Golomb Ruler Sequence Generation and Optimization using Modified Firefly Algorithm

Yogita Wadhwa[1], Parvinder Kaur[2], Baljeet Kaur[3]
Department of Electronics and Communication, MMU, Mullana, Haryana, India[1,2]
Department of Electronics and Communication, SSIET, Derabassi, Punjab, India[3]

Abstract- Nature-inspired algorithms are among the most powerful algorithms for optimization problems. This paper intends to provide a detailed description of a Modified Firefly Algorithm (FA) Approach for Golomb Ruler Sequence Generation and optimization that allows suppression of the four wave mixing (FWM) crosstalk while maintaining channel bandwidth. We will compare the proposed modified firefly algorithm with other hard computing and soft computing algorithms such as Extended Quadratic Congruence (EQC), Search Algorithm (SA), Genetic Algorithm (GA), Biogeography Based Optimization (BBO), Firefly Algorithm (FA). Simulations results indicate that the proposed modified firefly algorithm is superior to existing hard computing and soft computing algorithms in terms of computational complexity.

Keywords-Four wave Mixing (FWM), Optimal Golomb Ruler (OGR), Firefly Algorithm (FA), Modified Firefly Algorithm (MFA)

I. INTRODUCTION

In optical WDM systems, channels are usually assigned with center frequencies (or wavelength) equally spaced from each other. Due to equal spacing among the channels there is very high probability that FWM signals may fall into the WDM channels, resulting in severe crosstalk [1].

FWM crosstalk is analogous to third–order inter–modulation distortion in silica fiber whereby two or more optical waves at different frequencies (or wavelengths) mix to produce new optical waves at other frequencies [2], [3].

Performance can be substantially improved if FWM crosstalk generation at the channel frequencies is avoided.

In order to suppress the FWM crosstalk in optical WDM systems, several unequally spaced channel allocation (USCA) algorithms have been proposed. However, the algorithms [4]–[10] have the drawback of increased optical bandwidth requirement compared to equally spaced channel allocation (ESCA).

This paper proposes a method for finding the solutions to channel allocation problem by using the concept of Optimal Golomb Rulers (OGR) [11], [12] – [14]. This method for channel allocation achieves reduction in FWM effect with the WDM systems without inducing additional cost in terms of bandwidth.

Golomb Rulers represent a class of problems known as NP – complete [15]. 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*. The exhaustive search [16], [17] of such problems is impossible for higher order models. As another mark is added to the ruler, the time required to search the permutations and to test the ruler becomes exponentially greater. The success of Soft Computing approaches such as Genetic Algorithms (GAs) [18] – [20] Biogeography based Optimization (BBO) [21], Firefly Algorithm (FA) [22] in finding relatively good solutions to NP – complete problems provides a good starting point for methods of finding Optimal Golomb Ruler sequences. Hence, soft computing approaches seem to be very effective solutions for the NP – complete problems. No doubt, these approaches do not give the exact or best solutions but reasonably good solutions are available at given cost. In this paper, a optimization algorithm based on the behavior of firefly called Modified Firefly Algorithm is being applied to generate the optimal Golomb Ruler Sequences for various marks.

The remainder of this paper is organized as follows: Section II introduces the concept of Golomb Rulers. Section III presents the problem formulation. Section IV describes a brief introduction about MFA and steps to generate the Golomb Ruler sequences by using proposed approach. Section V provides comparison of simulation results with conventional classical approaches of generating unequal channel spacing i.e. Extended Quadratic Congruence (EQC) and Search Algorithm (SA) and existing soft computing approaches. Section VI presents some concluding remarks.

II. GOLOMB RULERS

The idea of Golomb Rulers‘ was first introduced by W.C. Babcock [11] in 1952, and further derived in 1977 from the relevant work by Professor Solomon W. Golomb [12], a professor of Mathematics and Electrical Engineering at the University of Southern California. According to Colannino [23] and Dimitromanolakis [24], W. C. Babcock [11] first discovered Golomb Rulers up to 10– marks, while analyzing positioning of radio channels in the frequency spectrum. He investigated inter–modulation distortion appearing in consecutive radio bands and observed that when positioning each pair of channels at a distinct distance, then third order distortion was eliminated and fifth order distortion was lessened greatly. According to William T. Rankin [25], all of rulers‘ upto eight are optimum, the nine and ten mark rulers that W. C. Babcock presents are near optimum.

The term ‘‘Golomb ruler’’ [26] refers to a set of nonnegative integer values, such that any two different pairs of numbers from the set have not the same difference. It is similar to a ruler constructed in a way that no two pairs of marks measure the same distance. An example of the Golomb ruler is shown in Fig. 1. [9] An Optimal Golomb Ruler is the shortest ruler possible for a given number of marks [5]. Therefore applying OGR to the channel allocation problem, it is possible to achieve the smallest distinct number to be used for the channel allocation. 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.

An n-mark Golomb ruler is a set of n distinct non negative integers (a1, a2, .., an) called marks, such that the positive differences |ai-aj|, computed overall possible pairs of different i, j = 1,… n with i≠j are distinct. Let an be the largest integer in an n-mark Golomb ruler. Then an OGR with n marks (0,...., an) is an n-mark Golomb ruler if

1. There exists no other n-mark Golomb ruler having smaller largest mark an, and

2. The ruler is written in a canonical form as the‘smaller’ of the equivalent rulers (0, a2, …, an) and(0,...an–a2,an), where smaller means the first differing entry is less than the corresponding entry in the other ruler.

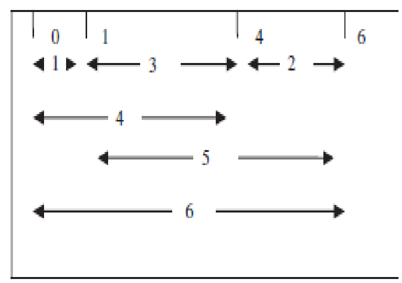

Figure 1: Perfect Golomb Ruler.

III. PROBLEM FORMULATION

To prevent FWM signals from falling onto the channels of an unequal–spaced WDM system, the frequency separation between any two channels must be distinct [1], [18]. This unequal channel spacing can be achieved by a mathematical term called ‘Golomb rulers’. So, the first problem formulation is to generate the optimal Golomb ruler sequences with distinct pairs of numbers.

The second problem formulation is to obtain the Optimal Golomb rulers (OGRs) by optimizing (minimizing) the length of the ruler and hence the total bandwidth occupied by the channels. Thus, if the spacing between any pair of channels is denoted as *CS* and the total number of channels is *n*, then the objective of this dissertation is to minimize the length of ruler denoted as RL, where RL is given as:

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

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

where i, j = 1, 2, …….. , (n-1) with i ≠ j are distinct.

If each individual element is a Golomb ruler, the sum of all elements of an individual forms the total bandwidth of the channels. Thus, if an individual element is denoted as and the total number of elements (called channels) is *n*, then the another objective of this dissertation is to minimize the total bandwidth denoted as BW, which is given by the equation

$$BW = \sum_{i=1}^{n} (\mathrm{IE})_i \quad \ldots\ldots\ldots\ldots.(2)$$

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

where i, j = 1, 2, …….. , (n-1) with i ≠ j are distinct.

IV. SOFT COMPUTING APPROACHES

In this section, the capabilities of a new technique based on the behavior of firefly called MFA for the generation of optimal Golomb Ruler sequences will be discussed

A. Firefly Algorithm

The Firefly Algorithm was developed by the author (Yang 2008, Yang 2009), and it is based on the idealized behavior of the flashing characteristics of fireflies and has demonstrated promising superiority over many other algorithms. The search strategy in this multi-agent algorithm is controlled randomization, efficient local search and selection of the best solutions. However, the randomization typically uses uniform distribution or Gaussian distribution.

Behavior of Fireflies

We know that the light intensity at a particular distance r from the light source obeys the inverse square law. That is to say, the light intensity I decreases as the distance r increases in terms of $I \alpha 1/r^2$. Furthermore, the air absorbs light which becomes weaker and weaker as the distance increases.

These two combined factors make most fireflies visible only to a limited distance, usually several hundred meters at night, which is usually good enough for fireflies to communicate.

The flashing light can be formulated in such a way that it is associated with the objective function to be optimized, which makes it possible to formulate new optimization algorithms.

Now we can idealize some of the flashing characteristics of fireflies so as to develop firefly-inspired algorithms. For simplicity in describing our new Fireflies Algorithm (FA), we now 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 [27].

B. Modified Firefly Algorithm

Modified Firefly Algorithm is very similar to Firefly Algorithm except search strategy. The search strategy in this multi-agent algorithm is controlled randomization, efficient local search as well as global search with parallel implementation of FA Algorithm. This makes Modified Firefly Algorithm much faster than Firefly Algorithm. This algorithm is specially designed for higher order marks as firefly algorithm and other soft computing approaches for higher order marks has high computational complexity and very time consuming, but modified firefly algorithm gives better results in comparatively less time.

Basic steps of the Modified firefly algorithm can be summarized as the pseudo code in Figure 2. All other factors of Modified Firefly Algorithm like behavior of Fireflies, Attractiveness, Distance and Movement are totally similar to Firefly Algorithm.

V. SIMULATION RESULTS

The modified firefly algorithm to generate OGR sequences has been written and tested in Matlab–7 language under Windows 7 operating system. The algorithm has been executed on Laptop with Intel core3 processor with a RAM of 4 GB. This section is devoted to the performance of

Modified 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 *Popsize* /* *Popsize* 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 */
For k= 1: No. of Parallel Iterations
While not *T* /* *T* is a termination criterion */
/* Movement */
For *i* = 1 : *n* /* all *n* fireflies */
For j = 1 : i
If (Ij > Ii)
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 insert new golombness satisfied population 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
End for /* End for *k* */
Display the optimal Golomb ruler sequences;
End

Figure 2: Pseudo Code for MFA to generate OGR Sequences

proposed firefly algorithm to generate OGRs and its comparison with existing known OGR [14] and two of the existing classical algorithms of generating unequal channel spacing i.e. EQC and SA [1], [9].

A. 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.

Table 1: Simulation parameters for Modified Firefly Algorithm

Parameter	Value
Number of Fireflies (Popsize)	10
MaxGeneration (Number of Iterations)	10*10
N (Number of Parallel Iterations)	10
α (Randomness)	0.025
β (Attractiveness)	0.2
γ (Absorption Coefficient)	1

With these parameters settings, Performance of modified firefly algorithm for different marks is shown in Table 2. Table 2 represents the minimum bandwidth after number of trials consumed by the multiplexed signals

* - Number of marks or channels

** - Ruler Length.

n*	RL**	No. of Iterations	BW (Hz)
1	0	100	0
2	1	100	1
3	3	100	4
4	7	100	11
5	12	100	23
6	20	100	42
7	27	100	73
8	34	100	117
9	59	100	193
10	74	100	274

Table 2: Performance of Modified Firefly Algorithm for different marks

B. Influence of increasing Iterations on the performance of MFA

Influence of increasing iterations on the performance of modified firefly algorithm is shown in Figure 3 to Figure 9. All these figures represent that as the number of iterations are increasing, bandwidth is decreasing.

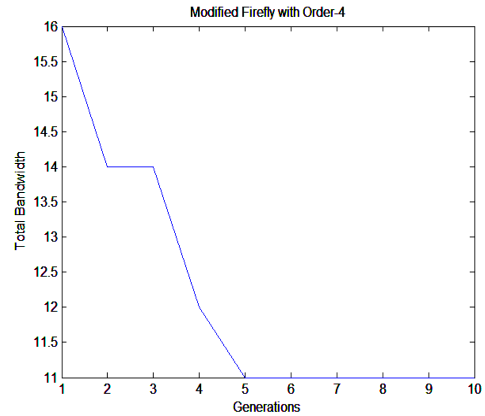

Figure 3: Effect of increasing iterations on performance of MFA for 4-marks

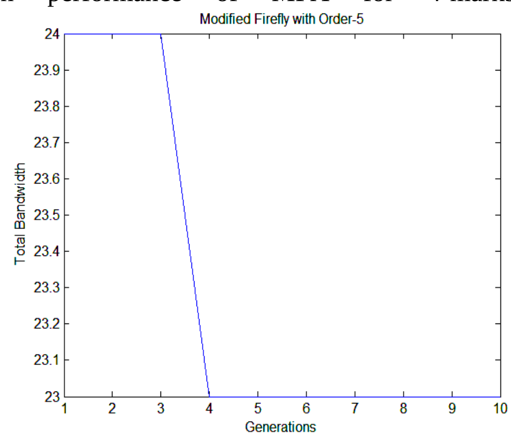

Figure 4: Effect of increasing iterations on performance of MFA for 5-marks.

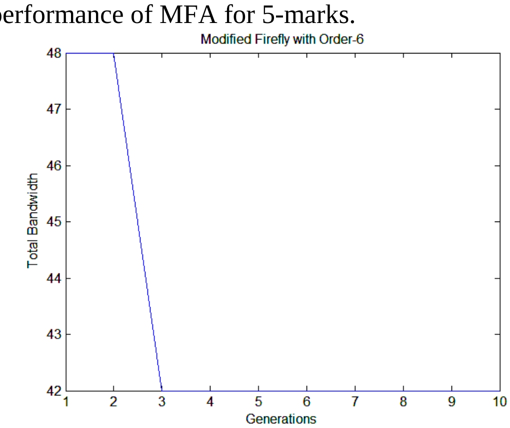

Figure 5: Effect of increasing iterations on performance of MFA for 6-marks.

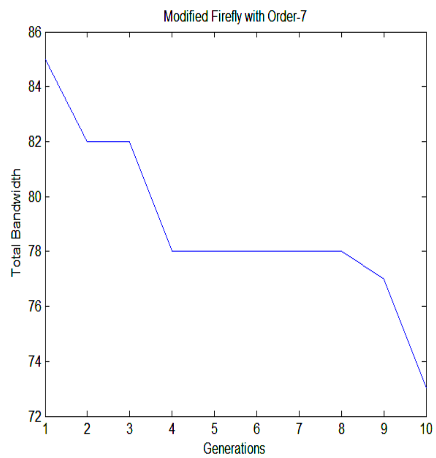

Figure 6: Effect of increasing iterations on performance of MFA for 7-marks.

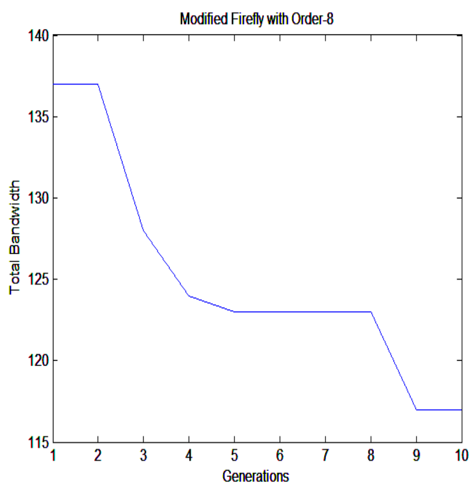

Figure 7: Effect of increasing iterations on performance of MFA for 8-marks.

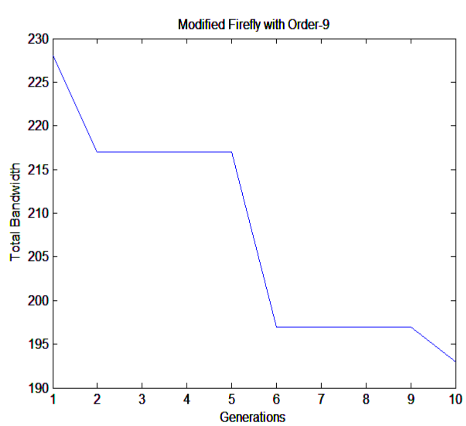

Figure 8: Effect of increasing iterations on performance of MFA for 9-marks

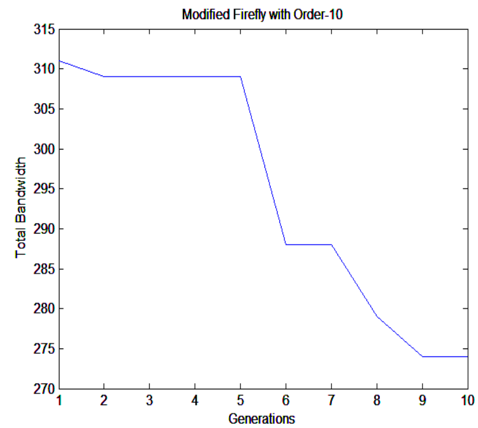

Figure 9: Effect of increasing iterations on performance of MFA for 10-marks

C. Performance Comparison of Modified Firefly Algorithm with Firefly Algorithm

Performance comparison of modified firefly algorithm with firefly algorithm is shown in Table 3. As shown in Table 3, for smaller mark values upto 7 the ruler length and thus bandwidth occupied by MFA and FA is same. But for higher mark values, the ruler length and hence total bandwidth obtained by MFA slightly approaches to their optimal values as compared with FA.

It has also been shown in Table 3 that time consumed to compute optimize result by Modified Firefly Algorithm is much lesser than that of Firefly Algorithm or in other words it can be concluded that MFA has very less computational complexity as compared to FA .

Table 3: Performance Comparison of MFA with FA

n	**RL (FA)**	**BW (FA) Hz**	**Time Consumed by FA**	**RL MFA)**	**BW (MFA) Hz**	**Time Consumed by MFA**
1	0	0	1sec	0	0	1sec
2	1	1	3sec	1	1	1sec
3	3	4	5sec	3	4	1sec
4	6	11	7sec	7	11	2sec
5	12	23	10sec	12	23	5sec
6	18	42	9min	20	42	1min
7	27	73	3hr 40min	27	73	51min
8	35	123	5hr 50min	34	117	2hr 16min
9	56	195	12hr 17min	59	193	5hr 19min
10	75	283	19hr 6min	74	274	11hr 22min

n	Known OGR [28]		EQC [1]		SA [1]		GA [18]		BBO [21]		FA		MFA	
	RL	Total BW (Hz)	RL	Total BW (Hz)	RL	Total BW (Hz)	RL	Total BW (Hz)	RL	Total BW (Hz)	RL	BW (Hz)	RL	BW Hz
3	3	4	6	10	6	4	3	4	3	4	3	4	3	4
4	6	11	15	28	15	11	6	11	6	11	6	11	7	11
5	11	25	-	-	-	-	13	29	12	23	12	23	12	23
6	17	44	45	140	20	60	18	44	18	44	18	42	20	42
7	25	81	-	-	-	-	27	78	27	73	27	73	27	73
8	34	117	91	378	49	189	35	121	34	121	35	123	34	117
9	44	206	-	-	-	-	59	196	56	200	56	195	59	193
10	55	249	-	-	-	-	75	287	74	274	75	283	74	274

Table 4: Performance Comparison of Proposed Algorithm with Previous existing algorithms.

D. Performance Comparison of Proposed Algorithm with Previous existing algorithms.

In this subsection, comparison of the results obtained by MFA with known OGR [28], EQC and SA [1], [18], FA [1], BBO and GA [2] in terms of Ruler Length and Bandwidth is described. Table 4 illustrates the total bandwidth (BW) and length of ruler (RL) occupied by different sequences obtained by proposed algorithm for various channels '*n*' and Figure 10 represents the comparison of MFA with previous existing algorithms in terms of Bandwidth. The aim to use soft computing approach MFA in this dissertation was to optimize the length of the ruler so as to conserve the total bandwidth occupied by the channels.

Wing and Yang [1] noted that the application of EQC and SA is limited to prime powers, so the total bandwidth and ruler length for EQC and SA are shown by a dash line in Table 4.

Comparing the simulation results of MFA with classical approaches i.e. 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 soft computing methods that is, the results gets better.

In Figure 10, lines colour explanation is as follows:

Magenta-OGR; Cyan-EQC; Red-SA; Green-GA Blue-BBO; Yellow-FA; Black-MFA

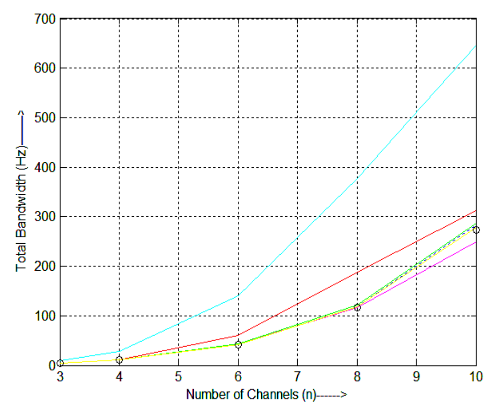

Figure 10: Comparison of MFA with Previous Existing Algorithms

References

[1]. Wing C. Kwong, and Guu-Chang Yang, ―An Algebraic Approach to the Unequal-Spaced Channel-Allocation Problem in WDM Light wave Systems, *IEEE Transactions on Communications*, Vol. 45, No. 3, 352-359, March–1997.

[2]. Andrew R. Chraplyvy, ―Limitations on Light wave Communications Imposed By Optical–Fiber Nonlinearities, *Journal of Light wave Technology*, Vol. 8, No. 10, pp. 1548–1557, October–1990.

[3]. Gerd Keiser, ―Optical Fiber Communications, Third Edition, *McGraw-Hill*, New York, 2000.

[4]. 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.

[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]. 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.

[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]. 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.

[10]. http://www.compunity.org/events/pastevents/ewomp2004/jaillet_krajecki_pap_ew04.pdf

[11]. W. Babcock,Intermodulation interference in radio systems, Bell Systems Technical Journal, pages: 63–73, 1953.

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

[13]. Vrizlynn L. L. Thing, M. K. Rao and P. Shum, ―Fractional Optimal Golomb Ruler Based WDM Channel Allocation‖, The 8th Opto–Electronics and Communication Conference (OECC– 2003), Vol. 23, pp. 631–632, October 2003.

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

[15]. http://theinf1.informatik.uni–jena.de/teaching/ss10/oberseminar–ss10

[16]. Johan P. Robinson, ―Optimum Golomb Rulers, IEEE Transactions on Computers, vol. C–28, No. 12, December 1979.

[17]. James B. Shearer, ―Some New Optimum Golomb Rulers, IEEE Transactions on Information Theory, IT–36:183–184, January 1990.

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

[19]. Stephen W. Soliday, A. Homaifar, Gary L. Lebby, ―Genetic Algorithm Approach to the Search for Golomb Rulers, Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA–95), Morgan Kaufmann (1995), 528–535.

[20]. John P. Robinson, "Genetic Search for Golomb Arrays, IEEE Transactions on Information Theory, Vol. 46, No. 3: 1170–1173, May 2000.

[21]. 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.

[22]. Bansal, S. and Kuldeep Singh, P.2014. A Novel Soft-Computing Algorithm for channel allocation in WDM Systems Vol. 85, No. 9, pp. 19-25. International Journal of Computer Applications(0975-8887)

[23]. A. Dimitromanolakis, ―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, June 2002.

[24]. Apostolos Dollas, William T. Rankin, and David McCracken, ―A New Algorithm for Golomb Ruler Derivation and Proof of the 19 Mark Ruler‖, IEEE Transactions on Information Theory, Vol. 44, No. 1, 379–382, January 1998.

[25]. Project OGR‖, http://www.distributed.net/OGR.

[26]. J. Singer, A theorem in finite projective geometry and some applications to number theory, Trans. Am. Math. Soc. 43 (3) (1938) 377–385.

[27]. X.-S. Yang 2010. Firefly Algorithm, Stochastic Test Functions and Design Optimization. Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp.78–84.

[28]. Pereira, F.B., Tavares, J., Costa, E., ―Golomb Rulers: The Advantage of Evolution, Proceedings of the 11th Portuguese Conference on Artificial Intelligence, Workshop on Artificial Life and Evolutionary Algorithms (ALEA), EPIA'03. (2003), pp. 27–33, 2003.