Constrained ILS Reduction - Chang vs Su

Consider the ILS problem,

$$\begin{aligned}
&\min_{x \epsilon X} \|Ax - y\| \\
&= \min_{x \epsilon X} \left\|Rx - Q^T y\right\| \\
&= \min_{x \epsilon X} \|Rx - \hat{y}\|
\end{aligned}$$

Define $G = (A^{-1})^T$ and G_i references the i^{th} column of G. Note that $\forall j \neq i \quad G_i \perp A_j$.

The algorithms preposed by Chang and Su follow the same procedure with the key difference that Su's algorithm does not perform QR factorization on the matrix A and instead uses the the columns of G to acheive the same results. The following will describe both algorithms in a common framework and then show that the values they compute are equivalent at each step. Values with superscript 'c' are from Chang's algorithm and superscript 's' are from Su's

In the first step of Chang's algorithm, for each $i \epsilon 1 \ldots n$ we interchange columns i and n in the matrix R, then return R to upper-triangular form with a series of Givens rotations. We then compute $a_i^c = \arg\min_{x \epsilon X} |R_{n,n} x_n - \hat{y_n}| = \lfloor \hat{y_n}/R_{n,n} \rceil$ and $b_i^c = a_i^c \pm 1$ so that b_i^c is the second closest integer in X to $\hat{y_n}/R_{n,n}$. Finally, we compute $dist_i^c = |R_{n,n} b_i^c - \hat{y_n}|$ which represents the partial residual given when x_n is fixed to b_i^c and column i is chosen to be the n^{th} column in the matrix R. The n^{th} column is chosen to be the one that maximizes $dist_i^c$.

The first step of Su's algorithm is much the same. For each $i \epsilon 1 \ldots n$ we compute $a_i^s = \arg\min_{x \epsilon X} \left|y^T G_i - x\right|$ and $b_i^c = a_i^c \pm 1$ so that b_i^c is the second closest integer in X to $y^T G_i$. Then compute $dist_i^s = \left\| \frac{G_i G_i^T}{G_i^T G_i}(y - A_i b_i^s) \right\|_2$. The n^{th} column is chosen to be the one that maximizes $dist_i^s$.

If a_i^c, b_i^c and $dist_i^c$ are equal to a_i^s, b_i^s and $dist_i^s$, then both algorithms will choose the same column as the n^{th}. It is obvious that a_i^c is just the last element of the real least squares solution rounded to the nearest integer (because that is how we obtained a_i^c). Also, $a_i^s = \lfloor y^T G_i \rceil = \lfloor (A^{-1} y)_i^T \rceil$, since we are assuming that we will swap column i and n, this is also by definition the last element of the real solution. Therefore $a_i^s = a_i^c$. It is obvious that if $a_i^s = a_i^c$, $b_i^s = b_i^c$ since it is computed the same way in both algorithms given a_i.

The following is a proof that $dist_i^c = dist_i^s$:

$$\begin{aligned}
dist_i^s &= \left\| \frac{G_i G_i^T}{G_i^T G_i}(y - A_i b_i) \right\|_2 \\
&= \left\| \frac{G_i G_i^T}{G_i^T G_i} y - \frac{G_i}{G_i^T G_i} b_i \right\|_2 \\
&= \left\| \frac{G_i}{G_i^T G_i}(A^{-1}y)_n - \frac{G_i}{G_i^T G_i} b_i \right\|_2 \\
&= \left\| \frac{G_i}{G_i^T G_i}(\frac{\hat{y_n}}{R_{n,n}} - b_i) \right\|_2 \\
&= \left\| \frac{G_i}{G_i^T G_i} \right\|_2 \left| \frac{\hat{y_n} - R_{n,n} b_i}{R_{n,n}} \right| \\
&= \frac{1}{\|G_i\|_2} \left| \frac{\hat{y_n} - R_{n,n} b_i}{R_{n,n}} \right| \\
&= \frac{1}{\|(R^{-1}Q^{-1})_n^T\|_2} \left| \frac{\hat{y_n} - R_{n,n} b_i}{R_{n,n}} \right| \\
&= |\hat{y_n} - R_{n,n} b_i|
\end{aligned}$$

Next, Chang's algorithm sets $\hat{y}_{1:n-1} = \hat{y}_{1:n-1} - R_{1:n-1,n} a_i$. And then continues to work on the subproblem, $\|R_{1:n-1,1:n-1} x_{1:n-1} - \hat{y}_{1:n-1}\|_2^2$. The effect of this is obvious, it fixes $x_n = a_i$ and continues the algorithm.

Su acheives the same thing by setting $y = (y - A_i a_i) - \frac{G_i G_i^T}{G_i^T G_i}(y - A_i a_i)$. Since $\forall j \neq i \quad G_i \perp A_j$, we are subtracting from y the part that is orthoganol to the remaining columns of A, this moves y onto the affine set defined by $x_n = a_i$. The columns of G are similarly projected $\forall j \neq i \quad G_j = G_j - \frac{G_i G_i^T}{G_i^T G_i} G_j$. Removing the part of the normal that is orthoganol to G_i. The algorithm is then repeated with column i removed from G.