Complejidad de MKM

Analizemos primero, com siempre, la complejidad del pasobloqueante de MKM. Sean $x_1, ...x_r$ los distintos vertices usados en los PUSH-PULL. Entonces, tenemos que:

$$Compl(MKM - block) = \sum_{i=1}^{r}(Compl(PP(x_i)) + Compl(REMOVER(x_i)))$$

Ahora bien, en cada $Remover(x_i)$ se revisan los vecinos de x_i y se borran esos lados. Por lo tanto, $Compl(REMOVER(x_i) = O(d(x_i))$ y por lo tanto:

$$\sum_{i=1}^{r} Compl(REMOVER(x_i)) = \sum_{i=1}^{r} O(d(x_i)) = O(m) \qquad (1)$$

(la ultima igualdad por el Lema del apreton de manos: $\sum_{x \in V} d(x) = 2m$).

Analizemos entonces la complejidad de PULL-PUSH. Tenemos dos partes ditintas del algoritmo: la parte en que el algoritmo esta mandando flujo por un lado que se satura, y la parte en la cual el algoritmo manda flujo por un lado que no se satura. Describamos con la notacion $()_s$ la parte de saturacion y con $()_{ns}$ la parte de no saturacion. Observemos que la operacion de mandar flujo por un lado, cambiar los desbalanceos, etc, es $O(1)$. Cada vez que hacemos esto con un lado que se satura, el lado se borra del network, por lo tanto no se vuelve a usar. Asi, tenemos que:

$$\sum_{i=1}^{r}(Compl(PP(x_i))_s = O(m) \qquad (2)$$

Nos queda por ver la parte de ns. Ahora bien, en cada vez que hacemos $PULL - PUSH(x_i)$ en cualquier vertice recorremos todos los vertices (que quedan). En este recorrido, en cada vertice se hace A LO SUMO UNA (a los umo dos en el caso del x_i) operación de tipo ns, pues cuando se balancea un vertice, se manda flujo por cada lado de forma tal de ir saturando todos los lados, salvo quizas el ultimo que se uso. Por lo tanto, en cada vertice, la complejidad de la parte ns es $O(1)$. Como se recorren todos los vertices en cada PULL-PUSH, tenemos que

$$Compl(PP(x_i))_{ns} = O(n) \ \forall i$$

y por lo tanto:

$$\sum_{i=1}^{r} Compl(PP(x_i))_{ns} = O(nr) = O(n^2) \qquad (3)$$

(1), (2) y (3) nos dicen que:

$$Compl(MKM - block) = O(m) + O(m) + O(n^2) = O(n^2)$$

Por lo tanto:

$$Compl(MKM) = nO(n^2) = O(n^3)$$