Algorithmes de Couplage - corrigé

(1) v_1 propose à u_2 $\quad A = \{(v_1, u_2)\}$

v_2 propose à u_1 $\quad A = \{(v_1, u_2), (v_2, u_1)\}$

v_3 propose à u_1 $\quad A = \{(v_1, u_2), (v_3, u_1)\}$

v_2 propose à u_2 $\quad A = \{(v_2, u_2), (v_3, u_1)\}$

v_1 propose à u_1 $\quad A = \{(v_1, u_1), (v_2, u_2)\}$

v_3 propose à u_3 $\quad A = \{(v_1, u_1), (v_2, u_2), (v_3, u_3)\}$

(2) on peut utiliser le nb de couples (v, u) t.q. v n'a pas encore proposé à u comme variant strictement décroissant.

(3) A est un couplage. A la fin comme tt le monde sera apparié, il sera parfait.

(4)

Soit (u, v) ; (u', v') dans A

Supposons que $u' >_{v} u$ (*) (ds le cas contraire il n'y a rien à prouver) alors on doit garantir que $v' >_{u'} v$

L'algo garantit d'après (*) que v propose à u' avant de proposer à u et puisqu'à la fin u est apparié avec v c'est que u' s'est apparié avec un elt $x >_{u'} v$ au cours de l'algo (il a rejeté v au moment de sa proposition ou plus tard) et finalement il s'est ultimement apparié avec v' (qui peut tout à fait être x) tel que $v' \geq_{u'} x >_{u'} v$ donc on a bien $v' >_{u'} v$

⑤ le nb de proposit° total~~e~~ est n^2.

⑥ Dans A : (u, v)

Dans A' : (u', v) et (u, v')

On suppose que v préfère A càd que $u >_v u'$.
A' est stable donc si $u >_v u'$, on a forcément $v' >_u v$ ce qui signifie exactement que u préfère A'.

⑦ Soit u qui rejette v alors au cours de l'algo, un sommet v' t.q $v' >_u v$ a proposé à u.
Supposons par l'absurde qu'il existe un couplage parfait stable qui contient (u, v).

On a alors l'algo qui produit un couplage parfait stable

$$A_1 : (u, x) \text{ avec } x \geq_u v' >_u v$$
$$\text{et } (u', v)$$

et si u est réalisable] $A_2 : (\overset{\in U}{u}, \overset{\in V}{v}) \quad (\overset{\in U}{y}, \overset{\in V}{x})$

avec $x >_u v$

donc comme A_2 est stable

$$y >_x u$$

Ainsi, dans l'algo x a proposé à y avant de proposer à u ce qui implique que y a rejeté x dans l'algo et on est essentiellement ramené à un pb analogue à celui qui précède. Pour éviter cela et réellement abouti à une contradict°, on va utiliser un argument de minimalité.

Reprenons donc entièrement la question.
Supposons qu'il existe v rejeté par un élément u qui est réalisable pour v et prenons v et u qui correspondent au premier tel rejet au cours de l'execution de l'algo.

Dans ce cas, $\exists x >_u v$ t.q l'algo renverra A_1 contenant (x, u) et (v, u')

et il existe A_2 stable qui contient (v, u) et (x, y) pour un certain y.

On sait que $x >_u v$ donc par stabilité de A_2

$y >_x u$ et donc x propose à y avant u dans l'algo qui renverra A_1 contenant (x, u) ce qui veut dire que y a rejeté x dans l'algo. Il suffit de justifier que c'est avant que u rejette v pour avoir la contradiction.

Au moment où u rejette v, x lui a déjà proposé donc a forcément déjà été rejeté par y

(8) 1) v proposera à $M(v)$ avant de proposer à $u <_v M(v)$. Puisque $M(v)$ est réalisable, on sait qu'il ne rejettera pas v par la q. 7 et donc v ne proposera jamais à aucun $u <_v M(v)$ et donc ne sera jamais apparié à un tel sommet. On a bien v apparié à u t.q $u \geq_v M(v)$.

2) Si v n'est pas apparié alors il n'a pas encore proposé à $M(v)$ qui ne peut pas le rejeter (Q7) et donc n'a proposé qu'à des $u >_v M(v)$.

9) v est apparié à $u \geq_v M(v)$ d'après 8.1 à la fin de l'algo. De plus ~~en~~ u est réalisable pour v car le couplage renvoyé par l'algo est stable (Q4) donc $u \leq M(v)$ par def de $M(v)$.

<u>Finalement $u = M(v)$</u>.