

A Novel Soft-Computing Algorithm for Channel Allocation in WDM Systems

Shonak Bansal

Department of Electronics and Communication Engineering, Institute of Science and Technology, Klawad, Haryana, INDIA

Kuldeep Singh

Department of Electronics and Communication Engineering, Institute of Science and Technology, Klawad, Haryana, INDIA

ABSTRACT

Nature-inspired algorithms are the most powerful algorithms for optimization problems. This paper presents a novel optimization channel allocation algorithm inspired by the flash pattern of fireflies that allows suppression of the four-wave mixing (FWM) crosstalk while maintaining channel bandwidth. It is composed of a fractional bandwidth channel allocation algorithm by using the concept of Optimal Golomb ruler (OGR) sequences. The simulation results conclude that the proposed novel optimization algorithm outperforms the other two existing conventional algorithms i.e. Extended Quadratic Congruence (EQC) and Search Algorithm (SA) in terms of the total optical bandwidth.

General Terms

Four wave mixing, Conventional and Soft computing, Nature-inspired, Optimization.

Keywords

Channel Spacing, Optimal Golomb ruler, Firefly Algorithm, Equally and Unequally spaced channel allocation.

1. INTRODUCTION

FWM crosstalk is an optical Kerr effect and is one of the major limiting sources of performance degradation in all optical wave length division multiplexing (WDM) systems. FWM crosstalk is due to the interaction of two or more channels which results in ghost channels. These ghost channels can coincide with other channels resulting in distortion. The performance of WDM systems can be substantially improved if the generation of FWM crosstalk signal at the channel frequencies is avoided. It is therefore important to develop algorithms to allocate the channel frequencies in order to suppress the FWM crosstalk. The efficiency of FWM crosstalk signals depends on both the channel spacing and fiber dispersion [1]–[3]. To suppress the crosstalk due to FWM signals the frequency separation of any two channels of an optical WDM system is kept different from that of any other pair of channels [4]–[7].

In literature [1], [8]–[14], to suppress the FWM crosstalk in optical WDM systems, several unequally spaced channel allocation (USCA) algorithms have been proposed. An optimum-USCA (O-USCA) algorithm ensures if the frequency separation of any two channels is different from any other pair of channels no FWM crosstalk will ever be generated at any of the channel frequencies [11]. However, the algorithms [8]–[14] have the drawback of increased optical bandwidth requirement compared to equally spaced channel allocation (ESCA). This paper proposes, a fractional bandwidth channel allocation algorithm taking into consideration the concept of Optimal Golomb ruler sequences [7], [15]–[17]. This method for USCA achieves suppression in FWM crosstalk with the optical WDM systems without

inducing additional cost in terms of total optical channel bandwidth.

Golomb rulers represent a class of problems known as NP-complete [18]. Unlike the traveling salesman problem (TSP), which may be classified as a *complete ordered set*, the Golomb ruler may be classified as an *incomplete ordered set*. For higher order models, the exhaustive search [19], [20] of such NP-complete problems is impossible. As another mark is added to the ruler, the time required to search the permutations and to test the ruler becomes exponentially larger. Several different algorithms to tackle the Golomb ruler problem such as exact methods [19], [20], constraint programming [21], local searches [22] and exhaustive parallel search [23] have been studied. The success of soft computing algorithms such as Genetic Algorithms (GAs) [24]–[27] and Biogeography Based Optimization (BBO) [28]–[30] and Big Bang–Big Crunch (BB–BC) evolution theory [31], [32] in finding relatively good solutions to such NP-complete problems provides a good starting point for algorithms of finding OGR sequences. Hence, soft computing based algorithms seem to be very effective solutions for such problems. No doubt, these algorithms do not give the best/exact solutions but reasonably good solutions are available at given cost. This paper introduces a novel algorithm of generating OGR sequences for various marks making use of Firefly Algorithm (FA) and its comparison with two existing conventional/classical algorithms i.e. Extended Quadratic Congruence and Search Algorithm. The generated OGR sequences can be used as a bandwidth allocation algorithm in WDM systems.

The remainder of this paper is organized as follows: Section II introduces the concept of Golomb rulers. Section III presents the problem formulation using FA. Section IV describes a brief account of FA optimization algorithm. Section V provides simulation results comparing with conventional/classical algorithms of generating unequal channel spacing. Section VI presents some concluding remarks.

2. GOLOMB RULERS

The concept of *Golomb rulers* was first introduced by W. C. Babcock [7], and further described by Professor Solomon W. Golomb [15]. According to Colannino [33] and Dimitromanolakis [34], W. C. Babcock was first who discovered Golomb ruler's upto 10-marks, while analyzing positioning of radio channels in the frequency spectrum to suppress the third and fifth order distortion. According to William T. Rankin [35], all of rulers' upto eight are optimum, the nine and ten mark rulers that were presented by W. C. Babcock are near-to-optimum.

Golomb ruler refers to a set of positive integers such that no distinct pairs of numbers from the set have the same difference [36], [37]. These numbers are referred to as *marks*

that correspond to positions on a linear scale [15], [24], [28], [29]. The difference between the values of any two marks is called the *distance* between those marks. The difference between the largest and smallest number is referred to as the *length* of the ruler. The number of marks on a ruler is sometimes referred to as the *size* of the ruler. Normally the first mark of the ruler [15], [16], [22] is set on position 0. Figure 1 [24] shows an example of Golomb ruler in which all the distances between each pair of marks are also shown.

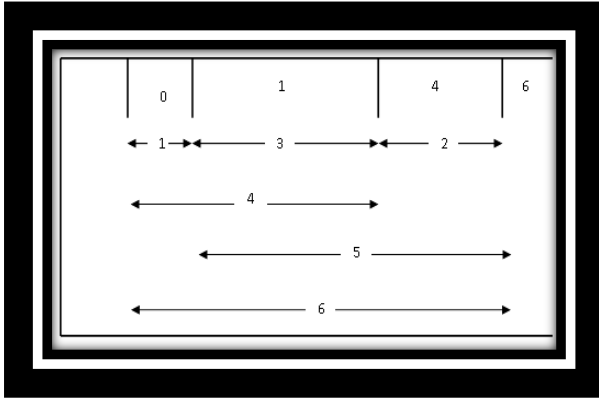


Figure 1: A Golomb Ruler with 4-Marks and Length 6

A *Perfect Golomb ruler* measures all the integer distances from 0 to L , where L is the length of the ruler [18], [24], [25]. In other words, the difference triangle of a perfect Golomb ruler contains all numbers between one and the length of the ruler. The length [23] of an n -mark perfect Golomb ruler is $\frac{1}{2}n(n-1)$.

An *Optimal Golomb ruler* is defined as the shortest length ruler for a given number of marks [24], [38], [39]. There can be multiple different OGRs for a specific number of marks.

For example, as shown in Figure 2 the set (0, 1, 3, 7) is a non-optimal 4-mark Golomb ruler since its differences are (1 = 1 - 0, 2 = 3 - 1, 3 = 3 - 0, 4 = 7 - 3, 6 = 7 - 1, 7 = 7 - 0), all of which are distinct. As from the differences it is clear that the number 5 is missing so it is not a perfect Golomb ruler sequence.

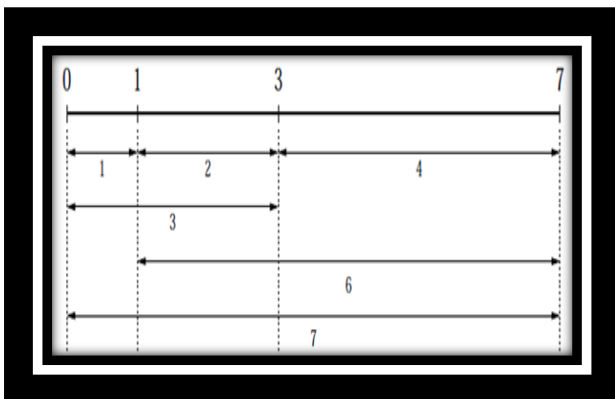


Figure 2: A Non Optimal Golomb Ruler of 4-Marks and Length 7

However, the unique optimal Golomb 4-mark ruler is (0, 1, 4, 6), which measures all the integer distances from 0 to 6 (and is therefore also a perfect ruler) as shown in Figure 1.

The OGRs found their applications in Communications and Radio Astronomy, X-Ray Crystallography, Coding Theory,

Linear Arrays, Computer Communication Network, PPM Communications, Circuit Layout, Geographical Mapping and Self-Orthogonal Codes [7], [15], [24], [25], [35].

Since the difference between any two numbers is distinct, the new FWM frequencies generated would not fall into the one already assigned for the carrier channels. For n -channels, Golomb ruler for n -marks is used.

3. PROBLEM FORMULATION

If the spacing between any pair of channels is denoted as CS and the total number of channels is n , then the objective is to optimize the length of the ruler denoted as RL , which is given by the equation (1):

$$RL = \sum_{i=1}^{n-1} (CS)_i \quad (1)$$

subject to $(CS)_i \neq (CS)_j$

where $i, j = 1, 2, \dots, n-1$ with $i \neq j$ are distinct.

If each individual element is a Golomb ruler, the sum of all elements of an individual forms the total optical bandwidth of the channels. Thus, if an individual element is denoted as (IE) , then the second objective is to minimize the total optical bandwidth (TBW), which is given by the equation (2):

$$TBW = \sum_{i=1}^n (IE)_i \quad (2)$$

subject to $(IE)_i \neq (IE)_j$

where $i, j = 1, 2, \dots, n$ with $i \neq j$ are distinct.

To generate optimal Golomb ruler sequences, firefly algorithm is being proposed in this paper.

4. FIREFLY ALGORITHM

This section outlines the capabilities of a novel soft computing algorithm called firefly algorithm for the generation of OGR sequences. Xin-She Yang [40], inspired by the characteristics and flash pattern of fireflies, introduced a novel optimization algorithm named firefly algorithm. For describing this novel algorithm the author in [40] use the following three idealized rules:

- 1) All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex;
- 2) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly;
- 3) The brightness of a firefly is affected or determined by the landscape of the objective function. For a maximization problem, the brightness can simply be proportional to the value of the objective function.

There are two important issues in firefly algorithm:

- 1) the variation of light intensity and
- 2) the formulation of attractiveness.

For simplicity, it was assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function [40]–[43].

In the simplest case for maximum optimization problems, the brightness I of a firefly at a particular location X can be chosen as $I(X) \propto f(X)$.

The attractiveness β should be judged by the other fireflies that will vary with the distance r_{ij} between firefly i and firefly j . In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so we should allow the attractiveness to vary with the degree of absorption. In the simplest form, the light intensity $I(r)$ varies according to the inverse square law as given by equation 3 [40]:

$$I(r) = \frac{I_s}{r^2} \quad (3)$$

where I_s is the intensity at the source. The light intensity I for a given medium varies with the distance r between any two fireflies as given by equation 4 [40]:

$$I = I_0 e^{-\gamma r} \quad (4)$$

where I_0 is the original light intensity and γ is fixed light absorption coefficient.

The firefly's attractiveness is proportional to the light intensity seen by the adjacent fireflies. Therefore the attractiveness β of a firefly is given by equation 5 [40]:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (5)$$

where β_0 is the attractiveness at $r = 0$.

The distance between any two fireflies i and j at X_i and X_j , respectively, is the Cartesian distance as given by equation 6 [40]:

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (6)$$

where $x_{i,k}$ is the k th component of the spatial coordinate X_i of i th firefly and d is the number of dimensions.

The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by the equation 7 [40]:

$$X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2} (X_j - X_i) + \alpha(\text{rand} - 0.5) \quad (7)$$

where the second term is due to the attraction. The third term is randomization with a control parameter α , which makes the more efficient exploration of the search space. For most cases in the implementation, $\beta_0 = 1$ and $\alpha \in [0, 1]$. The general pseudo-code for firefly algorithm [40]–[43] is shown in Figure 3.

4.1 Firefly Algorithm to Generate Optimal Golomb Ruler Sequences

The proposed pseudo-code for firefly algorithm to generate OGR sequences in this paper is shown in Figure 4.

5. SIMULATION RESULTS AND DISCUSSION

The firefly algorithm to generate OGR sequences has been written and tested in Matlab-7 language [44] under Windows 7 operating system. The algorithm has been executed on Laptop with Intel core2 Duo 2.20 GHz processor with a RAM of 3 GB. This section is devoted to the performance of proposed firefly algorithm to generate OGRs and its comparison with existing known OGR [15], [20], [33], [35], [39], [45], [46] and two of the existing classical algorithms of generating unequal channel spacing i.e. EQC and SA [1], [13], [24].

5.1 Simulation Parameters for Firefly Algorithm

To get optimal solutions after a number of careful experimentation, following optimum parameter values of FA have finally been settled as shown in Table 1.

With these parameters settings, the large numbers of sets of trials for various marks were conducted. A set of 10 trials for $n = 4$ to 6 are given in Table 2. The performance of all the sets is nearly the same as given in Table 2.

Firefly Algorithm

Begin

/* FA parameter initialization */

Define objective function $f(X)$; $X = (x_1, \dots, x_d)^T$;

Generate initial population of fireflies x_i ($i = 1, 2, \dots, n$);

Compute the light intensity I_i at X_i by $f(X_i)$;

Define light absorption coefficient γ ;

/* End of FA parameter initialization */

While not T

/* T is a termination criterion */

For $i = 1 : n$

/* all n fireflies */

For $j = 1 : i$

If $(I_j > I_i)$

Move firefly i towards j in d -dimension;

End if

Attractiveness varies with distance r via $\exp[-\gamma r]$;

Evaluate new solutions and update light intensity;

End for

/* End for j */

End for

/* End for i */

Rank the fireflies and find the current best;

End while

Postprocess results and visualization;

End

Figure 3: Pseudo-code for Firefly Algorithm

Table 1. Simulation Parameters for Firefly Algorithm

Parameter	Value
Number of fireflies (<i>Popsi</i> ze)	10
α	0.05
β	0.2
Γ	1

5.2 Sequences

The OGR sequences generated by proposed algorithm for various marks are shown in Appendix–A.

5.3 Influence of Increasing Iterations on Total Bandwidth

With the increase of number of iterations, the total bandwidth occupied by the sequence tends to decrease; it means that the rulers reach their optimum values after a certain number of iterations. This means, it is the point where we are approaching towards the optimal solution. This can be seen in tabular form for proposed FA in Table 3 for $n = 3$ to 10.

It is noted that the iterations has little effect for low value marks such as $n = 3$ and $n = 4$. But for higher order marks, the iterations have a great effect on the total bandwidth. It means bandwidth gets optimized after a certain numbers of iterations. By carefully observation, the paper fixed the iterations of 100 for firefly algorithm.

5.4 Comparison of Proposed Approach with Previous Existing Algorithms in terms of Ruler Length and Total Bandwidth

The objective to use FA in this paper was to optimize the ruler length (*RL*) so as to conserve the total bandwidth (*TBW*) occupied by the channels. Table 3 enlist the ruler length and total bandwidth occupied by different sequences obtained by the proposed firefly algorithm for various channels n and its comparison with known OGR [15], [20], [33], [35], [39], [45], [46], EQC and SA [1], [13], [24]. All results have been obtained after a set of 10 trials. The numbers of iterations and relative error in the length of the ruler by the proposed firefly algorithm are also represented in Table 3.

Firefly Algorithm

Begin

/ FA parameter initialization */*

Define operating parameters for firefly algorithm

Initialize the number of channels (marks), lower and upper bound on the length of ruler;

While not *Popsi*ze */* Popsi*ze is the population size input by the user */

Generate a random set of firefly (integer population);

/ Number of integers in firefly is being equal to the number of channels */*

Check Golombness of each firefly;

If Golombness is satisfied

Retain that firefly;

Else

Remove that particular firefly from the generated population;

End if

End while

Compute the light intensity (total bandwidth);

/ Light intensity represents the fitness (cost) value */*

Rank the fireflies from best to worst based on fitness value;

/ End of FA parameter initialization */*

While not T */* T is a termination criterion */*

/ Movement */*

For $i = 1 : n$ */* all n fireflies */*

For $j = 1 : i$

If ($I_j > I_i$)

A: Move firefly i towards the brighter firefly j ;

Recheck Golombness of updated firefly;

If Golombness is satisfied

Retain that firefly and then go to B;

Else

Retain the previous generated firefly and then go to A;

/ Previous generated firefly is being equal to the firefly generated into the FA parameter initialization step */*

End if

End if

B: Vary attractiveness β with distance r via $\exp[-\gamma r]$;

Determine the new position of each firefly and update light intensity;

End for */* End for j */*

End for */* End for i */*

/ End of Movement */*

Rank the fireflies from best to worst based on fitness value and find the current best;

End while

Display the optimal Golomb ruler sequences;

End

Figure 4: Pseudo-code for FA to Generate OGR Sequences

Table 2. Performance of Firefly Algorithm for Different Marks in a Set of 10 Trials

Trials	<i>n</i> = 4		<i>n</i> = 5		<i>n</i> = 6	
	Ruler Length	Total Bandwidth (Hz)	Ruler Length	Total Bandwidth (Hz)	Ruler Length	Total Bandwidth (Hz)
1	6	11	11	25	18	42
2	6	11	12	23	17	44
3	7	11	11	25	17	50
4	7	11	11	25	17	44
5	6	11	13	24	18	46
6	6	11	12	23	17	44
7	6	11	12	24	18	48
8	6	11	13	24	18	42
9	7	11	12	23	18	42
10	6	11	11	25	17	44
Optimal Ruler Length = 6 Optimal Total Bandwidth (Hz)= 11			Optimal Ruler Length =11 Optimal Total Bandwidth (Hz)= 23		Optimal Ruler Length =17 Optimal Total Bandwidth (Hz)= 42	

Wing et. al. [1] noted that the application of EQC and SA was limited to prime powers, so in Table 3, the ruler length and total bandwidth for EQC and SA are shown by a dash line.

Comparing the simulation results of proposed FA with best known OGR, EQC and SA; it is observed that there is a significant improvement with respect to the length of the ruler and thus the total bandwidth occupied by the use of firefly algorithm. Figure 5 and Figure 6 illustrate the comparison of FA to generate OGR sequences with known OGR, EQC and SA in terms of the ruler length and total optical bandwidth occupied by the sequences for $n = 3, 4, 6$ and 8 respectively. It

is also clear from Table 3 that the proposed algorithm can find near-optimal solutions for larger rulers but at the expense of a significant computational time.

The results obtained for rulers upto 6-marks can be performed in few seconds whereas minutes were necessary for other instances. From Table 3 it is clear that the proposed FA was able to generate the shortest ruler in $n = 3, 4, 5, 6, 7, 8$ and 10 . In $n = 9$ the FA generate the ruler that is 5 unit longer than the shortest best known solutions. But the total optical bandwidth occupied by $n = 9$ is same as that of the best known solutions.

Table 3. Comparison of Ruler Length and Total Bandwidth Obtained by FA with Known OGR, EQC and SA

<i>n</i>	Best Known OGR [15], [20], [33], [35], [39], [45], [46]		EQC [1], [13], [24]		SA [1], [13], [24]		FA (Proposed Algorithm)			
	Ruler Length	TBW (Hz)	Ruler Length	TBW (Hz)	Ruler Length	TBW (Hz)	Ruler Length	TBW (Hz)	Iterations	Relative Error (%)
3	3	4	6	10	6	4	3	4	1	0.0
4	6	11	15	28	15	11	6 7	11	2	0.0
5	11	25 28	—	—	—	—	11 12 13	23 24 25	3	0.0
6	17	44 47 50 52	45	140	20	60	17 18	42 44 46 47 50 52	5	0.0
7	25	81 87 95 77 90	—	—	—	—	25 26 27	73 77 80 81	8	0.0
8	34	117	91	378	49	189	34 39	113 117	50	0.0
9	44	206	—	—	—	—	49	206	70	11.36
10	55	249	—	—	—	—	55	249	100	0.0

It is pertinent to mention here that OGR obtained in [25] by non-heuristic exhaustive searches for 10-marks took 12.57 minutes, whereas for the our proposed approach the execution time varied from few seconds for rulers of 6-marks to 15 minutes for 10-marks ruler with a maximum iterations of 100. Of course computers today play dominant role in reducing the computation time; efficient algorithms can further reduce the time complexity drastically by substituting best for reasonably good solutions.

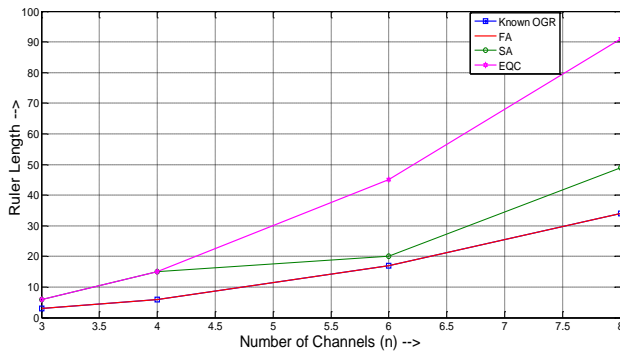


Figure 5: Comparison of the Results Obtained By Proposed Firefly Algorithm with Known OGR, EQC and SA in Terms of Length of the Ruler

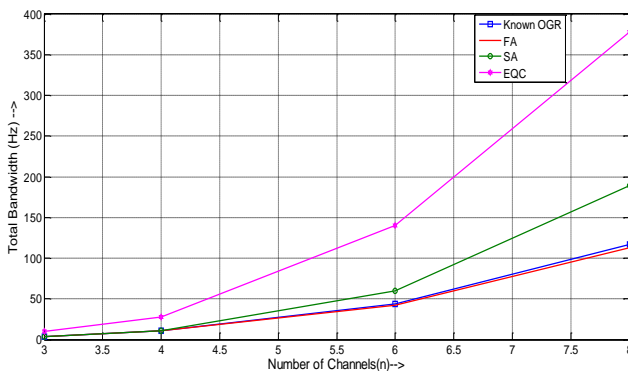


Figure 6: Comparison of the Results Obtained By Proposed Firefly Algorithm with Known OGR, EQC and SA in Terms of Total Optical Bandwidth

6. CONCLUSION

This paper presented the application of FA to solve optimal Golomb ruler sequences problem. It has been observed that FA produces Golomb ruler sequences very efficiently. The performance is being compared with the two existing classical algorithm i.e. EQC and SA in terms of the ruler length and total bandwidth obtained by the sequences. The preliminary results indicate that FA out performs the existing algorithms in terms of total optical bandwidth i.e. FA appears to be most efficient algorithm to such NP-complete problems.

In the future, in order for these algorithms to be of practical use, it is desired that the performance of these algorithms for higher mark OGRs upto about 3000 marks may be evaluated. Though this process will be very time consuming yet this needs be done for this work to be of some use in the field of communication engineering.

7. REFERENCES

- [1] Kwong, Wing 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, pp. 352–359.
- [2] Chraplyvy, A. R. 1990. Limitations on Lightwave Communications Imposed by Optical-Fiber Nonlinearities. J. Lightwave Technol., Vol. 8, pp. 1548–1557.
- [3] Aggarwal, G.P. 2001. Nonlinear Fiber Optics. Second Edition, Academic Press, San Diego.
- [4] Thing, Vrizlynn L. L., Shum, P., and Rao, M. K. 2004. Bandwidth-Efficient WDM Channel Allocation for Four-Wave Mixing-Effect Minimization. IEEE Transactions on Communications, Vol. 52, No. 12, pp. 2184–2189.
- [5] Saaied, Nordiana M. 2010. Nonlinear Optical Effects Suppression Methods in WDM Systems with EDFAs: A Review. In Proceedings of the International Conference on Computer and Communication Engineering (ICCE 2010), (11–13 May 2010), Kuala Lumpur, Malaysia.
- [6] Forghieri, F., Tkach, R. W., Chraplyvy, A. R., and Marcuse, D. 1994. Reduction of Four-Wave Mixing Crosstalk in WDM Systems Using Unequally Spaced Channels. IEEE Photonics Technology Letters, Vol. 6, No. 6, pp. 754–756.
- [7] Babcock, W. C. 1953. Intermodulation interference in radio systems, Bell Systems Technical Journal, pp. 63–73.
- [8] 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.
- [9] Forghieri, F., Tkach, R. W., and Chraplyvy, A. R. 1995. WDM systems with unequally spaced channels. J. Lightwave Technol., Vol. 13, pp. 889–897.
- [10] Hwang, B. and Tonguz, O. K. 1998. A generalized suboptimum unequally spaced channel allocation technique—Part I: In IM/DDWDMsystems. IEEE Trans. Commun., Vol. 46, pp. 1027–1037.
- [11] 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.
- [12] 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.
- [13] Randhawa, R., Sohal, J. S. and Kaler, R.S. 2009. Optimum Algorithm for WDM Channel Allocation for Reducing Four-Wave Mixing Effects. Optik 120, pp. 898–904.
- [14] http://www.compunity.org/events/pastevents/ewomp2004/jaillet_krajecki_pap_ew04.pdf
- [15] 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.
- [16] 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.

- [17] Shearer, James B. 1998. Some New Disjoint Golomb Rulers. *IEEE Transactions on Information Theory*, Vol. 44, No. 7, pp. 3151-3153.
- [18] <http://theinfl.informatik.uni-jena.de/teaching/ss10/oberseminar-ss10>
- [19] Robinson, Johan P. 1990. Optimum Golomb Rulers. *IEEE Transactions on Computers*, Vol. C-28, No. 12, (December 1979).
- [20] James B. Shearer. Some New Optimum Golomb Rulers. *IEEE Transactions on Information Theory*. IT-36, pp. 183-184.
- [21] 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)*.
- [22] Leitaio, Tiago 2004. Evolving the Maximum Segment Length of a Golomb Ruler. *Genetic and Evolutionary Computation Conference, USA*.
- [23] Rankin, William T. 1993. Optimal Golomb Rulers: An exhaustive parallel search implementation. M.S. thesis, Duke University, (Available at <http://people.ee.duke.edu/~wrankin/golomb/golomb.html>).
- [24] 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.
- [25] Soliday, Stephen W., Homaifar, A. and Lebby, Gary 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.
- [26] Robinson, John P. 2000. Genetic Search for Golomb Arrays. *IEEE Transactions on Information Theory*, Vol. 46, No. 3, pp. 1170-1173.
- [27] Ayari, N., Thé Van Luong and A. Jemai. 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] Colannino, J. 2003. Circular and Modular Golomb Rulers. URL:<http://cgm.cs.mcgill.ca/~athens/cs507/Projects/2003/JustinColannino/>.
- [34] 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.
- [35] Dollas, A., Rankin, William T., and McCracken, D. 1998. A New Algorithm for Golomb Ruler Derivation and Proof of the 19 Mark Ruler. *IEEE Transactions on Information Theory*, Vol. 44, No. 1, pp. 379-382.
- [36] "Project OGR", <http://www.distributed.net/OGR>.
- [37] Cotta, C., Dotu, I., Fernandez, Antonio J., and Hentenryck, Pascal V. 2007. Local Search-Based Hybrid Algorithms for Finding Golomb Rulers. *Kluwer Academic Publishers, Boston*, Vol. 12, Issue 3, pp. 263-291.
- [38] <http://mathworld.wolfram.com/PerfectRuler.html>
- [39] <http://mathworld.wolfram.com/GolombRuler.html>
- [40] X.-S. Yang 2009. Firefly Algorithms for Multimodal Optimization. In: *Stochastic Algorithms: Foundations and Applications, SAGA-2009, Lecture Notes in Computer Science*, 5792, Springer-Verlag, Berlin, pp. 169-178.
- [41] X.-S. Yang 2010. Firefly Algorithm, Stochastic Test Functions and Design Optimization. *Int. J. Bio-Inspired Computation*, Vol. 2, No. 2, pp.78-84.
- [42] X.-S. Yang and S. Deb 2010. Eagle Strategy Using Levy Walk and Firefly Algorithms for Stochastic Optimization. In: *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* (Eds. J. R. Gonzalez et al.), *Studies in Computational Intelligence*, Springer Berlin, 284, pp. 101-111.
- [43] X.-S. Yang 2010. Firefly algorithm, levy flights and global optimization. *Research and Development in Intelligent Systems XXVI* (Eds M. Bramer, R. Ellis, Petridis), Springer London, pp 209-218.

[44] Pratap R. 2010. Getting Started with Matlab A Quick Introduction for Scientists and Engineers. Oxford University Press, New York.

[45] <http://www.research.ibm.com/people/s/shearer/grtab.html>

[46] <http://www.research.ibm.com/people/s/shearer/gropt.html>

APPENDIX–A

The table below illustrates the optimal Golomb ruler sequences generated by the proposed FA for various marks:

Table 4. Optimal Golomb Ruler Sequences Generated By Proposed FA

<i>Order</i>	<i>Length</i>	<i>Marks</i>
1	0	0
2	1	0 1
3	3	0 1 3
4	6	0 1 4 6

	7	0 1 3 7
5	11	0 1 4 9 11
	12	0 1 3 7 12
	12	0 1 3 8 12
	13	0 1 4 6 13
6	17	0 1 4 10 12 17
	17	0 1 4 10 15 17
	17	0 3 5 9 16 17
	17	0 4 6 9 16 17
	18	0 1 3 8 12 18
	18	0 1 3 8 14 18
	18	0 1 5 7 15 18
7	25	0 2 6 9 14 24 25
	25	0 1 4 10 18 23 25
	26	0 1 7 9 12 22 26
	27	0 1 5 7 15 18 27
8	34	0 1 4 9 15 22 32 34
	39	0 1 3 8 14 18 30 39
9	49	1 5 11 12 20 33 36 38 50
10	55	0 1 6 10 23 26 34 41 53 55