Optimal Golomb Ruler Sequences asWDM 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 refer 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 cuckoos 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 $p_a \in [0,1]$. 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_a 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^t are generated say for a cuckoo i, by using Lévy flight are performed by equation (1)[42]:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda) \quad (1)$$

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 inthe proposed natureinspired 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^t of each solution x_i at running iteration index t,can be represented by the equation (2):

$$MR_i^t = \frac{f_i^t}{Max(f^t)} \quad (2)$$

where f_i^t is the fitness value of each solution $x_i$at the iteration index t, and $Max(f^t)$ 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):

$$x_i^t = x_i^{t-1} + p_m(x_{best}^{t-1} - x_i^{t-1}) + p_m(x_{r_1}^{t-1} - x_{r_2}^{t-1}) \quad (3)$$

where x_i^t is the population at running iteration index t, $x_{best}^{t-1} = x_*^{t-1}$ is the current best solution at iteration one less than running iteration index t, p_m is the mutation operator, r_1 &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 continuous 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):

$$L(\lambda) \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s >> s_0 > 0) \quad (4)$$

Here, $\Gamma(\lambda)$ is the standard gamma distribution. Throughout the paper, λ= 1.5 is used. In theory, it is required that $|s_0| >> 0$, but in practice s_0 can be as small as 0.1 [43]. The proposed algorithm CSAM canexplore new search space by the mutation and random walk A fundamental benefit of using mutation Lévy flight strategies in proposed algorithm is its ability to improve its solutions over time which does not seemin the existing algorithms [20], [22], [44-46] to find near OGRs.Based on the above discussion Pseudo code for CSAM is shown in Figure 1.

Cuckoo search algorithm with mutation(CSAM)
Begin
/*CSAM parameter initialization*/
Define objective function $f(x)$, $x=(x_1, \ldots, x_d)^T$;
Generate initial population of n host nests $x_i (i = 1, 2, \ldots, n)$;
/*End of CSAM parameter initialization*/
While not *T*/* ***T* is a termination criterion */**
  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*/
Evaluate its quality/fitnessF_i;
Choose a nest among *n* (say *j*) randomly;
If (*Fi >Fj*),
Replace *j* by the new solution;
End if
  A fraction (p_a) of worse nests are abandoned and new ones are built;
  Keep the best solutions(or nests with quality solutions);
  Rank the solutions and find the current best;
End While
Postprocess the results and display the sequences;
End

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 obtainthe 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]:

$$RL = \sum_{i=1}^{n} (CS)_i \qquad (5)$$

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_{un} [26]:

$$TBW_{un} = \sum_{i=1}^{n} (IE)_i \qquad (6)$$

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

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

$$TBW_{un} \geq [1 + ((n/2 - 1)/N)] TBW_{eq} \qquad (7)$$

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]:

$$TBW_{eq} = (n-1)\Delta f_c \qquad (8)$$

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

$$BEF = \frac{TBW_{un}}{TBW_{eq}} \qquad (9)$$

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–R2011a language under Windows 7operating 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 three of the nature inspired algorithm i.e. GAs [26], BBO [24, 25] and CSA [31] algorithms of generating unequal spaced WDM channel–allocation OGR sequences.

4.1 Simulation Parameters for Cuckoo Search Algorithm with Mutation

To get optimum results, following parameter values of CSAM have finally been settled as shown in Table 1.

Table 1. Simulation parameters for CSAM

Parameter	Value
α	0.01
p_a	0.5
p_m	0.05
Popsize	20
Iterations	1000

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 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.

```
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 values 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

n	Known OGRs [12-15], [35]		Algorithms																
			Classical Algorithms				Existing Nature-Inspired Algorithms									Proposed Algorithm			
			EQC [3], [10], [22]		SA [3], [10], [22]		GAs [26]			BBO [24]-[26]			CSA [31]			CSAM			
	RL	*TBW_{un}* (Hz)	*RL*	*TBW_{un}*(Hz)	*RL*	*TBW_{un}*(Hz)	*RL*	*TBW_{un}*(HZ)	CPU Time (sec.)	*RL*	*TBW_{un}*(Hz)	CPU time (sec)	*RL*	*TBW_{un}*(Hz)	CPU Time (sec.)	*RL*	*TBW_{un}*(Hz)	CPU Time (sec.)	
4	6	11	15	28	15	28	6 7	11	0.001	6 7	11 11	0.000	6 7	11	0.000	6	11	0.000	
5	11	25 28	-	-	-	-	12 13	23 25 29	0.021	12 13	23 24	0.020	11 12	23 24	0.001	11 12	23 24	0.001	
6	17	44 47 50 52	45	140	20	60	17 18 21	42 44 45	0.780	17 18 20 21	42 43 44 45 49	0.743	17 18	42 44 47 50	0.0517	17 18	42 44 47	0.0505	
7	25	77 81 87 90 95	-	-	-	-	27 28 29 30 31 32	73 78 79 80 83 86 95	1.120	27 29 30 31 32 33	73 82 83 84 91 95	1.180	25 26 27	73 77 80	0.0819	25 26 27	73 77	0.0798	
8	34	117	91	378	49	189	35 41 42 45 46	121 126 128 129 131 133	1.241	34 39 40 42	121 125 127 131	1.239	34 39	113 117	0.1379	34 39	113 117	0.1365	
9	44	206	-	-	-	-	52 56 59 61 63 65	192 193 196 203 225	1.1711	49 56 59 61 62 64	187 200 201 206 215	1.699	44	206	1.177	44 47 55	176 185 206	1.086	
10	55	249	-	-	-	-	75 76	283 287 301	5.499e +01	74	274	5.491e +01	55	249	3.127+ 01	55	249	3.091e+ 01	
11	72	389 391	-	-	-	-	94 96	395 456	7.200e +02	86 103 104 114 118	378 435 440 491	7.11e+ 02	72	391	4.558e +02	72 103	378 386	4.521e+ 02	
12	85	503	231	1441	132	682	123 128 137	532 581 660	8.602e +02	116 124 138	556 590 605	8.6e+ s02	85	503	5.342e +02	85	503	5.318e+ 02	
13	106	660	-	-	-	-	203 241	1015 1048	1.070e +03	156 171 187	768 786 970	1.03e+ 03	106	660	8.711e +02	106	660	8.341e+ 02	
14	127	924	325	2340	286	1820	206 228 230	1172 1177 1285	1.028e +03	169 206 221	999 1001 1166	1.027e +03	127	924	1.014e +03	127	924	0.890e+ 03	
15	151	1047	-	-	-	-	275 298	1634 1653	1.440e +03	260 267	1322 1554	1.48e+ 03	151	1047	1.149e +03	151	1047	1.078e+ 03	
16	177	1298	-	-	-	-	316	1985	1.680e +03	283	1804	1.677e +03	177	1298	1.332e +03	177	1298	1.145e+ 03	
17	199	1661	-	-	-	-	355	22055	5.048e +04	354 369	2201 2208	5.04e +04	199	1661	3.43e+03	199	1661	3.211e +03	
18	216	1894	561	5203	493	5100	427 463	2599 3079	6.840e +04	362 445	2566 2912	6.839e +04	216	1894	4.067e +0.04	216	1894	3.67e+ 04	
19	246	2225	-	-	-	-	567 597	3432 5067	8.280e +04	467 475 584	3337 3408 4101	8.28e+ 04	246	2225	6.571e +04	246	2225	5.378e +04	
20	283	2794	703	7163	703	6460	615 673 680 691	4660 4826 4905 4941	1.1242 8e+05	578 593 649	4306 4517 4859	1.1196 e+05	283	2794	7.118e +04	283	2794	6.974e +04	

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

n	TBW_{eq} (Hz)	*BEF*						
		Known OGRs [12-15], [35]	EQC [3], [10], [22]	SA [3], [10], [22]	GAs [26]	BBO [24], [25]	CSA [31]	CSAM (Proposed Algorithm)
4	12	0.9166	2.3333	2.3333	0.9166	0.9166	0.9166	0.9166
6	35	1.2571	4	1.7142	1.2857	1.4	1.4285	1.3428
8	70	1.6714	5.4	2.7	1.9	1.8714	1.6714	1.6714
12	176	2.8579	8.1875	3.875	3.75	3.4375	2.8579	2.8579
14	247	3.7408	9.4736	7.3684	5.2024	4.7206	3.7408	3.7408
18	425	4.4564	12.2423	12	7.2447	6.8517	4.4564	4.4564
20	532	5.2518	13.4642	12.1428	9.2875	9.1334	5.2518	5.2518

6. REFERENCES

[1] Chraplyvy, R. 1990. Limitations on Lightwave Communications Imposed by Optical-Fiber Nonlinearities. J. Lightwave Technology, Vol. 8, pp.1548–1557.

[2] Marcuse,D., Chraplyvy, R. and Tkach, R. W. 1991 Effect of Fiber Nonlinearity on Long-Distance Transmission. J. Lightwave Technology, Vol.9, pp. 121–128.

[3] Kwong, W.C. and Yang, G.C. 1997. An Algebraic Approach to the Unequal-Spaced Channel-Allocation Problem in WDM Lightwave Systems‖, IEEE Transactions on Communications, Vol. 45, No. 3, 352-359 .

[4] Babcock, W. C. 1953. Intermodulation interference in radio systems, Bell Systems Technical Journal, pp. 63–73.

[5] Forghieri, F., Tkach, R. W. and Chraplyvy, A. R. 1995. WDM systems with unequally spaced channels. J. Lightwave Technol., Vol. 13, pp. 889–897.

[6] Sardesai, H. P. 1999. A Simple Channel Plan to Reduce Effects of Nonlinearities In Dense WDM Systems. Lasers and Electro–Optics, (23–28, May–1999), pp. 183–184.

[7] Tonguz, O. K. and Hwang B. 1998. A Generalized Suboptimum Unequally Spaced Channel Allocation Technique—Part II: In coherent WDM systems. IEEE Trans. Commun., Vol. 46, pp. 1186–1193.

[8] Atkinson, M. D., Santoro, N. and Urrutia, J. 1986. Integer sets with distinct sums and differences and carrier frequency assignments for nonlinear repeaters. IEEE Trans. Commun., Vol. COM-34.

[9] Hwang, B. and Tonguz, O. K. 1998. A Generalized Suboptimum Unequally Spaced Channel Allocation Technique—Part I: In IM/DD WDM systems. IEEE Trans. Commun., Vol. 46, pp. 1027–1037.

[10] Randhawa, R., Sohal, J. S. and Kaler, R. S. 2009. Optimum Algorithm for WDM Channel Allocation for Reducing Four-Wave Mixing Effects. Optik120, pp. 898–904.

[11] http://www.compunity.org/events/pastevents/ewomp204/jaillet_krajecki_pap_ew04.pdf

[12] Bloom, Gray S. and Golomb, S.W. 1977. Applications of Numbered Undirected Graphs. In Proceedings of the IEEE, Vol. 65, No. 4, (April 1977), pp. 562–570.

[13] Dimitromanolakis, A. 2002. Analysis of the Golomb Ruler and the Sidon Set Problems, and Determination of Large, Near-Optimal Golomb Rulers. Master's Thesis, Department of Electronic and Computer Engineering, Technical University of Crete.

[14] Dollas, A., Rankin, W. T., and McCracken, D. 1998. A New Algorithm for Golomb Ruler Derivation and Proof of the 19 Mark Rulers. IEEE Transactions on Information Theory, Vol. 44, No. 1, pp. 379–382.

[15] Distributed.net, "Project OGR". Available at http://www.distributed.net/ogr.

[16] Yang, X.–S., Karamanoglu, M., and He., X. S. 2014. Flower Pollination Algorithm: A Novel Approach for Multi-objective Optimization. Engineering Optimization, Vol. 46, Issue 9, pp. 1222–1237.

[17] http://mathworld.wolfram.com/PerfectRuler.html

[18] http://mathworld.wolfram.com/GolombRuler.html

[19] http://theinf1.informatik.unijena.de/teaching/ss10/oberseminar-ss10

[20] Soliday, S. W., Homaifar, A. and Lebby, G. L. 1995. Genetic Algorithm Approach to the Search for Golomb Rulers. In Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA–95), Morgan Kaufmann, pp. 528–535.

[21] Robinson, J. P. 2000. Genetic Search for Golomb Arrays. IEEE Transactions on Information Theory, Vol. 46, No. 3, pp. 1170–1173.

[22] Shobhika. 2005. Generation of Golomb Ruler Sequences and Optimization Using Genetic Algorithm. M.Tech. Thesis, Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Deemed University, Patiala.

[23] Ayari, N., Luong, T. V. and Jemai, A. 2010. A Hybrid Genetic Algorithm for Golomb Ruler Problem. In Proceeding of ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2010), pp.1–4.

[24] Bansal, S., Kumar, S., Sharma, H. and Bhalla, P. 2011. Generation of Golomb Ruler Sequences and Optimization Using Biogeography Based Optimization. In Proceedings of 5th International Multi Conference on Intelligent Systems, Sustainable, New and Renewable Energy Technology and Nanotechnology (IISN–2011), Institute of Science and Technology Klawad, Haryana, pp 282–288.

[25] Bansal, S., Kumar, S., Sharma, H. and Bhalla, P. 2011. Golomb Ruler Sequences Optimization: A BBO Approach. International Journal of Computer Science and Information Security (IJCSIS), Pittsburgh, PA, USA, Vol. 9, No. 5, pp. 63–71.

[26] Bansal, S., 2014. Optimal Golomb Ruler Sequence Generation for FWM Crosstalk Elimination: Soft Computing Versus Conventional Approaches. Applied Soft Computing Journal (Elsevier), Vol. 22, pp. 443–457.

[27] Bansal, S., Kumar, S., and Bhalla P. 2013. A Novel Approach to WDM Channel Allocation: Big Bang–Big Crunch Optimization. In the proceeding of Zonal Seminar on Emerging Trends in Embedded System Technologies (ETECH-2013) organized by The Institution of Electronics and Telecommunication Engineers (IETE), Chandigarh Centre, Chandigarh, pp. 80–81.

[28] Kumar, S., Bansal, S., and Bhalla, P. 2012. Optimal Golomb Ruler Sequence Generation for FWM Crosstalk Elimination: A BB–BC Approach. In Proceedings of 6th International Multi Conference on Intelligent Systems, Sustainable, New and Renewable Energy Technology and Nanotechnology (IISN–2012), Institute of Science and Technology Klawad–133105, Haryana, India, pp. 255–262.

[29] Bali, S., Bansal, S., Kamboj, A., 2015.A Novel Hybrid Multi-objective BB-BC based Channel Allocation Algorithm to Reduce FWM Crosstalk and its Comparative Study.International Journal of Computer Applications(IJCA) Vol. 124, No.12, pp. 38-45.

[30] Bansal, S. and Singh, K., 2014. A Novel Soft–Computing Algorithm for Channel Allocation in WDM Systems. International Journal of Computer Applications (IJCA), Vol. 85, No. 9, pp. 19–26.

[31] Bansal, S., Chauhan, R. and Kumar, P., 2014. A Cuckoo Search based WDM Channel Allocation Algorithm.

International Journal of Computer Applications (IJCA), Vol. 96, No. 20, pp. 6–12.

[32] Jain, P., Bansal, S., Singh, A. K. and Gupta, N., 2015. Golomb Ruler Sequences Optimization for FWM Crosstalk Reduction: Multi–population Hybrid Flower Pollination Algorithm. Progress in Electromagnetics Research Symposium (PIERS), Prague, Czech Republic, pp. 2463–2467.

[33] Ritu, Bansal, S., and Sharma, S., 2016.A Novel Bat Algorithm for Channel Allocation to Reduce FWM Crosstalk in WDM Systems.International Journal of Computer Applications (IJCA),Vol. 136, No. 4, pp. 33-42.

[34] Thing, Vrizlynn L. L., Rao, M. K. and Shum, P. 2003. Fractional Optimal Golomb Ruler Based WDM Channel Allocation. In Proceedings of the 8th Opto-Electronics and Communication Conference (OECC–2003), Vol. 23, pp. 631-632.

[35] Shearer, James B. 1998. Some New Disjoint Golomb Rulers. IEEE Transactions on Information Theory, Vol. 44, No. 7, pp. 3151–3153

[36] Galinier, Jaumard, Morales, and Pesant G. 2001. A constraint–Based Approach to the Golomb Ruler Problem. In Proceeding of 3rd International workshop on integration of AI and OR techniques (CP-AI-OR 2001).

[37] Leitao, Tiago 2004. Evolving the Maximum Segment Length of a Golomb Ruler. Genetic and Evolutionary Computation Conference, USA.

[38] Rankin, William T. 1993. Optimal Golomb Rulers: An exhaustive parallel search implementation. M.S. thesis, Duke University, (Availabe at http://people.ee.duke.edu/~wrankin/golomb/golomb.html).

[39] Storn, R., and Price, K. V. 1997. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. Journal of Global Optimization, Vol. 11, No. 4, pp. 341–359.

[40] Price, K., Storn, R. and Lampinen. J. 2005. Differential Evolution–A Practical Approach to Global Optimization. Springer, Berlin, Germany.

[41] Yang, X. S. and Deb, S. 2009. Cuckoo search via Levy flights, in: Proc. of World Congress on Nature & Biologically Inspired Computing (NaBic 2009), IEEE Publications, USA, pp. 210-214.

[42] Yang, X.-S., and Deb, S. 2010. Optimization by Cuckoo Search. Int. J. Mathematical Modelling and Numerical Optimisation. Vol. 1, No. 4, pp. 330–343.

[43] Yang, X.-S., 2012. .Flower Pollination Algorithm for Global Optimization: in Unconventional Computation and Natural Computation. Lecture Notes in Computer Science, Vol. 7445, Springer, Berlin, pp. 240–249.

[44] Cotta, C., Dotú, I., Fernández, A.J. and P. V. Hentenryck, P.V. 2006. A Memetic Approach to Golomb Rulers:Parallel Problem Solving from Nature–PPSN IX, Lecture Notes in Computer Science, Vol. 4193, pp. 252–261.

[45] Bansal, S. 2011. Golomb Ruler Sequences Optimization: Soft Computing Approaches. M.Tech. Thesis, Department of Electronics and Communication Engineering, Maharishi Markandeshwar Engineering College, Deemed University, Mullana.

[46] Dotú, I., Hentenryck, P.V. 2005. A Simple Hybrid Evolutionary Algorithm for Finding Golomb Rulers. Evolutionary Computation, The 2005 IEEE Congress on, Vol. 3, pp. 2018–2023.