East-West J. of Mathematics: Vol. 19, No 2 (2017) pp. 105-148

A SURVEY ON COMPUTATIONAL ALGEBRAIC STATISTICS AND ITS APPLICATIONS

Nguyen V. Minh Man

Center of Excellency in Mathematics, Ministry of Education;
Dept. of Mathematics, Faculty of Science, Mahidol University, Thailand;
and
Faculty of Environment and Resources
University of Technology, VNUHCM, Vietnam
e-mail: man.ngu@mahidol.edu

Abstract

Computational Algebraic Statistics is a new mathematics-based scientific discipline that is combined of three disciplines of Computing, Algebra and Statistics. This review introduces integrating powerful techniques of - among other things- Polynomial Algebra, Algebraic Geometry, Group Theory and Computational Statistics in this new field of Computational Algebraic Statistics (in brief CAS). CAS currently plays an essential and powerful role in some importantly applicable fields growing very fast in recent years, from science to engineering, such as: statistical quality control, process engineering, computational biology, complex biological networks, life sciences, data analytics and finance studies.

Furthermore, by combining the theory of algebraic geometry with graph theory, we point out connections between integer linear programming (ILP), mathematical modeling in traffic engineering, logistics management and transportation science.

These methods and algorithms are based on elegant ideas from some active fields of mathematics and statistics, and their useful applications

Key words: algebraic geometry, algebraic statistics, computational science, computer algebra, experimental design, Groebner basis methodology, graph theory, group-theoretic computation, industrial statistics, industrial manufacturing, integer programming, Lie and Weyl algebra, mixed orthogonal arrays, policy makers, process control, quality control, statistical inference, statistical optimization, symbolic computation
2010 AMS Mathematics classification: 05B15, 05C50, 05E18, 05E20, 12Y05, 13P10, 13P15, 15A18 , 20B25, 20B35, 62K15, 62P30, 68W30, 90B06, 90C11, 92B05

can be potentially found in various scientific and technological sectors.

Introduction

Computer Algebra - a briefly mixed name of computing and algebra, as a part of both fields of computational mathematics and scientific computing, also called *symbolic computation* or algebraic computation, is a scientific area developed around 1970s that refers to the study and development of algorithms and software for manipulating mathematical expressions and other mathematical objects. Though it is a sub-field of scientific computing, while naming symbolic computation we emphasize exact computation with expressions containing variables that have no given value and are manipulated as symbols [11]. The core machinery, which makes all computations algebraically feasible and computationally tractable is the *Groebner Basis* method (see Appendix A in Section 4) being invented 1965, by Bruno Buchberger, an Austrian mathematician.

Recently, around the year 2000 *Algebraic Statistics* [3] was briefly named for the study of the algebraic structures underlying *statistical inference and modeling*. We could simply think the algebraic structures consist of linear algebra, commutative algebra, algebraic combinatorics, and the most important subject is algebraic geometry. Saying statistical inference we mean, in the broadest sense, any meaningful reasoning on samples of a population that could be made by using mathematical tools. The mathematical area named *Computational Algebraic Statistics* - CAS, therefore essentially is a newly scientific domain being intertwined from three subjects of computing, algebra and statistics.

There are two major parts in this review, in both parts **algebraic structures** play a vital role. At first, the *computer algebra* based approach is used for Pure Mathematics in Section 1 and furthermore, for Operations Research in Section 3. Secondly, Section 2 shows how the *algebraic statistics* based approach is employed in Quality Engineering, more specifically in Industrial Manufacturing.

The major aim of this writing is to express that, although each field has distinct strength, solving complex problems in various domains nowadays basically requires a multi-dimensional view and integrated thinking, but joyful and worthy to do.

Why study Computational Algebraic Statistics (CAS)?

We aim to understand statistical problems by looking through algebraic glasses; hope the study might explain unusual phenomena being observed but we could not produce elucidated explanations. Furthermore, the process could lead to purely mathematical problems, and possibly lead us to a unifying framework for discussing and exploring new connections to active research fields of computational biology, finance, and reliability of complex systems. The key philosophy of using CAS is two folds: focusing on the modeling and representation phase, of either of objects of interest or complex data sets; and artfully putting heavy and a bit boring computational tasks to computer-based algorithms.

Related Literature

Briefly speaking, CAS appears in or is intimately used in the below themes that use algebraic geometry, Groebner bases, quantifier elimination ... as major tools:

- Exact hypothesis tests of conditional independence (Diaconis, 1998 [14]);
- Experiment designs (G. Pistone and H. Wynn, [18, 20]; Nguyen, 2005 [37]);
- Geometric intersection in automobile industry, (A. Morgan, 2009 [2]);
- Reliability theory and engineering (Ron Kenett, 2014 [39]);
- Computational biology, e.g. biological multiple sequence alignments, and Phylogenetics (Pachter and Sturmfels, 2005 [25], Olson et al. [35]);
- Life sciences: in health care alone, computer algebra has been used in work that bears on cancer, public health issues (which include risk analysis, survival analysis, drug testing, epidemiology), clinical medicine (specifically medical imaging), population and evolutionary genetics, bioengineering (including computer vision and ergonomic design), biochemical kinetics ... kindly see Barnett, 2002 [34] for a full survey;
- Algebraic biology- goes beyond key themes of CAS- a new way of applying algebraic computation and statistical inference to the study of biological problems, especially molecular structures in general; see more in [7].

Currently active researches are included in two major schools:

The European School of Algebraic Statistics created by G. Pistone and H. P. Wynn in their pioneering paper titled *Generalised confounding with Groebner bases* in Biometrika, 1996 [18], and then the work have been concretely shaped in 2000–2001 respectively [see [19] and [20]].

Their innovative ideas are elegant and precise formulations of complex questions in Statistics, in particular in Designs of Statistical Experiments. With those formulations, efficient algebraic techniques are used to obtain solutions (predictions or quantitative inferences ...), and then powerful computer algebra systems are employed to speed up the computation.

The United States School of Algebraic Statistics. Almost the same time, Diaconis and Sturmfels in [14] firstly proposed an algebraic approach for classical sampling problems. Bernd Sturmfels at Berkeley and a group of multidisciplinary scientists studying Statistics and Computational Biology with the various algebraic tools in other emerging areas apparently.

The problems they have concerned mostly are life sciences-related questions including biologically sequence alignments, key mechanisms of complex biology networks [Lior Pachter and Bernd Sturmfels, 2005 [25]].

Most recent applications using CAS presented in this work include:

1. *Pure Mathematics*: Weyl algebras, non-commutative algebras in general (Nguyen, 1998 [36])

2. *Statistical Quality Control*: Factorial Designs in industrial manufacturing (Nguyen, 2005 [37], Kenett, 2014 [39], Pistone [20])

3. *Operations Research*: Optimal Vehicle Routing in logistics planning.

The remaining parts are shown as follows. Firstly, the well known Dixmier conjecture on Weyl algebras (being related to the Jacobean conjecture) is formulated in Section 1, coupling with a solution of using computer algebra to disprove it. Looking to quality engineering, Section 2 discusses major methodologies proposed by the European school of algebraic statisticians. As an illustration, we sketch an industry-oriented problem that can be handled by CAS. In Section 3 we consider a problem of finding optimal routes in manufacturing with a balance of source and sink. Last but not least, mathematical treatments for the discussed applications are reviewed in Appendix A- about the Groebner basis methodology, a computational machinery being essential for handling multivariate polynomial systems; and in Appendix B on basic facts of permutation group.

1 Testing conjectures on a Weyl algebra

We discuss about two conjectures both formulated on Weyl algebra (the algebra of polynomial differential operators), namely Jacques Dixmier's conjecture (1968) and Nguyen Huu Anh's conjecture (1997). The first one, raised by Jacques Dixmier, is the conjecture that whether every algebra endomorphism of the first Weyl algebra over a characteristic zero field is an automorphism. Some authors (Tsuchimoto in 2005, Belov-Kanel and Kontsevich in 2007) showed that the Dixmier conjecture is stably equivalent to the well known Jacobian conjecture, whereby the Jacobian conjecture itself is ranked number 16 in Stephen Smale's list of Mathematical Problems for the 21st Century, see [22].

1.1 *Jacques Dixmier and Nguyen Huu Anh conjectures*

Our contribution: In 1997 the first author tried to find a counter-example for the Dixmier conjecture for smallest valid parameters, and it turned out that Dixmier conjecture is still valid for that case, see details in Section 1.2. Years later, in 2008 Hoang V. Dinh [17] extended searching for a counter-example for a next pair of valid parameters, but the stubborn Dixmier conjecture still resists to failing!

1.1.1 Weyl algebra A_n and its canonical representation

Weyl algebra $A_n(k)$ or just A_n over a field k is an algebra being determined by $2n$ generators $p_1, q_1, \cdots, p_n, q_n$ such that the following conditions are hold:

- the Lie product $[p_i, q_i] = p_i q_i - q_i p_i = 1$, for $i = 1, 2, \cdots, n$; and
- $[p_i, q_j] = [p_i, p_j] = [q_i, p_j] = 0$ if $i \neq j$.

The canonical representation of A_n: Let $E = k[X_1, X_2, \cdots, X_n]$ be a vector space defined on the field k and n variates $X_1, X_2, \cdots, X_n$. Denote by $P_i = \dfrac{\partial}{\partial X_i}$ the partial differential morphism with respect to X_i, and Q_i the multiplicative morphism by X_i, then clearly $P_i, Q_i \in \mathbf{End}(E)$- the set of all endomorphisms on the space E. Moreover, we easily see that they satisfy constraints

$$[P_i, Q_i] = 1, \quad \text{and} \quad [P_i, Q_j] = [P_i, P_j] = [Q_i, Q_j] = 0, \quad \text{for all } i \neq j.$$

For instance, the fact that $[P_i, Q_i] = 1$ (the identity on E) is true because

$$\begin{aligned}[P_i, Q_i](f) &= (\frac{\partial}{\partial X_i}\ X_i - X_i\ \frac{\partial}{\partial X_i})(f) = (\frac{\partial}{\partial X_i})(X_i f) - X_i\ (\frac{\partial}{\partial X_i})(f) \\ &= f + X_i(\frac{\partial}{\partial X_i})(f) - X_i\ (\frac{\partial}{\partial X_i})(f) = f.\end{aligned}$$

Therefore, there exists an morphism ρ from A_n to $\mathbf{End}(E)$ so that $\rho(p_i) = P_i$, $\rho(q_i) = Q_i, \forall i$. As a result, elements $p_1^{i1} q_1^{j1} \cdots p_n^{in} q_n^{jn}$ make a basis of A_n as a vector space, and ρ is an injection. We write

$$A_n = k\left[X_1, X_2, \cdots, X_n,\ \frac{\partial}{\partial X_1}, \cdots, \frac{\partial}{\partial X_n}\right]$$

with

$$[X_i, \frac{\partial}{\partial X_j}] = \delta_{ij} \text{ (the Knonecker notation).}$$

[0]Hermann KH. Weyl (1885 - 1955) was a German mathematician, theoretical physicist and philosopher; one of the most influential mathematicians of the 20th century, and an important member of the Institute for Advanced Study during its early years.

The representation ρ of the algebra A_n in $\mathbf{End}(E)$ is called the canonical representation of A_n. When $n = 1$ we write $p_1 = p, q_1 = q$, the Weyl algebra A_1 is determined by two generators p, q where the Lie product $[p, q] = 1$.

The algebra A_1 plays an important role in harmonic analysis over unimodule Lie groups having integrable square representation. Jacques Dixmier (1968) investigated the algebraic representation of A_1 and related the theory with the automorphism group $\mathrm{Aut}(A_1)$ of A_1.

1.1.2 The Dixmier conjecture

From now on, for convenience we write x, y instead of p, q, so, as discussed above we can identify $A_1 = A_1(k)$ with the non-commutative polynomial ring $k_D[x, y]$ (D means the Lie bracket) in which the Lie product $[x, y] = xy - yx = 1$. Let $\tau : A_1 \longrightarrow A_1$ be an injective morphism. Then

$$\begin{cases} \tau(x) = P,\ \tau(y) = Q, \\ [P, Q] = [\tau(x),\ \tau(y)] = \tau([x, y]) = \tau(1) = 1 \dots \end{cases} \qquad (\alpha)$$

P, Q obviously are polynomials in the non-commutative ring $k_D[x, y]$, with degrees $p = \deg(P)$, $q = \deg(Q)$, and $\{P,\ Q\}$ generates $\tau(A_1)$. Furthermore it is a basis of $\tau(A_1)$ (i.e. $\{P^i\ Q^j\}$ is a basis of $\tau(A_1)$ as a vector space). As a result, the injective morphism τ can be determined by two generators $P,\ Q \in A_1$ such that $[P,\ Q] = 1$. Note that $P = P(x, y),\ Q = Q(x, y)$ are non-commutative polynomials with respect to two variates x, y satisfying $[x, y] = xy - yx = 1$.

Dixmier made his well known conjecture in 1968 as follows: **Every injective morphism τ from the Weyl algebra A_1 to itself is an isomorphism**.

A solution of the Dixmier conjecture - although only touching the algebra A_1- could meaningfully help studying the automorphism group $\mathrm{Aut}(A_n)$ of the general algebra A_n, a problem closely related to the enveloping algebra of a Heisenberg group. Up to the 1996 there was no proof of correctness of this conjecture, hence we tried to find a counter-example. A possible counter-example is just one specific injective morphism τ such that $\tau(A_1) \subset A_1$, meaning it is not surjective. Equivalently, tt means a pair of P, Q that do not generate A_1, or just "x, y can not be represented as polynomials of P and Q"!

<u>The automorphism group $\mathrm{Aut}(k[x, y])$ of $k[x, y]$</u>: Assume that the ground field k is algebraically closed with characteristic 0, let commutative polynomials

$$f = f(x, y),\ g = g(x, y) \in k[x, y]$$

[0]The discrete Heisenberg group is just a certain group of 3×3 upper triangular matrices, but the continuous one arises in the description of one-dimensional quantum mechanical systems, especially in the context of the Stonevon Neumann theorem.

in two variables x, y. Denote by $\theta: \ k[x,y] \longrightarrow k[x,y]$ a polynomial morphism such that $\theta(x) = f, \ \theta(y) = g$, that means θ is determined by the pair of f, g. Put $J(f,g) := \partial(f,g)$ the Jacobian matrix of θ, then the determinant of $J(f,g)$

$$D(\theta) := \det(J(f,g)) = \partial(f, \ g)/\partial(x,y) = \frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial g}{\partial x}\frac{\partial f}{\partial y} \tag{1}$$

is a polynomial with degree $\deg(D(\theta)) = \deg(f) + \deg(g) - 2$ over k.

If θ is isomorphism then there exists the inverse θ^{-1} and we have

$$D(\theta \circ \theta^{-1}) = D(\theta).D(\theta^{-1}) = D(\mathbf{Id}) = 1.$$

Therefore $D(\theta)$ is an invertible element in $k[x,y]$.

Now if we call $\mathrm{Aut}(k[x,y])$ the group of all isomorphisms of $k[x,y]$, then by the above reasoning this group is defined by such pairs of polynomials f, g.

1.1.3 Nguyen Huu Anh's conjecture

In the process of disproving the Dixmier conjecture, Nguyen Huu Anh proposed in 1997 a stronger conjecture, formulated as follows.

Given two polynomials $P, \ Q \in A_(k) = k_D[x,y]$**, with degrees** p, q**,** $p \geq 2$ **or** $q \geq 2$**,** $\gcd(p,q) < \min(p,q)$**, and also** $[P, \ Q] = c, \ c \in k$**; where** k **is an algebraically closed field.**

There exists a polynomial $u = u(x,y) \in A_(k)$ **with degree** $d = \gcd(p,q)$ **and two univariate polynomials** F, G **with respect to** u **such that**

$$\begin{cases} P(x,y) = F(u(x,y)) \ \textbf{and} \\ Q(x,y) = G(u(x,y)). \end{cases}$$

Theorem 1.1. *If Nguyen Huu Anh's conjecture would true then the Dixmier conjecture is true as well.*

We need the lemma below for proving this theorem.

Lemma 1.1. *Let $f = f(x,y), \ g = g(x,y)$ be commutative polynomials in $k[x,y]$. If f is homogeneous with degree $p \geq 1$, g is homogeneous with degree $q \geq 1$, and moreover, the determinant $D(f,g) := \det(J(f,g)) \equiv 0$ then there exists a homogeneous polynomial $u_0 \in k[x,y]$ of degree $d = \gcd(p,q)$ so that*

$$f = c_1 \ u_0^{p/d}, \quad g = c_2 \ u_0^{q/d}.$$

Proof. (Theorem 1.1) Denote by τ an arbitrary injection from A_1 to itself. We prove that τ is surjective, by induction on $\max\{p,q\}$, where p, q the degrees of generating polynomials P, Q of τ. It means we check P, Q generate A_1, with the above assumption (α) that $\tau(x) = P,\ \tau(y) = Q$, and $[P, Q] = 1$.

Case of $p = q = 1$: P, Q have degree 1, so we write $P = ax+by$, $Q = cx+dy$ with $D(\tau) = ad - bc = [P, Q] = 1 \neq 0$ (see Equation 1). Hence, x, y are represented as polynomials of degree 1 of P, Q, so P, Q generate the algebra A_1.

Case of $p > 1$ or $q > 1$: Obviously $p + q \geq 3$. We must have $p \geq 1$ or $q \geq 2$. Suppose $\gcd(p, q) < \min(p, q)$. Since $[P, Q] = 1$ and assumptions of Anh's conjecture are satisfied we imply that there exists a polynomial $u(x, y) \in A_1(k)$ such that

$$\begin{cases} P(x, y) = F(u) \text{ and} \\ Q(x, y) = G(u), \end{cases}$$

as a result $[P, Q] = [F(u), G(u)] = 0$ (contradiction)! Therefore, we must have $\gcd(p, q) = \min(p, q)$, and can take $p < q$, then p is a divisor of q, and so $d = \gcd(p, q) = p$, $p/d = 1$. Now let f_p, g_q be respectively the homogeneous components of degree p and q of polynomials P, Q. Then, as Equation 1 suggests, polynomial

$$E = \det(J(f_p,\ g_q)) = \partial(f_p,\ g_q)/\partial(x, y)$$

is the homogeneous components of highest degree $p + q - 2$ of polynomial $D(\tau) = [P, Q] = 1 \neq 0$. Because $p + q \geq 3$ or $p + q - 2 \geq 1$, besides $D(\tau) \neq 0$ so we get $E = 0$. The pair of f_p, g_q fulfills Lemma 1.1, so we can firstly find a homogeneous polynomial $u_0 \in k[x, y]$ of degree $d = \gcd(p, q)$.

Secondly we employ the concept of commutative graded algebra on the algebraically closed field k (see Man Nguyen 1997, [36, Part C, Chapter 3]), to finally reduce polynomials f_p, g_q further to

$$\begin{aligned} f_p &= u_0^{p/d} \mod \deg \leq p - 1 = u_0 \quad \mod \deg \leq p - 1, \\ g_q &= u_0^{q/d} \mod \deg \leq q - 1. \end{aligned}$$

With this result, we write the polynomial

$$\begin{aligned} Q^* := Q - P^{q/p} &= (g_q + \ldots) - (f_p + \ldots)^{q/p} = (g_q + \ldots) - (u_0 + \ldots)^{q/p} \\ &= (u_0^{q/d} + \ldots) - u_0^{q/d} + \ldots \end{aligned}$$

consequently $\deg(Q - P^{q/p}) = \deg(Q^*) < \deg(Q) = q$, and so

$$1 = [P, Q] = [P, Q^* + P^{q/p}] = [P, Q^*].$$

Because of $\deg(Q^*) < \deg(Q)$ we see that

$$\max(\deg(P), \deg(Q^*)) < \max(\deg(P), \deg(Q)) = q.$$

Besides, $[P, Q^*] = 1$, hence by inductive assumption, the pair of P, Q^* generates the Weyl algebra A_1. Finally, since $Q = Q^* + P^{q/p}$ and P, Q^* generates A_1, we conclude P, Q generates A_1 as well! $\square$

1.2 *Finding counter-examples of the two conjectures*

1.2.1 Reduce finding a counter-example to a polynomial problem

We now design efficiently computational procedures allowing us to deal with computation on multi-layer Lie brackets, specifically to define non-commutative multiplication between x, y so that $[x, y] = 1$. These ensure transforming $k[x, y]$ to the non-commutative ring $k_D[x, y]$, viewed as the Weyl algebra $A_1(k)$. To the first conjecture, by algebraic transformations, non-trival cases lead us to searching for a counter-example that fulfills:

I. Degrees $p = \deg(P) > 1$ or $q = \deg(Q) > 1$ (so $p + q \geq 3$), $[P,\ Q] = 1$ and P, Q do not generate A_1;

II. p, q satisfy one of the two conditions a) $\gcd(p, q) = \min(p, q)$, or b) $\gcd(p, q) <\ \min(p, q)$ [p is not a divisor of q and q is not a divisor of p].

It turns out that if a counter-example P, Q would exist in II.a) case then there exists a pair of P_1, Q_1 in II.b) case, kindly see [36, Chapter 3] for details. Hence, finding a counter-example is reduced to solving the following problem:

Find two polynomials $P, Q \in A_(k) = k_D[x, y]$ with degrees p, q ($p \geq 2$ or $q \geq 2$), k is an algebraically closed field, such that $\gcd(p, q) < \min(p, q)$, and satisfying $[P,\ Q] = c,\ c \in k$.

Dixmier proved his famous conjecture correct for the case of $\gcd(p, q) = 1$ in 1966. No one checks the case of $\gcd(p, q) = 2$, and the computational load is huge for the case of $\gcd(p, q) = 4$. Therefore from 1996, we have tried the case of $\gcd(p, q) = 3$ for which the smallest degrees are $p = 6$ and $q = 9$.

1.2.2 Computational setting for the two conjectures

Our key identities for computation are, firstly the non-commutative product of a monomial $f = a\ x^m.y^n$ with a polynomial $g = \sum_{ij} b_{ij} x^i.y^j$, that is

$$f.g = a\ x^m.y^{n-1}[g.y - diff_x(g)];$$

and secondly the recursive formula below:

$$[x^m,\ y^n] = m.n.\ x^{m-1}\ y^{n-1} - m\left[\sum_{0}^{n-1} [x^{m-1},\ y^k]\ .y^{n-k-1}\right]. \tag{2}$$

The Lie product $[P,\ Q]$ is a nonlinear polynomial with degree $6 + 9 - 2 = 13$, hence the condition $[P,\ Q] = c,\ c \in k$ gives us a system of nonlinear polynomial equations with $14 + 13 + \cdots 2 + 1 = 105$ equations in terms of 83 unknowns (28 unknowns from P and 55 unknowns from Q).

Using the concept of commutative graded algebra (see [36, Part C, Chapter 3]), when the ground field k is algebraically closed, we can algebraically simplify patterns of polynomials P, Q further to

$$P = u_0^3 + P_1, \ \deg(P_1) \leq 5, \tag{3}$$

$$Q = u_0^2 + Q_1, \ \deg(Q_1) \leq 8, \tag{4}$$

where u_0 is a homogeneous polynomial of degree $d = \gcd(p, q) = \gcd(6, 9) = 3$.

When homogeneous polynomials appear in computation we need to know what is the non-commutative $y^n.x^m$ and the non-commutative product of $x^i.\ y^n$ with $x^m.\ y^j$. Again by induction, when $n > m$ we got

$$y^n.\ x^m = \sum_k (-1)^k.C_m^k A_n^n \ x^{m-k} y^{n-k} \tag{5}$$

and

$$x^i.y^n.\ x^m.y^j = \sum_k (-1)^k.C_m^k A_n^n \ x^{i+m-k} y^{j+n-k}. \tag{6}$$

Then if set $i + n = p, \ j + m = q$ and call $f = c\ x^i.y^n$ be the monomial with highest degree in the homogeneous polynomial of degree p of P, call $g = d\, x^m.y^j$ be the monomial with highest degree in the homogeneous polynomial of degree q of Q, then the Lie bracket

$$[f, g] = cd\ [x^i.y^n, x^m.y^j] = cd\ (x^i.\underline{y^n.\ x^m}\ .y^j - x^m.\underline{y^j.\ x^i}\ .y^n)$$

gives rise the fact $\deg([f, g]) \leq p + q - 2$.

- These formulas, in general allow us to find a suitable homogeneous polynomial P_i of degree $i \leq p$ of P and a suitable homogeneous polynomial Q_j of degree $j \leq q$ of Q to build up a right partial system of nonlinear equations of the whole system $[P, Q] = c$. This constraint of

$$\deg(P_i, Q_j) \leq i + j - 2 \tag{7}$$

  is crucially necessary for step-wise truncating the huge system $[P, Q] = c$. Specifically, when $i = 6, j = 9$ (max degree monomials of P, Q to start with) we know the max degree of the Lie product $[P, Q]$ is $i + j - 2 = 13$, as seen above.

- The major idea of our step-wise truncating algorithm is exploiting Condition (7) at each truncation. We firstly start with P_6, the homogeneous polynomial of degree 6 of P and Q_9, the homogeneous polynomial of degree 9 of Q, compute the Lie $[P_6, Q_9]$ and add to $[P, Q]$; secondly at any iteration k form P_i and Q_j, generate $[P_i, Q_j]$ in order to append to the system $[P, Q] = c \mod \deg < k$ (β); and finally extract coefficients and find a Groebner basis of (β). Kindly see details in [ManNguyen, [36]].

1.3 *Summary*

By exploiting the power of certain computer algebra systems (such as Maple [16], Mathematical [45], the specialized polynomial system Singular [43] or the package GAP [specializing in Group, Algorithm and Programming, [23]]) by which solutions of nonlinear polynomial equations are effectively computed, we obtained the following conclusions.

Jacques Dixmier's conjecture

Polynomials P, Q with degrees $p = 6,\ q = 9$ satisfying $[P,\ Q] = c$ always imply $c = 0$. It means there is no counter-example for Dixmier's conjecture in the case of $\deg(P) = 6$ and $\deg(Q) = 9$, see [Man Nguyen [36]]. The latest work in 2008 by Hoang V. Dinh [17] strongly confirmed that Dixmier's conjecture is true for polynomials P, Q with degrees $p = 6,\ q = 9$ satisfying the condition $[P,\ Q] = c, c \in k$, where k is an algebraically closed field. With other parameters as $p = 8, q = 12$ no further work have been found, to the best of our knowledge.

Nguyen Huu Anh's conjecture

However, Hoang Van Dinh [17] found a counter-example for Nguyen Huu Anh's conjecture for all possible options of the homogeneous polynomial u_0 of degree $\gcd(p, q)$, given in Equation 3. The argument is based on seeking for the triple of $[F, G, u]$ such that $P(x, y) = F(u(x, y))$ and $Q(x, y) = G(u(x, y))$, where the pair of P, Q already satisfy Dixmier's conjecture, i.e. $[P,\ Q] = 0$.

If for such pair of P, Q, there is a triple of $[F, G, u]$ then Nguyen Huu Anh's conjecture is incorrect. Polynomials P, Q, u are $u = x^3 + y^2$, $P = u^2 + 2x$, and $Q = u^3 + 3ux + 3y$. What we must do to make sure that P, Q, u build up a counter-example is just checking $[P, Q] = 0$ (see [17, Part 3, Chapter 4]).

Our first conjecture

- Dixmier conjecture is still valid for $\gcd(p, q) = 3 < \min(p, q)$, as parameters $p = 9, q = 12$; or $\gcd(p, q) = 4 < \min(p, q)$, as parameters $p = 8, q = 12$?
- For the general case of $\gcd(p, q) < \min(p, q)$ new concepts (such as *slope of generators*) should be proposed before we prove/disprove this conjecture. E.g., if the case of $p = 8, q = 12$ would be true, then we may think that Dixmier conjecture is true for any case with the *slope* $s = p/q$ is constant, such as the cases of $(p, q) = (6, 9), (8, 12)$ give $p/q = 6/9 = 8/12 = 2/3$.

We have illustrated how useful the computer algebra approach is when solving theoretic problems of pure mathematics in the last part. In the subsequent sections, we switch to more practical sciences and engineering in which various data sets are available and algebraic thinking still plays an essential role.

2 Constructing designs for quality control

Quality is a broad concept, often it refers to a **grade of excellence**, literally means consistently meeting standards appropriate for a specific product or service. There are another two key views, saying **quality is fitness for use** [by Joseph M. Juran, a pioneer in *Total Quality Management*], and **quality is inversely proportional to variability** [by Douglas Montgomery, Arizona University]. Thus, if we follow the last definition, then *quality improvement* - in various industries and services- is just the reduction of variability in processes and products.

2.1 *Statistical Quality Control - Overview and Methodology*

Statistical Quality Control (SQC) - and Quality Engineering, its broader domain- among other things means to mathematically design goods/products from which we could monitor and control *quality characteristics* of those products before actually manufacture them in factories. SQC also means using the *prototypes of products* (being mathematically designed beforehand) to conduct life testing from which we are able to measure responses, collect numerical data, then analyze and control theirs quality characteristics **before** actual mass manufacturing them on assembly lines. The first phase uses designed experiments (DOE or *Experimental Designs*) - a sequence of trials or tests performed under controlled conditions which produces measurable outcomes; and in the second phase we could employ various popular *control charts* (as Shewhart types), Six-Sigma methodology and DMAIC (Define, Measure, Analyze, Improve, and Control) process. At least two major reasons for studying are:

1. Industry and service sectors always need cutting-edge ideas/outcomes [the richer countries the higher demand of quality R & D].

2. The problem comes down to a matter of cost: conducting R & D activities costs money; but this spending is worthy to make in a pre-production phase (i.e. **offline production-** meaning not implemented on assembly lines yet) of industrial manufacturing, or broader in the new 4.0 science and technology revolution.

The concept of "Organization's quality" with the focus on management was proposed since the 1980s. The company-wide quality approach emphasizes

[0] Joseph Moses Juran (1904 2008) was a Romanian-born American engineer and management consultant. He was an evangelist for quality and quality management.

on *i) Competence* such as knowledge, skills, experience and qualifications; *ii) Hard elements* such as job management, adequate processes and performance criteria; and *iii) Soft elements*, such as personnel integrity, confidence, organizational culture, and team spirit. **The quality of the outputs is at risk** if any of these aspects is deficient in any way.

Regarding specifically Quality Engineering, **Malcolm Bridge**, a former U.S. Secretary of Commerce, said in the article *Designing for productivity* (**Design News**, Vol. 38. No. 13., 1982) about few most practical demands for a competitive economy that (i) for top managers, the challenge is to create an organizational environment that fosters creativity, productivity and quality consciousness; that (ii) 40 percent of all costs in getting a product to the marketplace are in the design cycle; and last but not least, (iii) top management must better emphasize prevention than correction.

Prevention means conducting statistically designed experiments in the design cycle or off-line manufacturing. More general we have discussed Joseph Juran's Total Quality Management (TQM) methodology above, and we view DOE belongs to this broader category.

Total Quality Management (TQM) and Statistical Process Control

In Total Quality Management we are interested in the following activities.

a/ *Quality Planning*: the development of strategic activities designed to improve the quality of a product. The planning will include both statistical methods and management activities.

b/ *Quality Assurance*: a system of activities whose purpose is to provide an assurance that the overall quality control is in fact being done effectively. It includes the regulation of

the quality of raw materials, assemblies, products and components;

the services related to production; and

the processes of management and inspection.

c/ *Quality Control*: a few concepts broadly accepted nowadays, including

1. the operational techniques, activities and their uses sustaining a quality of product or service that will satisfy given needs,

[0]The U.S. Congress established in 1987 the **Malcolm Baldrige National Quality Award** (MBNQA), an award to raise awareness of quality management and recognize U.S. companies that have implemented successful quality management systems. Awards are presented annually by the President of the United States to organizations that demonstrate quality and performance excellence, in six categories: manufacturing, service, small business, education, healthcare and nonprofit.

2. the application of *statistical principles and techniques* in all stages of design, production, maintenance and service, directed toward the economic satisfaction of demand [by Deming (1971)].

d/ *Quality Improvement*: the improvement process, measures of process effectiveness, employs methods of DOE (Design of Experiments, also called Experimental Designs)...

e/ *Statistical Process Control*- SPC: can be considered as SQC applied to a process, or to a product resulting from a process. SPC is the totality of all process activities directed at improving process consistency through detecting changes in measured characteristics, identifying causes of changes, and preventing recurrence of those causes.

Large firms have applied major principles of SQC in manufacturing high-tech products, for instance in dairy industry at Campina - Thailand ([10], originally a Dutch dairy firm), in telecommunication at Samsung [40], AT & T, or in automobile sector at GE, Ford, Toyota, Audi or BMW [46].

2.2 *Experimental Designs with Computer Algebra in SQC*

The study of computer algebra in Quality Engineering historically began from the European Algebraic Statisticians, focused in Experimental Designs, and its core topic is *Factorial Experimental Design-FED or Factorial Design*.

We first recall some key terms of Factorial Design. In SQC, when causes of a response (or components of a product) all receive only discrete values (choices or levels) then those causes are said to be factors. Factorial designs is a very useful solution for our *industrial manufacturing* problems. We use regression models to capture relationships between random variables into a response of interest, determine the magnitude of the relationships between variables in that response, and make predictions based on those statistical models.

Both SQC and SPC intensively use factorial designs and various regressions to eliminate uncertainty of product's quality; see specific industries currently employing SPC, at [24] and [41].

2.2.1 What really are factorial designing experiments and why them?

Formally, for a natural number $d > 1$, we fix d finite sets $Q_1, Q_2, \ldots, Q_d$ called *factors*. The elements of a factor are called its *levels*. The (*full*) *factorial design* (also factorial experiment design- FED) with respect to these factors is the Cartesian product $D = Q_1 \times Q_2 \times \ldots \times Q_d$. FED help us in doing the followings: perform experiments to evaluate the effects the factors that could have on the characteristics of interest, and discover possible relationship among the factors, called *factor interactions* which could affect the characteristics.

Mathematically, the main aim of using FED (and similar structures of Experimental Designs) is to identify an unknown function

$$\phi : D \to \mathbb{Q},$$

a mathematical model of a quantity of interest (favor, usefulness, best-buy, quality ...) which has to be computed or optimized. When a firm's budget is limited, practically the firm's manager must accept using a subset F of D when investigating properties of a new product or service.

Definition 2.1. *A* fractional design *or* fraction *F of D is a subset consisting of elements of D (possibly with multiplicities). Put $r_i := |Q_i|$ be the number of levels of the ith factor. We say that F is* symmetric *if $r_1 = r_2 = \cdots = r_d$, otherwise F is* mixed.

Moreover, F is said to be strength t orthogonal array *(*OA*) or t-balanced if, for each choice of t coordinates (columns) from F, each combination of coordinate values from those columns occurs equally often; here t is a natural number. If some of r_i are identical we can group them in distinct level s_i and write* $\mathrm{OA}(N;\ s_1^{a_1} \cdots s_k^{a_k}; t)$ *where $a_1 + a_2 + \ldots + a_k = d$ and N is the runsize.*

The structure of orthogonal array even has more useful properties in statistical optimization and industrial statistics. Specifically, strength 3 OAs permit estimation of all the main effects of the experimental factors free from confounding with two-factor interactions. Strength 4 OAs furthermore, allow us to theoretically separate all two-factor interactions during the analysis of data obtained from experimentation. We want to find such designs, investigate it in practice, specifically interested in:

a) Constructing and/or designing: to learn how to construct those experiments, given the scope of expected commodities and the parameters of components;

(b) Exploring and selecting: to investigate some design characteristics (proposed by researchers) to choose good designs. For instance, in factorial designs we learn how to detect interactions between factors; if they exist, calculate how strongly they could affect on outcomes; finally

(c) Implementing, analyzing & consulting: study how to use (i.e., conduct experiments in applications, measure outcomes, analyze data obtained, and consult clients what they should do).

The goal is to use these new understanding to improve product, to answer questions such as:

1. What are the key factors in a process?

2. At what settings would the process deliver *acceptable performance?*

3. What are the *main interaction effects* in the process?

4. What settings would bring about *less variation* in the output?

2.2.2 Important steps in designing experiments for R & D

1. *State objective*: write a *mission statement for the project*; as in household furniture production;

2. *Choose response*: it is about consultation, have to ask clients what they want know, or ask yourself; focus on the nominal-the-best responses;

3. *Perform pre-experiment data analysis*?

4. *Choose factors and levels*: you have to use flowchart to represent the process or system, use cause-effect diagram to list the potential factors that may impact the response;

5. *Select experimental plan* (if available, otherwise have to compute?)

6. *Perform the experiment* (in lab or in real industrial settings)

7. *Analyze the data*

8. *Draw conclusions* and make recommendations.

Experimental Deigns in general, fractional designs in specific, and other data analytics tools are intensively employed in the above steps, except Step 6.

2.3 *Illustration of theses procedural steps*

We illustratively consider a particular fractional design here and a cost optimal problem in furniture industry, with 8 factors of interest. Let N be the number of experimental runs in the experiment; each run will be assigned to a particular *combination of factor levels.* Let $M := 6 \cdot 4^2 \cdot 2^5$ denote the number of possible level combinations of the factors A, B, C, D, E, F, G and H.

The goal: we study only one response Y, the *wood furniture hardness.*

Various targets: we distinguish three terms of main effects, two-factor interactions, and higher-order interactions.

The method: To maximize the hardness of new products, we study the *combined influence of the factors* using *linear regression models*. If we study only the main effects then such a linear model takes the form

$$Y = \theta_0 + \sum_{i=1}^{5} \theta_{A_i} a^i + \sum_{j=1}^{3} \theta_{B_j} b^j + \sum_{l=1}^{3} \theta_{C_l} c^l + \theta_D d + \theta_E e + \ldots + \theta_H h + \epsilon,$$

where ϵ is a random error term, $a = 0, 1, 2, 3, 4, 5$; $b, c = 0, 1, 2, 3$; besides $d, e, f, g, h = 0$ or 1, and the parameters θ_* are the *regression coefficients*.

In a dreamed situation we need a budget to carry out $M := 6 \cdot 4^2 \cdot 2^5 = 3072$ experiments, to estimate all effects on the quality (hardness of furniture). But if scale down our study to measuring only main effects and a few two-interactions then a design with 96 runs (experiments) would be suitable for practical usages. If we want to know all two-interactions and main effects we need at least

$$1 + \sum_{i=1}^{8} (r_i - 1) + \sum_{\substack{i,j=1 \\ i<j}}^{8} (r_i - 1)(r_j - 1) = 121 \text{ runs.}$$

Few essential questions are raised now: **Why 96 runs? Would any suitable design with type $6 \cdot 4^2 \cdot 2^5$ given in Table 1 does exist for our purpose**?

Table 1: A workable factorial plan with type $6 \cdot 4^2 \cdot 2^5$

			Level					
Factor	**Description**	r_i	**0**	**1**	**2**	**3**	**4**	**5**
(A)	wood	6	pine	oak	birch	chestnut	poplar	walnut
(B)	glue	4	a (least)	b	c	d (most)		
(C)	moisture content	4	10%	20%	30%	40%		
(D)	process time	2	1 h	2h				
(E)	pretreatment	2	no	yes				
(F)	indenting of wood	2	no	yes				
(G)	pressure	2	1 pas	10 pas				
(H)	hardening condition	2	no	yes				

A clearly immediate answer then is: No, in general! You have to find them, the so-called *orthogonal arrays* of strength $t \geq 3$, or a t-balanced fraction. Industrialists say that such designs must be firstly determined by its runsize N, via the divisibility and the Rao bound [38]. Generally the Rao bound gives an lower bound on the runsize N in terms of the factor's levels $r_1 > r_2 \ldots > r_d$. When $t = 2$, N is bounded below by $N \geq 1 + \sum_1^d (r_i - 1) = 1 + 5 + 2.3 + 5 = 17$. When t is odd, in general we have

$$N \geq r_1 \sum_{j=0}^{(t-1)/2} \sum_{|K|=j, K \subset \{2,\ldots,d\}} \prod_{i \in K} (r_i - 1).$$

If $t = 3$, and use the design type $6 \cdot 4^2 \cdot 2^5$ then $N \geq 6[2.(4-1) + 5.(2-1)] = 66$, coupling with divisibility give us the runsize $N = 96$. Our problem now is that the existence of an OA$(96;\ 6 \cdot 4^2 \cdot 2^5; 3)$ is still in questionable!

We will present two distinct mathematical approaches for computing such designs in the next two parts, about using computational algebraic geometry in Section 2.4 and non-abelian group computation in Section 2.5. Kindly see full treaments in [Man Nguyen, 2005 [37]] and [Man Nguyen, 2011 [28]].

2.4 *Computer-algebraic construction of mixed orthogonal arrays*

Linear-algebraic method for design construction

Suppose $F \subseteq D$ be a fraction with d factors, considered $D \subset \mathcal{F}^d$.

We represent the factors $Q_1, \ldots, Q_d$ by variables $x_1, \ldots, x_d$.

- Let $J = \mathrm{I}(F)$ and let $V = P/J$. Then

$$E = \mathrm{Est}(F) = \{h_1, \ldots, h_\mu\}$$

  is a set of monomials such that $\overline{E} = \{\overline{h}_1, \ldots, \overline{h}_d\}$ is a basis for P/J as a $\mathcal{F}$-vector space.

- Let $M = \boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_d^{\alpha_d}$, the M's left action induces an endomorphism of V.

- Let L_M be the matrix of this action with respect to the basis $\overline{E}$.

The matrices $L_{x_1}, \ldots, L_{x_d}$ are called the *elementary multiplication matrices.*

Key results for the existence of designs

Theorem 2.1. *Suppose that F has no repeated runs. The characteristic polynomial of L_M is*

$$\prod_{\boldsymbol{p}=(p_1,\ldots,p_d)\in F} (X - p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_d^{\alpha_d}).$$

The trace of L_M is $\sum_{\boldsymbol{p}\in F} p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_d^{\alpha_d}$. We observe that:

- If F is a 1-balanced fraction, the size of F must be a multiple of the number of levels of each of the factors which form F.

- If F is a 2-balanced fraction, then the size of F must be a multiple of the products of each pair of levels, and so on for any strength $t > 2$.

To appreciate the beauty and power of computer algebra we recall here a proof of the theorem.

Proof. Suppose F have N runs, and denote $\boldsymbol{p} = (p_1, \ldots, p_d)$ for a run in F. The vanishing ideal of $\boldsymbol{p}$ is

$$\mathrm{I}(\boldsymbol{p}) = \big\langle \{x_1 - p_1, \ldots, x_d - p_d\} \big\rangle. \tag{8}$$

The vanishing ideal of the fraction F is

$$\mathrm{I}(F) = \bigcap_{\boldsymbol{p}\in F} \mathrm{I}(\boldsymbol{p}).$$

The Chinese Remainder Theorem for ideals (see [42, Corollary 2.2]) gives us the decomposition:

$$P/\,\mathrm{I}(F) = \bigoplus_{\boldsymbol{p}\in F} P/\,\mathrm{I}(\boldsymbol{p}). \tag{9}$$

Consider a run $\boldsymbol{p} = (p_1, \ldots, p_d)$ as a variety. Each $P/\,\mathrm{I}(\boldsymbol{p})$ is isomorphic to $\mathcal{F}[\boldsymbol{p}] = \mathcal{F}$ (see [20, Definition 19], e.g. for the definition of $\mathcal{F}[\boldsymbol{p}]$), so $P/\,\mathrm{I}(\boldsymbol{p})$ is a 1-dimensional sub-algebra of the quotient algebra $P/\,\mathrm{I}(F)$. Hence, $P/\,\mathrm{I}(F)$ is isomorphic to the algebra $\mathcal{F}^d$.

From Equation (8), since $x_i - p_i \in \mathrm{I}(\boldsymbol{p})$, so we have $x_i^{\alpha_i} = p_i^{\alpha_i}$ in $P/\,\mathrm{I}(\boldsymbol{p})$, for all $i = 1, \ldots, d$. As a result, for each $v \in P/\,\mathrm{I}(\boldsymbol{p})$:

$$(x_i^{\alpha_i} - p_i^{\alpha_i})\, v = 0, \text{ so } \quad L_{x_i}^{\alpha_i}(v) = L_{{x_i}^{\alpha_i}}(v) = x_i^{\alpha_i}.v = p_i^{\alpha_i} v, \text{ for } i = 1, \ldots, d,$$

that means v is an eigenvector of the matrix $L_{x_i}^{\alpha_i} = (L_{x_i})^{\alpha_i}$ with eigenvalue $p_i^{\alpha_i}$. Hence p_i is an eigenvalue of the matrix L_{x_i} $(i = 1, 2, \cdots, d)$. If we choose a term $M = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_d^{\alpha_d}$, then the left multiplication matrix by M is given by

$$L_M = L_{{x_1}^{\alpha_1} \ldots x_d^{\alpha_d}} = L_{x_1}^{\alpha_1} \ldots L_{x_d}^{\alpha_d}, \text{ and } \quad L_M(v) = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_d^{\alpha_d} v.$$

Therefore, F consists of all vectors $\boldsymbol{p} = (p_1, \ldots, p_d)$ where v is some common eigenvector with eigenvalue p_i with respect to the matrix L_{x_i}. We conclude that v is an eigenvector of L_M with eigenvalue $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_d^{\alpha_d}$. In other words, the N subalgebras $P/\,\mathrm{I}(\boldsymbol{p})$ are N eigenspaces for L_M, with corresponding eigenvalues $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_d^{\alpha_d}$ for each run $\boldsymbol{p} = (p_1, \ldots, p_d)$. As a result, since L_M is an $N \times N$ matrix, the theorem is now proved. □

Corollary 2.1 (Using key result for a necessary condition)**.**

Let F be a t-balanced fraction of a design D in $\mathcal{F}^d$. Assume that factor x_i has levels $0, 1, \ldots, r_i - 1$.

(a) If $t \geq 1$ and $\alpha_i \in \{0, 1, \ldots, r_i - 1\}$, then the matrix $L_{{x_i}^{\alpha_i}}$ has trace

$$\frac{N}{r_i} \sum_{l=0}^{r_i - 1} l^{\alpha_i}.$$

In particular, L_{x_i} has trace $|F|(r_i - 1)/2$.

(b) If $t \geq 2$, $\alpha_i \in \{0, 1, \ldots, r_i - 1\}$ and $\alpha_j \in \{0, 1, \ldots, r_j - 1\}$, then $L_{{x_i}^{\alpha_i}{x_j}^{\alpha_j}}$ has trace

$$\frac{N}{r_i\ r_j} \sum_{l=0}^{r_i-1} l^{\alpha_i} \sum_{m=0}^{r_j-1} m^{\alpha_j}.$$

Proof. (See [29, Section 6]). For each factor i, the number $\lambda_i = |F|/r_i$ must be a positive integer. The fraction F can be decomposed into λ_i blocks $F_1, \ldots, F_{\lambda_i}$, each block has r_i runs such that their ith coordinates are $0, 1, \ldots r_i - 1$. Hence, Item (a) is proved, due to the fact

$$\sum_{\boldsymbol{p} \in F_l} p_i^{\alpha_i} = \sum_{m=0}^{r_i-1} m^{\alpha_i}, \text{ for every } l = 1, \ldots, \lambda_i.$$

By considering the designs combined by each pair of two factors i, j as a full design, applying a similar argument, we get (b). □

Our second conjecture

Let F be a fraction of a full design D in $\mathcal{F}^d$. Assume that factor x_i has levels $0, 1, \ldots, r_i - 1$.

For any natural $t \geq 2$, take parameters

$$\alpha_i \in \{0, 1, \ldots, r_i - 1\}, \alpha_j \in \{0, 1, \ldots, r_j - 1\}, etc,$$

and assume that

- the matrix $L_{{x_i}^{\alpha_i}}$ has trace

$$\frac{N}{r_i} \sum_{l=0}^{r_i-1} l^{\alpha_i},$$

- the matrix $L_{{x_i}^{\alpha_i}{x_j}^{\alpha_j}}$ has trace

$$\frac{N}{r_i\ r_j} \sum_{l=0}^{r_i-1} l^{\alpha_i} \sum_{m=0}^{r_j-1} m^{\alpha_j},$$

- the matrix $L_{{x_i}^{\alpha_i}{x_j}^{\alpha_j}{x_k}^{\alpha_k}}$ has trace

$$\frac{N}{r_i\ r_j\ r_k} \sum_{l=0}^{r_i-1} l^{\alpha_i} \sum_{m=0}^{r_j-1} m^{\alpha_j} \sum_{h=0}^{r_k-1} h^{\alpha_k}, \ldots$$

then F would be a t-balanced fraction. In other words the size $|F|$ would be a multiple of the products of each pair of levels, $|F|$ would also be a multiple of the products of each triple of levels, and so on.

2.5 *Computational Group Theory for mixed orthogonal array*

It is not immediately obvious how to define isomorphisms of a factorial design, given in Definition 2.1. In fact, there is more than one sensible definition that could be made. We give the definition that is most useful for our purposes in this section. The following notations will be used through out this section.

- Let N be a positive integer and $T := r_1 \cdot r_2 \cdots r_d$ be a design type, as Definition 2.1; equivalently we could group a_i factors with the same s_i levels in $T := s_1^{a_1} \cdot s_2^{a_2} \cdots s_m^{a_m}$, $s_i \neq s_j$ when $i \neq j$. Denote by $\mathbf{OA}(N;T)$ the set of all OAs with given type T and run size N.
- Set $U := \{(i,j,x) \mid i = 1,\ldots,N,\ j = 1,\ldots,d,\ x \in Q_j\}$, and call it the *underlying set* of $\mathbf{OA}(N;T)$. In other words, U consists of all possible triples of a row i, a column j, and an entry F_{ij} for any matrix $F \in \mathbf{OA}(N;T)$. The k-th column index set $J_k \subseteq \mathbb{N}_d := \{1, 2, \cdots, d\}$ precisely consists of column indices of factors having s_k levels, for each $k = 1, ..., m$.

2.5.1 Fraction transformations (or isomorphism) of arrays

We now define group actions (see Appendix B for basic concepts) on the set U:

- The *row permutation group* is $R := \mathrm{Sym}_N$. It acts via $\phi_R : R \to \mathrm{Sym}(U)$ defined by
$$(i,j,x)^{\phi_R(r)} = (i^r, j, x).$$
- The *column permutation group* is $C := \prod_{k=1}^{m} C_k$ where $C_k := \mathrm{Sym}(J_k)$. It acts via $\phi_C : C \to \mathrm{Sym}(U)$ defined by
$$(i,j,x)^{\phi_C(c)} = (i, j^c, x).$$
- The *level permutation group* is $L := \prod_{j=1}^{d} L_j$, here $L_j = \mathrm{Sym}_{r_j}$. This acts via $\phi_L : L \to \mathrm{Sym}(U)$ defined by
$$(i,j,x)^{\phi_L(l)} = (i, j, x^{l_j}),$$
where l_j is the projection of l onto L_j.

Definition 2.2. *The full group G of fraction transformations of U is defined as*
$$G := \phi_R(R) \quad \phi_C(C) \quad \phi_L(L) \leq \mathrm{Sym}(U). \tag{10}$$

Hence, we can now identify G with the wreath product $R \times (C \ltimes L)$ where
$$C \ltimes L = \prod_{k=1}^{m} \mathrm{Sym}_{s_k} \wr C_k.$$

Corollary 2.2. *We get the followings.*

- *The full group or the permutation group acting on the space* $\mathbf{OA}(N;T)$ *is*
$$G = R \times (C \ltimes L). \tag{11}$$

- *As a result, the order of* G *can be calculated from OA parameters, as*
$$|G| = N!\, a_1! \cdots a_m!\, (s_1!)^{a_1} \cdots (s_m!)^{a_m}.$$

The next concept plays a crucial role in the remaining parts.

Definition 2.3. *Let* F *and* F' *be in* $\mathbf{OA}(N;T)$*.*

- *An* isomorphism *from* F *to* F' *is* $g \in G$ *such that* $F^g = F'$*.*
- *The* automorphism group *of an orthogonal array* $F \in \mathbf{OA}(N;T)$ *is the normalizer of* F *in the group* G*, i.e.,* $\mathrm{Aut}(F) := \{g \in G \mid F^g = F\}$*.*
- *Any subgroup* $A \leq \mathrm{Aut}(F)$ *is called a group of automorphisms of* F*.*

We next formulate necessary algebraic conditions for extending a known orthogonal design $F = \mathrm{OA}(N; r_1 \cdots r_d; t)$ of strength t by a factor X to get a new design $[F|X]$ with the same strength.

2.5.2 An integer linear approach solves the extension problem

Assume $t = 3$, given an orthogonal array $F = \mathrm{OA}(N; r_1 \cdots r_d; 3)$ with columns $S_1, \ldots, S_d$, S_i has r_i levels $(i = 1, \ldots, d)$.

An s-level factor X is orthogonal to a pair of factors (S_i, S_j) of F, written $X \perp [S_i, S_j]$, if the frequency of all tuples $(a, b, x) \in [S_i, S_j, X]$ is $N/(r_i r_j s)$. Extending F by X means constructing an $\mathrm{OA}(N; r_1 \cdots r_d \cdot s; 3)$, denoted by $[F|X]$. By the definition of OAs, $[F|X]$ exists if and only if X is orthogonal to any pair of columns of F. We can find a set P of necessary constraints for the existence of array $[F|X]$ in terms of polynomials in the coordinate indeterminates of X, by the following rules.

(a) Calculate frequencies of 3-tuples, and locate positions of symbol pairs of (S_i, S_j).

(b) Set the sums of coordinate indeterminates of X (corresponding to these positions) equal to the product of those frequencies with the constant $0 + 1 + 2 + \ldots + s - 1 = \frac{s(s-1)}{2}$. The number of equations of P then is $\sum_{i \neq j}^{d} r_i r_j$, since each pair of (S_i, S_j) can be coded by a new factor with $r_i\, r_j$ levels. If $s = 2$, the constraints P are in fact the sufficient conditions for the existence of X.

For instance, let $F = \mathrm{OA}(16; 4 \cdot 2^2; 3) = [S_1|S_2|S_3]$ be a full design. By transformation rule (b), the sums of coordinates of X corresponding to the Y symbols and the Z symbols must equal a multiple of the appropriate frequencies. That means:

$$X \perp [S_1, S_2] \Leftrightarrow X \perp Y \Leftrightarrow x_1+x_2 = x_3+x_4 = \ldots = x_{15}+x_{16} = \lambda\cdot(0+1) = 1, \ldots,$$

and $X \perp [S_2, S_3] \Leftrightarrow X \perp Z \Leftrightarrow x_1+x_5+x_9+x_{13} = \ldots = x_4+x_8+x_{12}+x_{16} = \mu \cdot (0+1) = 2$. One solution of P is given in the last row of the matrix below:

$$\left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \end{array}\right]^T .$$

Generally, the set P of linear constraints with integer coefficients is described by the matrix equation $AX = b$, in which $A \in \mathrm{Mat}_{\mathrm{m_1,N}}(\mathbb{N})$,

$$X = (x_1, \ldots, x_N) \in \{0, 1, \ldots, s-1\}^N \subseteq \mathbb{N}^N \tag{12}$$

is a vector of unknowns, $b \in \mathbb{N}^{m_1}$, and $m_1 := \sum_{i\neq j}^d r_i r_j = |P|$. Since each orthogonal array is isomorphic to an array having the first row zero, we let $x_1 = 0$ throughout. By Gaussian elimination, we get the reduced system

$$M\, X = c, \tag{13}$$

where $M \in \mathrm{Mat}_{m,N}(\mathbb{Z})$, the set of all $m \times N$ ($m \leq m_1$) matrices with integral entries, $c \in \mathbb{Z}^m$, and the vector of unknowns $X = (0, x_2, \ldots, x_N) \in \mathbb{Z}^N$.

The extension $K := [F|X] = \mathrm{OA}(N; r_1 \cdots r_d \cdot s; t)$ clearly depends on solving the integer linear system (13) $M.X = c$ in terms of $X = (x_j) \in \{0, 1, \ldots, s-1\}^N$ for $j = 1, \ldots, N$. This approach is useful if a few constraints, structures or pruning techniques would be found and used to delete out some (not all) isomorphic vectors in each isomorphic class, and we then retain isomorph-free vectors. From that point, the search for all isomorph-free designs becomes feasible.

2.5.3 The row permutation group of F for computing X in $[F|X]$

Fix an array $F \in \mathbf{OA}(N; T; t)$, recall that $\mathrm{Aut}(F) := \{g \in G \mid F^g = F\}$, with G is the full group of isomorphisms, see Eq. (10). We first define the row permutation group of a fractional design F.

Let $g \in \mathrm{Aut}(F)$. Then g induces a permutation g_1 in the full group G_K of K, see Formula (11). Let g_R be the row permutation component of g, then g_R is also the row permutation component of g_1. Due to Definition 2.3, we have

Theorem 2.2. *For $g \in \mathrm{Aut}(F)$, g induces $g_1 \in G_K$ and generates the image K^{g_1} which is isomorphic to K.*

Proof. Formula (10) says any permutation g acting on F has the decomposition $g = g_R\, g_C\, g_S$ where g_C and g_S are the column and symbol permutations acting on F, respectively. Besides, the row permutation g_R induces a row permutation $g_1 \in G_K$, we furthermore have

$$K^{g_1} = [F|X]^{g_1} = [F^g|X^{g_R}] = [F|X^{g_R}] \tag{14}$$

since g already fixes F, and only g_R acts on the column X by moving its coordinates. As a result, $K^{g_1} = [F|X^{g_R}]$ is isomorphic to $K := [F|X]$. $\square$

Definition 2.4. *Let $H := \mathrm{Row}(\mathrm{Aut}(F))$ be the group of all row permutations g_R extracted from the group* $\mathrm{Aut}(F)$*. We call H the* row permutation group *of F.*

The *direct product* of H and τ is very useful for pruning later on, given by

$$\sigma := H \times \tau, \tag{15}$$

where $\tau := \mathrm{Sym}_s$, the symbol permutation group acting on the X's coordinates.

2.5.4 Row permutation subgroups for pruning solution spaces

It is now obvious that, by recursion, the process of building X can be brought back to strength 1 derived designs. We can effectively prune $\mathrm{Z}(P)$ from those smallest sub-designs by finding some subgroups of $H = \mathrm{Row}(\mathrm{Aut}(F))$ acting on strength 1 derived designs. Those subgroups, discussed in next parts, must have the property that they act separately on the row-index sets corresponding to the derived designs.

Fix $I_N := [1, 2, \ldots, N]$ the row-index list of F, and recall that $r_1 \geq r_2 \geq \ldots \geq r_d$. We explicitly distinguish the list I_N with $\{1, 2, \ldots, N\}$ in this section. Then H acts naturally on X' indices. Furthermore, we employ the following.

Definition 2.5. *We say a row permutation $g_R \in H$* acts fixed-point free*, or globally on X if it moves every index. Otherwise, if the moved points of g_R form a proper subset J of $\{1, \ldots, N\}$, i.e., it fixes point-wise the complement 'list' of J in I_N, we say g_R* acts locally *at that subset.*

The first step is to localize the formation of a vector X of the form (12) by taking the derived designs of strength $t-1$. We get the r_1 derived designs $F_1, \ldots, F_{r_1}$, each of which is an $\mathrm{OA}(r_1^{-1}N; r_2 \cdots r_d; t-1)$. Clearly, if a solution vector X exists, then it is formed by r_1 sub-vectors u_i of length $\frac{N}{r_1}$:

$$X = [u_1; u_2; \ldots; u_{r_1}], \text{ where } u_i = \left(x_{\frac{(i-1)N}{r_1}+1}, \ldots, x_{\frac{iN}{r_1}} \right). \tag{16}$$

Denote by V_i the set of all sub-vectors u_i which can be added to the ith derived design F_i to form an $\mathrm{OA}(r_1^{-1}N; r_2 \cdots r_d \cdot s; t-1)$. Let $V = V_1 \times V_2 \times \ldots \times V_{r_1}$.

We propose a simple scheme, Algorithm 1 to find all non isomorphic solution vectors $X \in V$. Algorithm 1 can be mathematically realized in 3 steps as follows.

Algorithm 1 Find all non isomorphic vectors X in $[F|X]$

EXTEND-ONE-FACTOR(F)

Input F is a strength t design;

Output All non-isomorphic extensions of F to $[F|X]$

a/ Find all candidate sub-vectors $u_i \in V_i$, $i = 1, \ldots, r_1$, using associated permutation subgroups

b/ Discard (prune) them as many as possible by using subgroups of H

c/ Plug those u_is together, then compute the representatives of the $\sigma = H \times \tau$-orbits in V, the solution space $\mathrm{Z}(P)$ of P.

a) Forming permutation subgroups of the derived designs

Remind that we viewed $F \in \mathrm{OA}(N; r_1 \cdot r_2 \cdots r_d; 3)$ as an $N \times d$-matrix with the $[l, j]$-entry is written as $F[l, j]$. For each derived design F_i w. r. t. the first column of F, the row-index set of F_i, denoted by $\mathrm{RowInd}(F_i)$ for $1 \le i \le r_1$, is defined as

$$\mathrm{RowInd}(F_i) := \big\{ l \in \{1, 2, \ldots, N\} : F[l, 1] = i - 1 \big\}.$$

Define the stabilizer in H of F_i by

$$\begin{aligned} N_H(F_i) &:= \text{Normalizer}\,\big(H, \mathrm{RowInd}(F_i)\big) \\ &= \big\{ h \in H : \mathrm{RowInd}(F_i)^h = \mathrm{RowInd}(F_i) \big\}. \end{aligned} \tag{17}$$

In this way, we find r_1 subgroups of H corresponding to the derived designs F_i. But it can happen that $\mathrm{RowInd}(F_l)^h \neq \mathrm{RowInd}(F_l)$ for some $h \in N_H(F_i)$ and $1 \le l \neq i \le r_1$. To make sure that the row permutations act independently on the F_i, we define the group of row permutations acting locally on each F_i as:

$$L(F_i) := \text{Centralizer}\,\big(N_H(F_i), J(F_i)\big), \tag{18}$$

where $J(F_i) := I_N \setminus \mathrm{RowInd}(F_i)$ is the sublist of I_N consisting of elements not in $\mathrm{RowInd}(F_i)$.

The group $L_i := L(F_i)$ acts locally at $\mathrm{RowInd}(F_i)$, i.e. it acts on the row-indices of F_i and fixes pointwise any row-index outside F_i.

Definition 2.6. *These subgroups* L_i *- of the group* $H = \mathrm{Row}(\mathrm{Aut}(F))$ *- are called the* row permutation subgroups associated with strength 2 derived designs.

These subgroups can be determined further as follows.
For an integer $m = 1, 2, \ldots, t-1$ and for $j = 1, 2, \ldots m$, denote by

$$F_{i_1,\ldots,i_m} := \mathrm{OA}\left(\frac{N}{r_1 r_2 \cdots r_m}; r_{m+1} \cdots r_d; t-m\right) \tag{19}$$

the derived designs of F taken with respect to symbols $i_1, \ldots, i_m$, where symbol i_j in column j and $i_j = 1, \ldots, r_j$. Define the row-index set of $F_{i_1,\ldots,i_m}$ by

$$\mathrm{RowInd}(F_{i_1,\ldots,i_m}) := \bigcap_{j=1}^{m} \big\{l \in \{1, 2, \ldots, N\} : F[l, j] = i_j - 1\big\}. \tag{20}$$

Let $J(F_{i_1,\ldots,i_m}) := I_N \setminus \mathrm{RowInd}(F_{i_1,\ldots,i_m})$. Generalizing (17) and (18) gives:

$$\begin{aligned} N_H(F_{i_1,\ldots,i_m}) &:= \mathrm{Normalizer}\big(H, \mathrm{RowInd}(F_{i_1,\ldots,i_m})\big), \\ L(F_{i_1,\ldots,i_m}) &:= \mathrm{Centralizer}\big(N_H(F_{i_1,\ldots,i_m}), J(F_i)\big), \text{ for } 1 \le i_j \le r_j. \end{aligned}$$

b) Using permutation subgroups of the derived designs

Definition 2.7. $L(F_{i_1,\ldots,i_m})$ *is called the subgroup associated with the derived design* $F_{i_1,\ldots,i_m}$*. We say* $L(F_{i_1,\ldots,i_m})$ *acts locally on the derived design* $F_{i_1,\ldots,i_m}$*, and write* $L_{i_1,\ldots i_m} := L(F_{i_1,\ldots,i_m})$*, for* $1 \le i_j \le r_j$*,* $j = 1, 2, \ldots m$*, if no ambiguity occurs.*

For $t = 3$, we compute these subgroups for $m = 1$ and $m = 2$. If $m = 1$, we have s_1 subgroups $L(F_i)$ acting locally on strength 2 derived designs; and if $m = 2$, then $s_1\ s_2$ subgroups $L(F_{i,j})$ acting locally on strength 1 designs.

We now show how to use the subgroups $L_{i_1,\ldots,i_m}$. Recall that $\mathrm{Z}(P)$ is the set of all natural solutions X. From Eq. (14) in Theorem 2.2, K^g is an isomorphic array of $K = [F|X]$, hence the vector X^g can be pruned from $\mathrm{Z}(P)$, for any solution X and any permutation $g \in \mathrm{Aut}(F)$.

We use the following notations in the remaining parts. For a fixed m-tuple of symbols $i_1, \ldots, i_m$, let $V_{i_1,\ldots,i_m}$ be the set of solutions of fraction

$$F_{i_1,\ldots,i_m} = \mathrm{OA}((r_1 r_2 \cdots r_m)^{-1} N; r_{m+1} \cdots r_d; t-m), \text{ for } 1 \le m \le t-1.$$

For any sub-vector $u \in V_{i_1,\ldots,i_m}$, from (20) and (16), let

$$\begin{aligned} &I(u) := \mathrm{RowInd}(F_{i_1,\ldots,i_m}); \quad J(u) := I_N \setminus I(u); \text{ and} \\ &\mathrm{Z}(u) := \big\{(x_j) : j \in J(u) \text{ and } \exists X \in \mathrm{Z}(P) \text{ s.t. } X[I(u)] = u\big\}, \end{aligned}$$

here $X[I(u)] := (x_i : i \in I(u))$. For instance, if $m = 1$ and $u \in V_1$ then

$$\mathrm{Z}(u) \; = \{\, [u_2; \ldots ; u_{r_1}] : X = [u; u_2; \ldots ; u_{r_1}] \in \mathrm{Z}(P) \,\}.$$

Theorem 2.3 (Key theorem)**.** *For any pair of sub-vectors* $u, v \in V_{i_1,\ldots,i_m}$*, if* $v = u^{g_R}$ *for some row permutation* $g_R \in L_{i_1,\ldots,i_m}$*, we have* $\mathrm{Z}(u) = \mathrm{Z}(v)$*.*

We prove this key theorem in the next two claims. In Lemma 2.1, without loss of generality, it suffices to give the proof for the first strength 2 derived array. Theorem 2.4 then shows the induction step.

Lemma 2.1 (Case $m = 1$)**.**

Let u_1 *and* v_1 *be two arbitrary sub-solutions in* V_1*, ie, they form strength 2 OAs* $[F_1|u_1]$ *and* $[F_1|v_1]$ *of the form* $\mathrm{OA}(r_1^{-1}N; r_2 \cdots r_d \cdot s; 2)$*. Let*

$$\mathrm{Z}_X(u_1) = \{\, [u_2; \ldots ; u_{r_1}] : X = [u_1; u_2; \ldots ; u_{r_1}] \in \mathrm{Z}(P) \,\},$$
$$\mathrm{Z}_Y(v_1) = \{\, [v_2; \ldots ; v_{r_1}] : Y = [v_1; v_2; \ldots ; v_{r_1}] \in \mathrm{Z}(P) \,\}.$$

Suppose that there exists a nontrivial subgroup, say $L(F_1)$*, and if* $v_1 = u_1^h$ *for some* $h \in L_1$*, we have* $\mathrm{Z}_X(u_1) = \mathrm{Z}_Y(v_1)$*.*

Proof. See Appendix C in Section 4. □

As a result, we can wipe out all solutions $Y = [v_1; v_2; \ldots ; v_{r_1}] \in \mathrm{Z}(P)$ if $v_1 \in u_1^{L_1}$, the L_1- orbit of u_1 in V_1. In other words, if we get $V_1 \neq \emptyset$, then it suffices to find the first sub-vector of vector X by selecting $|V_1|/|L_1|$ representatives u_1 from the L_1- orbits in V_1. Furthermore, the above proof is independent of the original choice of derived design. Hence it can be done simultaneously at all solution sets $V_1, V_2, \ldots , V_{r_1}$, using the subgroups $L_1, \ldots , L_{r_1}$.

We call this procedure, that results from Main Theorem 2.3, the *local pruning process* using strength 2 derived designs. Next, if $t \geq 3$ we extend the proof of Lemma 2.1 to cases $2 \leq m \leq t - 1$.

Theorem 2.4 (Case $m > 1$.)**.** *For any pair of sub-vectors* $u, v \in V_{i_1,i_2}$*, if* $v = u^{g_R}$ *for some* $g_R \in L_{i_1,i_2}$*, we have* $\mathrm{Z}(u) = \mathrm{Z}(v)$*.*

Proof. See [Man Nguyen, [31, 37]. □

c) Operations on derived designs- An agent-based localization

The above-proposed localizing idea can be enhanced further when we consider each derived design as an agent that receives data from its lower strength derived designs, make some appropriate operations, then pass the result to its parent design. Specifically, notice that strength 1 and strength t designs require special operations. To be precise, at the global scale of strength t design, it suffices to find only the representatives of the $H \times \tau$-orbits [see Formula (15)] in the solution space $\mathrm{Z}(P)$ of P.

We now formalize our new agent-based localization. Recall from (19) that the symbols $i_1, \dots, i_m$ $(1 \le i_j \le r_j)$ indicate the derived design having symbol i_j in column j, for $j = 1, \dots, m$.

From Definition 2.7, $L_{i_1,\dots,i_m}$ are the subgroups associated with the derived designs $F_{i_1,\dots,i_m}$ having strength $t-m$. When $m = t-1$, write $L_{i_1,\dots,i_{t-1}}$ for the subgroup associated with the strength 1 derived design $F_{i_1,\dots,i_{t-1}}$. The agents of derived designs can be described as follows.

At initial designs $F_{i_1,\dots,i_{t-1}}$ (Initial step when $m = t-1$):

Input: $F_{i_1,\dots,i_{t-1}}$;

Operation:

- form $V_{i_1,\dots,i_{t-1}}$, the set of all strength 1 vectors of length $(r_1 r_2 \cdots r_{t-1})^{-1} N)$ being appended to $F_{i_1,\dots,i_{t-1}}$,
- compute $L_{i_1,\dots,i_{t-1}}$, and
- find the representatives of $L_{i_1,\dots,i_{t-1}}$ - orbits in the set $V_{i_1,\dots,i_{t-1}}$;

Output: these representatives, ie, solutions of $F_{i_1,\dots,i_{t-1}}$.

At strength k derived designs $(1 < k \le t-1)$: let $m := t-k$, we have

Input: the vector solutions having length $(r_1 r_2 \cdots r_m \cdot r_{m+1})^{-1} N$ of strength $k-1$ sub-designs; and the subgroup $L_{i_1,\dots,i_m}$;

Operation:

- form sub-vector solutions having length $(r_1 r_2 \cdots r_m)^{-1} N)$ of $F_{i_1,\dots,i_m}$,
- prune these solutions by $L_{i_1,\dots,i_m}$;

Output: representatives of the $L_{i_1,\dots,i_m}$- orbits in the set $V_{i_1,\dots,i_m}$.

At the (global) design F:

Input: the sub-vectors from strength $t-1$ derived designs;

Operation: find the representatives of σ-orbits in the Cartesian product $V = V_1 \times V_2 \times \dots \times V_{r_1} = \{\text{vectors } X \text{ of length } N\}$ where V_i had been already pruned by the subgroup L_i $(i = 1, 2, \dots, m)$;

Output: Two steps

a/ (Isomorph-free test 1) returns solution vectors X which are non-isomorphic up to $\sigma = H \times \tau$,

b/ (Isomorph-free test 2) forms orthogonal arrays $K = [F|X]$ of the same strength t, then select only non-isomorphic arrays, by computing their canonical arrays.

We brief ideas in Algorithm 2, **Pruning-Uses-Symmetry**(F, d).

Algorithm 2 Pruning uses subgroups of derived designs

Pruning-Uses-Symmetry(F, d)

Input F is a strength t design; d is the number of columns required

Output All non-isomorphic extensions of F

$\diamond$ STEP 1: *Local pruning at strength k derived designs.*

1a) Find sub-vectors of $F_{i_1,\ldots,i_m}$, for $m := t-k$, and $k = 1, \ldots, t-1$,

1b) prune these sub-vectors locally and simultaneously by using $L_{i_1,\ldots,i_m}$,

1c) concatenate these sub-vectors to get sub-vectors in $V_{i_1,\ldots,i_{m-1}}$.

Comment: For $t = 3$, in Step 1), form subvectors $u_{i,j} \in V_{i,j}$ simultaneously at the $r_1 r_2$ sets $V_{i,j}$, then concatenate $u_{i,j}$ $(1 \le i \le r_1, 1 \le j \le r_2)$ to get $u_i \in V_i$.

$\diamond$ STEP 2: *Pruning at strength t design F.*

2a) Select the representative vectors X from the $\sigma = H \times \tau$-orbits of V
Comment: Each vector in V is formed by sub-vectors found from Step 1

2b) append non-isomorphic vectors X to F to get strength t OAs $[F|X]$,

2c) compute and store only their distinct canonical arrays,
(see Man Nguyen, [31, Section 2.2])

2d) get back non-isomorphic orthogonal arrays into a list Lf, return Lf.

$\diamond$ STEP 3: *Repeating step.*
`If` # current columns $< d$ Call **Pruning-Uses-Symmetry**(f, d) for $f \in Lf$
`Else` Return Lf `EndIf`

Example 2.1. *Let* $U := \big[[3,1],[2,3]\big]$, $F = \mathrm{OA}(24; 3.2^3; 3)$,

$$F = \begin{bmatrix} 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&2&2&2&2&2&2&2&2 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&0&1&1&1&1&0&0&0&1&0&1&0&1&0&1&1&1&0&0&0&0&1&1 \end{bmatrix}^T.$$

$\mathrm{Aut}(F)$ *has order 12288. Compute the group* $H = \mathrm{Row}(\mathrm{Aut}(F))$ *(from Definition 2.4), and update* $H = \mathrm{Stabilizer}(H, [1])$, *which is a permutation group of size 768. The three strength 2 derived designs give 8, 8, and 16 candidates respectively, so we must check* $8.8.16 = |V| = 1024$ *cases.*

The row permutation subgroups of the three strength 2 derived designs are

$$L_0 = [(), (7,8), (5,6), (5,6)(7,8), (3,4), (3,4)(7,8), (3,4)(5,6), (3,4)(5,6)(7,8)],$$
$$L_1 = [()], \ \textit{and}$$
$$L_2 = [(), (23,24), (21,22), (21,22)(23,24), (19,20), (19,20)(23,24), (19,20)(21,22), (19,20)(21,22)(23,24), (17,18), (17,18)(23,24), (17,18)(21,22), (17,18)(21,22)(23,24), (17,18)(19,20), (17,18)(19,20)(23,24), (17,18)(19,20)(21,22), (17,18)(19,20)(21,22)(23,24)]$$

with corresponding orders 8,1,16. And the subspaces are pruned to 1,8, and 1 vectors respectively. That is we just check 8 cases.

New strength 3 OA obtained with the group-theoretic approach

Some unknown OAs that previous well-known methods failed to compute (e.g. Man Nguyen [9, 15, 33]), found by our combined approach, are listed in Table 2. We have used multiplicity notation for automorphism group orders. The (**IS**) construction means employing the **I**nteger linear formulation and **S**ymmetries of automorphism groups of OAs, fully developed in this Section 2.5.

Table 2: New strength 3 mixed OAs of sizes $N \leq 100$.

N	Type; Strength t	#	Size of the group Aut(F)	Methods
80	$5 \cdot 4 \cdot 2^5$; $t = 3$	≥ 1		(IS)
80	$5 \cdot 4 \cdot 2^6$; $t = 3$	≥ 5	$2^2, 4^3$	(IS)
96	$6 \cdot 4^2 \cdot 2^5$; $t = 3$	≥ 1199	1^{411}, 2^{370}, 4^{250}, 8^{137}, 12, 16^{29}	,,

3 Finding best routes in logistics management

In sustainable economic development, besides of quality engineering (targeted mostly to industries), effective urban transportation is another side of the story. Urban traffic certainly is not just affected by households demands, but also substantially influenced by transactional activities and logistics expenditures of firms, both production and service. We describe a well-known logistical transportation problem with a discrete optimization setting in this part, then present few newly related results from which optimal solutions can be obtained.

3.1 *A balanced source-sink transportation problem*

Consider m assembly factories A_i and n warehouses W_j for some *integral* products (that is they are *indivisible* as car, air conditioner or trucks of these goods). Suppose that both the A_i and W_j belong to the same cooperation. Operations researchers would requite that the ith factory supplies daily r_i products, the jth warehouse need c_j products, and furthermore, that the total supply must agree with the total demand. i.e. $r_1 + r_2 + \ldots + r_m = c_1 + c_2 + \ldots + c_n$, or

$$\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j, \tag{21}$$

for any given vector $\mathbf{r} \in \mathbb{N}^m$ and $\mathbf{c} \in \mathbb{N}^n$. We aim to find a minimum cost plan to transport goods from assembly factories to warehouses. Our problem is mathematically formulated as follows.

Let $W = (w_{ij}) \in \mathbb{R}_*^{m\times n}$ be an $m \times n$ matrix of non-negative real numbers, called *cost matrix*, representing the transportation costs. Fix vectors $\mathbf{r} \in \mathbb{N}^m$ and $\mathbf{c} \in \mathbb{N}^n$ so that $\sum_{i=1}^m r_i = \sum_{j=1}^n c_j$. A *transportation plan* is a matrix $X = (x_{ij}) \in \mathbb{N}_*^{m\times n}$ in which x_{ij} is the number of items to be brought from factory A_i to warehouse W_j.

Problem 1. We need to find if there exists a matrix $X \in \mathbb{N}^{m\times n}$ such that

$$\begin{aligned}
&\sum_{j}^{n} x_{ij} = r_i, \text{ for each } i = 1, 2, \ldots, m,\\
&\sum_{i}^{m} x_{ij} = c_j, \text{ for each } j = 1, 2, \ldots, n, \text{ and}\\
&\langle W, X\rangle = \sum_{i,j} w_{ij}x_{ij} \text{ is minimized.}
\end{aligned} \tag{22}$$

A much simpler companion problem, namely its LP-relaxation, is

Problem 2. To determine whether there exists a matrix $X \in \mathbb{R}_*^{m\times n}$ (where $R_* = \{x : x \in \mathbb{R} \wedge x \geq 0\}$) such that $\sum_j^n x_{ij} = r_i$, $\sum_i^m x_{ij} = c_j$ for each $i = 1..m$, $j = 1..n$, and $\langle W, X\rangle$ is minimized.

Problem 1 belongs to the NP-class of complexity, where the input size is $m.n$, in general. However, the second one can be solved in *polynomial time*, as we all know (e.g. see [13, 21]). We can prove the followings.

Theorem 3.1. *Given vectors* $\mathbf{r} = [r_1, r_2, \ldots, r_m] \in \mathbb{N}^m$, $\mathbf{c} = [c_1, c_2, \ldots, c_n] \in \mathbb{N}^n$, *such that* $\sum_{i=1}^m r_i = \sum_{j=1}^n c_j$. *If there exists a matrix* $X \in \mathbb{R}_*^{m\times n}$ *such that*

$$\langle W, X\rangle = \sum_{i,j} w_{ij}x_{ij}$$

is minimum, then there exists a matrix $X' \in \mathbb{N}^{m\times n}$ *such that*

$$\langle W, X'\rangle = \sum_{i,j} w_{ij}x'_{ij} = \langle W, X\rangle.$$

This is somewhat similar to Matousek, [21, Theorem 3.2.1.], but our proof - shown later in Section 3.2- is different and shorter. Furthermore we state

Theorem 3.2. *If $X \in \mathbb{N}^{m\times n}$ is a solution of Problem 1 then X has at most $m+n-1$ nonzero elements.*

We will prove these theorems using graph theory, among other tools. Before presenting proofs, let us introduce a few more useful concepts.

Definition 3.1. *Let matrix $X \in \mathbb{R}_*^{m\times n}$ be an $m \times n$ matrix of non-negative real numbers. A set $T = \{X_{i_p,j_p} : X_{i_p,j_p} \notin \mathbb{N}\}$ is called a k-cycle on X, denoted $T\diamondsuit X$, if it satisfies that $|T| = k > 3$ and that for all $p = 1..k$:*

$$[i_p = i_{p+1}, j_{p+1} = j_{p+2}] \quad or \quad [j_p = j_{p+1}, i_{p+1} = i_{p+2}],$$

in which $i_{k+1} = i_1, i_{k+2} = i_2, j_{k+1} = j_1$ and $j_{k+2} = j_2$.

For instance, if X is a 7×5 matrix given as in Table 3 then a 6-cycle $T = \{X_{3,3}, X_{1,3}, X_{1,5}, X_{5,5}, X_{5,1}, X_{3,1}\}$. Let $X \in \mathbb{R}_*^{m\times n}$ and we fix an order on elements of $T = \{X_{i_p,j_p} : X_{i_p,j_p} > 0, \forall p = 1..k\} \diamondsuit X$, a k-cycle on X with all positive entries. We define two subsets of T as follows:

$$T_e = \{T[p] : p \equiv 0 \mod 2\}, \text{ and } T_o = \{T[p] : p \equiv 1 \mod 2\}.$$

In the above example, it is obvious that $T_e = \{X_{1,3}, X_{5,5}, X_{3,1}\}$ and $T_o = T\backslash T_e$.

Table 3: **A 6-cycle T in a 7×5 matrix X**

1	2	**1.1** $\rightarrow$	1 $\rightarrow$	**1.2** $\downarrow$
2	2	3 $\uparrow$	1	2 $\downarrow$
0.3 $\rightarrow$	$\rightarrow$ 4	**1.3** $\uparrow$	3	3 $\downarrow$
4 $\uparrow$	3	2	0.4	4 $\downarrow$
0.4 $\uparrow$	1 $\leftarrow$	2 $\leftarrow$	1 $\leftarrow$	$\longleftarrow$ **2.1**
5	2	0.5	2	0.3
6	7	1	1	2

Now let $X \in \mathbb{R}_*^{m\times n}$, $T \diamondsuit X$ with a fixed order, and let $\varepsilon > 0$. We define two matrices $X^{+\varepsilon}$ and $X^{-\varepsilon}$ in $\mathbb{R}^{m\times n}$ as follows.

Definition 3.2. *Matrices $X^{+\varepsilon}$ and $X^{-\varepsilon}$ are respectively determined by*

$$X^{+\varepsilon}_{i,j} = X_{i,j}, \text{ if } X_{i,j} \notin T,$$
$$X^{+\varepsilon}_{i,j} = X_{i,j} + \varepsilon, \text{ if } X_{i,j} \in T_o, \quad X^{+\varepsilon}_{i,j} = X_{i,j} - \varepsilon, \text{ if } X_{i,j} \in T_e;$$

$$X^{-\varepsilon}_{i,j} = X_{i,j}, \text{ if } X_{i,j} \notin T,$$
$$X^{-\varepsilon}_{i,j} = X_{i,j} - \varepsilon, \text{ if } X_{i,j} \in T_o, \quad X^{-\varepsilon}_{i,j} = X_{i,j} + \varepsilon, \text{ if } X_{i,j} \in T_e.$$

Definition 3.3. *Let $X \in \mathbb{R}_*^{m\times n}$, we associate with X a graph $G(X) := (V, E)$ where $V = \{A_1, A_2, \ldots, A_m,\ B_1, B_2, \ldots, B_n\}$ represents the row and column indexes of X, and $E = \{A_iB_j : X_{i,j} \notin \mathbb{N}\}$ describes non-natural entries of X.*

Using the cycle given in Table 3, we have:

Table 4: **Matrix $X^{+\varepsilon}$ and $X^{-\varepsilon}$.**

$X^{+\varepsilon} =$

1	2	**1.1**- ε	1	**1.2** + ε
2	2	3	1	2
0.3 - ε	4	**1.3** + ε	3	3
4	3	2	0.4	4
0.4 + ε	1	2	1	**2.1**- ε
5	2	0.5	2	0.3
6	7	1	1	2

$X^{-\varepsilon} =$

1	2	**1.1**+ ε	1	**1.2** - ε
2	2	3	1	2
0.3 + ε	4	**1.3** - ε	3	3
4	3	2	0.4	4
0.4 - ε	1	2	1	**2.1**+ ε
5	2	0.5	2	0.3
6	7	1	1	2

With the above 7×5 matrix X, its corresponding graph $G(X)$ has 12 vertices, drawn in Figure 1.

Lemma 3.1. *If $X \in \mathbb{R}_*^{m\times n}$ has a cycle T then the abovedefined graph $G(X)$ also has a cycle T_G that corresponds to T and the number of elements of T equals the number of vertices of T_G.*

Lemma 3.2. *If T is a cycle on $X \in \mathbb{R}_*^{m\times n}$ then T has an even number of elements, and furthermore, the matrices $X^{+\varepsilon}$, $X^{-\varepsilon}$ and X have the same vectors of row sums and column sums.*

Proof. T has an even number of elements since its companion cycle T_G, being in the bipartite graph $G(X)$, has an even number of edges/vertices. The fact that $X^{+\varepsilon}$, $X^{-\varepsilon}$ have the same vectors of row sums and column sums as X's comes from their definitions. □

Lemma 3.3. *If a graph G is connected and has no degree-one vertex then G has a cycle.*

Proof. If $G = (V(G), E(G))$ is connected and has no degree-one vertex, all nodes have degree at least two, then the number of edges is at least $|V(G)|$, so G is not a tree. Hence, G must contain a cycle. □

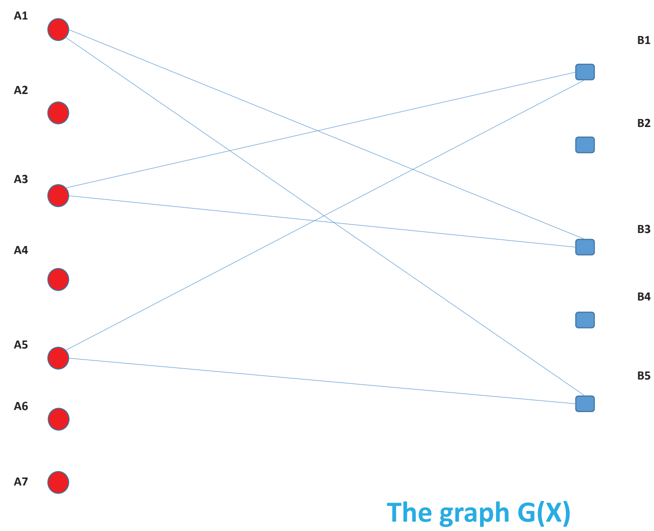


Figure 1: The associated graph $G(X)$ of matrix X

3.2 *Proving of theorems*

Proof of Theorem 3.1

Proof. Suppose that there does not exist matrix $X' \in \mathbb{N}^{m\times n}$ such that $\langle W, X\rangle = \langle W, X'\rangle$ (*). Then any $X^* \in \mathbb{R}_*^{m\times n}$ that satisfies $\langle W, X\rangle = \langle W, X^*\rangle$ must have at least an element $X^*_{i,j}$ at some row i and column j not being integer. But the sums of entries in the row i and the column j are integers. Hence, there are elements $X^*_{i,k}$ and $X^*_{l,j}$, where $1 \leq k \leq n$ and $1 \leq l \leq m$, also not integers.

Thus $G := G(X^*)$ or any its connected component must have no one-degree vertex, so $G(X^*)$ or any its connected component has a cycle (Lemma 3.2). Let T_G is a cycle of $G(X^*)$ then there is a cycle T on X^*. We set $\varepsilon := T_{min}$ be the minimum element of T, then $X^{*^{-\varepsilon}}, X^{*^{+\varepsilon}} \in \mathbb{R}_+^{m\times n}$. and

$$\langle W, X^{*^{-\varepsilon}}\rangle = \langle W, X^*\rangle - \alpha, \qquad \langle W, X^{*^{+\varepsilon}}\rangle = \langle W, X^*\rangle + \alpha,$$

with $\alpha = \varepsilon\left(\sum_{X^*_{i,j}\in T_o} W_{i,j} - \sum_{X^*_{i,j}\in T_e} W_{i,j}\right)$.

If $\alpha \neq 0$ then $\langle W, X^*\rangle$ is not minimum, thus $\alpha = 0$ and

$$\langle W, X^{*^{-\varepsilon}}\rangle = \langle W, X^*\rangle = \langle W, X^{*^{+\varepsilon}}\rangle.$$

If $\varepsilon = T_{min} \in T_o$ then $X^{*^{-\varepsilon}}$ has a new element whose value is 0, thus $\langle W, X^{*^{-\varepsilon}}\rangle = \langle W, X^*\rangle$ and $X^{*^{-\varepsilon}}$ has non-integer elements less than those of X^*

(conflict to (*)). Similarly, if $T_{min} \in T_e$, then (*) is also false. Therefore (*) is always false, the theorem is proved. $\square$

Proof of Theorem 3.2

To prove this theorem, we define a variation of the concept of cycle given in Definition 3.1. More clearly, we replace the condition $X_{i_p,j_p} \notin \mathbb{N}$ with $X_{i_p,j_p} \neq 0$.

Definition 3.4. *Let $X \in \mathbb{R}_*^{m\times n}$, a set $T = \{X_{i_p,j_p} : X_{i_p,j_p} \neq 0\}$ is called a nonzero-cycle on X, denoted $T \boxdot X$, if it satisfies that $|T| = k > 3$ and that for all $p = 1..k$: $[i_p = i_{p+1}, j_{p+1} = j_{p+2}]$ or $[j_p = j_{p+1}, i_{p+1} = i_{p+2}]$, in which $i_{k+1} = i_1, i_{k+2} = i_2, j_{k+1} = j_1$ and $j_{k+2} = j_2$.*

Now for a given $T = \{X_{i_p,j_p} : X_{i_p,j_p} > 0, \forall p = 1..k\} \boxdot X$, a nonzero-cycle on X with all positive entries, we let

$$T_e = \{T[p] : p \equiv 0 \mod 2\}, \text{ and } T_o = \{T[p] : p \equiv 1 \mod 2\}.$$

Given a $\varepsilon > 0$, we define two matrices $X^{+\varepsilon}$ and $X^{-\varepsilon}$ in $\mathbb{R}^{m\times n}$ by the same formulas as introduced in Definition 3.2. Finally, we need the below.

Definition 3.5. *Let $X \in \mathbb{R}_*^{m\times n}$, we associate with X a graph $H(X) := (V, E)$ whose vertices and edges are:*

- *$V = \{A_1, A_2, A_3, \ldots, A_m, B_1, B_2, B_3, \ldots, B_n\}$ representing the row and column indexes of X,*
- *$E = \{A_iB_j : X_{i,j} \neq 0\}$ describing nonzero entries of X.*

Proving Theorem 3.2. Suppose $X \in \mathbb{N}_*^{m\times n}$ is a solution of transportation problem which has the least number of nonzero elements and X has more than $m + n - 1$ elements nonzero (*).

So $H(X)$ is a graph with $m+n$ nodes and has more than $m+n$ edges then $H(X)$ has a cycle (because $H(X)$ can not be a tree or a set of trees). Let T be a cycle of $H(X)$ then there is a non-zero cycle T on X. Denote by T_{min} the minimum element of T. Let $\varepsilon = T_{min}$ then $X^{-\varepsilon}, X^{+\varepsilon} \in \mathbb{R}_+^{m\times n}$ and

$$\langle W, X^{-\varepsilon}\rangle = \langle W, X\rangle - \alpha, \ \langle W, X^{+\varepsilon}\rangle = \langle W, X\rangle + \alpha$$

with

$$\alpha = \varepsilon\Big(\sum_{X_{ij}\in T_o} W_{ij} - \sum_{X_{ij}\in T_e} W_{ij}\Big).$$

If $\alpha \neq 0$ then $\langle W, X\rangle$ is not minimum, so $\alpha = 0$ and

$$\langle W, X^{-\varepsilon}\rangle = \langle W, X\rangle = \langle W, X^{+\varepsilon}\rangle.$$

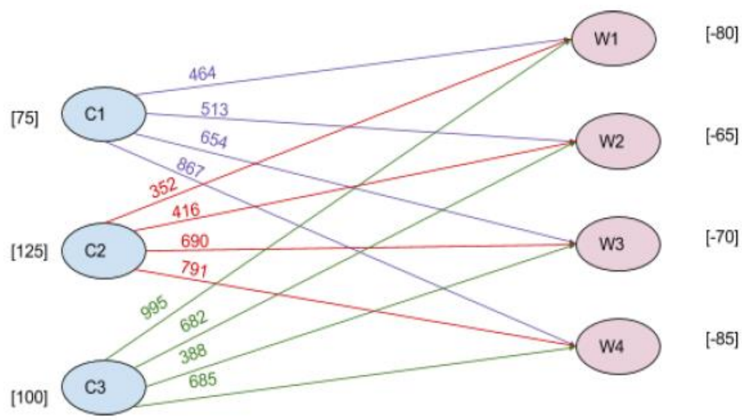
If $T_{min} \in T_o$ then $X^{*-\varepsilon}$ has a new element which value is 0, thus

$$< W, X^{*-\varepsilon} >= \langle W, X\rangle$$

and $X^{-\varepsilon}$ has nonzero elements less than X (conflict to (*)). Similar, if $T_{min} \in T_e$, then (*) is also false. Hence (*) is always false, the theorem is proved.

$\square$

3.3 *Experimental computation results*



A berry cannery manufacturer has 3 cannery and 4 warehouse

Figure 2: A typical balanced source-sink transportation scheme

As an illustration for Theorem 3.2, we consider a transportation plan of a berry cannery with 3 canneries and 4 warehouses in Fig. 2. The cost of transporting a unit (a truckload) from factory C_i to warehouse W_j is w_{ij}, e.g. $w_{12} = 513...$, $w_{32} = 682, w_{34} = 685$. This plan obviously is balanced source-sink since the total goods coming out from the sources equals the total goods entering the sinks, $\sum_{i=1}^{3} r_i \; = r_1 + r_2 + r_3 = 300 = \sum_{j=1}^{4} c_j = c_1 + c_2 + c_3 + c_4$.

Let x_{ij} be the number of truckloads being shipped from cannery C_i to warehouse W_j, where $i = 1, 2, 3;\; j = 1, 2, 3, 4$. The ILP model of Problem 1 is

minimize $Z = \langle W, X\rangle = \sum_i^3 \sum_j^4 \; w_{ij}\; x_{ij} = 464\; x_{11} + \cdots + 388\; x_{33} + 685\; x_{34}$
subject to

a) Decision variable constraints: $x_{ij} \in \mathbb{N}$

b) Cannery constraints: $\sum_j^4 \; x_{1j} = 75,\;\; \sum_j^4 \; x_{2j} = 125,\;\; \sum_j^4 \; x_{3j} = 100$

c) Warehouse (sink) constraints:

$$\sum_{i}^{3} x_{i1} = 80, \sum_{i}^{3} x_{i2} = 65, \sum_{i}^{3} x_{i3} = 70, \sum_{i}^{3} x_{i4} = 85.$$

Using LINGO software we got the optimal cost $Z = \$152535$ from the solution of $x_{12} = 20,\ x_{14} = 55$, $x_{21} = 80$, $x_{22} = 45$, $x_{33} = 70,\ x_{34} = 30$, and all other $x_{ij} = 0$. Clearly, there are exactly $m + n - 1 = 4 + 3 - 1 = 6$ nonzero values. Few stronger and diverser constraints will be proposed in Section 4, and see APPENDIX D on how to code our ILP problem.

4 Summary and Conclusion

What have been done through this short excursion? We have seen that the *algebraic language* and *statistical formulation* are essential for addressing complicated problems of reality, that quality engineering, statistical quality control and logistics management are rich sources of CAS.

Two conjectures in the first two topics might be interesting open problems. Moreover, would computer algebra be more useful in the third topic of logistics? Besides of shipping cost optimality, we can think about integration of various demands from both suppliers (sources) and customers (sinks) in a framework named **collaborative logistics**. If the firm's manager want to further improve transportation plans, not only in terms of *shipping cost* but also *load balancing* at both sources and sinks, then he can impose extra constraints like

$$\text{at each source } i = 1, \cdots, m: \ 0 \le x_{ij} \le k_j \ \lfloor r_i/n \rfloor, \ \forall j = 1, \cdots, n;$$

$$\text{at each sink } j = 1, \cdots, n: \ 0 \le x_{ij} \le l_i \ \lfloor c_j/m \rfloor, \ \forall i = 1, \cdots, m;$$

where balancing weights $0 < k_1, k_2, \cdots, k_n \le n$, and $0 < l_1, l_2, \cdots, l_m \le m$, to avoid over-sending goods to warehouses j, and also disregard receiving goods too often from big suppliers. The values of x_{ij} then are roots of polynomials with degrees much more lower than the primary upper bounds r_i and c_j, or better $\min(r_i, c_j)$. E.g, with $m = 3,\ n = 4$ suitable weights can be $k_1 = 1,\ k_2 = 1, k_3 = 0, k_4 = 2$, meaning the 3rd warehouse doesn't receive goods from any supplier. Additionally using such *utility* and/or *load balancing* constraints for Model (22) of Problem 1 possibly is a promising move, not only to better balance utilities or benefits of both parties (producers and customers), but also to reduce the complexity and root domains (the limited values of decision variables provide a much more reduced feasibility region comparing to our original feasibility region).

What CAS topics can be investigated next? Few questions we may raise, and this emerging approach might provide sound methodology for modeling complex problems arising in sectors of finance, health-care, and geo-statistics ...[25].

Acknowledgment

I firstly express my sincere gratitude to my mentors, Nguyen Huu Anh (Vietnam) and Arjeh M. Cohen (Netherlands) for their valuable guidance and discussions. Nguyen Huu Anh suggested a study of testing conjectures on Weyl algebra $A_1(\mathcal{F})$ in Section 1, while Arjeh M. Cohen particularly pointed the use of multiplication matrices in Theorem 2.1 of Section 2.4. Second, I highly appreciate the helps of Nguyen Van Sanh and Yongwimon Lenbury (Thailand) in many aspects. Last but not least, this review can not be done without generous supports of Center of Excellency in Mathematics (CEM), Ministry of Education, (Thailand), Department of Mathematics, Faculty of Science Mahidol University (Thailand); and University of Technology, VNUHCM (Vietnam).

APPENDICES

APPENDIX A: Groebner basis methodology

We recall here the essential computational machinery of Groebner bases to solve systems of polynomial equations appearing in Computational Algebraic Statistics. The **Groebner basis methodology** basically is about computing on *multivariate polynomial systems*. Let us firstly brief a few basic notation.

A short polynomial algebraic background

We start with a set of input polynomials $F = \{f_1, \ldots, f_s\}$ on a field of numbers $\mathcal{F}$ [$\mathcal{F} = \mathbb{R}$, or $\mathbb{C}$ the complex numbers], with its algebraic closure denoted as $\overline{\mathcal{F}}$. We distinguish the variables $\boldsymbol{X} := (X_1, X_2, X_3, \ldots, X_n)$ from the coordinates $\boldsymbol{x} := (x_1, x_2, x_3, \ldots, x_n)$. So we talk about the ring (over the field $\mathcal{F}$)

$$\mathcal{F}[\boldsymbol{X}] := \mathcal{F}[X_1, X_2, X_3, \ldots, X_n]$$

and denote the coordinates of a point $x \in \mathcal{F}^n$ by $x = (x_1, x_2, x_3, \ldots, x_n)$, in general. An **ideal** in $\mathcal{F}[\boldsymbol{X}]$ is a subset $J \subset \mathcal{F}[\boldsymbol{X}]$ consists of 0 and closed under the *addition* of its polynomials and the *multiplication* with an arbitrary polynomial in $\mathcal{F}[\boldsymbol{X}]$, we write $J \trianglelefteq \mathcal{F}[\boldsymbol{X}]$. Furthermore,

- An ideal $I \leq \mathcal{F}[\boldsymbol{X}]$ is *prime* iff $F \in I$ or $G \in I$ whenever $FG \in I$.
- In the space $\mathcal{A}^n := \mathcal{F} \times \mathcal{F} \times \cdots \times \mathcal{F}$ (n times), an *algebraic set* is the set of zeros of an ideal $J \leq \mathcal{F}[\boldsymbol{X}]$ that is determined by a finite set of polynomials $f_1, f_2, f_3, \ldots, f_s$:

$$\mathrm{Z}(J) = \mathrm{Z}(f_1, f_2, f_3, \ldots, f_s) := \{\boldsymbol{p} \in \mathcal{A}^n : f_i(\boldsymbol{p}) = 0, \quad \forall f_i\} = \bigcap_{i=1}^{s} \mathrm{Z}(f_i).$$

Hence, finding $\mathrm{Z}(J)$ is reduced to computing all $\mathrm{Z}(f_i)$ for $i = 1, 2, \ldots, s$. The algebraic set $\mathrm{Z}(J)$ of a prime ideal $J \leq \mathcal{F}[\boldsymbol{X}]$ is named an affine variety, denoted by $\mathbb{X} := \mathrm{Z}(J)$. For example, let $J = \langle F(X, Y, Z) := X^2 + Y^2 + Z^2 - 1 \rangle$ (note that principal ideal generated by an irreducible polynomial $F(X, Y, Z) \in \mathcal{F}[\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}]$ is a prime ideal if $\mathcal{F} \neq \mathbb{C}$).

Quotient ring and Zero-dimensional systems

- We say f and g are congruent modulo J, written $f \equiv g \mod J$ if $f - g \in J$.
- Relation $\equiv_J$ (congruent modulo J) is an equivalence relation on $\mathcal{F}[\boldsymbol{X}]$.
- The quotient set $\mathcal{F}[\boldsymbol{X}]/J$ (read $\mathcal{F}[\boldsymbol{X}]$ modulo J), is the set of all equivalence classes $[f] = \{g : g - f \in J\}$ with respect to relation $\equiv_J$:

$$\mathcal{F}[\boldsymbol{X}]/J := \{\, [f] : f \in \mathcal{F}[\boldsymbol{X}] \,\}.$$

Lemma 4.1.

- *The set $\mathcal{F}[\boldsymbol{X}]/J$ is a commutative ring with two operation $+$ and $\cdot$ being determined by:*

$$[f] + [h] = [f + h]; \quad \text{and } [f] \cdot [h] = [fh].$$

- *Every ideal in $\mathcal{F}[\boldsymbol{X}]/J$ is finitely generated.*
- *$\mathcal{F}[\boldsymbol{X}]/J$ has a linear space structure, and also it is an algebra.*

Definition 4.1. *The ideal J is called* zero-dimensional *if it has a finite number of solutions, that means $|\,\mathrm{Z}(J)| < \infty$.*

What is a Groebner basis G of a polynomial ideal J?

The key idea of Groebner basis method is to transform the given set $F = \{f_1, \ldots, f_s\}$ to a new set of output polynomials $G = \{g_1, \ldots, g_m\}$ so that information about F can be understood more easily through inspection of G. The computation of G from F uses *Buchbergers Algorithm* (1965).

This algorithmic method generalizes well-known algorithms:

- Gaussian Elimination (solving *linear system* in many variables)
- Euclidean Algorithm (computing gcd of two polynomials in one variate, then finding root of nonlinear *univariate polynomial* systems), and
- Simplex Algorithm (finding global optimum from local optimums).

Three associated questions are:

1. Can Gaussian elimination be extended to handle *nonlinear systems*?
2. Can Euclidean algorithm be generalized to factor *multivariate polynomials*?
3. Can Simplex algorithm be utilized to a scale where we have more rules to help us moving faster towards global optimum (in certain concerned polyhedral)?

Essentially, the *Gröbner basis* method was created by combining the three major techniques above.

Example 4.1 (Gaussian elimination)**.** *From the system* $F = \{2x+3y+4z = 5,\ 3x+4y+5z = 2\}$ *we get* $G = \{x = z-14,\ y = 11-2z\}$.

Example 4.2 (Euclidean algorithm finds gcd of two polynomials)**.** *From the system* $F = \{f_1,\ f_2\}$ *where* $f_1 = x^4 - 12x^3 + 49x^2 - 78x + 4$, $f_2 = x^5 - 5x^4 + 5x^3 + 5x^2 - 6x$ *we get the* $\gcd(f_1, f_2) = x^2 - 3x + 2$.

How about the last component? – the Simplex algorithm

Example 4.3 (ATM Utilization)**.** *Consider minimizing a integer-valued linear functional*

$$Z = P + N + D + Q, \text{ where } P + 5N + 10D + 25Q = 117 \text{ and } P, N, D, Q \in \mathbb{N}.$$

We also know that $5P(enny) = N(ickel), 10P = D(ecimal), 25P = Q(uarter)$, so could encode these additive relations in a multiplicative way, by introducing new variables p, n, d, q and constraints:

$$p^5 = n, p^{10} = d, p^{25} = q \text{ or, in terms of polynomials}$$

$$F := \{p^5 - n, p^{10} - d, p^{25} - q\}.$$

Then a feasible point, in our feasible polyhedron H, is represented by the term $p^{17}n^{10}d^5$. Using F we move slowly in H, but if we use $G := F \cup \{\text{some extra terms}\}$ like $G := F \cup \{n^2 - d, d^2n - q, d^3 - nq\}$ then the moving could be much faster to the global optimum $(2, 1, 1, 4)$.

Properties of Gröbner basis

Property 1: $\mathrm{Z}(J) = \emptyset \iff 1 \in J$.

Property 2: If G is a *Gröbner basis* of J, then $\mathrm{Z}(J)$ is finite iff for each variate X_i, G consists of a polynomial in only that X_i.

Example 4.4 (Experimenting in Maple soft)**.**

We find roots of a system of polynomial equations via computing *Gröbner basis* in Maple as follows.

$f := x^2 + y + z - 1; g := x + y^2 + z - 1; h := x + y + z^2 - 1;\ J := [f, g, h]$
$\mathit{with}(\mathit{Groebner});\ G := \mathrm{Basis}(J, \mathit{plex}(x, y, z));$ # and get a basis

$$G := [z^6 + 4*z^3 - 4*z^4 - z^2, 2*z^2*y + z^4 - z^2, y^2 - z^2 + z - y, x + y + z^2 - 1];$$

To check whether $\mathrm{Z}(J) = \mathrm{Z}(G)$ is finite, and solve the system we use:
$\mathrm{IsZeroDimensional}(G)$;
true #i.e. the system has a finite number of solutions
$\mathrm{solve}(G, \{x, z, y\})$; # The zero set $\mathrm{Z}(J)$ then is
$\{\{y = 0, z = 0, x = 1\}, \{x = 0, y = 1, z = 0\}, \{z = 1, x = 0, y = 0\},$
$\{z = 1, x = 0, y = 0\}, \{x = \mathit{RootOf}(Z^2 + 2Z - 1, \mathit{label} =_L 1), y = x; z = x\}\}.$

If we use Singular instead [43], the function $\mathit{RootOf}()$ will returns up to complex roots. Kindly see more in [Arjeh Cohen, [4]], [Alicia Dickenstein, [6]] and [Sturmfels, [8]].

APPENDIX B: Permutation group

Given a set X, a *permutation* of X is a bijection from X to itself. We write $\mathrm{Sym}(X)$ for the *symmetric group* on X, ie, the group of all permutations of X. We denote Sym_N instead of $\mathrm{Sym}(\{1, 2, \ldots, N\})$, for a natural number N. We write elements of Sym_N in *cycle notation*, so the permutation $p = (1, 2, 3)(4, 5)$ is defined by $1^p = 2,\ 2^p = 3,\ 3^p = 1,\ 4^p = 5,\ 5^p = 4$. We say a group K *acts* on a set X if we have a group homomorphism $\phi : K \to \mathrm{Sym}(X)$. We abbreviate $x^{\phi(g)}$ by x^g. Let $p \in \mathrm{Sym}_N$. The *action of* p *on a subset* $B \subseteq \{1, 2, \ldots, N\}$ is given by $B^p := \{x^p : x \in B\}$. The *action of* p *on a list* of length N is given by

$$[y_1, y_2, \ldots, y_N]^p := [y_{1^{p^{-1}}}, y_{2^{p^{-1}}}, \ldots, y_{N^{p^{-1}}}].$$

In other words, we compute the ith position of Y^p by $Y^p[i] = y_{i^{p^{-1}}} = Y[i^{p^{-1}}]$.

APPENDIX C: Proof of Lemma 2.1

Proof. Pick up a nontrivial permutation h in $L(F_1)$. Then it acts locally on $\mathrm{RowInd}(F_1)$. By symmetry, we only check that $\mathrm{Z}_X(u_1) \subseteq \mathrm{Z}_Y(v_1)$. We choose any sub-vector

$$\boldsymbol{u}^* := [u_2; \ldots; u_{r_1}] \in \mathrm{Z}_X(u_1)$$

then $X = [u_1; u_2; \ldots; u_{r_1}]$ is in $\mathrm{Z}(P)$. We view $h \in \mathrm{Aut}(F)$, so

$$\begin{aligned} D^h = [F|X]^h &= \left[F^h|\, X^h\right] = \left[F|\, X^h\right] = \left[F|\, [u_1; u_2; \ldots; u_{r_1}]^h\right] \\ &= \left[F|\, [u_1^h; u_2; \ldots; u_{r_1}]\right] = \left[F|\, [v_1; u_2; \ldots; u_{r_1}]\right]. \end{aligned}$$

This implies that $[v_1; u_2; \ldots; u_{r_1}]$ is a solution, hence $u^* \in \mathrm{Z}_Y(v_1)$. □

APPENDIX D: LINGO environtment for ILP

We used two structures

SETS: ... ENDSETS to set the name and denote all variables.

DATA: ... ENDDATA to put the value that we get from the question.

To express constraints we need LINGO commands below:

- $@SUM$: adds all the numbers or variables together.
- $MIN =$: finds the minimum value of the objective behind the equal sign.
- $@FOR$: gives specific condition to some variables or equations.

Here is the LINGO code for our problem with data given in Figure 2.

```
! LINGO code for a balanced source-sink plan in Logistics using ILP;

SETS:
Cannery: CanProduce, Output;
Warehouse: WarProduce, Allocation;
Links(Cannery,Warehouse): ShipCost, Ship;    ! W and X;
ENDSETS

DATA:
! the Canneries (source) output;
Cannery, Output=
C1    75
C2    125
C3    100;

! the Warehouses (sink) output;
Warehouse, Allocation=
W1    80
W2    65
W3    70
W4    85;

! the shipping cost per trucload as given in the ship cost matrix W;
ShipCost =
464   513   654   867
352   416   690   791
995   682   388   685;
ENDDATA

! Minimize total cost Z;
MIN = @SUM(Links: ShipCost*Ship);

! the Canneries (source) constraints;
@FOR(Cannery(i):
 @SUM(Warehouse(j): Ship(i,j)) = Output(i));

! the Warehouses (sink) constraints;
@FOR(Warehouse(j):
 @SUM(Cannery(i): Ship(i,j)) = Allocation(j));
```

References

[1] A. R. Ravindran. *Operations research and management science handbook*, 2008, Taylor & Francis, CRC Press

[2] A. Morgan. *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, Classics in Applied Mathematics, SIAM 2009

[3] Online `https://en.wikipedia.org/wiki/Algebraic-statistics`

[4] Arjeh M. Cohen, Hans Cuypers and Hans Sterk. *Some Tapas of Computer Algebra*, Springer 1999

[5] Arjeh M. Cohen. *Computer algebra in industry: Problem Solving in Practice*, Wiley, 1993

[6] Alicia Dickenstein and Ioannis Z. Emiris. *Solving Polynomial Equations*, Springer 2005

[7] Online `www.risc.jku.at/conferences/ab2007/` and `www.springer.com/gp/book/9783540851004`

[8] Bernd Sturmfels. *Solving Systems of Polynomial Equations*, CBMS Regional Conference Series in Mathematics Volume: 97; AMS 2002

[9] Brouwer, Andries E., Cohen, Arjeh M. and Nguyen, Man V. M.
Orthogonal arrays of strength 3 and small run sizes, Journal of Statistical Planning and Inference, Vol 136, Issue 9, (2006) pp. 3268-3280.

[10] Friesland Campina Thailand,

`https://www.frieslandcampina.com/en/organisation/organogram/frieslandcampina-thailand/`

[11] Online `https://en.wikipedia.org/wiki/Computer-algebra`

[12] David Cox, John Little and Donal O'Shea. *Ideals, Varieties, and Algorithms: An introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer 1997

[13] David J. Rader. *Deterministic operations research: models and methods in linear optimization*, John Wiley & Sons, 2010

[14] Diaconis P. and Sturmfels B. *Algebraic Algorithms for sampling from conditional distribution*, Ann. Statistics **26**, 363-397, 1998

[15] Eric D. Schoen, Pieter T. Eendebak and Nguyen, Man V. M.
Complete Enumeration of Pure-Level & Mixed-Level Orthogonal Arrays, Journal of Combinatorial Designs, 18(2) (2010) pp. 123 - 140, Wiley.

[16] Frank Garvan. *The Maple book*, Chapman & Hall/CRC (2002)

[17] Hoang V. Dinh. *Using Maple to test Dixmier conjecture and Nguyen Huu Anhs conjecture for degree 6 and 9 polynomials*, University of Science, VNUHCM, 2008

[18] G. Pistone and H.P. Wynn. *Generalized confounding with Grobner bases*, Biometrika, **83**, 653- 666, 1996

[19] Pistone G., Riccomagno, E. and Wynn, H. P. *Computational commutative algebra in discrete statistics*, in **Algebraic methods in statistics and probability**, Contemporary Mathematics, **287**, 267–282, Amer. Math. Soc.

[20] Pistone G., Riccomagno, E., Wynn, H. P. *Algebraics statistics*, CRC, 2001

[21] J. Matousek and Bernd Gartner. *Understanding and Using Linear Programming*, Springer-Verlag, 2007

[22] Online `https://en.wikipedia.org/wiki/Jacobian-conjecture`

[23] Online `https://www.gap-system.org/`

[24] Online **www.kpack.com.vn/home**

[25] Lior Pachter and Bernd Sturmfels, Editors. *Algebraic Statistics for Computational Biology*, Cambridge University Press, (2005), `http://bio.math.berkeley.edu/ascb/`

[26] Man V.M. Nguyen and Phan Phuc Doan. *A Combined Approach to Damage Identification for Bridge*, Proceeding of the 5th Asian Mathematical Conference, pp 629- 636, (2009), Universiti Sains Malaysia in collaboration with UNESCO, Malaysia

[27] Man V.M. Nguyen and Tran Vinh Tan. *Selecting Meaningful Predictor Variables: A Case Study with Bridge Monitoring Data*, Proceeding of the First Regional Conference on Applied and Engineering Mathematics (RCAEM I) (2010), University of Perlis, Malaysia.

[28] Man V.M. Nguyen, Scott Murray and Thien An Ngoc Vo.
Mixed Orthogonal Arrays: Constructions and Applications,
International Conference on Advances in Probability and Statistics - Theory and Applications, 2011, The Chinese University of Hong Kong.

[29] Man V.M. Nguyen and Scott H. Murray.
Algebraic Methods for Construction of Mixed Orthogonal Arrays,
Southeast Asian Journal of Sciences, Vol 1, No. 2 (2012) pp. 155-168; ISSN 2286-7724, link
`science.utcc.ac.th/sajs/wp-content/uploads/2013/06/3-MinhMan.pdf`.

[30] Man VM. Nguyen, Tran V. Tan and Phan P. Doan. *Statistical Clustering and Time Series Analysis for Bridge Monitoring Data*, Lecture Notes in Electrical Engineering 156, (2013) pp. 61 - 72, Springer.

[31] Man V.M. Nguyen.
Permutation Groups and Integer Linear Algebra for Enumeration of Orthogonal Arrays, East-West Journal of Mathematics, Vol. 15, No 2 (2014)

[32] Man V.M. Nguyen, Linh V. Huynh. *Algebraic computation for few operational research problems*, Technical report (in Vietnamese), HCMUT, 2006

[33] Man V.M. Nguyen. *Some New Constructions of strength 3 Orthogonal Arrays*, the Memphis 2005 Design Conference Special Issue of the J. of Statistical Planning and Inference, Vol 138, Issue 1 (Jan 2008) pp. 220-233.

[34] Michael P. Barnett, Princeton University. *Computer Algebra in the life sciences*, ACM SIGSAM Bulletin, Vol. 36, No 4 (2002)

[35] Maynard V. Olson et al., National Academy of Sciences, The U.S. Government. *Mathematics and 21st Century Biology* (2005)

[36] Nguyen, V. M. Man. *About Jacques Dixmier's conjecture on Weil algebra* $A_1(K)$, M.Sc thesis, 1998, School of Natural Sciences, Vietnam National University in HCM City, Vietnam.

[37] Nguyen, VM Man. *Computer-Algebraic Methods for the Construction of Designs of Experiments*, PhD. thesis 2005, Technische Univ. Eindhoven.

[38] C.R. Rao. Factorial experiments derivable from combinatorial arrangements of arrays, Suppl. J. Roy. Statistics Soc., vol 9, pp. 128- 139, 1947

[39] Ron S. Kenett, Shelemyahu Zacks. *Modern Industrial Statistics with applications in R, MINITAB*, 2nd edition, (2014), Wiley

[40] Online `https://news.samsung.com/global/samsung-announces-new-and-enhanced-quality-assurance-measures-to-improve-product-safety`

[41] Online `www.samsungengineering.com/sustainability/quality/common/suView`

[42] Serge Lang, *Algebra*, Third Edition, *Springer* (2002)

[43] Online `https://www.singular.uni-kl.de/`

[44] Sohn, Hoon et al., *A Review of Structural Health Monitoring Literature*: 1996-2001. Los Alamos National Laboratory Report, 2004.

[45] Stephen Wolfram. *Mathematica book*, 5th edition, Wolfram Media (2003)

[46] Online `www.toyota-global.com/company/history-of-toyota/75years/data/company-information/management-and-finances/management/tqm/change.html`