Algorithm 1: Pauli Strings Simplification in BSF

Input : Pauli strings list *pls*
Output: Reconfigured circuit components list *cfg*

cfg ← ∅; *bsf* ← BSF(*pls*); *cliffs_with_locals* ← ∅;
while *bsf*.totalWeight() > *2* **do**
    local_bsf ← *bsf*.popLocalPaulis();
    C ← ∅ ; // Clifford2Q candidates
    B ← ∅ ; // Each element of *B* results from applying each Clifford2Q candidate on *bsf*
    costs ← ∅ ; // Cost functions calculated on each element of *B*
    for *cg* ***in*** CLIFFORD_2Q_SET **do**
        for *i, j* ***in*** combinations(range(*n*), 2) **do**
            cliff ← *cg*.on(*i*, *j*) ; // qubits to act on
            bsf′ ← *bsf*.applyClifford2Q(*cliff*);
            cost ← calculateBSFCost(*bsf′*);
            C.append(*cliff*);
            B.append(*bsf′*);
            costs.append(*cost*);
        end
    end
    bsf ← BSFWithMinCost(*B*, *costs*);
    cliff ← CliffordWithMinCost(*C*, *costs*);
    cliffs_with_locals.append((*cliff*, *local_bsf*));
end
cfg.append(*bsf*);
for *cliff, local_bsf* ***in*** *cliffs_with_locals* **do**
    // Clifford2Q operators are added as conjugations, with local Pauli strings removed before each search epoch
    cfg.prepend(*cliff*);
    cfg.append(*local_bsf*);
    cfg.append(*cliff*);
end