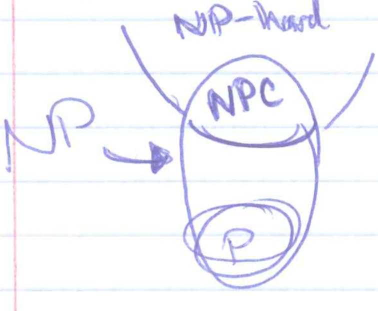
Ch 13: ILP.

Satisfiability (and Complexity [Ch. 15].)



Reductions

~~NP~~ Def: P, NP, NP-complete

~~NP Complete P.~~

Linear Programming is in P!

NP-Complete Problem: 3-CNF-SAT

~~3~~ Conjunctive Normal Form: $\bigwedge_{i=1}^{m} C_j(x)$

where each $C_j(x) = \bigvee(\hat{x})$ clause.

Example:

$$\varphi(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee \bar{x}_4) \wedge (\bar{x}_2 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4)$$

$$\varphi(1,0,1,0) = (1 \vee 0 \vee 1) \wedge (1 \vee 1 \vee 0) \wedge (0 \vee 0 \vee 0)$$

$$= 1 \wedge 1 \wedge 0 = \underline{\underline{0}}.$$

Satisfiability: Does there exist a vector $\underline{x} \in \{0,1\}^n$ ~~that sat~~ s.t. $\varphi(\underline{x}) = 1$.
($\underline{x}$ satisfies φ).

* ~~Linear~~

Integer Programming is NP-Hard

(Feasibility of Bounded ILP is NP-complete.)

* Feas. in NP by selecting an integer point and solving the resulting LP for feasibility.

ILP is NP-Hard: Reduce CNF-SAT to ILP.

Given $\varphi, (x_1, \ldots, x_n)$, make the following ILP:

Variables $x_1, \ldots, x_n \in \{0,1\}$.

For each clause, C_j, we make the following sum:

$$\sum_{i : x_i \in C_j} x_i + \sum_{i : \bar{x}_i \in C_j} (1 - x_i) = |C_j|$$

~~If~~ A feasible point $\underline{x}$ ~~exists~~, $\Leftrightarrow \varphi(\underline{x}) = 1$.

Traveling Salesman on $\underbrace{n+1 \text{ vertices}}_{\{0,\dots,n\}}$

$$\min z = \sum_{\substack{i,j=0 \\ i \neq j}}^{n} c_{i,j}\, x_{i,j}$$

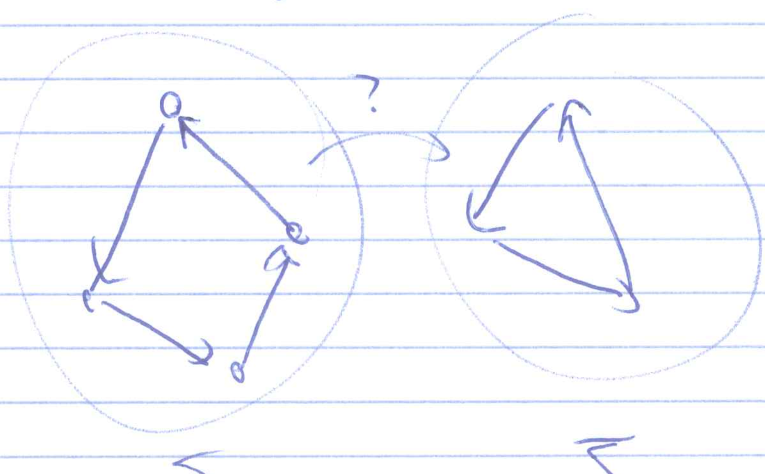
$$\text{s.t.} \quad \left.\begin{array}{ll} 0 \leq x_{i,j} \leq 1 & \forall i,j \\ x_{i,j} & \text{integer} \end{array}\right\} \text{ 0,1-ILP}$$

$$\text{(a)} \quad \sum_{i=1}^{n} x_{i,j} = 1 \quad \forall j$$

$$\text{(b)} \quad \sum_{i=1}^{n} x_{j,i} = 1 \quad \forall j$$

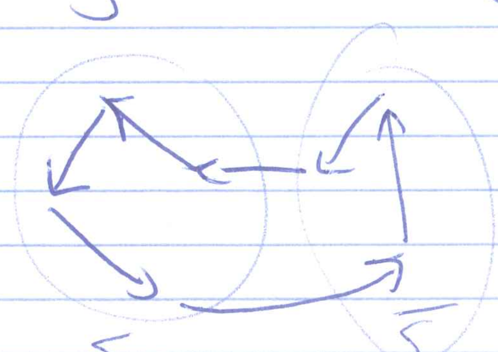
$$\text{(c)} \quad \sum_{\substack{i \in S \\ j \notin S}} x_{i,j} \geq 1 \quad \forall S \subsetneq [n],\ S \neq \emptyset.$$

$2^n - 2$ such sets!



S $\quad$ $\bar{S}$

(Iterative Constraints?)



S $\quad$ $\bar{S}$

← Better!

A.W. Tucker replaces (c) w/

(c'): $u_i - u_j + n x_{ij} \leq n-1 \quad \forall\ 1 \leq i \neq j \leq n$

important! 0 is possible!

where $u_i \in \mathbb{R}$ for each $i \in [n]$. (MILP)

Prop: Constraints (a), (b) & (c') define the TSP.

Pf: **First: Every Solution is a tour.**

(By symmetry) We only need to show that every cycle includes 0.

Suppose $i_1, \ldots, i_k$ is a cycle. $i_j \neq 0$.

$u_{i_j} - u_{i_{j+1}} + n \leq n-1 \quad j = 1, \ldots, k-1$

$u_{i_k} - u_{i_1} + n \leq n-1.$

Add all rows to find

$$kn \leq k(n-1). \quad ↯$$

Second: If T is a tour, there is a feasible point.

Let $i_0, \ldots, i_n$ be a tour. w/ $i_0 = 0$.

Let $u_{i_j} = j$ for $j \in \{0, \ldots, n\}$.

Now: $u_{i_j} - u_{i_{j+1}} + n = j - (j+1) + n = n-1 \leq n-1.$

~~Also $u_{i_t} - u_{i_0} + n = (n-1) - 0$~~

But $u_{i_n} - u_{i_0} + n$ is excluded! $\square$

Scheduling Problems

Multi-core Processors.

A list of computation requests come in, each has an approximate time to compute, some require others to be done before they start!

$\mathcal{J} = \{J_1, \dots, J_n\} \leftarrow n$ jobs / tasks.

J_i requires processing time $\tau_i \geq 0$.

J_i requires R_{ji} of resource j.

There is B_j amount of resource j available at any given time.

The job J_i has a deadline d_i

There are m processors, $a_{i,j} \in \{0,1\}$ indicates

$$a_{ij} = \begin{cases} 1 & \text{the } i^{th} \text{ job can run on the } j^{th} \text{ processor} \\ 0 & \text{otherwise} \end{cases}$$

Precedence Relation: $(\mathcal{J}, A)$, a digraph w/ no cycles.

$$T = \sum_{i=1}^{n} \tau_i$$

Formulation as MILP:

Let $t, s_i \in \mathbb{R}_{\geq 0}$, and $p_{i,j} \in \{0,1\}$ be variables.

s_i = starting time of job J_i.

$p_{i,j} = \mathbb{1}[J_i \text{ runs on processor } P_j]$

Optimize: min t = time to completion.

Constraints:

① $p_{i,j} \leq a_{i,j}$ (don't run on incompatible processors)

② (Let $\delta_{i,k} = \mathbb{1}[s_i < s_k] \in \{0,1\}$)

enforces property of $\delta_{i,k}$
$$\begin{cases} s_k - s_i \leq \delta_{i,k} \cdot T & (\forall i \neq k) \\ \delta_{i,k} + \delta_{k,i} = 1 & (\text{exactly one has value } 1.) \end{cases}$$

If $\delta_{i,k} = 1$ and $p_{i,j} = p_{k,j} = 1$, then $s_i + \tau_i \leq s_k$.

$$\{ s_i + \tau_i - s_k \leq T \cdot (1 - \delta_{i,k} + 2 - p_{i,j} - p_{k,j})$$

③ If $(J_i, J_k) \in A$, then J_k starts after J_i.

$$\left.\begin{array}{l} \delta_{i,k} = 1. \\ s_i + \tau_i - s_k \leq 0 \end{array}\right\}$$

④ Resource Constraints.

At a given moment, a resource is not overused.

Let $\varepsilon_{i,k} \geq 0$ be integers.

$$\varepsilon_{i,k} = \mathbb{1}\left[J_i \text{ is } \underline{\text{active}} \text{ when } J_k \text{ starts} \right]$$

$$\begin{cases} \varepsilon_{i,k} \leq \delta_{i,k} \\ 1 + (\varepsilon_{i,k} - 1)T \leq s_i + \tau_i - s_k \leq (\varepsilon_{i,k} + \delta_{i,k})T. \end{cases}$$

Resource Constraints:

$$R_{j,k} + \sum_{i \neq k} (\varepsilon_{i,k} + \varepsilon_{k,i}) R_{j,i} \leq B_j$$

⑤ Deadline Constraints

$$s_i + \tau_i \leq d_i$$

⑥ Total Finishing Time

$$\cancel{t} \; s_i + \tau_i \leq t.$$

Strength of ILP

Disjunction: $x \geq a$ or $y \geq b$ — constants

Make new 0/1-variable δ

$$\left.\begin{aligned} x &\geq \delta \cdot a \\ y &\geq (1-\delta)\cdot b. \end{aligned}\right\} \text{ exactly one will be enforced.}$$

Implication: If $x > a$ then $y \geq b$.

Same as $x \leq a$ or $y \geq b$.

(Strict inequalities are tricky.)

~~Discrete Variables~~: $x \in S = \{s_1, \dots, s_k\} \subseteq \mathbb{R}$.

$$\varepsilon_i = \mathbb{1}[x = s_i].$$

$$\sum_{i=1}^{k} \varepsilon_i = 1.$$

$$x = \sum_{i=1}^{k} s_i \varepsilon_i.$$