Why are Some Problems Difficult to Solve ?

Some Problems

1. SAT (satisfiability problem)
   - Boolean 식을 TRUE 로 만드는 변수를 찾는 문제

(ex) Find $x_i \ (i = 1, \ldots, 100)$

s.t. $F(\boldsymbol{x})$ =TRUE

where

$$F(\boldsymbol{x}) = (x_{17} \vee \bar{x}_{37} \vee x_{73}) \wedge (\bar{x}_{11} \vee x_{56}) \wedge \cdots \wedge (x_2 \vee \bar{x}_{77} \vee x_{43} \vee \bar{x}_{89})$$

- Search space S

  |S| = 2^n

$$n = 100 \Rightarrow 2^{100} \approx 10^{30}$$

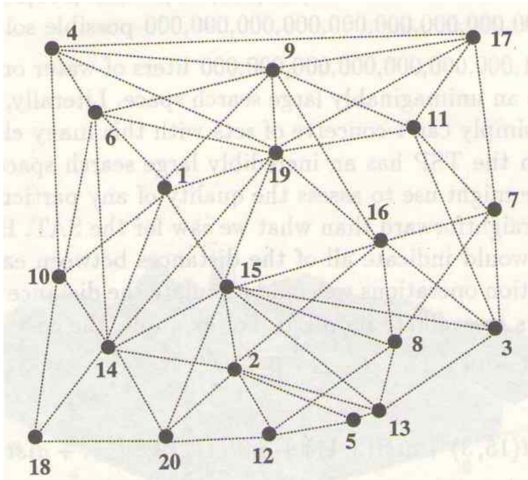
1000 strings / sec → 15 billion years (150억년)

- Evaluation Function

  해의 worse / better 를 판단하기 어려움

Some Problems

2. TSP (travelling salesman problem)
   - 모든 도시를 한번씩 방문하고 돌아오는 순서 결정 문제
   - 최소 시간 or 최단 거리
     - Symmetric TSP: $d(i,j) = d(j,i)$
     - Asymmetric TSP: $d(i,j) \neq d(j,i)$



   - Search space

$$|S| = \frac{n!}{2n} = \frac{(n-1)!}{2}$$

n=6 → 60
n=7 → 360
n=10 → 181,000
n=20 → 10,000,000,000,000,000
n=50 → 10^{62}

   - Evaluation function
     해의 worse / better 를 판단할 수 있음

Some Problems

3. NLP (nonlinear programing problem)
   – 비선형 함수의 최대/최소 값을 찾는 문제

(ex) maximize G2(x)

$$G2(\mathbf{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|,$$

subject to

$\prod_{i=1}^{n} x_i \geq 0.75,\ \sum_{i=1}^{n} x_i \leq 7.5n$, and bounds $0 \leq x_i \leq 10$ for $1 \leq i \leq n$.

– Search space

각 x_i를 0.000001 단위로 분할

$|S| = 10^{7n}$

n=50 → 10^{350}

– Evaluation function

해의 worse / better 를 판단할 수 있음

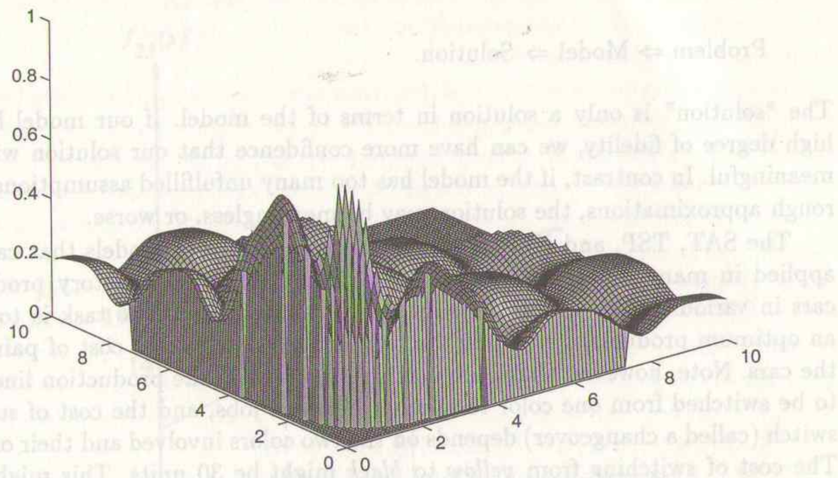


Fig. 1.2. The graph of function $G2$ for $n = 2$. Infeasible solutions were assigned a value of zero.

Modeling the Problem

- Problem → Model → Solution

- Car production schedule problem
  - n 개의 color 를 painting
  - Cost of switching

    Yellow -> black: 30, black -> yellow: 80

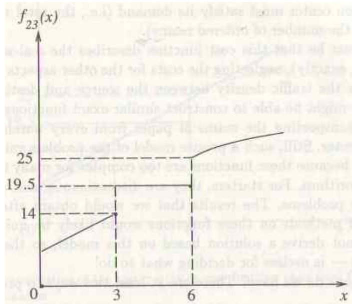
    Yellow -> Green: 50, ……
  - Find a job sequence to minimize the total cost of switching

  ☞ n-city asymmetric TSP

Modeling the Problem

- Transportation problem
  - n warehouses (source)
  - k distribution center (destination)
  - f_{ij} : transportation cost between warehouse i and destination center j $(i = 1, \cdots, n, \quad j = 1, \cdots, k)$

$$f_{23}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 4 + 3.33x & \text{if } 0 < x \le 3 \\ 19.5 & \text{if } 3 < x \le 6 \\ 0.5 + 10\sqrt{x} & \text{if } 6 < x, \end{cases}$$



x: quantity of supplies

  - Find quantity of supplies to minimize the total transportation cost

☞ NLP

$$\text{minimize } \sum_{i=1}^{n} \sum_{j=1}^{k} f_{ij}(x_{ij}),$$

subject to

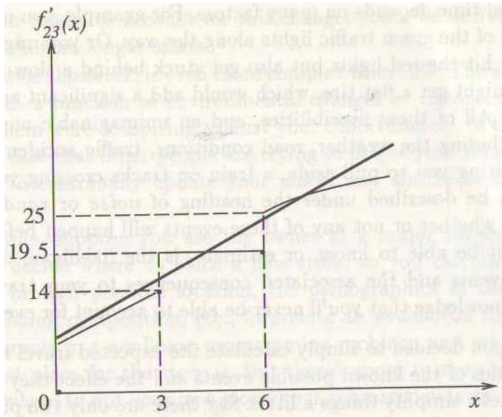
$$\sum_{j=1}^{k} x_{ij} \le sour(i), \text{ for } i = 1, 2, \ldots, n,$$
$$\sum_{i=1}^{n} x_{ij} \ge dest(j), \text{ for } j = 1, 2, \ldots, k,$$
$$x_{ij} \ge 0, \text{ for } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, k,$$

x_{ij} : quantity from source i to destination j

Modeling the Problem

- Transportation Problem
  - Approximation

$$f'_{23}(x) = 2.66x + 8.25$$



☞ LP

- Two approaches

1) Problem → $Model_a$ → $Solution_p(Model_a)$
2) Problem → $Model_p$ → $Solution_a(Model_p)$

   a: approximated, *p*: precise

– 일반적으로 2) 가 바람직한 접근 방법

Basic Concepts

Search Problem

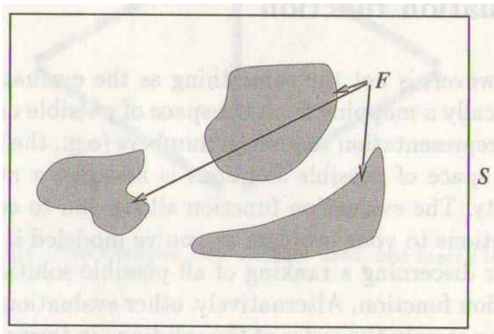
- Search Problem (Optimization Problem)

  Given a search space S and its feasible part $F \subseteq S$,
  find $x \in F$ such that

  $$eval(x) \leq eval(y)$$

  for all $y \in F$.

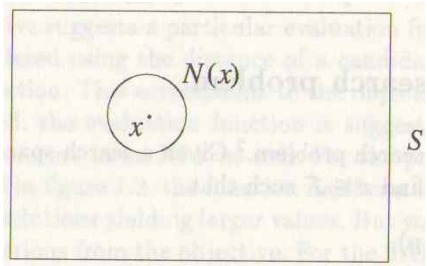
  ☞ x : global solution



Neighborhoods and Local Optima

- Neighborhood $N(x)$

  Set of all points of the search space S that are **close** in some **measurable sense** to the given point x



(ex) NLP

$$N(x) = \{y \in S : dist\ (x, y) \le \epsilon\}$$

$dist\ (x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ (Euclidean distance)

Neighborhoods and Local Optima

- Neighborhood $N(x)$

  (ex) TSP : 2-swap mapping

  x: 3-2-4-1-5

  $N(x)$: **2-3**-4-1-5

  4-2-**3**-1-5

  1-2-4-**3**-5

  5-2-4-1-**3**

  3-**4**-**2**-1-5

  3-**1**-4-**2**-5

  3-**5**-4-1-**2**

  3-2-**1**-**4**-5

  3-2-**5**-1-**4**

  3-2-4-**5**-**1**

  n city TSP: $\frac{n(n-1)}{2}$ neighbors

Neighborhoods and Local Optima

- Neighborhood $N(x)$

  (ex) SAT : 1-flip mapping

  x: 01101101

  $N(x)$: **1**1101101

  0**0**101101

  01**0**01101

  011**1**1101

  0110**0**101

  01101**0**01

  011011**1**1

  0110110**0**

Neighborhoods and Local Optima

- Local optima

$$eval\ (x) \le eval\ (y)$$

for all $y \in N(x)$

☞ x : local solution (local optimum)

- Local search strategy

Locate solutions within a neighborhood of the current point that have better evaluations

(ex) minimize $f(x) = x^2$

- Size of the neighborhood vs. efficiency of the search
  - Smaller size of neighborhood
    → quick search, local optimum
  - Larger size of neighborhood
    → better decisions, huge computation

Hill-Climbing Methods

- Hill-climbing procedure
  - Start from a single point (current point)
  - Improve the current point by searching the neighborhood
  - Terminates if no further improvement

```
procedure iterated hill-climber
begin
  t ← 0
  initialize best
  repeat
    local ← FALSE
    select a current point v_c at random
    evaluate v_c
    repeat
      select all new points in the neighborhood of v_c
      select the point v_n from the set of new points
        with the best value of evaluation function eval
      if eval(v_n) is better than eval(v_c)
        then v_c ← v_n
        else local ← TRUE
    until local
    t ← t + 1
    if v_c is better than best
      then best ← v_c
  until t = MAX
end
```

t: number of iterations
local : 해의 개선 여부
best : 최적해
$\mathbf{v}_c$: current point
$\mathbf{v}_n$: new point

Hill-Climbing Methods

- Weakness
  - Terminates at solutions that are only locally optimal
  - Depends on the initial configuration
  - Not possible to provide an upper bound for computation time