S polynomial transformation for permutation argument

Decomposition

Following the original suggestion $s(X,Y) = X^{-N-1}Y^N s_1(X,Y) - X^N s2(X,Y)$

$$s_1(X,Y) = \sum_{i=1}^{N} u_i'(Y)X^{-i+N+1} + \sum_{i=1}^{N} v_i'(Y)X^{i+N+1} + \sum_{i=1}^{N} w_i'(Y)X^{i+2N+1}$$

$s_2(X,Y)$ is not important for this discussion. $s_1(X,Y)$ is in total a polynomial of degree $3N+1$.

$$u_i'(Y) = \sum_{q=1}^{Q} Y^q u(q,i)$$

and with a similar form for $v'(Y)$ and $w'(Y)$

$u(q,i)$ by itself is a constant in q-th linear constraint in front of a variable $a(i)$. $v(q,i)$ and $w(q,i)$ have the same meaning for $b(i)$ and $c(i)$.

In total $s_1(X,Y)$ can be represented as a large convolution in a form $M_{q,i}N^qK^i$ where summing is over the same index that is placed up and down. Vectors are $N^q = [Y, Y^1, ..., Y^Q]$ and $K^i = [X, X^2, ..., X^{3N+1}]$, so the matrix $M_{q,i}$ is sparse and q-th row is formed by the concatenation of coefficients of $u(q,i)$, $v(q,i)$ and $w(q,i)$ (i notation is abused). For two multiplication gates (giving variables $a(1), a(2), ..., c(2)$) and a linear constraint $10a(1) - b(1) - c(2) = 0$ a first row would look like

$$[10, 0, -1, 0, 0, -1]$$

There are three questions:

- Original paper states that $s_1(X,Y)$ can be represented as a sum of three polynomials, each of those being a permutation by itself. Why three? One could try to transform a whole matrix $M_{q,i}$ to have one permutation argument.
- If s_1 is split into sum of three polynomials, are those polynomials each form an individual permutation argument for components like $\sum_{i=1}^{N} u_i'(Y)X^{-i+N+1}$?
- What would be the most efficient procedure to do such a reduction? Just from an example above with a single constraint in a form $[10, 0, -1, 0, 0, -1]$ a first element will contribute in a summand $10X^2Y^1$, while to make a permutation argument one has to first create a term $10X^2Y^2$.