Advanced Studies in Pure Mathematics 6, 1985
Algebraic Groups and Related Topics
pp. 515–523

Computation of New Schubert Tables for Quadrics and Projectivities

Corrado De Concini, Patrizia Gianni and Carlo Traverso

In the persent paper we show the details and the results of an implementation of algorithms described in [3], to compute: a) the number of quadrics in $\mathbf{P}^n$ with given tangency conditions, b) the number of projective transformations of $\mathbf{P}^n$ with prescribed incidence conditions.

Case a), for $n=2$, is classical and completely solved by Chasles, [2], and, for $n=3$, was computed by Schubert, [6]. In case b), partial results were due to Schubert and to Giambelli.

The algorithm a) has been run for $n=4$, 5, and the algorithm b) has been run for $n=3$, giving new Schubert numbers.

To our knowledge this is the first systematic extension of Schubert's results in enumerative geometry, and the number of quadrics tangent to 20 quadrics in $\mathbf{P}^5$ is by far the largest known result of this kind.

We recall a few facts: given a semisimple adjoint group G over $\mathbf{C}$, with Lie algebra $\mathfrak{g}$ and an order 2 automorphism $\sigma\colon G\to G$, if we let $H=G^\sigma$ one can construct a canonical projective G-equivariant embedding X of G/H which is defined as follows: let $\mathfrak{h}\subseteq\mathfrak{g}$ be the Lie algebra of H; $\mathfrak{h}$ can be considered as a point in a suitable Grassman variety of linear subspaces of $\mathfrak{g}$; then X is defined as the closure of the G-orbit of $\mathfrak{h}$.

X enjoys some remakable properties: X is smooth, $X-G/H=\bigcup S_i$, where S_i are G-stable smooth divisors which meet transversally; $\bigcap S_i$ is non-empty and is the unique closed orbit of X.

In the classical cases considered in this paper, the construction of X and its properties were known, (see [5] to [10],) and in the case of quadrics (resp. projectivities) X is classically called the variety of complete quadrics (resp. projectivities). We refer to [3] for the general setting and the geometrical interpretation of our algorithm; here we concentrate on the algebraic part.

The frame of the algorithm is as follows: the conditions are interpreted as product of "simple conditions", $x_1\cdots x_n$, which are seen as elements of

Received April 3, 1984.
This work is part of a research supported by C.N.R. (G.N.A.S.A.G.A and Progetto Finalizzato Trasporti).

Pic $X=H^2(X, Z)$. To evaluate the number of elements satisfying the condition $x_1^{m_1}\cdots x_n^{m_n}$, where $N=m_1+\cdots+m_n=\dim X$, one has to evaluate the corresponding element of $H^{2N}(X)$ on the class of a point. This can be reduced to the evaluation of the classes $s_1\cdots s_n x_1^{k_1}\cdots x_n^{k_n}$, where $s_1, \cdots, s_n$ are the cohomology classes of the divisors S_i of X. Since the closed orbit of X is the transversal intersection of S_i, one has to evaluate $x_1^{k_1}\cdots x_n^{k_n}$ against a point in the cohomology of the closed orbit. An explicit description of the closed orbit and of the "simple conditions" on it allows to compute explicitly the result.

We do not distinguish in the notation a cohomology class in the highest cohomology group and its evaluation against a point whenever the situation is clear.

We are considering two cases:

a) We want to compute the number of quadrics in $\mathbf{P}^n$ satisfying $N=(n+1)(n+2)/2-1$ simple tangency conditions, where a simple tangency condition is either

i) tangency to a given k-dimensional linear subspace,

ii) tangency to a given quadric.

This is obtained taking $G=\mathrm{PSL}(n+1)$, σ is the involution induced on G by the involution σ' of SL $N(n+1)$, $\sigma'(A)={}^tA^{-1}$.

With x_{k+1} we indicate the element in Pic X corresponding to the tangency to a k-dimensional subspace. Then the tangency to a given quadric hypersurface is given by $2(x_1+\cdots+x_n)$. The problem is thus solved evaluating all monomials $x_1^{m_1}\cdots x_n^{m_n}$ with $N=\sum m_i$. In this case the s_i are $s_i=2x_i-x_{i-1}-x_{i+1}$ (where $x_0=x_{n+1}=0$), and the closed orbit is the complete flag variety in $\mathbf{P}^n$ (see [3], [10]); the class $x_1^{k_1}\cdots x_n^{k_n}$, $\sum k_i=N-n$, on the flag variety corresponds to $2^{N-n}\xi_1^{k_1}\cdots\xi_n^{k_n}$, where ξ_i is the class corresponding to the fundamental weight w_i for $\mathrm{SL}(n+1)$, [3], i.e. ξ_i represents the elementary condition on the flag variety determinated by the flags whose i-1th dimensional subspace meets a fixed n-i-dimensional subspace.

b) We want to compute the number of projectivities of $\mathbf{P}^n$ satisfying $N=n^2-1$ simple conditions, where the simple condition x_k corresponds to the condition that the transform of a given $k-1$-dimensional subspace meets another given $n-k$-dimensional subspace.

This is obtained taking $G=\mathrm{PSL}(n+1)\times\mathrm{PSL}(n+1)$ and the involution σ is defined putting $\sigma(A, B)=(B, A)$.

Here again $s_i=2x_i-x_{i-1}-x_{i+1}(x_0=x_n=0)$ but the closed orbit here is the product of two complete flag varieties in $\mathbf{P}^n$. The class $x_1^{k_1}\cdots x_n^{k_n}$, $k_i=N-n$, on this product variety, corresponds to $(\xi_1\otimes 1+1\otimes\xi_1)^{k_1}\cdots(\xi_n\otimes 1+1\otimes\xi_n)^{k_n}$.

The algorithm in both cases follows three steps:

1) Evaluation of $\xi_1^{k_1}\cdots\xi_n^{k_n}$ for the flag variety;
2) evaluation of $x_1^{k_1}\cdots x_n^{k_n}$ for the closed orbit;
3) evaluation of $x_1^{m_1}\cdots x_n^{m_n}$ for X.

Step 1 is the same for both algorithms, step 2 is different (and quite elementary), step 3 is formally the same with one difference that we shall pinpoint when discussing it.

To accomplish step 1, one has to consider the following facts, [1]: $-\xi_1^n\xi_2^{n-1}\cdots\xi_n$ evaluates to 1, $-$ setting $\xi_i=y_1+\cdots+y_i$ (hence $y_i=\xi_i-\xi_{i-1}$, with $\xi_0=0$), and defining $\Psi_k(T_1\cdots T_n)$ being the polynomial sum of all monomials in $T_1\cdots T_k$ of degree $n-k+1$, we have $\Psi_k(y_1,\cdots,y_k)=0$, Substituting $y_i=\xi_i-\xi_{i-1}$, one obtains relations $\xi_k^{n-k}=\Phi_k(\xi_1,\cdots,\xi_k)$ and Φ_k has degree in ξ_k lower than $n-k$. Using these relations as substitution rules, one can evaluate all monomials of degree $n(n+1)/2$ in the ξ_i. One may simplify considerably the algorithm, by introducing several conditions on $k_1\cdots k_n$ that insure that $\xi_1^{k_1}\cdots\xi_n^{k_n}$ vanishes.

Step 2 is trivial for algorithm a); for algorithm b), to evaluate $x_1^{k_1}\cdots x_n^{k_n}$, $\sum k_i=n(n+1)$, substitute $x_i=(\xi_i\otimes 1+1\otimes\xi_i)$; then

$$(x_1^{k_1}\cdots x_n^{k_n})=\sum_{m_1\cdots m_n}\prod\binom{k_i}{m_i}(\xi_1^{m_1}\cdots\xi_n^{m_n})\otimes(\xi_1^{k_1-m_1}\cdots\xi_n^{k_n-m_n})$$

and to evaluate just replace $\otimes$ with multiplication. To simplify the computation one may remark that if $m_1+\cdots+m_n>n(n+1)/2$ then $\xi_1^{m_1}\cdots\xi_n^{m_n}=0$, and if $m_1+\cdots+m_n<n(n+1)/2$ then $\xi_1^{k_1-m_1}\cdots\xi_n^{k_n-m_n}=0$, hence the sum is extended to $m_1+\cdots+m_n=n(n+1)/2$. Many other vanishing conditions on $k_1,\cdots,k_n$ are easy to obtain; for example if $k_i\leqslant 1$, then $x_1^{k_1}\cdots x_n^{k_n}$ vanishes.

To accomplish Step 3, one has to evaluate $x_1^{m_1}\cdots x_n^{m_n}$, with $N=\sum m_i$, and $N=(n+1)(n+2)/2$, in case a), $N=n^2-1$ in case b).

To do this, one evaluates all $s_1^{\varepsilon_1}\cdots s_n^{\varepsilon_n}x_1^{k_1}\cdots x_n^{k_n}$, with $\sum\varepsilon_i+\sum k_i=N$, $\varepsilon_i\in\{0,1\}$, The evaluation is made through a recursive formula, equal for both algorithm, and taking in account various vanishing conditions. The first one, (which is essential, since insures that the recursion can be made, and is the only necessary one), is equal for both cases and states that if, for a monomial $s_1^{\varepsilon_1}\cdots s_n^{\varepsilon_n}x_1^{k_1}\cdots x_n^{k_n}$ we have $\varepsilon_i=0\Rightarrow k_i=0$, and not all ε_i are equal to 1, then the monomial evaluates to 0. The other vanishing conditions are useful to shorten the algorithm, and are different for the two algorithms.

Now we define the recursion; take a monomial $M=s_1^{\varepsilon_1}\cdots s_n^{\varepsilon_n}x_1^{m_1}\cdots x_n^{m_n}$. We proceed as follows:

a) if $\varepsilon_i=1$ $\forall i$, then look Step 2,
b) otherwise, check vanishing conditions,

c) if these fail, find i such that $\varepsilon_i=0$, $m_i\neq 0$; then define $y_1, \cdots, y_n$, putting $y_j=x_j$ if $\varepsilon_j=1$, $y_j=s_j$ if $\varepsilon_j=0$. Then express $x_j=\sum a_j y_j$. Define M' such that $x_i M'=M$. We call M the "father", and $M_j=M'y_j$ the "ith-child of M" Then all "children" of M are monomials of the considered type, and, evaluating $M=\sum a_j M_j$.

Remark that the a_j are rational numbers, but M evaluates to an integer, so it is convenient, to avoid rational arithmetic or floating point approximations, to find integer multiples $a'_j=da_j$, and to evaluate $M=(\sum a'_j M_j)/d$, (this is, in effect, what the encoded algorithms shown in the appendix do).

It is easy to show that the recursion converges in a finite number of steps. In practice, the recursion may be organized in different ways:

a) one may organize a big recursion machine, incorporating all the steps of the algorithm,

b) one may tabulate the $s_1 \cdots s_n x_1^{k_1} \cdots x_n^{k_n}$.

The latter way is quite convenient, since many vanishing conditions reduce the table of results to a quite manageable size, and otherwise one has to recompute the same monomials again and again.

So the recursions is broken in two steps. The single step may be organized:

a) in a purely recursive way, which bounds the storage requirements, but duplicates many computations,

b) in a recurrent way, by organizing an "evaluation path" which ensures that, when one evaluates the "father", the "children" have already been evaluated. This method is quite convenient from the point of view of computation, but requires a large amount of storage, and gives a complicate encoding, with needs of storage maps and evaluation path.

c) a compromise may be the following: use pure recursion, but keep track of intermediate results to avoid computing duplications. This method does not avoid storage need and storage maps, but allows to design a more transparent code, with a lower price in terms of execution speed.

The choice among the methods is essentially a matter of evaluation of the resources; recursion may be preferred either in low dimension where the execution time is anyway low, or in high dimension, where the physical limitations of storage do not allow to keep track of all computations already done.

We have done the computations in many different ways, also to allow to check the results.

With these computations we have found that the number of quadrics in $\mathbf{P}^4$ tangent to 14 quadrics is 48942189946470400 and the number of quadrics in $\mathbf{P}^5$ tangent to 20 quadrics is 641211464734373953791690014720. In Table 1 and 2 we give the complete results for quadrics in $\mathbf{P}^4$ with given tangency conditions to linear spaces, and for projectivities in $\mathbf{P}^3$ with given incidence conditions. The table for quadrics in $\mathbf{P}^5$ (10626 entries) is available upon request.

Different versions of the programs were written in PASCAL and FORTRAN; they were tested on IBM 3033 and VAX 780; the computation of the table for quadrics in $\mathbf{P}^5$ lasted two minutes; all other programs have run in few seconds.

References

[1] Borel, A., Sur la cohomologie des espaces fibrés principaux et des espaces homogènes de groupes de Lie compacts, Ann. of Math., **57** (1953), 116–207.

[2] Chasles, M., Comptes Rendues de l'Académie des Sciences de Paris 1864.

[3] DeConcini, C. and Procesi, C., Complete symmetric varieties; Springer Lecture Notes, **996** 1983.

[4] ——, Complete Symmetric Varieties II (Intersection Theory), this volume.

[5] Demazure, M., Limites de groupes ortogonaux ou symplectiques, Preprint 1980.

[6] Schubert, H., Kalkü der abzählenden Geometrie, Leipzig 1897 (reprinted Springer Verlag 1979).

[7] Semple, J. G., On complete quadrics I, J. London Math. Soc., **23** (1948), 258–267.

[8] ——, The variety whose points represent complete collineations of S_r on S_r' Rend. Mat., **10** (1951), 201–280.

[9] ——, On complete quadrics II, J. London Math. Soc., **27** (1952), 280–287.

[10] Tyrell, J. A., Complete quadrics and collineations in S_n. Mathematika, **3** (1956), 69–79.

[11] Vaisencher, I., Schubert calculus for complete quadrics, Preprint.

Corrado De Concini
Istituto Matematico della II Università
Via Orazio Raimondo
I–00173 ROMA

Partrizia Gianni
Carlo Traverso
Dipartimento di Matematica
Via Buonarroti 2
I–56100 PISA

TABLE 1. $(a\ b\ c)$=number of projectivities sending a points in a planes, b lines to meet b lines, c planes to pass through c points.

(0 0 15)= 1	(3 1 11)= 54	(7 1 7)= 286
(0 1 14)= 2	(3 2 10)= 108	(7 2 6)= 532
(0 2 13)= 4	(3 3 9)= 216	(7 3 5)= 824
(0 3 12)= 8	(3 4 8)= 432	(7 4 4)= 976
(0 4 11)= 16	(3 5 7)= 864	(7 5 3)= 864
(0 5 10)= 32	(3 6 6)=1568	(7 6 2)= 576
(0 6 9)= 64	(3 7 5)=2416	(7 7 1)= 384
(0 7 8)= 128	(3 8 4)=3080	(7 8 0)= 256
(0 8 7)= 256	(3 9 3)=3220	(8 0 7)= 143
(0 9 6)= 492	(3 10 2)=2816	(8 1 6)= 266
(0 10 5)= 864	(3 11 1)=2320	(8 2 5)= 412
(0 11 4)=1344	(3 12 0)=1832	(8 3 4)= 488
(0 12 3)=1832	(4 0 11)= 61	(8 4 3)= 432
(0 13 2)=2200	(4 1 10)= 122	(8 5 2)= 288
(0 14 1)=2384	(4 2 9)= 244	(8 6 1)= 192
(0 15 0)=2384	(4 3 8)= 488	(8 7 0)= 128
(1 0 14)= 3	(4 4 7)= 976	(9 0 6)= 133
(1 1 13)= 6	(4 5 6)=1752	(9 1 5)= 206
(1 2 12)= 12	(4 6 5)=2624	(9 2 4)= 244
(1 3 11)= 24	(4 7 4)=3184	(9 3 3)= 216
(1 4 10)= 48	(4 8 3)=3080	(9 4 2)= 144
(1 5 9)= 96	(4 9 2)=2412	(9 5 1)= 96
(1 6 8)= 192	(4 10 1)=1824	(9 6 0)= 64
(1 7 7)= 384	(4 11 0)=1344	(10 0 5)= 103
(1 8 6)= 728	(5 0 10)= 103	(10 1 4)= 122
(1 9 5)=1236	(5 1 9)= 206	(10 2 3)= 108
(1 10 4)=1824	(5 2 8)= 412	(10 3 2)= 72
(1 11 3)= 232	(5 3 7)= 824	(10 4 1)= 48
(1 12 2)=2568	(5 4 6)=1488	(10 5 0)= 32
(1 13 1)=2568	(5 5 5)=2216	(11 0 4)= 61
(1 14 0)=2384	(5 6 4)=2624	(11 1 3)= 54
(2 0 13)= 9	(5 7 3)=2416	(11 2 2)= 36
(2 1 12)= 18	(5 8 2)=1744	(11 3 1)= 24
(2 2 11)= 36	(5 9 1)=1236	(11 4 0)= 16
(2 3 10)= 72	(5 10 0)= 864	(12 0 3)= 27
(2 4 9)= 144	(6 0 9)= 133	(12 1 2)= 18
(2 5 8)= 288	(6 1 8)= 266	(12 2 1)= 12
(2 6 7)= 576	(6 2 7)= 532	(12 3 0)= 8
(2 7 6)=1072	(6 3 6)= 984	(13 0 2)= 9
(2 8 5)=1744	(6 4 5)=1488	(13 1 1)= 6
(2 9 4)=2412	(6 5 4)=1752	(13 2 0)= 4
(2 10 3)=2816	(6 6 3)=1568	(14 0 1)= 3
(2 11 2)=2816	(6 7 2)=1072	(14 1 0)= 2
(2 12 1)=2568	(6 8 1)= 728	(15 0 0)= 1
(2 13 0)=2200	(6 9 0)= 492	
(3 0 12)= 27	(7 0 8)= 143	

TABLE 2. (a, b, c, d)=number of quadrics in P^4 through a points, tangent to b lines, c planes and d spaces.

(0, 0, 0,14)=(14, 0, 0, 0)=	1	(0, 2, 4, 8)=(8, 4, 2, 0)=	144	(0, 5, 1, 8)=(8, 1, 5, 0)=	486	(0, 9, 1, 4)=(4, 1, 9, 0)=	15784
(0, 0, 1,13)=(13, 1, 0, 0)=	2	(0, 2, 5, 7)=(7, 5, 2, 0)=	288	(0, 5, 2, 7)=(7, 2, 5, 0)=	972	(0, 9, 2, 3)=(3, 2, 9, 0)=	23840
(0, 0, 2,12)=(12, 2, 0, 0)=	4	(0, 2, 6, 6)=(6, 6, 2, 0)=	576	(0, 5, 3, 6)=(6, 3, 5, 0)=	1944	(0, 9, 3, 2)=(2, 3, 9, 0)=	28096
(0, 0, 3,11)=(11, 3, 0, 0)=	8	(0, 2, 7, 5)=(5, 7, 2, 0)=	1152	(0, 5, 4, 5)=(5, 4, 5, 0)=	3888	(0, 9, 4, 1)=(1, 4, 9, 0)=	25440
(0, 0, 4,10)=(10, 4, 0, 0)=	16	(0, 2, 8, 4)=(4, 8, 2, 0)=	2240	(0, 5, 5, 4)=(4, 5, 5, 0)=	7264	(0,10, 0, 4)=(4, 0,10, 0)=	14993
(0, 0, 5, 9)=(9, 5, 0, 0)=	32	(0, 2, 9, 3)=(3, 9, 2, 0)=	4064	(0, 5, 6, 3)=(3, 6, 5, 0)=	11968	(0,10, 1, 3)=(3, 1,10, 0)=	22626
(0, 0, 6, 8)=(8, 6, 0, 0)=	64	(0, 2,10, 2)=(2,10, 2, 0)=	6672	(0, 5, 7, 2)=(2, 7, 5, 0)=	16896	(0,10, 2, 2)=(2, 2,10, 0)=	26172
(0, 0, 7, 7)=(7, 7, 0, 0)=	128	(0, 2,11, 1)=(1,11, 2, 0)=	9784	(0, 5, 8, 1)=(1, 8, 5, 0)=	20352	(0,10, 3, 1)=(1, 3,10, 0)=	22784
(0, 0, 8, 6)=(6, 8, 0, 0)=	256	(0, 2,12, 0)=(0,12, 2, 0)=	12988	(0, 5, 9, 0)=(0, 9, 5, 0)=	22464	(0,11, 0, 3)=(3, 0,11, 0)=	20089
(0, 0, 9, 5)=(5, 9, 0, 0)=	512	(0, 3, 0,11)=(11, 0, 3, 0)=	27	(0, 6, 0, 8)=(8, 0, 6, 0)=	694	(0,11, 1, 2)=(2, 1,11, 0)=	22938
(0, 0,10, 4)=(4,10, 0, 0)=	1008	(0, 3, 1,10)=(10, 1, 3, 0)=	54	(0, 6, 1, 7)=(7, 1, 6, 0)=	1388	(0,11, 2, 1)=(1, 2,11, 0)=	19396
(0, 0,11, 3)=(3,11, 0, 0)=	1896	(0, 3, 2, 9)=(9, 2, 3, 0)=	108	(0, 6, 2, 6)=(6, 2, 6, 0)=	2776	(0,12, 0, 2)=(2, 0,12, 0)=	19167
(0, 0,12, 2)=(2,12, 0, 0)=	3312	(0, 3, 3, 8)=(8, 3, 3, 0)=	216	(0, 6, 3, 5)=(5, 3, 6, 0)=	5552	(0,12, 1, 1)=(1, 1,12, 0)=	15854
(0, 0,13, 1)=(1,13, 0, 0)=	5284	(0, 3, 4, 7)=(7, 4, 3, 0)=	432	(0, 6, 4, 4)=(4, 4, 6, 0)=	10208	(0,13, 0, 1)=(1, 0,13, 0)=	12541
(0, 0,14, 0)=(0,14, 0, 0)=	7703	(0, 3, 5, 6)=(6, 5, 3, 0)=	864	(0, 6, 5, 3)=(3, 5, 6, 0)=	16192	(1, 0, 0,13)=(13, 0, 0, 1)=	4
(0, 1, 0,13)=(13, 0, 1, 0)=	3	(0, 3, 6, 5)=(5, 6, 3, 0)=	1728	(0, 6, 6, 2)=(2, 6, 6, 0)=	21504	(1, 0, 1,12)=(12, 1, 0, 1)=	8
(0, 1, 1,12)=(12, 1, 1, 0)=	6	(0, 3, 7, 4)=(4, 7, 3, 0)=	3328	(0, 6, 7, 1)=(1, 7, 6, 0)=	23808	(1, 0, 2,11)=(11, 2, 0, 1)=	16
(0, 1, 2,11)=(11, 2, 1, 0)=	12	(0, 3, 8, 3)=(3, 8, 3, 0)=	5888	(0, 6, 8, 0)=(0, 8, 6, 0)=	24576	(1, 0, 3,10)=(10, 3, 0, 1)=	32
(0, 1, 3,10)=(10, 3, 1, 0)=	24	(0, 3, 9, 2)=(2, 9, 3, 0)=	9280	(0, 7, 0, 7)=(7, 0, 7, 0)=	1802	(1, 0, 4, 9)=(9, 4, 0, 1)=	64
(0, 1, 4, 9)=(9, 4, 1, 0)=	48	(0, 3,10, 1)=(1,10, 3, 0)=	12896	(0, 7, 1, 6)=(6, 1, 7, 0)=	3604	(1, 0, 5, 8)=(8, 5, 0, 1)=	128
(0, 1, 5, 8)=(8, 5, 1, 0)=	96	(1, 3,11, 0)=(0,11, 3, 0)=	16192	(0, 7, 2, 5)=(5, 2, 7, 0)=	7208	(1, 0, 6, 7)=(7, 6, 0, 1)=	256
(0, 1, 6, 7)=(7, 6, 1, 0)=	192	(0, 4, 0,10)=(10, 0, 4, 0)=	81	(0, 7, 3, 4)=(4, 3, 7, 0)=	13136	(1, 0, 7, 6)=(6, 7, 0, 1)=	512
(0, 1, 7, 6)=(6, 7, 1, 0)=	384	(0, 4, 1, 9)=(9, 1, 4, 0)=	162	(0, 7, 4, 3)=(3, 4, 7, 0)=	20256	(1, 0, 8, 5)=(5, 8, 0, 1)=	1024
(0, 1, 8, 5)=(5, 8, 1, 0)=	768	(0, 4, 2, 8)=(8, 2, 4, 0)=	324	(0, 7, 5, 2)=(2, 5, 7, 0)=	25536	(1, 0, 9, 4)=(4, 9, 0, 1)=	2000
(0, 1, 9, 4)=(4, 9, 1, 0)=	1504	(0, 4, 3, 7)=(7, 3, 4, 0)=	648	(0, 7, 6, 1)=(1, 6, 7, 0)=	26112	(1, 0,10, 3)=(3,10, 0, 1)=	3672
(0, 1,10, 3)=(3,10, 1, 0)=	2784	(0, 4, 4, 6)=(6, 4, 4, 0)=	1296	(0, 7, 7, 0)=	25344	(1, 0,11, 2)=(2,11, 0, 1)=	6144
(0, 1,11, 2)=(2,11, 1, 0)=	4728	(0, 4, 5, 5)=(5, 5, 4, 0)=	2592	(0, 8, 0, 6)=(6, 0, 8, 0)=	4166	(1, 0,12, 1)=(1,12, 0, 1)=	9228
(0, 1,12, 1)=(1,12, 1, 0)=	7256	(0, 4, 6, 4)=(4, 6, 4, 0)=	4928	(0, 8, 1, 5)=(5, 1, 8, 0)=	8332	(1, 1, 0,12)=(12, 0, 1, 1)=	12
(0, 1,13, 0)=(0,13, 1, 0)=	10122	(0, 4, 7, 3)=(3, 7, 4, 0)=	8448	(0, 8, 2, 4)=(4, 2, 8, 0)=	15192	(1, 1, 1,11)=(11, 1, 1, 1)=	24
(0, 2, 0,12)=(12, 0, 2, 0)=	9	(0, 4, 8, 2)=(2, 8, 4, 0)=	12672	(0, 8, 3, 3)=(3, 3, 8, 0)=	23056	(1, 1, 2,10)=(10, 2, 1, 1)=	48
(0, 2, 1,11)=(11, 1, 2, 0)=	18	(0, 4, 9, 1)=(1, 9, 4, 0)=	16512	(0, 8, 4, 2)=(2, 4, 8, 0)=	27936	(1, 1, 3, 9)=(9, 3, 1, 1)=	96
(0, 2, 2,10)=(10, 2, 2, 0)=	36	(0, 4,10, 0)=(0,10, 4, 0)=	19488	(0, 8, 5, 1)=(1, 5, 8, 0)=	26688	(1, 1, 4, 8)=(8, 4, 1, 1)=	192
(0, 2, 3, 9)=(9, 3, 2, 0)=	72	(0, 5, 0, 9)=(9, 0, 5, 0)=	243	(0, 9, 0, 5)=(5, 0, 9, 0)=	8628	(1, 1, 5, 7)=(7, 5, 1, 1)=	384

TABLE 2. (Continued).

(1, 1, 6, 6)=(6, 6, 1, 1)= 768	(1, 4, 6, 3)=(3, 6, 4, 1)=15488	(2, 0, 0,12)=(12, 0, 0, 2)= 16	(2, 3, 3, 6)=(6, 3, 3, 2)= 3456
(1, 1, 7, 5)=(5, 7, 1, 1)= 1536	(1, 4, 7, 2)=(2, 7, 4, 1)=21120	(2, 0, 1,11)=(11, 1, 0, 2)= 32	(2, 3, 4, 5)=(5, 4, 3, 2)= 6912
(1, 1, 8, 4)=(4, 8, 1, 1)= 2976	(1, 4, 8, 1)=(1, 8, 4, 1)=24192	(2, 0, 2,10)=(10, 2, 0, 2)= 64	(2, 3, 5, 4)=(4, 5, 3, 2)=12672
(1, 1, 9, 3)=(3, 9, 1, 1)= 5344	(1, 5, 0, 8)=(8, 0, 5, 1)= 902	(2, 0, 3, 9)=(9, 3, 0, 2)= 128	(2, 3, 6, 3)=(3, 6, 3, 2)=19968
(1, 1,10, 2)=(2,10, 1, 1)= 8616	(1, 5, 1, 7)=(7, 1, 5, 1)= 1804	(2, 0, 4, 8)=(8, 4, 0, 2)= 256	(2, 3, 7, 2)=(2, 7, 3, 2)=26176
(1, 1,11, 1)=(1,11, 1, 1)=12312	(1, 5, 2, 6)=(6, 2, 5, 1)= 3608	(2, 0, 5, 7)=(7, 5, 0, 2)= 512	(2, 4, 0, 8)=(8, 0, 4, 2)= 1156
(1, 2, 0,11)=(11, 0, 2, 1)= 36	(1, 5, 3, 5)=(5, 3, 5, 1)= 7216	(2, 0, 6, 6)=(6, 6, 0, 2)= 1024	(2, 4, 1, 7)=(7, 1, 4, 2)= 2312
(1, 2, 1,10)=(10, 1, 2, 1)= 72	(1, 5, 4, 4)=(4, 4, 5, 1)=13152	(2, 0, 7, 5)=(5, 7, 0, 2)= 2048	(2, 4, 2, 6)=(6, 2, 4, 2)= 4624
(1, 2, 2, 9)=(9, 2, 2, 1)= 144	(1, 5, 5, 3)=(3, 5, 5, 1)=20416	(2, 0, 8, 4)=(4, 8, 0, 2)= 3952	(2, 4, 3, 5)=(5, 3, 4, 2)= 9248
(1, 2, 3, 8)=(8, 3, 2, 1)= 288	(1, 5, 6, 2)=(2, 6, 5, 1)=26112	(2, 0, 9, 3)=(3, 9, 0, 2)= 7016	(2, 4, 4, 4)=(4, 4, 4, 2)=16704
(1, 2, 4, 7)=(7, 4, 2, 1)= 576	(1, 5, 7, 1)=(1, 7, 5, 1)=27264	(2, 0,10, 2)=(2,10, 0, 2)=11088	(2, 4, 5, 3)=(3, 5, 4, 2)=25344
(1, 2, 5, 6)=(6, 5, 2, 1)= 1152	(1, 6, 0, 7)=(7, 0, 6, 1)= 2216	(2, 1, 0,11)=(11, 0, 1, 2)= 48	(2, 4, 6, 2)=(2, 6, 4, 2)=31104
(1, 2, 6, 5)=(5, 6, 2, 1)= 2304	(1, 6, 1, 6)=(6, 1, 6, 1)= 4432	(0, 1, 1,10)=(10, 1, 1, 2)= 96	(2, 5, 0, 7)=(7, 0, 5, 2)= 2628
(1, 2, 7, 4)=(4, 7, 2, 1)= 4416	(1, 6, 2, 5)=(5, 2, 6, 1)= 8864	(2, 1, 2, 9)=(9, 2, 1, 2)= 192	(2, 5, 1, 6)=(6, 1, 5, 2)= 5256
(1, 2, 8, 3)=(3, 8, 2, 1)= 7712	(1, 6, 3, 4)=(4, 3, 6, 1)=16064	(2, 1, 3, 8)=(8, 3, 1, 2)= 384	(2, 5, 2, 5)=(5, 2, 5, 2)=10512
(1, 2, 9, 2)=(2, 9, 2, 1)=11888	(1, 6, 4, 3)=(3, 4, 6, 1)=24320	(2, 1, 4, 7)=(7, 4, 1, 2)= 768	(2, 5, 3, 4)=(4, 3, 5, 2)=18976
(1, 2,10, 1)=(1,10, 2, 1)=16008	(1, 6, 5, 2)=(2, 5, 6, 1)=29568	(2, 1, 5, 6)=(6, 5, 1, 2)= 1536	(2, 5, 4, 3)=(3, 4, 5, 2)=28224
(1, 3, 0,10)=(10, 0, 3, 1)= 108	(1, 6, 6, 1)= 28416	(2, 1, 6, 5)=(5, 6, 1, 2)= 3072	(2, 5, 5, 2)= 33024
(1, 3, 1, 9)=(9, 1, 3, 1)= 216	(1, 7, 0, 6)=(6, 0, 7, 1)= 4728	(2, 1, 7, 4)=(4, 7, 1, 2)= 5856	(2, 6, 0, 6)=(6, 0, 6, 2)= 5024
(1, 3, 2, 8)=(8, 2, 3, 1)= 432	(1, 7, 1, 5)=(5, 1, 7, 1)= 9456	(2, 1, 8, 3)=(3, 8, 1, 2)=10080	(2, 6, 1, 5)=(5, 1, 6, 2)=10048
(1, 3, 3, 7)=(7, 3, 3, 1)= 864	(1, 7, 2, 4)=(4, 2, 7, 1)=17248	(2, 1, 9, 2)=(2, 9, 1, 2)=15160	(2, 6, 2, 4)=(4, 2, 6, 2)=18432
(1, 3, 4, 6)=(6, 4, 3, 1)= 1728	(1, 7, 3, 3)=(3, 3, 7, 1)=25856	(2, 2, 0,10)=(10, 0, 2, 2)= 144	(2, 6, 3, 3)=(3, 3, 6, 2)=27392
(1, 3, 5, 5)=(5, 5, 3, 1)= 3456	(1, 7, 4, 2)=(2, 4, 7, 1)=30336	(2, 2, 1, 9)=(9, 1, 2, 2)= 288	(2, 7, 0, 5)=(5, 0, 7, 2)= 8392
(1, 3, 6, 4)=(4, 6, 3, 1)= 6528	(1, 8, 0, 5)=(5, 0, 8, 1)= 8924	(2, 2, 2, 8)=(8, 2, 2, 2)= 576	(2, 7, 1, 4)=(4, 1, 7, 2)=15504
(1, 3, 7, 3)=(3, 7, 3, 1)=11008	(1, 8, 1, 4)=(4, 1, 8, 1)=16376	(2, 2, 3, 7)=(7, 3, 2, 2)= 1152	(2, 7, 2, 3)=(3, 2, 7, 2)=23392
(1, 3, 8, 2)=(2, 8, 3, 1)=16064	(1, 8, 2, 3)=(3, 2, 8, 1)=24624	(2, 2, 4, 6)=(6, 4, 2, 2)= 2304	(2, 8, 0, 4)=(4, 0, 8, 2)=12028
(1, 3, 9, 1)=(1, 9, 3, 1)=20128	(1, 8, 3, 2)=(2, 3, 8, 1)=28256	(2, 2, 5, 5)=(5, 5, 2, 2)= 4608	(2, 8, 1, 3)=(3, 1, 8, 2)=18200
(1, 4, 0, 9)=(9, 0, 4, 1)= 324	(1, 9, 0, 4)=(4, 0, 9, 1)=14202	(2, 2, 6, 4)=(4, 6, 2, 2)= 8640	(2, 9, 0, 3)=(3, 0, 9, 2)=13692
(1, 4, 1, 8)=(8, 1, 4, 1)= 648	(1, 9, 1, 3)=(3, 1, 9, 1)=21412	(2, 2, 7, 3)=(3, 7, 7, 2)=14304	(3, 0, 0,11)=(11, 0, 0, 3)= 44
(1, 4, 2, 7)=(7, 2, 4, 1)= 1296	(1, 9, 2, 2)=(2, 2, 9, 1)=24248	(2, 2, 8, 2)=(2, 8, 2, 2)=20240	(3, 0, 1,10)=(10, 1, 0, 3)= 88
(1, 4, 3, 6)=(6, 3, 4, 1)= 2592	(1,10, 0, 3)=(3, 0,10, 1)=17552	(2, 3, 0, 9)=(9, 0, 3, 2)= 432	(3, 0, 2, 9)=(9, 2, 0, 3)= 176
(1, 4, 4, 5)=(5, 4, 4, 1)= 5184	(1,10, 1, 2)=(2, 1,10, 1)=19704	(2, 3, 1, 8)=(8, 1, 3, 2)= 864	(3, 0, 3, 8)=(8, 3, 0, 3)= 352
(1, 4, 5, 4)=(4, 5, 4, 1)= 9600	(1,11, 0, 2)=(2, 0,11, 1)=15396	(2, 3, 2, 7)=(7, 2, 3, 2)= 1728	(3, 0, 4, 7)=(7, 4, 0, 3)= 704

TABLE 2. (Continued).

(3, 0, 5, 6)=(6, 5, 0, 3)= 1408	(3, 3, 1, 7)=(7, 1, 3, 3)= 2096	(4, 0, 4, 6)=(6, 4, 0, 4)= 1376	(4, 5, 0, 5)=(5, 0, 5, 4)= 4048
(3, 0, 6, 5)=(5, 6, 0, 3)= 2816	(3, 3, 2, 6)=(6, 2, 3, 3)= 4192	(4, 0, 5, 5)=(5, 5, 0, 4)= 2752	(5, 0, 0, 9)=(9, 0, 0, 5)= 137
(3, 0, 7, 4)=(4, 7, 0, 3)= 5360	(3, 3, 3, 5)=(5, 3, 3, 3)= 8384	(4, 0, 6, 4)=(4, 6, 0, 4)= 5168	(5, 0, 1, 8)=(8, 1, 0, 5)= 274
(3, 0, 8, 3)=(3, 8, 0, 3)= 9184	(3, 3, 4, 4)=(4, 4, 3, 3)=15104	(4, 1, 0, 9)=(9, 0, 1, 4)= 258	(5, 0, 2, 7)=(7, 2, 0, 5)= 548
(3, 1, 0,10)=(10, 0, 1, 3)= 132	(3, 3, 5, 3)=(3, 5, 3, 3)=22720	(4, 1, 1, 8)=(8, 1, 1, 4)= 516	(5, 0, 3, 6)=(6, 3, 0, 5)= 1096
(3, 1, 1, 9)=(9, 1, 1, 3)= 264	(3, 4, 0, 7)=(7, 0, 4, 3)= 2304	(4, 1, 2, 7)=(7, 2, 1, 4)= 1032	(5, 0, 4, 5)=(5, 4, 0, 5)= 2192
(3, 1, 2, 8)=(8, 2, 1, 3)= 528	(3, 4, 1, 6)=(6, 1, 4, 3)= 4608	(4, 1, 3, 6)=(6, 3, 1, 4)= 2064	(5, 1, 0, 8)=(8, 0, 1, 5)= 376
(3, 1, 3, 7)=(7, 3, 1, 3)= 1056	(3, 4, 2, 5)=(5, 2, 4, 3)= 9216	(4, 1, 4, 5)=(5, 4, 1, 4)= 4128	(5, 1, 1, 7)=(7, 1, 1, 5)= 752
(3, 1, 4, 6)=(6, 4, 1, 3)= 2112	(3, 4, 3, 4)=(4, 3, 4, 3)=16640	(4, 1, 5, 4)=(4, 5, 1, 4)= 7584	(5, 1, 2, 6)=(6, 2, 1, 5)= 1504
(3, 1, 5, 5)=(5, 5, 1, 3)= 4224	(3, 4, 4, 3)= =24576	(4, 2, 0, 8)=(8, 0, 2, 4)= 704	(5, 1, 3, 5)=(5, 3, 1, 5)= 3008
(3, 1, 6, 4)=(4, 6, 1, 3)= 7904	(3, 5, 0, 6)=(6, 0, 5, 3)= 4152	(4, 2, 1, 7)=(7, 2, 1, 4)= 1408	(5, 2, 0, 7)=(7, 0, 2, 5)= 848
(3, 1, 7, 3)=(3, 7, 1, 3)=13008	(3, 5, 1, 5)=(5, 1, 5, 3)= 8304	(4, 2, 2, 6)=(6, 2, 2, 4)= 2816	(5, 2, 1, 6)=(6, 1, 2, 5)= 1696
(3, 2, 0, 9)=(9, 0, 2, 3)= 396	(3, 5, 2, 4)=(4, 2, 5, 3)=15328	(4, 2, 3, 5)=(5, 3, 2, 4)= 5632	(5, 2, 2, 5)= 3392
(3, 2, 1, 8)=(8, 1, 2, 3)= 792	(3, 6, 0, 5)=(5, 0, 6, 3)= 6416	(4, 2, 4, 4)=(4, 4, 2, 4)=10176	(5, 3, 0, 6)=(6, 0, 3, 5)= 1504
(3, 2, 2, 7)=(7, 2, 2, 3)= 1584	(3, 6, 1, 4)=(4, 1, 6, 3)=11936	(4, 3, 0, 7)=(7, 0, 3, 4)= 1552	(6, 0, 0, 8)=(8, 0, 0, 6)= 188
(3, 2, 3, 6)=(6, 3, 2, 3)= 3168	(3, 7, 0, 4)=(4, 0, 7, 3)= 8552	(4, 3, 1, 6)=(6, 1, 3, 4)= 3104	(6, 0, 1, 7)=(7, 1, 0, 6)= 376
(3, 2, 4, 5)=(5, 4, 2, 3)= 6336	(4, 0, 0,10)=(10, 0, 0, 4)= 86	(4, 3, 2, 5)=(5, 2, 3, 4)= 6208	(6, 0, 2, 6)=(6, 2, 0, 6)= 752
(3, 2, 5, 4)=(4, 5, 2, 3)=11584	(4, 0, 1, 9)=(9, 1, 0, 4)= 172	(4, 3, 3, 4)= 11264	(6, 1, 0, 7)=(7, 0, 1, 6)= 424
(3, 2, 6, 3)=(3, 6, 2, 3)=18112	(4, 0, 2, 8)=(8, 2, 0, 4)= 344	(4, 4, 0, 6)=(6, 0, 4, 4)= 2736	(6, 1, 1, 6)= 848
(3, 3, 0, 8)=(8, 0, 3, 3)= 1048	(4, 0, 3, 7)=(7, 3, 0, 4)= 688	(4, 4, 1, 5)=(5, 1, 4, 4)= 5472	(7, 0, 0, 7)= 212