Using a parallel tabu search to approximate uniform design

Alberto Rodríguez Sánchez[1], Antonin Ponsich[1], Antonio López Jaimes[2], and Saúl Zapotecas Martínez[2]

[1] Dpto. de Sistemas, Universidad Autónoma Metropolitana Azcapotzalco
{ars,aspo}@correo.azc.uam.mx
[2] Dpto. de Matemáticas Aplicadas y Sistemas, Universidad Autónoma Metropolitana Cuajimalpa
alopez@correo.cua.uam.mx, saul.zapotecas@gmail.com

Abstract. In Multi-objective Optimization (MO), diversity assessment is one of the most important concern in order to produce an approximated set of solutions evenly distributed over the Pareto front. To deal with this issue, recent algorithms such as MOEA/D[2] make use of a uniformly scattered set of reference points/vectors that indicates search directions in the objective space. This issue becomes critical in Many-objective Optimization, promoting the development of many generation techniques having a rich underlying theory mainly arising from chemistry and statistics areas.

Among these methods, the Uniform Design (UD) is based on the minimization of a discrepancy metric, which measures how well equidistributed the points are in a sample space. Of particular interest in this work is the centered L_2 discrepancy metric proposed in [1]. An exponentially increasing number of candidate sets can be generated using the Good Lattice Point (GLP) technique, which involves a huge computational cost (memory and time). It was demonstrated that the problem of finding a uniform design under a given discrepancy metric is NP-hard when the number of runs, $n \to \infty$ and the number of factors, $s > 1$.

In order to solve this optimization problem, a parallel Tabu Search (TS) is implemented in this work. A specific feature is the tabu list that only reports "generator parameters", which are the input needed by the GLP algorithm to generate the set of final reference points. The best reference sets founded by TS were subsequently used to solve two classical MO problems with MOEA/D (using the Tchebycheff scalarizing function). The results are compared with those of the Simplex Lattice Design (SLD) in terms of the Hypervolume (HV) and Δ-diversity indicators. Results highlight that the UD allows a significant improvement, particularly regarding diversity and when the number of objectives increases.

1 Introduction

2 Overview on weight vector design for reference-based MOEAs

2.1 Mixture design

Experiments with mixtures are experiments in which the variants are proportions of ingredients in a mixture. An example is an experiment for determining the proportion of ingredients in a polymer mixture that will produce plastics products with the highest tensile strength. Similar experiments are very commonly encountered in industries. Designs for deciding how to mix the ingredients are called experimental designs with mixtures.

A design of N runs for mixtures of m ingredients is a set of N points in the domain:

$$T_m = \{(\lambda_1, \dots, \lambda_m) : \lambda_j \geq 0, j = 1, \dots, m, \lambda_1 + \dots + lambda_m = 1\} \quad (1)$$

The original MOEA/D[2] just adopts the simplex-lattice design to set the aggregation coefficient vectors. There are at least two problems can be found with that design:

- The experimental points are not very uniformly distributed on the experimental domain T_m.
- There are too many experimental points at the boundary of the experimental domain, i.e, the unitary simplex.

2.2 Based on mixture designs

a lot of work which appeared in the statistical literature proposed many kinds of designs. Scheffé introduced the simplex-lattice designs and the corresponding polynomial models (cita de Scheffé simplex lattice,). Later he introduced an alternative design, the simplex-centroid design (citar Experiments with mixtures), to the general simplex-lattice. Cornell gave a suggestion of axial design and he gave a comprehensive review of nearly all the statistical articles on designs of experiments with mixtures and data analysis (buscar cita en Experiments with mixtures, designs, models, and the analysis of mixture data).

Simplex Lattice

Two layer simplex Lattice

Simplex-centroid design

Axial design

2.3 Based on discrepancy functions

3 UD : problem statement

3.1 Good lattice point

3.2 Discrepancy functions

3.3 CD_2 function and optimization problem for UD

$$\operatorname{argmin} CD_2 = \left(\frac{13}{12}\right)^m - \left(\frac{2}{N}\right) \sum_{k=1}^{N} \prod_{i=1}^{m} \left(1 + \frac{1}{2}|X_{ki} - \frac{1}{2}| - \frac{1}{2}|X_{ki} - \frac{1}{2}|^2\right) + \left(\frac{1}{N^2}\right) \sum_{k=1}^{N} \sum_{j=1}^{N} \prod_{i=1}^{m} \left(1 + \frac{1}{2}|X_{ki} - \right. \tag{2}$$

where:

- X is the induced Matrix of a U-type design.
- N Rows of X.
- m Columns of X.

4 Solution technique : parallel Tabu Search

Algoritmo 4.1: Parallel Tabu Search

```
Data: IterMax: Stop Condition, W: workers, T_max: Tabu List size, N:
      Number of points wanted, dim: number of dimensions
Result: NUD:Nearly Uniform Design
bag ← SearchCoprime(N);
Solutions ← RandomDisjoint(bag,dim,W);
bestdis ← ∞, bestsol ← ∅, TabuList ← ∅ ;
while ¬Stopping(IterMax) do
    /* Parallel For                                              */
    for i := 1 a W do
        U_i ← GLP(Solutions_i) ;
        X_i ← InducedMatrix(U_i) ;
        discrepancy_i ← CD_2(X_i) ;
        if discrepancy_i < bestdis_i and Solutions_i ∉ TabuList then
            bestsol_i ← Solutions_i;
            bestdis_i ← discrepancy_i;
            TabuList ← TabuList ∪ Solutions_i;
        /* Critical Zone                                         */
        Solutions_i ← UpdateSol(Solutions_i,bag);
        if |TabuList| > T_max then
            TabuList ← MaintainTabuList(TabuList) ;
    NUD ← min(Solutions, bestdis) ;
return (NUD)
```

4.1 Decision variables and codification

4.2 Neighborhood and Tabu list

4.3 Specific features

5 Computational experiments and discussion

5.1 Uniform designs obtained

5.2 Testing on classical MO instances

6 Conclusions

References

1. Fang, K.T., Ma, C.X., Winker, P.: Centered l2-discrepancy of random sampling and latin hypercube design, and construction of uniform designs. Mathematics of Computation **71**(237), 275–296 (2002)
2. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (Dec 2007). https://doi.org/10.1109/TEVC.2007.892759