Optimal Golomb Ruler Sequences as WDM Channel-Allocation Algorithm Generation: Cuckoo Search Algorithm with Mutation

Nisha Kumari
Department of Electronics and Communication Engineering, Institute of Science and Technology, Klawad, India

Tapeshwar Singh
Department of Electronics and Communication Engineering, Institute of Science and Technology, Klawad, India

Shonak Bansal
Department of Electronics and Communication Engineering, PEC University of Technology, Chandigarh, India

ABSTRACT
With the hailing winds of development, humans are still can’t fully act individually, the dependency over the nature is indispensable. In this paper, a hybrid nature inspired based algorithm named Cuckoo search algorithm with mutation (CSAM) has been used to solve the channel-allocation problem presents in optical wavelength division multiplexing (WDM) systems. The channels can be allocated by using the concept of shortest length rulers called optimal Golomb ruler (OGR) sequences to suppress four-wave mixing (FWM) crosstalk. The simulation results reveals that computational time taken by CSAM to generate channel-allocation algorithm has been abated substantially unlike other existing nature inspired based algorithms such as Genetic algorithms (GAs), Biographically based optimization (BBO), and Cuckoo search algorithm (CSA). The simulation results obtained by proposed hybrid algorithm demonstrates better and efficient in terms of length of the ruler, total channel bandwidth, and bandwidth expansion factor compared to simple classical approaches such as Extended quadratic congruence (EQC) and Search algorithm (SA) and other nature inspired based algorithms.

Keywords
Bandwidth expansion factor, Channel-allocation, Cuckoo search algorithm with mutation, Four-wave mixing, Optimal Golomb ruler, Wavelength division multiplexing.

1. INTRODUCTION
The optical communication system has been used most widely due to the ability of silica doped fibers to carry large amounts of information over long distances. To utilize the available bandwidth, numerous channels at different wavelengths/frequencies have been multiplexed on the single fiber called wavelength division multiplexing (WDM). To increase system margins, high transmitter powers or low attenuation losses are required. All these attempts to fully utilize the capabilities of silica doped fibers will ultimately be limited by nonlinear optical interactions [1],[2] between the information bearing light waves and the transmission medium. These optical nonlinearities can lead to interference, distortion and excess attenuation of the optical signals, resulting in system degradations. Four–wave mixing (FWM) is one of the fiber nonlinear effects whereby three frequencies intersect through electric susceptibility of optical fiber to generate new wave of different frequency [2]. In WDM systems channels are equally spaced from each other. But there is high probability that FWM signals may fall in to the WDM system, resulting in discrepancies, which leads to crosstalk [3]. In order to reduce the FWM crosstalk, there is a need to develop algorithms to allocate the channel frequencies. Several unequally spaced channel-allocation algorithms [4]-[11] have been proposed which have the drawback of increased optical channel bandwidth requirement compared to equally spaced channel allocation. This paper proposes a hybrid nature inspired based unequally spaced channel algorithm by using the concept of Optimal Golomb ruler (OGR) sequences.

The concept of Golomb rule had been introduced by Babcock [4] and further derived by Golomb et. al. [12]. Golomb ruler refers to a set of non-negative integer such that no distinct pairs of numbers called marks from the set have same difference [13]-[16]. The difference between largest and smallest marks is known as length of the ruler. A ruler is called perfect Golomb ruler only when it measures all the non–negative integer distances from 0 to length L of the ruler [15], [17], [18]. The shortest length ruler for a given mark is named as optimum Golomb ruler. There can be numerous different OGRs for a specific marks value. Golomb rules represent a class of NP-complete problems [19]. Some algorithms like Genetic algorithms (GAs) and its hybrid form [20]-[23], Biogeography based optimization (BBO) [24]-[26], Big bang crunch algorithm [27], [28] and its hybridization [29], Firefly algorithm (FA) [30], Cuckoo search algorithm (CSA) [31], Multi–objective flower pollination algorithm (MOFPA) and its hybridization form [32], and Bat inspired algorithm [33] had been applied in finding the better solution of such NP-complete problems. In attempt to reduce the FWM crosstalk effect in WDM, unequally spaced channel allocation by using the concept of optimal Golomb ruler (OGR) sequences [34]-[38] has been used in this paper. The OGR sequences have been generated in a reasonable time by using hybrid Cuckoo search. For the hybridization, the differential evolution (DE) mutation strategy [39], [40] has been used with simple CSA, named Cuckoo search algorithm with mutation (CSAM).

2. CUCKOO SEARCH ALGORITHM WITH MUTATION (CSAM)
Cuckoo search algorithm is a nature inspired algorithm that is derived by observing the problem solving capability of nature. Inspired by obligate brood parasitism of cuckoo spices by laying down their eggs in the nest of other host bird, Yang et. al. [41], [42] implemented Cuckoo search algorithm with the three following idealized rules [41]:

1. Every cuckoo lay one egg at a time, and dump it in a randomly chosen nest;
2. The best nest with high quality of eggs (solution) are carried over to the next generations;
3. The number of available host nests is fixed, and a host can discover an alien egg with probability \( p_h \in [0,1] \). In this case, the host bird can either throw the egg away or simply abandon the nest so as to build a completely new nest in a new location.

To make the things simpler, the last assumption can be approximated by a fraction of the \( p_h \) of \( n \) nests that are replaced by new nests with new random solutions. Considering the maximization problem, the quality or fitness of solution can simply be proportional to the value of its objective function. To optimize the CSA one egg in a nest represent a solution and a cuckoo egg represents a new solution is used. The objective is to replace the new and better solutions with worst solutions that are in nests.

New solutions \( x' \) are generated say for a cuckoo \( i \), by using Lévy flight are performed by equation (1)[42]:

\[
x'^{(i+1)} = x^{(i)} + \alpha \odot \text{Lévy}(\lambda)
\]

Where \( \alpha > 0 \) is the step size, which should be related to the scale of the specified problem.

Although the algorithm CSA in its simplified form works well in the exploitation as compared to other optimization algorithms but there has been some problems in the global exploration of the search space. If all of the solutions in the initial phase of the optimization algorithm has been collected in a small part of search space, the algorithm may not find the optimal result and with a high probability, it may be trapped in that sub-domain. One can consider a large number for solutions to avoid this shortcoming, but it causes an increase in the function calculations as well as the computational costs and time. So for the optimization algorithm CSA, there is a need by which exploration and exploitation can be enhanced and the algorithms can work more efficiently. In order to improve the performance of simple CSA the differential evolution (DE) mutation strategy [39], [40] has been applied to get fitness of solutions in the proposed nature-inspired metaheuristic CSA algorithm, which is the main technical contribution of this paper. The new modified algorithm is known as Cuckoo search algorithm with mutation (CSAM).

In CSAM the mutation rate probability is determined on the fitness value. Objective of the proposed algorithm is to have quick convergence and to reduce the computational time. The mutation rate probability \( MR^i \) of each solution \( x_i \) at running iteration index \( t \), can be represented by the equation (2):

\[
MR^i_t = \frac{f^i_t}{\text{Max}(f^i)}
\]

where \( f^i_t \) is the fitness value of each solution \( x_i \), at the iteration index \( t \) and \( \text{Max}(f^i) \) is the maximum fitness value in the population at iteration \( t \). For the proposed algorithm, fitness of each solution is updated by the mutation equation [39], [40] which is used throughout this paper is given by the equation (3):

\[
x'_i = x^i_t + p_m(x^T_{\text{best}} - x^i_t) + p_m(x^T_i - x^i_{r_1})
\]

where \( x^i_t \) is the population at running iteration index \( t \), \( x^T_{\text{best}} = x^T_{k-1} \) is the current best solution at iteration one less than running iteration index \( t \), \( p_m \) is the mutation operator, \( r_1 \) and \( r_2 \) are different from running index. Typical values of \( p_m \) are 0.001 to 0.05.

The mutation strategy increases the chance for good solution, but if mutation rate exceed \( (p_m > 1) \) causes too much search exploration and is disadvantageous to the improvement of candidate solutions. Even if mutation rate continues to decrease and if it becomes \( (p_m < 0.01) \) optimization ability decreases rapidly. A small value of \( p_m \) is not able to sufficiently increase solution diversity [26].

The Lévy flight distribution used for the proposed algorithm is given by the equation (4):

\[
L(\lambda) \sim \frac{\lambda^{\lambda} \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{s+2}}, \quad (s > s_0 > 0)
\]

Fig. 1. Pseudo Code for CSAM

3. PROBLEM FORMULATION

In this paper the application of Cuckoo search based algorithm has been proposed to solve unequally spaced channel-allocation algorithm problem in WDM systems. So, the first
problem formulation is to obtain the OGR sequences by optimizing the length of the ruler length, total bandwidth occupied by the channel and hence the bandwidth expansion factor. Thus, if channel spacing between any pair of channel is \( CS \) and total numbers of channels is \( n \) then optimized ruler length is [26]:

\[
RL = \sum_{i=1}^{n} (CS) \quad (5)
\]

Subject to \((CS)_i \neq (CS)_j\)

Thus, if an individual element is denoted as \( IE \), then sum of all individual elements form the total bandwidth that is denoted as \( TBW_{in} \) [26]:

\[
TBW_{in} = \sum_{i=1}^{n} (IE) \quad (6)
\]

Subject to \((IE)_i \neq (IE)_j\)

Lower bound on the total optical bandwidth can be achieved from the condition [5]:

\[
TBW_{eq} = [1 + ((n/2-1)/N)]TBW_{eq} \quad (7)
\]

where \( N \) is minimum channel separation and \( TBW_{eq} \) is total bandwidth of a WDM system where channel are equally spaced, which is given by equation (8) [10]:

\[
TBW_{eq} = (n-1)\Delta f_c \quad (8)
\]

where \( \Delta f_c = CS \) is the channel-to-channel separation. Thus our objective is to optimize the bandwidth expansion factor (BEF) [10]:

\[
BEF = \frac{TBW_{in}}{TBW_{eq}} \quad (9)
\]

The pseudo code for the proposed Cuckoo search algorithm with mutation to generate OGRs is given in Figure 2.

4. SIMULATION RESULTS AND DISCUSSION

To generate OGRs the CSAM algorithm has been written and tested in Matlab\textsuperscript{\textregistered} 2011a language under Windows \textregistered operating system. The algorithm has been executed on Laptop with Intel Pentium 2.20 GHz processor with a RAM of 4 GB. To show the effectiveness of the proposed algorithm, its performance is being compared with known OGRs [12]-[15], [35], two of the existing classical computing algorithms i.e. EQC and SA [3], [10], [22] and nature inspired algorithms GAs [26], BBO [24]-[26] and CSA [31] algorithms. The application of classical algorithms i.e. EQC and SA is limited to prime power [2], so the length of ruler and total bandwidth for EQC and SA are shown by a dash line in Table 2 whereas Table 3 list the comparison of bandwidth expansion factor of CSAM algorithm with known OGRs [12]-[15], [35], simple classical algorithms i.e. EQC and SA [3], [10], [22] and nature inspired algorithms GAs [26], BBO [24], 25 and CSA [31].

Comparing the simulation results obtained from Cuckoo search algorithm with mutation with known OGRs, EQC, SA, GAs, BBO and CSA is perceived that there is a significant improvement with respect the length of the ruler, the total bandwidth occupied, bandwidth expansion factor and average CPU time that is, the results gets better. So, it is concluded from Table 2 and Table 3 that the performance of proposed CSAM algorithm is better than the existing algorithms.

4.2 Comparison of CSAM with Previous Existing Algorithms in Terms of Ruler Length, Total Bandwidth, Average CPU Time and Bandwidth Expansion Factor

The CSAM algorithm has been used in this paper to optimize the length of the ruler so as to conserve the total bandwidth occupied by the channels and hence the bandwidth expansion factor in less computational time. Table 2 list the length of ruler (\( RL \)), total optical bandwidth (\( TBW \)) and average CPU time occupied by different sequences obtained by proposed algorithm for various channels \( n \) and its comparison with known OGRs [12]-[15], [35], two of the existing classical computing algorithms i.e. EQC and SA [3], [10], [22] and nature inspired algorithms GAs [26], BBO [24]-[26] and CSA [31] algorithms. The application of classical algorithms i.e. EQC and SA sequence is limited to prime power [2], so the length of ruler and total bandwidth for EQC and SA are shown by a dash line in Table 2 whereas Table 3 list the comparison of bandwidth expansion factor of CSAM algorithm with known OGRs [12]-[15], [35], simple classical algorithms i.e. EQC and SA [3], [10], [22] and nature inspired algorithms GAs [26], BBO [24], [25] and CSA [31].

Comparing the simulation results obtained from Cuckoo search algorithm with mutation with known OGRs, EQC, SA, GAs, BBO and CSA is perceived that there is a significant improvement with respect the length of the ruler, the total bandwidth occupied, bandwidth expansion factor and average CPU time that is, the results gets better. So, it is concluded from Table 2 and Table 3 that the performance of proposed CSAM algorithm is better than the existing algorithms.

5. CONCLUSION

This paper presented the application of nature-inspired Cuckoo search algorithm with mutation to generate optimal Golomb ruler sequences needed for optical WDM systems.

The Cuckoo search algorithm with mutation comprised of fitness based on differential mutation strategy to explore the search space. The optimal Golomb ruler’s sequence provides the distinct channel–allocation in optical WDM systems to reduce the FWM crosstalk. It has been observed that proposed CSAM produces Golomb ruler sequences very efficiently and effectively. The performance is being compared with the existing classical and nature-inspired algorithms in terms of the length of ruler, total optical channel bandwidth, average CPU time and bandwidth expansion factor obtained by the different sequences. The preliminary results indicate that proposed CSAM appears to be most efficient algorithm to generate OGRs for optical WDM systems and outperforms the existing algorithms.

---

Table 1. Simulation parameters for CSAM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>0.01</td>
</tr>
<tr>
<td>( p_s )</td>
<td>0.5</td>
</tr>
<tr>
<td>( p_m )</td>
<td>0.05</td>
</tr>
<tr>
<td>Popsize</td>
<td>20</td>
</tr>
<tr>
<td>Iterations</td>
<td>1000</td>
</tr>
</tbody>
</table>

---

International Journal of Computer Applications (0975 – 8887)
Volume 142 – No.9, May 2016

23
Cuckoo search algorithm with mutation to generate OGRs

Begin
/* CSAM parameter initialization */
Define operating parameters for CSAM;
Initialize the number of channels/marks, lower and upper bound on the ruler length;
/* End of CSAM parameter initialization */
While not Popsize /* Popsize is the population size input by the user */
   Generate a random set of integer population of host nests;
   /* Number of integers in host nests is being equal to the number of channels */
   Check Golombness of each nests;
   If Golombness is satisfied
      Retain that nest;
   Else
      Remove that particular nest from the generated population;
   End if
End while
Compute the fitness value i.e. ruler length and total channel bandwidth;
Rank the population from best to worst based on fitness value;
ion */
A: Get a cuckoo (say i) randomly by Lévy flights
/* Mutation*/
Compute mutation rate probability $MR_i$ via equation (2);
   If $(MR_i < rand)$
      Perform mutation via equation (3);
   End if
/* End of mutation */
   Recheck Golombness of updated candidates;
   If Golombness is satisfied
      Retain that candidate and go to B;
   Else
      Retain the previous generated solution and then go to A;
   /* Previous generated solution is being equal to the solution generated into the parameter initialization step */
End if
B: Recompute the fitness values $F_i$ of the modified candidates;
   Choose a nest among $n$ (say j) randomly;
   If ($F_i > F_j$),
      Replace j by the new solution;
End if
A fraction of worse nests are abandoned and new ones are built;
Keep the best solutions (or nests with quality solutions);
Rank the candidates from best to worst based on fitness values and find the current best;
End While
Display the near OGR sequences;
End

Fig. 2. Pseudo Code for CSAM to generate OGR sequences
Table 2. Performance Comparison of proposed CSAM Algorithm with Known OGR, EQC, SA, GAs, BBOand CSA in terms of Ruler Length, Total Bandwidth, and Average CPU Time

<table>
<thead>
<tr>
<th>r</th>
<th>Known OGs [12-15],[31]</th>
<th>Classical Algorithms</th>
<th>Proposed Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL</td>
<td>TBW (Hz)</td>
<td>RL</td>
<td>TBW (Hz)</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>25</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>54</td>
<td>47</td>
</tr>
<tr>
<td>7</td>
<td>77</td>
<td>117</td>
<td>81</td>
</tr>
<tr>
<td>8</td>
<td>54</td>
<td>117</td>
<td>91</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>206</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>55</td>
<td>249</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>72</td>
<td>369</td>
<td>391</td>
</tr>
<tr>
<td>12</td>
<td>85</td>
<td>505</td>
<td>231</td>
</tr>
<tr>
<td>13</td>
<td>106</td>
<td>660</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>127</td>
<td>924</td>
<td>325</td>
</tr>
<tr>
<td>15</td>
<td>181</td>
<td>1045</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>175</td>
<td>1298</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>198</td>
<td>1661</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>216</td>
<td>1804</td>
<td>561</td>
</tr>
<tr>
<td>19</td>
<td>246</td>
<td>2229</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>283</td>
<td>2794</td>
<td>703</td>
</tr>
</tbody>
</table>

Table 3. Performance Comparison of proposed CSAM Algorithm with Known OGR, EQC, SA, GAs,BBOand CSA in terms of Bandwidth Expansion Factor(BEF)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>12</td>
<td>0.9166</td>
<td>2.3333</td>
<td>2.3333</td>
<td>0.9166</td>
<td>0.9166</td>
<td>0.9166</td>
<td>0.9166</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>1.2571</td>
<td>4</td>
<td>1.7142</td>
<td>1.2857</td>
<td>1.4</td>
<td>1.4285</td>
<td>1.3428</td>
</tr>
<tr>
<td>8</td>
<td>70</td>
<td>1.6714</td>
<td>5.4</td>
<td>2.7</td>
<td>1.9</td>
<td>1.8714</td>
<td>1.6714</td>
<td>1.6714</td>
</tr>
<tr>
<td>12</td>
<td>176</td>
<td>2.8579</td>
<td>8.1875</td>
<td>3.875</td>
<td>3.75</td>
<td>3.4375</td>
<td>2.8579</td>
<td>2.8579</td>
</tr>
<tr>
<td>20</td>
<td>532</td>
<td>5.2518</td>
<td>13.4624</td>
<td>12.1428</td>
<td>9.2875</td>
<td>9.1334</td>
<td>5.2518</td>
<td>5.2518</td>
</tr>
</tbody>
</table>

6. REFERENCES


Distributed.net, “Project OGR”. Available at http://www.distributed.net/og.


http://mathworld.wolfram.com/PerfectRuler.html

http://mathworld.wolfram.com/GolombRuler.html

http://theinf1.informatik.unijena.de/teaching/ss10/oberse


