

CSIE 5452, Fall 2020 — Homework 2

Due October 26 (Monday) at Noon

When you submit your homework on Gradescope, please select the corresponding page(s) of each problem. Points may be deducted if no appropriate intermediate step is provided.

1 MILP Linearization (12pts)

We will prove or make the following propositions are equivalent so that we can transform constraints to linear forms and thus apply the Mixed Integer Linear Programming (MILP). Note that “ \iff ” denotes “equivalence” and “ \wedge ” denotes “logical conjunction” (AND).

- (4pts) Given α, β, γ which are binary variables, prove

$$\alpha + \beta + \gamma \neq 2 \iff \alpha + \beta - \gamma \leq 1 \wedge \alpha - \beta + \gamma \leq 1 \wedge -\alpha + \beta + \gamma \leq 1$$

by filling “T” (True) or “F” (False) in the following table (if LHS=RHS in all cases, then LHS and RHS are equivalent):

α	β	γ	LHS	$\alpha + \beta - \gamma \leq 1$	$\alpha - \beta + \gamma \leq 1$	$-\alpha + \beta + \gamma \leq 1$	RHS	LHS=RHS?
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

- (4pts) Given α, β, γ which are binary variables, prove

$$\alpha\beta = \gamma \iff \alpha + \beta - 1 \leq \gamma \wedge \gamma \leq \alpha \wedge \gamma \leq \beta$$

by filling “T” (True) or “F” (False) in the following table (if LHS=RHS in all cases, then LHS and RHS are equivalent):

α	β	γ	LHS	$\alpha + \beta - 1 \leq \gamma$	$\gamma \leq \alpha$	$\gamma \leq \beta$	RHS	LHS=RHS?
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

3. (4pts) Given β which is a binary variable, x, y which are non-negative real variables, and a constraint $x \leq 2020$, select a value of M to guarantee

$$\beta x = y \iff 0 \leq y \leq x \wedge x - M(1 - \beta) \leq y \wedge y \leq M\beta,$$

where you can refer to the following table:

β	LHS	$0 \leq y \leq x$	$x - M(1 - \beta) \leq y$	$y \leq M\beta$	RHS
0	$0 = y$	$0 \leq y \leq x$	$x - M \leq y$	$y \leq 0$	$x - M \leq y = 0 \leq x$
1	$x = y$	$0 \leq y \leq x$	$x \leq y$	$y \leq M$	$0 \leq y = x \leq M$

2 Signal Packing (12pts)

Bit stuffing does not need to be considered in this problem, *i.e.*, you can assume that the length of a message is the length of its data field plus 44 plus 3. Note that the length of a data field must be 8, 16, 24, ..., or 64 bits, even if the message itself is shorter. Assume that there are 4 Electronic Control Units (ECUs), $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3$, and 4 messages, $\mu_0, \mu_1, \mu_2, \mu_3$, as follows:

Message	Sender	Receiver(s)	Number of Bits	Period (msec)
μ_0	ε_0	ε_1	6	50
μ_1	ε_0	ε_1	10	50
μ_2	ε_1	$\varepsilon_2, \varepsilon_3$	10	50
μ_3	ε_0	ε_3	16	100

A system designer redesigns the messages as follows:

Message	Sender	Receiver(s)	Number of Bits	Period (msec)
μ'_0	ε_0	ε_1	16	50
μ_2	ε_1	$\varepsilon_2, \varepsilon_3$	10	50
μ_3	ε_0	ε_3	16	100

where the first 6 bits of μ'_0 are the bits from μ_0 and the following 10 bits of μ'_0 are the bits from μ_1 .

- (4pts) Regarding the number of bits which need to be transmitted, do you think that the new design is better? Please explain.
- (4pts) Can you further merge μ_2 into μ'_0 ?
- (4pts) In most cases, it does not hurt to have more frequent messages, but it is not allowed to have less frequent messages. Following this policy, can you further improve the number of bits which need to be transmitted? Please explain.

3 Simulated Annealing for Priority Assignment (36pts)

Please download the benchmark "Input.dat" from NTU COOL. In the benchmark, the first number is n , the number of messages. The second number is τ . Each of the following lines contains the priority (P_i), the transmission time (C_i), and the period (T_i) of each message. Now, you are asked to use the Simulated Annealing to decide the priority of each message. The requirements are:

- The objective is to minimize the summation of the worst-case response times of all messages.
- The priority of each message must be an integer in the range $[0, n - 1]$.
- The priority of each message must be unique.
- The worst-case response time of each message must be smaller than or equal to the period of each message.
- The given priorities are the initial solution in the Simulated Annealing.
- We expect the total runtime less than 15 seconds.

You are required to do three things in your submission:

1. You should print out n numbers (one number per line) representing the priorities of those messages. Note that you need to follow the message ordering in the benchmark, *e.g.*, the first number in the list is the priority of the first message in the benchmark.
2. You should print out 1 number representing your objective value (best one during your run). We do not expect an optimal solution in this problem.
3. You should also print out your source codes. We may ask you to provide your source codes which must be the same as those on your printout. If the worst-case response times above are correct but the source codes are clearly wrong implementation, it is regarded as academic dishonesty.