬을 구한 후, 극점을 대입했을 때 양수이면 극소의 조건이 된다.
  제시된 식의 그래디언트 벡터는 다음과 같은 방법으로 구할 수 있다.

```
f <- function(x, y) x^2 - x*y +2*y^2 -2*x + exp(x+y)
f
Deriv(f, "x")
Deriv(f, "y")
grad <- function(x, y) c(2 * x + exp(x + y) - (2 + y), 4 * y + exp(x + y) - x)
grad(0,0)
```

위 함수를 선언하고, x와 y에 편미분하여 grad 벡터를 만든다.

극소점으로 추정되는 (0, 0) 을 대입하여 값을 확인한다.

```
> Deriv(f, "x")
function (x, y)
2 * x + exp(x + y) - (2 + y)
> Deriv(f, "y")
function (x, y)
4 * y + exp(x + y) - x
> grad <- function(x, y) c(2 * x + exp(x + y) - (2 + y), 4 * y + exp(x + y) - x)
> grad(0,0)
[1] -1  1
```

그래디언트 벡터의 결과는 -1, 1 이 되기 때문에 극점이 될 수 없다.

(2) 위 문제의 최적해와 목적함수값

위 함수는 제약식이 없는 비선형 문제이다.

풀이과정 :

```
f <- function(x) x[1]^2 - x[1]*x[2] +2*x[2]^2 -2*x[1] + exp(x[1]+x[2])
grad <- function(x) c(2 * x[1] + exp(x[1] + x[2]) - (2 + x[2]), 4 * x[2] + exp(x[1] + x[2]) - x[1])
grad(c(0,0))
x0 <- c(0,0)
option <- list("algorithm" = "NLOPT_LD_LBFGS", tol = 1e-04) # tol : 좀 더 자세하게 찾아봐라
result <- nloptr(x0 = x0,
        eval_f = f,
        eval_grad_f = grad,
        opts = option)
result$solution
result$objective
```

x를 x[1] y를 x[2] 변수로 하여 f를 선언한다.

f를 각 변수에 대해 편미분한 그래디언트 벡터를 만든다.

답 :

nloptr을 사용하여 최적해를 구한다.

최적해는 x = 0.3304495, y = -0.2017303

목적함수값은 0.7337205 이다.

```
> result$solution
[1]  0.3304495 -0.2017303
> result$objective
[1] 0.7337205
```

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371　성명: 하승연	출 제 자	이두호
			채점성적	

#6. 비선형 문제의 최적해와 목적함수값

풀이과정 :

Quadratic Progtamming을 이용하여 풀어보자.

위 비선형 목적함수의 변수는 변수 제곱의 합 밖에 없다. 따라서 P 행렬(목적함수 이차식 계수 행렬)은 대각선이 모두 2인 100*100 행렬이다.

A 행렬 (제약식 계수 행렬) 은 3*100 행렬이다.

첫 번째 행은, $i = 0, 1 \ldots 49$ 일 때 x_{2i+1} 의 계수가 $(-1)^i$ 이다.

두 번째 행은, $i = 1, 2 \ldots 50$ 일 때 x_{2i} 의 계수가 $(-1)^i$ 이다.

세 번째 행은, $i = 1, 2 \ldots 25$ 일 때 x_{4i} 의 계수가 $(-1)^{i+1}$ 이다.

위 계수를 A 행렬에 채워넣는다.

q 벡터 (목적함수 1차식 계수 행렬)는 원소의 개수가 100인 벡터이며, 값은 모두 0이다.

b 벡터 (제약식 우변상수 상수 행렬)은 제약식의 개수(3개)를 원소의 개수로 두며 값은 27, 50, 16 이다.

위에 명시한 행렬과 벡터를 바탕으로 K = $\begin{pmatrix} P & A^T \\ A & 0 \end{pmatrix}$ 행렬을 제작한다.

$\begin{pmatrix} x \\ y \end{pmatrix} = K^{-1} \begin{pmatrix} -q \\ b \end{pmatrix}$ 를 이용하여 최적 해 x를 얻어낸다.

최적해를 바탕으로 $\sum_{i=1}^{100} x_i^2$ 을 구하여 목적함수값을 구한다.

```
# 실습을 이용한 문제풀이

P <- diag(2, 100)
A <- matrix(rep(0, 100*3), ncol = 100)
for (i in 0:49){
  A[1, 2*i+1] <- (-1)^i
}
for (i in 1:50){
  A[2, 2*i] <- (-1)^i
}
for (i in 1:25){
  A[3, 4*i] <- (-1)^(i+1)
}
q <- rep(0, 100)
b <- c(27, 50, 16)
K<- matrix(0, ncol = 103, nrow = 103)
K[1:100, 1:100] <- P
K[101:103, 1:100] <- A
K[1:100, 101:103]  <- t(A)
K
result <- solve(K)%*%c(-q, b) # 행렬곱은 %*% 를 씁니다
x <- result[1:100,]
solution <- 0
for (i in 1:100){
  solution <- solution + (x[i]^2)
}
solution
x
```

답:

최적해는 x 벡터이고, 목적함수의 값은 73.58271 이다.

```
> solution
[1] 73.58721
> x
  [1]  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904
 [11] -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708
 [21]  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904
 [31] -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100
 [41]  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904
 [51] -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708
 [61]  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904
 [71] -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100
 [81]  0.5400000 -0.9879904 -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904
 [91] -0.5400000  1.5884708  0.5400000 -0.9879904 -0.5400000  0.3875100  0.5400000 -0.9879904 -0.5400000  1.5884708
```

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

#7. 비선형 최적화 문제를 R을 이용하여 최적해와 목적함수 구하기

위 문제는, 부등식 제약식을 가진 비선형 최적화 문제이다. 이 문제는 KKT 방식을 활용하여 해결할 수 있다.

1. 목적함수

먼저, 목적함수를 선언하고 각 변수에 대해 편미분 하여 dx, dy 함수를 제작한다. 이를 이용해 그래디언트 벡터를 만든다.

2. 제약식 1

부등식 제약식은 총 2개이다. (lower bound, upper bound 제외)

이때, 부등식 제약식의 그래디언트 벡터를 구해야 하는데 제약식에 절대값이 있으므로

x2 >= 0 와 x2 <= 0 로 나눠서 푼다.

이 중 목적함수가 최소인 것이 정답이 된다.

3. 제약식 2

x2를 양수인 경우와 음수인 경우로 나누었으므로, lower bound 와 upper bound 도 맞는 값으로 조절하여 설정한다.

풀이과정 :

1) x2>= 0 일 때

```
# 1. x2 가 양수
f <- function(x, y) exp((x+y)^8)*(1/cos((x+y)^4))
dx <- Deriv(f, 'x')
dy <- Deriv(f, 'y')
# objective function & its gradient vector
eval_f <- function(x)
  list("objective" = exp((x[1]+x[2])^8)*(1/cos((x[1]+x[2])^4)),
       "gradient" = c(dx(x[1], x[2]), dy(x[1], x[2]))
  )
# equality constraints & their gradient vectors
eval_g_ineq <- function(x)
  list("constraints" = c(x[1]^2 + x[2]^2 - 1,
                         exp(-x[1]) + x[2] -1 ),
       "jacobian" = rbind(c(2*x[1], 2*x[2]),
                          c(-exp(-x[1]), 1))
  )

lb <- c(0.5, 0)
ub <- c(1, 1)
x_init <- c(0.6,0.7)

optimizer <- list("algorithm" = "NLOPT_LD_MMA",
                  "xtol_rel"  = 1.0e-30)

options <- list("algorithm" = "NLOPT_LD_AUGLAG",
                "xtol_rel"  = 1.0e-30,
                "maxeval"   = 5000,
                "local_opts" = optimizer)

# model and solve problem
nlp <- nloptr(x0 = x_init,
              eval_f = eval_f,
              eval_g_ineq = eval_g_ineq,
              lb = lb,
              ub = ub,
              opts = options)
nlp$objective
nlp$solution
```

답 :

위 경우에서 목적함수는 1.005878 이다.

최적해는 x1 = 0.5, x2 = 0 이다.

```
> nlp$objective
[1] 1.005878
> nlp$solution
[1] 0.5 0.0
```

강원대학교 시험 답안지

			감독교수확인	이 두 호
2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

1) x2<= 0 일 때

```r
# x2 가 음수
f <- function(x, y) exp((x+y)^8)*(1/cos((x+y)^4))
dx <- Deriv(f, 'x')
dy <- Deriv(f, 'y')
# objective function & its gradient vector
eval_f <- function(x)
  list("objective" = exp((x[1]+x[2])^8)*(1/cos((x[1]+x[2])^4)),
       "gradient" = c(dx(x[1], x[2]), dy(x[1], x[2]))
  )

# equality constraints & their gradient vectors
eval_g_ineq <- function(x)
  list("constraints" = c(x[1]^2 + x[2]^2 - 1,
                         exp(-x[1]) - x[2] -1 ),
       "jacobian" = rbind(c(2*x[1], 2*x[2]),
                          c(-exp(-x[1]), -1))
  )

lb <- c(0.5, -1)
ub <- c(1, 0)
x_init <- c(0.6,-0.7)

optimizer <- list("algorithm" = "NLOPT_LD_MMA",
                  "xtol_rel"  = 1.0e-30)

options <- list("algorithm" = "NLOPT_LD_AUGLAG",
                "xtol_rel"  = 1.0e-30,
                "maxeval"   = 5000,
                "local_opts" = optimizer)
# model and solve problem
nlp <- nloptr(x0 = x_init,
              eval_f = eval_f,
              eval_g_ineq = eval_g_ineq,
              lb = lb,
              ub = ub,
              opts = options)
nlp$objective
nlp$solution
```

답 :
위 경우에서 목적함수는 1이다.
최적해는 x = 0.5, y = -0.3934547 이다.

```
> nlp$objective
[1] 1
> nlp$solution
[1]  0.5000000 -0.3934547
```

최종 답 :
두 경우를 종합하였을 때, 가장 작은 목적함수를 가지는 2번째 경우를 최종 답이라고 말할 수 있다.
목적함수는 1이다.
최적해는 x = 0.5, y = -0.3934547 이다.

강원대학교 시험 답안지

감독교수확인	이 두 호

2024 학년도 2학기 기말고사	시험과목 : 최적화 (2반)	전공 : AI소프트웨어학과 학번: 202322371 성명: 하승연	출 제 자	이두호
			채점성적	

#8. 겨울 방학 계획

이번 방학은 의미 있는 활동으로 채울 예정입니다. 먼저, 학교에서 진행하는 해외 봉사 활동을 통해 캄보디아에서 10일 간 건축과 교육 봉사에 참여하며 현지 문화를 체험할 계획입니다. 이후 삼척에서 영어 공부를 할 예정입니다. 여름방학 때는 영어 공부에 실패했기 때문에 이번에는 실현 가능한 범위에서 해보려고 합니다. 또 프로그래밍 언어를 공부 하며 실력을 쌓겠습니다. 특히 책을 이용한 이론 공부와 백준으로 실습을 해보려고 합니다. 내년에 3학년으로서 앞으로 진로에 대한 고민과 목표를 설정하기 위해 시간을 갖고 깊이 생각해보려 합니다.