社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－MPS－50　(6)
2004／6／22

不完全な部品の組合せ問題について (III) —
誤差がベクトルの場合の最大誤差最小化

嘉村 友作　　　　　中森 眞理雄

東京農工大学工学部 情報コミュニケーション工学科
〒 184-8588 東京都小金井市中町 2-24-16

概要

半導体露光装置のレンズシステムに対するレンズ個体の最適な組合せ法を考える．レンズシステムは m 枚のレンズから構成されるとする．また，レンズ個体は n 枚ずつのロット単位として与えられ，m 個のロット $A_1, A_2, \ldots, A_m$ のすべてのレンズ個体を用いて n 個のレンズシステムを作る．レンズ個体の誤差は多次元ベクトルで与えられ, システムの誤差は，システムを構成する個体の誤差ベクトルの和の最大成分とする．本論文の目的は，n 個のレンズシステムの中の最大誤差を最小とする組合せ方法を提案することである．まず問題を，辺のコストがベクトルであり組合せの中の最大値を最小にすることを目的とする，一般の割当て問題を変形した割当て問題として記述する．そして，多項式時間の新しいアルゴリズムを提案し，提案したアルゴリズムによる計算例を示す.

Combining Imperfect Components (III) —
Minimax Optimization of Multidimensional Cost Error

Yuusaku Kamura　　Mario Nakamori

Department of Computer, Information, and Communication Sciences
Tokyo A & T University
Koganei, Tokyo 184-8588, Japan

Abstract

We consider the problem of optimally combining individual lenses for the lens systems used in semiconductor exposure equipment. Given a lens system with m lenses $A_1, A_2, \ldots, A_m$, then for n lens systems we need n individual lenses for each A_i. Our objective is to combine the n individual lenses in such a way that the maximum aberration occurring in any of the n combinations is minimized. We formulate the problem as a modified assignment problem with a vector cost and a minimax objective function. A new algorithm of polynomial time is proposed and the results from numerical experiments are presented.

1 Introduction

We consider a lens system for semiconductor exposure equipment consisting of m lenses, which we denote as $A_1, A_2, \ldots, A_m$. (We will refer to $A_1, A_2, \ldots, A_m$ as "components.") Factories producing such equipment normally receive these components in lots of many individual lenses. For the sake of simplicity, suppose we have from these lots n lenses for component A_1, n lenses for component A_2, …, n lenses for component A_m, and we want to combine them to make n lens systems. Let us choose i_1th lens from the lot of A_1, the i_2th lens from the lot of A_2, …, the i_mth lens from the lot of A_m; then choose the i'_1th lens from the lot of A_1, the i'_2th lens from the lot of A_2, …, the i'_mth lens from the lot of A_m; …; and finally, choose the $i_1^{(m)}$th lens from the lot of A_1, the $i_2^{(m)}$th lens from the lot of $A_2, \dots$, the $i_m^{(m)}$th lens from the lot of A_m. Thus, we have selected n combinations of lenses for n lens sysmtems. Although such lenses are manufactured very precisely, each lens will have its own unique aberration (error). Therefore, a variety of compounded aberration combinations (i.e., a variety of lens systems of differing total aberration) will result. Some of the combinations (some of the lens systems) will have a small compounded aberration while others will have a large compounded aberration. Our goal is to select optimum combinations, that is, we want to combine lenses so as to minimize the maximum aberration occurring in any one lens system. Normally, aberration for each lens is given as a vector (of more than 300 dimensions). We denote the aberration vector of the ith component and jth lens by $\mathbf{a}_{ij}$ $(i = 1, 2, \dots, m;\ j = 1, 2, \dots, n)$. In the following discussion, we consider the case of $m = 2$ and assume that all entries of $\mathbf{a}_{ij}$'s are nonnegative.

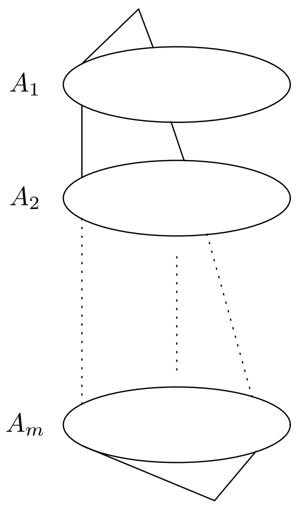

Figure 1: Lens system

In a former study [1] we proposed an algorithm of obtaining optimal combinations when the aberration vector is one dimensional (i.e., scalar). That algorithm is as follows:

Algorithm 1

Step 1: Arrange n lenses for the A_1 component in ascending order.

Step 2: Arrange n lenses for the A_2 component in ascending order.

Step 3: Combine the best lens for A_1 and the worst lens for A_2; combine the second best lens for A_1 and the second worst lens for A_2; combine the third best lens for A_1 and the third worst lens for A_2; and so on. □

In this paper, we propose an algorithm for selecting optimal combinations in the case where the aberration vector is two dimensional.

2 Formulation as an Integer Programming Problem

We first formulate the problem as a 0-1 integer programming problem.

Problem 1

Minimize z

subject to

$$\begin{aligned}
&\sum_j x_{jk} = 1 \quad (k = 1, 2, \dots, n),\\
&\sum_k x_{jk} = 1 \quad (j = 1, 2, \dots, n),\\
&\sum_k (a_{1j}^{(1)} + a_{2k}^{(1)}) x_{jk} \le z,\\
&\sum_k (a_{1j}^{(2)} + a_{2k}^{(2)}) x_{jk} \le z,\\
&z \ge 0,\\
&x_{jk} = 0 \quad \text{or} \quad 1. \qquad \square
\end{aligned}$$

Here $a_{1j}^{(1)}$ is the first entry of $\mathbf{a}_{1j}$, and $a_{1j}^{(2)}$ is the second entry. Similarly, $a_{2k}^{(1)}, a_{2k}^{(2)}$ are entries of $\mathbf{a}_{2k}$. Variable x_{jk} takes the value 1 when $\mathbf{a}_{1j}$ is combined with $\mathbf{a}_{2k}$, and 0 otherwise. The integral condition for x_{jk} is essential, otherwise we would usually obtain a noninteger solution.

3 Algorithm for Problem 1

In this section, we propose an algorithm that makes use of the problem character rather than trying to solve Problem 1 directly as an integer programming problem. The fundamental idea is to repeatedly solve the well known assignment problem.

3.1 Preliminaries

Lemma 1 For $(a_{1j}^{(1)}, a_{1j}^{(2)}) \in A_1, (a_{2k}^{(1)}, a_{2k}^{(2)}) \in A_2$, we assume that

$$m_1^{(1)} = E(a_{1j}^{(1)}), \quad m_1^{(2)} = E(a_{1j}^{(2)}), \quad (j = 1, 2, \dots, n);$$

$$m_2^{(1)} = E(a_{2k}^{(1)}), \quad m_2^{(2)} = E(a_{2k}^{(2)}), \quad (k = 1, 2, \dots, n).$$

We consider the permutation π on $\{1, 2, \dots, n\}$ which minimizes

$$\max_{1 \le j \le n} \{a_{1j}^{(1)} + a_{2\pi(j)}^{(1)}, a_{1j}^{(2)} + a_{2\pi(j)}^{(2)}\}.$$

The permutation π is that which for each j makes the larger of $a_{1j}^{(1)}+a_{2\pi(j)}^{(1)}$ and $a_{1j}^{(2)}+a_{2\pi(j)}^{(2)}$ to $m_1^{(1)}+m_2^{(1)}$ or $m_1^{(2)}+m_2^{(2)}$ as close as possible. □

From Lemma 1, we can conclude that our strategy is to find the permutation producing the larger of the $\mathbf{a}_{1j}+\mathbf{a}_{2k}$ entries in the sum of their averages.

Let X be a probabilistic variable that satisfies $E(X)=m$ and $V(X)=\sigma^2$. We can transform X to another probabilistic variable Z that satisfies $E(Z)=0$ and $V(Z)=1$ by the z-transformation

$$Z=\frac{X-m}{\sigma}.$$

We call such a transformation the normalization of X.

Regarding $a_{1j}^{(1)}$ as a probabilistic variable and assuming that $\tilde{a}_{1j}^{(1)}$ represents its normarilzation, evidently the average of $\tilde{a}_{1j}^{(1)}$ is 0 and $\tilde{a}_{1j}^{(1)}$ becomes 0 only when $a_{1j}^{(1)}=m_1^{(1)}$. The z-transformation is a linear transformation, so if $a_{1j}^{(1)}$ increases from $m_1^{(1)}$ linearly, and $\tilde{a}_{1j}^{(1)}$ increases from 0 linearly.

Similarly applying the z-transformation to $a_{1j}^{(2)}$, we denote its normalization by $\tilde{a}_{1j}^{(2)}$. Putting $(\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)})$ on a Euclid $(\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)})$-plane, then $(\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)})$ corresponding to $(a_{1j}^{(1)},a_{1j}^{(2)})$ satisfies $a_{1j}^{(1)}>m_{(1)}^1$ and $a_{1j}^{(2)}>m_{(1)}^1$ is in the first quadrant. This is similar for $(\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)})$ in the second,third and fourth quadrants. The origin of the plane corresponds to $(a_{1j}^{(1)},a_{1j}^{(2)})=(m_1^{(1)},m_1^{(2)})$ and the position of $(\tilde{a}_{1j}^1,\tilde{a}_{1j}^2)$ depends linearly on $a_{1j}^{(1)}-m_1^{(1)}$ and $a_{1j}^{(2)}-m_1^{(2)}$, respectively. We also apply z-transformation to $a_{2k}^{(1)}$ and $a_{2k}^{(2)}$ and denote their normalizations by $\tilde{a}_{2k}^{(1)}$ and $\tilde{a}_{2k}^{(2)}$, respectively.

Def. 1 (Maximum norm)
For $\mathbf{u}=(u_1,u_2)$ and $\mathbf{v}=(v_1,v_2)$,

$$d_{\max}(\mathbf{u},\mathbf{v})=\max\{|u_1-v_1|,|u_2-v_2|\}$$

defines a norm, which we will call the "maximum norm." □

By the normalization, $a_{1j}^{(1)}+a_{2k}^{(1)}=m_1^{(1)}+m_2^{(1)}$ and $a_{1j}^{(2)}+a_{2k}^{(2)}=m_1^{(2)}+m_2^{(2)}$ become equivalent to $\tilde{a}_{1j}^{(1)}+\tilde{a}_{2k}^{(1)}=0$ and $\tilde{a}_{1j}^{(2)}+\tilde{a}_{2k}^{(2)}=0$, respectively. Therefore we can state our problem as follows:

To find the permutation π on $\{1,2,\ldots,n\}$ such that the largest value of the $2n$ values

$$|\tilde{a}_{1j}^{(1)}+\tilde{a}_{2\pi(j)}^{(1)}|,|\tilde{a}_{1j}^{(2)}+\tilde{a}_{2\pi(j)}^{(2)}|,$$

is as small as possible.

To realize this, for every $(\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)})$ we take the symmetrical point $(-\tilde{a}_{1j}^{(1)},-\tilde{a}_{1j}^{(2)})$ as the origin. It is then sufficient to find the permutation π that combines with $(\tilde{a}_{2k}^{(1)},\tilde{a}_{2k}^{(2)})$ to produce the smallest as possible maximum norm. That is, apply the symmetric transformation for the origin to all $(\tilde{a}_{2k}^{(1)},\tilde{a}_{2k}^{(2)})$ so they are transformed to $(-\tilde{a}_{2k}^{(1)},-\tilde{a}_{2k}^{(2)})$, then find the permutation that minimizes the maximum norm of $(\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)})$ and $(-\tilde{a}_{2k}^{(1)},-\tilde{a}_{2k}^{(2)})$.

Here, we restate our problem as follows:

Problem 1′
Find the permuation π on $\{1,2,\ldots,n\}$ that minimizes

$$\max_{j=1,2,\ldots,n}\{d_{\max}((\tilde{a}_{1j}^{(1)},\tilde{a}_{1j}^{(2)}),(-\tilde{a}_{2\pi(j)}^{(1)},-\tilde{a}_{2\pi(j)}^{(2)}))\}.$$

3.2 Calculate an Approximate Solution

In order to obtain an approximate solution for Problem 1′, we solve it as a minimum cost assignment problem. Let V_1 and V_2 be vertex sets, and for $v(\mathbf{a}_{1j})\in V_1$ and $v(\mathbf{a}_{2k})\in V_2$ let e_{jk} be the directed edge from $v(\mathbf{a}_{1j})$ to $v(\mathbf{a}_{2k})$ and c_{jk} be the its weight defined as:

$$c_{jk}=\max\{|\tilde{a}_{1j}^{(1)}+\tilde{a}_{2k}^{(1)}|,|\tilde{a}_{1j}^{(2)}+\tilde{a}_{2k}^{(2)}|\}.$$

Then, our problem is to minimize

$$z=\sum_{(j,k)}c_{jk}x_{jk}$$

subject to

$$\sum_k x_{jk}=1,\quad \sum_j x_{jk}=1,\quad x_{jk}\geq 0.$$

This solution is not exact for Problem 1′, because it is not the maximum cost but the total cost that is minimized by the matching.

3.3 Algorithm for Problem 1′

We first define an algorithm $A(R)$ as follows:
(1) Replace all c_{ij}'s such that $c_{ij}\geq w$ by a sufficiently large value R.
(2) Solve the assignment problem using the above c_{ij}'s.

We are going to find the smallest value $\hat{R}$ (which we will call "*best_max*") of R such that the objective function of the assignment by $A(R)$ is not greater than R. We find $\hat{R}$ by binary search. Since the assignment problem is solved in polynomial time (e.g., $O(n^3)$) and the binary search is executed through n^2 values, the total computational time is also polynomial (e.g., $O(n^3 \log n)$. Note that $\log(n^2) = 2 \log n$).

Algorithm 2

```
Step 1: For all $j, k$ $(j, k = 1, 2, \ldots, n)$ compute
    $$d_{jk} = \max(a_{1j}^{(1)} + a_{2k}^{(1)}, a_{1j}^{(2)} + a_{2k}^{(2)})$$
and store these values in an array $w$. Since there are $n^2$ of the $d_{jk}$'s, the size of array $w$ is $n^2$.
Step 2: Sort the entries of $w$ in ascending order.
Step 3:
(0) $sl \leftarrow 1$;   $su \leftarrow n^2$.
(1) $mid \leftarrow \lceil (sl + su)/2 \rceil$.
(2) Execute $A(w[mid])$.
(3) If the objective function of the assignment in (2) is greater than $best\_max$, then $sl \leftarrow mid$; else $su \leftarrow mid$.
(4) If $su - sl > 1$, then go to (1).
Step 4: The assignment in Step 3 is the solution.
```

4 Numerical Experiments

We present the results obtained by solving the integer programming problem directly and apply our proposed algorithm to the same examples. Our computational environment is as follows:

CPU	Intel Celeron 2.0 GHz
Memory	512 MB
OS	MS-Windows 2000

We used the NUOPT Ver. 5.1.3 mathematical programming and modelling solver by the Mathematical Systems Institute, Inc.[1].

In every problem, the number of data $n = 50$. For each $\mathbf{a}_{1j}$ and $\mathbf{a}_{2k}$, we give correlational coefficients ρ_1 for $a_{1j}^{(1)}$ and $a_{1j}^{(2)}$ and ρ_2 for $a_{2k}^{(1)}$ and $a_{2k}^{(2)}$. Probabilistic variables $a_{1j}^{(1)}, a_{1j}^{(2)}, a_{2k}^{(1)}, a_{2k}^{(2)}$ follow the normal distribution and all have average and variance of 10.0 and 0.0, respectively.

Table 1 : Calculate by Algorithm 2 and NUOPT

ρ_1	ρ_2	optimal value	time(s)	
			Algorithm 2	NUOPT
-1.0	-1.0	21.371368342	2.88	220.75
-1.0	1.0	21.576122778	2.12	258.91
-0.5	0.5	20.985355631	3.04	2694.45
0.0	0.0	20.902542908	3.21	1464.02
0.5	0.5	21.251067699	3.11	2147.94
1.0	1.0	20.174854698	2.58	10946.95

[1]http://www.msi.co.jp/nuopt/

5 Conclusions

In this paper, we considered a permutation that minimizes the maximum component of n combinations of two vectors. We first formulated this problem as an integer programming problem. We then proposed an $O(n^3 \log n)$ algorithm for the problem and presented results from computational experiments using the algorithm. The problem of finding a permutation that minimizes the total cost of vector combinations, like the general assignment problem, is left to further research.

References

[1] Y.Kamura and M.Nakamori. Combining Imperfect Components to Minimize the System's Error, *Proc. PDPTA'01*, pp.1277-1283, 2001.

[2] Y.Kamura, M.Nakamori and Y.Shinano. Combining Imperfect Components(II) — The Case of Multidimensional Error, *Proc. PDPTA'02*, pp.228-232, 2002.

[3] J.E.Hopcroft and R.M.Karp. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs, *SIAM J. Comput.*, pp.225-231, 1973.

[4] M.Fushimi. *Probabilistic Methods and Simulations* (in Japanese), Iwanami Shoten, 1994.

[5] N.Inagaki. *Statistical Mathematics* (in Japanese), Shokabo, 2003.

[6] G.L.Nemhauser, A.H.G.Rinnooy Kan and M.J.Tood (eds.). *OPTIMIZATION, Handbooks in Operations Research and Management Science*, Vol.1, Elesevier Sci. Pub., 1989.

[7] J.van Leeuwen (ed.). *Handbook of Theoretical Computer Science*, Elesevier Sci. Pub.,1990.