

SAPIENZA UNIVERSITY OF ROME

---

Algorithm Design Second Homework

# Alternative Solutions

---

Andrea Fioraldi 1692419

January 15, 2019

## EXERCISE 4

Let  $k$  an ordered multiset of genes  $g \in G$  such that  $g_1 || \dots || g_j || \dots || g_{|k|} = D$  with  $g_j \in k$  ( $||$  is the concatenation operator).  $K$  is the set of all possible  $k$ .

As instance, if  $D = ACCA$  we can have  $K = \{\{AC, CA\}, \{ACC, A\}\}$  if  $G = \{AC, CA, ACC, A\}$ .

Let the variables  $x_i$  with  $i = 1 \dots m$  binary variables that represents if a gene  $G_i \in G$  is in  $D$ .

Let  $y_k \forall k \in K$  a binary variable that tells if the multiset  $k$  is used to form  $D$ .

The ILP formulation is the following:

$$\begin{aligned}
 & \min \sum_{i=1}^m w_i * x_i \\
 (1) \quad & \sum_{k \in K | G_i \in k} y_k - x_i \leq 0 \quad \forall i \in \{1 \dots m\} \\
 (2) \quad & \sum_{k \in K} y_k \geq 1 \\
 & x_i \in \{0, 1\} \quad \forall i \in \{1 \dots m\} \\
 & y_k \in \{0, 1\} \quad \forall k \in K
 \end{aligned}$$

(1) means that if a gene  $G_i$  is not taken all the  $k$  that contains  $G_i$  must be excluded. (2) means that at least one  $k$  must be taken, and it is our solution that has minimum cost genes.

Consider now the LP-relaxion with  $x_i \geq 0 \wedge y_k \geq 0$ . Let's compute the dual of it. In the dual, we have that the number of variables is the same of the number of constraints in the primal ( $m + 1$ ) and vice-versa the number of constraints is the number of variables in the primal ( $m + |K|$ ). Only constraint (2) contribute to the objective function.

When  $G_i$  is used the index  $i$  is always relative to  $G$  and when we use  $G_i \in k$  we are considering all the distinct genes in  $k$ .

$$\begin{aligned}
 & \max u \\
 & \sum_{G_i \in k} v_i - u \geq 0 \quad \forall k \in K \\
 & v_i \leq w_i \quad \forall i \in \{1 \dots m\} \\
 & u \geq 0, v_i \geq 0 \quad \forall i \in \{1 \dots m\}
 \end{aligned}$$

## EXERCISE 5

The proposed problem is a finite zero-sum game theory problem.

Comet, Dasher	Head	Tail
Head	4, -4	-1, 1
Tail	-2, 2	2, -2

We create a matrix  $A_{2,2}$  associated to the previous table taking the first value of each table entry so that each entry of the matrix is the payoff for Comet and the cost for Dasher. We encode payoff/cost and mixed strategy as variables in a LP problem for each player:

Comet	Dasher
$\max y$	$\min v$
$4 * x_1 - 2 * x_2 \geq y$	$4 * u_1 - u_2 \leq v$
$-x_1 + 2 * x_2 \geq y$	$-2 * u_1 + 2 * u_2 \leq v$
$x_1 + x_2 = 1$	$u_1 + u_2 = 1$
$x_1 \geq 0, x_2 \geq 0$	$u_1 \geq 0, u_2 \geq 0$

Comet wants to maximize the payoff choosing the mixed strategy  $x$  in a way that no matter what Dasher plays. Dasher wants to minimize the cost in the same way. Each problem is the dual of the other so  $\max y = \min v = k$ . This implies that this is a Mixed Nash Equilibrium with  $k$  that is the value of the zero-sum game.

Solving the two problems with a solver (see the Appendix to know how) we have that  $\max y = \min v = \frac{2}{3}$ ,  $x_1 = \frac{4}{9}$ ,  $x_2 = \frac{5}{9}$ ,  $y_1 = \frac{1}{3}$ ,  $x_2 = \frac{2}{3}$ . Decoding the index of the variables to the strategies the result is that Santa will bias the coins with this probabilities:

$$\begin{array}{ll} Pr[\text{Comet coin} = \text{Head}] = \frac{4}{9} & Pr[\text{Comet coin} = \text{Tail}] = \frac{5}{9} \\ Pr[\text{Dasher coin} = \text{Head}] = \frac{1}{3} & Pr[\text{Dasher coin} = \text{Tail}] = \frac{2}{3} \end{array}$$

## APPENDIX

### EXERCISE 5

Yes, AMPL+CPLEX should be used insted of Z3 but we do not like the AMPL syntax and in addition we use Z3 almost every day for SMT/SAT so...

Comet:

```
(declare-fun y () Real)
(declare-fun x2 () Real)
(declare-fun x1 () Real)
(assert (>= (- (* 4.0 x1) (* 2.0 x2)) y))
(assert (>= (+ (- x1) (* 2.0 x2)) y))
(assert (= (+ x1 x2) 1.0))
(assert (>= x1 0.0))
(assert (>= x2 0.0))
(maximize y)
(check-sat)
```

Dasher:

```
(declare-fun v () Real)
(declare-fun y2 () Real)
(declare-fun y1 () Real)
(assert (<= (- (* 4.0 y1) y2) v))
(assert (<= (+ (* (- 2.0) y1) (* 2.0 y2)) v))
(assert (= (+ y1 y2) 1.0))
(assert (>= y1 0.0))
(assert (>= y2 0.0))
(minimize v)
(check-sat)
```