General formulation:

$$\min_{x\in\mathcal{X}} \ \boxed{f(x)} \quad \text{objective function}$$
$$s.t.\ \ \boxed{g_i(x)=0, \quad 1\le i\le q;} \quad \text{equality constraints}$$
$$\boxed{h_i(x)\le 0, \quad q+1\le i\le m} \quad \text{inequality constraints}$$

A solution is **(in)feasible** if it does (not) satisfy the constraints

The goal: find a feasible solution minimizing the objective f

目的：找到满足约束的可行解

介绍经典的处理约束的方式，还对设计算法有帮助，从理论指导算法设计

一、常见约束处理策略

1、penalty function

> 把约束转化没

$$\begin{aligned} &\min \ f(x) \\ &s.t. \ \ g_i(x)=0, \quad 1\le i\le q; \\ &\qquad h_i(x)\le 0, \quad q+1\le i\le m \end{aligned}$$

$$\min \ f(x)+\sum_{i=1}^{m}\lambda_i\cdot \boxed{f_i(x)}$$

the i-th constraint violation degree $$f_i(x)=\begin{cases} |g_i(x)| & 1\le i\le q \\ \max\{0,h_i(x)\} & q+1\le i\le m \end{cases}$$

要确保转后的最优解不变

应用示例--MST

$$\arg\min_{x\in\{0,1\}^m}\sum_{i=1}^{m} w_i x_i \quad s.t.\ c(\boldsymbol{x})=1$$

Fitness function:

Constraint violation degree

$n^2\cdot w_{max}$

Original objective function

$$\min \ (c(\boldsymbol{x})-1)\cdot w_{ub}+\sum\nolimits_{i:x_i=1} w_i$$

2、Stochastic ranking

给罚函数引入随机性

> φ(x)罚函数penalty：整体约束违反程度
>
> λ因为很难设，所以随机来设

λ小时取决定作用的是f，否则是φ

Consider two solutions x_1 and x_2 satisfying

$$f(x_1) < f(x_2) \text{ and } \phi(x_1) > \phi(x_2)$$

- λ is small: the comparison is based on the objective function

$$\lambda < \frac{f(x_2) - f(x_1)}{\phi(x_1) - \phi(x_2)} \Rightarrow f(x_1) + \lambda\phi(x_1) < f(x_2) + \lambda\phi(x_2)$$

- λ is large: the comparison is based on the penalty function

$$\lambda > \frac{f(x_2) - f(x_1)}{\phi(x_1) - \phi(x_2)} \Rightarrow f(x_1) + \lambda\phi(x_1) > f(x_2) + \lambda\phi(x_2)$$

The value of λ determines whether the comparison is based on the objective function or the penalty function

冒泡排序

3、Repair functions

repair infeasible solutions to feasible

4、Restricting search to the feasible region

把搜索限定到可行区域里，生成新的可行解

5、Decoder functions

和repair类似，以解码方式进行修复

区别：repair不做变换，遇到不可行的就修复成可行的，只在子空间上搜索

decode解码后一定是可行解，可以得到的种群更完善

二、约束转化成二目标问题

> Bi-objective reformulation

(1+1)-EA for MST

GSEMO: Given a pseudo-Boolean function vector $\boldsymbol{f}$:

$\boldsymbol{x} :=$ randomly selected from $\{0,1\}^n$.
$P := \{\boldsymbol{x}\}$.
Repeat until some termination criterion is met
  Choose $\boldsymbol{x}$ from P uniformly at random.
  $\boldsymbol{x}' :=$ flip each bit of $\boldsymbol{x}$ with probability $1/n$.
  if $\nexists\, \boldsymbol{z} \in P$ such that $\boldsymbol{z} \prec \boldsymbol{x}'$
    $P := (P - \{\boldsymbol{z} \in P \mid \boldsymbol{x}' \preccurlyeq \boldsymbol{z}\}) \cup \{\boldsymbol{x}'\}$.

随机抽一个解，变异看看有没有更好

MST by MOEAs

$$\arg\min_{x \in \{0,1\}^m} \sum_{i=1}^{m} w_i x_i \quad s.t. \quad c(\boldsymbol{x}) = 1$$

$$\Downarrow$$

Bi-objective reformulation $\quad min \quad (c(\boldsymbol{x}), \sum_{i:x_i=1} w_i)$

Theorem. [Neumann & Wegener, Nature Computing'05] The expected running time of the GSEMO solving the MST problem is $O(mn\,(n + \log w_{max}))$.

Penalty functions: $O(m^2(\log n + \log w_{max}))$

Bi-objective reformulation: $O(mn(n + \log w_{max}))$

Bi-objective reformulation is better for dense graphs, e.g., $m \in \Theta(n^2)$

proof

Problem	Penalty functions	Bi-objective reformulation
Set cover	*exponential*	$O(mn(\log c_{max} + \log n))$ [Friedrich et al., ECJ'10]
Minimum cut	*exponential*	$O(Fm(\log c_{max} + \log n))$ [Neumann et al., Algorithmica'11]
Minimum label spanning tree	$\Omega(ku^k)$	$O(k^2 \log k)$ [Lai et al., TEC'14]
Minimum cost coverage	*exponential*	$O(Nn(\log n + \log w_{max} + N))$ [Qian et al., IJCAI'15]

Better

对比其他np-hard问题，也发现二目标转化后比罚函数法时间节省不少

转化例子

1. transform the original constrained optimization problem into a bi-objective optimization problem

constrained

$$\min \ f(x)$$
$$s.t. \ \ g_i(x) = 0, \quad 1 \le i \le q;$$
$$h_i(x) \le 0, \quad q+1 \le i \le m$$

bi-objective

$$\min \ \left(f(x), \ \sum_{i=1}^{m} \boxed{f_i(x)}\right)$$

constraint violation degree $f_i(x) = \begin{cases} |g_i(x)| & 1 \le i \le q \\ \max\{0, h_i(x)\} & q+1 \le i \le m \end{cases}$

2. employ a multi-objective EA to solve the transformed problem

constraint violation degree = 0

3. output the feasible solution from the generated non-dominated solution set

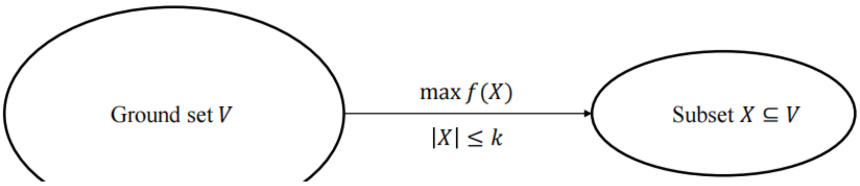
三、应用--子集选择

> 应用很广

Subset selection is to select a subset of size k from a total set of n items for optimizing some objective function

Formally stated: given all items $V = \{v_1, \ldots, v_n\}$, an objective function $f: 2^V \to \mathrm{R}$ and a budget k, to find a subset $X \subseteq V$ such that

$$max_{X \subseteq V} \quad f(X) \quad s.t. \quad |X| \le k.$$



- 挑的准则，不一定是啥：f
- 挑的个数上限：k

sparse regression

目标：预测MSE小

影响力最大化Influence maximization

每条边表示一个用户对另一个用户的影响力

文本摘要Document summarization

传感器放置Sensor placement

在可选位置中选可监测信息量最大的组合

四、POSS算法

先转化为2目标转化（不完全等价）

POSS algorithm [Qian, Yu and Zhou, NIPS'15]

Transformation:

$$max_{X \subseteq V} \ f(X) \quad s.t. \quad |X| \le k \qquad \text{original}$$

$$\Downarrow$$

$$min_{X \subseteq V} \ (-f(X), |X|) \qquad \text{bi-objective}$$

Algorithm 1 POSS

Input: all variables $V = \{X_1, \ldots, X_n\}$, a given objective f and an integer parameter $k \in [1, n]$
Parameter: the number of iterations T
Output: a subset of V with at most k variables
Process:

Let $s = \{0\}^n$ and $P = \{s\}$.
Let $t = 0$.
while $t < T$ **do**
  Select s from P uniformly at random.
  Generate s' by flipping each bit of s with prob. $\frac{1}{n}$.
  Evaluate $f_1(s')$ and $f_2(s')$.
  if $\nexists z \in P$ such that $z \prec s'$ **then**
    $Q = \{z \in P \mid s' \preceq z\}$.
    $P = (P \setminus Q) \cup \{s'\}$.
  end if
  $t = t + 1$.
end while
return $\arg\min_{s \in P, |s| \le k} f_1(s)$

Initialization: put the special solution $\{0\}^n$ into the population P

Parent selection & Reproduction: pick a solution $\boldsymbol{x}$ randomly from P, and flip each bit of $\boldsymbol{x}$ with prob. $1/n$ to generate a new solution

Evaluation & Survivor selection: if the new solution is not dominated, put it into P and weed out bad solutions

Output: select the best feasible solution

把全0放到种群中

每轮迭代随机从种群选一个，变异一下

理论保证：没见过的数据集也不会太坏

Theorem 1. For subset selection with monotone objective functions, POSS using $E[T] \le 2ek^2n$ finds a solution X with $|X| \le k$ and

$$f(X) \ge (1 - e^{-\gamma}) \cdot \mathrm{OPT}.$$

已被证实这是最好的界