Entropy **2007**, *9*, 198-232

entropy
ISSN 1099-4300

www.mdpi.org/entropy

Full Paper

Design of Experiments: Useful Orthogonal Arrays for Number of Experiments from 4 to 16

Sorana D. Bolboacă [1,]* and Lorentz Jäntschi [2]

1 Iuliu Hatieganu University of Medicine and Pharmacy Cluj-Napoca, 6 Louis Pasteur, 400349 Cluj-Napoca, Romania; E-mail: sbolboaca@umfcluj.ro
2 Technical University of Cluj-Napoca, 103-105 Muncii Bvd., 400641 Cluj-Napoca, Romania; E-mail: lori@academicdirect.org

* Author to whom correspondence should be addressed. E-mail: sbolboaca@umfcluj.ro

Received: 22 August 2007 / Accepted: 2 November 2007 / Published: 31 December 2007

Abstract: A methodology for the design of an experiment is proposed in order to find as many schemes as possible with the maximum number of factors with different levels for the smallest number of experimental runs. An algorithm was developed and homemade software was implemented. The abilities in generation of the largest groups of orthogonal arrays were analyzed for experimental runs of 4, 6, 8, 9, 10, 12, 14, 15, and 16. The results show that the proposed method permits the construction of the largest groups of orthogonal arrays with the maximum number of factors.

Keywords: Design of experiment, orthogonal arrays, Taguchi method.

AMS Classification: 05B15 Orthogonal arrays, Latin squares, Room squares, 94C12 Fault detection; testing, 11Y35 Analytic computations, 11Y40 Algebraic number theory computations, 11Y55 Calculation of integer sequences, 65F25 Orthogonalization, 62Kxx Design of experiments

---

Introduction

Manufacturing process optimizations are powerful methods that provide simulation scenarios that yield the desired outcome [1]. The optimization techniques could contain metaheuristic procedures and/or classical optimization methods [1,2] that involve setting a series of parameters in order to obtain:

- Maximum return on budgets
- Most effective configuration of machines
- Most effective allocation of raw materials
- Optimal workforce allocations to minimize labor and total time

Although the design of experiments concept was introduced by Fisher in the early 1920s [3], the most research on this topic was carried out in the academic environment [4]. One year later, Fisher [5] demonstrated the usefulness of his concept in agricultural experiments; he analyzed the optimum water, rain, sunshine, fertilizer, and soil conditions needed to produce the best crop. Taguchi [6] went further with the design of experiment concept by introducing his approach in 1986. According to the nature of the problem, the Taguchi approach divides optimization problems in two categories, using a log function of desired output as objective functions for optimization (called Signal-to-Noise ratios):

- Static problems (there are several control factors that directly decide the desired value of the output):
    - Smaller-the-Better approach is used when:
        - The ideal value for all undesirable characteristics is zero
        - The ideal value is finite and its maximum or minimum value is defined
    - Larger-the-Better
    - Nominal-the-Best approach is used when a specified value is most desired and neither a smaller nor a larger value is desirable.
- Dynamic problems (there is a signal input that directly decides the output):
    - Sensitivity of the slope: the slope should be at the specified value (usually 1) when the output is:
        - An undesired characteristic (it can be treated as Smaller-the-Better)
        - A desirable characteristic (it can be treated as Larger-the-Better)
    - Linearity (Larger-the-Better): is used when the dynamic characteristics are required to have direct proportionality between the input and output.

A triad could better characterize the aim of manufacturing process optimization: best quality – less failures – higher productivity. Factorial analysis can be used in order to find the best values for parameters implied in the manufacturing process [7]. Opposite to full factorial analysis, the Taguchi method reduces the number of experimental runs to a reasonable one, in terms of cost and time, by using orthogonal arrays [8].

The Taguchi method is used whenever the settings of interest parameters are necessary, not only for manufacturing processes. Therefore, the Taguchi approach is used in many domains such as: environmental sciences [9,10], agricultural sciences [11], physics [12], chemistry [13], statistics [14], management and business [15], medicine [16].

Choosing the proper orthogonal arrays suitable for the problem of interest is the main difficulty of the Taguchi's approach. The available literature identified the use of the orthogonal arrays summarized in Table 1.

The literature reported many orthogonal arrays; however, a full scheme that includes all the possibilities of orthogonal arrays, even for a small number of experimental runs, could not be found yet [17]. Starting from this observation the aim of the present study was to generate the largest groups of orthogonal arrays for number of experimental runs from four to sixteen, with the maximum number of factors by using a series of homemade software.

Table 1. Design of experiments: reported scheme

Experimental runs	Scheme	Reference
4	2^3	[18]
8	2^7	[19-23]
	2^5	[24]
	2^3	[25-27]
9	3^4	[28-33]
	3^3	[34]
16	4^5	[35-37]
	2^{15}	[38]
18	$2^1\times3^7$	[39-41]
	$2^2\times3^6$	[42]
	3^7	[43]
	$2^1\times3^6$	[44]
	$2^1\times3^4$	[45,46]
	$2^1\times3^3$	[47]
27	3^3	[48]

Method

Note that searching for as many as possible orthogonal arrays with the highest number of factors possible for the smallest number of experimental runs is not a trivial task. Table 2 presents an example of 12 experimental runs, in which adding of a new orthogonal array (D) to the existent ones (A, B, and C) is not possible, although the maximum number of factors with two levels for which the orthogonal arrays could be obtained is equal with eleven. The information regarding the maximum number of factors with two levels can be checked with Statistica software, Experimental design - Taguchi robust design experiments [49].

Table 2. Design of experiment: 12 experimental runs

Experimental runs	Factor(levels)			
	A(2)	B(2)	C(2)	D(2)
1	0	0	0	-
2	0	0	0	-
3	0	0	0	-
4	0	1	1	-
5	0	1	1	-
6	0	1	1	-
7	1	0	1	-
8	1	0	1	-
9	1	0	1	-
10	1	1	0	-
11	1	1	0	-
12	1	1	0	-

As already mentioned, the objective of the research was to obtain the largest set of orthogonal arrays by using a series of homemade software. Note that there was no rule about generating these sets, as proven by the previous example from Table 2.

An application was designed for generating orthogonal arrays from a list of factor levels (`level` array) using a recursive function (`recurs`), a class (`liste`) for storing current array (`lst` instance of `liste`), and an initialization function (`fill_first_oa`):

```
class liste{
      function __construct(b,n){
            this->b=b;//b: base of numeration
            this->n=n;//n: number of experiments (n%b=0)
            this->m=n/b; //number of repetitions
            for(s=0,i=0;i<b;i++){
                  this->d[i]=this->m;//clusters initialization
                  s+=i*this->m;
            }
            this->s=s;//sum of elements
            for(i=0;i<n;i++){
                  this->v[i]=0;//elements initialization
            }
      }
}
function recurs(&ar,&o_a_a,$it,&is_OA){
      if(it>=ar->n) return;//nothing to recourse
      if(check_empty(ar)){
            if(check_orto(ar,o_a_a)){
                  is_OA=TRUE;
                  return;
            }//ar is OA with o_a_a
      }else{
            for(i=1;i<ar->b;i++){//0 is the default
                  $ar->v[it]=i;//try with i
                  $ar->d[i]--;
                  recurs(ar,o_a_a,it+1,is_OA);
                  if(is_OA) return;
                  ar->d[i]++;//try with 0
                  ar->v[it]=0;
                  recurs(ar,o_a_a,it+1,is_OA);
                  if(is_OA) return;
            }
      }
}
function fill_first_oa(lvn,expn,&o_a_a_var){
      for(k=0,i=0;i<expn;i++){
            o_a_a_var[0][i]=k++;
            k%=lvn;
      }//fill like 0,1,2,0,1,2 (lvn=3)
}
//main program for Orthogonal Arrays (OA)
      ...
      fill_first(levels[0],expn,o_a_a);// first OA
      for(i=1;i<n;i++){//n: number of planned OAs
            ...
            lst = new liste(levels[i],expn);
            ...
            recurs(lst,o_a_a,0,stop);//stop: no more OAs;
            ...
            //display intermediary OAs (o_a_a)
            ...
      }
```

Another application was designed for generating the orthogonal arrays having the same number of levels (`levels`) for a given number of experimental runs (`expn`) starting with a list of already found orthogonal arrays (`orto_list`, which is an array of arrays), using a recursive function (`rec`), and an orthogonal testing function (`ort`, which also add the new OA to the list on successful):

```
function rec(&a,b,na,va,nb,vb,pa,pb,bufb,nbufb,
                &orto_list,&norto_list){//output data
    if(va==0){//all combinations were exhausted
        for(i=0;i<na;i++){
            c[bufb[i]]=(i-i%nb)/nb;
        }//it's time to check our OA
        nc=(na/nb-1)*na/2;
        ort(c,na,nc,orto_list,norto_list);
    }else //do recursion for all remained combinations
        if(vb==0){
        j=0;
        for(i=0;i<va;i++){
            if(a[i]==b[j]){
                bufb[]=b[$j];
                nbufb++;
                if(j<nb-1)j++;
            }else{
                newa[]=a[i];
            }
        }
        rec(newa,array(),na,va-nb,nb,nb,0,0,bufb,nbufb,
                    orto_list,norto_list);
    }else{
        for(i=pa;i<va-vb+1;i++){
            b[pb]=a[i];
            rec1(a,b,na,va,nb,vb-1,i+1,pb+1,bufb,nbufb,
                    orto_list,norto_list);
        }
    }
}
```

The main program calls the `rec` function after initializing the identical permutation (`a[i]=i`). Then, the results are displayed (`orto_list`):

```
rec(a,array(),expn,expn,expn/levels,expn/levels,0,0,array(),0,
                orto_list,norto_list);
```

The `rec` function generates all possible distinct permutations of, for example, a set like:

`{0,0,0,1,1,1,2,2,2}`

by taking into account that the changing of a pair of positions (from 0 to 9 in our case) is relevant only for the positions that point to different values.

Thus, the complexity of the problem solved by `rec` function is (`b` = base, 3 in above example, `n` = number of experimental runs, 9 in the example above, which give a total number of combinations equal with 1680):

`(n,b)·(n-b,b)·(n-2b,b)·…`

Results and Discussion

The programs presented in the previous section were run in order to reach the aim of the research for the following number of experiments: 4, 6, 8, 9, 10, 12, 14, 15, and 16. The largest groups of orthogonal arrays were generated. The results according to the number of factors and associated levels by the number of experimental runs were summarized and presented in Table 3. Note that only the maximum number of factors with associated levels according to the number of experimental runs was reported. The schemes that were included into the reported ones were not displayed; for example, for L_9 only the 3^5 scheme was reported and not the 3^4 scheme, as this was obviously included into the 3^5 scheme. As presented in Table 3, a total number of sixty-three schemes were identified: 2 for L_4, 3 for L_6, 12 for L_8, 4 for L_9, 2 for L_{10}, 18 for L_{12}, 2 for L_{14}, 9 for L_{15}, and 11 for L_{16}.

Note that the following are true (see Table 3):

- Any level of any factor is a number that divides the number of experimental runs. This is the explanation for missing the orthogonal arrays for L_5, L_7, L_{11}, and L_{13} (the number of the experimental runs could be divided just by themselves);
- In every experimental runs, in at least one case, the highest level of factor is equal with the number of experimental runs;

For the number of experimental runs equal with 4, 8, 12, and 16, the highest number of factors is given by the expression: $n_F = n_E - 1$ (where n_F = number of factors, n_E = number of experimental runs), as the Hadamard matrix shown [50].

The comparison of the resulted schemes (Table 3) with the table of orthogonal arrays maintained by Sloane [51] (who identified one scheme for L_4, L_6, L_9, L_{10}, L_{14}, and L_{15}, 2 schemes for L_8, four schemes for L_{12}, and seven schemes for L_{16}) indicates that the number is greater and the contribution to the orthogonal arrays database is significant. Furthermore, there are not any schemes reported by Sloane [51] that cannot be identified by using the implemented software (see the Material section).

The analysis of the available literature and software suggests that orthogonal arrays that are reported for the first time were identified.

Table 3. Number of experimental runs (n_{exp}), maximum number of factors (n_f), levels (L_i) and maximum number of factors for specified levels ($\sum MF_i$)

n_{exp}	n_f	L_1	$\sum MF_1$	L_2	$\sum MF_2$	L_3	$\sum MF_3$
4	3	4	2	2	1		
		2	3				
6	3	6	1	3	2		
		3	3				
		2	1	3	2		
8	7	4	4	2	3		
		4	2	2	5		
		2	7				
	6	8	3	4	2	2	1
		8	1	4	3	2	2
		8	1	4	1	2	4
		4	6				
	5	8	3	2	2		
		8	2	2	3		
		8	1	4	3	2	1
		8	1	2	4		
	4	8	4				
9	5	3	5				
	4	9	4				
		9	2	3	2		
		9	1	3	3		
10	6	10	1	5	5		
	3	5	2	2	1		
12	11	2	11				
	10	4	1	2	9		
	9	4	2	2	7		
	7	4	4	2	3		
		4	3	2	4		
		3	7				
	6	4	6				
		4	3	3	1	2	2
		4	2	3	1	2	3
		3	4	2	2		
		3	3	2	3		
12	5	12	1	2	4		
		6	5				
		4	2	3	2	2	1
		4	1	3	2	2	2
		3	1	2	4		
	4	12	4				
	3	12	1	6	2		
14	6	7	5	2	1		
	5	14	1	7	4		
15	8	5	4	3	4		
	7	5	5	3	2		
		15	1	3	6		
		5	2	3	5		
		5	1	3	6		
		3	7				
	6	15	1	5	5		
		5	6				
		5	3	3	3		
16	15	2	15				
	14	4	1	2	13		
	13	8	1	2	12		
		4	2	2	11		
	12	16	1	2	11		
	10	16	1	8	1	2	8
	9	4	9				
		4	3	2	6		
	7	8	7				
	5	16	5				
		16	2	2	3		

n_{exp} = number of experimental runs
n_f = total number of factors
L_i = levels for associated number of factors *i*
$\sum MF_i$ = total number of maximum factors *i*

One observation that resulted from the investigated cases refers to the modality of constructing orthogonal arrays. The orthogonal arrays could be classified as fixed-level (all factors have the same number of levels) and mixed-level (the factors have different levels). The linear programming can be used for fixed-level orthogonal arrays construction [52]. The mixed-level orthogonal arrays can be constructed using the expansive replacement method [17] or the "mixed spreads" approach [53]. The analysis of the orthogonal arrays obtained by using the developed programs revealed that a new factor could be constructed by the linear combination of two existing factors. Thus, a factor that is independent from all the other orthogonal arrays factors results ($(10)_x \cdot A + B$, $(100)_x \cdot A + (10)_x \cdot B + C$, where x = number of levels, $A, B, and\ C$ = elements of vector as modulo x values (from zero to $n_E - 1$, n_E = number of experimental runs)).

Let us take for example the L_8 (2^7) orthogonal array (see Table 4).

Table 4. L_8 (2^7) orthogonal array

Experimental run	Factor(Levels)						
	A(2)	B(2)	C(2)	D(2)	E(2)	F(2)	G(2)
1	0	1	1	1	0	0	0
2	1	1	1	0	1	1	0
3	0	1	0	0	1	0	1
4	1	1	0	1	0	1	1
5	0	0	1	0	0	1	1
6	1	0	1	1	1	0	1
7	0	0	0	1	1	1	0
8	1	0	0	0	0	0	0

The L_8 ($4^1 \times 2^5$) orthogonal array is obtained (see Table 5) by the linear combination of the two-level factors A and B (Table 4).

Table 5. L_8 ($4^1 \times 2^5$) orthogonal array

Experimental run	Factor(Levels)					
	Y(2)	C(2)	D(2)	E(2)	F(2)	G(2)
1	1	1	1	0	0	0
2	3	1	0	1	1	0
3	1	0	0	1	0	1
4	3	0	1	0	1	1
5	0	1	0	0	1	1
6	2	1	1	1	0	1
7	0	0	1	1	1	0
8	2	0	0	0	0	0

The L_8 ($4^2 \times 2^3$) orthogonal array presented in Table 6 is obtained by the linear combination of C and D factors (Table 4).

Table 6. L_8 ($4^2\times2^3$) orthogonal array

Experimental run	Factor(Level)				
	Y(4)	Z(4)	E(4)	F(4)	G(4)
1	1	3	0	0	0
2	3	2	1	1	0
3	1	0	1	0	1
4	3	1	0	1	1
5	0	2	0	1	1
6	2	3	1	0	1
7	0	1	1	1	0
8	2	0	0	0	0

The L_8 ($8^1\times2^4$) orthogonal array could also been obtained (see Table 7) by the linear combination of A, B and C factors (see Table 4).

Table 7. L_8 ($8^1\times2^4$) orthogonal array

Experimental run	Factor(Level)				
	Q(8)	D(2)	E(2)	F(2)	G(2)
1	3	1	0	0	0
2	7	0	1	1	0
3	2	0	1	0	1
4	6	1	0	1	1
5	1	0	0	1	1
6	5	1	1	0	1
7	0	1	1	1	0
8	4	0	0	0	0

Although some new factors with associated levels could be obtained, the linear combination of factors could not retrieve the maximum numbers of possible combinations. This can be observed by looking at the obtained results presented in Table 3. For example, the proposed method identified a number of twelve schemes as the largest groups of orthogonal arrays for L_8:

- Seven factors: $4^4\times2^3$, $4^2\times2^5$, 2^7
- Six factors: $8^3\times4^2\times2^1$, $8^1\times4^3\times2^2$, $8^1\times4^1\times2^4$, 4^6
- Five factors: $8^3\times2^2$, $8^2\times2^3$, $8^1\times4^3\times2$, $8^1\times2^4$
- Four factors: 8^4

The aim of the research was reached: the largest groups of orthogonal arrays for the studied number of experimental runs were identified. It is already known that the Taguchi approach can satisfy the needs of optimum process design and can reduce manufacturing costs [4]. The orthogonal arrays used by the Taguchi approach allow the study of the simultaneous effect of several factors efficiently, providing better results in smaller number of experimental runs [1,4]. The orthogonal arrays resulted from the present research have an advantage as compared with known orthogonal arrays: they allow the investigation of a greater number of factors with different levels. Having a clear list of all the largest sets of orthogonal arrays possible for a given number of experimental runs, the design of experiments could be improved and simplified. A more useful optimization of manufacturing design could be obtained by using the greatest number of factors' levels and the smallest number of experimental runs.

The generation of the largest groups of orthogonal arrays could be further developed. The proposed algorithm could be applied for other desired numbers of experiments. Note that, as the number of experimental runs increases, the time needed for generating the maximum numbers of orthogonal arrays with the highest levels increases too.

Acknowledgements

The research was partly supported by UEFISCSU Romania through projects ID458/2007 and ID1051/2007.

References

1. Ross, P.J. *Taguchi Techniques for Quality Engineering*; McGraw-Hill: New York, 1998.
2. Czyzak, P.; Jaszkiewicz, A. Pareto simulated annealing – A metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis* **1998**, *7(1)*, 34-47.
3. Fisher, R.A. *Statistical Methods for researcher Workers*; Oliver and Boyd: London, UK, 1925.
4. Ranjit, K.R. *Design of Experiments Using the Taguchi Approach: 16 Steps to Product and Process Improvement*; John Wiley & Sons: Hoboken, NJ, 2001.
5. Fisher, R.A. The arrangement of field experiments. *Jour. Min. Agr. Engl*. **1926**, *33*, 503-513.
6. Taguchi, G. *Introduction to Quality Engineering: Designing Quality into Products and Processes*; Asian Productivity Organization/UNIPUB, White Plain, Unipub/Kraus, NY, 1986.
7. Montgomery, D.C.; Runger, G.C. *Applied Statistics and Probability for Engineers*; John Wiley & Sons: Hoboken, NJ, 2006.
8. Taguchi, G.; Jugulum, R.; Taguchi, S. *Computer-based Robust Engineering: Essentials for DFSS*; ASQ Quality Press: Milwoukee, WI, 2004.
9. Daneshvar, N.; Khataee, A.R.; Rasoulifard, M.H.; Pourhassan, M. Biodegradation of dye solution containing Malachite Green: Optimization of effective parameters using Taguchi method. *Journal of Hazardous Materials* **2007**, *143(1-2)*, 214-219.
10. du Plessis, B.J.; de Villiers, G.H. The application of the Taguchi method in the evaluation of mechanical flotation in waste activated sludge thickening. *Resources, Conservation and Recycling* **2007**, *50(2)*, 202-210.
11. Tasirin, S.M.; Kamarudin, S.K.; Ghani, J.A.; Lee, K.F. Optimization of drying parameters of bird's eye chilli in a fluidized bed dryer. *Journal of Food Engineering* **2007**, *80(2)*, 695-700.
12. Wu, C.-H.; Chen, W.-S. Injection molding and injection compression molding of three-beam grating of DVD pickup lens. *Sensors and Actuators A: Physical* **2006**, *125(2)*, 367-375.
13. Houng, J.-Y.; Liao, J.-H.; Wu, J.-Y.; Shen, S.-C.; Hsu, H.-F. Enhancement of asymmetric bioreduction of ethyl 4-chloro acetoacetate by the design of composition of culture medium and reaction conditions. *Process Biochemistry* **2007**, *42(1)*, 1-7.
14. Romero-Villafranca, R.; Zúnica, L.; Romero-Zúnica, R. Ds-optimal experimental plans for robust parameter design. *Journal of Statistical Planning and Inference* **2007**, *137(4)*, 1488-1495.

15. Elshennawy, A.K. Quality in the new age and the body of knowledge for quality engineers. *Total Quality Management and Business Excellence* **2004**, *15(5-6)*, 603-614.
16. Ng, E.Y.K.; Ng, W.K. Parametric study of the biopotential equation for breast tumour identification using ANOVA and Taguchi method. *Medical and Biological Engineering and Computing* **2006**, *44(1-2)*, 131-139.
17. Hedayat, A.S.; Sloane, N.J.A.; Stufken, J. *Orthogonal Arrays: Theory and Applications*; Springer-Verlag: New York, 1999.
18. Fontani, S.; Niccolai, A.; Kapat, A.; Oliveri, R. Studies on the Maximization of Recombinant Helicobacter Pylori Neutrophil-Activating Protein Production in Escherichia Coli: Application of Taguchi Robust design and Response Surface Merhodology for Process Optimization. *World Journal of Microbiology & Biotechnology* **2003**, *19*, 711-717.
19. del Alamo, J.; Fernandez, J.C.; Hernandez, M.; Nunez, Y.; Irusta, R.; Del Valle, J.L. Environmental optimisation of a hydro-moulding process. *Journal of Cleaner Production* **2004**, *12*, 153-157.
20. Kim, S.-T.; Park, M.-S.; Kim, H.-M. Systematic approach for the evaluation of the optimal fabrication conditions of a H_2S gas sensor with Taguchi method. *Sensors and Actuators B* **2004**, *102(2)*, 253-260.
21. Kim, K.D.; Choi, K.Y.; Kim, H.T. Experimental optimization of the formation of silver dendritic particles by electrochemical technique. *Scripta Materialia* **2005**, *53*, 571-575.
22. Lin, T.-S.; Wu, C.-F.; Hsieh, C.-T. Enhancement of water-repellent performance on functional coating by using the Taguchi method. *Surface & Coatings Technology* **2006**, *200(18-19)*, 5253-5258.
23. Trabelsi, K.; Rourou, S.; Loukil, H.; Majoul, S.; Kallel, H. Optimization of virus yield as a strategy to improve rabies vaccine production by Vero cells in a bioreactor. *Journal of Biotechnology* **2006**, *121*, 261-271.
24. Lee, H.-C.; Park, O.O. Round pinholes in indium-tin-oxide thin films on the glass substrates: a Taguchi method analysis and theoretical approach to their origins. *Vacuum* **2004**, *72(4),* 411-418.
25. Poon, G.K.K.; Williams, D.J.; Chin, K.S. Optimising the Lithographic Patterning Effect in an Acid Copper Electroplating Process. *The International Journal of Advanced Manufacturing Technology* **2000**, *16*, 881-888.
26. Rocak, D.; Kosec, M.; Degen, A. Ceramic suspension optimization using factorial design of experiments. *Journal of the European Ceramic Society* **2002**, *22*, 391-395.
27. Reddy, T.A.J.; Kumar, Y.R.; Rao C.S.P. Determination of Optimum Process Parameters using Taguchi's Approach to Improve the Quality of SLS Parts. In *Proceeding of the 17^{th} IASTED International Conference Modeling and Simulation,* Montreal, Canada, 2006; pp. 228-233.
28. Lin, T.-S. The Use of Reliability in the Taguchi Method for the Optimisation of the Polishing Ceramic Gauge Block. *International Journal of Advanced Manufacturing Technology* **2003**, *22*, 237-242.
29. Shaji, S.; Radhakrishnan, V. Analysis of process parameters in surface grinding with graphite as lubricant based on the Taguchi method. *Journal of Materials Processing Technology* **2003**, *141(1)*, 51-59.

30. Cho, M.H.; Bahadur, S.; Pogosian, A.K. Friction and wear studies using Taguchi method on polyphenylene sulfide filled with a complex mixture of MoS_2, Al_2O_3, and other compounds. *Wear* **2005**, *258*, 1825-1835.
31. Chua, B.W.; Lu, L.; Lai, M.O.; Wong, G.H.L. Investigation of complex additives on the microstructure and properties of low-temperature sintered PZT using the Taguchi method. *Journal of Alloys and Compounds* **2005**, *386*, 303-310.
32. Fung, C.-P.; Kang, P.-C. Multi-response optimization in friction properties of PBT composites using Taguchi method and principle component analysis. *Journal of Materials Processing Technology* **2005**, *170(3)*, 602-610.
33. Ha, J.-L.; Kung, Y.-S.; Hu, S.-C.; Fung, R.-F. Optimal design of a micro-positioning Scott-Russell mechanism by Taguchi method. *Sensors and Actuators A* **2006**, *125(2)*, 565-572.
34. Ting, J.-H.; Shiau, S.-H.; Chen, Y.-J.; Pan, F.-M.; Wong, H.; Pu, G.M.; Kung C.-Y. Preparation and properties of sputtered nitrogen-doped cobalt silicide film. *Thin Solid Films* **2004**, *468*,155-160.
35. Alsaran, A.; Celik, A.; Celik, C. Determination of the optimum conditions for ion nitriding of AISI 5140 steel. *Surface and Coatings Technology* **2002**, *160*, 219-226.
36. Ali, N.; Neto, V.F.; Mei, S.; Cabral, G.; Kousar, Y.; Titus, E.; Ogwu, A.A.; Misra, D.S.; Gracio, J. Optimisation of the new time-modulated CVD process using the Taguchi method. *Thin Slid Films* **2004**, *469-471*, 154-160.
37. Tan, O.; Zaimoglu, A.S.; Hinislioglu, S.; Altun, S. Taguchi approach for optimization of the bleeding on cement-based grouts. *Tunnelling and Underground Space Technology* **2005**, *20(2)*, 167-173.
38. Sahin, Y. The prediction of wear resistance model for the metal matrix composites. *Wear* **2005**, *258*, 1717-1722.
39. Cheng, C.-C.; Young, M.-S.; Chuang, C.-L.; Chang, C.-C. Fabrication optimization of carbon fiber electrode with Taguchi Method. *Biosensors and Bioelectronics* **2003**, *18*, 847-855.
40. Houng, J.-Y.; Hsu, H.-F.; Liu, Y.-H., Wu, J.-Y. Applying the Taguchi robust design to the optimization of the asymmetric reduction of ethyl 4-chloro acetoacetate by bakers' yeast. *Journal of Biotechnology* **2003**, *100*, 239-250.
41. Oguz, E.; Keskinler, B.; Celik, C.; Celik, Z. Determination of the optimum conditions in the removal of Bomaplex Red CR-L dye from the textile wastewater using O_3, H_2O_2, HCO_3^- and PAC. *Journal of Hazardous Materials B* **2006**, *131*, 66-72.
42. Yan, B.H.; Wang, C.C.; Liu, W.D.; Huang, F.Y. Machining Characteristics of Al_2O_3/6061Al Composite using Rotary EDM with a Disklike Electrode. *International Journal of Advanced Manufacturing Technology* **2000**, *16*, 322-333.
43. Yu, Y.-C.; Chen, X.-X.; Hung, T.-R.; Thibault, F. Optimization of Extrusion Blow Molding Processes Using Soft Computing and Taguchi's Methos. *Journal of Intelligent Manufacturing* **2004**, *15*, 625-634.
44. Ming-Der, J.; Yih-Fong, T. Optimization Of Electron-Beam Surface Hardening Of Cast Iron For High Wear Resistance Using Taguchi Method. *International Journal of Advanced Manufacturing Technology* **2004**, *24*, 190-198.

45. Wu, D.H.; Tsai, Y.J.; Yen, Y.T. Robust design of quartz crystal microbalance using finite element and Taguchi method. *Sensors and Actuators B* **2003**, *92(3)*, 337-344.
46. Liu, C.H.; Chen, C.-C.A.; Huang, J.-S. The polishing of molds and dies using a compliance tool holder mechanism. *Journal of Materials Processing Technology* **2005**, *166*, 230-236.
47. Tosun, N.; Cogun, C.; Tosun, G. A study on kerf and material removal rate in wire electrical discharge machining based on Taguchi method. *Journal of Materials Processing Technology* **2004**, *152(3)*, 316-322.
48. Ghani, J.A.; Choudhury, I.A.; Hassan, H.H. Application of Taguchi method in the optimization of end milling parameters. *Journal of Materials Processing Technology* **2004**, *145(1)*, 84-92.
49. Statistica; Statsoft Inc., V.6.0., 2001; http://www.statsoft.com; accessed 20 April 2006.
50. Sylvester, J.J. Thoughts on inverse orthogonal matrices, simultaneous sign successions, and tessellated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers. *Philosophical Magazine* **1867**, *34*, 461-475.
51. Sloane, N.J.A. Table of Orthogonal Arrays of Strength 2 with up to 100 Runs, 2007; http://www.research.att.com/~njas/doc/cent4.html#T1; accessed 23 May 2007.
52. Sloane, N.J.A.; Stufken, J. A linear programming bound for orthogonal arrays with mixed levels. *Journal of Statistical Planning and Inference* **1996**, *56*, 295-305.
53. Rains, E.M.; Sloane, N.J.A.; Stufken, J. The lattice of N-run orthogonal arrays. *Journal of Statistical Planning and Inference* **2002**, *102(2)*, 477-500.

Appendix

L_4

Table A1. $4^2 \times 2^1$ scheme.

Experimental runs	Factor (levels) A(4)	B(4)	C(2)
1	0	2	0
2	1	0	1
3	2	3	1
4	3	1	0

Table A2. 2^3 scheme.

Experimental runs	Factor (levels) A(2)	B(2)	C(2)
1	0	1	0
2	1	1	1
3	0	0	1
4	1	0	0

L_6

Table A3. $6^1 \times 3^2$ scheme.

Experimental runs	Factor (levels) A(6)	B(3)	C(3)
1	0	1	0
2	1	1	2
3	2	0	1
4	3	2	2
5	4	2	0
6	5	0	1

Table A4. 3^3 scheme.

Experimental runs	Factor (levels) A(3)	B(3)	C(3)
1	0	1	1
2	1	0	2
3	2	2	2
4	0	1	1
5	1	2	0
6	2	0	0

Table A5. $3^2 \times 2^1$ scheme.

Experimental runs	Factor (levels) A(3)	B(3)	C(2)
1	1	1	0
2	1	1	1
3	2	0	0
4	2	2	1
5	0	2	0
6	0	0	1

L_8

Table A6. $4^4 \times 2^3$ scheme.

Experimental runs	Factor (levels)						
	A(4)	B(4)	C(4)	D(4)	E(2)	F(2)	G(2)
1	1	1	0	0	0	1	1
2	0	2	2	3	1	1	1
3	2	0	3	2	0	1	0
4	3	3	1	1	1	1	0
5	2	3	3	1	0	0	1
6	3	0	1	2	1	0	1
7	1	2	0	3	0	0	0
8	0	1	2	0	1	0	0

Table A7. $4^2 \times 2^5$ scheme.

Experimental runs	Factor (levels)						
	A(4)	B(4)	C(2)	D(2)	E(2)	F(2)	G(2)
1	1	0	0	1	1	1	1
2	2	3	1	1	1	1	0
3	2	3	0	1	0	0	1
4	1	0	1	1	0	0	0
5	0	2	0	0	1	0	0
6	3	1	1	0	1	0	1
7	3	1	0	0	0	1	0
8	0	2	1	0	0	1	1

Table A8. 2^7 scheme.

Experimental runs	Factor (levels)						
	A(2)	B(2)	C(2)	D(2)	E(2)	F(2)	G(2)
1	0	1	1	1	0	0	0
2	1	1	1	0	1	1	0
3	0	1	0	0	1	0	1
4	1	1	0	1	0	1	1
5	0	0	1	0	0	1	1
6	1	0	1	1	1	0	1
7	0	0	0	1	1	1	0
8	1	0	0	0	0	0	0

Table A9. $8^3 \times 4^2 \times 2^1$ scheme.

Experimental runs	Factor (levels)					
	A(8)	B(8)	C(8)	D(4)	E(4)	F(2)
1	1	1	1	0	1	1
2	0	4	5	1	3	0
3	2	6	7	2	0	1
4	3	3	3	3	2	0
5	6	7	2	0	1	0
6	7	2	6	1	3	1
7	5	0	4	2	0	0
8	4	5	0	3	2	1

Table A10. $8^1 \times 4^3 \times 2^2$ scheme.

Experimental runs	Factor (levels)					
	A(8)	B(4)	C(4)	D(4)	E(2)	F(2)
1	1	1	1	0	0	1
2	0	1	0	2	1	0
3	4	0	2	3	0	1
4	5	0	3	1	1	0
5	6	2	1	0	0	0
6	7	2	0	2	1	1
7	3	3	2	3	0	0
8	2	3	3	1	1	1

Table A11. $8^1 \times 4^1 \times 2^4$ scheme.

Experimental runs	Factor (levels) A(8)	B(4)	C(2)	D(2)	E(2)	F(2)
1	1	1	0	1	1	1
2	0	1	1	1	0	0
3	4	0	0	0	1	0
4	5	0	1	0	0	1
5	6	2	0	1	0	1
6	7	2	1	1	1	0
7	3	3	0	0	0	0
8	2	3	1	0	1	1

Table A12. 4^6 scheme.

Experimental runs	Factor (levels) A(4)	B(4)	C(4)	D(4)	E(4)	F(4)
1	0	0	0	2	2	2
2	0	3	3	2	1	1
3	1	0	3	0	2	1
4	1	3	0	0	1	2
5	2	1	1	3	0	0
6	2	2	2	3	3	3
7	3	1	2	1	0	3
8	3	2	1	1	3	0

Table A13. $8^3 \times 2^2$ scheme.

Experimental runs	Factor (levels) A(8)	B(8)	C(8)	D(2)	E(2)
1	1	1	1	0	1
2	0	4	5	1	1
3	6	6	6	0	1
4	7	3	2	1	1
5	2	7	3	0	0
6	3	2	7	1	0
7	5	0	4	0	0
8	4	5	0	1	0

Table A14. $8^2 \times 2^3$ scheme.

Experimental runs	Factor (levels) A(8)	B(8)	C(2)	D(2)	E(2)
1	1	0	1	1	1
2	5	1	1	0	0
3	7	2	0	1	0
4	3	3	0	0	1
5	0	4	0	0	0
6	4	5	0	1	1
7	6	6	1	0	1
8	2	7	1	1	0

Table A15. $8^1 \times 2^4$ scheme.

Experimental runs	**Factor (levels)**				
	A(8)	**B(2)**	**C(2)**	**D(2)**	**E(2)**
1	0	0	0	0	0
2	1	0	1	1	1
3	2	1	0	1	1
4	3	1	1	0	0
5	4	1	1	0	1
6	5	1	0	1	0
7	6	0	1	1	0
8	7	0	0	0	1

Table A16. 8^4 scheme.

Experimental runs	**Factor (levels)**			
	A(8)	**B(8)**	**C(8)**	**D(8)**
1	0	0	3	4
2	1	7	0	2
3	2	6	6	6
4	3	1	5	0
5	4	5	7	3
6	5	2	1	7
7	6	3	4	5
8	7	4	2	1

L_9

Table A17. 3^5 scheme.

Experimental runs	**Factor (levels)**				
	A(3)	**B(3)**	**C(3)**	**D(3)**	**E(3)**
1	0	0	0	0	0
2	0	0	2	2	1
3	0	2	0	2	2
4	1	1	2	0	2
5	1	2	1	0	1
6	1	2	2	1	0
7	2	0	1	1	2
8	2	1	0	1	1
9	2	1	1	2	0

Table A18. 9^4 scheme.

Experimental runs	Factor (levels)			
	A(9)	B(9)	C(9)	D(9)
1	0	0	7	5
2	1	8	0	4
3	2	1	1	1
4	3	7	8	6
5	4	6	6	0
6	5	5	3	7
7	6	2	2	8
8	7	4	5	3
9	8	3	4	2

Table A19. $9^2 \times 3^2$ scheme.

Experimental runs	Factor (levels)			
	A(9)	B(9)	C(3)	D(3)
1	0	1	1	0
2	1	2	0	2
3	2	4	2	1
4	3	7	2	2
5	4	8	1	1
6	5	6	0	0
7	6	5	1	0
8	7	3	0	2
9	8	0	2	1

Table A20. $9^1 \times 3^3$ scheme.

Experimental runs	Factor (levels)			
	A(9)	B(3)	C(3)	D(3)
1	0	1	1	1
2	1	1	0	0
3	2	0	2	2
4	3	1	0	2
5	4	2	2	1
6	5	2	1	0
7	6	0	2	0
8	7	2	1	2
9	8	0	0	1

L_{10}

Table A21. $10^1 \times 5^5$ scheme.

Experimental runs	Factor (levels)					
	A(10)	B(5)	C(5)	D(5)	E(5)	F(5)
1	0	0	0	0	2	2
2	1	0	4	4	2	2
3	2	4	4	1	0	3
4	3	4	0	3	1	0
5	4	3	1	4	3	4
6	5	3	3	0	4	1
7	6	2	3	2	4	1
8	7	2	1	2	3	4
9	8	1	2	3	1	0
10	9	1	2	1	0	3

Table A22. $5^2 \times 2^1$ scheme.

Experimental runs	Factor (levels)		
	A(5)	B(5)	C(2)
1	0	0	1
2	1	0	0
3	2	1	1
4	3	1	0
5	4	2	1
6	0	4	0
7	1	4	1
8	2	3	0
9	3	3	1
10	4	2	0

L_{12}

Table A23. 2^{11} scheme.

Experimental runs	Factor (levels)										
	A(2)	B(2)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)	J(2)	K(2)
1	0	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	1	1	1	0	0
3	0	1	1	0	0	0	1	0	0	1	1
4	1	1	0	1	1	0	0	1	0	1	1
5	0	1	0	0	0	1	0	1	1	1	0
6	1	1	0	0	1	1	1	0	1	0	1
7	0	0	1	0	1	0	0	1	1	0	1
8	1	0	1	1	0	1	0	0	1	1	1
9	0	0	0	1	1	0	1	0	1	1	0
10	1	0	1	0	1	1	1	1	0	1	0
11	0	0	0	1	0	1	1	1	0	0	1
12	1	0	0	0	0	0	0	0	0	0	0

Table A24. $4^1 \times 2^9$ scheme.

Experimental runs	Factor (levels)									
	A(4)	B(2)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)	J(2)
1	0	0	1	1	1	1	1	0	0	0
2	0	1	1	1	1	0	0	1	1	1
3	3	0	1	1	0	0	0	1	0	0
4	3	1	1	0	1	1	0	0	1	0
5	1	0	1	0	0	0	1	0	1	1
6	2	1	1	0	0	1	1	1	0	1
7	2	0	0	1	0	1	0	0	1	1
8	3	1	0	1	1	0	1	0	0	1
9	1	0	0	0	1	1	0	1	0	1
10	1	1	0	1	0	1	1	1	1	0
11	2	0	0	0	1	0	1	1	1	0
12	0	1	0	0	0	0	0	0	0	0

Table A25. $4^2 \times 2^7$ scheme.

Experimental runs	Factor (levels)								
	A(4)	B(4)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)
1	0	0	0	1	1	1	1	1	0
2	1	1	1	1	1	1	0	0	1
3	2	2	0	1	1	0	0	0	1
4	2	3	1	1	0	1	1	0	0
5	1	3	0	1	0	0	0	1	0
6	3	0	1	1	0	0	1	1	1
7	3	1	0	0	1	0	1	0	0
8	3	2	1	0	1	1	0	1	0
9	1	2	0	0	0	1	1	0	1
10	0	3	1	0	1	0	1	1	1
11	2	1	0	0	0	1	0	1	1
12	0	0	1	0	0	0	0	0	0

Table A26. $4^4 \times 2^3$ scheme.

Experimental runs	Factor (levels)						
	A(4)	B(4)	C(4)	D(4)	E(2)	F(2)	G(2)
1	1	1	1	2	0	1	1
2	1	1	1	2	1	1	1
3	0	2	2	0	0	1	1
4	1	2	3	3	1	1	0
5	3	0	2	1	0	1	0
6	3	3	0	1	1	1	0
7	3	2	0	2	0	0	1
8	2	0	2	3	1	0	1
9	0	3	1	3	0	0	0
10	2	3	3	0	1	0	1
11	2	1	3	1	0	0	0
12	0	0	0	0	1	0	0

Table A27. $4^3 \times 2^4$ scheme.

Experimental runs	Factor (levels)						
	A(4)	B(4)	C(4)	D(2)	E(2)	F(2)	G(2)
1	1	1	0	0	1	1	1
2	1	1	3	1	1	1	1
3	0	2	1	0	1	1	0
4	2	0	2	1	1	0	1
5	3	2	1	0	1	0	0
6	2	3	2	1	1	0	0
7	3	0	3	0	0	1	0
8	3	3	0	1	0	1	1
9	0	3	3	0	0	0	1
10	1	2	2	1	0	1	0
11	2	1	1	0	0	0	1
12	0	0	0	1	0	0	0

Table A28. 3^7 scheme.

Experimental runs	Factor (levels)						
	A(3)	B(3)	C(3)	D(3)	E(3)	F(3)	G(3)
1	0	1	1	1	1	1	0
2	1	1	1	1	0	2	1
3	2	1	1	1	1	1	2
4	0	0	0	0	2	2	2
5	1	1	1	1	2	0	1
6	2	0	2	2	1	1	2
7	0	0	2	2	1	1	0
8	1	2	0	2	0	2	1
9	2	2	2	0	2	2	0
10	0	2	2	0	0	0	2
11	1	2	0	2	2	0	1
12	2	0	0	0	0	0	0

Table A29. $3^4 \times 2^2$ scheme.

Experimental runs	Factor (levels)					
	A(3)	B(3)	C(3)	D(3)	E(2)	F(2)
1	1	1	1	1	0	1
2	1	1	1	1	1	1
3	1	1	1	0	0	1
4	1	1	1	0	1	1
5	0	0	2	2	0	1
6	2	2	0	2	1	1
7	0	2	0	1	0	0
8	0	2	2	2	1	0
9	2	0	0	2	0	0
10	2	0	2	1	1	0
11	2	2	2	0	0	0
12	0	0	0	0	1	0

Table A30. $3^3 \times 2^3$ scheme.

Experimental runs	Factor (levels)					
	A(3)	B(3)	C(3)	D(2)	E(2)	F(2)
1	1	1	1	0	1	1
2	1	1	0	1	1	1
3	0	0	2	0	1	1
4	1	1	1	1	1	0
5	1	1	0	0	1	0
6	2	2	2	1	1	0
7	2	2	0	0	0	1
8	0	2	1	1	0	1
9	0	2	2	0	0	0
10	2	0	2	1	0	1
11	2	0	1	0	0	0
12	0	0	0	1	0	0

Table A31. $4^3 \times 3^1 \times 2^2$ scheme.

Experimental runs	Factor (levels)					
	A(4)	B(4)	C(4)	D(3)	E(2)	F(2)
1	1	1	1	1	0	1
2	1	1	1	1	1	1
3	1	0	2	2	0	1
4	0	3	2	2	1	1
5	3	2	0	0	0	1
6	3	2	3	0	1	1
7	0	3	3	0	0	0
8	2	2	1	1	1	0
9	2	0	3	1	0	0
10	3	1	2	2	1	0
11	2	3	0	2	0	0
12	0	0	0	0	1	0

Table A32. $4^2 \times 3^1 \times 2^3$ scheme.

Experimental runs	Factor (levels)					
	A(4)	B(4)	C(3)	D(2)	E(2)	F(2)
1	1	1	1	0	1	1
2	1	1	2	1	1	1
3	0	2	0	0	1	1
4	1	2	1	1	1	0
5	3	0	2	0	1	0
6	3	3	0	1	1	0
7	3	2	0	0	0	1
8	2	0	1	1	0	1
9	0	3	2	0	0	0
10	2	3	2	1	0	1
11	2	1	1	0	0	0
12	0	0	0	1	0	0

Table A33. 4^6 scheme.

Experimental runs	Factor (levels)					
	A(4)	B(4)	C(4)	D(4)	E(4)	F(4)
1	0	0	0	0	0	2
2	1	0	0	3	3	0
3	2	0	3	0	3	1
4	3	1	1	3	1	3
5	0	1	3	3	0	2
6	1	3	0	1	3	3
7	2	1	3	1	2	2
8	3	2	1	0	0	1
9	0	3	2	1	2	1
10	1	3	2	2	1	0
11	2	2	2	2	2	3
12	3	2	1	2	1	0

Table A34. $12^1 \times 2^4$ scheme.

Experimental runs	Factor (levels)				
	A(12)	B(2)	C(2)	D(2)	E(2)
1	0	1	1	1	0
2	1	1	0	0	1
3	2	0	1	0	0
4	3	0	0	1	0
5	4	0	0	1	1
6	5	1	0	0	1
7	6	0	1	0	1
8	7	0	1	1	1
9	8	1	1	0	0
10	9	1	0	1	0
11	10	1	1	1	1
12	11	0	0	0	0

Table A35. 6^5 scheme.

Experimental runs	Factor (levels)				
	A(6)	B(6)	C(6)	D(6)	E(6)
1	0	0	0	0	3
2	1	0	5	5	0
3	2	1	0	5	4
4	3	1	5	0	2
5	4	2	4	1	5
6	5	2	2	4	3
7	0	5	4	2	4
8	1	5	1	2	0
9	2	4	3	4	5
10	3	4	3	3	1
11	4	3	2	3	2
12	5	3	1	1	1

Table A36. $4^2 \times 3^2 \times 2^1$ scheme.

Experimental runs	Factor (levels)				
	A(4)	B(4)	C(3)	D(3)	E(2)
1	1	1	0	1	1
2	1	1	1	1	1
3	1	2	2	1	0
4	0	2	0	0	1
5	3	0	1	1	0
6	3	2	2	0	1
7	3	3	0	0	0
8	0	3	1	2	0
9	2	3	2	2	1
10	2	1	0	2	0
11	2	0	1	2	1
12	0	0	2	0	0

Table A37. $4^1 \times 3^2 \times 2^2$ scheme.

Experimental runs	Factor (levels)				
	A(4)	B(3)	C(3)	D(2)	E(2)
1	1	1	1	0	1
2	2	1	1	1	1
3	1	1	1	0	1
4	2	1	1	1	1
5	3	0	0	0	1
6	0	2	2	1	1
7	1	0	2	0	0
8	2	0	2	1	0
9	0	2	0	0	0
10	3	2	0	1	0
11	3	2	2	0	0
12	0	0	0	1	0

Table A38. $3^1 \times 2^4$ scheme.

Experimental runs	Factor (levels)				
	A(3)	B(2)	C(2)	D(2)	E(2)
1	1	0	1	1	1
2	1	1	1	1	1
3	0	0	1	1	0
4	0	1	1	0	1
5	2	0	1	0	0
6	2	1	1	0	0
7	1	0	0	1	0
8	2	1	0	1	1
9	0	0	0	0	1
10	1	1	0	1	0
11	2	0	0	0	1
12	0	1	0	0	0

Table A39. 12^4 scheme.

Experimental runs	Factor (levels) A(12)	B(12)	C(12)	D(12)
1	0	0	0	0
2	1	1	10	11
3	2	11	1	10
4	3	10	11	1
5	4	9	2	7
6	5	2	9	6
7	6	8	7	5
8	7	7	8	2
9	8	6	5	4
10	9	5	6	8
11	10	3	4	9
12	11	4	3	3

Table A40. $12^1 \times 6^2$ scheme.

Experimental runs	Factor (levels) A(12)	B(6)	C(6)
1	0	1	1
2	1	1	1
3	2	0	5
4	3	4	5
5	4	5	2
6	5	5	0
7	6	4	2
8	7	3	3
9	8	3	4
10	9	2	4
11	10	2	3
12	11	0	0

L_{14}

Table A41. $7^5 \times 2^1$ scheme.

Experimental runs	Factor (levels)					
	B(7)	C(7)	D(7)	E(7)	F(7)	A(2)
1	0	0	0	0	5	0
2	0	0	6	4	1	1
3	1	6	0	6	0	0
4	1	6	1	2	5	1
5	2	1	4	5	2	0
6	2	5	5	0	2	1
7	3	5	6	1	3	0
8	3	4	4	6	4	1
9	4	4	5	3	6	0
10	4	1	2	5	6	1
11	5	2	3	2	1	0
12	5	3	2	3	3	1
13	6	3	3	4	4	0
14	6	2	1	1	0	1

Table A42. $14^1 \times 7^4$ scheme.

Experimental runs	Factor (levels)				
	A(14)	B(7)	C(7)	D(7)	E(7)
1	0	0	0	0	0
2	1	0	5	5	6
3	2	6	0	6	4
4	3	1	5	4	3
5	4	6	1	0	6
6	5	5	6	1	4
7	6	5	6	3	1
8	7	4	3	6	0
9	8	3	4	1	1
10	9	4	4	3	2
11	10	3	1	5	2
12	11	1	2	4	5
13	12	2	3	2	5
14	13	2	2	2	3

L_{15}

Table A43. $5^4 \times 3^3$ scheme.

Experimental runs	Factor (levels)							
	A(5)	B(5)	C(5)	D(5)	E(3)	F(3)	G(3)	H(3)
1	0	0	0	0	0	0	0	1
2	1	0	0	2	2	2	2	0
3	2	0	4	3	0	1	2	2
4	3	1	4	3	1	0	0	0
5	4	1	0	4	2	0	1	2
6	0	1	4	2	2	2	1	1
7	1	4	1	4	1	2	0	2
8	2	2	3	1	1	1	0	2
9	3	2	1	3	0	2	1	1
10	4	2	3	1	1	1	2	1
11	0	4	2	4	0	0	2	0
12	1	4	2	0	2	0	2	2
13	2	3	3	2	2	1	0	0
14	3	3	2	0	0	2	1	1
15	4	3	1	1	1	1	1	0

Table A44. $15^1 \times 3^6$ scheme.

Experimental runs	Factor (levels)						
	A(15)	B(3)	C(3)	D(3)	E(3)	F(3)	G(3)
1	0	0	0	0	0	1	1
2	1	0	0	2	1	0	1
3	2	0	2	0	2	2	1
4	3	2	2	2	0	2	0
5	4	2	2	0	1	1	2
6	5	2	0	2	2	1	0
7	6	2	1	1	0	0	2
8	7	2	0	1	2	2	2
9	8	0	2	2	2	1	1
10	9	1	2	1	1	0	0
11	10	1	1	1	0	1	1
12	11	1	1	1	2	0	2
13	12	1	1	0	1	0	0
14	13	1	0	0	1	2	0
15	14	0	1	2	0	2	2

Table A45. $5^5 \times 3^2$ scheme.

Experimental runs	Factor (levels)						
	A(5)	B(5)	C(5)	D(5)	E(5)	F(3)	G(3)
1	0	0	0	0	0	0	0
2	1	0	0	2	3	2	2
3	2	0	4	3	4	0	1
4	3	1	4	3	0	2	0
5	4	1	0	4	4	1	0
6	0	1	4	2	3	1	2
7	1	4	1	4	2	1	1
8	2	2	3	1	1	2	1
9	3	2	1	3	0	2	2
10	4	2	3	1	1	0	2
11	0	4	2	4	1	0	1
12	1	4	2	0	4	2	1
13	2	3	3	2	2	1	0
14	3	3	2	0	3	1	0
15	4	3	1	1	2	0	2

Table A46. $5^2 \times 3^5$ scheme.

Experimental runs	Factor (levels)						
	A(5)	B(5)	C(3)	D(3)	E(3)	F(3)	G(3)
1	0	0	0	0	0	0	0
2	1	0	0	0	2	2	2
3	2	0	2	2	0	0	2
4	3	1	0	2	1	1	2
5	4	1	2	0	0	2	0
6	0	1	2	2	2	2	0
7	1	4	0	2	0	1	1
8	2	2	2	1	2	0	1
9	3	2	1	2	1	1	0
10	4	2	1	1	1	2	2
11	0	4	2	0	1	1	2
12	1	4	1	1	0	2	1
13	2	3	1	1	2	0	1
14	3	3	0	1	2	1	0
15	4	3	1	0	1	0	1

Table A47. $5^1 \times 3^6$ scheme.

Experimental runs	Factor (levels)						
	A(5)	B(3)	C(3)	D(3)	E(3)	F(3)	G(3)
1	0	0	0	0	0	0	0
2	1	0	0	2	2	1	1
3	2	0	0	0	2	2	2
4	3	0	2	2	0	0	2
5	4	0	2	0	0	2	0
6	0	1	2	2	1	2	1
7	1	1	2	1	2	0	0
8	2	1	2	0	2	1	2
9	3	1	0	2	0	1	2
10	4	1	1	2	2	1	0
11	0	2	1	1	0	2	1
12	1	2	1	1	1	0	1
13	2	2	1	1	1	2	1
14	3	2	1	0	1	0	2
15	4	2	0	1	1	1	0

Table A48. 3^7 scheme.

Experimental runs	Factor (levels)						
	A(3)	B(3)	C(3)	D(3)	E(3)	F(3)	G(3)
1	0	0	0	0	0	0	1
2	1	0	0	0	2	2	0
3	2	0	0	2	0	1	2
4	0	0	2	2	0	2	1
5	1	0	2	0	2	0	1
6	2	1	0	2	2	0	1
7	0	1	2	2	2	0	2
8	1	1	1	2	1	2	0
9	2	1	2	0	1	2	2
10	0	2	0	1	2	2	2
11	1	2	1	0	0	1	2
12	2	1	2	1	1	1	0
13	0	2	1	1	1	1	0
14	1	2	1	1	0	0	0
15	2	2	1	1	1	1	1

Table A49. $15^1 \times 5^5$ scheme.

Experimental runs	Factor (levels)					
	A(15)	B(5)	C(5)	D(5)	E(5)	F(5)
1	0	0	0	0	0	0
2	1	0	4	3	3	3
3	2	0	2	3	3	4
4	3	4	0	4	2	1
5	4	4	0	1	4	3
6	5	4	4	0	0	4
7	6	3	4	0	3	0
8	7	3	3	3	2	2
9	8	3	2	4	1	2
10	9	2	3	4	2	0
11	10	2	2	1	4	1
12	11	2	1	2	0	4
13	12	1	3	2	1	1
14	13	1	1	2	1	2
15	14	1	1	1	4	3

Table A50. 5^6 scheme.

Experimental runs	Factor (levels)					
	A(5)	B(5)	C(5)	D(5)	E(5)	F(5)
1	0	0	0	0	0	0
2	1	0	0	4	3	2
3	2	0	4	0	4	3
4	3	1	4	0	0	3
5	4	1	0	2	4	4
6	0	1	4	4	3	2
7	1	4	1	1	2	4
8	2	2	3	4	1	1
9	3	2	1	3	0	3
10	4	2	3	3	1	1
11	0	4	2	2	1	4
12	1	4	2	1	4	0
13	2	3	3	3	2	2
14	3	3	2	1	3	0
15	4	3	1	2	2	1

Table A51. $5^3 \times 3^3$ scheme.

Experimental runs	Factor (levels)					
	A(5)	B(5)	C(5)	D(3)	E(3)	F(3)
1	0	0	0	0	0	0
2	1	0	0	2	2	2
3	2	0	4	0	0	2
4	3	1	4	0	2	0
5	4	1	0	2	1	0
6	0	1	4	2	2	1
7	1	4	1	0	2	0
8	2	2	3	2	0	1
9	3	2	1	0	1	2
10	4	2	3	1	2	1
11	0	4	2	1	1	2
12	1	4	2	1	1	1
13	2	3	3	2	0	0
14	3	3	2	1	0	1
15	4	3	1	1	1	2

L_{16}

Table A52. 2^{15} scheme.

Experimental runs	Factor (levels)														
	A(2)	B(2)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)	J(2)	K(2)	L(2)	M(2)	N(2)	O(2)
1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0
3	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1
4	1	1	1	0	0	1	1	0	0	1	1	0	1	1	0
5	0	1	0	0	1	1	0	0	0	1	0	1	1	0	1
6	1	1	0	1	0	1	0	1	0	0	1	1	0	1	1
7	0	1	0	0	0	0	1	1	1	1	1	0	0	0	1
8	1	1	0	0	1	0	1	1	1	0	0	1	1	1	0
9	0	0	1	0	1	0	1	0	0	0	1	1	0	1	1
10	1	0	1	1	0	0	1	1	0	1	0	1	1	0	1
11	0	0	1	0	0	1	0	1	1	1	0	1	0	1	0
12	1	0	1	0	1	1	0	1	1	0	1	0	1	0	1
13	0	0	0	1	1	0	0	1	0	1	1	0	1	1	0
14	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1
15	0	0	0	1	0	1	1	0	1	0	1	1	1	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table A53. $4^1 \times 2^{13}$ scheme.

Experimental runs	**Factor (levels)**													
	A(4)	**B(2)**	**C(2)**	**D(2)**	**E(2)**	**F(2)**	**G(2)**	**H(2)**	**I(2)**	**J(2)**	**K(2)**	**L(2)**	**M(2)**	**N(2)**
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	1	1	1	1	1	1	1
3	2	0	0	0	1	1	1	0	0	0	1	1	1	1
4	3	0	0	0	1	1	1	1	1	1	0	0	0	0
5	0	0	1	1	0	1	1	0	1	1	0	0	1	1
6	1	0	1	1	0	1	1	1	0	0	1	1	0	0
7	2	0	1	1	1	0	0	0	1	1	1	1	0	0
8	3	0	1	1	1	0	0	1	0	0	0	0	1	1
9	0	1	0	1	1	0	1	1	0	1	0	1	0	1
10	1	1	0	1	1	0	1	0	1	0	1	0	1	0
11	2	1	0	1	0	1	0	1	0	1	1	0	1	0
12	3	1	0	1	0	1	0	0	1	0	0	1	0	1
13	0	1	1	0	1	1	0	1	1	0	0	1	1	0
14	1	1	1	0	1	1	0	0	0	1	1	0	0	1
15	2	1	1	0	0	0	1	1	1	0	1	0	0	1
16	3	1	1	0	0	0	1	0	0	1	0	1	1	0

Table A54. $8^1 \times 2^{12}$ scheme.

Experimental runs	**Factor (levels)**												
	A(8)	**B(2)**	**C(2)**	**D(2)**	**E(2)**	**F(2)**	**G(2)**	**H(2)**	**I(2)**	**J(2)**	**K(2)**	**L(2)**	**M(2)**
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1	1	1	1	1	1	1
3	2	0	0	1	1	1	0	0	0	1	1	1	1
4	3	0	0	1	1	1	1	1	1	0	0	0	0
5	4	0	1	0	1	1	0	1	1	0	0	1	1
6	5	0	1	0	1	1	1	0	0	1	1	0	0
7	6	0	1	1	0	0	0	1	1	1	1	0	0
8	7	0	1	1	0	0	1	0	0	0	0	1	1
9	0	1	1	1	0	1	1	0	1	0	1	0	1
10	1	1	1	1	0	1	0	1	0	1	0	1	0
11	2	1	1	0	1	0	1	0	1	1	0	1	0
12	3	1	1	0	1	0	0	1	0	0	1	0	1
13	4	1	0	1	1	0	1	1	0	0	1	1	0
14	5	1	0	1	1	0	0	0	1	1	0	0	1
15	6	1	0	0	0	1	1	1	0	1	0	0	1
16	7	1	0	0	0	1	0	0	1	0	1	1	0

Table A55. $4^2 \times 2^{11}$ scheme.

Experimental runs	Factor (levels)												
	A(4)	B(4)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)	J(2)	K(2)	L(2)	M(2)
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1	1	1	1	1	1	1
3	2	0	0	1	1	1	0	0	0	1	1	1	1
4	3	0	0	1	1	1	1	1	1	0	0	0	0
5	0	1	1	0	1	1	0	1	1	0	0	1	1
6	1	1	1	0	1	1	1	0	0	1	1	0	0
7	2	1	1	1	0	0	0	1	1	1	1	0	0
8	3	1	1	1	0	0	1	0	0	0	0	1	1
9	0	2	1	1	0	1	1	0	1	0	1	0	1
10	1	2	1	1	0	1	0	1	0	1	0	1	0
11	2	2	1	0	1	0	1	0	1	1	0	1	0
12	3	2	1	0	1	0	0	1	0	0	1	0	1
13	0	3	0	1	1	0	1	1	0	0	1	1	0
14	1	3	0	1	1	0	0	0	1	1	0	0	1
15	2	3	0	0	0	1	1	1	0	1	0	0	1
16	3	3	0	0	0	1	0	0	1	0	1	1	0

Table A56. $16^1 \times 2^{11}$ scheme.

Experimental runs	Factor (levels)											
	A(16)	B(2)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)	J(2)	K(2)	L(2)
1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	1	1	1	1	1	1	1
3	2	0	1	1	1	0	0	0	1	1	1	1
4	3	0	1	1	1	1	1	1	0	0	0	0
5	4	1	0	1	1	0	1	1	0	0	1	1
6	5	1	0	1	1	1	0	0	1	1	0	0
7	6	1	1	0	0	0	1	1	1	1	0	0
8	7	1	1	0	0	1	0	0	0	0	1	1
9	8	1	1	0	1	1	0	1	0	1	0	1
10	9	1	1	0	1	0	1	0	1	0	1	0
11	10	1	0	1	0	1	0	1	1	0	1	0
12	11	1	0	1	0	0	1	0	0	1	0	1
13	12	0	1	1	0	1	1	0	0	1	1	0
14	13	0	1	1	0	0	0	1	1	0	0	1
15	14	0	0	0	1	1	1	0	1	0	0	1
16	15	0	0	0	1	0	0	1	0	1	1	0

Table A57. $16^1 \times 8^1 \times 2^8$ scheme.

Experimental runs	Factor (levels)									
	A(16)	B(8)	C(2)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)	J(2)
1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	1	1	1	1	1	1
3	2	1	0	1	0	0	0	1	1	1
4	3	1	1	1	1	1	1	0	0	0
5	4	2	1	1	0	1	1	0	1	1
6	5	2	1	1	1	0	0	1	0	0
7	6	3	1	0	1	0	1	1	0	1
8	7	3	1	0	0	1	0	0	1	0
9	8	7	0	0	0	1	1	1	0	0
10	9	7	1	0	1	0	0	0	1	1
11	10	6	0	1	1	0	1	0	1	0
12	11	6	1	1	0	1	0	1	0	1
13	12	5	0	1	1	1	0	1	1	0
14	13	5	0	1	0	0	1	0	0	1
15	14	4	0	0	1	1	0	0	0	1
16	15	4	1	0	0	0	1	1	1	0

Table A58. 4^9 scheme.

Experimental runs	Factor (levels)								
	A(4)	B(4)	C(4)	D(4)	E(4)	F(4)	G(4)	H(4)	I(4)
1	0	0	0	0	0	0	0	2	2
2	1	0	0	0	3	2	3	0	2
3	2	0	0	3	2	3	1	3	0
4	3	0	3	2	0	0	1	1	0
5	0	1	3	2	0	3	3	2	3
6	1	1	3	1	3	0	3	3	1
7	2	1	3	1	1	3	1	0	2
8	3	1	2	2	3	1	1	2	3
9	0	2	2	3	2	1	0	0	2
10	1	2	2	1	3	3	0	1	0
11	2	2	0	3	0	2	3	1	1
12	3	2	1	2	2	1	2	2	3
13	0	3	1	3	2	0	2	1	1
14	1	3	2	0	1	2	2	3	0
15	2	3	1	1	1	2	0	3	3
16	3	3	1	0	1	1	2	0	1

Table A59. $4^3 \times 2^6$ scheme.

Experimental runs	Factor (levels)								
	A(4)	B(4)	C(4)	D(2)	E(2)	F(2)	G(2)	H(2)	I(2)
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	1	1	1	1	1
3	2	0	3	0	0	0	0	1	1
4	3	0	2	1	0	1	1	0	1
5	0	1	2	1	1	0	1	1	0
6	1	1	1	1	0	1	0	1	0
7	2	1	1	1	1	0	1	0	1
8	3	1	2	1	1	0	0	0	0
9	0	2	3	1	1	1	0	1	1
10	1	2	1	0	1	1	1	0	0
11	2	2	3	0	1	1	0	0	0
12	3	2	2	0	0	1	1	1	0
13	0	3	3	0	0	0	1	0	1
14	1	3	0	1	0	1	0	0	1
15	2	3	1	1	0	0	1	1	0
16	3	3	0	0	1	0	0	1	1

Table A60. 8^7 scheme.

Experimental runs	Factor (levels)						
	A(8)	B(8)	C(8)	D(8)	E(8)	F(8)	G(8)
1	0	0	0	0	0	0	3
2	1	0	0	7	7	7	4
3	2	1	7	0	7	4	5
4	3	1	7	7	0	5	3
5	4	2	6	1	5	4	1
6	5	2	6	6	1	2	4
7	6	3	1	6	4	1	6
8	7	3	3	5	5	2	0
9	0	7	5	5	4	0	2
10	1	7	3	4	3	5	7
11	2	6	5	4	6	3	2
12	3	6	1	3	3	6	1
13	4	5	4	3	2	3	7
14	5	5	2	2	1	6	0
15	6	4	4	1	2	7	6
16	7	4	2	2	6	1	5

Table A61. 16^5 scheme.

Experimental runs	Factor (levels) A(16)	B(16)	C(16)	D(16)	E(16)
1	0	0	0	0	0
2	1	1	13	9	14
3	2	2	12	13	8
4	3	15	1	1	15
5	4	14	2	15	3
6	5	13	15	2	1
7	6	12	14	8	7
8	7	11	3	14	10
9	8	10	11	11	2
10	9	3	4	12	12
11	10	9	9	4	11
12	11	8	10	3	13
13	12	7	7	7	9
14	13	4	8	6	5
15	14	6	5	10	4
16	15	5	6	5	6

Table A62. $16^2 \times 2^3$ scheme.

Experimental runs	Factor (levels) A(16)	B(16)	C(2)	D(2)	E(2)
1	0	0	0	0	0
2	1	13	0	1	1
3	2	12	1	0	1
4	3	1	0	1	1
5	4	2	1	0	0
6	5	15	0	1	0
â7	6	14	1	0	0
8	7	3	1	1	1
9	8	11	1	0	1
10	9	4	1	1	0
11	10	9	1	1	0
12	11	10	0	1	0
13	12	7	0	0	1
14	13	8	0	0	1
15	14	5	1	1	1
16	15	6	0	0	0