Algorithm 1: SNC-SIMP for two objectives. The main passages of the algorithm are presented. The optimised points are generated iteratively by the MultiSimp function or by OptiStruct.

// Generate the m reference points by solving the corresponding SOPs.
for $j \leftarrow 1$ ***to*** m **do**
    $\{\mu_j\} \leftarrow$ j-th anchor point solution of SOP.
// Anchor points are normalised.
$\bar{\mu}_j = \frac{\mu_j - \min \mu_j}{\max \mu_j - \min \mu_j}$
$\bar{\mu} = \{\bar{\mu}_1, \ldots, \bar{\mu}_m\}$
// Calculate utopia vectors.
$N_j = \bar{\mu}_j - \bar{\mu}_m$
// Assign PIT region parameters.
$a \leftarrow \{a_1, \ldots, a_m\}$
$p = 0.4$
// Generate N evenly spaced approximation points associated to each anchor point.
$\alpha = linspace(0, 1, N)$
$\{U_i\} = \sum_{j=1}^{m} \alpha_i \bar{\mu}_j \quad \forall i = 1, \ldots, N$
while $\{U_i\}$ *is not empty* **do**
    // Call SmartDistance function to calculate the smart distances between anchor points and approximation points and select the approximation point with the largest smart distance.
    $\{s\} \leftarrow SmartDistance(\bar{\mu}, a, p)$
    // Select the two approximation points adjacent to U_ℓ.
    $X_{p1} = U_{\ell-1}$
    $X_{p2} = U_{\ell+1}$
    // Call MultiSimp function to solve the SOP associated to the point U_ℓ and subject to the addtional normal constraints. The MultiSimp function is replaced with a call to OptiStruct for the case of the wing model.
    $\{\mu_p\} = MultiSimp(N_j, \bar{\mu}, X_{p1}, X_{p2})$
    if μ_p *is optimal* **then**
        $\mu \leftarrow \{\mu, \mu_p\}$
    // Remove the approximation point used at this iteration and restart.
    $\{U_i\} \leftarrow (U_i \setminus U_\ell)$

Algorithm 2: MultiSimp function to solve SOP subject to normal constraints. This pseudocode describes only the normal constraints associated to the bi-objective problem described in this work.

Input: $(N_j, \bar{\mu}, X_{p1}, X_{p2})$

Output: $\mu_p, \bar{\mu}_p$

while $change > tol \wedge iter < maxiter$ **do**

    `// Calculate non-dimensional compliance` $\bar{C}$ `and first eigenfrequencies` $\bar{\omega}_1$`.`

    $\mu_p \leftarrow \{C, \omega_1\}$;

    `// Update constraint vector.`

    $g(x) = (X_{p1}(2) - \bar{\omega}_1) + \left(\bar{C} - X_{p1}(1)\right), (\bar{\omega}_1 - X_{p2}(2)) + \left(X_{p2}(1) - \bar{C}\right)$;

    `// Call mmasub function. See relative documentation for input arguments.`

    $x_{new} = mmasub(x, f, dfdx, g, dgdx)$;

    $change = x - x_{new}$