

A Comparative Benchmark Study of using Different Multi-Objective Optimization Algorithms for Restraint System Design

Monica Majcher
Oakland Univ.

Hongyi Xu
Northwestern Univ.

Yan Fu, Ching-Hung Chuang, and Ren-Jye Yang
Ford Motor Co.

ABSTRACT

Vehicle restraint system design is a difficult optimization problem to solve because (1) the nature of the problem is highly nonlinear, non-convex, noisy, and discontinuous; (2) there are large numbers of discrete and continuous design variables; (3) a design has to meet safety performance requirements for multiple crash modes simultaneously, hence there are a large number of design constraints. Based on the above knowledge of the problem, it is understandable why design of experiment (DOE) does not produce a high-percentage of feasible solutions, and it is difficult for response surface methods (RSM) to capture the true landscape of the problem. Furthermore, in order to keep the restraint system more robust, the complexity of restraint system content needs to be minimized in addition to minimizing the relative risk score to achieve New Car Assessment Program (NCAP) 5-star rating. These call for identifying the most appropriate multi-objective optimization algorithm to solve this type of vehicle restraint system design problem. In this paper, several advanced multi-objective optimization algorithms are employed to solve a large-scale restraint system design problem. The accuracy to the "true" Pareto set and percentage of feasible solutions are employed as the performance metrics to judge how well each algorithm performs. A detailed analysis of why some algorithms perform better than others will be investigated, and the experience gained from this benchmark study will offer some guidance as to how to choose multi-objective optimization algorithms for solving future restraint system design problems.

CITATION: Majcher, M., Xu, H., Fu, Y., Chuang, C. et al., "A Comparative Benchmark Study of using Different Multi-Objective Optimization Algorithms for Restraint System Design," *SAE Int. J. Trans. Safety* 2(2):2014, doi:10.4271/2014-01-0564.

1. INTRODUCTION

Early on computers helped engineers increase computational efficiency dramatically by using tools such as computer-aided design programs and finite-element modeling. Now, many years later, optimization techniques and algorithms give engineers the ability to explore and exploit a design space, whether it is simple or complex. Optimization, when properly understood and utilized, can save many hours in wasted effort exploring infeasible or undesirable designs. However, the complex design spaces of most engineering problems make the selection of the proper optimization algorithm very challenging. Many engineering design problems faced in industry are not simple, convex optimization problems which

can be solved using gradient-based methods. The available heuristic algorithms have the ability to handle the complexity of modern engineering problems, but they have many parameters which must be carefully tuned in order to properly use the algorithm for a specific problem. This can make optimization unappealing and a deterrent for its use for a person without an optimization background.

Automotive safety is an area that has benefitted from optimization techniques in the design of restraint systems [1] and structural design [2]. Safety is a crucial part of the automotive design and development process, with not only the safety of the occupants at stake, but also critical classifications such as NCAP safety star ratings [3], and meeting applicable

FMVSS requirements. With the availability of computer-simulations, results from physical testing can be combined with simulation results to design the most effective restraint system.

Restraint system design problems have been shown to be highly non-linear, non-convex, and contain numerous design variables and constraints. Therefore, these complex design problems cannot be solved using simple gradient-based methods, such as sequential quadratic programming (SQP) or Simplex methods. There are other techniques, however, that can be used for these types of problems. One option is to use a Design of Experiments (DOE). This involves using a space-filling algorithm such as uniform Latin hypercube or random sampling to sample the design space. The problem is then solved directly for these design points. The information gathered from a DOE can provide insight into which variables affect the results of interest the most, known as characterization or screening [4]. These DOE results can also be used to train a metamodel or response surface [4]. Although DOE has become very popular and beneficial in design problems [5], it often is not able to capture the full essence of a design problem with a large number of constraints and design variables.

Evolutionary algorithms, such as genetic algorithms, have become very popular for use in complex engineering design problems. These algorithms start by doing a wide search of the design space, and then concentrate more on local and global optimum. They are heuristic, which means that they may require a large number of iterations in order to find a reasonable, let alone, optimal solution.

In addition to DOE and evolutionary algorithms, hybrid and adaptive algorithms have recently been applied to engineering problems. These algorithms combine several other existing algorithms, such as generic algorithms (GAs) and gradient-based methods, to traverse the design landscape and find feasible design faster. GAs and other heuristics such as neural networks find solutions iteratively, or sequentially [6]. Adaptive algorithms often explore the design space more globally [6], learn from it interactively, and then change the type of algorithm that is used in different areas or at different times in the optimization in order to most effectively move toward the optimal solution.

In this work, we used five different commercially-available algorithms to solve a vehicle restraint system design optimization problem. The algorithms include both traditional evolutionary algorithms and hybrid algorithms. We will begin by giving a brief description of each algorithm, followed by describing the design landscape and setup of the restraint system problem. Then we will show how effective each algorithm was in finding feasible and optimal solutions by using a simple percentage calculation and a Pareto front performance evaluation metric. Some discussion and conclusion are given in the end.

2. ALGORITHM DESCRIPTIONS

Algorithm 1

Algorithm 1 is a combination of the genetic algorithm and sequential quadratic programming. For every generation, the original objectives are converted to one objective, and SQP is applied. The user specifies the percentage of SQP runs allowed out of the total desired evaluations, as well as the crossover probability, mutation probability, etc.

Algorithm 2

Algorithm 2 is a GA-like evolutionary optimization algorithm. Instead of using bit strings, it uses a natural problem-dependent representation in which the chromosome is a vector of design variables and control parameters. In mutation, new designs are generated by adding normally-distributed random values to the chromosomes of existing designs. In crossover, two chromosomes exchange elements randomly to generate new design variables. Starting from an initial population of random designs, the operators are applied in a loop to evolve the designs generation by generation. The sequence of generations is continued until a termination criterion is met.

Algorithm 3

Algorithm 3 is a proprietary hybrid and adaptive search method. For a single objective optimization problem, this algorithm uniquely combines elements of several different search methods and uses them simultaneously to find the optimal solution. The adaptive element tunes the internal parameters of the current algorithms based on the knowledge it gains of the design space. It also determines how much and at what times each optimization method contributes to the overall search strategy. This algorithm only uses direct calculations. No approximate response surfaces are used. For multi-objective problems, the algorithm handles the objectives independently of each other, and uses a unique non-dominated sorting technique to provide a Pareto front of optimal designs.

Algorithms 4 & 5

These algorithms utilize the computational efficiency of response surfaces to enhance the optimization process. Existing designs are used for training different metamodels. An exploration occurs around the current Pareto front points to find virtual promising designs. Next, an optimization algorithm of the user's choice, such as MOGA-II or NSGA-II, is used on the metamodels. Algorithms 4 and 5 throughout this paper each use a different available algorithm within the software only for the portion of the optimization which utilizes the metamodels to find the best potential designs. After this optimization, the best points are validated by using direct evaluation. When these solutions are obtained, they are combined with existing solutions and stored as training points for the metamodels.