CSIE 5452, Fall 2019 — Homework 2

Due October 28 (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 "$\Longleftrightarrow$" denotes "equivalence" and "$\wedge$" denotes "logical conjunction" (AND).

1. (4pts) Given α, β, γ which are binary variables, prove

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

   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	T	T	T	T	T	T
0	0	1	T	T	T	T	T	T
0	1	0	T	T	T	T	T	T
0	1	1	F	T	T	F	F	T
1	0	0	T	T	T	T	T	T
1	0	1	F	T	F	T	F	T
1	1	0	F	F	T	T	F	T
1	1	1	T	T	T	T	T	T

2. (4pts) Given α, β, γ which are binary variables, prove

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

   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	T	T	T	T	T	T
0	0	1	F	T	F	F	F	T
0	1	0	T	T	T	T	T	T
0	1	1	F	T	F	T	F	T
1	0	0	T	T	T	T	T	T
1	0	1	F	T	T	F	F	T
1	1	0	F	F	T	T	F	T
1	1	1	T	T	T	T	T	T

3. (4pts) Given β which is a binary variable, x, y which are non-negative real variables, and a constraint $x \leq 2019$, select a value of M to guarantee

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

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	$x = y$	$0 \leq y \leq x$	$x \leq y$	$y \leq M$	$0 \leq y = x \leq M$

β==0:
if LHS==RHS==True:
Y==0 -> 0≦X≦2019
X-M ≦0 -> X≦M
if LHS==RHS==False:
Y>0 -> Y≦0 always False
β==1:
if LHS==RHS==True:
X==Y -> X≦M
if LHS==RHS==False:
X!=Y -> Y≦X and X≦Y always have 1 False

Because X≦2019 and X≦M, 2019≦M
M -> 2019

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.

1. (4pts) Regarding the number of bits which need to be transmitted, do you think that the new design is better? Please explain.

Originally, μ0 needs 8+44+3=55 and μ1 needs 16+44+3=63 bits. Therefore, total 55+63=128 bits for 50 msec.
After redesigned, μ0' needs 16+44+3=63 bits for 50 msec. Because of 63 bits < 128 bits and with the same period, the new design is better.

2. (4pts) Can you further merge μ_2 into μ_0'?

The senders are different. They can't be merged.

3. (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.

μ0' and μ2 have different sender, so they can't merge. However, μ0' and μ3 have the same sender, so they can be merged. Before merged, μ0' and μ3 take total 63*2+63=189 bits per 100 period. After merged, it takes total 79 bits per 50 msec. In this case, the period of original μ3 should be altered to 50 msec. In 100 msec, new design has 79*2=158 bits to be transmitted, and original design needs 189 bits. The performance is improved.

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.