A HYBRID ALGORITHM FOR TOPOLOGY OPTIMIZATION OF ADDITIVE MANUFACTURED STRUCTURES

Aremu A., Ashcroft I., Wildman R., Hague R., Tuck C., Brackett D.

Wolfson School of Mechanical and Manufacturing Engineering,
Loughborough University, Loughborough, LE11 3TU, UK.

REVIEWED, August 17 2011

Abstract

Most topology (TO) algorithms involve the penalization of intricate structural features to eliminate manufacturing difficulties. Since additive manufacturing is less dependent on manufacturing constraints, it becomes necessary to adapt these algorithms for AM. We propose a hybrid algorithm consisting of an adaptive meshing strategy (AMS) and a modified form of the bidirectional evolutionary structural optimization (BESO) method. By solving a standard cantilever problem, we show that the hybrid method offers improved performance over the standard BESO method. It is proposed that the new method is more suitable for optimizing structures for AM in a computational efficient manner.

Introduction

Topology optimization (TO) is the structural optimization method least sensitive to the initial design and so has the greatest capacity for realizing improved design for additive manufacture (AM). It began when Mitchell [1] established principles for the TO of truss-like structures and later, work by Rozvany and Kirsch [2, 3] extended this to allow the optimization of grillages and bar systems. Bendsoe and Kikuchi's [4] homogenization technique significantly advanced TO as it reduced the optimization of continuum structures to that of determining the optimum parameters of holes in unit cells that constituted such structures. Rozvany and Zhou [5] further simplified TO by introducing the SIMP (*Solid Isotropic Microstructure with Penalization*) algorithm to obtain practical designs. Steven and Xie [6] also proposed a simple approach to TO, called evolutionary structural optimization (ESO), though Zhou and Rozvany [7] attribute much of its ground work to Mattheck et al. [8] who used a similar procedure in adaptive biological growth. Querin et al. [9] enhanced the efficiency of the ESO algorithm with the bi-directional evolutionary structural optimization (BESO) algorithm. Huang and Xie [10] improved on this to solve mesh dependency and non-convergent problems. Stochastic optimization algorithms have also been used for TO. Sandgren et al. [11] and Chapman et al. [12, 13] performed genetic algorithm based TO. Kaveh et al. [14], and Luh and Lin [15] used ant colony optimization algorithms. Luh et al. [16] applied particle swarm optimization while Bureerat and Limtragool [17] preferred simulated annealing. Hybrids have also emerged, Zuo et al. [18] combined BESO and a genetic algorithm and Garcia-Lopez [19] coupled SIMP with simulated annealing. These stochastic algorithms operate on a population of solutions for a complete exploration of the design space. However, computational cost increases significantly when they are used to solve complicated three dimensional problems. Similar to non-stochastic algorithms, stochastic algorithms rely on results from a finite element analysis (FEA) to move an initial topology to an optimum.

FEA is a numerical technique used to estimate solutions for problems governed by partial differential equations [20]. It assumes a piecewise form of the equation for a given domain. An assembly of each piece or element constitutes a mesh improved either by $h-$, $p-$ or $r-$ refinement [21]. A common approach to TO involves prior preparation and improvement of the mesh to an acceptable quality, with the mesh remaining fixed for the duration of the TO. Authors illustrate the practicality of their TO algorithm by solving simple problems with this sort of meshing strategy. For three dimensional problems with complicated geometry, satisfactory mesh quality can only be achieved with a high number of elements. Following TO, post processing operations are often a necessity owing to the characteristic rough boundaries of optima. Again, using a finer mesh would reduce the extent of the roughness, however the computational resource required for this often makes them undesirable, especially for three dimensional

problems. Both Kim et al. [22] and Mariano et al. [23] developed algorithms that achieve smoother boundaries at reasonable mesh sizes. However, similar to other TO, their reliance on a fixed mesh limit the appearance of intricate features that could potentially enhance structural performance.

Until recently, the appearance of such intricate features was not desirable due to manufacturing constraints. These small feature were penalized either during or after a TO to allow manufacturability. Recent developments in the field of additive manufacturing (AM) have enabled a significant increase in the level of geometric complexity achievable without a significant increase in the cost of manufacture. This is due to the inherent layer-by-layer approach to manufacturing which eliminates the need for tooling or fixtures [24, 25]. According to Hague et al. [26] *"the advent of additive manufacturing will have profound implications for the way in which designers work."*. Rosen [27] commenced a domain exploration based on sizing optimization of cellular materials called mesostructures. Interestingly, the versatility of this exploration could be enhanced by coupling an adaptive meshing strategy (AMS) with TO algorithms. In this paper, we couple the BESO algorithm to an AMS to solve a two dimensional problem. The discreteness of intermediate topologies in the BESO algorithm makes it suitable for our strategy. To demonstrate the performance of our method, a cantilever plate problem was solved to minimize total strain energy, C, for a given area fraction constraint, A^*. This work is an aspect of a project titled *"Topological optimized additively manufactured structural metallic components"* funded by the engineering and physical science research council (EPSRC) and the innovative manufacturing and construction research centre (IMCRC).

Method

Certain aspects of Huang and Xie's [10] BESO algorithm were modified to allow an effective coupling of the AMS with the BESO algorithm. This hybrid method was then based on an iterative improvement of an unstructured triangular mesh by refining elements at boundaries of intermediate topologies and coarsening other regions. Using Engwirda's [28] code, elements were selectively refined while coarsening was achieved by edges collapse. This necessitated the development of a logic for recognition of elements to refine and edge to collapse. A consequence of this strategy was the degradation of mesh quality minimized by subjecting the emergent mesh to Laplacian smoothening [31] (an $r-$refinement method) method. To illustrate performance, we solve a cantilever plate problem with area, A_i, at iteration, i, experiencing load, F, with an objective to minimize C given A^*. Expressed mathematically as,

$$min : C = \frac{1}{2}\boldsymbol{F}^{\boldsymbol{t}}u, \qquad subject\ to : \frac{A_i}{A_0} = A^* \tag{1}$$

where u is a displacement vector.

Modified BESO

The BESO algorithm was modified to involve,

- FEA,
- Computation of elemental strain energies, λ_a,
- Calculation of filtered elemental strain energy densities, $\overline{\chi}_a$,
- Calculation of a target area, A_1,
- Deletion and addition of elements based on thresholds, $\overline{\chi}^{th}_{del}$, $\overline{\chi}^{th}_{add}$
- Adaptive mesh improvement,
- Comparison of change in strain energy, ΔC, against a tolerance, θ,
- Iterations of steps 1 to 5 until ΔC is lower than θ,

We implemented this BESO version by passing a discretized design domain, filter radius factor, $R^{'}$, and evolution rate, v, to the algorithm. The $R^{'}$ was a scalar multiple used to include contributions of neighboring nodes to the filtered strain energy densities of an element being filtered. The v was the iterative rate of change of area of the undeleted elements. An explicit description of their use is given later. The first step, FEA, was performed using MSC Nastran® (MSC Software, California, Santa Ana). Elemental strain energies, λ_a, were outputted from MSC Nastran® and inputted into MATLAB® (Mathworks, Natick, Massachusetts) where subsequent steps of the BESO were implemented. We then computed elemental strain energy densities, χ_a, from λ_a through

$$\chi_a = \frac{\lambda_a}{A_a} \tag{2}$$

where A_a was the area of element, a. Filtering χ_a suppressed the appearance of checkerboard patterns, and was achieved in two stages. The first involved a distribution of χ_a over connecting nodes to obtain nodal sensitivities, η_b, by

$$\eta_b = \frac{\sum_{a=1}^{N} A_a \chi_a}{\sum_{a=1}^{N} A_a} \tag{3}$$

where N was the last element connected to node b. Secondly, calculating filtered elemental strain energy densities, $\overline{\chi}_a$, from η_b and $R^{'}$ through,

$$\overline{\chi}_a = \frac{\sum_{b=1}^{N} \psi(d_{ab}) \eta_b}{\sum_{b=1}^{N} \psi(d_{ab})} \tag{4}$$

where $\psi(d_{ab}) = R'y - d_{ab}$, y was the radius of a circle that circumscribed the element, a and d_{ab} was the distance between node b and the center of element a. Next, a target area, A_1, was computed from v according to

$$A_1 = \begin{cases} A_i(1-v), \ if & \frac{A_i}{A_0} > A^* \\ A^* A_0, \quad if & \frac{A_i}{A_0} \leq A^* \end{cases} \tag{5}$$

A set of areas, $\widehat{\Omega}$, belonging to undeleted elements was sorted in descending order of $\overline{\chi}_a$. This order allowed the construction of a cumulative area set, $\widehat{\kappa}$, for each element through,

$$\widehat{\kappa}(a) = \sum_{j=1}^{a} \widehat{\Omega}(a) \tag{6}$$

The $\overline{\chi}_a$ of an element whose corresponding cumulative area, $\widehat{\kappa}_a$ was nearest to A_1 was assigned as a threshold, $\overline{\chi}_{del}^{th}$. Undeleted elements with $\overline{\chi}_a$ below $\overline{\chi}_{del}^{th}$ were deleted. The was achieved by re-associating a predefined void property to them with an insignificant Young's modulus (1% of that of undeleted elements). A similar process was then applied to pre-existing deleted elements. The $\overline{\chi}_a$ of the element whose cumulative area, $\widehat{\Omega}(a)$, approximately equaled $A_1 - A_i$ was assigned to a different threshold, $\overline{\chi}_{add}^{th}$. Elements with $\overline{\chi}_a$ above this threshold were reassociated to the existing property for undeleted elements. The mesh was then improved through the AMS detailed in the next section.

Adaptive Meshing

The AMS was basically composed of selecting, refining, coarsening and smoothening. We define the following sets to enhance the description of the AMS:

- $\widehat{U}$ = {All elements in design domain},
- $\widehat{Q}$ = {Undeleted elements connected to boundary edges},
- $\widehat{A}$ = {Elements in $\widehat{Q}$ whose A exceed a lower limit, A_{min} },
- $\widehat{R}$ = {Elements to refine},
- $\widehat{N}$ = {New elements derived from $\widehat{R}$},
- $\widehat{O}$ = {Old elements,}
- $\widehat{C}$ = {Elements to delete by collapsing (thereby coarsening the mesh).}
- $\widehat{X}$ = {Points whose X Cartesian coordinates lie on the design domain boundary},
- $\widehat{Y}$ = {Points whose Y Cartesian coordinates lie on the design domain boundary},

The first stage in the AMS was to select the elements at the edeges of the current topology that would be subject to re-meshing. A connectivity matrix, L, was constructed to contain the total number of elements in the mesh, h. Each row, a in L held the node identification numbers, b, of nodes connected to element, a. An edge matrix $\overline{E}$ with $3h$ rows and 2 columns was then constructed from L through

$$\overline{E}_{i,k} = L_{x,y} \tag{7}$$

where,

$$x = \begin{cases} i - \lfloor \frac{i}{h} \rfloor, \ if & i - \lfloor \frac{i}{h} \rfloor \neq 0 \\ h \quad , \ if & i - \lfloor \frac{i}{h} \rfloor = 0 \end{cases}, \quad y = \begin{cases} \lceil \frac{i}{h} \rceil + (k-1), \ if & \lceil \frac{i}{h} \rceil < 3 \\ 1 \qquad , \ if & \lceil \frac{i}{h} \rceil = 3 \end{cases}, \quad 1 \leq i \leq 3h, \ \texttt{and} \ 1 \leq k \leq 2$$

Each row in $\overline{E}$ held node numbers connected to each edge. Subsets occurring only once represent edges lying on the boundary of the domain. These subsets were placed in a matrix $\overline{E}_1$. The set of elements, $\widehat{Q}$, connected to these edges was then determined by

$$\widehat{Q} = \{a : \overline{E}_1^j \subseteq L_{a,(1,2,3)}\} \tag{8}$$

where j is an integer used to select edges in $\overline{E}_1$ for each element, a. The size of $\widehat{Q}$ was further reduced by subjecting it to two further criteria. The first involved a lower area limit, A_{min}, imposed on members of $\widehat{Q}$ so that only elements with area, A, above A_{min} were passed to a new set $\widehat{R}$. Secondly, $\widehat{R}$ was reduced by removing elements whose A_a falls below a certain percentile (90%). These two steps prevented an infinite growth in the number of elements.

As mentioned earlier, Engwirda's [28] method was employed for refinement purposes. This replaced elements in $\widehat{R}$ with new smaller elements based on two refinement templates. This was achieved by first storing edges of elements to refine in $\overline{E}_1$. The mid point of these edges were computed and including in the mesh. Elements with all three edges belonging to $\overline{E}_1$ were divided into four elements, while elements with a single edge in $\overline{E}_1$ were divided into two on the edge found in $\overline{E}_1$. A much detailed description of this algorithm can be found in [28]. Elemental properties of refined elements were inherited by corresponding new elements. After refinement, $\widehat{O}$ was constructed from $\widehat{N}$

$$\widehat{O} = \widehat{N}^c = \widehat{U} \setminus \widehat{N} \tag{9}$$

Construction of $\widehat{C}$ was accomplished by extracting the smallest 1% of elements in $\widehat{O}$. This lower range was appropriate to prevent excessive distortion of elements in the mesh. Also, elements lying on the boundaries of the design domain were removed from $\widehat{C}$. The $\widehat{C}$ and the mesh were then passed to the coarsening subroutine where the element collapsing [30] operation was performed. The flow chart for this subroutine is shown in Fig. 1. Set $\widehat{C}$ was iteratively checked until it was empty which initiated the termination of the coarsening operation. However, a populated $\widehat{C}$ caused a move into a loop where a restoration point was initiated by duplicating the current mesh into N_{mesh}. The n^{th} edge of the first element in $\widehat{C}$ was then collapsed in N_{mesh} by setting both the first and second nodes of the edge to the mid point, P_{mid}, of that edge. This operation caused a degeneration in elements connected to the edge

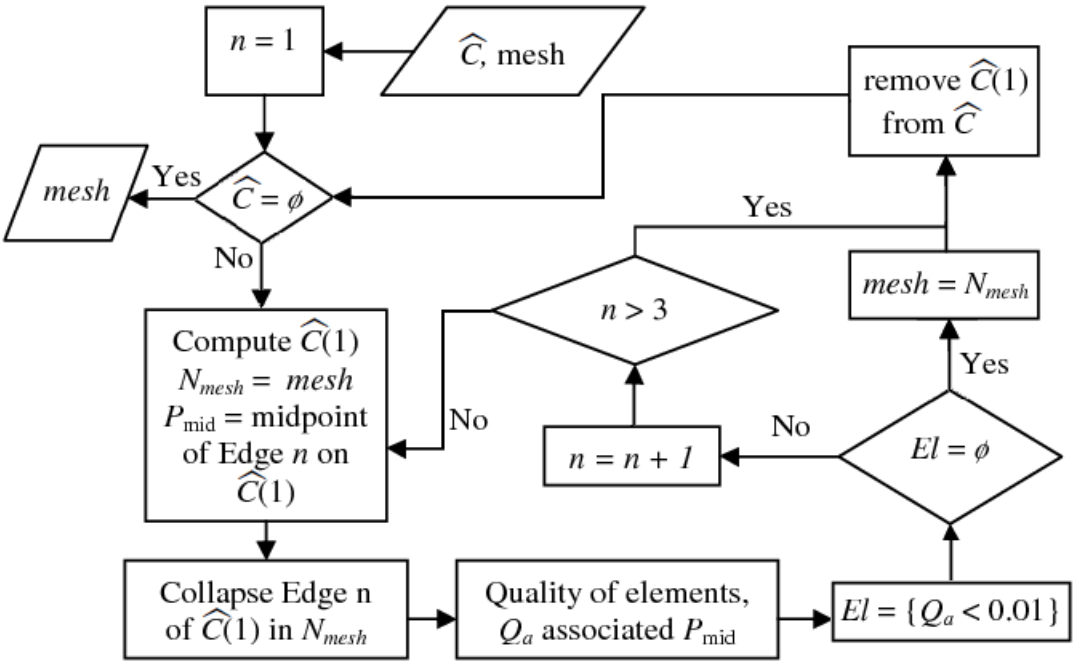

Figure 1: Flow chart for coarsening subroutine based on element edge collapse.

and an enlargement of other elements connected to just one node of same edge. Degraded elements were then simply removed from N_{mesh} without compromising it's topology. The quality, Q_a, of elements in the vicinity of the deleted edge was evaluated according to Bhatia and Lawrence [29] metrics expressed as

$$Q_a = \frac{4\sqrt{3}A_a}{l_1^2 + l_2^2 + l_3^2} \tag{10}$$

where l_1, l_2, l_3 are the element edge lengths and $0 \leq Q_a \leq 1$. For degenerate elements Q_a approaches a value of zero while regular elements were characterized by a Q_a value of one. The variable *mesh* was replaced by N_{mesh} only if low quality elements ($Q_a < 0.01$) were absent from N_{mesh}. Otherwise, a move is forced to the next edge and the collapsing process repeated. An element was removed from $\widehat{C}$ once collapsed or if no collapsing operation for its three edges resulted in adjacent elements of acceptable quality. Further improvement in the quality of *mesh* was achieved by Laplacian smoothening [31]. Convergence of the Laplacian smoothening algorithm was based on the quality metric stated earlier with a minimum quality of 0.05. The mesh was passed back to BESO after each improvement and convergence tested by calculating ΔC

$$\Delta C = \frac{|\sum_{k=1}^{T}(C_{i-k+1} - C_{i-T-k+1})|}{|\sum_{k=1}^{T} C_{i-k+1}|} \tag{11}$$

where $T = 5$, k is a counter varying from 1 to T, used to selectively pick C for the last T iterations. Variable ΔC was compared against θ to either terminate or repeat the BESO algorithm. θ assumed a value of 1×10^{-3}. We hypothesize that coupling our AMS with the BESO algorithm would provide greater efficiency, reduced sensitivity to the starting mesh and smoother boundaries thereby reducing subsequent geometric post processing. We proceed to give a detailed account of experiments conducted to investigate these hypotheses by solving a cantilever plate problem to minimize C.

Test Problem

To test the performance of the AMS, we solved a cantilever plate problem (Fig. 2) with an objective to minimize C subject to an area fraction constraint, A^* of 0.5. A Young's modulus of 100GPa and Poisson's ratio of 0.3 was assumed.

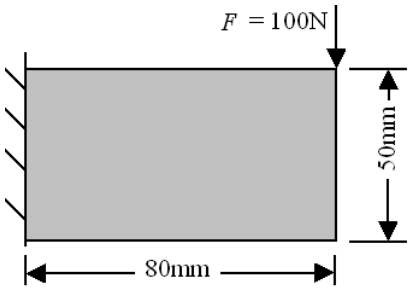

Figure 2: Cantilever plate subject to a point load, F, and constrained on the left side.

We investigated the validity of our hypotheses with two difference set of experiments. Using different mesh sizes we compared our strategy with the original BESO [10]. The $R^{'}$ and v were fixed at 1.5 and 1% respectively for all experiments. It was anticipated that by selecting these values for the optimization parameters, penalization of finer features would be minimized. Experiments were performed on a desktop computer with Intel(R) Core(TM) CPU, 3.20GHz and 3.24GB of RAM. Where the results are labelled 'BESO', this refers to Huang and Xie's version [10].

Test of Efficiency

Four experiments were performed to ascertain the computational efficiency for both the AMS and BESO algorithm (table 1). The starting mesh size and area limit, A_{min}, were varied across these experiments. An initial value of $0.1mm^2$ was assumed for A_{min}, while solving the problem with the AMS algorithm with 2000 elements initially in the design domain (Exp.1). A corresponding uniform mesh was created with elemental areas equal to $0.1mm^2$ (40,000 elements) was utilized in Exp. 2 where the BESO algorithm was applied. Two other experiments were performed using a lower value of A_{min} ($0.02mm^2$). Starting with a mesh populated with 40,000 elements, the first experiment was repeated (Exp. 3). The corresponding uniform mesh for this value of A_{min} amounted to 200,000 elements in Exp. 4 where the BESO was applied. Optima truss-like topologies for these four experiments are shown in Fig. 3.

Table 1: Optimization run details and results for Exps. 1 to 4 comparing CPU time used, iterations to convergence and converged strain energy, C_∞ and stiffness, K_∞.

Exp.	Algorithm	Start No. elems.	A_{min} (mm^2)	Time (hr:min)	No. iterations	C_∞ (Nmm)	K_∞ (N/mm)
1	AMS	2,000	0.1	0:18	75	2.21	27.11
2	BESO	40,000	0.1	0:39	78	2.21	26.85
3	AMS	40,000	0.02	1:42	77	2.25	27.03
4	BESO	200,000	0.02	9:39	78	2.28	27.00

It can be seen that the boundaries of the topologies shown in Figs. 3a and 3c appear darker than the interior regions. A magnified view B shown in these Figs. allows a better visualization of the mesh. This is due to the higher density of smaller elements at these boundaries, in contrast to an even shade

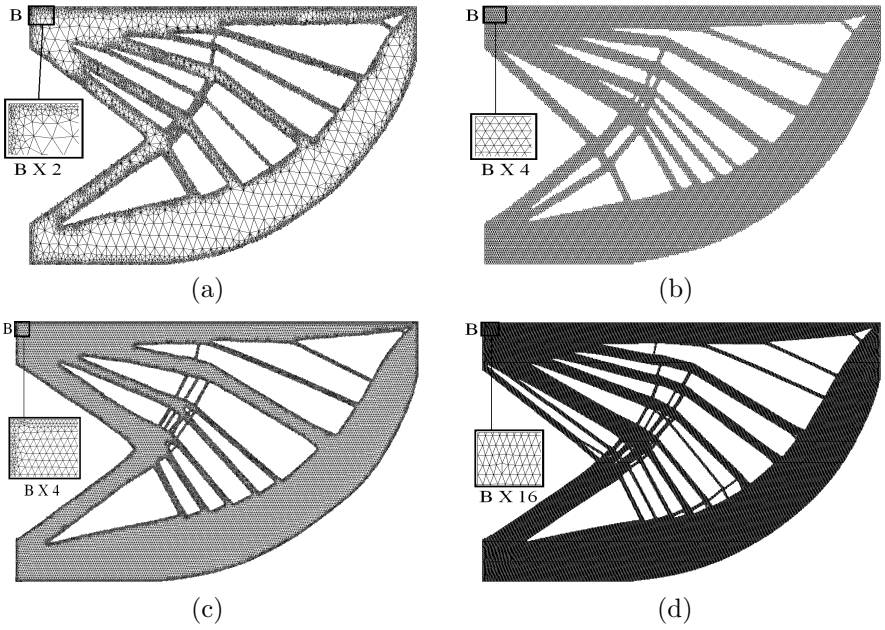

Figure 3: Optima topologies and a magnified view B for (a) Exp.1 (b) Exp. 2 (c) Exp. 3 (d) Exp. 4

observed in Figs. 3b and 3d due to a near constant element size. We characterize each experiment by the CPU time consumed, number of iterations to convergence and the asymptotic strain energy, C_∞. Results for these measures are contained in table 1.

Exps. 1 and 2 converged to the same C_∞, however Exp. 2 took twice as long as Exp. 1, also requiring a greater number of iterations. A similar trend is repeated for Exp. 3 and Exp. 4, though Exp. 3 converged approximately five times faster than Exp. 4. Also while Exp. 1 and 2 attained the same C_∞, Exp. 3 and 4 differed slightly. Similarity in C_∞ for Exp. 1 and 2 can be attributed to approximately the same number of nodes at the end of both Exps. as shown in Fig. 4a.

The number of nodes increased in Exp. 1 as the TO progressed to end with a similar number of nodes to Exp. 2. However, Exp. 3 and Exp. 4 have a significantly different number of nodes at the last iteration as shown in Fig. 4b. The different number of degrees of freedom makes it difficulty to fairly compare these C_∞ values since different levels of FEA errors will exist in the calculation of C. Distortions at the loaded node compound this problem, necessitating a post analysis. Since the main aim of a strain energy minimization problem is to minimize deflection, the post analysis was performed to determining the stiffness of the topologies. Stiffness in this context refers to the ratio between load F and the magnitude of a nodal displacement. The chosen node was in the vicinity of the loaded node, however far enough to avoid FEA errors caused by distortion. The location of this node was fixed for all topologies.

Boundaries of topologies were first extracted, followed by a smoothening process to eliminate the rough boundaries without violating V^*. All, topologies were then remeshed with second order elements and then analyzed. The number of elements was increased until a converged stiffness, K_∞ was reached. The K_∞ for the four topologies are shown in table 1. The K_∞ for Exp. 1 was slightly greater (0.01%) than that of Exp. 2. Also, K_∞ of Exp. 3 was slightly greater (0.001%) than that of Exp. 4. This suggested topologies achieved by the AMS were marginally stiffer than those using the BESO algorithm. While it could be argued that the K_∞ values of Exp. 1 to 4 were approximately equal, the computational efficiency offered by the AMS makes it attractive for TO for AM.

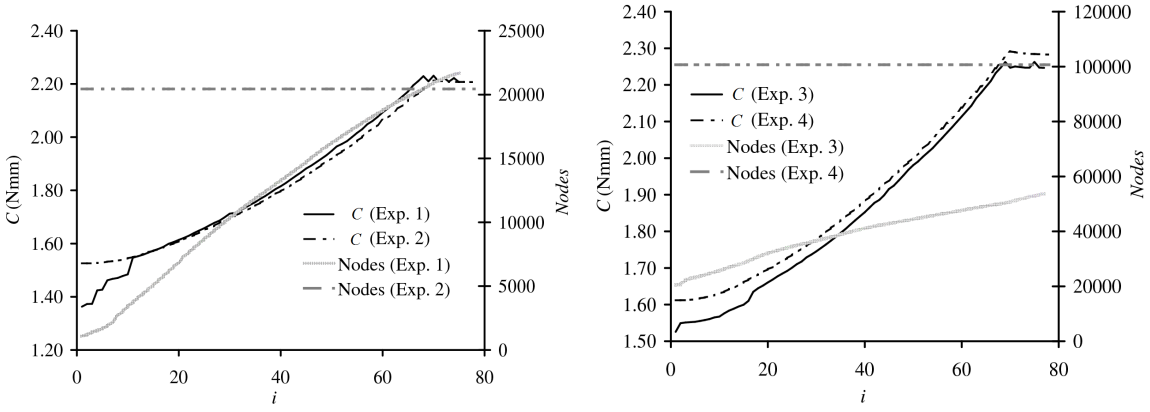

Figure 4: Plots of C and nodes against i for (a) Exps. 1 and 2 (b) Exps. 3 and 4

Sensitivity Test

A second set of experiments was performed to compare the sensitivity of the AMS to the starting mesh compared to that of the original BESO. Eight experiments were formulated for this purpose (table 2), the first four were performed with the AMS and the second four using the BESO algorithm. The A_{min} was set at a low value of $0.001mm^2$ for the AMS experiments while the initial mesh was populated with a different number of elements for the four experiments. The four experiments using the BESO algorithm had different fixed meshes corresponding to the initial meshes of the AMS experiment. Optima the topologies for these experiments are shown in Fig. 5. Again, these topologies appear truss-like, noting that Exps. 11 and 12 correspond to Exps. 2 and 4 respectively.

Table 2: Sensitivity test of AMS and BESO algorithms involving eight experiments (Exps. 5 to 12) characterized by their M, D_{min}, L_{min}, C_∞ and K_∞. Standard errors, σ_m, are also shown.

Exp.	Algorithm	Start No. elems.	A_{min} (mm^2)	M (mm)	D_{min} (mm)	L_{min} (mm)	C_∞ (Nmm)	K_∞ (N/mm)
5	AMS	2,000	0.001	28	0.36	0.37	2.25	27.11
6	AMS	20,000	0.001	26	0.38	0.42	2.29	27.18
7	AMS	40,000	0.001	31	0.23	0.38	2.31	26.99
8	AMS	75,000	0.001	28	0.34	0.32	2.30	27.10
			σ_m:	1.03	0.03	0.02		0.04
9	BESO	2,000	-	10	3.64	10.66	2.09	26.23
10	BESO	20,000	-	24	1.66	1.20	2.18	26.79
11	BESO	40,000	-	25	0.38	0.38	2.21	26.84
12	BESO	75,000	-	35	0.74	0.64	2.23	26.94
			σ_m:	5.14	0.73	2.48		0.32

It was difficult to ascertain the sensitivity of the AMS and the BESO to initial mesh by simple visual inspection of the topologies. Therefore, these topologies were quantified with the number of strut members, M, occurring in the optima (table 2), the minimum member thickness, D_{min}, the minimum member length, L_{min} and K_∞. For both the AMS and BESO algorithm, variability in each sample set was quantified with the standard error in the mean, σ_m, of these characteristics stated as,

$$\sigma_m = \frac{\sigma}{\sqrt{4}} \tag{12}$$

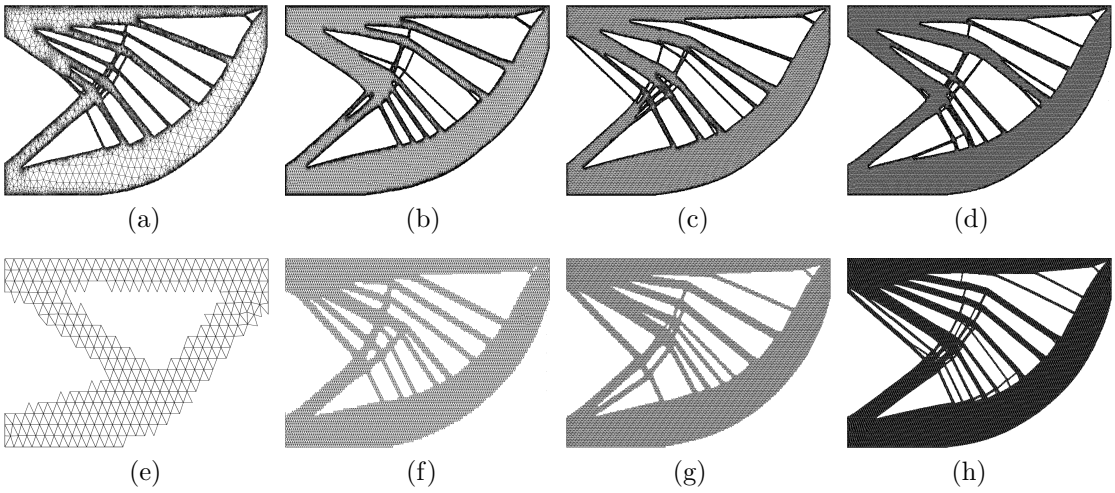

Figure 5: Optima topologies for (a) Exp. 5 (b) Exp. 6 (c) Exp. 7 (d) Exp. 8 (e) Exp. 9 (f) Exp. 10 (g) Exp. 11 (h) Exp. 12

where σ was the standard deviation of each sample set. The σ_m for both algorithms can be seen in table 2. These values show that there is less variability in the topological data collected for the AMS than with the BESO algorithm. Secondly, it was observed that the AMS realized members with smaller D_{min} and L_{min} than the BESO. Attaining these dimensions with the BESO, or any other fixed mesh algorithm, would require a much finer mesh, and have high computational time. These finer details are important as we are now able to produce part with such details via AM. A typical limit of minimum feature size in AM is about $10\mu m$ [32]. With regards to C_∞, these could not be reliably used to estimate variability in the performance for the reasons mentioned in the previous section. A useful measure was the standard error in the K_∞, which was also observed to be less for the AMS than that of the BESO as shown in table 2. It should also be noted that Figs. 3a and 3b, Figs. 3c and 3d shared similar boundary roughness since the size of boundary elements in both sets were largely similar. However, Figs. 3a-3d had smoother boundaries Figs. 3e-3h since elements at their boundaries were much smaller ($0.001mm^2$) than that of the uniform meshes used in Figs. 3e-3h. This demonstrates that the AMS is able to achieve smoother boundaries than the BESO algorithm at a lower computation cost.

Conclusion and further work

A hybrid TO algorithm is proposed constituted by a modified form of the BESO algorithm and an adaptive meshing strategy (AMS). The BESO algorithm was modified so that it was based on strain energy densities since the AMS caused a progressive variation in elemental sizes. A two dimensional plate problem was solved to benchmark performance of the hybrid against that of the BESO algorithm. The AMS was found to be substantially more efficient, with a reduction in computation time of 52% and 82% respectively for the cases investigated, with similar performance. It was also less sensitive to the starting mesh and able to attain smoother boundaries with a lower number of elements. Also, finer features are achievable with the AMS, starting with relatively coarse meshes. This is particularly interesting for topologies to be made via additive manufacture (AM). Since AM can achieve parts with high degrees of complexity. Further work could involve the determination of an element size that corresponds to AM's resolution. Also, the AMS could be extended to practical three dimensional parts experiencing single and multiple load cases.

Acknowledgements

The authors are grateful for the funding provided by the IMCRC and EPSRC.

References

[1] Mitchell, A.G.M., *The limits of economy in frame structures*, Philo, Mag. Sec, 6(8), p.589-597 (1904).

[2] Rozvany, G.I.N., *Aims, scope, methods, history, and unified terminology of computer-aided topological optimization in structural mechanics*. Struct Multidisc Optim, 21, p. 19 (2001).

[3] Kirsh, U., *On optimal topologies of grillage structures*, Engineering with computers, 42(8), p.223-239 (1986).

[4] Bendsoe, M.P., Kikuchi, *Generating optimal topologies in structural design using a homogenization method.* Computer Methods in Applied Mechanics and Engineering, 71, p. 28 (1988).

[5] Rozvany, G., I., N., Zhou, M., Birker, T., *Generalized shape optimization without homogenization*, Structural Optimization, 4, p.250-252 (1992).

[6] Xie, Y., M., Steven, G.P., *Evolutionary Structural Optimization*, 188p., Springer-Verlag, London (1997).

[7] Zhou, M., Rozvany, G.I.N., *On the validity of ESO type methods in topological optimization.* Struct Multidisc Optim, 21, p. 4 (2001).

[8] Mattheck, C., Burkhardt, S., Erb, D., *Shape optimization of engineering components by adaptive biological growth.* In: Eschenauer, H., A.; Mattheck, C.; Olhoof, N. (eds) Engineering optimization in design processes, pp.15-26. Berlin, Heidelberg, New York, Springer, 1991.

[9] Querin, O., M., Young, V., Steven, G., P., Xie, Y., M., *Computational Efficiency and validation of bi-directional evolutionary structural optimization*, Comput Methods Applied Mechanical Engineering, 189, p.559-573 (2000b).

[10] Huang, X., Xie, Y., M., *Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method*, Finite Elements in Analysis and Design, 43, p.1039-1040 (2007).

[11] Sandgren, E., Jensen, E., Welton, J., *Topological design of structural components using genetic optimization methods*, in Sensitivity Analysis and Optimization with Numerical Methods, S. Saigal and S. Mukherjee, eds., Proceedings of the Winter Annual Meeting of the American Society of Mechanical Engineers 115, pp. 31(43), Texas, 1990.

[12] Chapman, C., D., Saitou, K., Jakiela, M., J., *Genetic algorithms as an approach to configuration and topology design*, Journal of Mechanical Design, 116(105), 8p. (1994).

[13] Chapman, C., D., Jakiela, M., J., *Genetic algorithm-based structural topology design with compliance and topology simplification consideration*, Journal of Mechanical Design 118(89), 10p. (1996).

[14] Kaveh, A., Hassani, B., Shojaee, S., Tavakkoli, S., M., *Structural topology optimization using ant colony methodology*, Engineering Structures, 30, p.2559-2566 (2008).

[15] Luh, G., Lin, C., *Structural topology optimization using ant colony optimization algorithm*, Applied Soft Computing, 9, p.1343-1353 (2009).

[16] Luh, G., Lin, C., Lin, Y., *A binary particle swarm optimization for continuum using topology optimization*, Applied Soft Computing, 11, p.2833-2844 (2011).

[17] Bureerat, S., Limtragool, J., *Structural topology optimization using simulated annealing with multiresolution design variables*, Finite Elements in Analysis and Design, 44, p.738-747 (2008).

[18] Zuo, Z.H., Xie, Y.M., Huang X., *Combining genetic algorithms with BESO for topology optimization.* Struct Multidisc Optim, 38, p. 13, (2009).

[19] Garcia-Lopez, N., P., Sanchez-Silva, M., Medaglia, A., L., Chateauneuf, A., *An improved hybrid topology optimization approach coupling simulated annealing and SIMP*, IOP Conf. Series: Materials Science and Engineering, 10, WCCM/APCOM, (2010).

[20] Heubner, K., H., Dewhirst, D., L., Smith, D., E., Byrom, T., G., *The finite element method for engineers*, John Wiley and Sons, 4^{th} ed, New York, (2001).

[21] Kuo, Y., L., Cleghorn, W., L., Behdinan, K., Fenton, R., G., *The h-p-r-refinement finite element analysis of a planar high-speed four-bar mechanism*, Mechanism and Machine Theory, 41, p.505-524 (2006).

[22] Kim, H., Garcia, M., J., Querin, O., M., Steven, J., P., Xie, Y., M., *Introduction of fixed grid in evolutionary structural optimization*, Engineering Computation, 17(4), p.427-439 (2000).

[23] Mariano, V., Marti, P., M., Querin, O., M., *Topology design of two-dimensional continuum structures using isolines*, Computer and Structures, 87, p.101-109 (2009).

[24] Gibson, I., Rosen, D., W., Stucker, B., *Additive Manufacturing Technologies*, 459p., Springer, New York (2009).

[25] Tuck, C.J., Hague, R., J., M., Ruffo, M., Ransley, M., Adams, P., *Rapid manufacturing facilitated customization*, International Journal of Computer Integrated Manufacturing, 21(3), p.245-258 (2008).

[26] Hague, R., Mansour, S., Saleh N., *Design opportunities with rapid manufacturing*, Assembly Automation, 23(4), p.346-356 (2003).

[27] Rosen, D., W., *Computer-Aided design for additive manufacturing of cellular structures*, Computer-Aided design and applications, 4(5), p.585-594 (2007).

[28] Engwirda, D., *Unstructured Mesh Methods for the Navier Stokes Equations*, Undergraduate thesis, The school of Aerospace Engineerinf, The University of Sydney, 2005.

[29] Bhatia, R., P., Lawrence, K., L., *Two-dimensional finite element mesh generation based on stripwaise automatic triangulation*, Computer and structures, 36, p.309-319 (1990.)

[30] Zhigeng, P., Kun, Z., Jiaoying, S., *A New mesh simplification algorithm based on triangle collapses*, 16(1), p.8 (2001).

[31] Glen, A., Hansen, R., W., Douglas, A., Z., *Mesh enhancement: Selected elliptic methods, foundations and applications*, Imperial College Press, 515p., London, 2005.

[32] Cohen, A., Chen, R., Frodis, U., Wu, M., Folk, C., *Microscale metal additive manufacturing of multi-component medical devices*, Rapid Prototyping Journal, 16(3), p.209-215 (2010).