Hybrid Tabu Algorithm for the Synthesis and Fabrication of Fiber Bragg Gratings

Nam Quoc Ngo[1] and Ruitao Zheng[2]
[1]*Nanyang Technological University*
[2]*Avago Technologies Manufacturing (Singapore) Pte. Ltd.*
Singapore

1. Introduction

Fiber Bragg gratings (FBGs) are widely employed as optical filters for performing various functions such as add/drop multiplexers, dispersion compensators and multiplexers/demultiplexers for use in optical communication systems and optical sensors because of a number of advantages that include low insertion loss, low polarization sensitivity, all-fiber geometry, compactness, easy fabrication and low cost (Kashyap, 1999; Othonos & Kalli, 1999). In addition, the technology of ultraviolet (UV) photoinduced FBGs is quite mature to allow the fabrication of a wide variety of FBGs with complex characteristics. To meet the increasing demand for large capacity of the next generation of optical communication systems (i.e., wavelength division multiplexing (WDM) networks) and optical sensors, there is an important need for a powerful design tool that can be used for the synthesis or design of FBGs from the specified frequency responses that can be practically realized. This synthesis or inverse problem of determining a FBG structure from a given frequency response (i.e., magnitude and phase responses) is common in many application areas. The design tool must be efficient and reliable to enable the synthesis of FBG-based filters with prescribed frequency responses, depending on the application requirements. That is, the design tool must be able to determine the index modulation profile and hence the structure of an FBG from a given frequency response. In addition, the design tool must be powerful enough for use in the diagnosis or characterization during and after fabrication of an FBG. Although several synthesis methods such as those based on the layer-peeling algorithm have been proposed for the synthesis of FBGs from the specified frequency responses, the index modulation profiles of the synthesized FBGs are not optimized and are often complex, making practical realization difficult (Feced et al., 1999; Poladian, 2000; Skaar et al., 2001; Rosenthal & Horowitz, 2003). To overcome this problem, optimization methods have been demonstrated as an attractive approach because it allows weighting mechanisms to be incorporated into the desired frequency response of an FBG-based filter to be synthesized, resulting in an optimum and practically realizable index modulation profile of the FBG structure. Furthermore, the optimization algorithms also allow additional constraints to be included in the weighting mechanisms of the specified frequency response to suit certain condition(s) or constraint(s) of a particular fabrication system. In optimization, the FBG synthesis problems are formulated as nonlinear objective

Source: Local Search Techniques: Focus on Tabu Search, Book edited by: Wassim Jaziri, ISBN 978-3-902613-34-9, pp. 278, October 2008, I-Tech, Vienna, Austria

functions and the optimized solution of the FBG design is obtained by finding the global optimum of the objective function. Although several global optimization algorithms such as the global genetic algorithm (GA) (Skaar & Risvik, 1998; Gill et al., 2004; Cheng & Lo, 2004), the global simulated annealing (SA) (Dong, Azana & Kirk, 2003) and the local optimization method such as the Levenberg-Marquardt algorithm (Plougmann & Kristensen, 2004) have been applied to solving the FBG synthesis problems, the obtained solutions are, in general, not optimum, making practical implementation difficult. This is because, in general, it is important to employ a global optimization algorithm to solve an FBG synthesis problem to ensure that a global optimum can be obtained; however, convergence of the global optimization method is normally not as good as that of a local optimization algorithm. Unfortunately local optimization methods also cannot easily solve the FBG synthesis problems due to the multimodal and ill-conditioned character of the nonlinear objective functions. To improve the convergence of the global optimization algorithm, a hybrid algorithm combining a global optimization algorithm and a local optimization algorithm has been shown to be a better approach for the synthesis and fabrication of FBG-based bandpass filters (Zheng et al., 2004; Ngo et al., 2007). This is the motivation of this chapter which describes the use of a hybrid Tabu algorithm for the synthesis and fabrication of FBG-based bandpass filters and linear phase filters from the given frequency responses. (Ngo et al., 2004; Zheng et al., 2005) presented the first reports of employing the standard and improved Tabu search algorithms (Glover & Laguna, 1998; Chelouah & Siarry, 2000) for the synthesis of FBG-based bandpass filters and linear phase filters. The hybrid Tabu algorithm is a two-tier search that employs a global optimization algorithm (i.e., a staged continuous Tabu search (SCTS) algorithm (Zheng et al., 2005) which performs better than the standard Tabu search (Ngo et al., 2004)) and a local optimization algorithm (i.e., the Quasi-Newton method (Shanno, 1970) which has high efficiency in solving multimodal nonlinear optimization problems). First, the global SCTS algorithm, in which a dynamic mechanism for weighting of different requirements of the magnitude and phase responses is employed to enhance the optimization efficiency, is used to find a “promising” FBG structure that has a frequency response as close as possible to the target one. The local Quasi-Newton algorithm is then applied to further optimize this “promising” FBG structure obtained from the global SCTS algorithm to obtain the final optimum solution. To demonstrate the effectiveness of the hybrid Tabu method which has high convergence rate and high reliability, the synthesis and fabrication of several FBG-based bandpass filters and linear phase filters for application in optical communications are presented in this chapter. It is worth mentioning that other stochastic search algorithms of various types have also been applied to the design of FBGs: the Nelder-Mead Simplex hill climbing algorithm (Caucheteur et al., 2004) and particle swarm optimization (Baskar et al., 2005a), evolutionary strategies such as the covariance matrix adapted evolution (Baskar et al., 2005b; Baskar et al., 2006), and a multi-objective evolutionary algorithm (Manos & Poladian, 2005).

2. Transfer matrix method for solving non-uniform Bragg gratings

2.1 Theory of fiber Bragg grating

There are two main types of fiber Bragg gratings (FBGs), namely, uniform FBG (which has equal grating periods) and non-uniform FBG (which has unequal grating periods). Single-moded FBGs are considered here because they are commonly used in many areas of optics and photonics. For ease of discussion, the uniform FBG is considered in this section. An FBG is essentially a filter written into the core of a segment of optical fiber via the interference of two ultraviolet (UV) beams from a UV laser (see Fig. 1). The interference pattern forms a periodic refractive index change (or index perturbation) along the longitudinal direction of the fiber. Due to the index change, the FBG acts as a series of reflectors, reflecting back a small amount of the input light with wavelength components that are close to the Bragg wavelength.

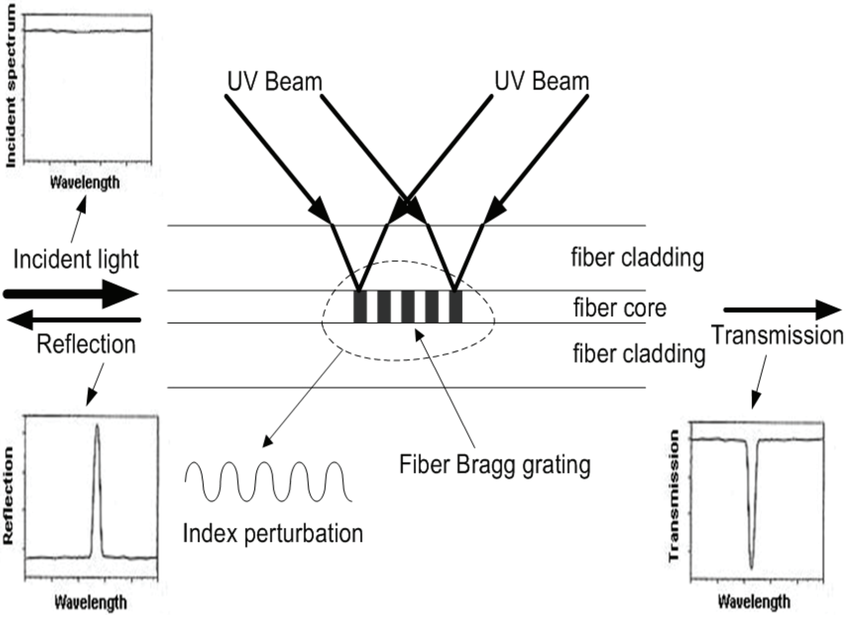

Fig. 1. Schematic diagram of a typical interferometric system used for the fabrication of fiber Bragg gratings (FBGs). The FBG shown here is of a uniform type with equal grating periods.

The Bragg wavelength, $\lambda_B(z)$, of a uniform FBG is the wavelength that fulfills the Bragg condition:

$$\lambda_B(z) = 2\bar{n}_{\text{eff}}(z)\Lambda(z) \tag{1}$$

where z is the longitudinal coordinate along the length of the grating and $\Lambda(z)$ is the perturbation period or grating period (see Fig. 2). The average effective refractive index on the grating is defined as (see Fig. 2)

$$\bar{n}_{\text{eff}}(z) = n_0 + \Delta n_{\text{dc}}(z) \tag{2}$$

where n_0 is the effective index without UV exposure and $\Delta n_{\text{dc}}(z)$ is the "dc" (or average) index change spatially averaged over the grating. Fig. 2 shows the quasi-sinusoidal profile of the effective index, $n_{\text{eff}}(z)$, which is described by

$$n_{\text{eff}}(z) = \bar{n}_{\text{eff}}(z) + \Delta n_{\text{ac}}(z) \cdot f(z) \cdot \cos\left[\frac{2\pi z}{\Lambda(z)} + \phi(z)\right] \tag{3}$$

where $\Delta n_{\text{ac}}(z)$ is the "ac" index change (i.e., index modulation), $f(z)$ is the normalized apodization function, and $\phi(z)$ is the chirp profile of the grating. Putting Eq. (2) into Eq. (3), the index perturbation, $\delta n_{\text{eff}}(z)$, is given by

$$\delta n_{\text{eff}}(z) = n_{\text{eff}}(z) - n_0 = \Delta n_{\text{dc}}(z) + \Delta n_{\text{ac}}(z) \cdot f(z) \cdot \cos\left[\frac{2\pi z}{\Lambda(z)} + \phi(z)\right] \tag{4}$$

Thus Eq. (4) shows that the optical properties of an FBG are essentially determined by the variation of the index perturbation along the grating length.

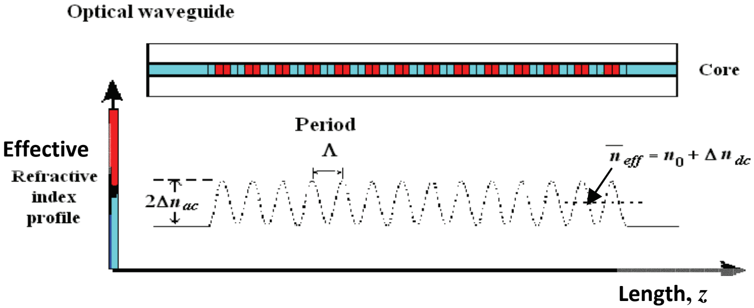

Fig. 2. Schematic showing an example of an FBG and its refractive index profile. The FBG shown here is of a uniform type with equal grating periods.

Figure 3 shows the index perturbation profiles of a uniform FBG, an apodized FBG with variable-dc index change, and an apodized FBG with zero-dc index change, which are considered in this work. Other types of index perturbation profiles such as chirped FBG, phase-shifted FBG and super-structured FBG can also be employed, depending on the desired filter responses (Erdogan, 1997).

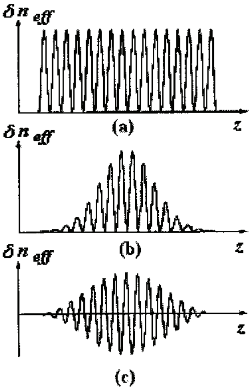

Fig. 3. Index modulation profiles of the types of FBGs considered here. (a) A uniform FBG with a constant dc index change. (b) An apodized FBG with variable-dc index change. (c) An apodized FBG with zero-dc index change.

The length of the FBG typically ranges from several millimeters to tens of centimeters, depending on the desired filter responses. Because there are tens of thousands of these perturbation periods of index changes or reflectors in a row, an FBG-based filter generally has a near-squared reflective magnitude response. In the reflection mode, the reflective magnitude response has a very narrow bandpass response (with a typical 3-dB bandwidth of 0.2 nm or 25 GHz in the 1550 nm wavelength window) at the Bragg wavelength; whereas in the transmission mode, the transmissive magnitude response has a very narrow notch response at the Bragg wavelength (see Fig. 1). The phase-mask technique is one of the most effective methods used in the fabrication of FBGs because it employs a simple diffractive optical element (or a phase mask) to spatially modulate the UV beam (Kashyap, 1999; Othonos & Kalli, 1999). Compared with two other main types of optical filters, namely, thin film filters and arrayed waveguide gratings, FBG-based filters have many unique advantages such as low loss, low polarization sensitivity, simple all-fiber geometry, easy fabrication, and low cost.

2.2 Transfer matrix method for solving non-uniform fiber Bragg gratings

The coupled-mode theory has been widely used for the analysis of FBGs because it allows one to determine the effect of the grating structure on the frequency response (i.e., magnitude and phase responses) (Erdogan, 1997; Hill & Meltz, 1997; Kashyap, 1999; Othonos & Kalli, 1999). A uniform FBG is the simplest type of FBG to design and fabricate because it simply has equal grating periods and a constant modulation depth of the refractive index (see Fig. 1, Fig. 2, Fig. 3(a)). The uniform FBG can be easily designed because an analytical solution to the coupled-mode equation can be easily obtained. However, due to the finite length of the uniform FBG, the roll offs on the two edges of the reflective magnitude response are not sharp enough due to the presence of sidelobes on both sides of the bandpass response. This drawback has limited the application of uniform FBGs. This limitation can be overcome by using a non-uniform FBG which has a more squared reflective magnitude response with much smaller amplitudes of the sidelobes. However, analytical solutions to the coupled-mode equations describing the non-uniform FBGs cannot be easily obtained. The transfer matrix method (TMM) has been widely employed for solving the non-uniform FBGs due to its high computational efficiency and high reliability (Erdogan, 1997; Kashyap, 1999). The TMM method allows the magnitude and phase responses of a non-uniform FBG to be easily obtained with reasonably high accuracy. In TMM, a non-uniform FBG can be modeled using two methods, depending on the requirements of the magnitude and phase responses: (1) A non-uniform FBG model using the cascade of serially-connected *uniform* sub-gratings (this model is sufficient for obtaining the magnitude response when the phase response is not important); and (2) A non-uniform FBG model using the cascade of serially-connected *apodized* sub-gratings (this model must be used when both the magnitude and phase responses are important). These two methods are described below.

2.2.1 A non-uniform FBG model using the cascade of *uniform* sub-gratings

When only the magnitude response of a filter (whose phase response is not important) is required to be designed, the TMM method can be used to model a non-uniform FBG as the cascade of serially-connected *uniform* sub-gratings. The non-uniform FBG can be divided into a number of serially-connected *N uniform* sub-gratings or sections (as shown in Fig. 4).

Each *uniform* sub-grating section can be described by an analytic transfer matrix. The transfer matrix for the entire non-uniform FBG structure can be obtained by simply multiplying the individual transfer matrices of the *uniform* sub-gratings.

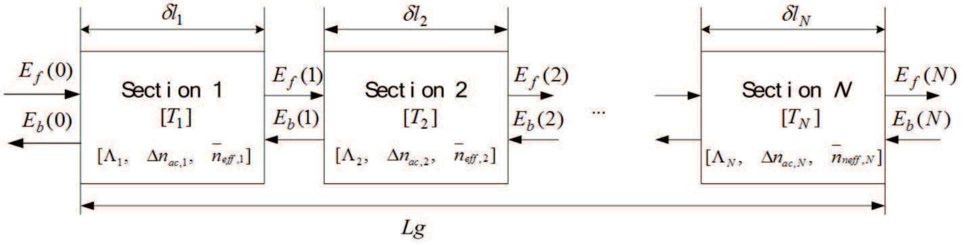

Fig. 4. Schematic diagram of a non-uniform FBG model based on the cascade of serially-connected *uniform* sub-gratings.

Each *uniform* sub-grating section is described by an index perturbation defined in Eq. (4). Note that Eq. (4) is a general equation describing the index perturbation of all kinds of FBGs. To apply Eq. (4) to each *uniform* sub-grating section, the parameters in this equation are re-defined as follows. A *uniform* sub-grating section requires $f(z)=1$ and $\phi(z)=0$ in Eq. (4). Putting these conditions into Eq. (4) gives a simpler index perturbation, $\delta n_{\text{eff}}(z)$, for each *uniform* sub-grating which is given by

$$\delta n_{\text{eff}}(z) = \Delta n_{\text{dc}}(z) + \Delta n_{\text{ac}}(z) \cdot \cos\left(2\pi z / \Lambda(z)\right) \tag{5}$$

Equation (5) is schematically shown in Fig. 3(a). In Fig. 4, $E_{\text{f}}(j;\lambda)$ and $E_{\text{b}}(j;\lambda)$ are the complex electric fields of the forward and backward propagation waves, respectively, describing the j^{th} section. Also, δl_j, Λ_j, $\Delta n_{\text{ac},j}$ and $\bar{n}_{\text{eff},j}$ (and hence $\Delta n_{\text{dc},j}$ according to Eq. (2)), which are the parameters to be optimized in this particular model, are the length, period, 'ac' index change (i.e., index modulation) and average effective index of the j^{th} section, respectively. In this model, the parameters that are not optimized are Λ_j and $\Delta n_{\text{dc},j}(z) = \Delta n_{\text{dc},j}$ (and hence $\bar{n}_{\text{eff},j}$ according to Eq. (2)), which are the period and the "dc" index change, respectively, and they are fixed or constant values. L_{g} is the total length of the non-uniform FBG. The designer has the choice of choosing which of these four (4) variables (i.e., δl_j, Λ_j, $\Delta n_{\text{ac},j}$ and $\bar{n}_{\text{eff},j}$) are to be optimized, depending on the fabrication condition(s) or constraint(s) of a particular fabrication system. The complex electric fields at the input ports ($E_{\text{f}}(0;\lambda), E_{\text{b}}(0;\lambda)$) and output ports ($E_{\text{f}}(N;\lambda), E_{\text{b}}(N;\lambda)$) of the non-uniform FBG are described by

$$\begin{bmatrix} E_{\text{f}}(0;\lambda) \\ E_{\text{b}}(0;\lambda) \end{bmatrix} = T \cdot \begin{bmatrix} E_{\text{f}}(N;\lambda) \\ E_{\text{b}}(N;\lambda) \end{bmatrix}; \qquad T = \prod_{j=1}^{N} T_j \tag{6}$$

where $T_j = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix}$ is the 2 × 2 transfer matrix of the j^{th} section. The elements of the transfer matrix are defined as

$$T_{11} = \cosh(S_j \delta l_j) - \frac{i\hat{\sigma}_j}{S_j} \sinh(S_j \delta l_j), \quad T_{22} = \cosh(S_j \delta l_j) + \frac{i\hat{\sigma}_j}{S_j} \sinh(S_j \delta l_j) \tag{7}$$

$$T_{12} = -\frac{i\kappa_j}{S_j} \sinh(S_j \delta l_j), \quad T_{21} = \frac{i\kappa_j}{S_j} \sinh(S_j \delta l_j) \tag{8}$$

where $i = \sqrt{-1}$. The 'ac' coupling coefficient, κ_j , is defined as

$$\kappa_j = i\frac{\pi}{\lambda} \Delta n_{\mathrm{ac},j} \tag{9}$$

where λ is the optical wavelength. The 'dc' coupling coefficient, $\hat{\sigma}_j$, is defined as

$$\hat{\sigma}_j = \delta_j + \sigma_j \tag{10}$$

where $\delta_j = \beta_j - \pi/\Lambda_j = 2\pi \bar{n}_{\mathrm{eff},j} \left(1/\lambda_j - 1/\lambda_{\mathrm{B},j} \right)$ is the detuning parameter at the wavelength λ_j, β_j is the propagation constant, $\lambda_{\mathrm{B},j} = 2\bar{n}_{\mathrm{eff},j}\Lambda_j$ is the Bragg wavelength, and $\bar{n}_{\mathrm{eff},j} = n_0 + \Delta n_{\mathrm{dc},j}$ (see Eq. (2)). The coefficient σ_j is defined as

$$\sigma_j = \frac{2\pi}{\lambda_j} \Delta n_{\mathrm{dc},j} \tag{11}$$

where $\Delta n_{\mathrm{dc},j}$ is given in Eqs. (2) and (4). S_j in Eqs. (7) and (8) is defined as

$$S_j^2 = \kappa_j^2 - \hat{\sigma}_j^2 \tag{12}$$

Applying the boundary condition $E_{\mathrm{b}}(N;\lambda) = 0$ (i.e., there is no input to the right side of the FBG), the reflection frequency response $\rho(\lambda)$ and the transmission frequency response $t_x(\lambda)$ are given by

$$\rho(\lambda) = \frac{E_{\mathrm{b}}(0;\lambda)}{E_{\mathrm{f}}(0;\lambda)} \tag{13}$$

$$t_x(\lambda) = \frac{E_{\mathrm{f}}(N;\lambda)}{E_{\mathrm{f}}(0;\lambda)} \tag{14}$$

The reflection magnitude response and the transmission magnitude response are simply given by $|\rho(\lambda)|^2$ and $|t_x(\lambda)|^2$, respectively. The reflection phase response and the transmission phase response are defined as $\arg[\rho(\lambda)]$ and $\arg[t_x(\lambda)]$, respectively, where arg stands for argument. We here consider only the reflection frequency response (i.e., magnitude and phase responses) or Eq. (13) because we are interested in the bandpass response of the filter. The transmission magnitude response will give a bandstop or notch response. The delay time, $\tau_\rho(\lambda)$, of light reflected off a grating corresponds to the phase change of $\rho(\lambda)$ relative to the optical wavelength λ, and is given by (Erdogan, 1997)

$$\tau_\rho(\lambda) = -\frac{\lambda^2}{2\pi c} \cdot \frac{d\theta_\rho}{d\lambda} \tag{15}$$

where $\theta_\rho = \arg\left[\rho(\lambda)\right]$ is the phase response of $\rho(\lambda)$ and c is the speed of light in vacuum. The dispersion of the grating, $d_\rho(\lambda)$, is therefore given by

$$d_\rho(\lambda) = \frac{d\tau_\rho}{d\lambda} = -\frac{\lambda^2}{2\pi c}\left(\frac{d^2\theta_\rho}{d\lambda^2} + \frac{2}{\lambda}\cdot\frac{d\theta_\rho}{d\lambda}\right) \tag{16}$$

The non-uniform FBG model based on the cascade of serially-connected *uniform* sub-gratings described here is useful for use in the design of the magnitude response (but not the phase response) of a bandpass filter. The model will be used for the design and fabrication of bandpass filters (where the phase responses are not of interest) which are described in Section 4.3. When both the magnitude and phase responses of the filter (e.g., a linear phase filter) are required to be designed, the non-uniform FBG model based on the cascade of serially-connected *apodized* sub-gratings must be used and it is described below.

2.2.2 A non-uniform FBG model using the cascade of *apodized* sub-gratings

When both the magnitude and phase responses of a filter are required to be designed, the TMM method can be used to model a non-uniform FBG as the cascade of serially-connected *apodized* sub-gratings. The non-uniform FBG can be divided into a number of serially-connected *N* *apodized* sub-gratings or sections (as shown in Fig. 5). Each *apodized* sub-grating section can be described by an analytic transfer matrix. The transfer matrix for the entire non-uniform FBG structure can be obtained by simply multiplying the individual transfer matrices of the apodized sub-gratings.

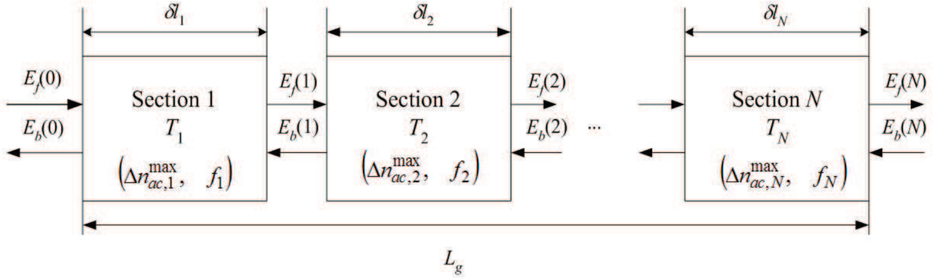

Fig. 5. Schematic diagram of a non-uniform FBG model based on the cascade of serially-connected *apodized* sub-gratings.

Each *apodized* sub-grating section is described by the index perturbation defined in Eq. (4). Note that Eq. (4) is a general equation describing the index perturbation of all kinds of FBGs. To apply Eq. (4) to each *apodized* sub-grating section, the parameters in this equation are re-defined as follows. An *apodized* sub-grating section requires $\phi(z)=0$ in Eq. (4). Putting this condition into Eq. (4) gives a simpler index perturbation, $\delta n_{\text{eff}}(z)$, for each *apodized* sub-grating which is given by

$$\delta n_{\text{eff}}(z) = \Delta n_{\text{dc}}(z) + \Delta n_{\text{ac}}(z)\cdot f(z)\cdot\cos\left(2\pi z/\Lambda(z)\right) \tag{17}$$

Equation (17) is schematically shown in Fig. 3(b). In Fig. 5, $E_{\text{f}}(j;\lambda)$ and $E_{\text{b}}(j;\lambda)$ are the complex electric fields of the forward and backward propagation waves, respectively, describing the j^{th} section. Also, δl_j, $\Delta n_{\text{ac},j}$ and f_j, which are the parameters to be optimized in this particular model, are the length, 'ac' index change (i.e., index modulation) and apodization function of the j^{th} section, respectively. In this model, the parameters that are not optimized are Λ_j and $\Delta n_{\text{dc},j}(z)=\Delta n_{\text{dc},j}$ (and hence $\overline{n}_{\text{eff},j}$ according to Eq. (2)), which are the period and the "dc" index change, respectively, and they are fixed or constant values. L_{g} is the total length of the non-uniform FBG. The designer has the choice of choosing which of these five (5) variables (i.e., δl_j, $\Delta n_{\text{ac},j}$, f_j, Λ_j and $\Delta n_{\text{dc},j}$) are to be optimized, depending on the fabrication condition(s) or constraint(s) of a particular fabrication system. Unlike a uniform FBG (i.e., no apodization) which has sidelobes of about -10 dB in the reflective magnitude response, applying apodization to a non-uniform FBG can significantly suppress the sidelobes to $>-40\text{ dB}$ in the reflective magnitude response. Listed below are several commonly used apodization functions which can be applied to each sub-grating section

$$\text{Raised cosine: } f_j(z)=\frac{1}{2}\left\{1+\cos\left[\frac{\pi\left(z-\delta l_j/2\right)}{\delta l_j}\right]\right\} \tag{18}$$

$$\text{Gaussian: } f_j(z)=\exp\left[-\frac{(4\ln 2)\left(z-\delta l_j/2\right)^2}{\left(\delta l_j/3\right)^2}\right] \tag{19}$$

$$\text{Sine: } f_j(z)=\sin\left(\pi z/\delta l_j\right) \tag{20}$$

$$\text{Quadratic sine: } f_j(z)=\sin^2\left(\pi z/\delta l_j\right) \tag{21}$$

Note that the only difference between a *uniform* sub-grating (as described in Section 2.2.1) and an *apodized* sub-grating is that the former has an index perturbation described by Eq. (5) while the latter has an index perturbation described by Eq. (17). In general, the number of *apodized* sub-gratings is smaller than the number of *uniform* sub-gratings to achieve a particular desired frequency response. Thus Eqs. (6)–(12) are also applicable to the *apodized* sub-gratings by putting the right condition defined in Eq. (17) into Eqs. (6)–(12). Putting Eq. (17) into Eqs. (6)–(12), the reflection frequency response, $\rho(\lambda)$, and the transmission frequency response, $t_x(\lambda)$, can be computed using Eqs. (13) and (14), respectively. Also, the delay time, $\tau_{\rho}(\lambda)$, of light reflected off a grating can be determined using Eq. (15) and the dispersion of the grating, $d_{\rho}(\lambda)$, can be calculated using Eq. (16). This model will be used for the design of FBG-based linear phase filters which is described in Section 4.4.

3. Staged continuous Tabu search algorithm

3.1 Review of Tabu search algorithm

Tabu search (TS) is an iterative search method originally developed by Glover and Laguna (Glover & Laguna, 1998) which has been successfully applied to a variety of combinatorial global optimization problems (Ho et al., 2001; Machado et al., 2001; Chelouah & Siarry, 2000). The TS algorithm has been used for the synthesis of FBG-based linear phase filters using the non-uniform FBG model based on the cascade of serially-connected *apodized* sub-gratings (Ngo et al., 2004). A good analogy of how TS works is mountain climbing, where the climber must selectively remember key elements of the path traveled (using *adaptive memory*) and must be able to strategize choices along the way (using *responsive exploration*). A rudimentary form of this algorithm can be summarized as follows. It starts from an initial solution *s* that is randomly selected. From this current solution *s*, a set of neighbors, called *s*′, is generated by pre-defining such a set of "moves" or perturbations of the current solution. To avoid the endless reiterative cycle, the neighbors of the current solution, which belong to a subsequently defined "tabu list", are systematically eliminated. The objective function to be minimized is then evaluated for each generated solution *s*′, and the best neighborhood of *s* becomes the new current solution *s*′ even if it is worse than *s*. The "move" that generates the new current solution will also be stored in the tabu list, which is circular. When the tabu list is full, it is updated by eliminating the previous estimated solution. Then a new "iteration" is performed; the previous procedure is repeated by starting from the new current solution until a pre-defined stopping condition is satisfied. The TS algorithm has a small probability of becoming trapped in a local optimum. Compared to other traditional methods such as the genetic algorithm (GA) and the simulated annealing (SA), the TS algorithm has one unique advantage in that it can also be organized to take advantage of problem-specific information and thus has higher convergence velocity as well as higher level of reliability. TS also includes "candidate list strategies" for generating and sampling neighbors. These candidate list strategies are very important because often only a relatively small subset of neighbors is generated at any given iteration, especially when a large number of neighborhoods is used, as in the case of multi-variable problems whose neighbors are generated in a multi-dimensional space.

3.2 Review of continuous Tabu search algorithm

As an enhancement to the TS algorithm described above in terms of higher convergence velocity and higher level of reliability, a continuous Tabu search (CTS) algorithm, which employs a special "candidate list strategy" to generate neighbors, has been proposed for the optimization of nonlinear objective functions (Siarry & Berthiau, 1997). In this method, the solution space is divided into several regions. Neighbors are generated in these regions and the remainder of the method consists of an elementary form of TS that uses only the simple tabu list construction as described in Section 3.1. A brief review of the CTS algorithm is described here because it will be used for the development of a staged continuous search (SCTS) algorithm in the subsequent section. For the following optimization problem:

$$\min_{s \in \Psi^k} \left[\Phi(s) \right], \tag{22}$$

where $\Phi(s)$ is the objective function to be minimized, and $s = [x_1, x_2, \cdots, x_k]^{\mathrm{T}}$ is defined as

$$s \in \Psi^k \text{ and } \Psi^k = \{s \mid a_j \le x_j \le b_j\}, \; j = 1, 2, \ldots, k. \tag{23}$$

where a_j and b_j are the boundary values of the j^{th} element (or x_j) of s, and k represents the dimension of the problem or the number of variables. The basic process of the CTS method, which is organized around a simple version of the tabu search, can be summarized as follows.

1. Generate a random point s that belongs to the space Ψ^k as the current solution.
2. A set of neighbors, $s' \in \Psi^k$, is then generated by applying s with a series of perturbations or "moves". Generation of neighbors is defined as follows. The neighborhood space Ψ^k of the current solution, s, is deemed as a ball $B(s,r)$ centered on s with a radius r. Considering a set of concentric balls with radii h_0, h_1, …, h_n, the space is partitioned into n concentric 'crowns'. Hence n neighbors of s are obtained by selecting one point randomly inside each crown and eliminating those neighbors that belong to the "tabu list". Figure 6 shows an example that demonstrates the generation of neighbors for a problem with two variables x_1 and x_2 (k = 2), where the space is partitioned into four (n = 4) concentric "crowns", and four neighbors are produced randomly in their own crown areas.
3. Evaluate these neighbors with the objective function, choose the best neighbor s^* and replace the starting point s with s^* even if it is worse than the current solution. Then update the "tabu list".
4. Clear the "tabu list": in particular those solutions that belong to the "tabu list" can release their tabu status if their "aspiration levels" are sufficiently high.
5. Check the stopping condition and return to step 2 if the condition is not met. Otherwise, stop the iteration procedure and report the results.

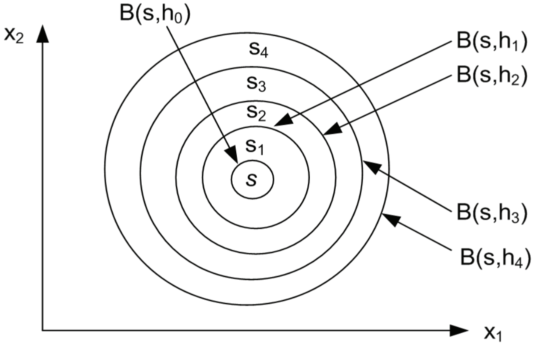

Fig. 6. Partition of the current solution neighborhood, where x_1 and x_2 are two variables (k = 2) and n = 4 concentric "crowns". The neighborhood s_j (j = 1, 2, 3, 4) is selected randomly in its own crown area (Siarry & Berthiau, 1997).

Figure 7 shows the flow chart of the CTS algorithm, where the main stages include the initial solution, generation of neighbors, selection of the solution and tabu list clearance. The strategy of generating neighbors in CTS is more efficient than a naive candidate-list strategy based solely on random sampling, and usually produces neighbors distributed over the whole solution space. Although the method is effective for optimizing functions with two or three variables, its efficiency decreases with an increase in the number of variables of a function such as the case for high-dimension problems. These difficulties can be overcome using a staged continuous Tabu search (SCTS) algorithm which is described next.

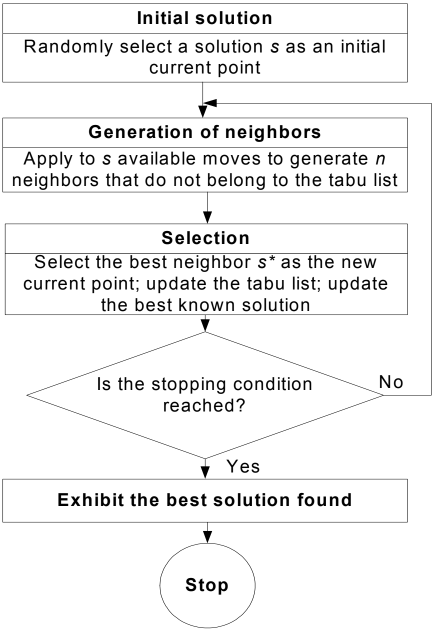
Initial solution
Randomly select a solution *s* as an initial current point
Generation of neighbors
Apply to *s* available moves to generate *n* neighbors that do not belong to the tabu list
Selection
Select the best neighbor *s** as the new current point; update the tabu list; update the best known solution
Is the stopping condition reached?
No
Yes
Exhibit the best solution found
Stop

Fig. 7. General flow chart of a continuous Tabu search (CTS) algorithm.

3.3 Staged continuous Tabu search algorithm

The staged continuous Tabu search (SCTS) algorithm employs the same rudimentary form of tabu search embodies in the CTS algorithm (Zheng et al., 2005). However, SCTS provides an enhanced candidate-list strategy that subdivides the CTS approach into three independent processes that generate candidate neighbors in a different way (see Fig. 8). The first stage attempts to survey the whole solution space to localize a "prospective point", which is a solution likely to produce a global optimum. The objective of the second stage is to find a point close to the global optimum. The third stage starts from the solution found in the second stage, and eventually converges to the global optimum point. The SCTS algorithm is described below with reference to Fig. 8 which shows the steps of the algorithm. The following notations are used in Fig. 8: Ψ^k: space of feasible solutions (k dimensions); s_0 : current solution; n_1 : length of neighbors generated in the first stage (which is equal to *k* here); n_2 : section number divided within the boundary of every element of s_0; n_3 : length of neighbors generated in the third stage; s': the neighborhood of s_0; s^*: the best solution in s'; s_{opt} : current best solution found; and $Mv(j)$: maximum number of iterations without improvement of s_{opt} in the j^{th} stage.

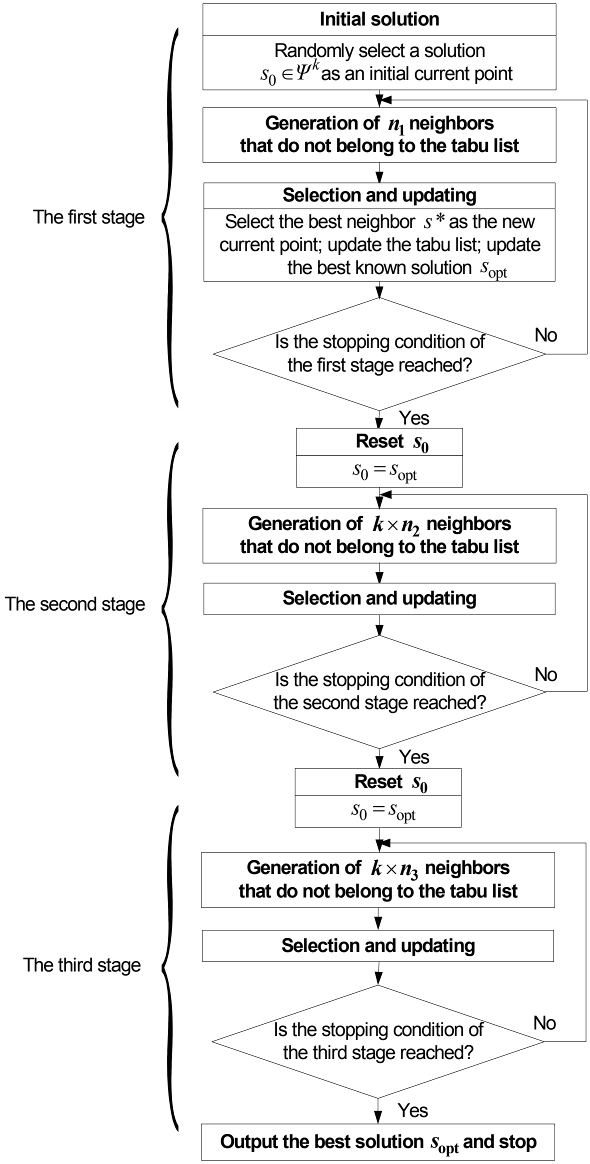
Initial solution
Randomly select a solution $s_0 \in \Psi^k$ as an initial current point
Generation of n_1 neighbors that do not belong to the tabu list
Selection and updating
Select the best neighbor s^* as the new current point; update the tabu list; update the best known solution s_{opt}
Is the stopping condition of the first stage reached?
No
Yes
The first stage
Reset s_0
$s_0 = s_{\text{opt}}$
Generation of $k \times n_2$ neighbors that do not belong to the tabu list
Selection and updating
Is the stopping condition of the second stage reached?
No
Yes
The second stage
Reset s_0
$s_0 = s_{\text{opt}}$
Generation of $k \times n_3$ neighbors that do not belong to the tabu list
Selection and updating
Is the stopping condition of the third stage reached?
No
Yes
The third stage
Output the best solution s_{opt} and stop

Fig. 8. Algorithmic description of the SCTS algorithm.

Generation of Neighborhoods: Neighborhoods are generated in a ball $B(s,r)$ centered on *s* with a radius *r* (see Fig. 6). All neighbors s' meet the condition: $|s'-s| \le r$. In the first stage, the radius r_1 is defined so that the ball $B_1(s,r_1)$ contains the whole *k*-dimension space Ψ^k. With radii $r_1^1, r_1^2, \cdots, r_1^{n_1}$, the ball is partitioned into k concentric "crowns" centered on the current solution. One neighbor is produced in each crown. Thus the *j*th neighbor, s_j', is generated with the condition

$$r_1^{j-1} \le |s_j'-s| \le r_1^j, \quad (r_1^0 = 0). \tag{24}$$

As the ball $B_1(s,r_1)$ includes the whole space Ψ^k, it should be possible for all solutions within it to become neighbors of the current solution, *s*, so that the process can investigate the whole solution space. We define the "moves" to generate neighbors such that some elements of the current solution are randomly replaced. The number of replaced elements depends on different crowns. For example, the *j*th neighbor, s_j', is generated by replacing any *j* element of the current solution. The radius r_2 for the generation of neighbors in the second stage is defined as the minimum radius of radii $r_1^1, r_1^2, \cdots, r_1^{n_1}$ defined in the first stage. Followed with another partition process with a set of radii $r_2^1, r_2^2, \cdots, r_2^{n_2}$, the ball $B_2(s,r_2)$ is divided into n_2 sections. The *j*th neighbor, s_j', is generated with a condition given by

$$r_2^{j-1} \le |s_j'-s| \le r_2^j, \quad (r_2^0 = 0). \tag{25}$$

The minimum radius defined in the first stage is propagated in only one dimension of the current solution. Considering the condition defined in Eq. (23), the boundary can be proportionally divided for every dimension into n_2 partitions. The neighbors can then be generated by replacing the *j*th element of the current solution, x_j', with a number computed by:

$$x_j' = a_j + (j_2 + \mu) \times \frac{b_j - a_j}{n_2}, \quad \text{where } j = 1, 2, \ldots, k;\ j_2 = 1, 2, \ldots, n_2; \tag{26}$$

where μ is a random value between 0 and 1, a_j and b_j are defined in Eq. (23), and $k \times n_2$ is the number of neighbors in the stage. The minimum of radii $r_2^1, r_2^2, \cdots, r_2^{n_2}$ is set as radius r_3 to generate neighbors in the third stage. Instead of partitioning to generate neighbors, the radius of the ball $B_3(s,r_3)$ decreases with an increase in the iteration number. The generation of the *j*th neighbor is defined as

$$x_j' = x_j + \mu_j \times \frac{b_j - a_j}{n_2} \times \frac{M_3 - m}{M_3} \tag{27}$$

where x_j and x_j' are the jth elements of the current solution s and the neighbor s' produced, respectively, m is the iteration number without any improvement on the current solution, and M_3 is the maximum allowable number of iterations without any improvement on the current solution. As pointed out previously, the neighbors in the first stage are generated in the largest possible range so as to explore most of the space. While in the last stage, only a reduced space is used so that the solution will eventually converge to the global optimum.

Tabu list: In the underlying TS algorithm, a tabu list stores some solutions that have recently been selected. It is used to qualify the algorithm to select solutions that have not been selected before so as to escape from being recycled. Because the three stages in the SCTS algorithm are independent of each other, the tabu lists in these stages are thus independent of each other. The list obtained in the first stage will store those 'prospective solutions' found in recent iterations. In the second and third stages, the lists will store the attributes of 'moves' or perturbations that generate the best neighbors in recent iterations. The tabu list in each stage is always reset at the beginning of each stage.

Stopping conditions: The stopping conditions for the three processes are defined below.

1. The program will stop after a given number of iterations when there is no improvement on the value of the objective function. The number of iterations varies in different stages.
2. The result satisfies the prescribed condition. An example of the prescribed condition is the known global optimum of a benchmark test function.
3. The search procedure will stop after it has completed a pre-defined maximum number of iterations.

All stages will be terminated if any one of these stopping conditions is satisfied. That is, if the process is in the first or second stage, it will move into the next stage. However, if the process is in the third stage, the algorithm will stop and report the result.

The sensitivities of several main parameters of the CTS algorithm were discussed (Siarry & Berthiau, 1997). Usually, these parameters should be adjusted empirically according to the nature of the problem so as to achieve an efficient optimization. As the SCTS algorithm is derived from the CTS algorithm, it is found that the properties of some parameters in the SCTS algorithm are similar to those of the CTS algorithm. These parameters are not analyzed individually, and instead a set of empirical values is applied in the experiments for testing the benchmark functions. These empirical values are listed in Table 1.

Parameters used in the SCTS algorithm	Parameters used in the benchmark functions
Number of neighbors in the first stage (n_1)	Number of variables
Number of sections in the second stage (n_2)	5
Number of neighbors in the third stage (n_3)	Number of variables
Number of neighbors in the second stage	5 × number of variables
Maximum number of iterations without any improvement on the objective function value (*Mv*)	{20, 8, 5} for {1st stage, 2nd stage, 3rd stage}
Maximum number of iterations of the SCTS algorithm	8000

Table 1. Parameters of the SCTS algorithm used for testing the benchmark functions.

Numerical experimental results: To demonstrate the effectiveness of the SCTS algorithm, the important parameters to be studied are convergence, speed and robustness. Convergence refers to the evaluation of the search for the global optimum of a function. The test for convergence employed here is the relative error between the optimum obtained by the algorithm, X_{opt}, and a theoretical (or known) value of the optimum, X_{theo}, of the function. The relative error, E_{relative}, is defined as (Andre et al., 2000)

$$E_{\text{relative}} = \frac{\left|X_{\text{opt}} - X_{\text{theo}}\right|}{X_{\text{theo}}} \tag{28}$$

If X_{theo} = 0, Eq. (28) is reduced to

$$E_{\text{relative}} = \left|X_{\text{opt}} - X_{\text{theo}}\right| \tag{29}$$

The criterion of speed refers to the time taken by the algorithm to find the global optimum of an objective function. As the computation time also depends on the speed of the computer, it is better to define the speed criterion by determining the number of evaluations of the objective function required till a global optimum is found. Robustness means that the algorithm is versatile and can be applied to solving a variety of functions. A set of commonly used benchmark functions with known global optima (as listed in the Appendix) is employed to test the algorithm. These benchmark test functions represent various practical problems in engineering. To obtain a statistical comparison of the optimization results, every test is performed 100 times (starting from various randomly selected points) to ensure that the results obtained are reliable.

Table 2 shows the results obtained from the CTS (Siarry & Berthiau, 1997) and SCTS algorithms for the four test functions with two and three variables. The criterion of success is the percentage of trials (out of 100 tests for each function) required to obtain the global optimum with a relative error of < 1%. From the table, both algorithms can successfully find the global optima of all the four test functions which are given in the Appendix. However, the number of evaluations of the Goldprice and Shubert test functions required by the SCTS algorithm (i.e., 696 and 521, respectively) is much smaller than those required by the CTS algorithm (i.e., 1636 and 1123, respectively). This means that the SCTS algorithm has a faster computation rate than that of the CTS algorithm for these two particular test functions. However, the two algorithms have about the same computation rate for the Hartmann1 and Branin functions.

Function	Success rate (%)		Number of evaluations of objective or test functions	
	CTS	SCTS	CTS	SCTS
Goldprice	100	100	1636	696
Hartmann1	100	100	528	691
Branin	100	100	668	491
Shubert	100	100	1123	521

Table 2. Experimental data of the SCTS and CTS algorithms.

Table 3 shows a comparison of the experimental data of various test functions obtained by the SCTS algorithm and an improved genetic algorithm (IGA). Note that the IGA algorithm can potentially yield a complete set of optima of multimodal problems (Andre, Siarry & Dognon, 2000). These test functions have variables ranging from 1 to 20 as given in the Appendix. In the table, Max and Min are, respectively, the maximum and minimum values of a set of optima found over 100 tests. The SCTS algorithm outperforms the IGA algorithm in two ways. The SCTS algorithm has one advantage in that it can find the global optima of the test functions (i.e., Brown1, Brown 3 and F10n functions) with 100% success rate and with very low relative errors; while the IGA algorithm has very low success rates and very high relative errors of these test functions. The other advantage of the SCTS algorithm over the IGA algorithm is that it has a much smaller computation time due to the smaller number of evaluations of the test functions. In addition, for 100% success rates of the test functions achieved by both algorithms, the SCTS algorithm has much smaller relative errors than those of the IGA algorithm. It can thus be concluded from the results shown in Tables 2 and 3 that the SCTS algorithm outperforms the CTS algorithm and the IGA algorithm in terms of higher success rate and higher computation efficiency. Next section describes the use of the SCTS algorithm for the development of a hybrid Tabu search algorithm which is a hybrid of the global SCTS algorithm and the local Quasi-Newton algorithm.

4. Hybrid Tabu search algorithm for optimization and fabrication of non-uniform fiber Bragg gratings

4.1 Staged continuous Tabu search algorithm for optimization of non-uniform fiber Bragg gratings

This section describes the SCTS algorithm (as described in Section 3.3) for use in the optimization of non-uniform FBGs, and it is schematically shown in Fig. 9.

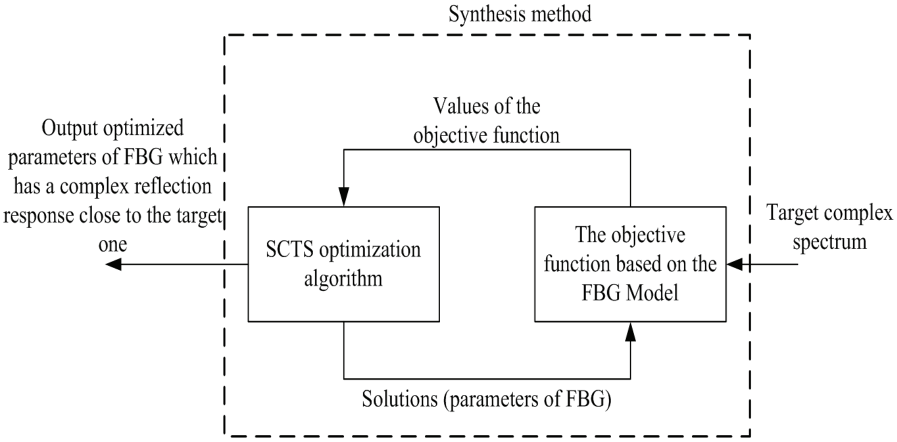
Synthesis method
Values of the objective function
Output optimized parameters of FBG which has a complex reflection response close to the target one
SCTS optimization algorithm
The objective function based on the FBG Model
Target complex spectrum
Solutions (parameters of FBG)

Fig. 9. Block diagram of the SCTS algorithm for use in the optimization of FBGs.

The synthesis problem of an FBG is formulated as an objective function which measures the error between the designed reflection frequency response of the FBG using the SCTS algorithm and the target frequency response. The synthesized FBG using the SCTS algorithm can be carried out using the steps as described below.