2010-06

Orthogonal Array Experiment in Systems

Huynh, Thomas V.

Regular Paper

Orthogonal Array Experiment in Systems Engineering and Architecting

Thomas V. Huynh*

Department of Systems Engineering, Naval Postgraduate School, 777 *Dwyer Road, Monterey,* CA 93943

Received 9 May 2008; Revised 23 March 2010; Accepted 4 June 2010, after one or more revisions
Published online 29 September 2010 in Wiley Online Library (wileyonlinelibrary.com)
DOI 10.1002/sys.20172

ABSTRACT

This paper espouses the application of orthogonal array experiment to solve a class of engineering optimization problems encountered in systems engineering and architecting. It also illustrates the applicability of orthogonal array experiment in systems engineering and architecting with two examples: verification and validation of the performance of a bandwidth allocation algorithm and architecting of a system of systems to respond to small boat attacks by terrorists. The orthogonal array experiment approach does not call for linearization of nonlinear engineering optimization problems; using orthogonal arrays, it solves them directly by carrying out the smallest possible number of experiments and determining their solutions from the results of the experiments. The orthogonal array experiment method has been found to be effective and efficient for these problems. The feasibility of applying the orthogonal array experiment approach to these problems suggests its potential application to other optimization problems encountered in systems engineering and architecting. © 2010 Wiley Periodicals, Inc. Syst Eng 14: 208–222, 2011

Key words: orthogonal array experiment; systems engineering and architecting; engineering optimization problem; assignment problem

1. INTRODUCTION

Three pillars of systems engineering are systems engineering management, systems engineering methodology, and systems engineering methods and tools [Sage, 1992]. Systems engineering methodology and system engineering management are the two pillars that must occur for a successful production of a system. Systems engineering methodology involves system definition, system design and development, and system deployment. Systems management consists of a task management structure, managing systems engineering tasks, and decision making with respect to system development. The third pillar, systems engineering methods and tools, is needed to support systems engineering management and systems engineering methodology. This paper deals with systems engineering methods and tools. Specifically, it deals with systems analysis used in systems engineering and systems architecting. Systems analysis supports many areas, such as requirements analysis, functional analysis, design evaluation, synthesis and allocation of design criteria, determination of system key drivers, system performance assessment, design, development, detail design performance analysis, system performance analysis, trade-off studies, etc. Systems analysis often involves solving optimization problems.

An optimization engineering problem can often be cast as an assignment problem (or a mathematical programming problem). There are many mathematical methods to solve assignment problems of different types, such as integer programming problems (linear and nonlinear), mixed integer programming problems, etc. [Minoux, 1986]. As the dimen-

* E-mail: thuynh@nps.edu

sion of the problem (i.e., the number of assignment variables and constraints) increases, the time it takes to solve the problem increases. Often it is impractical (because it is time-consuming) to solve these assignments problems when a Monte Carlo method is involved. Often a quick solution is needed to enable quick decision making. Often a heuristic algorithm would be preferred over mathematical optimization techniques. Often an optimal solution is needed without having to search for all possible optimal solutions. And often a problem in systems analysis calls for just an approximate solution of an optimization problem with or without constraints. In these cases, the orthogonal array experiment approach has been proven to be effective in providing solutions to the optimization problem.

This paper espouses the application of orthogonal array experiment to solve a class of engineering optimization problems encountered in systems engineering and architecting and illustrates the mechanism of applying orthogonal array experiment to the problems treated in Huynh and Gillen [2001] and Huynh et al. [2007]. The problem addressed in Huynh and Gillen [2001] is a system algorithm performance verification and validation problem in which heuristic algorithms are employed to determine bandwidth to be allocated on demand in a satellite communications system that maximizes satisfaction of requests for bandwidth. The problem addressed in Huynh et al. [2007] is a systems architecting problem in which architectures of a system of systems to respond in a cost-effective manner to terrorists using small boats to attack maritime commerce traffic and critical shore infrastructure. This problem is a subset of a larger problem of architecting a system of systems responding to maritime domain terrorism [Huynh et al., 2009]. The purpose of this paper is thus to demonstrate the applicability of the orthogonal array experiment approach to solving these engineering optimization problems.

These problems are not the only assignment problems encountered in systems engineering and architecting. Indeed, assignment problems abound in systems engineering. Some of these assignment problems tackled in recent times are now mentioned. Attagara [2006] uses heuristics to solve an NP-hard (nondeterministic polynomial-time hard) assignment problem of allocating both the type and the number of explosive scanning devices at airports to different groups of passengers with carry-on baggage so as to maximize the total airport security while satisfying budget, resource, and throughput constraints. Vidalis et al. [2005] model serial flow or production lines as tandem queuing networks and formulate them as continuous-time Markov chains to minimize the average work-in-process when the total service time and the total number of service phases among the stations are fixed. Pettit and Veley [2003] discuss risk allocation in airframe systems engineering in general and in particular the concept of allocating system-level risks in multidisciplinary design problems. Vidal [2003] solves service allocation problems in which a set of services must be allocated to a set of agents so as to maximize a global utility, using a global hill-climbing and a bidding-protocol algorithm. Grundel et al. [2005] apply exact and heuristic methods to solve an NP-complete, nonlinear integer programming problem of determining weapon-to-target pairings that minimize the total expected survival value of the targets after all the engagements. Gao et al. [2007] also solve the weapon-target assignment problem using immune system which serves as a local search mechanism for genetic algorithm. Holness et al. [2006] treat the personnel assignment problems from a systems view, using human factors methodologies and research methods such as macro-ergonomics, human-computer interaction, the skills-rules-knowledge framework, hierarchical task analysis, decision ladders, and abstraction decomposition spaces.

A matrix experiment consists of a set of experiments in which the settings of the various product or process parameters of interest are changed from one experiment to another and from which the data are then analyzed to determine the effects of the parameters on the response of the product or process [Phadke, 1989]. An orthogonal array experiment is a matrix experiment using special matrices, called orthogonal arrays. Section 3.1 discusses orthogonal arrays in some detail. Orthogonal array experiment is used heavily in quality engineering in general and robust design in particular. Quality engineering is concerned with reducing the costs incurred prior to and after the sale of a product [Taguchi, 1978, 1986, 1987; Taguchi, Wu, and Chowdhury, 2004; Taguchi and Wu, 1979; Taguchi and Phadke, 1984; Kackar, 1985, 1986; Clausing, 1988; Byrne and Taguchi, 1986; Bendell et al., 1989]. Robust design is a systematic and efficient method of design optimization for performance, quality, and cost [Phadke, 1989]. Software testing has also benefited from orthogonal array experiment [Taguchi et al., 2004; Phadke, 2009]. Jeang and Chang [2002] combine the use of orthogonal arrays, computer simulation, and statistical methods in the rapid development of new products and the planning and early implementation of product development. In this paper, again, orthogonal array experiment is used to solve engineering optimization problems in systems engineering and architecting formulated as assignment problems (or mathematical programming problems). This paper is not purported to serve as an introduction of robust design. Rather, it espouses the employment of orthogonal array experiment to solve assignment problems.

As discussed in Section 3.2, an orthogonal array used in an orthogonal array experiment employed in robust design does not capture all of the experiments of a full factorial design. Consequently, degradation of the results otherwise obtained with the full factorial design could occur. Sound engineering judgment during the planning phase of the experiment design is therefore necessary in order to properly incorporate potentially significant interactions of factors into the orthogonal array [Peace, 1992]. Similarly, the solutions of assignment problems obtained with the orthogonal array experiment as an approximation solution method could potentially deviate from the optimal solutions (i.e., degradation) obtained with the mathematical programming methods. The solutions of the assignment problems that have been treated exhibit some degradation as well as agreement [Huynh, 1997; Huynh and Gillen, 2001]. As discussed in Section 5.2, the deviations of the solutions of the dynamic bandwidth allocation problem, obtained with the orthogonal array experiment approach, from the optimal solutions, obtained with the commercial optimization tool AMPL and CPLEX [Fourer et al., 1993], range from 0% (the best case) to 41% (the worst case).

Also, as reported in Huynh [1997], the solution of the allocation problem treated by Chu [1969] obtained with the orthogonal array experiment approach is in excellent agreement with that obtained with the integer programming method.

Furthermore, computational time saving achieved with the orthogonal array experiment approach to solving assignment problems has been observed with the problems studied thus far [Huynh, 1997; Huynh and Gillen, 2001; Kessler et al., 2006; Huynh et al., 2007]. For example, a method to solve a nonlinear programming problem involves a transformation of the nonlinear programming problem to a linear programming problem, which is then solved using existing integer programming methods; but, as a penalty, the dimension (i.e., the number of assignment variables and constraints) of the resulting linear programming problem increases significantly. This is the case with a transformation technique [Chu, 1969] that turns a zero-one nonlinear programming problem, in which m functions are assigned to n subsystems of a system, to a zero-one linear programming problem; the dimension increases from $mn^2 + m + n$

$$\text{to } 3n(n-1) \sum_{i=1}^{\min(\mu+1,m)} \binom{m}{i} + mm + mn^2 + 2m + n,$$

where

$$\binom{m}{i} = \frac{m!}{i!(m-i)!}$$

and μ denotes the number of the new variables resulting from the transformation technique [Huynh and Kohfeld, 1994]. The increase in the dimension of the problem, hence an increase in the time to solve it, resulting from the linearization of the problem, can be avoided by using the orthogonal array experiment approach, which does not call for linearization of the nonlinear problem [Huynh, 1997]. Time saving is thus achieved in this case as well as in the problem of architecting a system of systems responding to small boat attacks (SBA) discussed in Section 5.1. Solving this SBA problem using a full factorial design would involve an evaluation of the effectiveness of 3072 possible combinations (architectures) of the SBA system concepts, using Monte Carlo simulation. Such an evaluative effort would take 704 days (or 2 years of around the clock) to complete 100 simulation runs for each combination on a Dell Intel Pentium (R) CPU 3.40 GHz. The orthogonal array experiment approach allows this architecting effort to be completed in a much shorter time [Kessler et al., 2006].

The goals of this paper are:

- Demonstrate the application of the orthogonal array experiment to solve assignment problems encountered in systems engineering and architecting.
- Sketch the steps in solving the engineering optimization problems with the orthogonal array experiment method.
- Illustrate the application of the steps to the problems treated in Huynh and Gillen [2001] and Huynh et al. [2007].

The rest of the paper is organized as follows. Section 2 provides a mathematical description of assignment problems. Section 3 discusses orthogonal array experiment. Section 4 delineates the steps to follow in the application of orthogonal array experiment to solve assignment problems. With excerpts from Huynh and Gillen [2001] and Huynh et al. [2007], Section 5 illustrates the application of orthogonal array experiment to solving, respectively, the problem of dynamically allocating bandwidth in a satellite communication network and the problem of architecting a system of systems responding to attacks by terrorists using small boats. Finally, Section 6 contains some concluding remarks.

2. MATHEMATICAL DESCRIPTION OF ASSIGNMENT PROBLEMS

Let F denote the set of elements f_j and S_j the set of elements $s_{jk_{(j)}}$ to which f_j is to be allocated (assigned). Let X denote a set of allocation variables defined according to

$$X_{jk_{(j)}} = X(k_{(j)}, j) = \begin{cases} 1, & \text{if } s_{jk_{(j)}} \text{ is assigned to } f_j \\ 0, & \text{otherwise} \end{cases},$$

where $j = 1, \ldots, |F|$ and $k_{(j)} = 1, \ldots, |S_j|$; $|S_j|$ denotes the cardinality of S_j. This formulation allows for S_j to be different for the different elements of F, in the sense that $|S_j|$ are different and also that $s_{jk_{(j)}}$ can be different for different values of j.

An assignment problem is of the form:

$$\text{Minimize } z(X_{jk_{(j)}})$$

$$\text{subject to} \begin{cases} g(X_{jk_{(j)}}) \le 0, \\ h(X_{jk_{(j)}}) = 0, \\ X_{jk_{(j)}} \in \{0, 1\}, \end{cases} \tag{1}$$

$$j = 1, \ldots, |F|, k_{(j)} = 1, \ldots, |S_j|.$$

The function z is called the objective function, which is real-valued. The feasible region is the collection of the values of X that simultaneously satisfy X in $\{0, 1\}$ and the constraints $g(X_{jk_{(j)}}) \le 0$ and $h(X_{jk_{(j)}}) = 0$ in (1), for some functions g and h of X. X^* is optimal if it is feasible and if the value of the objective function is not less than that of any other feasible solution: $z(X^*) \le z(X)$ for all feasible X. The problem in (1) involves minimization, but it could just as well involve maximization, with the appropriate change in the meaning of optimal solution: $z(X^*) \ge z(X)$ for all feasible X.

3. ORTHOGONAL ARRAY EXPERIMENT IN QUALITY ENGINEERING

The perspective of this paper is an introduction of the application of orthogonal array experiment to solve assignment problems encountered in systems engineering and architecting. The subject of orthogonal array experiment has been covered widely in quality engineering literature [Taguchi, 1978, 1987, 1993; Taguchi, Wu, and Chowdhury, 2004; Phadke, 1989; Roy, 1990]. This section provides only a sketch of orthogonal array experiment, beginning with a brief discussion of orthogonal arrays.

3.1. Orthogonal Arrays

Orthogonal arrays began with Euler as Latin squares in the 18th century [Euler, 1849]. A Latin square of order s is an $s \times s$ (square) array with s rows and s columns whose s^2 intersections are occupied by s distinct symbols such that each symbol occurs once in each row and once in each column [Bose and Mandel, 1984]. Orthogonal arrays can be obtained from combining orthogonal Latin squares. The columns of an orthogonal array correspond to factors. The factor levels are specified in each row of the orthogonal array. All combinations of levels occur and occur an equal number of times in every pair of columns of an orthogonal array. This combinatorial property ensures the orthogonality property: All columns in the array are orthogonal to each other [Pao, Phadke, and Sherrerd, 1989]. "Orthogonal" means "balanced," "separable," or "not mixed." Taguchi and Wu [1979] have tabulated a number of orthogonal arrays which can be used conveniently to construct orthogonal designs for many experimental situations. The methods of constructing orthogonal arrays can be found in Taguchi and Wu [1979], Taguchi [1987], Taguchi and Konishi [1987], Phadke [1989], Roy [1990], Hedayat, Sloane, and Stufken [1999], and Taguchi, Wu, and Chowdhury [2004].

Not only are orthogonal arrays used in quality engineering and robust design, but, being connected to statistics and coding theory, they are also employed in computer science, cryptography, agricultural, or medical research [Hedayat, Sloane, and Stufken, 1999]. For experiment design, the use of orthogonal arrays allows a significant reduction of the number of experiments and also the simplicity of data analysis [Phadke, 1989; Taguchi and Konishi, 1987].

3.2. Orthogonal Array Experiment

Experiment design has a long history. Its theory can be found in Cochran and Cox [1957], John [1971], Hicks [1973], Daniel [1976], and Box, Hunter, and Hunter [1978, 2005]. Factorial design of experiments, pioneered by R.A. Fisher in the 1920s [Pearson, 1974], has been in use since Fisher's work in agricultural experimentation [Fisher, 1951]. For a full factorial design, the number of possible conditions or experiments is L^m, where m is the number of factors and L is the number of levels for each factor. When the number of factors and levels per factor are large, the number of possible conditions (or experiments) become extremely large. An efficient form of factorial design of experiments is thus needed. Orthogonal array experiment, which meets that need, requires only a smaller number of unique factor/level combinations captured in an orthogonal array.

Again, an orthogonal array experiment is a matrix experiment using orthogonal arrays. In experiment design, the matrix experiments are also known as designed experiments or runs or conditions, settings as levels, and parameters are factors [Taguchi, 1978, 1987, 1993; Taguchi, Wu, and Chowdhury, 2004; Phadke, 1989; Roy, 1990]. An orthogonal array experiment, which is an important technique in robust design, allows efficient determination of the effects of several parameters. Robust design, whose foundations were developed by Taguchi in the early 1960s, is a systematic and efficient method of design optimization for performance, quality, and cost. The factors being examined and their associated levels constitute the so-called experimental region. Optimum product or design results from the best or the optimum level for each factor, which gives the optimum value of the response in the experimental region [Phadke, 1989].

3.3. Analysis of Data from Experiments

An orthogonal array experiment involves three main steps: use of an orthogonal array to plan the matrix experiment; run the experiments; and use of arithmetic averages of the responses in determining the effect of a factor level [Roy, 1990]. The last step is part of the data analysis in the orthogonal experiment, which is now briefly described. Section 4 discusses the data analysis in detail as it pertains to solving assignment problems.

The purpose of the data analysis is to achieve one or more of these objectives: to establish an optimum condition for a product or a process; to estimate the contribution of individual factors; and/or to estimate the response under an optimum condition [Roy, 1990]. The optimum condition is established from analyzing the main effects of each of the factors, which indicate the effects of a factor on the objective function when it goes from one level to another. The analysis of the main effects, or the estimation of factor effects**,** involves the calculation of the arithmetic averages of the responses or of the signal-to-noise ratios (S/N) of the responses for the levels of the factors. The S/N ratio measures the sensitivity of the quality characteristic (response) to those noise factors or uncontrollable factors. Product or process design with highest S/N ratios always yields the optimum quality with minimum variance [Roy, 1990]. Section 4.3 defines the S/N ratio, a concept developed by Taguchi. The response (objective function) under the optimum condition is then obtained by running a confirmatory experiment. This approach is sometimes called analysis of means (ANOM) [Phadke, 1989], which is employed in the illustrating problems in this paper.

The analysis of data obtained from an orthogonal array experiment also involves analysis of variance (ANOVA), needed for estimating the relative importance of the factors and the error variance. ANOVA is a standard statistical treatment of the experimental results, which provides a measure of confidence in the results and compares the variances of the factors to determine the relative contribution of each factor. The reader is referred to Taguchi and Wu [1979], Taguchi [1987], Phadke [1989], Roy [1990], and Taguchi et al. [2004] for detailed discussions of the ANOVA method used in quality engineering and robust design.

4. ORTHOGONAL ARRAY EXPERIMENT FOR SOLVING ASSIGNMENT PROBLEMS

The application of an orthogonal array experiment to solve assignment problems of the form in (1) that occur in systems engineering and architecting involves the following steps.

Step 1. Formulate the engineering optimization problem.
Step 2. Transform the engineering optimization problem to an assignment problem.
Step 3. Perform an orthogonal array experiment to solve the assignment problem.
a. Perform experiment design and planning.
b. Run the experiments.
c. Perform data analysis.
d. Obtain an optimal solution.
e. Run a confirmatory experiment with the optimal solution (to confirm the optimized objective function).

Step 1 varies from application to application and is unique to a specific application at hand. Step 1 will be illustrated with the problems in Section 5. Step 2 turns the engineering optimization problem formulated in Step 1 into an assignment problem of the form (2). Like Step 1, Step 2 varies with the specific application at hand. Step 3—Perform an orthogonal array experiment—is now elaborated.

4.1. Perform Orthogonal Array Experiment

Performing an orthogonal array experiment involves the following.

4.1.1. Perform Experiment Design and Planning

A key step in designing the orthogonal array experiment is defining the factors and their levels [Pao, Phadke, and Sherrerd, 1989; Phadke, 1989; Roy, 1990; Taguchi, 1987; Taguchi, Wu, and Chowdhury, 2004]. The factors are the causes which produce an effect, their levels are the ways in which the factors are changed, and the response is the result produced by the factors. There is no systematic way to define factors and their levels for assignment problems in general; care should be exercised in choosing factors and their levels appropriately so as to take advantage of the available orthogonal arrays and so as to minimize the dimensions of the arrays [Huynh, 1997].

As mentioned before, a full factorial design to explore all possible factor-level combinations would impractically require performing an exorbitant number of experiments. The use of orthogonal arrays drastically reduces the number of experiments. As pointed out in Huynh [1997], the nature of an assignment problem of interest dictates the selection of the factors and their levels and orthogonal arrays. Also, a large number of factors and levels will require a large orthogonal array. Large orthogonal arrays, if not already available, can be readily generated or requested from ASI [1987].

4.2. Run Experiment

An experiment in the orthogonal array experiment application espoused in this paper refers in general to a computation performed by a computer program or a computer simulation. In either case, input to the computation or to the simulation consists of the orthogonal array selected for the problem at hand and the data pertinent to the problem. The response is then calculated for each of the experiments. Finally, the obtained responses are then processed according to the data analysis described in Section 4.3.

4.3. Perform Data Analysis

Again, the purpose of the data analysis is to achieve one or more of these objectives: To establish an optimum condition for a product or a process; to estimate the contribution of individual factors; and to estimate the response under an optimum condition [Roy, 1990]. The optimum condition is established from analyzing the main effects of the factors, which involves the calculation of the arithmetic averages of the responses or of the signal-to-noise ratios of the responses for the levels of the factors. The response (objective function) under the optimum condition is then obtained by running a confirmatory experiment. The discussion of the two averages of the responses follows.

The average effects are the arithmetic averages of the responses, based on which the optimum levels of the factors are selected [Roy, 1990]. Let a_{ij} denote the level of the jth factor in the ith experiment (row), z_i the response (i.e., the objective function) corresponding to the ith experiment. Then the average performance (i.e., the arithmetic average of the objective function or of the response) of the jth factor at the α_jth level, denoted by $\overline{f}_{j\alpha_j}$, is calculated according to

$$\overline{f}_{j\alpha_j} = \frac{1}{N_{j\alpha_j}} \sum_{i=1}^{N_\varepsilon} \delta(a_{ij} - \alpha_j) z_i, \tag{2}$$

where

$$N_{j\alpha_j} = \sum_{i=1}^{N_\varepsilon} \delta(a_{ij} - \alpha_j),$$

is the number of experiments (rows) in which the α_jth level is assigned to the jth factor, the mathematical device

$$\delta(a_{ij} - \alpha_j) = \begin{cases} 1, & \text{if } a_{ij} = \alpha_j \\ 0, & \text{otherwise} \end{cases}$$

identifies the experiment (trial) in which a_{ij} is assigned the value α_jth level, where $j = 1, \ldots, |F|$, and $\alpha_j = 1, \ldots, |S_j|$. The expression of $\overline{f}_{j\alpha_j}$ in (2) is a compact way of expressing the arithmetic averages calculated in an orthogonal array experiment. Explicitly expressed, as an example, the average performance $\overline{f}$ of factor 1 at level 2, is, according to (2),

$$\overline{f}_{12} = \frac{1}{N_{12}}(z_1 + \cdots + z_{N_{12}}),$$

where N_{12} is the number of experiments in which the level of factor 1 is equal to 2 and z_k is the response from the kth experiment of those N_{12} experiments.

For the purpose of design optimization in a robust design experimentation, the response of a product/process is called a quality characteristic [Phadke, 1989]. The parameters (also called factors) influence the quality characteristic. In a maximization problem, the quality characteristic of the problem is the larger-the-better performance, using the quality engineering parlance [Taguchi, 1978]. If the quality characteristic of the problem is the larger-the-better performance, the levels of the factors are identified according to the maximum average performance. In this case, if $\overline{f}_{jk} = \max_{\alpha_j} \overline{f}_{j\alpha_j}$, then the kth level is assigned to the jth factor. That is, the selected level of a factor is the level that produces the highest average performance. If the quality characteristic of the problem is the smaller-the-better performance (again, using the quality engineering parlance [Taguchi, 1978]), the levels of the factors are identified according to the minimum average response. In this case, if $\overline{f}_{jk} = \min_{\alpha_j} \overline{f}_{j\alpha_j}$, then the kth level is assigned to the jth factor. That is, the selected level of a factor is the level that produces the smallest average performance. In either case, the levels assigned to these factors constitute an optimal solution (optimal condition).

For experiments with repetitions (i.e., when an experiment is independently repeated), the response η_i, which now stands for the S/N corresponding to the ith experiment, is calculated, according to Phadke [1989] and Roy [1990], as

$$\eta_i = -10 \log\left(\frac{1}{M}\sum_{r=1}^{M}(\eta_{ir})^2\right)$$
$$= -10 \log\left(\frac{1}{M}(\eta_{i1}^2 + \eta_{i2}^2 + \cdots + \eta_{iM}^2)\right), \tag{3}$$

in which M is the number of repetitions of each of N_ε experiments and η_{ir} is the response of the rth repetition of the ith experiment. The logarithmic function, log, is with respect to base 10. The average performance (i.e., the arithmetic average of the objective function or of the response) is similarly calculated using (2), and the optimum levels of the factors are then selected. The argument of the logarithmic function in (3) is the mean squared response for the ith experiment, which is the arithmetic average of the squares of the responses of the replications of the ith experiment. The use of (3) to determine the optimum levels of the factors is justified by an additive model, which approximates the relationship between the response variable and the factor levels. The reader is referred to Phadke [1989] for a detailed discussion of additive models and the justification for the use of S/N ratios in determining the main effects of the factors.

In this paper and for the illustrating problems, an experiment is carried out only once. Therefore, $\eta_i = -20 \log z_i$. Then, (2) becomes, with $\overline{\eta}_{j\alpha_j}$ replacing $\overline{f}_{j\alpha_j}$,

$$\overline{\eta}_{j\alpha_j} = -\frac{20}{N_{j\alpha_j}}\sum_{i=1}^{N_{j\alpha_j}}\log(z_i) = -20\log\left(\prod_{i=1}^{N_{j\alpha_j}} z_i\right)^{1/N_{j\alpha_j}},$$

where $j = 1, \ldots, |F|$ and $\alpha_j = 1, \ldots, |S_j|$.

Since the logarithm of the geometric mean and the arithmetic mean increase or decrease together with their arguments, one can use either the average S/N ratio $\overline{\eta}_{j\alpha_j}$ or the arithmetic mean $\overline{f}_{j\alpha_j}$ to determine the optimum levels of the factors. The ANOM is thus carried out to estimate effects of the factors, but with η_i now playing the role of z_i; if the averaged response is minimized, then the S/N ratio is maximized. The illustrating problems in this paper, in which all experiments are carried out only once, involve the use of the arithmetic mean $\overline{f}_{j\alpha_j}$ and do not necessarily involve ANOVA. The reader is referred to Phadke [1989] for the use of S/N ratios in robust design, where actual experiments in some case studies therein are carried out with repetitions.

Furthermore, when the optimization problem is without constraints, the data analysis is straightforward. When the optimization problem is with constraints, the constraints must be implemented judiciously, as shown in the example in Section 5.2. The implementation of the constraints can vary from problem to problem. In the end, the solution that satisfies the constraints is an optimal solution of the problem.

5. APPLICATION OF ORTHOGONAL ARRAY EXPERIMENT

For illustration purposes, again, this paper discusses the application of orthogonal array experiment to the problem of dynamically allocating bandwidth in a satellite communication network and the problem of architecting a system of systems to respond to attacks by terrorists using small boats. The discussion of the application of orthogonal array experiment to these problems is excerpted from Huynh and Gillen [2001] and Huynh et al. [2007], respectively, and focuses on the implementation of the steps (working mechanics) discussed in Section 4. Detailed discussions of the application of orthogonal array experiment to these problems can be found in the cited references.

5.1. Architecting a System of Systems Responding to Maritime Domain Terrorism

Considered to be the most likely attack in the future, a small boat attack (SBA) in U.S. waters and ports is an attack by a single terrorist already in the U.S. who, in an explosive-laden small boat, blends in with recreational boaters to get close to high-value units (HVU) and assaults them at a high speed. There is thus a need to develop a system of systems (SoS) to respond to such terrorist threats. This SoS is called the maritime threat response (MTR) SoS.

The crux of the MTR SoS architecting problem is to develop architectures of a conceptual, cost-effective, near-term system of systems (SoS) to respond to small boats used by terrorists to attack maritime commerce traffic and critical shore infrastructure and to do so with minimal impact on commerce and economic cost. The near-term MTR SoS consists of systems that are currently in service, in development, and commercial-off-the-shelf technologies or systems that would be available and/or could be developed within the next 5 years. The systems that constitute the SoS include hardware, software, and human resources.

Step 1. Formulate the Engineering Optimization Problem

The mission to thwart a small boat attack includes searching and detecting the threat (the attacker), neutralizing the detected threat, and supporting and maintaining the MTR SoS components. A functional analysis performed by Kessler et al. [2006] identifies five top-level SoS functions: (1) Command, Control, Computers, Communication, Intelligence, Surveillance, and Reconnaissance (C4ISR), (2) Prepare the Battlespace, (3) Find/Fix Threat, (4) Finish Threat, and (5) Sustain. The C4ISR function ensures that the SoS has the appropriate means to carry out a mission in terms of command and control and to have appropriate communication channels to keep the forces informed of the status of operations. The Prepare-the-Battlespace function ensures that the SoS has the appropriate personnel, equipment, and platforms to carry out the mission; it also renders the area of operations ready for countering a potential attack. The Find/Fix and Finish functions are executed as MTR forces actually carry out the mission. (Hereafter, for convenience, the word "Threat" will be excluded from the Find/Fix and Finish functions.) The Sustain function ensures that all units and equipment are properly supported and maintained for the duration of operations. As the system concepts for Sustain are unique, system concepts corresponding only to the first four top-level functions are identified for use in an MTR SoS. Sustain will thus be excluded from the formulation of the MTR SoS architecting problem as an assignment problem. A concise elaboration of these functions can be found in Kessler et al. [2006] and Huynh et al. [2007]. Some functions may be supported by as many as four different system concepts, while the others by as few as two concepts. The four systems concepts supporting the C4ISR function are: Area Control/Problem-solving; Area Control/Objective-oriented; Local Control/Problem-solving; and Local Control/Objective-oriented. The four systems concepts supporting the PBS function are: Small escorts, Medium escorts, Small and medium escorts, and HVU-based escort teams. The four systems concepts supporting the Finish function are: Organic weapons only, Organic weapons and armed helicopters (air support), Organic weapons and USVs, and Organic weapons, USVs, and armed helicopters (air support). Finally, the two systems concepts supporting the Find/Fix function are: Visual detection; and Visual detection with surface search radar support.

The counter-SBA mission is declared a success if the terrorist attack boat is prevented from reaching a certain lethal range of a protected asset. If the terrorist attack boat is still alive when reaching within the lethal range of the protected asset, then the counter-SBA mission is a failure.

The total cost of an SoS is contributed by the cost of procurement of both additional existing and new SoS components (platforms), the cost of operating and supporting (O&S) the SoS, and the cost associated with both the time delay suffered by commerce (the ferries and the oil tankers) in the course of responding to an attack and damage to the physical entities resulting from failures to neutralize the terrorist threat. Table I contains the cost estimates of the system concepts. Parenthetically, Table I does not show the costs of the Find/Fix system concepts, for they incur no cost as they are already accounted for under PBS and the radar is organic to both the small and mid-sized escorts; these Find/Fix system concepts are, however, included in the experiments for the purpose of assessing the SoS effectiveness. The delay and damage costs are generated by the mission-level modeling and simulation. The estimation of the remaining costs is discussed in [Kessler et al., 2006; Huynh et al., 2007].

The cost of procuring the SoS is fixed, but the remaining costs change with mission execution. The latter are determined by Monte Carlo simulation [Kessler et al., 2006]. The cost of procuring the SoS depends on the number of platforms in the SoS architecture and the cost of each platform. The O&S cost reflects the number of days per year during which the platforms in the SoS would be involved in MTR-related activities and the daily O&S rate, which accounts for both the system and personnel O&S rates. The average O&S costs for selected classes of ships and aircraft form the basis for MTR SoS platform O&S cost estimates. The details of the estimation of the O&S cost can be found in Kessler et al. [2006].

An MTR SoS architecture is a combination of the system concepts that perform the top-level functions. The problem is thus to determine which pertinent system concept is assigned to a top-level function, so that, put together, the assigned

Table I. Cost Estimates* [Huynh et al., 2007]

Function	System Concepts		Total Cost ($FY2006 Million)
C4ISR	1	Area Control/ Problem-solving	12.1
	2	Area Control/ Objective-oriented	12.1
	3	Local Control/ Problem-solving	60.1
	4	Local Control/ Objective-oriented	60.1
PBS	1	Small boats only	92.6
	2	Medium escort ships only	1534.8
	3	Small boats & medium escort ships	1583.9
	4	HVU-based team transport	36.1
Finish	1	Organic weapons only	0.8
	2	Organic weapons & armed helicopters	13.8
	3	Organic weapons and USVs	21.3
	4	Organic weapons, USVs, & armed helicopters	35.7

*Note that Table I does not include the Find/Fix system concepts, for they incur no cost as they are already accounted for under PBS and the radar is organic to both the small and mid-sized escorts.

system concepts result in an optimal MTR SoS architecture, in the sense that it maximizes some objective function of performance and cost.

Step 2. Transform the Engineering Optimization Problem to an Assignment Problem

In this case F denotes the set of the SoS top-level functions, f_j, $j = 1, \ldots, 4$, and S_j the set of the system concepts that can perform function f_j. The set of allocation variables is defined according to

$$X_{jk} = \begin{cases} 1, & \text{if system concept } k \text{ of } S_j \text{ is assigned to function } f_j, \\ 0, & \text{otherwise} \end{cases}$$

where $j = 1, \ldots, 4$ and $k = 1, \ldots, |S_j|$; $|S_j|$ denotes the number system concepts in S_j. In this case, $|S_3| = 2$ and $|S_1| = |S_2| = |S_4| = 4$.

The SBA mission success or failure is related to the allocation of the system concepts to the top-level functions; that is, it is a function of X_{jk}. Both the total cost of an SoS architecture and the probability of mission success, P_s, thus depend on the allocations X_{jk}. The probability of mission success is the fraction of the number of Monte Carlo simulation runs in which the SBA mission is a success.

A dimensionless objective function, z, is introduced as $z = \rho(P_s, C)$, which, by means of a rule ρ, is a function of the performance measure P_s, and the total system cost C. The objective function z is thus a function of the allocations X_{jk}. A specific rule ρ will be elaborated in the data analysis.

The problem of optimizing the MTR SoS architecture then amounts to determining an assignment of the system concepts to the four SoS top-level functions (i.e., the allocations X_{jk}) that maximizes the objective function z.

Step 3. Perform Orthogonal Array Experiment

- **Perform Experiment Design and Planning.** In this problem, the SoS top-level functions—C4ISR, PBS, Find/Fix, and Finish—are the factors. Thus, functions 1, 2, 3, and 4 correspond to C4ISR, PBS, Find/Fix, and Finish, respectively. The system concepts supporting a function are the levels of the corresponding factor.

As aforementioned, for a full factorial design 3072 possible combinations (architectures) of these system concepts need to be evaluated for their effectiveness, using Monte Carlo simulation. Each simulation run on a Dell Intel Pentium (R) CPU 3.40 GHz computer takes more than 3 min. It would therefore take 704 days (or 2 years around the clock) to evaluate those potential combinations, with each combination requiring 100 simulation runs. This would be impractical.

The appropriate orthogonal array for this application is a portion of the mixed orthogonal array $L_{32}(2^1 \times 4^9)$ [Taguchi and Konishi, 1987], shown in Table II. The columns of the array correspond to the functions (factors). Each of the 32 rows (or conditions) corresponds to an architecture trial (experiment). The values, ranging from 1 to 4, represent the system functions. This orthogonal array requires 32 separate experiments and can be used to study up to 9 factors with 4 levels per factor and 1 factor with 2 levels.

- **Run Experiment.** Carrying out an experiment (corresponding to a row of the orthogonal array) in this case means performing a Monte Carlo simulation of the MTR SoS response, z, to an attack of an HVU by a small boat attacker. The Monte Carlo simulation involves 2000 simulation runs of an SBA mission model, each of which produces success or failure of the SBA mission. The Monte Carlo simulation results are then processed to yield the probability of mission success.

The SBA mission model represents the C4ISR, PBS, Find/Fix, and Finish functions, the kinematics of the small boat attacker, the response engagement geometry, the engagement types (i.e., warning and lethal engaging), the engagement sequence (e.g., helicopter followed by close escorts or the onboard team), and the MTR SoS responses. The SBA mission model is elaborated in detail in Kessler et al. [2006] and Huynh et al. [2007].

For each Monte Carlo run, the output of the SBA mission model is mission success or failure. Postprocessing yields the probability of mission success for each of the 32 experiments. Table III displays the experimental results—the probability of mission success (in the second column) and the total cost (in the third column) for each of the 32 experiments (in the first column)—and the dimensionless response (in the fourth column) defined above.

Table II. The Orthogonal Array Reduced from $L_{32}(2^1, 4^9)$

Trial	C4ISR	PBS	Find/Fix	Finish
1	1	1	1	1
2	1	2	2	2
3	1	3	1	3
4	1	4	2	4
5	2	1	2	3
6	2	2	1	4
7	2	3	2	1
8	2	4	1	2
9	3	2	2	2
10	3	1	1	1
11	3	4	2	4
12	3	3	1	3
13	4	2	1	4
14	4	1	2	3
15	4	4	1	2
16	4	3	2	1
17	1	4	2	3
18	1	3	1	4
19	1	2	2	1
20	1	1	1	2
21	2	4	1	1
22	2	3	2	2
23	2	2	1	3
24	2	1	2	4
25	3	3	1	4
26	3	4	2	3
27	3	1	1	2
28	3	2	2	1
29	4	3	2	2
30	4	4	1	1
31	4	1	2	4
32	4	2	1	3

Table III. Experimental Results [Huynh et al., 2007]

Trial	Probability of Success, P_s	Cost ($M)	Response, z
1	0.53	105.5	72.7
2	0.76	1560.6	48.2
3	0.69	1617.3	40.8
4	0.68	83.8	86.5
5	0.59	140.3	76.8
6	0.82	1568.1	53.4
7	0.70	1609.8	41.7
8	0.63	49.0	83.7
9	0.77	1616.2	47.4
10	0.54	188.4	71.3
11	0.71	97.1	88.4
12	0.70	1657.9	40.0
13	0.81	1608.7	51.3
14	0.58	153.6	76.0
15	0.63	131.9	81.3
16	0.70	1665.3	39.8
17	0.39	61.9	62.8
18	0.77	1596.8	48.3
19	0.62	1582.5	35.7
20	0.61	125.9	79.0
21	0.26	69.4	51.1
22	0.80	1631.6	49.0
23	0.61	1547.6	35.7
24	0.70	118.4	87.7
25	0.76	1679.7	44.7
26	0.40	117.5	61.2
27	0.63	166.5	79.9
28	0.63	1595.7	35.9
29	0.81	1644.9	49.8
30	0.24	110.0	48.1
31	0.71	174.0	86.4
32	0.63	1630.6	35.1

• **Perform Data Analysis.** The rule ρ that amalgamates the cost C_i, and the probability of success, P_{S_i}, associated with the ith experiment (row) into a the dimensionless response z_i of the ith experiment is simply

$$z_i = \frac{1}{2}\left[\frac{100}{\pi_{max} - \pi_{min}}(P_{S_i} - \pi_{min}) + \frac{100}{\xi_{max} - \xi_{min}}(\xi_{max} - C_i)\right],$$

in which $\xi_{max} = max_{i\in N_\varepsilon} C_i$, $\xi_{min} = min_{i\in N_\varepsilon} C_i$, $\pi_{max} = max_{i\in N_\varepsilon} P_{S_i}$ and $\pi_{min} = min_{i\in N_\varepsilon} P_{S_i}$, and N_ε denotes the number of experiments (rows).

The average performance of the jth factor (function) at the α_jth level is calculated according to (2), with $\alpha_j = 1, \ldots, |S_j|$ and

$$N_{j\alpha_j} = \begin{cases} 8, & j = 1, 2, 4 \\ 16, & j = 3 \end{cases}.$$

• **Obtain an Optimal Solution.** Figure 1 displays the graphs of the calculated $\bar{f}_{j\alpha_j}$ against the system concepts for each function. [Note that the use of the lines to connect the discrete values in the graphs shown in Fig. 1 is a common practice in the realm of the Taguchi method. The graphs do not represent mathematical functional relationships between the average responses (average performances) and the factor levels. The connecting lines aid in visually explicating the differences of the values and also the existence of interactions of factors.] The calculation of $\bar{f}_{j\alpha_j}$ is now illustrated with that of $\bar{f}_{41}$, the average performance of the Finish function carried out by the Organic-weapons-only system concept. In this case, as the values of the elements in the Finish column and rows 1, 7, 10, 16, 19, 21, 28, and 30 in Table II are "1", (2) becomes $\bar{f}_{41} = \frac{1}{8}(z_1 + z_7 + z_{10} + z_{16} + z_{19} + z_{21} + z_{28} + z_{30})$, which, making use of the values of the corresponding responses, z_i, i = 1, 7, 10, 16, 19, 21, 28, and 30, in the fourth column of Table III, yields $\bar{f}_{41} \approx 50$. The remaining $\bar{f}_{j\alpha_j}$ are similarly calculated.

For this problem, the quality characteristic is the larger-the-better performance (the overall objective function), and the system concepts for the functions are therefore identified according to the maximum average response. The selected system concept for a function corresponds to the largest $\bar{f}_{j\alpha_j}$. As shown in Figure 1, $\bar{f}_{12}$ is the largest among the values $\bar{f}_{1\alpha_j}$, $\bar{f}_{21}$ the largest among the values $\bar{f}_{2\alpha_j}$, $\bar{f}_{32}$ the largest among the values $\bar{f}_{3\alpha_j}$, and $\bar{f}_{44}$ the largest among the values $\bar{f}_{4\alpha_j}$. In other words, the maximum average responses (performances) are obtained with function 1 performed by system concept 2, function 2 by system concept 1, function 3 by system concept 2, and function 4 by system concept 4. This means that the optimal cost-effective MTR SoS architecture consists of area control/objective-oriented for C4ISR, small escorts (boats) for PBS, visual detection with surface search radar support for Find/Fix, and organic weapons, USVs, and armed helicopters for Finish. This optimal cost-effective MTR SoS architecture results in a 0.72 probability of mission success and costs $188.6 M.

To confirm that the resulting cost-effective SoS architecture is indeed a best architecture, its performance and cost are compared with those of two additional architectures, namely, an optimal effective architecture (i.e., maximum-performance SoS architecture) and a heuristic cost-effective SoS architecture. The orthogonal array experiment approach is also employed to develop the optimal effective architecture, but with the objective function being the probability of mission success; the cost is not considered. The heuristic cost-effective SoS architecture consists of the lowest cost system concepts that would meet system effectiveness requirements [Kessler et al., 2006]. The components of these two architectures along with those of the optimal cost-effective SoS architecture can be found in Huynh et al. [2007]. Of the three architectures, the optimal cost-effective architecture is found to be the best MTR SoS architecture for the SBA mission [Huynh et al., 2007].

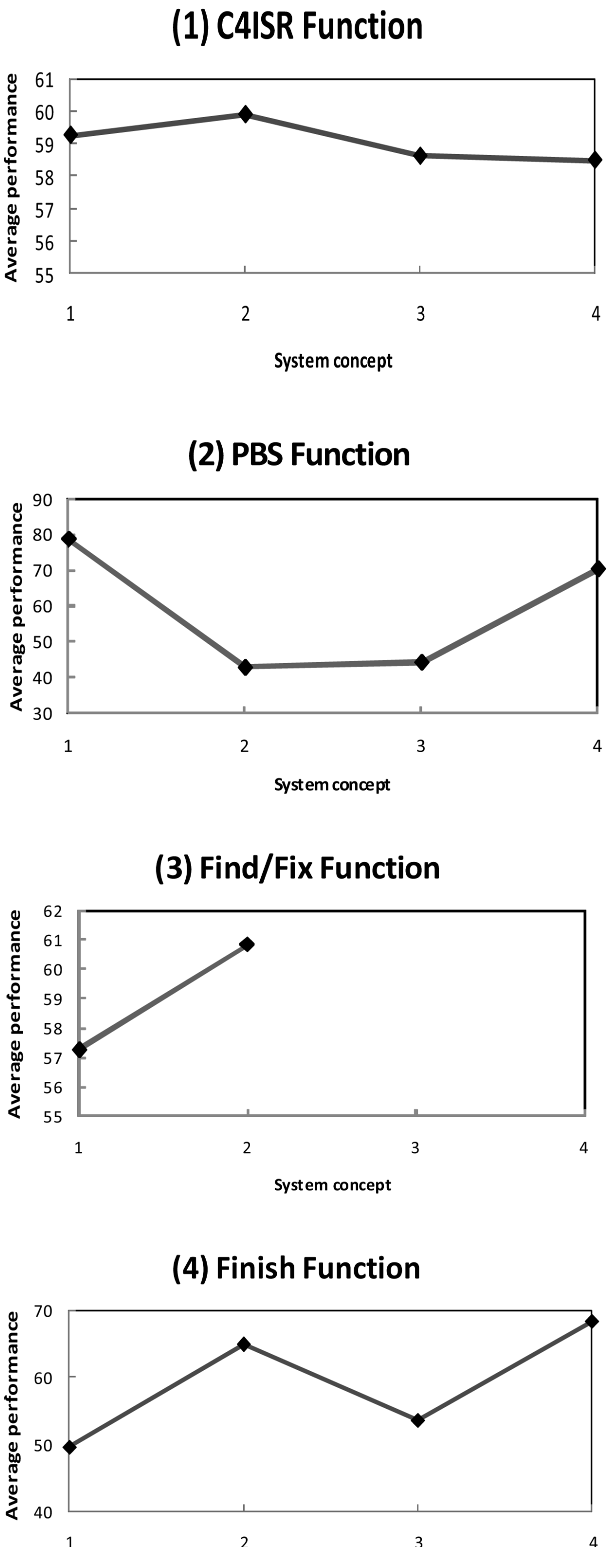

Figure 1. Effects of the SoS functions on the objective function.

5.2. Dynamic Bandwidth Allocation in a Satellite Communication Network

Since satellite communications networks offer wide earth coverage, high-bandwidth, and star topologies suited for common multimedia applications, extension of terrestrial networks to include satellite communication links becomes attractive. Asynchronous transfer mode (ATM), a technology that has promised to revolutionize satellite communications, enables efficient transmission of multimedia traffic with different quality-of-service (QoS) requirements [McDysan and Spohn, 1999]. Demand-Assignment Multiple-Access (DAMA) protocol, a channel access method, is important in providing system capacity and QoS parameters like cell loss, cell delay, and cell delay variation [Biswas and Izmailov, 2000]. Time slots in communication channels constitute bandwidth. It need be dynamically allocated on demand so as to maximize satisfaction of requests for bandwidth from connections while meeting certain transmission and capacity constraints, maintaining fairness among requests, and minimizing cell delay variation (CDV).

To this end, DAMA algorithms need be developed to optimize bandwidth allocation in real time. The algorithms must be efficient and robust. Their efficiency ensures a fast response to a connection's demand for bandwidth. Their robustness ensures satisfaction of the connection's required QoS and its integrity under unpredictable traffic burstiness. Computational practicality dictates that only heuristic DAMA algorithms be implemented in an ATM satellite communications system. These heuristic algorithms must be validated and benchmarked before the system using these algorithms is built. Since simulation is often expensive, optimization appears to be a preferred option used to validate and benchmark such algorithms. This application of the orthogonal array experiment approach deals with validation and benchmarking of such algorithms.

Step 1. Formulate the Engineering Optimization Problem

A generic satellite communications network consists of subscriber terminals (STs), a network controller (NC), and satellites. The network controller, the central control for the network, manages the uplink bandwidth allocation to all STs. All STs share a common pool of bandwidth. The NC, where the DAMA assignment algorithm resides, is responsible for allocating bandwidth to the STs.

Each ATM service category has a certain priority level. A request for bandwidth specifies the minimum and maximum numbers of time slots needed for each priority level. The DAMA assignment algorithm allocates available bandwidth (channels and time slots) to satisfy requests from the STs, collected at the beginning of each assignment cycle. In addition to maximizing the satisfaction of each request, the DAMA assignment algorithm must also minimize the CDV and maintain fairness among the STs. Fairness among the STs may be achieved by minimizing the differences in the percentage satisfaction levels of the STs. The CDV is minimized by equally distributing the assigned time slots across a frame [Huynh and Gillen, 2001].

The problem is then to determine the bandwidth allocation on demand that maximizes satisfaction of requests for bandwidth from connections while meeting certain transmission and capacity constraints, maintaining fairness among requests, and minimizing cell delay variation.

Step 2. Transform the Engineering Optimization Problem to an Assignment Problem

The allocation of bandwidth on demand in an ATM satellite communications system is formulated as a nonlinear mixed integer assignment problem. The fine points of formulating such a problem are elucidated in Huynh and Gillen [2001].

Let I denote the set of subscribers terminals (ST), J the set of traffic (or service) priority levels, K the set of channels (frequencies), and L the set of time slots (or slots) in a channel. Except where noted, i, j, k, and ℓ hereinafter represent the elements of the sets I, J, K, and L, respectively. Let $X_{ik\ell} \in \{0, 1\}$ represent the assignment of slots and channels to an ST, defined by

$$X_{ik\ell} = \begin{cases} 1, & \text{if ST } i \text{ is assigned to slot } \ell \text{ in channel } k \\ 0, & \text{otherwise} \end{cases}.$$

Let Z_{ij} denote the number of slots assigned to satisfy requests from the ith ST for traffic with priority j. Let $Y_{ik} \in \{0, 1\}$ denote allocation function that assigns any slot at all in channel to the ith ST according to

$$Y_{ik} = \begin{cases} 1, & \text{if ST } i \text{ is assigned to any slot in channel } k \\ 0, & \text{otherwise} \end{cases}.$$

Let $W_{i\ell_1\ell_2} \in \{0, 1\}$ denote the so-called spacing variable defined by

$$W_{i\ell_1\ell_2} = \begin{cases} 1, & \text{if ST } i \text{ uses both slots } \ell_1 \text{ and } \ell_2 \text{ but no slots in between} \\ 0, & \text{otherwise} \end{cases}.$$

The assignment problem in this illustration is the nonlinear mixed integer program [Huynh and Gillen, 2001]:

$$\text{Minimize } \xi = \lambda_\sigma \sum_i \left| W_{i\ell_1\ell_2}(\ell_2 - \ell_1) - W_{i\ell_1\ell_2}\frac{|L|}{D_i} \right| - \lambda_\chi \sum_{i,k,\ell} X_{ik\ell} + \lambda_\tau \sum_j (ub_j - lb_j) \tag{4}$$

subject to

$\sum_k X_{ik\ell} \le 1, \ \forall i, \ell$	(One ST transmitting on one channel during a time slot)
$\sum_i X_{ik\ell} \le v_{kl}, \ \forall k, \ell$	(One ST per available time slot)
$\sum_\ell X_{ik\ell} \le c_k Y_{ik}, \ \forall i, k$	(Time slots per channel assigned to an ST not to exceed available time slots)
$\sum_k Y_{ik} \le ch_{max}, \ \forall i$	(Channels assigned to an ST not to exceed maximum channels, ch_{max})
$\sum_{k,\ell} X_{ik\ell} \le \sum_j Z_{ij}, \ \forall i$	(Time slots in channels assigned to an ST to be equal to time slots needed to satisfy request with all priorities)
$min_{ij} \le Z_{ij} \le max_{ij}, \ \forall i, j$	(Size of each assignment within the range of the requested time slots)
$lb_j \le \dfrac{Z_{ij} - min_{ij}}{max_{ij} - min_{ij}} \le ub_j$	(Percentage satisfaction levels among STs to be as close as possible)
$W_{i\ell_1\ell_2} = \left(\sum_k X_{ik\ell_1}\right)\left(\sum_k X_{ik\ell_2}\right)\left(\prod_{\ell=\ell_1+1}^{\ell_2-1}\left(1 - \sum_k X_{ik\ell}\right)\right), \ \forall i, \ell_1 < \ell_2$	(Time slot spacing constraint)

(5)

In (4), the first component within the absolute value represents the distance (i.e., the number of slots) between the slots ℓ_1 and ℓ_2 assigned to the ith ST. The second component is the distance between any two assigned slots if D_i slots are assigned to the ith ST, where D_i is the average of the minimum and maximum number of slots, min_{ij} and max_{ij}, respectively, requested by the ith ST; that is,

$$D_i = \frac{\sum_j min_{ij} + \sum_j max_{ij}}{2}, \ \forall i.$$

As defined, D_i ensures that extreme distances are not chosen. Finally, the weights λ_σ, λ_χ, and λ_τ allow an adjustment of the contribution of each of the three components to the objective function. Their values are chosen such that $\lambda_\sigma + \lambda_\chi + \lambda_\tau = 1$.

In (4) the variables lb_j and ub_j denote the lower bound and upper bound, respectively, on the percentage satisfaction level of each request with priority j. The percentage satisfaction level is the ratio of the total number of slots allocated above the minimum number of requested slots to the difference between the maximum and minimum numbers of requested slots.

The specific problem treated in Huynh and Gillen [2001] involves $|I| = 3$, $|K| = 4$, $|L| = 10$, and $ch_{max} = 2$. Since a maximum bandwidth allocation is considered more important than optimal spacing of assignments, the objective function in (4) is evaluated with $\lambda_\sigma = 0.1$ and $\lambda_\chi = 0.9$. The traffic demand is randomized.

Step 3. Perform the Orthogonal Array Experiment

• **Perform Experiment Design and Planning.** This problem indicates two tiers of assignment: STs to channels and STs to time slots in the assigned channels. Since each channel has 10 time slots and there are 4 channels, the total number of time slots is 40. In this case, a time slot is a factor. Since each time slot can be assigned to only 1 ST, each factor has three levels: 1 for ST 1, 2 for ST 2, and 3 for ST 3.

Table IV. A Comparison of DAMA Assignments by Orthogonal Array Experiment and Optimization [Huynh and Gillen, 2001]

Run	Solution Method	Optimality Departure (%)	DAMA Assignment Results										
1	Orthogonal Array Experiment	0		Slot									
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			Channel **1**	x	3	x	x	3	1	2	x	x	2
			2	2	2	x	2	2	2	x	2	2	0
			3	3	x	x	3	0	3	x	3	3	3
			4	1	x	1	1	1	x	1	1	1	1
	Optimization			Slot									
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			Channel **1**	x	2	x	x	2	2	2	x	x	2
			2	2	1	x	2	1	1	2	2	2	0
			3	1	x	x	3	0	3	x	3	3	3
			4	1	x	3	1	3	x	1	1	1	1
2	Orthogonal Array Experiment	13		Slot									
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			Channel **1**	3	3	x	1	x	1	2	x	x	2
			2	1	x	1	0	x	0	1	1	x	1
			3	0	0	x	3	3	3	3	3	3	3
			4	x	2	x	x	2	2	0	x	x	0
	Optimization			Slot									
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			Channel **1**	1	1	x	1	x	0	0	x	x	1
			2	2	x	2	2	x	1	1	1	x	2
			3	3	2	x	3	2	2	2	3	2	3
			4	x	3	x	x	3	3	3	x	x	0
3	Orthogonal Array Experiment	41		x									
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			Channel **1**	x	3	x	1	x	1	2	3	1	2
			2	x	x	x	x	2	2	3	x	2	1
			3	x	0	x	x	0	0	0	0	x	0
			4	x	x	x	0	x	0	0	x	0	0
	Optimization			Slot									
				1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			Channel **1**	x	1	x	3	x	1	1	2	3	3
			2	x	x	x	x	2	2	2	x	2	2
			3	x	3	x	x	3	3	3	3	x	0
			4	x	x	x	1	x	0	0	x	1	1

An experiment here is neither a computer simulation nor a laboratory experiment with hardware; rather, it is the computation of the objective function associated with a trial allocation of the time slots (and hence channels) to the STs. A full factorial design for this problem would require 3^{40} or 1.2×10^{19} experiments. It would be impractical to run such an exorbitant number of experiments. The use of orthogonal arrays drastically reduces the number of experiments. This problem requires the orthogonal array $L_{81}(3^{40})$ for the 40 factors (i.e., time slots) with 3 levels (again corresponding to the 3 STs). This large orthogonal array can be found in Taguchi and Konishi [1987]. The 40 columns of the array correspond to the 40 slots. The values that go into all the columns correspond to the number of STs, namely, 1, 2, and 3. There are thus at most 81 experiments or conditions to carry out.

- **Run Experiment.** A computer program reads the orthogonal array $L_{81}(3^{40})$ and the input data such as the size and state of the network, the traffic demand, and the operational constraint parameters. The responses (values of the objective function) in (4), calculated for the 81 experiments (trials), are then analyzed by the computer program according to a modified data analysis, which is discussed next.

• **Perform Data Analysis.** Each column corresponds to a particular channel and a particular slot, i.e., j is a function of k and ℓ. Since four channels each have 10 slots, $j = 10(k - 1) = \ell$, with $k = 1, \ldots, 4$ and $\ell = 1, \ldots, 10$. Then the average performance (i.e., the objective function) $\overline{f}_{j\alpha}$ of the jth factor (slot) at the αth level is calculated according to (2). Here $N_{j\alpha}$ is the number of rows in which the jth slot is assigned to the αth ST and N_ε is the number of rows in the orthogonal array, which is 81.

For this problem, the quality characteristic is "the smaller the better" performance (the overall objective function), the levels of the factors are identified according to the minimum average response. In this illustration, the standard data analysis discussed in Section 4.2 is carried out differently. Instead of using $\overline{f}_{ji} = \min_\alpha \overline{f}_{j\alpha}$ to assign the jth slot to the ith ST, an ordered set of the average responses is formed, in which the average responses, $\overline{f}_{j\alpha}$, are ranked from the smallest to the largest value, and assignments are made from the ordered set.

• **Obtain an Optimal Solution.** Since j is a known function of the channel-slot pair (k, ℓ), an assignment of the ith ST to a channel-slot pair (k, ℓ) can be transformed to the usual assignment variable, $X_{ik\ell}$. As proven in Huynh and Gillen [2001], in the absence of constraints, these two ways of implementing the data analysis are equivalent, in the sense that they yield an identical final allocation. This implementation enhances the chance of satisfying the requested minimum time slot constraint as long as feasible solutions exist [Huynh and Gillen, 2001].

Furthermore, the constraints are handled in a specific manner. First of all, the minimum number of slots requested by the STs must be satisfied. Starting with the smallest value of the ordered set, denoted by Φ, its elements are mapped to assignments $X_{ik\ell}$ that satisfy the constraint in (5) that only one ST can be assigned to an available time slot and then the constraint in (5) that the number of channels assigned to an ST cannot exceed ch_{max} until

$$\sum_{k,\ell} X_{ik\ell} = min_i \quad \text{for all } i \in I.$$

Clearly, the elements of Φ that are successfully mapped to the total of the minimum numbers of requested slots need not be consecutive. Let $\Phi' = \Phi - \Phi_{min}$, where Φ_{min} is the set of all the average responses that correspond to the al1ocation of the minimum numbers of slots requested by the STs. If no such Φ_{min} exists, then the orthogonal array experiment provides no feasible solution to the DAMA assignment problem.

The set Φ' remains an ordered set. Again, starting with the smallest value of the ordered set Φ', its elements are mapped to the assignments $X_{ik\ell}$ that satisfy the constraints that only one ST can be assigned to an available time slot and then the constraint that the number of channels assigned to an ST cannot exceed ch_{max}, respectively, until

$$\sum_{k,\ell} X_{ik\ell} \le max_i \text{ for all } i \in I.$$

The final DAMA assignment is thus obtained, for which the objective function is then calculated.

Table IV shows a comparison of the Monte Carlo results produced by the orthogonal array experiment and those obtained with the commercial optimization tool AMPL and CPLEX [Fourer et al., 1993]. It also shows the DAMA assignments and the extent of departure from optimality. The slots in Table IV marked with "x" are unavailable at the time of assignment, while those with 0 remain unassigned. The slots marked with 1, 2, and 3 are assigned to ST 1, 2, 3, respectively. As shown, Run 1 corresponds to no departure (best case) from optimality, Run 2 to 13% departure (average case), and Run 3 to 41% (the worst case).

Based on the Monte Carlo results, the way the constraints are handled thus appear to be effective; the orthogonal array experiment approach appears to be an appropriate solution technique for multidimensional assignment problems of this kind. Additionally, the short runtime of the orthogonal array experiment makes it suitable for real-time bandwidth allocation in a satellite communications network.

Employed as a validation and benchmarking tool when developing real-time heuristic DAMA algorithms, this approach offers these benefits: ad hoc assessment of the performance of a heuristic algorithm is avoided and, hence, confidence in the algorithm is increased; and a quantitative conclusion can be made about the sufficiency of the performance of a heuristic algorithm. The latter benefit aids in determining when further development or improvement of an algorithm is no longer necessary.

6. CONCLUSION

This paper discusses the applicability of orthogonal array experiment to solving some assignment problems encountered in systems engineering and architecting. To solve the nonlinear engineering optimization problems which can be cast in nonlinear integer programming problems, the orthogonal array experiment approach, as an advantage, does not call for linearization of the nonlinear optimization problems. It solves them directly, by carrying out, through orthogonal arrays, the smallest possible number of experiments and determining the solution from the responses of the experiments. An engineering optimization problem that involves a large number of factors and levels will require a large orthogonal array, which can be generated or obtained from ASI [1987].

This paper also illustrates the mechanism of applying the orthogonal array experiment approach to the problem of DAMA algorithm performance verification and validation [Huynh and Gillen, 2001] and the problem of architecting an SoS responding to small boat attacks by terrorists [Huynh et al., 2007]. The orthogonal array experiment approach has been found to be effective and efficient for these problems. The feasibility of applying orthogonal array experiment to these problems suggests its potential application to other optimization problems encountered in systems engineering and architecting.

The area of application of orthogonal array experiment to solve assignment problems (or mathematical programming problems) is still evolving. Research in this area needs to look into a unified treatment of constraints in mathematical programming problems and methods to handle continuous decision variables.

7. ACRONYMS

ATM Asynchronous transfer mode
C2 Command and Control
C4ISR Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CDV Cell Delay Variation
DAMA Demand-Assignment Multiple-Access
HVU High-Value Unit
MTR Maritime Threat Response
NC Network Controller
NPS Naval Postgraduate School
O&S Operating and Support
PBS Prepare Battlespace
PC Patrol Coastal
Ps Probability of success
QoS Quality of Service
SBA Small Boat Attack
SEA Systems Engineering and Analysis
S/N Signal-to-Noise ratio
SoS System of Systems
ST Subscriber terminal
U.S. United States
USCG United States Coast Guard
USV Unmanned Surface Vehicle

ACKNOWLEDGMENT

The author thanks three reviewers for their constructive and helpful comments.

REFERENCES

ASI (American Supplier Institute), Dearborn, MI, 1987.

J. Attagara, The explosive scanning devices allocation problem for airport security systems, Doctoral dissertation, Texas Tech University, Lubbock, April 26, 2006, http://hdl.handle.net/2346/997.

A. Bendell, J. Disney, and W. A. Pridmore (Editors), Taguchi methods, applications in world industry, IFS Publications/Springer-Verlag, New York, 1989.

S.K. Biswas and R. Izmailov, Design of a fair bandwidth allocation policy for VBR traffic in ATM networks, IEEE/ACM Trans Networking 8(2) (April 2000), 212–223.

R. Bose and B. Mandel, Introduction to combinatorial theory, John Wiley and Sons, 1984.

G.E.P. Box, W.G. Hunter, and J.S. Hunter, Statistics for experimenters: An introduction to design, data analysis and model building, Wiley, New York, 1978.

G.E.P. Box, J.S. Hunter, and W.G. Hunter, Statistics for experimenters: Design, innovation, and discovery, 2nd edition, Wiley, Hoboken, NJ, 2005.

D.M. Byrne and G. Taguchi, The Taguchi approach to parameter design, ASQC Trans Annu Qual Congr, Anaheim, CA, 1986.

D.P. Clausing, Taguchi methods integrated into the improved total development, Proc IEEE Int Conf Commun, Philadelphia, PA, 1988, pp. 826–832.

W.G. Cochran and G.M Cox, Experimental design, Wiley, New York, 1957.

W.W. Chu, Optimal file allocation in a multiple computer system, IEEE Trans Comput 18(10) (October 1969), 885–889.

C. Daniel, Applications of statistics to industrial experimentation, Wiley, New York, 1976.

L. Euler, Recherches sur une espèce de carrés magiques, Commentationes Arithmetae Collectae II (1849), 302–361.

R.A. Fisher, Design of experiments, Oliver and Boyd, Edinburg, UK, 1951.

R. Fourer, D.M. Gay, and B.W. Kernighan, AMFL: A modeling language for mathematical programming, Duxbury, North Scituate, MA, 1993.

S. Gao, Z. Zhang, X. Zhang, and C. Cao, Immune genetic algorithm for weapon-target assignment problem, Workshop on Intell Inform Technol Appl, 2007, pp. 145–148.

D. Grundel, C.A. Oliveira, P.M. Pardalos, and E. Pasiliao, Asymptotic results for random multidimensional assignment problems, Comput Optim Appl 31(3) (July 2005), 275–293.

A.S. Hedayat, N.J.A. Sloane, and J. Stufken, Orthogonal arrays: Theory and applications, Springer Series in Statistics, Springer, New York, 1999.

C.R. Hicks, Fundamental concepts in the design of experiments, Holt, Rinehart, and Winston, New York, 1973.

K.S. Holness, C.G. Druny, and R. Batta, A systems view of personnel assignment problems, Human Factors and Ergonomics in Manufacturing 16(3) (2006), 285–307.

T.V. Huynh, Optimal file allocation in a distributed computer network by orthogonal array experiments, IEEE Aerospace Appl Conf Proc, 1997, Vol. 4, pp. 105–114.

T.V. Huynh and D.C. Gillen, Dynamic bandwidth allocation in a satellite communication network, IEEE Aerospace Appl Conf Proc, 2001, Vol. 3, pp. 1221–1232.

T.V. Huynh and J.J. Kohfeld, Optimal functional allocation in a tactical BM/C3I system, 1994 Symp Command Control Res Decision Aids, Monterey, CA, June 21–23, 1994, pp. 386–391.

T.V. Huynh and X.L. Tran, Optimal allocation of support to small manufacturing enterprises in defense projects, Proc Third Int Conf Syst Syst Eng, Monterey, CA, June 2–4, 2008.

T.V. Huynh, A. Kessler, J. Oravec, S. Wark, and J. Davis, Orthogonal array experiment for architecting a system of systems responding to small boat attacks, Syst Eng 10(3) (2007), 241–259.

T. Huynh, B. Connett, J. Chiu-rourman, J. Davis, A. Kessler, J. Oravec, M. Schewfelt, and S. Wark, Architecting a system of systems responding to maritime domain terrorism by orthogonal array experiment, Nav Eng J 121(1) (2009), 79–101.

A. Jeang and C.-L. Chang, Combined robust parameter and tolerance design using orthogonal arrays, Int J Adv Manuf Technol 19(6) (2002), 442–447.

P.W.M. John, Statistical design and analysis of experiments, Macmillan, New York, 1971.

R.N. Kackar, Off-line quality control, parameter design and the Taguchi method, J Qual Technol 17(4) (1985), 176–209.

R.N. Kackar, Taguchi's quality philosophy: Analysis and commentary, Quality Progress, December 21–29, 1986, pp. 23–24.

A. Kessler, M. Schewfelt, B. Connett, C. Chiurourman, J. Oravec, S. Wark, J. Davis, Ling Siew Ng, Seng Chuan Lim, Cheng Lock

Chua, Eng Choon Yeo, Kok Long Lee, Heng Hui Chew, Kwang Yong Lim, Ee Shen Tean, Sze Tek Ho, and Koh Choon Chung, Maritime threat response, Report NPS-97-06-004, National Park Service, Washington, DC, June 2006.

D. McDysan and D. Spohn, ATM theory and applications, McGraw-Hill, New York, 1999.

M. Minoux, Mathematical programming: Theory and algorithms, Wiley, New York, 1986.

T.W. Pao, M.S. Phadke, and C.S. Sherrerd, "Computer response time optimization using orthogonal array experiments," IEEE International Communications Conference, Chicago, IL, Vol. 2, pp. 890–895, 1985.

G.S. Peace, Taguchi methods, Addison Wesley, 1992.

E.S. Pearson, Memories of the impact of Fisher's work in the 1920s, Int Statist Rev/Rev Int Statist 42(1) (1974), 5–8.

C.L. Pettit and D.E. Veley, Risk allocation issues for systems engineering of airframes, 4th Int Symp Uncertainty Model Anal, 2003, p. 232.

M.S. Phadke, Quality engineering using robust design, Prentice Hall, Englewood Cliffs, NJ, 1989.

M.S. Phadke, http://www.isixsigma.com/library/content/c030106a.asp, accessed October 24, 2009.

R.K. Roy, A primer on the Taguchi method, Van Nostrand Reinhold, New York, 1990.

A.P. Sage, Systems engineering, Wiley, Hoboken, NJ, 1992.

G. Taguchi, Off-line and on-line quality control systems, Int Conf Qual Control, Tokyo, Japan, 1978.

G. Taguchi, Introduction to quality engineering, Asian Productivity Organization, Tokyo, 1986.

G. Taguchi, The system of experimental design: Engineering methods to optimize quality and minimize costs, American Supplier Institute, Livonia, MI, 1987, Vols. I and II.

G. Taguchi, Taguchi on robust technology development: Bringing quality engineering upstream, ASME, New York, 1993.

G. Taguchi and S. Konishi, Orthogonal arrays and linear graphs—tools for quality engineering, ASI Press, Trevose, PA, 1987.

G. Taguchi and M.S. Phadke, Quality engineering through design optimization, Conf Record, GLOBECOM 84 Meeting, IEEE Communications Society, Atlanta, GA, 1984, pp. 1106–1113.

G. Taguchi and Y.I. Wu, Introduction to off-line quality control, Central Japan Control Association, Meieki Nakamura Ku Nagaya, Japan, 1979.

G. Taguchi, Y. Wu, and S. Chowdhury, Taguchi's quality engineering handbook, Wiley, New York, 2004.

R. Unal, D.O. Stanley, and C.R. Joyner, Propulsion system design optimization using the Taguchi method, IEEE Trans Eng Management 40(3) (1993), 315–322.

J.M. Vidal, A method for solving distributed service allocation problems, Web Intell Agent Syst 1(2) (2003), 139–146.

M.I. Vidalis, C.T. Papadopoulos, and C. Heavey, On the workload and "phaseload" allocation problems of short reliable production lines with finite buffers, Comput Indust Eng 48(4) (June 2005), 825–837.

W.-C. Weng, Y. Fan, V. Demir, and A. Elsherbeni, Electromagnetic optimization using Taguchi method: a case study of linear antenna array design, IEEE Antennas Propagation Soc Int Symp 9(14) (2006), 2063–2066.

Tom Huynh is an associate professor of systems engineering at the Naval Postgraduate School in Monterey, CA. His research interests include complex systems and complexity theory, system-of-systems engineering methodology, and systems architecting. Prior to joining the Naval Postgraduate School, Dr. Huynh was a Fellow at the Lockheed Martin Advanced Technology Center, Palo Alto, CA, where he engaged in research in computer network performance, computer timing control, bandwidth allocation, heuristic algorithms, nonlinear estimation, perturbation theory, differential equations, and optimization. He was also a lecturer in the departments of Physics and Mathematics at San Jose State University. Dr. Huynh obtained simultaneously a B.S. (Hons) in Chemical Engineering and a B.A. in Applied Mathematics from UC Berkeley and an M.S. and a Ph.D. in Physics from UCLA. He is a member of INCOSE.