Golomb Ruler Sequences Optimization for FWM Crosstalk Reduction: Multi-population Hybrid Flower Pollination Algorithm

Prince Jain, **Shonak Bansal**, **Arun Kumar Singh**, **and Neena Gupta**
PEC University of Technology, Sector-12, Chandigarh, India

Abstract— Nature-inspired algorithms are becoming powerful methods for solving many tough optimization problems. This paper proposes a recent approach for solving the channel allocation problem in optical wavelength division multiplexing (WDM) systems based on nature-inspired optimization algorithm namely Flower pollination algorithm (FPA) and its improved form by taking into consideration the concept of optimal Golomb rulers (OGRs). The improvement in FPA proposes in this paper includes the partition of entire population into several sub-populations and its hybridization with differential evolution mutation strategy. The comparative study of simulation results concludes that the proposed hybrid algorithm outperform the existing two classical algorithms, i.e., Extended quadratic congruence (EQC) and Search algorithm (SA), one of the existing nature-inspired algorithm, i.e., Genetic algorithm (GA) and its simple form.

1. INTRODUCTION

The adverse nonlinear effects degrade the performance of optical WDM systems. One of the performance degradation effects is FWM crosstalk, a serious problem for WDM systems and can be reduced by unequal channel spacing [1–3]. In order to reduce FWM crosstalk in WDM systems, numerous unequally spaced channel allocation (USCA) algorithm have been studied in [1, 4–6] that have the drawback of increased bandwidth requirement compared to equally spaced channel allocation. In this paper, we present an USCA scheme based on the OGR sequences [7, 8]. It allows the reduction of FWM crosstalk signals preserving the effectiveness of optical channel bandwidth. *Golomb ruler*s are a sequence of non-negative integer numbers such that no distinct pairs of numbers called *marks* from the sequence have the unique difference. An OGR is the shortest length ruler for a given number of marks [9–12]. Golomb ruler, a class of non-deterministic polynomial (NP)-time hard problems [12], has been solved by nature–inspired optimization algorithm such as GA [12–14], Tabu Search (TS) [13] and its hybridization with GA [13]. This paper presents the application of multi-objective Flower pollination algorithm (MOFPA) and its novel improved form, namely, multi-population hybrid multi-objective Flower pollination algorithm (MH-MOFPA) to generate OGR sequences.

The remainder of this paper is as follows: Section 2 introduces the concept of Flower pollination based algorithms. Section 3 presents the problem formulation. Section 4 provides simulation results comparing with existing algorithms of generating unequal channel spacing. Section 5 presents some concluding remarks.

2. MULTI-OBJECTIVE FLOWER POLLINATION BASED ALGORITHMS

Inspired by flow pollination process of flowering plants, Yang et al. [15] developed Flower pollination algorithm. In FPA, global and local pollination are two main steps. The interaction of local and global pollination can be controlled by a switch probability p The global and local pollination process mathematically can be written by (1) and (2) respectively:

$$x_i^t = x_i^{t-1} + \gamma L(\lambda)\left(x_* - x_i^{t-1}\right) \tag{1}$$

$$x_i^t = x_i^{t-1} + \in \left(x_j^{t-1} - x_k^{t-1}\right) \tag{2}$$

where x_i^t is the solution vector x_i at iteration t, x_* is the current best solution found among all solutions, γ is step size control factor, x_j^{t-1} and x_k^{t-1} are pollens from different flowers of the same plant species with $\in$ is drawn in between [0, 1]. In any multi-objective algorithm, a general approach to design problem having M objectives with nonlinear equality and inequality constraints is to combine the individual objectives into a single objective by using weighted sum method:

$$f = \sum_{i=1}^{M} w_m f_m \quad \text{with} \quad \sum_{i=1}^{M} w_i = 1, \quad w_i > 0, \tag{3}$$

where w_i are randomly generated positive weights. By varying the weights, the Pareto front [15] can be expected correctly.

Although for lower-dimensional problems MOFPA has outstanding property as compared to numerous nature-inspired algorithms, but may become challenging for higher-dimensional problems because of the phenomenon of low accuracy and slow convergence rates. Therefore, this paper forward an improved hybrid MOFPA, namely, MH-MOFPA, based on multiple populations and fitness values based differential mutation strategy. In order to explore the search space of MOFPA, the entire population is divided into several sub-populations (multi-population), where each sub-population takes charge in exploring and exploiting the search space. In MH-MOFPA, the mutation rate probability MR_i^t of each solution x_i at running iteration index t is determined based on the fitness value f_i^t of each solution:

$$MR_i^t = \frac{f_i^t}{\max\left(f^t\right)} \tag{4}$$

where $\max(f^t)$ is maximum fitness value in the population of solutions at iteration t. Instead of keeping fixed differential mutation operator in DE mutation strategy, this research uses the varying mutation operators at running iteration t:

$$F_{1i}^t = \left((LB - UB)\frac{t}{\eta} + UB\right)\beta_1 \quad \text{and} \quad F_{2i}^t = \left((UB - LB)\frac{t}{\eta} + LB\right)\beta_2 \tag{5}$$

where LB, UB are lower and upper bound on the solutions respectively, β_1, $\beta_2 \in [0, 1]$ are random vectors drawn from uniform distribution, and η is large values positive fixed parameter. The values of β_1, β_2 and η are selected in such a way that the values of mutation operators F_{1i}^t and F_{2i}^t are less than unity. In simplest case,

$$\beta_1 = rand_1 * 0.0001 \quad \text{and} \quad \beta_2 = rand_2 * 0.0001 \tag{6}$$

where $rand_1$ and $rand_2$ are two random numbers between [0, 1]. To improve search efficiency and increase population diversity, based on mutation rate probability, positions of the solutions x_i are updated by using mutation Equation (7):

$$x_i^t = x_i^{t-1} + F_{1i}^{t-1}\left(x_{best}^{t-1} - x_i^{t-1}\right) + F_{2i}^{t-1}\left(x_{r1}^{t-1} - x_{r2}^{t-1}\right) \tag{7}$$

where $x_{best}^{t-1} = x_*^{t-1}$ is current global best solution at iteration one less than running iteration t, r_1 and r_2 are uniformly distributed random integer numbers between 1 to the problem size. The numbers r_1 and r_2 are different from running index.

3. GOLOMB RULERS OPTIMIZATION: PROBLEM FORMULATION

If each individual element (IE) sequence in non-negative integer location is Golomb ruler, the sum of all elements of an individual sequence forms the total occupied bandwidth. If CS is channel spacing with n as total number of channels, then the objective is to minimize both the ruler length RL and total occupied bandwidth TBW given by (8) and (9) respectively.

$$RL = \sum_{i=1}^{n-1} (CS)_i \text{ subject to } (CS)_i \neq (CS)_j, \text{ where } i, j = 1, 2, \ldots, n \text{ with } i \neq j. \tag{8}$$

$$TBW = \sum_{i=1}^{n} (IE)_i \text{ subject to } (IE)_i \neq (IE)_j, \text{ where } i, j = 1, 2, \ldots, n \text{ with } i \neq j. \tag{9}$$

Thus using OGRs as channel allocation in WDM systems, the two optimization objectives are: $f_1(x) = RL$ and $f_2(x) = TBW$ which that are composited into a single objective $f(x)$. The proposed pseudo-code for MH-MOFPA to generate OGRs in this paper is shown in Figure 1.

4. SIMULATION RESULTS AND DISCUSSION

The proposed algorithms to generate OGRs have been tested in Matlab-7 language with different parameter values. As the number of iterations increases, the rulers are approaching towards their optimal values. From simulation results, it is observed that to generate OGRs up to 20-marks,

```
/* Parameter initialization */
   Define the number of channels n, initial lower and upper bound on the ruler length, Pareto fronts point N and switch probability p ϵ [0, 1];
   Generate MP populations of NP size integers randomly and each NP population corresponding to Golomb ruler to the specified channels;
      /* MP is multi population size, NP is the size of sub populations in MP and integers in flower is being equal to the number of channels n */
   For i = 1 : MP
      For j = 1 : NP
         Find the local best x^i_{lbest,j} among ith population of NP flowers by using Equations (3), (8) and (9);    /* x_lbest is local best solution */
      End for j
   End for i
   Based on fitness value, among MP x_lbest solutions select the globally best solution x*;
/*End of parameter initialization */
For i = 1 : N                                                     /* N is the points on Pareto fronts (PF) */
   Generate M weights which satisfies Equation (3);
   While (TC < Maxiter)       / * TC is a termination criterion and Maxiter is maximum number of iterations */
      For j = 1 : MP
         For k = 1 : NP                                           /*all NP flowers */
A:          If (rand (0,1) < p),
               Draw a (d dimensional) step vector L via Lévy flights and perform global pollination (Equation (1));
            Else
               Draw from a uniform distribution in [0,1] and perform local pollination (Equation (2));
            End if
            /* Mutation */
               Based on the mutation rate probability MRk perform mutation by using Equations (4) to (7);
            /* End of Mutation */
            Check Golombness of updated solutions;
            If Golombness is satisfied
               Retain that solution and then go to B;
            Else
               Remove that particular generated solution and then go to A;
            End if
B:          Evaluate fitness values of the generated NP solutions of jth population and form a single optimize objective f (x_k);
         End for k
         If new solutions are better, update and rank them in the population, and find the current best Pareto optimal solution x^j_lbest;
      End for j
      Find global best solution x* among the MP x_lbest solutions;
   End while
   Record x* as a non dominated solution;
End for i
Display the optimal Golomb ruler sequences;
```

Figure 1: Pseudo-code for MHCMOFPA to generate OGRs.

MOFPA stabilized within *Maxiter* of 800, while MH-MOFPA generates the same solutions with *Maxiter* of 600. This means that the convergence rate of proposed novel algorithm MH-MOFPA is much higher than MOFPA. By varying MP and $NP = 5$ to 100, $\gamma = 0.0$ to 1.0, and $p = 0.0$ to 1.0, it is found that the best parameters for OGR problem are: MP and $NP = 10$ to 30, $\gamma = 0.9$ to 1.0, $p = 0.6$ to 0.8, $Maxiter$ = number of marks $(n) * 100$, $\eta = 2*$ $Maxiter$ and Pareto front points $N = 100$. The size of entire search space in MOFPA is NP while for MH-MOFPA is $MP * NP$.

4.1. Performance Comparison of Proposed Algorithms with Existing Algorithms

Figures 2(a), 2(b) and 2(c) show the optimization in ruler length, total bandwidth and average CPU time for various channels respectively. Although RL and TBW for proposed algorithms are same, but difference is in the convergence and success rates. It is pertinent to mention here that OGRs found from heuristic based exhaustive search, the times varied from 0.035 seconds for 5-marks to 6 weeks for 13-marks, whereas by non-heuristic exhaustive search, the times varied from around 12.57 minutes for 10-marks to 9.36e+20 years for 20-marks [12]. The OGRs found by the exhaustive search [10] for 14 and 16-marks, took nearly one hour and hundred hours respectively, while 17 to 19-marks OGRs found in [11], the times varied from 1440 to 36200 CPU hours. In [13], it is mentioned that CPU time by TS to generate OGRs varied from 0.1 second for 5-marks to 2516 seconds for 13-marks with *Maxiter* of 30000. The OGRs generated by GA [14], the average CPU time was around 31 hours for 20-marks with *Maxiter* of 5000. But to generate OGRs, the average CPU time for proposed MOFPA is 19 hours for 20-marks ruler, while for MH-MOFPA; it is about 2 hours for 20-marks. Thus it is concluded that the success rate for MH-MOFPA is higher than MOFPA.

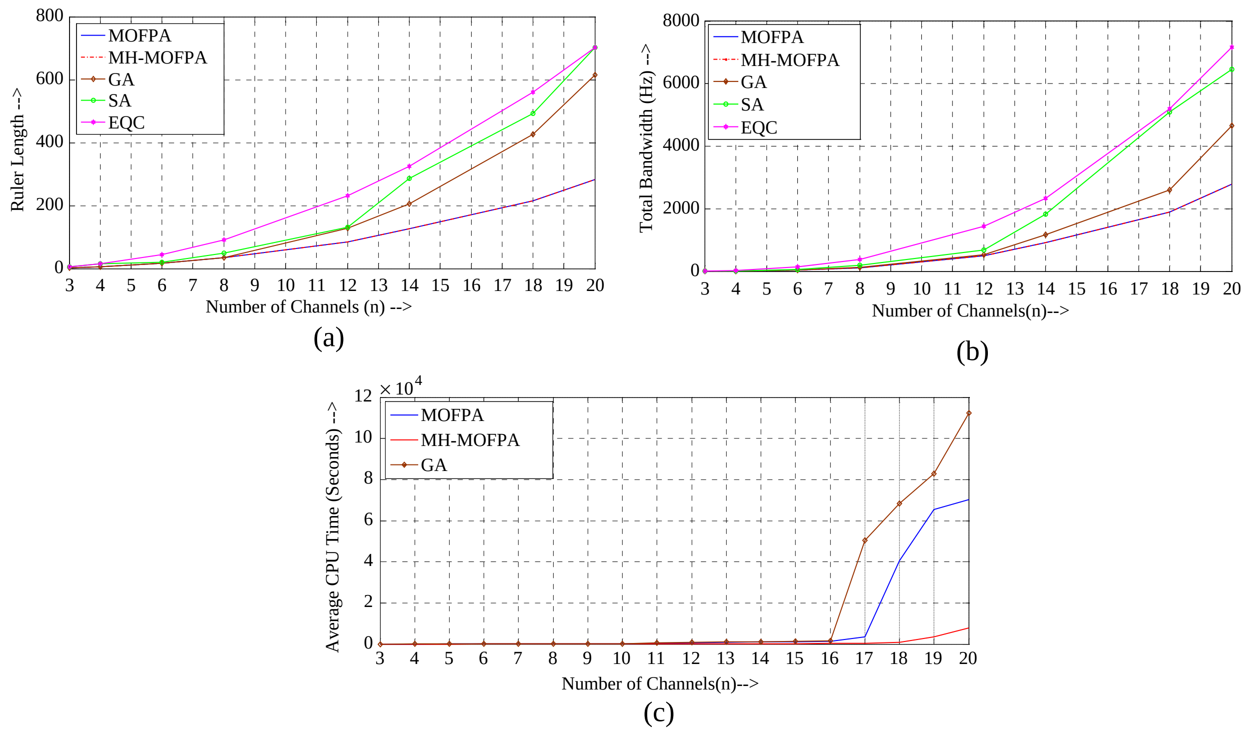

Figure 2: Proposed algorithms demonstrates the significant reduction in (a) ruler length, and (b) total channel bandwidth, (c) average CPU time in comparison to the existing algorithms.

5. CONCLUSION

The channel allocation scheme was presented in this paper, intending to reduce FWM crosstalk in WDM systems. The purpose of nature-inspired algorithms is to produce the best results under the constraints of given cost and time. Even if exact algorithms are able to generate OGRs, they remain unpractical in terms of computational complexity. Indeed, many months or sometimes years on many computers are necessary to prove the optimality of large mark rulers. The main technical contribution of this paper was to successfully formulate a novel hybrid algorithm for OGRs generation, named, MH-MOFPA. The preliminary results indicate that the proposed algorithms appear to be most efficient to such NP-hard problems. From the simulations results it is also concluded that the proposed MH-MOFPA can significantly improve the performance of MOFPA.

REFERENCES

1. Kwong, W. C. and G. C. Yang, "An algebraic approach to the unequal-spaced channel-allocation problem in WDM lightwave systems," *IEEE Trans. on Communication*, Vol. 45, No. 3, 352–359, 1997.
2. Thing, V. L. L., P. Shum, and M. K. Rao, "Bandwidth-efficient WDM channel allocation for four-wave mixing-effect minimization," *IEEE Trans. on Communication*, Vol. 52, No. 12, 2184–2189, 2004.
3. Forghieri, F., R. W. Tkach, A. R. Chraplyvy, and D. Marcuse, "Reduction of four-wave mixing crosstalk in WDM systems using unequally spaced channels," *IEEE Photonics Tech. Letters*, Vol. 6, No. 6, 754–756, 1994.
4. Sardesai, H. P., "A simple channel plan to reduce effects of nonlinearities in dense WDM systems," *Lasers & Electro-Optics*, 183–184, 1999.
5. Hwang, B. and O. K. Tonguz, "A generalized suboptimum unequally spaced channel allocation technique — Part I: In IM/DDWDM systems," *IEEE Trans. Commun.*, Vol. 46, 1027–1037, 1998.
6. Atkinson, M. D., N. Santoro, and J. Urrutia, "Integer sets with distinct sums and differences and carrier frequency assignments for nonlinear repeaters," *IEEE Trans. on Communication*, Vol. 34, No. 6, 614–617, 1986.
7. Bloom, G. S. and S. W. Golomb, "Applications of numbered undirected graphs," *Proceedings of the IEEE*, Vol. 65, No. 4, 562–570, 1977.

8. Shearer, J. B., "Some new disjoint Golomb rulers," *IEEE Trans. on Information Theory*, Vol. 44, No. 7, 3151–3153, 1998.
9. Robinson, J. P., "Optimum Golomb rulers," *IEEE Trans. on Computers*, Vol. 28, No. 12, 183–184, 1979.
10. Shearer, J. B., "Some new optimum Golomb rulers," *IEEE Trans. on Information Theory*, Vol. 36, 183–184, 1990.
11. Dollas, A., W. T. Rankin, and D. McCracken, "A new algorithm for Golomb ruler derivation and proof of the 19 mark ruler," *IEEE Trans. on Information Theory*, Vol. 44, No. 1, 379–382, 1998.
12. Soliday, S. W., A. Homaifar, and G. L. Lebby, "Genetic algorithm approach to the search for Golomb rulers," *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA-95)*, 528–535, Morgan Kaufmann, 1995.
13. Ayari, N., T. Van Luong, and A. Jemai, "A hybrid genetic algorithm for Golomb ruler problem," *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, 1–4, 2010.
14. Bansal, S., "Optimal Golomb ruler sequence generation for FWM crosstalk elimination: Soft computing versus conventional approaches," *Applied Soft Computing*, Vol. 22, 443–457, 2014.
15. Yang, X.-S., M. Karamanoglu, and X. S. He, "Flower pollination algorithm: A novel approach for multiobjective optimization," *Engineering Optimization*, Vol. 46, No. 9, 1222–1237, 2014.