Multi-Objective Design Optimization of Grillage Systems by Scatter Search Methodology

Tugrul Talaslioglu
Osmaniye Korkut Ata University, Department of Civil Engineering, Osmaniye, TURKEY
ttalaslioglu@oku.edu.tr
doi:10.6088/ijcser.00202010039

ABSTRACT

In this study the design of grillage systems is minimized considering both its entire weight and joint displacements. Scatter search methodology as an optimization tool is utilized for design minimization. Scatter search methodology executes its optimization procedures using predefined values of genetic operators. However, this approach causes to an increase in computational cost of optimization procedures. This bottleneck is overcome by dynamically adjusting the parameter values of genetic operators according to values of two performance metrics, Spread and Inverted Generational Distance (IGD). Furthermore, the flexibility of proposed optimization tool based on scatter search methodology is increased by automatically generating longitudinal-transversal beams of grillage system and correspondingly distributing the loads to the related joints. It is demonstrated that proposed optimization tool has a capability of producing better optimal designations compared to those obtained by a pure usage of scatter search methodology.

Keywords: Multi-objective, optimization, grillage system, Scatter Search, AISC_LRFD

1. Introduction

A grillage system contains the traversal and longitudinal which of cross-section properties are assigned from a practically available set of standard steel sections. Grillage systems are utilized in different structures, for example bridge or ship decks, building floors and space structures etc. The cross-sections of beams are determined according to certain design criterion prescribed by any code of practice, for example LRFD. After specified the design criterion, the entire weight of grillage system is minimized (Saka et.al, 2000, Erdal and Saka, 2008, Saka and Erdal, 2009, Kahev and Talatahari, 2010). The values of joint displacements used in displacement constraints are limited by max span/400 according to LRFD-AISC V3 (LRFD, 2001). This safety margin on displacement constraints can be large especially for ship decks, floors of industrial buildings which bears special machines. If the positions of these machines are not balanced horizontally, they don’t regularly work. This task is overcome thereby utilizing multi-objective optimization concept (Kelesoglu and Ulker, 2005, Coello and Christiansen, 2000). Therefore, two conflicting and commensurable objective functions, weight of grillage system and joint displacement are included into optimal design stage. In spite of a wide variety of multi-objective optimization algorithms (MOAs), for example NSGA II, SPEA II, PAES etc., Archeive Based Hybrid Scatter Search which of fundamentals are based on these algorithms mentioned achieves to take more attention in last few years (JMETAL, 2008). Therefore, AbYSS is proposed to optimize the design of grillage systems. For this purpose, a ready optimization tool called as JMETAL (2008) and coded in Java Language is executed the optimization procedures of AbYSS.
2. Multi-objective Optimization Mechanism and Its Extended Algorithm: Archive Based Hybrid Scatter Search (AbYSS)

The fundamentals of MOAs are based on an exploration of a solution set with multiple objective functions considering the penalty values of problem-related constraints. This solution set, also called as pareto solution that contains a number of solutions with different dominance degree. In this regard, a non-dominated solutions set is comprised of a number of solutions set. These non-dominated solutions form a curve, called as pareto front. The solution set located on just this pareto front is named as true pareto front and not improved further. The pareto front is also responsible to asses the performance of any proposed MOA according to its current pareto front that is obtained in the end of evolutionary search.

In the performance evaluation of MOAs, one of the major approaches is to examine the quality of non-dominated solution set considering its approximation degree to the true pareto front. The approximation quality becomes good when the evolutionary search exhibits a fast and accurate convergence towards true pareto front including all non-dominated solutions. However, it is a difficult task to achieve both a fast and accurate convergence at the same time due to inconsistent variation in the distribution quality of population (Laumanns et.al, 2002). Although a categorization of MOAs is out of scope in this paper (see a detailed grouping of MOAs in Coello (2005)), it was shown that the archive based approaches of MOAs, for example AbYSS, achieved to handle these two issues, fast and accurate convergence (see Nebro et.al, 2008).

2.1. Structure of AbYSS

Scatter Search firstly introduced by Glover (1997) has been successfully applied to both combinatorial and linear or non-linear optimization problems (Deb et.al, 2002). AbYSS emerged by a hybridization of scatter search algorithm with basic methods of well-known MOAs, NSGA II, SPEA II and Evolutionary Algorithm (EA). These methods are called in the context of AbYSS as: diversication generation borrowed from scatter search, improvement from EA and NSGA II, reference set update from SPEA II, subset generation and solution combination (see a java code in Figure 1).

The execution of AbYSS’ optimization procedure starts by initializing a population called as “solution set”. Thus, “diversication generation method” is activated. Each individual of population is created by use sub-solution-regions obtained by division of entire solution region into smaller parts. The number of sub-solution-regions is specified by a parameter named as “number of sub-ranges. Following the generation of first solution set, solutions contained in “solutions set” are mutated according to a dominance test used in (1+1) Evolutionary Algorithm (EA)

The main cycle of evolutionary search starts by invoking “reference set update method” to update “reference set”. In this regard, the activation option of reference set update method must be set to “true”. Thus, solutions contained in “solutions set” are combined to generate a population called as “reference set”. Then, the solutions generated are employed to update “reference set” in a way of setting the activation option of reference set method to “false”. Two subsets, reference set 1 with “RefSet1” individuals and reference set 2 with “RefSet2”
individuals are utilized. Reference set update method is terminated when a condition “newsolution > 0” is satisfied. Whereas “reference set1” is created by use of solutions generated according to the selection rules of SPEA II algorithm, the distance measures of these solutions are used to create “reference set 2”. Then, these new solutions are combined by activating “subset generation method”. If remaining evolution number does not exceed the maximum evolution number “max evaluations”, then solution set is completely re-generated.

2.2. Introduction of Dynamic AbYSS Algorithm (DAbYSS)

In order to obtain a computationally faster and more accurate convergence to pareto front by preserving the diversity, one of the common approaches is to integrate a local search strategy into evolutionary search (Knowles, 2002). Whereas AbYSS explores globally the entire search space for promising solution sets, a local search strategy exploits available solution set to provide a precise search for previously visited solution region. The mutation operation managed by the method of “Mutation Local Search” is responsible to perform local search for AbYSS. Combining operation based on a compare-eliminate-update process which is managed by methods of “Reference Set Update” and “Subset Generation” supports the mutation operator. The crossover operator employed to perform the combining operation creates a new solution set “new solution”.

In order to enhance the performance of evolutionary search, AbYSS make use of an increased population size and evaluation number causing to an increase in the computational cost of its optimization procedure. Therefore, a feed-back based adaptation with an external control is offered to direct the optimization procedures of AbYSS. This adaptation is based on adjusting the parameter values of AbYSS’ genetic operators according to two performance metrics; Spread and Inverted Generational Distance (IGD) (see Eqns. 1-2). In this regard, while Spread is employed to measure efficiency of mutation and crossover operators, IGD is utilized to assess the variation degree throughout the evolutionary search. Furthermore, reintroduction of non-dominated solutions gathered in the end of each run into the evolutionary search is also included into of the evolutionary search (see the activation of introducing operation by NewStrategy=1 in Figure 1).

2.2.1. Quality-Measuring Metrics

2.2.1.1. Inverted Generational Distance (Igd)

Igd estimates the far of non-dominated solutions included in current pareto front generated by the proposed MOA, from those included in true pareto front.

\[
Igd = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{n} d_i^2}}{n}
\]

(1)

where \( n \) is number of non-dominated solutions found by proposed MOA and \( d_i \) is Euclidian distance between each of these and nearest member of true pareto front. A lower value of Igd indicates an increase in approximation of current pareto front obtained to the true pareto front in terms of convergence.
2.2.1.2. Spread

This quality metric is used to measure an expanding spread exhibited by non-dominated solutions obtained and computed as,

\[
\text{Spread} = \frac{d_f + d_1 + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_1 + (N-1) \times \bar{d}}
\]  

(2)

where \(d_i\) is Euclidian distance between consecutive non-dominated solutions, \(\bar{d}\) is mean of these distances, \(d_f\) and \(d_l\) are distances to extreme solutions of current pareto front. A lower Spread value points out a better and arranged distribution among non-dominated solutions. In this work, a higher value of Spread is aimed in order to provide a divergent distribution among non-dominated solutions. Thus, differentiation among solution set is kept at a maximum level.

2.2.2. Parameters Specified for DAbYSS

Two main parameters “NumNewSolutions” (abbreviated for Number of New Solutions) and “NumSolutionSetSize” (abbreviated for Number of Solutions Set Size) governs the optimization procedures of AbYSS (see the parameters NumNewSolutions and NumSolutionSetSize in Figure 1). In fact, NumNewSolutions controls a closed loop that is responsible to update “reference set” using “ReferenceSetUpdate method” and generate new solutions using “SubSetGeneration method”. If the value of NumNewSolutions increases, the exploitation capacity of AbYSS is elevated due to an increased activating number of SubSetGeneration method. The exploration capacity is elevated by use of higher number of randomly generated solution. If the value of NumSolutionSetSize is decreased; the variable “insert” is increased (see the parameter “insert” in Figure 1). Thus, the number of randomly generated solution is elevated. A parameter “IndependentRuns” that externally controls the evolutionary search is utilized to perform DAbYSS’ computational procedures.

In order to increase the performance DAbYSS, appropriate parameter values must be assigned to its genetic operators (see these parameters indicated by bolt or underlying characters in Figure 1). Therefore, DAbYSS is performed for design examples proposed here using different parameter values. Considering optimal designations obtained by DAbYSS, two basic combination sets, named as exploring and exploiting based approaches are offered (see Tables 1 and 2). While the exploring based approach is based on a mutation-based management, the exploiting based approach make uses of combining operation to control the evolutionary search. In addition, each approach is managed by four sub-combination sets. The activation of these sub-combination sets are controlled by the values of Dynamic Spread Limit (DSL) and Dynamic Inverted Generational Distance Limit (DIGDL). DSL and DIGDL values are dynamically assigned in an increased order. According to DSL and DIGDL values, evolutionary search is either intensified by use of higher values of population size, evolutionary number and related parameter or quickly completed by use of lower values of these parameters.
In order to examine the efficiency of DAbYSS’ exploiting and exploring capacities, total three cases: Case I (IndependentRuns=5, PopSize=300, EvaluationNumber=2000), Case II (IndependentRuns=10, PopSize=200, EvaluationNumber=900) and Case III (IndependentRuns=20, PopSize=100, EvaluationNumber=600) are used to represent the varying population size and evaluation number. These three cases are performed for both exploring and exploiting approaches separately. Furthermore DSL and DIGDL are assigned in a three equal interval by depending on IndependentRuns. A general arrangement of these parameters and their related combination numbers are presented by a pseudo code in Figure 2.

```java
NewStrategy=1
Initialize
Built solutionSet

if NewStrategy==0 || evaluations==1{
    for int i=0;i<SolutionSetSize;i++{
        solution=DiversificationGeneration()
        input for diversificationGeneration: Number of Subranges, Lower Limit of Variables, Upper Limit of Variables,
        SolutionsSetSize=PopulationSize
        Insert solution into SolutionSet
        evaluations++
    }
    else {
        solution=PreviousSolutions
    }
    Execute MutationLocalSearch
    evaluations+= evaluations + number of evaluation calculated in MutationLocalSearch
}
while evaluations < Max Evaluations {
    Update ReferenceSetupdate(true)
    input for referenceSetupdate: RefSet1Size, RefSet2Size, SubSet=(SolutionSetSize*1000
    newsolution=SubSetGeneration()
    evaluations= evaluations + number of evaluation calculated in subsetGeneration
    if newsolution > 0 {
        while newsolution > NumNewSolutions {
            Update ReferenceSetupdate(false)
            if evaluations >= Max Evaluations {
                return archieve
            }
            newsolution=SubSetGeneration()
            evaluations= evaluations + number of evaluation calculated in subsetGeneration
        }
        if evaluations >= Max Evaluations {
            Clear solutionSet
            for int i=0;i<RefSet1.size;i++{
                Execute MutationLocalSearch
                evaluations+= evaluations + number of evaluation calculated in MutationLocalSearch
                Insert solution into SolutionSet
            }
            Clear RefSet1
            Clear RefSet2
            Calculate crowding distances of individuals stored in archvive
            //Insert= solutionSetSize/2
            Insert= solutionSetSize/NumSolutionSetSize
            if insert > Archive.size{
                Insert= Archive.size
            }
            if insert > (SolutionSetSize - SolutionSet.size){
                Insert= SolutionSetSize - SolutionSet.size
            }
            Insert Archive into SolutionSet
            if SolutionSetSize < SolutionSet.size {
                Create the rest of solutions randomly and insert solutions into solutionSet
            }
        }
    }
}
```

Figure 1: A Java Code Summarized the Computation Procedures of DAbYSS (see the Java Code of AbYSS in Reference 8)
CaseNo=1

    if CaseNo==1 { //Case I
        IndependentRuns=5, MaxEvaluation=2000, PopulationSize=300,
        //if CaseNo==2 { //Case II
        //IndependentRuns=10, MaxEvaluation=900, PopulationSize=200,
        //if CaseNo==3 { //Case III
        //IndependentRuns=20, MaxEvaluation=600, PopulationSize=100,
            for int k=0;k<IndependentRuns;k++{
                if ((IndependentRuns >= 0) && (IndependentRuns <=1)){
                    //Case I
                    DSL=0.20;
                    DIGDL=2.00;
                } else if ((IndependentRuns >1) && (IndependentRuns <=2)){
                    //Case I
                    DSL =0.30;
                    DIGDL =3.00;
                } else if ((IndependentRuns >2) && (IndependentRuns <= 4)){
                    //Case I
                    DSL =0.40;
                    DIGDL =4.00;
                } else if ((IndependentRuns >4) && (IndependentRuns <=6)){
                    //Case II
                    DSL=0.50;
                    DIGDL=3.50;
                } else if ((IndependentRuns >6) && (IndependentRuns <=8)){
                    //Case II
                    DSL =0.60;
                    DIGDL =4.50;
                } else if ((IndependentRuns >8) && (IndependentRuns <=13)){
                    //Case III
                    DSL=0.70;
                    DIGDL=5.00;
                } else if ((IndependentRuns >13) && (IndependentRuns <=19)){
                    //Case III
                    DSL =0.80;
                    DIGDL =5.50;
                }
                SpreadRatio=(Spread_new-Spread_old)/Spread_old) //coming from DAbYSS
                IGDRatio=(IGD_new-IGD_old)/IGD_old //coming from DAbYSS
                if (SpreadRatio  DSL) \& (IGDRatio  DIGDL) {
                    //Choose one of the exploiting or exploring based approaches for execution of DAbYSS
                    //Assignment of the parameter values represented by Combination No 0 for Execution of DAbYSS
                } else if (SpreadRatio  DSL) \& (IGDRatio  DIGDL) {
                    //Choose one of the exploiting or exploring based approaches for execution of DAbYSS
                    //Assignment of the parameter values represented by Combination No 1 for Execution of DAbYSS
                } else if (SpreadRatio  DSL) \& (IGDRatio  DIGDL) {
                    //Choose one of the exploiting or exploring based approaches for execution of DAbYSS
                    //Assignment of the parameter values represented by Combination No 2 for Execution of DAbYSS
                } else if (SpreadRatio  DSL) \& (IGDRatio  DIGDL) {
                    //Choose one of the exploiting or exploring based approaches for execution of DAbYSS
                    //Assignment of the parameter values represented by Combination No 3 for Execution of DAbYSS
                }
            }
        }
    }

Figure 2: A Pseudo Code for Assignment of Related Combination No According to IGD and Spread Values Associated with Case I, II and III
Table 1: Parameter Sets of Related Sub-combination Numbers for Exploiting based Approach

<table>
<thead>
<tr>
<th></th>
<th>Combination No 0</th>
<th>Combination No 1</th>
<th>Combination No 2</th>
<th>Combination No 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>PS*0.25</td>
<td>PS*0.70</td>
<td>PS*0.30</td>
<td>PS*1.00</td>
</tr>
<tr>
<td>Evolution Number</td>
<td>EN*0.20</td>
<td>EN*0.70</td>
<td>EN*0.30</td>
<td>EN*1.00</td>
</tr>
<tr>
<td>Archive Size</td>
<td>PS*0.15</td>
<td>PS*0.25</td>
<td>PS*0.15</td>
<td>PS*0.25</td>
</tr>
<tr>
<td>Reference Set 1 Size</td>
<td>PS*0.10</td>
<td>PS*0.20</td>
<td>PS*0.05</td>
<td>PS*0.20</td>
</tr>
<tr>
<td>Reference Set 2 Size</td>
<td>PS*0.05</td>
<td>PS*0.10</td>
<td>PS*0.10</td>
<td>PS*0.10</td>
</tr>
<tr>
<td>Crossover Prob.</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Improvement Rounds</td>
<td>EN*0.0025</td>
<td>EN*0.0050</td>
<td>EN*0.0025</td>
<td>EN*0.0050</td>
</tr>
<tr>
<td>Mutation Prob.</td>
<td>0.50</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
</tr>
<tr>
<td>NumNewSolution</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>NumSolSetSize</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

PS: Population Size, EN: Evolutionary Number

Table 2: Parameter Sets of Related Sub-combination Numbers for Exploring based Approach

<table>
<thead>
<tr>
<th></th>
<th>Combination No 0</th>
<th>Combination No 1</th>
<th>Combination No 2</th>
<th>Combination No 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>PS*0.20</td>
<td>PS*0.30</td>
<td>PS*0.70</td>
<td>PS*1.00</td>
</tr>
<tr>
<td>Evolution Number</td>
<td>EN*0.20</td>
<td>EN*0.30</td>
<td>EN*0.70</td>
<td>EN*1.00</td>
</tr>
<tr>
<td>Archive Size</td>
<td>PS*0.15</td>
<td>PS*0.15</td>
<td>PS*0.25</td>
<td>PS*0.25</td>
</tr>
<tr>
<td>Reference Set 1 Size</td>
<td>PS*0.05</td>
<td>PS*0.10</td>
<td>PS*0.10</td>
<td>PS*0.10</td>
</tr>
<tr>
<td>Reference Set 2 Size</td>
<td>PS*0.10</td>
<td>PS*0.05</td>
<td>PS*0.20</td>
<td>PS*0.20</td>
</tr>
<tr>
<td>Crossover Prob.</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Improvement Rounds</td>
<td>EN*0.0020</td>
<td>EN*0.0020</td>
<td>EN*0.0020</td>
<td>EN*0.0020</td>
</tr>
<tr>
<td>Mutation Prob.</td>
<td>0.50</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
</tr>
<tr>
<td>NumNewSolution</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>NumSolSetSize</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
3. Optimum Design of Grillage System

In this work, a design problem of grillage system is represented by two objectives, weight of entire grillage system and displacement of nodal joint and expressed as:

\[
\min W = \sum_{k=1}^{m} (w \cdot l) \quad \text{and} \quad \min d_{ij} \quad (k = 1, \ldots, m, \ i = 1, \ldots, 12 \text{ and } j = 1, \ldots, n) \quad (3)
\]

Subject to

\[
\frac{d_{ij}}{d_{\max}} \leq 1 \quad (i = 1, \ldots, 12 \text{ and } j = 1, \ldots, n) \quad (4)
\]

\[
\frac{M_{uk}}{(\phi_b \cdot M_{nk})} \leq 1 \quad (k = 1, \ldots, m) \quad (5)
\]

\[
\frac{V_{uk}}{(\phi_s \cdot V_{nk})} \leq 1 \quad (k = 1, \ldots, m) \quad (6)
\]

Where term \( W \) is total weight of all grid members and computed using \( w \) and \( l \) which are unit weight to be selected from \( W \)-sections list and length of a grid member. While \( d \) is termed as a joint displacement corresponding to related degree of freedom which is denoted \( i \) and \( n \), \( m \) indicates total numbers of joint and grid member. \( d_{\max} \) is taken as max span/400. In constraint inequalities, while displacements of joints are constrained by an upper limit \( d_{\max} \), bending moment strength of grid members \( M_{nk} \) is limited by allowable nominal moment strength \( M_{nk} \). Shear strength of grid members \( V_{uk} \) is limited by allowable nominal shear strength \( V_{nk} \) (see \( V_n \) in Eqn. (6)). In equations 5 and 6, \( \phi_b \) and \( \phi_s \) are resistance factors for moment and shear and taken as 0.9. In the design of grillage systems bending moment and shear force of any grid member are obtained from results of structural analysis and employed to design of that member according to AISC-LRFD V3 (LRFD, 2001). In this work, a design approach for the grillage systems is proposed. According to a pre-specified magnitude of uniform loading, the grillage system is automatically generated. Some application examples solved by use of AISC-LRFD V3 provisions are presented in Segui’ book (2007). In addition, structural analysis of the grillage system is formulated by Saka and Erdal (2009).


In this study, design of grillage systems is optimized only considering the magnitude of load in lb/ft\(^2\). The nodal coordinates, member connectivity are automatically generated by an intelligent grillage model generation module coded in java language and invoked by JMETAL (2008). Thus, both member cross sections and geometry of grid system simultaneously contributes to optimal design of grillage system. After defining the division numbers, the beams and related nodal coordinates are automatically generated. In this design facility, load in lb/ft\(^2\) is distributed to related latticed beams considering their spans in z
direction. Thus, loads in lb/ft are carried only uniformly by latticed beams along x directions. In Figure 3, the projection of distributed loads on latticed beams is indicated by dotted lines.

![Figure 3: Projections of Distributed Loads on Latticed Beams along x Direction](image-url)

4. Search Methodology

Due to fact that application problems are real-world engineering design problems with design variable of discrete type, a reliable and consistent search strategy must be established for a performance evaluation of MOAs. Associating with the traditional approaches, the performances of MOAs are assessed considering closeness of these designations to pareto front known beforehand. Furthermore, accuracy in the assessment of MOAs must be confirmed by outcomes of statistical tests. Therefore, a reasonable approach is to obtain a pareto front for design the example by running current optimization model in bigger and repeated generation numbers. In this regard, independent 10 runs with an increased size of generation and population are executed by use of AbYSS algorithm. Due to nature of stochastic algorithms, a statistical test for analysis of these quality-measuring metrics computed must be performed with a certain level of confidence. Details about quality-measuring metrics and statistical tests employed for MOAs’ performance assessment are presented in following sub-section.

4.1. Statistical Tests

After computing means and standard deviations of quality indicators obtained by independent 10 executions, consistency of these results are checked through statistical analysis in a certain level of confidence. If a probability value resulted from a statistical testing procedure satisfies a user defined significance level, then it is said that distribution of current MOAs approximation set is acceptable.

Computational procedures of statistical analysis are performed in MATLAB (2008). Firstly, Kolmogorov-Smirnov test is carried out to check quality indicator values for whether to be exhibited a normal distribution at 5% significance level. Then, existence of a variance
homogeneity is controlled through Levene’s test. If there exists homogeneity in the variance, Anova test is performed, otherwise Welch test (Walls, 2003).

5. Discussion of results

It is mentioned that DAbYSS is managed by dynamically adjusted parameter values. These parameters and their related values are reported in Section 2.2. In this regard, three Cases, each of which utilizes three different parameter combination sets for their evolutionary search, are separately performed for exploiting and exploring based approaches. Therefore, there are possible six combination sets. Total trial number for each of these six cases is 10. One of 10 trials is best. Mean and standard deviation values of quality metrics obtained from 10 trials. Furthermore, the solution set obtained from each independent runs is re-introduced into evolutionary search as a beginning population. The effect of this introducing operation on the behavior of DAbYSS is investigated according to quality degree of optimal designations.

The characteristics of steel material are represented by a 50 mild steel with a yield stress of 50 ksi, a elastic modulus of 29,000 ksi and a shear modulus of 14,500 ksi. The sectional properties of grid members are stored in a discrete set with 273 w-sections. Design variables are represented by binary strings. Thus, a binary length of l=9 is adequate to represent 273 ready sections. Therefore, there are \(2^9=512\) possible gene combinations. The division numbers of each span in x and z directions are coded in 4-digit binary strings. Thus, total length of the string is l=9+9+4+4=26. Two different cross sections one in x direction and one in z direction are used to represent the design variables. Design examples of real-world engineering structures are presented in an order of increasing size of their loadings.

5.1. Design Example 1

The design of grillage system subject to a load of 1000 lb/ft\(^2\) is optimized (see its covered area of 30x45 ft\(^2\) in Figure 4). Maximum deflection of nodal joints are limited by 45/400 = 0.113 ft (1.356 inch) according to design criteria indicated by max span/400. DAbYSS performs its optimization procedures for optimal design considering both exploiting&exploring based approaches and introducing operation. The variation in combination numbers through Spread and Igd is depicted in Figures 5 and 6. The variation in pareto fronts obtained by use of both exploiting&exploring based approaches and AbYSS algorithm is presented in Figure 7. While the activation number of Combination 2 is higher for both approaches, the evolutionary search is dominantly managed by the parameter values of Case II and III (see Figure 5a and 5b). By usage of the introducing operation, the parameter values of Case II and Combination 2 become dominant for the management of both approaches (see pareto fronts in Figure 7b and 7c).

Considering means and standard deviations of Spread and Igd, statistical tests are performed. The statistical test (Welch’ F statistic test) is performed to evaluate means of their Spread and Igd values. It is resulted by p values: 0.8523 for Spread and 0.5729 for Igd respectively. Thus, it is said that there is not any considerably difference among exploiting and exploring based approaches due to satisfying required condition as \(p \geq 0.050\).

According to some of the optimal designations reported in Table 3, when the division number for each span is decreased, carrying capacity of grillage system is reduced due to a decrease in the number of lattice beams. But, using the bigger cross-section of steel profiles leads to an
increase in the carrying capacity of grillage system and its weight, but, a decrease in its joint displacements.

Table 3: Values of Design Variables and Corresponding Weight of Grillage System and Displacement of Joints (Design Example 1)

<table>
<thead>
<tr>
<th>Variable 1</th>
<th>Variable2</th>
<th>Division Number in X Direction</th>
<th>Division Number in Z Direction</th>
<th>Weigh (lb)</th>
<th>Displacement (inch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>229</td>
<td>14</td>
<td>15</td>
<td>2107799.99</td>
<td>0.04799</td>
</tr>
<tr>
<td>4</td>
<td>231</td>
<td>2</td>
<td>9</td>
<td>2160720.00</td>
<td>0.04731</td>
</tr>
<tr>
<td>4</td>
<td>260</td>
<td>15</td>
<td>9</td>
<td>2290320.00</td>
<td>0.04821</td>
</tr>
<tr>
<td>28</td>
<td>241</td>
<td>12</td>
<td>9</td>
<td>2663100.00</td>
<td>0.04814</td>
</tr>
<tr>
<td>4</td>
<td>186</td>
<td>5</td>
<td>14</td>
<td>3672000.00</td>
<td>0.03012</td>
</tr>
<tr>
<td>27</td>
<td>220</td>
<td>8</td>
<td>7</td>
<td>2619900.00</td>
<td>0.04838</td>
</tr>
<tr>
<td>27</td>
<td>132</td>
<td>8</td>
<td>7</td>
<td>254699.99</td>
<td>0.04830</td>
</tr>
<tr>
<td>27</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>5057100.00</td>
<td>0.00326</td>
</tr>
</tbody>
</table>

Figure 4: Geometry of Grillage System Used in Design Example 1
Figure 5: Variation in Combination Number associated with Spread & Igd for both Exploiting and Exploring Approaches (Omitting Introducing Operation) (Design Example 1)
Figure 6: Variation in Combination Number associated with Spread & Igd for both Exploiting and Exploring Approaches (Including Introducing Operation) (Design Example 1)
Figure 7: Pareto Fronts Obtained by Random Point Sets (a), Omitting Introducing Operation (b) and Including Introducing Operation (c) (Design Example 1)
5.2. Design Example 2

The design of grillage system tackled in previous design example is optimized by altering the load by 1500 lb/ft\(^2\) but keeping the area of 30x45 ft\(^2\). Thus, the variation in parameter combination sets is investigated for severe loadings conditions. The variation in combination number through Spread and Igd is presented for exploiting and exploring based approaches in Figures 8 and 9. The pareto fronts obtained by use of both these approaches and AbYSS algorithm is presented depending on states of introducing operation (see Figure 10a-10c).

It is displayed that solution set is maintained by use parameter sets of Combination 3 for both approaches when omitting introducing operation (see Figure 8a and 8b). However, by including introducing operation, the parameter values of Combination No 2 become more dominant in the management of evolutionary search (see Figure 9a and 9b). Considering pareto fronts in Figure 10b and 10c, it is observed that the evolutionary search achieves a close converge to the true pareto front using the parameter values of Combination No2 and Case II for exploring based approach.

Welch’ F statistic test is performed to evaluate means of their Spread and Igd values and resulted by p values: 0.8536 for Spread and 0.3446 for Igd. Thus, it is said that that there is a negligible difference between exploiting and exploring based approaches due to \(p \geq 0.050\).

According to some of the optimal designations reported in Table 4, an increase in the division number leads to elevate the carrying capacity of grid system. An increase in the number of lattice beams cause to use smaller cross-section of steel profiles for design of grillage system. Although the use of smaller cross sections leads to a decrease in the weight of grid system, corresponding joint displacements are increased.

Table 4: Values of Design Variables and Corresponding Weight of Grillage System and Displacement of Joints (Design Example 2)

<table>
<thead>
<tr>
<th>Variable 1</th>
<th>Variable2</th>
<th>Division Number in X Direction</th>
<th>Division Number in Z Direction</th>
<th>Weight (lb)</th>
<th>Displacement (inch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>227</td>
<td>4</td>
<td>11</td>
<td>2897639.99</td>
<td>0.05903</td>
</tr>
<tr>
<td>4</td>
<td>163</td>
<td>5</td>
<td>12</td>
<td>2921040</td>
<td>0.05235</td>
</tr>
<tr>
<td>27</td>
<td>111</td>
<td>7</td>
<td>9</td>
<td>3242880</td>
<td>0.05562</td>
</tr>
<tr>
<td>28</td>
<td>125</td>
<td>6</td>
<td>14</td>
<td>3834540</td>
<td>0.04697</td>
</tr>
<tr>
<td>28</td>
<td>118</td>
<td>2</td>
<td>15</td>
<td>4024440</td>
<td>0.04387</td>
</tr>
<tr>
<td>27</td>
<td>201</td>
<td>7</td>
<td>14</td>
<td>4449599</td>
<td>0.03528</td>
</tr>
<tr>
<td>27</td>
<td>130</td>
<td>9</td>
<td>15</td>
<td>4867200</td>
<td>0.03364</td>
</tr>
<tr>
<td>27</td>
<td>69</td>
<td>5</td>
<td>13</td>
<td>4650839</td>
<td>0.03864</td>
</tr>
<tr>
<td>28</td>
<td>195</td>
<td>13</td>
<td>11</td>
<td>3277799</td>
<td>0.05917</td>
</tr>
<tr>
<td>27</td>
<td>184</td>
<td>13</td>
<td>10</td>
<td>4498559</td>
<td>0.05168</td>
</tr>
<tr>
<td>5</td>
<td>28</td>
<td>15</td>
<td>12</td>
<td>7987320</td>
<td>0.05396</td>
</tr>
</tbody>
</table>
Figure 8: Variation in Combination Number associated with Spread & Igd for both Exploiting and Exploring Approaches (Omitting Introduction of Solution Set into Evolutionary Search)
Figure 9: Variation in Combination Number associated with Spread & Igd for both Exploiting and Exploring Approaches (Including Introduction of Solution Set into Evolutionary
Figure 10: Pareto Fronts Obtained by Random Point Sets (a), Omitting Introduction of Solution Set (b) and Including Introduction of Solution Set (c) (Design Example 2)
6. Conclusion

In this study, the design of grillage system is optimized by use of Archieve Based hYbrid Scatter Search (AbYSS) optimization algorithm. Two objective functions, the weight of grid system and displacement of its joints are minimized by AbYSS’ optimization procedures. Design constraints are implemented from LRFD_AISC V3. A ready evolutionary tool named JMETAL is employed to perform computational procedures of AbYSS. In order to decrease the computational cost of AbYSS’ optimization procedures, evolutionary number and population size are dynamically decreased considering two quality metrics, named Spread and Igd. In this regard, two approaches named exploiting and exploring based approaches are offered to arrange the parameter sets. Each of these approaches is managed by three combination sets of (IndependentRun-EvaluationNumber-PopulationSize) including four sub-combination sets of genetic parameters values. Thus, these parameter sets are assigned for the optimization procedures of DAbYSS according to the values of Spread and Igd. Furthermore, solution sets obtained from each run of evolutionary search are introduced to the evolutionary search as a beginning population. Thus, the effect of introducing operation on solution quality is also investigated.

It is demonstrated that an use of parameter values of Combination No 2 and Case II (IndependentRuns=10, PopSize=200, EvaluationNumber=900) for both approaches including introducing operation leads to an increase the convergence degree of current pareto fronts according to the true pareto front. Thus, the population size and evaluation number is decreased as % 30 (Pop. Size=200-200*0.30=140 and Eval. Num.=900-900*0.30=630) for exploiting based approach and % 70 (Pop. Size=200-200*0.70=60 and Eval. Num.=900-900*0.70=270) for exploring based approach. It is also displayed that an increase in the division number leads to elevate the carrying capacity of grid system. An increase in the number of lattice beams cause to use smaller cross-section of steel profiles for design of grillage system. Consequently, DAbYSS is suggested to optimize the design of grillage systems as a fast and accurate evolutionary optimization tool.

Acknowledgment

I would like to thank Prof. Dr. Antonio J. Nebro, Prof. Dr. Juan J. Durillo and their team members for releasing source code of JMETAL to execute computational procedures of multi-objective optimization methods.

7. References


