1 Project scheduling in the boatyard industry

- t_i time at which operation i finishes
- z the end of everithing
- $z \geq t_i \forall i$
- $t_A \geq d_A$
- $t_B \geq t_A + d_B$
- $t_E \geq t_B + d_B$ this constraint should be here only if operation B is executed. You introduce new variable $y_B = 1$ if B is executed 0 otherwise. $t_E \geq t_B + d_E - M(1 - y_B)$ if B is not executed the constraint becomes redundant.
- $t_E \geq t_C + d_E$

2 A (shift) covering problem

(Real problem) i = index of pharmacy k = index of shift We introduce the variable $y_{i,k} = 1$ if pharmacy i si open in shift k, 0 otherwise $i \in \mathcal{P}$ $k = 1 \dots K$ $j \in C$ set of clients. d_{jk} distance between j and the shift k (the closest pharmacy opened in shift k), it is a variable this means that we have to find the d_{jk}

- $min \sum_{j \in C} \sum_{k=1}^{K} d_{jk}$

It is a common tecnhique to decompose the problem in smaller one. In this case we have to determine d_{jk}

- $d_{jk} = y_{ik}$

we have a relation that relates $x_{jik} = 1$ if j goes to i in k, 0 otherwise

$d_{jk} = \sum_{i \in \mathcal{P}} D_{ji} x_{jik}$, where D_{ji} is the constant, just one of the xs will be equal to one

$\sum_{i \in P} x_{jik} = 1 \forall i \in C, \forall k = 1 \dots K$ (constraint, it says that the client must go to just one pharmacy forall shifts. $x_{ijk} \leq y_{ik}, \forall i, j, k$ With this 2 constraints we say that the client must go only to one of the opened pharamacy.

We have to add the fact that the pharmacy must not be *always* opened, each pharamcy should be in one shift.

$\sum_{k=1} K y_{ik} = 1 \forall i \in P$

The domains are

$$y_{ik} \in \{0, 1\}$$
$$x_{jik} \in \{0, 1\}$$
$$d_{jk} \in \mathbb{R}$$

We want to balance

$$\left\lceil \frac{|P|}{K} \right\rceil \leq \sum_{i \in P} y_{ik} \leq \frac{|P|}{K} \forall k$$

2.1 A comment

A problem with this model is that there are a lot of symmetric solution due to the fact that we modelled the problem in this way (using names, since the index are different you can have). The problem is the index k because it is an artificial method -¿ we have only to determine subsets.

2.2 Alternative

$P \rightsquigarrow 2^P$ we have to determine the subsets of P that are used in a shift we determine the subset of pharmacies. We imagine a shift

$$x_J \forall J \in 2^P$$

$$x_J = 1 \text{ if subset} J \subset Pisashift0otherwise$$

here we have no symmetries. We have to minimize

$$min \sum_{j \in 2^P} \sum_{j \in C} D_{jJ} x_J$$

in this case D is not a variable, it is a number, because all the subsets have a minimal distance. We can preprocess it. We must guarantee

$$\sum_{J \in 2^P} x_J = K$$

We have to say that:

$$\sum_{J \in 2^P} A_{iJ} x_J = 1 \forall i \in P$$

A_{iJ} is a constant/number that takes value 1 if $i \in J$ 0 otherwise.

This model is equivalent solution to the first model. This model has no symmetry is very flexible, BUT The problem with this model is that it is very difficult (big data) to implement you have to create an exponential number of parameters/variables The powerset is $2^{|P|}$. But we will see methods to manage an exponential number of variables.

3 27/10

4 3/11

Heuristics methods. There are two categories:

- Exact Methods. Designed to obtain provably optimal solution
- Heuristics methods: provides "good" solution with no optimality guarantee.

Before applying heuristics methods you have at least to scatch an exact approach (Example:Cisco and shortest path problem). Heuristic methods are defined by some people: quick and dirty. Quick is a very important characteristics. Heuristic means find/search a solution. You have to apply it when the problem is very difficult or when you have a very short time to solve it.

Sometimes you cannot use heuristics, because the nature of the problem, e.g. an optimal solution is mandatory, because it is in the requirement of the problem

4.1 Heuristic Methods

Heuristic methods are a broad class of methods. There is no standard classification, but one possible is:

- Specific Heuristic:
- General Heuristic:
- Constructive Heuristics: Greedy algorithms, you have to choose the element that at is moment looks (Miope)

4.2 10/11

You can have at most :

$$n(n-p) \leq (\frac{n^2}{2})$$

swap solutions