Article

An SQP Algorithm for Structural Topology Optimization Based on Majorization–Minimization Method

Weilong Liao [ID] **, Qiliang Zhang * and Huanli Meng**

School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China; liaowlong@mail2.sysu.edu.cn (W.L.); menghli@mail2.sysu.edu.cn (H.M.)
* Correspondence: zhangqliang@mail.sysu.edu.cn

Abstract: When applying the sequential quadratic programming (SQP) algorithm to topology optimization, using the quasi-Newton methods or calculating the Hessian matrix directly will result in a considerable amount of calculation, making it computationally infeasible when the number of optimization variables is large. To solve the above problems, this paper creatively proposes a method for calculating the approximate Hessian matrix for structural topology optimization with minimum compliance problems. Then, the second-order Taylor expansion transforms the original problem into a series of separable and easy-to-solve convex quadratic programming (QP) subproblems. Finally, the quadratic programming optimality criteria (QPOC) method and the QP solver of MATLAB are used to solve the subproblems. Compared with other sequential quadratic programming methods, the advantage of the proposed method is that the Hessian matrix is diagonally positive definite and its calculation is simple. Numerical experiments on an MBB beam and cantilever beam verify the feasibility and efficiency of the proposed method.

Keywords: topology optimization; majorization–minimization; Hessian matrix; sequential quadratic programming; compliance

1. Introduction

Structural topology optimization aims to optimize the material layout in a given design space under a given set of loads, boundary conditions, and constraints to achieve the optimal performance of the structure. At present, structural topology optimization has been widely used in related technical fields, such as aerospace [1], 3D print [2], and reliability study of structures [3]. Since Bendsøe and Kikuchi conducted seminal work [4] on numerical topology optimization in 1988, more and more structural topology optimization methods have been proposed, such as density-based method [5], level set method (LSM) [6], moving morphable components (MMC) [7], and bidirectional evolutionary structural optimization (BESO) [8]. Among them, the density-based method has received wide attention due to its strong adaptability and simplicity of implementation. Accordingly, a series of mature solution algorithms have been developed. These algorithms can be divided into two main categories: one is the optimality criteria (OC) [9] algorithm and its extensions, such as the PDOC [10] method based on the idea of proportional differential control. The advantages of this type of method are the high computational efficiency, and the speed of the solving is less sensitive to the number of design variables. However, this type of method can be used for single-constraint optimization problems, while it is difficult to be extended to multi-constraint optimization problems. Another class of algorithms to solve the density-based method is represented by the approximation algorithms. The main idea of this class of methods is to transform the original problem into a simpler subproblem, and then use the solution of the subproblem as the initial point for the next iteration until convergence. The main approximation algorithms are sequential linear programming (SLP) [11] SQP [12], the method of moving asymptote (MMA) [13], and its globally convergent counterpart

(GCMMA) [14]. This type of method can solve multi-constraint problems, but its solving speed is slower than that of the previous methods.

The SQP algorithm has been one of the most successful general methods for solving nonlinear constrained optimization problems. The solving of nonlinear problems with SQP is divided into two main steps. The first step is calculating the Hessian matrix to obtain the subproblem, and the second step is solving this subproblem. Compared with other nonlinear programming problems, QP problems are more attractive. Etman et al. [15] transformed the sequential convex programming subproblem into a Newton Lagrange QP subproblem, and Rong et al. [16] directly used the Hessian matrix in the MMA subproblem to construct the quadratic programming subproblem. In practice, the Hessian matrix of most optimization problems is difficult to calculate. The SQP algorithms generally use quasi-Newton [17] methods to approximate the Hessian matrix to obtain the QP subproblem. In topology optimization, TopSQP [18] uses information from part of the second-order derivatives to ensure the positive definiteness of the Hessian matrix.

Albeit successful in solving nonlinear optimization problems, the SQP method is not as widely used as the MMA and OC methods in structural topology optimization. The main reason is that the calculation of the Hessian matrix may become a heavy burden for large-scale topology optimization problems. On the one hand, the existence of the equilibrium equation makes it difficult to calculate the exact Hessian matrix, which may also not be positive definite; on the other hand, using the quasi-Newton method to approximate the Hessian matrix will lead to a massive amount of computation and storage. Facing these obstacles, using the problem-driven majorization–minimization (MM) [19] algorithm to compute the approximate Hessian matrix would be a good breakthrough point. This paper proposes a method to calculate the approximate Hessian matrix according to the MM algorithm, which takes advantage of the problem structure. Then, the original problem is expanded by second-order Taylor expansion to form a QP subproblem. Finally, to verify the correctness of the approximate Hessian matrix method, this paper uses the proposed QPOC method and the QP solver of MATLAB to solve the subproblem. As the numerical experiments show, the proposed method in this work has certain advantages in terms of convergence rate, cost time, and optimal solution compared with the OC and MMA methods.

The paper is organized as follows: Section 2 describes the MM algorithm and how to calculate the approximate Hessian matrix. The algorithms to solve the subproblem are presented in Section 3. Section 4 provides two benchmark problems to validate the effectiveness of the proposed method. Concluding remarks are provided in Section 5.

2. Quadratic Programming Subproblem

2.1. Optimization Model

To produce an almost void-and-solid design, the density-based method uses the solid isotropic material with penalization (SIMP) [20] material interpolation model to penalize elements with intermediate densities. The relationship between Young's modulus and density is defined as

$$E_i(x_i) = E_{\min} + x_i^p(E_0 - E_{\min}) \quad x_i \in [0,1] \quad , \tag{1}$$

where E_0 is the stiffness of the material, x_i is the design variable, $E_{\min}$ is a very small positive number to avoid singularity of the stiffness matrix, and p is a penalization factor introduced to ensure black-and-white solutions.

The mathematical formulation of the minimum compliance problem can be expressed as

$$
\begin{cases}
min & C = \mathbf{F^T U(x)} = \mathbf{U(x)^T K(x) U(x)} = \sum_{i=1}^{N} E_i(x_i)\mathbf{u_i^T k_0 u_i} \\
st. & \frac{V(\mathbf{x})}{V_0} \leq f \quad, \\
& 0 < x_{\min} \leq x_i \leq x_{\max} = 1 \\
& \mathbf{F=K(x)U(x)}
\end{cases}
\tag{2}
$$

where C is compliance, $\mathbf{F}$ and $\mathbf{U}$ are the force and global displacement vectors, respectively, N is the number of elements used to discretize the design domain, $\mathbf{K}$ is the global stiffness matrix, $\mathbf{u_i}$ is the element displacement vector, $\mathbf{k}_0$ is the element stiffness matrix for an element with unit Young's modulus, V_0 and $V(\mathbf{x})$ are the design domain volume and material volume, respectively, and f is the prescribed volume fraction. The derivative of the objective function is found as

$$
\frac{\partial C}{\partial \mathbf{x}} = \mathbf{F^T} \frac{\partial \mathbf{U}}{\partial \mathbf{x}} = \mathbf{U^T K} \frac{\partial \mathbf{U}}{\partial \mathbf{x}}.
\tag{3}
$$

Deriving the derivatives of the equilibrium equation with respect to the design variables and manipulating it mathematically, one can write

$$
\mathbf{K} \frac{\partial \mathbf{U}}{\partial \mathbf{x}} = -\frac{\partial \mathbf{K}}{\partial \mathbf{x}} \mathbf{U}.
\tag{4}
$$

Substituting Equation (4) into Equation (3) and then generalizing the sensitivity of the compliance in relation to the "N" design variables yields

$$
\frac{\partial C}{\partial \mathbf{x}} = -\left[\mathbf{U^T} \frac{\partial \mathbf{K}}{\partial x_1} \mathbf{U}, \dots, \mathbf{U^T} \frac{\partial \mathbf{K}}{\partial x_N} \mathbf{U}\right]^T.
\tag{5}
$$

In this way, the sensitivity of the objective function can be rewritten as

$$
\frac{\partial C}{\partial x_i} = -p(E_0 - E_{min})x_i^{p-1}\mathbf{u_i^T k_0 u_i}.
\tag{6}
$$

The exact Hessian matrix of the objective function can be expressed as

$$
\begin{aligned}
\mathbf{H} &= \frac{\partial^2 C}{\partial \mathbf{x}^2} = -\mathbf{U^T} \frac{\partial^2 \mathbf{K}}{\partial \mathbf{x}^2} \mathbf{U} - 2\frac{\partial \mathbf{U^T}}{\partial \mathbf{x}} \frac{\partial \mathbf{K}}{\partial \mathbf{x}} \mathbf{U} \\
&= -diag\left(\mathbf{U^T} \frac{\partial^2 \mathbf{K}}{\partial x_i^2} \mathbf{U}\right) + 2\left(\frac{\partial \mathbf{K}}{\partial x_1} \mathbf{U}, \dots, \frac{\partial \mathbf{K}}{\partial x_N} \mathbf{U}\right)^T \mathbf{K}^{-1}\left(\frac{\partial \mathbf{K}}{\partial x_1} \mathbf{U}, \dots, \frac{\partial \mathbf{K}}{\partial x_N} \mathbf{U}\right) \\
&= -\mathbf{A} + \mathbf{B}
\end{aligned}
\tag{7}
$$

Here, the matrices $\mathbf{A}$ and $\mathbf{B}$ are both positive definite symmetric matrices; $\mathbf{A}$ is a diagonal matrix whose each term on the diagonal can be expressed as

$$
\mathbf{U^T} \frac{\partial^2 \mathbf{K}}{\partial x_i^2} \mathbf{U} = p(p-1)(E_0 - E_{min})x_i^{p-2}\mathbf{u_i^T k_0 u_i} = \frac{-(p-1)C'(x_i)}{x_i} > 0.
\tag{8}
$$

2.2. Majorization–Minimization Algorithm

The MM algorithm is an algorithmic framework commonly used in machine learning and signal processing, and many algorithms can be interpreted by the MM algorithm framework, such as the EM [21] algorithm. The main idea of the MM algorithm is to find a surrogate function at the current iteration point and then use the extreme point of the surrogate function for the next iteration point, which is similar to the MMA method. The

function $g(x, x_k)$ is called the surrogate function of the original function $f(x)$ if it satisfies the following two properties:

$$f(x) \leq g(x, x_k)$$
$$f(x_k) = g(x_k, x_k)$$

$$(9)$$

where the x_k is the current iteration point. The procedure of the MM is shown pictorially in Figure 1. Generally speaking, there are two principles for finding a surrogate function. One is that the surrogate function should approximate the original function as closely as possible to achieve faster convergence. The other is that the surrogate function should be as simple as possible to reduce the computational cost. For more methods of constructing surrogate functions, the reader is referred to [22]. The MM algorithm is locally convergent and converges to the stationary point, while its good results in practice have led to its wide application.
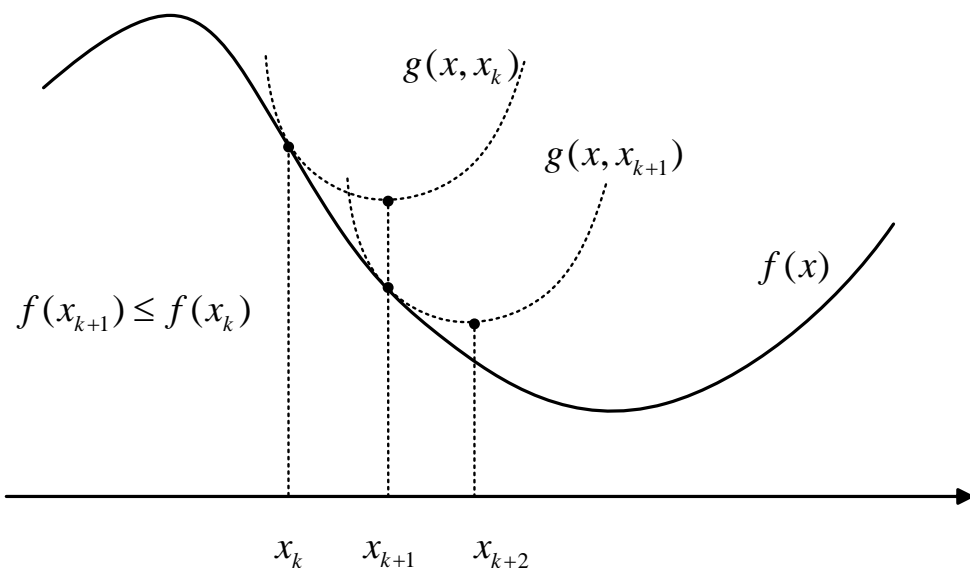

Figure 1. MM algorithm procedure.

2.3. Quadratic Programming Subproblem

The method [22] of combining the MM algorithm with the SQP algorithm is to find an approximate Hessian matrix $\mathbf{H}$, where $\mathbf{H} \succ \mathbf{B} - \mathbf{A}$. Since matrices $\mathbf{A}$ and $\mathbf{B}$ are positive definite, the most straightforward approach is to take $\mathbf{H} = \mathbf{B}$ like the TopSQP method, which can ensure that the matrix $\mathbf{H} - (\mathbf{B} - \mathbf{A})$ is positive definite. However, matrix $\mathbf{B}$ needs to calculate the inverse of the global stiffness matrix, which is very expensive for large-scale topology optimization problems and does not meet the requirements of low computational cost. Since the matrix $\mathbf{A}$ is diagonally positive definite, there must exist a curvature parameter $q(q > 0)$, such that $q\mathbf{A} \succ \mathbf{B} - \mathbf{A}$. Then, the surrogate function of Equation (2) can be expressed as

$$\Phi(\mathbf{x}) = \frac{q}{2}(\mathbf{x} - \mathbf{x}^*)^T \mathbf{A}(\mathbf{x} - \mathbf{x}^*) + C'(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + C(\mathbf{x}^*). \tag{10}$$

The solution of Equation (10) is given by

$$\mathbf{x} = \mathbf{x}^* - \frac{\mathbf{A}^{-1}C'(\mathbf{x}^*)}{q}. \tag{11}$$

In this paper, we do not directly calculate the value of q, but use numerical experiments to verify the effect of different q on the convergence result. The QP subproblem of Equation (2) is as follows:

$$
\begin{cases}
min & \Phi(\mathbf{x}) = \frac{q}{2}(\mathbf{x} - \mathbf{x}^*)^T \mathbf{A}(\mathbf{x} - \mathbf{x}^*) + C'(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + C(\mathbf{x}^*) \\
st. & V(\mathbf{x}) = fV_0 = \sum\limits_{i=1}^{N} x_i v_i \\
& x_i^{lb} \le x_i \le x_i^{ub}
\end{cases}
\tag{12}
$$

where

$$
\begin{aligned}
x_i^{lb} &= \max(x_{\min}, x_i - move) \\
x_i^{ub} &= \min(x_{\max}, x_i + move)
\end{aligned}
\tag{13}
$$

where *move* is the moving limit, which is usually set in topology optimization to ensure the stability of the solutions. The surrogate function has several properties desired by the literature [22]: convex separable and the existence of a closed-form minimizer. The design of efficient and neat solution algorithms to verify the feasibility of the surrogate function is presented in the next section.

3. Solution of Subproblem

3.1. QPOC Method

Since the subproblem in Equation (12) is convex and separable, the algorithm for solving it should take full advantage of this property. First, before solving Equation (12), we construct an easy to solve quadratic programming problem, which is

$$
\begin{cases}
min & \Phi(\mathbf{x}; \lambda) = \frac{q}{2}(\mathbf{x} - \mathbf{x}^*)^T A(\mathbf{x} - \mathbf{x}^*) + C'(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + C(\mathbf{x}^*) + \lambda\left(\sum\limits_{i=1}^{N} x_i v_i - V^*\right), \\
st. & x_i^{lb} \le x_i \le x_i^{ub}
\end{cases}
\tag{14}
$$

where λ is a scalar parameter. Because **A** is a diagonal positive definite matrix, Equation (14) can be decomposed into N problems; the optimal solution of $\Phi(\mathbf{x}; \lambda)$ is

$$
\begin{aligned}
\mathbf{x}(\lambda) &= \frac{\mathbf{A}^{-1}}{q}(-C'(\mathbf{x}^*) - \lambda v \mathbf{I}) + \mathbf{x}^* \\
x_i(\lambda) &= \frac{(q(p-1)+1)C'(x_i^*) + \lambda v_i}{q(p-1)C'(x_i^*)} x_i^*
\end{aligned}
\tag{15}
$$

Constrained by upper and lower bounds, the solution of Equation (14) is given by

$$
x_i(\lambda) = \text{middle}(x_i^{lb}, x_i(\lambda), x_i^{ub}).
\tag{16}
$$

Then, we solve the univariate nonlinear equation,

$$
r(\lambda) = \sum\limits_{i=1}^{N} x_i(\lambda) v_i - fV_0 = 0.
\tag{17}
$$

According to Equations (15) and (16), $x_i(\lambda)$ and $r(\lambda)$ are monotonically decreasing continuous functions. Note that $r(0) > 0, r(\infty) < 0$; hence, Equation (17) must have a unique solution. The literature [23] has a detailed description of the solution of single-dimensional functions and gives several efficient solutions, such as the dichotomous method, secant method, and brent method. To better combine the 88 lines of code (OC) [24], the dichotomous method was selected in this paper. Because the KT conditions of Equation (14) and Equation (17) constitute the KT conditions of Equation (12), once λ^* is located such that Equation (17) holds, $\mathbf{x}(\lambda^*)$ is a global minimizer since it meets the KT conditions and convexity of Equation (12).

The efficiency of the proposed method is mainly reflected in two aspects. One is the low computation and storage cost of the Hessian matrix. The other is to transform

the N-dimensional QP subproblem into the root-finding problem of a one-dimensional function, which significantly accelerates the solving speed.

In fact, the update formula of the OC method is also explicitly expressed as

$$x_i(\lambda) = \text{middle}(x_i^{lb}, x_i^* (\frac{-C'(x_i^*)}{\lambda v_i})^{1/2}, x_i^{ub}).$$ (18)

Equation (18) also uses the dichotomous method to solve the Lagrange multipliers, which means that reproducing the above algorithm requires only initializing q and modifying the update formula for the design variables in 88 lines of code. In summary, the proposed algorithm is more like the application of the OC method in quadratic programming; therefore, it is called the quadratic programming optimality criteria (QPOC) method.

3.2. IPC Method

The QPOC method can only be used to solve single linearly constrained quadratic programs subject to lower and upper bounds. For topology optimization, the constraints are not only the volume constraint; fixed value constraints and perimeter constraints [25] can also be added. At present, the mathematical community has systematically studied the solution algorithm of multi-constraint quadratic programming, mainly including the interior point method [26], trust-region method [27], and gradient projection method [28]. To verify the feasibility of the QP subproblem for multiple constraints, we used the interior-point convex (IPC) algorithm that comes with the built-in function *quadprog* of MATLAB.

4. Numerical Examples

In this paper, two benchmark examples of an MBB beam and cantilever beam are used to demonstrate the performance of the proposed algorithm. The Young's modulus $E_0 = 1, E_{\min} = 0$, and penalty factor $p = 3$; to avoid the singular value phenomenon, $x_{\min} = 0.001$. Regarding the filters, the QPOC, OC, and MMA methods use sensitivity filters, while the IPC method uses density filters that do not use chain rule, yielding better results. As for the convergence criterion, QPOC uses the same convergence criterion as the OC method. For MMA and IPC, due to the large change of design variables, it is necessary to set new convergence criteria $mean(sum(abs(\mathbf{x}^{k+1} - \mathbf{x}^k))) < 4 \times 10^{-4}$ and $iterations < 200$. To measure whether an optimized design has converged to a discrete solution, a measurement index [29] is introduced.

$$M_{\text{nd}} = \frac{\sum\limits_{i=1}^{N} 4x_i(1 - x_i)}{N} \times 100\%.$$ (19)

If all elements' densities are equal to 1 or 0, the optimized design is fully discrete, and M_{nd} is 0%. A smaller value of M_{nd} means that there are fewer intermediate densities in the structure, and the diagram of the optimized structure will be clearer. In addition, all the experiments were executed on a personal computer with an Intel CORE i7 11700 processor, Windows 10 (64 bit), and MATLAB R2021a.

4.1. MBB Beam

The MBB beam is a typical benchmark example used in topology optimization. In accordance with [24], the design domain, the boundary conditions, and the external load for the MBB beam are shown in Figure 2. The design domain was discretized with 120×40 square elements, and the volume fraction f was set to 0.5. The optimized designs obtained using the OC, IPC, and QPOC methods are illustrated in Figure 3.
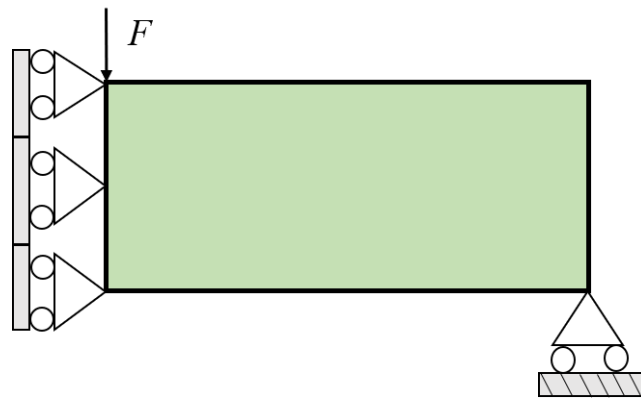
Figure 2. Design domain of the MBB beam.

| (**a**) OC r = 4.8 | (**b**) QPOC q = 1, r = 4.8 | (**c**) QPOC q = 2, r = 4.8 |

| (**d**) QPOC q = 3, r = 4.8 | (**e**) IPC q = 1, r = 4.8 | (**f**) IPC q = 1.5, r = 3.6 |

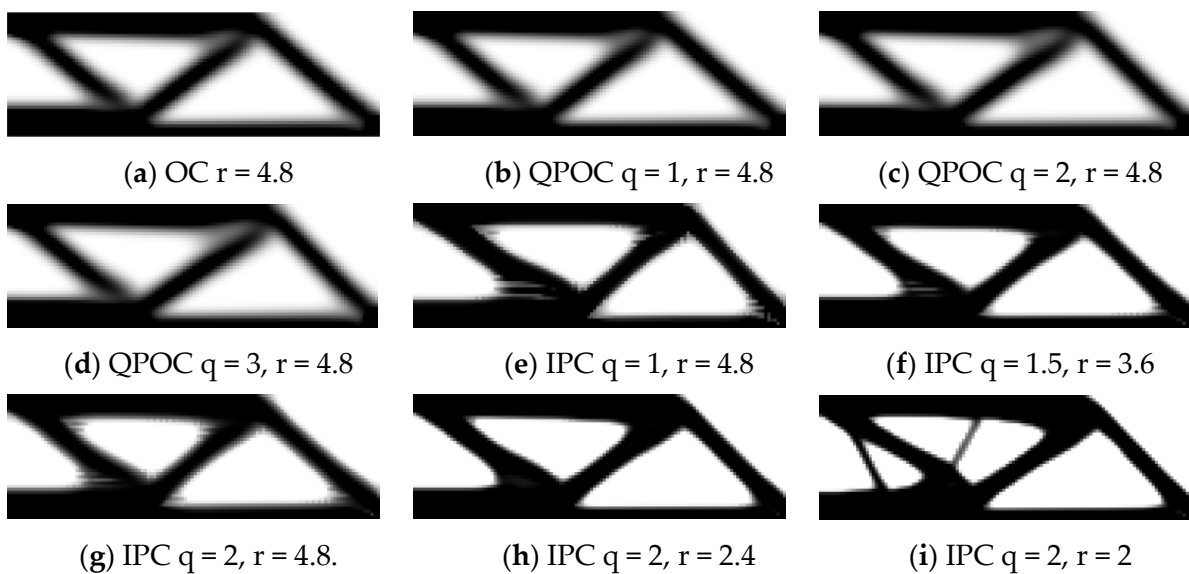| (**g**) IPC q = 2, r = 4.8. | (**h**) IPC q = 2, r = 2.4 | (**i**) IPC q = 2, r = 2 |

Figure 3. Optimized designs of the MBB beam with different methods and parameters.

An excellent solution algorithm is mainly reflected in the optimal solution, convergence rate, and calculation cost. The advantages and disadvantages of the method proposed in this paper was compared with the OC method from these aspects.

4.1.1. Optimal Solution and Optimized Design

It can be seen in Figure 3 that, by selecting appropriate curvature parameters and filter radius, the QPOC method and IPC method could obtain the optimized structures similar to the OC method, which verifies the feasibility of the proposed method. The optimized structures obtained using the QPOC method and OC method had the phenomenon of fuzzy boundary, while the optimized structures (Figure 3h,i) obtained using the IPC method were clearer, and the structure contour was more distinct.

The curvature parameter and the filter radius have a crucial influence on the structure, and it is necessary to explore the optimal combination of the two parameters. Because QPOC is a derivative algorithm of the OC method, its filter radius should be chosen consistently with the OC method, which is also verified by the fact that Figure 3b–d exhibits a similar structure with the same filter radius as the OC method. For the IPC method, reducing the filter radius can obtain a clearer structure as shown in Figure 3g,h, but too small a filter radius will lead to fine branches in the optimized structure and affect the fabrication, as shown in Figure 3i. It can be seen from Figure 3e–g that when the curvature parameter was increased, the filter radius could be appropriately reduced to obtain a clearer structure without fine branches.

To further analyze the influence of the QPOC and IPC methods and parameters on the optimized structure, the compliance and discrete index M_{nd} were calculated, and the results are shown in Table 1. As can be seen from Table 1, the values of M_{nd} for the QPOC and OC methods were comparable, while the M_{nd} for the IPC method was much smaller than that of the other two methods. When $q = 2$ and $r = 2$, the M_{nd} of the IPC method was the smallest, 8.5%, approximately 72% smaller than that of the OC method, which indicates that the IPC method could significantly reduce the intermediate densities and reduce the boundary-blurring effect.

Table 1. Optimization data for different methods and parameters.

Method	q	Figure 3	Radius r	f^* *	Iterations	Compliance	M_{nd} (%)	Time Cost (s)	Per_Time Cost (s)
OC	None	a	4.8	0.5	93	219.2	26.5	4.0	0.04
QPOC	1	b	4.8	0.5	101	219.2	26.5	4.2	0.04
	1.5	None	4.8	0.5	72	219.5	26.8	3.1	0.04
	2	c	4.8	0.5	65	220.5	27.3	2.8	0.04
	3	d	4.8	0.5	64	222.4	28.4	2.7	0.04
IPC	1	e	4.8	0.5	200	209.0	15.2	49.2	0.25
	1.5	f	3.6	0.5	200	202.2	11.7	41.2	0.21
	2	g	4.8	0.5	200	211.4	18.0	38.7	0.19
	2	h	2.4	0.5	137	199.1	8.7	27.0	0.20
	2	i	2	0.5	200	196.1	8.5	42.9	0.21
	3	None	2.2	0.5	142	201.4	9.6	29.0	0.20

* f^* is the volume fraction of the final optimized structure.

In addition, the compliance of QPOC and OC methods was approximately equal; the compliance obtained by the IPC method was smaller than that of the previous two methods. It is worth noting that the compliance of the IPC ($q = 2, r = 2.4$) method was about 12% smaller than that of the OC method. Because the optimized structures of both had similar geometries, the reason for this phenomenon was that the M_{nd} of the IPC method was smaller, which means that the optimized structure had fewer intermediate densities, while SIMP penalized the stiffness of elements with intermediate density, resulting in a decrease in the integral structural stiffness.

In summary, the optimized structure and compliance of the QPOC method were consistent with those of the OC method, and the selection of the filter radius should be the same as that of the OC method. The optimized structure of the IPC was clearer, with a distinct boundary and smaller compliance. The selection of filter radius was negatively correlated with curvature parameters.

4.1.2. Convergence Rate

Figure 4a shows the convergence curves of the compliance under different methods and parameters. For the QPOC method, when the curvature parameter decreased, the corresponding purple ($q = 3$), red ($q = 2$), and black ($q = 1$) convergence curves were below the previous curve, and the same phenomenon was also observed for the IPC method. This indicates that the curvature parameter was negatively related to the convergence rate. However, this does not mean that the curvature parameter can be arbitrarily small; otherwise, it may lead to the surrogate function being smaller than the original function resulting in non-convergence of the algorithm.

Figure 4b illustrates that the curvature parameter and the average iteration step of the design variables were negatively correlated, and a larger average iteration step could enhance its capability to find the extreme points and improve the convergence rate. It is worth noting that Figure 4b also indirectly illustrates that the selection of the filter radius in the IPC method was positively correlated with the average iteration step of the design variables.
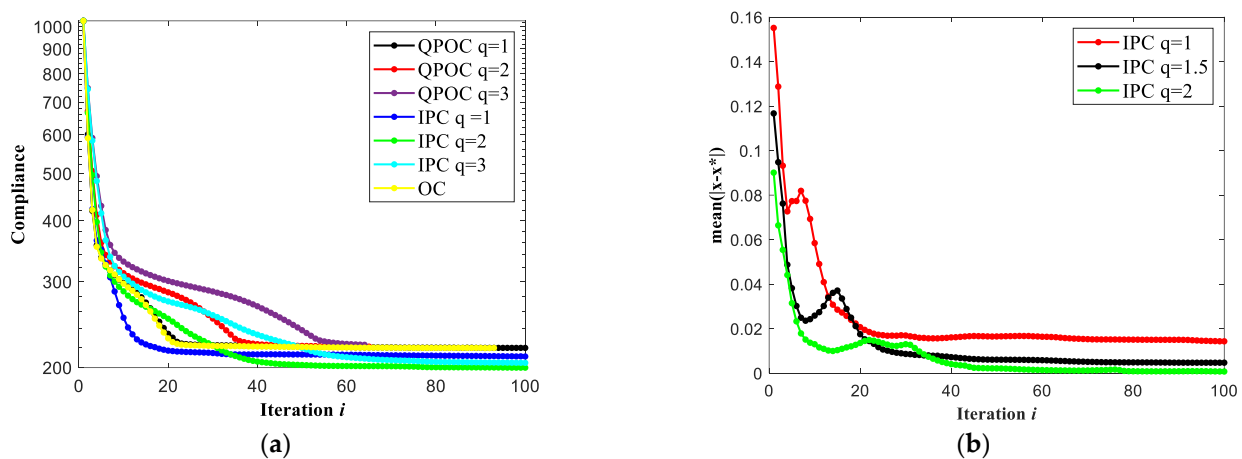
Figure 4. Convergence curves for the MBB beam using different methods and parameters: (**a**) the changes in the compliance; (**b**) average iteration step size using different curvature parameters.

Overall, the IPC ($q = 1$) method had the fastest convergence rate, and the OC and QPOC ($q = 1$) converged at about the same rate. The compliance of all the methods rapidly decreased in the first few steps, and then the rate of decrease slowed down, with the compliance of all methods reaching convergence in fewer than 60 steps.

4.1.3. Calculation Cost

As can be seen from Table 1, the single-step computation time of OC and QPOC was the same, about one-fifth of the single-step computation time of the IPC method. The reason for the basic agreement between the two computational costs is that both update formulas can be expressed explicitly and both use the dichotomous method to compute Lagrange multipliers. The QPOC method transforms the QP subproblem into a root-finding problem for a single-dimensional function, which is much simpler than solving a system of nonlinear equations using the Newton–Raphson method in the IPC method. This is the reason why the computational cost of QPOC is much cheaper than that of the IPC method.

The comparison between different methods of compliance and calculation time and discrete indices can be found in Figure 5, where the data in the figure are all normalized. In general, the OC and QPOC methods had an advantage over the IPC method in terms of computation time, and the QPOC could be adjusted by the curvature parameter ($q = 2$) such that its computation time was less than that of the OC method. IPC method could obtain better optimized structures, and it could be used to solve multi-constrained problems.
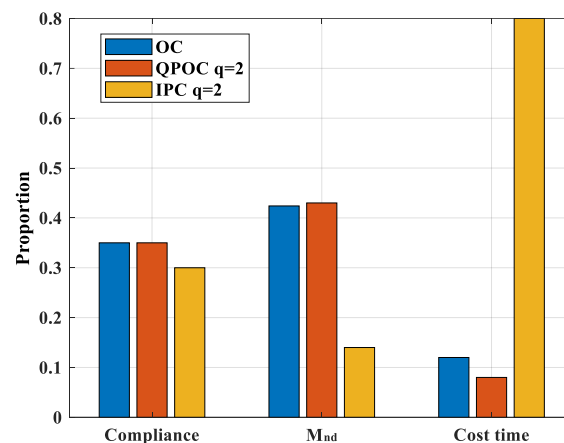

Figure 5. Comparison of optimization indices obtained using different methods.

4.2. Cantilever Beam

To demonstrate the effectiveness of the surrogate function under multiple constraints, a cantilever beam problem is provided in this section. The design domain of the cantilever beam is illustrated in Figure 6. It was fully constrained at the left end and subjected to a downward unit load at the right center. To set multiple constraints, a fixed value constraint was added, and the virtual densities corresponding to the first four columns of finite elements in the design area were set to 1, expressed as follows:

$$\sum_{i=1}^{4*nely} x_i = 4 \times nely. \tag{20}$$

Figure 6. Design domain of the cantilever beam.

In terms of the other parameters, $move = 1, q = 2, f = 0.5$. The design domain was discretized with 60×30, 80×40, and 120×60 square elements. Both the IPC and MMA methods were applied to this problem, and the optimization results are shown in Figure 7.

(**a**) MMA r = 1.6

(**b**) MMA r = 2.4

(**c**) MMA r = 3.6

(**d**) IPC r = 1.6

(**e**) IPC r = 2.4
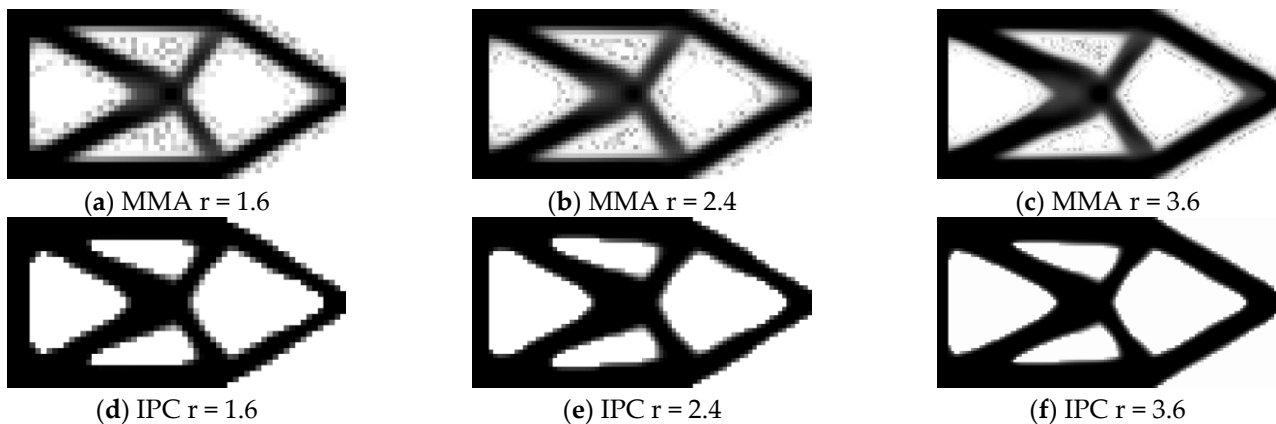

(**f**) IPC r = 3.6

Figure 7. Optimized designs of the cantilever beam using IPC and MMA methods.

As shown in Figure 7, some checkerboard and porous elements appeared around the structure optimized using the MMA method, and the same phenomenon occurred in the literature [30]. Compared with the MMA method, the optimized structure of the IPC method was clearer, and the boundary contours were more distinct, with no checkerboard phenomenon. These benefits suggest that the optimized structure of the IPC method was superior to that of the MMA method in terms of manufacturing.

Table 2 shows the optimization data of both methods for different sizes of design domains. The IPC method performed better than the MMA method. In the design domain, from small to large, the compliance of the IPC method was about 18.8%, 18.9%, and 13% smaller than that of MMA, respectively, while the number of iterations was 76%, 76%, and 42% smaller, respectively, and the overall calculation time was about 60%, 74%, and 4.2% smaller, respectively. It is worth noting that, in Figure 8a, the compliance curve of the MMA method showed a peak value, while the volume ratio convergence curve of MMA

in Figure 8b showed a corresponding valley value. The decrease in volume ratio was the reason for the sudden increase of compliance, which also indicates that the MMA method was not as capable of maintaining constraints as the IPC method.

Table 2. Optimization data for different methods and finite element meshes.

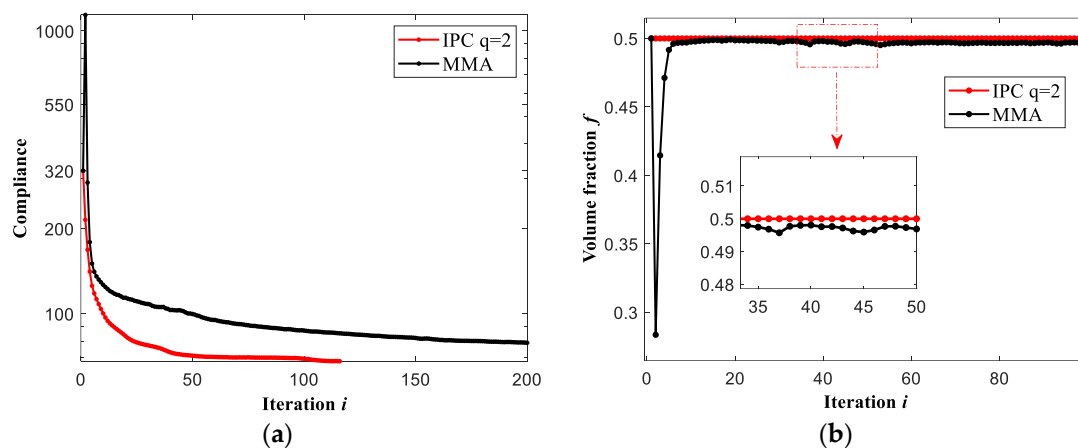| Meshes	Method	Figure 7	Radius r	f^*	Iterations	Compliance	M_{nd} (%)	Time Cost (s)	Per_Time Cost (s)
60 × 30	MMA	a	1.6	0.489	200	87.1	25.6	9.3	0.05
	IPC	d	1.6	0.500	48	70.7	7.1	3.7	0.07
80 × 40	MMA	b	2.4	0.491	200	87.5	27.9	24.7	0.12
	IPC	e	2.4	0.500	47	71.0	8.0	6.2	0.13
120 × 60	MMA	c	3.6	0.500	200	78.9	34.8	44.2	0.22
	IPC	f	3.6	0.496	116	67.9	9.1	42.3	0.36

Figure 8. Convergence curves for the cantilever beam using MMA and IPC methods: (**a**) the changes in the compliance; (**b**) the changes in the volume fractions.

5. Conclusions and Discussion

This work introduced the MM algorithm to topology optimization, providing a new perspective for the application of the SQP algorithm in topology optimization. Numerical experiments on an MBB beam and cantilever beam verified the feasibility and efficiency of the proposed method. The contribution of our method can be concluded as follows:

1. We proposed a novel way to calculate the Hessian matrix for structural topology optimization with minimum compliance problems. The Hessian matrix could be given almost simultaneously when solving the first derivative, and the calculation cost was almost negligible.
2. The QP subproblem is formed by the second-order Taylor expansion of the original problem, which is convex separable and easy to be imported into the QP solver. The convergence rate can be controlled by the curvature parameter.
3. To solve the QP subproblem, we proposed the QPOC method. The derivation of the update formula of this method is supported by rigorous mathematical theory, while the update formula of the OC method is heuristic.

Compared to using general-purpose solvers, a specialized algorithm devised according to the structure of the problem might be a better option.

Author Contributions: Conceptualization, W.L. and H.M.; methodology, W.L. and H.M.; software, W.L.; validation, W.L. and H.M.; formal analysis, W.L., H.M. and Q.Z.; investigation, W.L. and H.M.; resources, Q.Z.; data curation, Q.Z.; writing—original draft preparation, W.L.; writing—review and editing, W.L.; visualization, W.L.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, Q.Z. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Wang, Q.; Wang, C.; Gong, P.; Shi, Y.; Yu, Y.; Liu, Z. Topology Optimization Design and Experimental Research of a 3D-Printed Metal Aerospace Bracket Considering Fatigue Performance. *Appl. Sci.* **2021**, *11*, 6671. [CrossRef]
2. Vantyghem, G.; De Corte, W.; Shakour, E.; Amir, O. 3D printing of a post-tensioned concrete girder designed by topology optimization. *Automat. Constr.* **2020**, *112*, 103084. [CrossRef]
3. Simonetti, H.L.; Almeida, V.S.; de Assis Das Neves, F.; Del Duca Almeida, V.; de Oliveira Neto, L. Reliability-Based Topology Optimization: An Extension of the SESO and SERA Methods for Three-Dimensional Structures. *Appl. Sci.* **2022**, *12*, 4220. [CrossRef]
4. Bendsøe, M.P.; Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Comput. Method. Appl. M.* **1988**, *71*, 197–224. [CrossRef]
5. Bendsøe, M.P. Optimal shape design as a material distribution problem. *Struct. Optim.* **1989**, *1*, 193–202. [CrossRef]
6. Wang, M.Y.; Wang, X.; Guo, D. A level set method for structural topology optimization. *Comput. Method. Appl. M.* **2003**, *192*, 227–246. [CrossRef]
7. Guo, X.; Zhang, W.; Zhong, W. Doing Topology Optimization Explicitly and Geometrically—A New Moving Morphable Components Based Framework. *J. Appl. Mech.* **2014**, *81*, 081009. [CrossRef]
8. Huang, X.; Xie, Y.M. Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials. *Comput. Mech.* **2008**, *43*, 393. [CrossRef]
9. Ananiev, S. On Equivalence Between Optimality Criteria and Projected Gradient Methods with Application to Topology Optimization Problem. *Multibody Syst. Dyn.* **2005**, *13*, 25–38. [CrossRef]
10. Zhao, L.; Chen, Z.; Qiu, H.; Gao, L. *A PDOC Method for Topology Optimization Design, International Conference on Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2010. [CrossRef]
11. Gomes, F.A.; Senne, T.A. An SLP algorithm and its application to topology optimization. *Comput. Appl. Math.* **2011**, *30*, 53–89. [CrossRef]
12. Park, S.; Jeong, S.H.; Yoon, G.H.; Groenwold, A.A.; Choi, D. A globally convergent sequential convex programming using an enhanced two-point diagonal quadratic approximation for structural optimization. *Struct. Multidiscip. Optim.* **2014**, *50*, 739–753. [CrossRef]
13. Svanberg, K. The method of moving asymptotes-a new method for structural optimization. *Int. J. Numer. Meth. Eng.* **1987**, *24*, 359–373. [CrossRef]
14. Svanberg, K. A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations. *SIAM J. Optim.* **2002**, *12*, 555–573. [CrossRef]
15. Etman, L.F.P.; Groenwold, A.A.; Rooda, J.E. First-order sequential convex programming using approximate diagonal QP subproblems. *Struct. Multidiscip. Optim.* **2012**, *45*, 479–488. [CrossRef]
16. Rong, J.H.; Tang, Z.L.; Xie, Y.M.; Li, F.Y. Topological optimization design of structures under random excitations using SQP method. *Eng. Struct.* **2013**, *56*, 2098–2106. [CrossRef]
17. Lewis, A.S.; Overton, M.L. Nonsmooth optimization via quasi-Newton methods. *Math. Program.* **2013**, *141*, 135–163. [CrossRef]
18. Rojas-Labanda, S.; Stolpe, M. An efficient second-order SQP method for structural topology optimization. *Struct. Multidiscip. Optim.* **2016**, *53*, 1315–1333. [CrossRef]
19. Hunter, D.R.; Lange, K. A Tutorial on MM Algorithms. *Am. Stat.* **2004**, *58*, 30–37. [CrossRef]
20. Rozvany, G.I.; Zhou, M.; Birker, T. Generalized shape optimization without homogenization. *Struct. Optim.* **1992**, *4*, 250–252. [CrossRef]
21. Wu, T.T.; Lange, K. The MM alternative to EM. *Stat. Sci.* **2010**, *25*, 492–505. [CrossRef]
22. Sun, Y.; Babu, P.; Palomar, D.P. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. *IEEE T. Signal Proces.* **2017**, *65*, 794–816. [CrossRef]
23. Sauer, T. *Numerical Analysis*, 2nd ed.; Addison-Wesley Publishing Company: Boston, MA, USA, 2011.

24. Andreassen, E.; Clausen, A.; Schevenels, M.; Lazarov, B.S.; Sigmund, O. Efficient topology optimization in MATLAB using 88 lines of code. *Struct. Multidiscip. Optim.* **2011**, *43*, 1–16. [CrossRef]

25. Haber, R.B.; Jog, C.S.; Bends, E.M.P. A new approach to variable-topology shape design using a constraint on perimeter. *Struct. Optim.* **1996**, *11*, 1–12. [CrossRef]

26. Mehrotra, S. On the Implementation of a Primal-Dual Interior Point Method. *SIAM J. Optim.* **1992**, *2*, 575–601. [CrossRef]

27. Wright, S.J.; Tenny, M.J. A feasible trust-region sequential quadratic programming algorithm. *SIAM J. Optim.* **2004**, *14*, 1074–1105. [CrossRef]

28. Serafini, T.; Zanghirati, G.; Zanni, L. Gradient projection methods for quadratic programs and applications in training support vector machines. *Optim. Methods Softw.* **2005**, *20*, 353–378. [CrossRef]

29. Sigmund, O. Morphology-based black and white filters for topology optimization. *Struct. Multidiscip. Optim.* **2007**, *33*, 401–424. [CrossRef]

30. Zuo, K.; Chen, L.; Zhang, Y.; Yang, J. Study of key algorithms in topology optimization. *Int. J. Adv. Manuf. Technol.* **2007**, *32*, 787–796. [CrossRef]