TSP:

1) a) Folosim metoda reducerii la absurd. Pp. că problema nu este în NP-Hard. Fie $G_1=(V_1,E_1)$ și $G_2=(V_2,E_2)$. Muchiile au costul 1 dacă $\in E_1$ sau 2 dacă $\notin E_1$ $\Rightarrow$ în G_2 muchiile au costul 1 sau 2, iar după TSP $\Rightarrow$ costul total minim $= N$ doar dacă G_1 este hamiltonian. $\Rightarrow$ trb. să aflăm dacă există un ciclu hamiltonian în G_2, dar alg. de găsire a unui ciclu hamiltonian este în NP-Hard, iar $NP \neq P \Rightarrow$ (≪↯) $\Rightarrow$ problema este în NP-Hard

b) Pentru a arăta că aceste ponderi satisfac în continuare inegalitatea triunghiului, vom analiza toate cazurile:

I $\left.\begin{array}{l} a=1 \\ b=1 \\ c=1 \end{array}\right] \Rightarrow 1+1 \geq 1$ (A)

II $\left.\begin{array}{l} a=1 \\ b=1 \\ c=2 \end{array}\right] \Rightarrow 1+1 \geq 2$ (A)

III $\left.\begin{array}{l} a=1 \\ b=2 \\ c=2 \end{array}\right] \Rightarrow 1+2 \geq 2$ (A)

IV $\left.\begin{array}{l} a=2 \\ b=2 \\ c=2 \end{array}\right] \Rightarrow 2+2 \geq 2$ (A)

c) Fie K_n = graful complet cu n noduri, toate muchiile au costul 1 $\Rightarrow$ $\Rightarrow$ din TSP va rezulta costul total $= N$.

Un arbore parțial de cost minim ce începe cu nodul 1 va conține o muchie către fiecare nod, adică n-1 muchii.

Alg. din curs parcurge APM-ul descris mai sus de 2 ori, adică toate muchiile de $2*(n-1)$ ori. $\Rightarrow$ nu poate fi $\frac{3}{2}$ aprox. pt. că $2*(n-1) > \frac{3n}{2}$, $\forall n > 4$;

Load Balance

3) Găsim un alt lower bound

$L(M_i)$ = load-ul mașinii i după asignarea ~~celor~~ primelor j-1 activ.,

j = ultimul job asignat ~~m~~ mașinii x

x = mașina cu load-ul cel mai mare

$$L(M_x) \leq \frac{m+1}{2m} * \sum_{i=1}^{m} L(M_i) = \frac{m+1}{2m} * \sum_{p=1}^{j} t_p < \frac{m+1}{2m} * \sum_{p=1}^{m} t_p \leq LB \text{ (lower bound nou)}$$

$$ALG = L(M_x) + t_j \leq t_j + LB \leq \max\{t_p \,/\, 1 \leq p \leq n\} + LB \leq LB + LB =$$
$$= 2\,LB \leq 2\,OPT$$

$$L(M_x) \le \frac{m+1}{2m} \sum_{i=1}^{m} L(M_i) = \frac{m+1}{2m} \cdot \sum_{p=1}^{j} t_p \le \frac{m+1}{2m} \left(\sum_{p=1}^{n} t_p - t_j \right) \le$$

$$\le \frac{m+1}{2m} \sum_{p=1}^{n} t_p - \frac{m+1}{2m} t_j$$

$$ALG = L(M_x) + t_j \le \frac{m+1}{2m} \sum_{p=1}^{n} t_p - \frac{m+1}{2m} t_j + t_j \le OPT - \frac{m+1}{2m} t_j + t_j \le$$

$$\le OPT - \frac{m+1}{2m} t_{max} + t_{max} \le OPT + OPT - \frac{m+1}{2m} OPT = 2OPT - \frac{m+1}{2m} OPT =$$

$$= OPT \left(2 - \frac{m+1}{2m}\right) = \frac{3m-1}{2m} OPT = \left(\frac{3m}{2m} - \frac{1}{2m}\right) OPT = \left(\frac{3}{2} - \frac{1}{2m}\right) OPT$$

$\Rightarrow$ alg. poate fi îmbunătățit la $\left(\frac{3}{2} - \frac{1}{2m}\right) OPT$

1) a) Fie setul $\{40, 80, 50, 30\}$

Sol. optimă $\max(\{40, 50\}, \{80, 30\}) = 110$

cu alg dat obținem $\max(\{50, 30\}, \{40, 80\}) = 120$

$110 \cdot 1,1 = 121 > 120 \Rightarrow$ alg. propus poate fi ≥ 1.1 aprox.

Knapsack

1) b) Vom avea 2 cazuri:

I suma $\ge k/2 \Rightarrow$ condiție îndeplinită

II suma $\le k/2 \Rightarrow$ urm: $x_i \ge k - k/2 \Rightarrow x_i \ge k/2 \Rightarrow$ suma $= x_i$

$\Rightarrow$ condiție îndeplinită

Vertex Cover:

a) Pentru a analiza factorul de aproximare (worst case) al algoritmului, vom da următorul exemplu

$$C = (x_1 \vee x_4 \vee x_5) \wedge (x_1 \vee x_3 \vee x_4) \wedge (x_6 \vee x_2 \vee x_1)$$

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

în care x_i selectați ~~de~~ în algoritm sunt cei încercuiți. Worst case ar fi ca elementele selectate dintr-o mulțime să nu mai apară în nicio altă mulțime, ceea ce înseamnă că vom evalua la true toate ~~variabilele~~ elementele din mulțimea X_i și toate predicatele $\Rightarrow$ alg. este n-aproximativ

b) Putem încerca să îmbunătățim algoritmul prin atribuirea fiecărui element dintr-un predicat valoarea true.

1: $C = \{c_1, \dots, c_m\}$, mulțimea de predicate, $X = \{x_1, \dots x_n\}$ - mulțimea de variabile
2: cât timp $C \neq \emptyset$ execută
 3: Alegem aleator $c_j \in C$
 4: Pentru fiecare x_i din c_j
 5: $x_i \leftarrow$ true
 6: Eliminăm din C toate predicatele ce îl conțin pe x_i.
7: return X

- cu ajutorul acestui alg., vom selecta 3 elem. din primul predicat, eliminându-le ~~C=~~ pe restul.

$$C = (x_1 \vee x_4 \vee x_5) \wedge \cancel{(x_1 \vee x_3 \vee x_4)} \wedge \cancel{(x_6 \vee x_2 \vee x_1)}$$

În cel mai rău caz vor fi selectate 3 elemente pt. fiecare pas $\Rightarrow$ alg. este cel puțin 3-aproximativ

c) Exemplu de reformulare:
Fie $X = \{x_1, x_2, \dots x_n\}$, $0 \le x_i \le 1$, $(\forall)\ i \in \{1,..n\}$

$C = \{c_1, \dots, c_m\}$, $\forall\ c_j \in C$ și $x_{j_1}, x_{j_2}, x_{j_3} \in c_j$, $\Rightarrow$

$$\Rightarrow x_{j_1} + x_{j_2} + x_{j_3} \ge 1$$

Cerință: Minimizați $\sum_{i=1}^{n} x_i$.