

# LINFO2266 - Solutions

November 9, 2021

## 1 Lagrangian Relaxation

### 1.1 Set Covering Problem

1. We introduce one Lagrangian multiplier for each constraint to obtain the Lagrangian relaxation:

$$\min \sum_{i=1}^m c_i x_i + \sum_{j=1}^n \lambda_j \left( 1 - \sum_{\substack{i=1 \\ j \in S_i}}^m x_i \right) \quad (1)$$

with  $\lambda_j \geq 0, j \in \{1, \dots, n\}$ . Note how we integrated the constraints: for any *feasible* solution of the original problem, we will have

$$1 - \sum_{\substack{i=1 \\ j \in S_i}}^m x_i \leq 0 \quad (2)$$

which means a *negative* value will be added to the objective function and that we will thus obtain a *lower bound* on the optimal value of the problem.

For clarity, we rewrite eq. (1) as:

$$\min \sum_{i=1}^m \left( c_i - \sum_{j \in S_i} \lambda_j \right) x_i + \sum_{j=1}^n \lambda_j \quad (3)$$

For fixed values of the Lagrangian multipliers  $\lambda_j$ , we can obtain a lower bound by solving eq. (3). This problem is very simple to solve if we notice that:

- each  $x_i$  is multiplied by a fixed coefficient  $\left( c_i - \sum_{j \in S_i} \lambda_j \right)$
- the second term  $\left( \sum_{j=1}^n \lambda_j \right)$  is constant

In order to minimize the objective function, it is sufficient to select the sets which have a negative coefficient:

$$x_i = \begin{cases} 1 & \text{if } c_i - \sum_{j \in S_i} \lambda_j \leq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

2. By applying exactly what we described above, we get a lower bound value of 4.2.
3. We adapt the subgradient procedure covered in the lectures for this particular problem, and for multiple Lagrange multipliers in algorithm 1.

---

**Algorithm 1** Subgradient procedure for the Set Covering Problem.

---

```

1:  $\mathcal{L}^* \leftarrow -\infty, k \leftarrow 1, \mu_0 \leftarrow 1$ 
2:  $\lambda_{0,j} \leftarrow 0, 1 \leq j \leq n$ 
3:  $\mathcal{C}^* \leftarrow$  a trivial solution, or none
4: while  $\mu_k \geq \epsilon$  do
5:   Compute cover  $\mathcal{C}_k$  with weights  $\left(c_i - \sum_{j \in S_i} \lambda_{k,j}\right)$  for  $1 \leq i \leq m$ 
6:    $\mathcal{L}_k \leftarrow \text{value}(\mathcal{C}_k) + \sum_{j=1}^n \lambda_{k,j}$ 
7:   if  $\mathcal{L}_k > \mathcal{L}^*$  then
8:      $\mathcal{L}^* \leftarrow \mathcal{L}_k$ 
9:   end if
10:  if  $\text{value}(\mathcal{C}_k) < \text{value}(\mathcal{C}^*)$  then
11:     $\mathcal{C}^* \leftarrow \mathcal{C}_k$ 
12:  end if
13:  if  $\mathcal{L}^* = \text{value}(\mathcal{C}^*)$  then
14:    break
15:  end if
16:   $\lambda_{k+1,j} \leftarrow \max\left(0, \lambda_{k,j} + \mu_k \left(1 - \sum_{\substack{i=1 \\ j \in S_i}}^m x_i\right)\right)$  for  $1 \leq j \leq n$ 
17:   $\mu_{k+1} \leftarrow \frac{1}{k}$  (or another valid update rule)
18:   $k \leftarrow k + 1$ 
19: end while
20: return  $lb$ 

```

---