Lotto Design Tables

P. C. Li and G. H. J. van Rees*

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
Canada R3T 2N2
lipakc@cs.umanitoba.ca vanrees@cs.umanitoba.ca

Abstract

An $LD(n, k, p, t; b)$ lotto design is a set of b k-sets (blocks) of an n-set such that any p-set intersects at least one k-set in t or more elements. Let $L(n, k, p, t)$ denote the minimum number of blocks in any $LD(n, k, p, t; b)$ lotto design. We will list the known lower and upper bound theorems for lotto designs. Since many of these bounds are recursive, we will incorporate this information in a set of tables for lower and upper bounds for lotto designs with small parameters. We will also use back-track algorithms, greedy algorithms and simulated annealing to improve the tables.

1 Introduction

There are many lotteries in the world. They tend to be very similar in operation. The way most lotteries work is as follows. For a small fee, a person chooses k numbers from n numbers or has the numbers chosen randomly for her/him. This is the ticket. At a certain point no more tickets are sold and the government or casino picks p numbers from the n numbers, usually in a random fashion. These are called

*research supported by NSERC grant OGP # 0003558

the winning numbers. If any of the tickets sold match t or more of the winning numbers, then a prize is given to the holder of the matching ticket. The larger the value of t, the larger the prize. Usually t must be three or more to receive a prize. Many people and researchers are interested to know what is the minimum number of tickets necessary to ensure that at least one ticket will intersect the winning numbers in t or more numbers, for varying values of t. See the following for research papers on the subject [1], [2], [5], [6], [8]and [11]. For an enlightening overview on this subject see Colbourn and Dinitz[7].

More formally we may define an (n, k, p, t) lotto design to be a set of k-sets (also called blocks) of an n-set such that any p-set intersects at least one k-set in t or more elements. A p-set is said to be *represented by a block* B if B intersects the p-set in at least t elements. Further it is said that the p-set is *represented by the design* if it is represented by one block in the design. An LD$(n, k, p, t; b)$ is a (n, k, p, t) lotto design with exactly b blocks. Let $L(n, k, p, t)$ denote the minimum number of blocks in any $LD(n, k, p, t; b)$. Clearly these definitions mirror the lottery situation and the fundamental problem is to determine $L(n, k, p, t)$. The following is a typical example of a lotto design.

Example 1.1 *: The set of* 6*-sets* $\{\{1, 2, 3, 4, 5, 6\}$, $\{1, 2, 3, 4, 5, 7\}$, $\{1, 2, 3, 4, 6, 7\}$, $\{1, 2, 3, 5, 6, 7\}$, $\{8, 9, 10, 11, 12, 13\}\}$ *forms an* $LD(13, 6, 5, 3; 5)$ *lotto design.*

There are many theoretical results known about the upper and lower bounds for $L(n, k, p, t)$. Many of these results are recursive. There are various computer results known, some published, some not. It is getting difficult to know if a particular result is new or not. So the focus in this paper is to produce tables of upper and lower bounds for $L(n, k, p, t)$, for small parameters. We have written two papers [15] [16] that give lower and upper bounds on lotto designs. Most of the known results can be found in these papers. There are also some other results in the thesis of Li [13] that we will discuss briefly in this paper. We will also have a section in this paper on computer algorithms for finding $L(n, k, p, t)$. All this information will then be incorporated into the tables. We have decided that we would calculate and maintain all bounds for the parameters up to $n = 20$.

Two special subclasses of lotto designs are covering designs and Turán designs. A (n, k, t) covering design is an (n, k, t, t) lotto design and a (n, p, t) Turán design is an (n, t, p, t) lotto design. The minimum number of blocks in any (n, k, t) covering design is denoted by $C(n, k, t)$ and the minimum number of blocks of any (n, p, t) Turán design is denoted by $T(n, p, t)$. The complement of a covering design is a Turán design. Both covering and Turán designs have been thoroughly studied. Gordon et al. [9] [10] have excellent tables on covering designs so we will not produce lotto design numbers with these parameters.

Another subclass of lotto designs that have been studied extensively (see Füredi et al. [8]) are lotto designs with $t = 2$. Their properties are closely related to graph properties and they have obtained good non-recursive results.

Since there too many entries to publish in a journal, we have decided that only the most interesting parameters could be published in this paper but that the interested reader could get all the numbers from the following web site, [14]. This web site will be up-dated as new results are made known to us. In this paper, we publish upper and lower bounds for $t = 2, 3,\ 4$; $p = t + 1, \ldots, 10$; $k = t + 1 \ldots, 12$ and $n = p + 1, \ldots, 20$.

2 Theoretical Results

Li's thesis [13] contains most of the theoretical results known about lotto designs. Most of these results have been published in Li & van Rees [15] [16]. However, this leaves a few results that merit discussion.

There are a set of theorems that we call the Monotonicity Theorems. They show the relationship in size between $L(n, k, p, t)$ and $L(n+a, k+b, p+c, t+d)$ where $a, b, c, d \in 0, 1$. They are easy to prove (see Li [13]) but quite useful. We will prove just one of the results. There should be 15 of them but one is missing. It is missing because $L(n, k, p, t)$ can not be directly compared with $L(n, k+1, p+1, t+1)$. Sometimes it is bigger, sometimes less. It would be nice to have a theorem of some sort relating these two numbers.

Theorem 2.1 $L(n, k, p, t) \leq L(n + 1, k, p, t)$.
$L(n, k, p, t) \geq L(n, k, p + 1, t)$.
$L(n, k, p, t) \geq L(n + 1, k + 1, p, t)$.
$L(n, k, p, t) \geq L(n + 1, k + 1, p + 1, t)$.
$L(n, k, p, t) \geq L(n, k + 1, p, t)$.
$L(n, k, p, t) \leq L(n, k, p, t + 1)$.
$L(n, k, p, t) \leq L(n + 1, k + 1, p + 1, t + 1)$.

$L(n,k,p,t) \leq L(n+1,k+1,p,t+1)$.
$L(n,k,p,t) \leq L(n+1,k,p,t+1)$.
$L(n,k,p,t) \leq L(n,k+1,p,t+1)$.
$L(n,k,p,t) \leq L(n,k,p+1,t+1)$.
$L(n,k,p,t) \leq L(n+1,k,p+1,t+1)$.
$L(n,k,p,t) \geq L(n+1,k,p+1,t)$.
$L(n,k,p,t) \geq L(n,k+1,p+1,t)$.

Proof of $L(n,k,p,t) \leq L(n,k+1,p,t+1)$. Let $\mathcal{B}$ be the blocks of an $(n,k+1,p,t+1)$ lotto design. Let $\mathcal{B}_1 = \{B' : B' = B \setminus \{\text{the largest element in B}\}, B \in \mathcal{B}\}$. Consider an arbitrary set, P, of size p. It intersects some block of $\mathcal{B}$ in at least $t+1$ elements. So it intersects the corresponding block in $\mathcal{B}_1$ in at least t elements. Hence $\mathcal{B}_1$ is a (n,k,p,t) lotto design. □

The following lower bound appears in Li [13] and is a generalization of Bate's result [1].

Theorem 2.2 *If $k_1 > k_2$, then $L(n,k_1,p_1,t) \geq \frac{L(n,k_2,p,t)}{L(k_1,k_2,t,t)}$.*

Proof Consider a (n,k_1,p_1,t) lotto design $\mathcal{B}$ containing b blocks. For each of its b blocks, construct a (k_1,k_2,t,t) lotto design. Let $\mathcal{C}$ denote the set of all k_2-sets in the b (k_1,k_2,t,t) lotto designs. Then $\mathcal{C}$ is an (n,k_2,p,t) lotto design. To show this consider an arbitrary set P of size p. P must intersect some block of $\mathcal{B}$ in t elements as $\mathcal{B}$ is an (n,k_1,p,t) lotto design. These t elements must be a block in $\mathcal{C}$ by definition. Hence $\mathcal{C}$ is an (n,k_2,p,t) lotto design. So $L(n,k_2,p,t) \leq L(n,k_1,p,t) * L(k_1,k_2,t,t)$. □

The next three theorems are closely linked.

Theorem 2.3 *If $L(n,k,p,t) = b < \frac{n}{k}$, then $L(n,k,p,t) \geq L(n-1,k,p-1,t)$.*

Proof Assume that $b < L(n-1,k,p-1,t)$. Then there exists an (n,k,p,t) lotto design with b blocks whose elements have average frequency in the design of $\frac{kb}{n} < \frac{kn}{nk} = 1$. This means that there is an element of frequency 0 in the lotto design. These blocks can be considered a $(n-1,k,p-1,t)$ lotto design. Hence there exists an $(n-1,k,p-1,t)$ lotto design on b blocks. This is a contradiction as we assumed $b < L(n-1,k,p-1,t)$. □

Corollary 2.1 *If* $L(n,k,p,t) = b < \frac{n}{k}$*, then* $L(n,k,p,t) = L(n-1,k,p-1,t)$.

Proof Apply Theorem 2.3 and Theorem 2.1. □

Theorem 2.4 *If* $L(n,k,p,t) = b < \frac{2n}{k}$*, then* $L(n,k,p,t) \geq L(n-k+t-2,k,p-1,t)+1$.

Proof Assume $b < L(n-k+t-2,k,p-1,t)+1$. Then there exists an (n,k,p,t) lotto design on b blocks with elements whose average frequency in the design is $\frac{kb}{nk} < \frac{k2n}{nk} = 2$. This means that there is at least one element of frequency 0 or 1 in the design. As in the previous theorem, if there is an element of frequency 0, then there exists an $(n-1,k,p-1,t)$ lotto design on $b < L(n-k+t-2,k,p-1,t)+1$ blocks. But by the generalized Schönheim bound (listed as number 30 in Figure 1), $L(n-k+t-2,k,p-1,t)+1 \leq L(n-1,k,p-1,t)$. This is a contradiction. If there is an element, x, of frequency 1, then consider the blocks, B, that do not contain x. Delete $k-t+2$ of the elements in B that occur with x and replace them with arbitrary elements. These blocks can be considered to be a $(n-k+t-2,k,p-1,t)$ lotto design. So there exists an $(n-k+t-2,k,p-1,t)$ lotto design on $b-1 < L(n-k+t-2,k,p-1,t)$ blocks which is a contradiction. □

Theorem 2.5 *If* $L(n,k,p,t) = b < \frac{3n}{k}$*, then* $L(n,k,p,t) \geq \min(\frac{2n}{k}, L(n-k+t-2,k,p-1,t)+1)$.

Proof Assume $b < min(\frac{2n}{k}, L(n-k+t-2,k,p-1,t)+1)$. Then there exists an (n,k,p,t) lotto design on b blocks with elements whose average frequency in the design is less than 3. This means that there is an element of frequency 0, 1 or 2 in the design. As in the previous theorem, if there is an element of frequency 0, then there exists an $(n-1,k,p-1,t)$ lotto design on $b < L(n-k+t-2,k,p-1,t)+1$ blocks. Again, by the generalized Schönheim bound (listed in Figure 1), $L(n-k+t-2,k,p-1,t)+1 < L(n-1,k,p-1,t)$, which is a contradiction. Again as in the previous theorem, if there is an element of frequency 1, then there exists an $(n-k+t-2,k,p-1,t)$ lotto design on $b-1 < L(n-k+t-2,k,p-1,t)$ blocks which is also a contradiction. Therefore, every element has frequency at least two and so there are at least $\frac{2n}{k}$ blocks in the design which again is a contradiction. □

The next theorem lists 27 results from Li's thesis [13] that were proved by hand in mostly case work. Because the proofs are not that interesting and are repetitive, they are left out in this paper. However these cases ensure that all values of $L(n, k, p, 2)$ are solved for parameters less than 21 and $t = 2$.

Theorem 2.6 *L(14,3,4,2)=11, L(16,3,4,2)=14, L(18,3,4,2)=17, L(19,3,4,2)=18, L(20,3,4,2)=20, L(18,3,5,2)=13, L(19,3,5,2)=15, L(20,2,5,2)=16, L(13,4,3,2)=8, L(14,4,3,2)=9, L(15,4,3,2)=11, L(16,4,4,2=12), L(17,4,3,2)=14, L(18,4,3,2)=15, L(19,4,3,2)=16, L(20,4,3,2)=18, L(19,4,4,2)=11, L(20,4,4,2)=12, L(15,5,3,2)=7, L(16,5,3,2)=8, L(17,5,3,2)=9, L(18,5,3,2)=10, L(19,5,3,2)=11, L(20,5,3,2)=12, L(20,6,3,2)=8, L(11,5,4,3) $\geq$ 8, L(9,4,4,3)=9.*

3 Computer Algorithms

We use three types of algorithms to improve our tables. They are exhaustive backtracking, greedy methods and simulated annealing. These techniques are very similar to the techniques described by Gordon et al. [9]. The ideas of combinatorial computing can be found in the excellent book of Kreher & Stinson [12].

Exhaustive backtrack algorithms for computing lotto designs date back to Bate [1]. The idea is to find an uncovered p-set. Then find all k-sets or blocks that represent that p-set. Choose one of these blocks to add to the design. Repeat until a lotto design is found or there are no more blocks to choose. If there are no more blocks to choose at the r^{th} level, then reject the current choice of block at the $(r-1)^{th}$ level and choose a different block there. In this way all possible sets of blocks can be generated. To find a minimal block design, one backtracks as soon as the number of blocks in the design reaches the current upper bound.

This is a very slow algorithm only suitable for very small parameters. Furthermore, a lot of memory may be required to store all data structures required by the algorithm.

Bate did two things to speed up his search. First, he used a simplified version of isomorphic rejection. When adding the r^{th} block to the design, a check is made to see if these r blocks are isomorphic to some previously considered r blocks. If so, the block is rejected as the r^{th} block. Complete isomorphism testing is expensive so Bate only checked easy to find isomorphisms. Secondly, he used look-ahead to see which set of blocks were never going to produce a minimal design. He did this by dividing how many p-sets were represented by a block into the number of p-sets not yet represented to get a lower bound on the number of blocks still needed. If this number is added the level number and it is bigger than the current upper bound, rejection may immediately take place.

To conserve memory, both the blocks of a design and p-sets were represented by integers. Each p-set and k-set were assigned a unique integer called an index. To do this, good algorithms are needed to relate a number and a set of n numbers. These exist but space is being traded for time. These methods of Bate, even when updated, are still very slow and only suitable for small parameters. However, these algorithms are easily parallelized and slightly larger parameters could be attacked in the future.

Another exhaustive algorithm that we investigated was integer programming. It is easy to turn the lotto design problem into an integer programming problem. The CPLEX linear programming package was used to evaluate small parameters. It was used sequentially but not with much success. The only result computed using this technique was $L(10, 4, 5, 3) = 7$. Perhaps the parallel version would do better.

Unfortunately, exhaustive backtrack is the only algorithm that we know that gives lower bounds. Therefore, it is worthwhile to spend time on these algorithms to provide careful and efficient implementations. It would certainly be worthwhile to write a parallel exhaustive search that would improve many of the lower bounds in this and expanded tables.

Since exhaustive techniques are too slow, greedy algorithms were used to generate better upper bounds for the tables. The idea of a greedy algorithm is to order the p-sets and blocks. Then pick the first block in the ordered list that represents the most unrepresented p-sets. This is really fast compared to the exhaustive techniques but is not guaranteed to give a minimal lotto design. This technique requires a fair bit of memory to run and has to do a fair bit of work finding the block which represents the most unrepresented k-sets. Again, this can be speeded up for the cost of more memory.

There are many orderings that we can choose for the blocks. We have implemented the greedy algorithms with lexicographical, reverse lexicographical, co-lexicographical and random orderings. Each of these orderings were useful. Greedy algorithm found many of the best upper bounds. It takes a great deal of time with simulated annealing, our next algorithm, to improve on its results.

The last type of search probably holds out the most hope for getting really good results. But at a great cost. These are the heuristic techniques such as hill-climbing, simulating annealing and tabu search. Simulated annealing has been extensively applied to a good proportion of the table. Bluskov [3] got some of his good results using simulated annealing. We have re-implemented Nurmela and Ostergard's [17] simulated annealing algorithm to get many of the results in our tables.

The idea of simulated annealing is to pick b random blocks. Of course, many p-sets are not represented. We then think about replacing a particular random block in the solution with a neigbouring block. If the number of unrepresented p-sets decreases, this is done. If not, then there is a small probability, which decreases over time, that we will accept it anyways. This way the algorithm will, hopefully, not get stuck in a local minimum. After a fixed number of iterations, we stop with no design found or before that we find one with b blocks and then we try the algorithm with $b-1$ blocks. This algorithm is quite costly in time and space, but does give better and better results if enough computing power is used. Simulated annealing is the most useful overall algorithm for upper bounds. Parallel simulated annealing is not well-understood. The side-effects of communication between processors is not known. There is the possibility in these parallel algorithms to get very good speed-ups.

Nevertheless, one algorithm will not be suitable for all parameters. Lotto designs are just too large a problem. Although an algorithm may do quite well with one set of parameters, other parameters will be totally inappropriate to try. This means that there must be a variety of algorithms to obtain better bounds just as there must be a variety of theoretical constructions to produce good highly-structured lotto designs. We suggest that $LD(n,5,4,3)'s$ are a good place to try to improve the tables. The number of blocks is relatively small but the gap between the upper and lower bound is large.

4 Tables

In the tables, There are upper and lower bounds for $t = 3,4,5$, $p = t+1,\ldots,10$, $k = t+1\ldots,12$, and $n = p+1,\ldots,20$. For each table, the value of p and t are fixed and the values of n and k vary. The column labels represent the value of n and the row labels represent the value of k. The upper entry is the lower bound and the lower entry is the upper bound. Any row which consists entirely of the value 1 or NA (not applicable) are not displayed.

There is a subscript for each entry telling how the number was obtained. This subscript could refer to a theorem in one of our two previous papers, a theorem in this paper or refer to an entry in Gordon's tables or Bate's tables or refer to a result reported to us by other researchers such as Bluskov [1]. We restate all these theorems. For some entries we could make several references. We have chosen the reference that is most convenient for us. This reference is not necessarily the first one in the literature. Some numbers are missing from the list. That is because there was a theorem or technique encoded that became redundant.

When a new result is put into the table, all known recursive results must be re-applied to the tables. This is repeated until the tables no longer change. These tables are constantly changing and we encourage researchers with new results to contact us so their results may be incorporated into the tables.

We now list the meanings of the subscripts that occur in this paper and the web site.

0-The lower bound of 1 and the upper bound of $\binom{n}{k}$
1-$L(n,k,p,t) \leq \frac{\binom{n}{t}}{\binom{p-1}{t-1}\binom{k}{t}} \frac{n-p+1}{n-t+1}$. Theorem 2-6 in Li & van Rees [15]
2-$L(n,k,t,t) \geq \lceil \frac{n}{k} L(n-1,k-1,t-1,t-1) \rceil$ Schönheim Bound for Coverings, [18].
3-Tables of Bate [1]
4-Tables of Gordon, Patashnik & Kuperberg [9]
7-$L(n,k,p,t) = 1$ if $n-k \leq p-t$.
8-$L(n,k,p,t) \leq L(n_1,k,p_1,t) + L(n_2,k,p_2,t)$, if $n = n_1 + n_2$ and $p = p_1 + p_2 - 1$. Proposition 8.8 in CRC Handbook of Combinatorial Designs [7].
9-Tables of Bluskov [3]
10-$L(v,k,p,t) \leq \lfloor \frac{pr}{t-1} \rfloor \binom{t-1}{2} + \binom{pr - \lfloor \frac{pr}{t-1} \rfloor (t-1)}{2}$ if a (v,b,r,k,λ)-BIBD exists. $L(xk,k,p,t) \leq \frac{\lambda x(xk-1)}{k-1}$ if $(\lfloor \frac{p}{t-1} \binom{t-1}{2} + \binom{p - \frac{p}{t-1}(t-1)}{2}))\lambda \frac{xk-1}{k-1} < \lambda \binom{p}{2}$ and if a resolvable BIBD $(xk, \frac{\lambda x(xk-1)}{k-1}, \frac{\lambda(xk-1}{k-1}, k, \lambda)$ exists. $L(4a+3, 2a+1, 2t-2, t) \leq 4a+3$ if $a > t-2$ and if there exists a $(4a+3, 4a+3, 2a+1, 2a+1, a)$ BIBD. Theorems 3.1, 3.2, 3.3 in Li & van

Rees [16].
11-$L(n,k,p,t) = L(n, n-k, n-p, n-k-p+t)$. Corollary to Theorem 2.1.1 in Bate [1].
12-25-Theorem 2.1 in the order presented.
26-$L(n+1, k+1, p+1, t+1) \leq L(n,k,p,t) + L(n, k+1, p+1, t+1)$. Theorem 5.1 in Li & van Rees [16]
28-$L(n,k,p,t) = r+1$ if $n-k \geq p-t+1$ and $\lceil \frac{r}{r+1} n \rceil \leq k$ where $r = \lfloor \frac{p}{p-t+1} \rfloor$. Theorem 4.3 in Li & van Rees [15]
29-$L(n+1, k+1, p+1, t) \leq \lfloor 2 - \frac{n}{k} L(n,k,p,t) \rfloor + L(n-1, k+1, p+1, t)$. Theorem 5.6 in Li & van Rees [16].
30-$L(n,k,p,t) \geq \lceil \frac{\binom{n}{k-t+1} L(n-k-t-1,k,p,t)}{\binom{n}{k-t+1} - \binom{k}{k-t+1}} \rceil$. Generalized Schönheim Bound. Theorem 3.2 in Li & van Rees [15]
31-$L(n,3,3,2) = \lceil \frac{n^2-2n}{12} \rceil$ if $n \equiv 2,4,6 (\mathrm{mod} 12)$
$L(n,3,3,2) = \lceil \frac{n^2-2n}{12} \rceil + 1$ if $n \equiv 0,8,10 (\mathrm{mod} 12)$
$L(n,3,3,2) = \lceil \frac{n^2-n}{12} \rceil$ if $n \equiv 1,3,5,7 (\mathrm{mod} 12)$
$L(n,3,3,2) = \lceil \frac{n^2-n}{12} \rceil + 1$ if $n \equiv 9,11 (\mathrm{mod} 12)$
Bate's Theorem in Theorem 2.2 in Li & van Rees [15] or Brouwer [5].
33-$L(n,k,t+1,t) \geq \min(L(n,k,t-1,t-1), L(n-t+1, k-t+2, 2, 2))$.
$L(n,k,t+2,t) \geq \min(L(n,k,t,t-1), L(n-t, k-t+2, 2, 2))$.
$L(n,k,6,3) \geq \min(L(n,k,4,2), L(n-4, k-1, 2, 2))$.
Theorem 4.1, 4.2, 4.4 in Li & van Rees [15] .
34-Theorem 2.2
35-$L(mn, 2m, zn+2, 2z+2) \leq \binom{n}{2}$ for $z \leq m$ for m an integer. Theorem 5.5 in Li & van Rees [16].
36-$L(2k+1, k, 5, 3) \geq 5$ if $k \geq 5$. Theorem 4.5 in Li & van Rees [15].
$L(2k+2, k, 6, 3) \geq 4$ if $k \geq 3$. Theorem 4.6 in Li & van Rees [15].
$L(3k+1, k, 7, 3) \geq 6$ if $k \geq 3$. Theorem 4.7 in Li & van Rees [15].
$L(k+2, k, 4, 3) \geq 3$ if $k \geq 3$. Theorem 4.8 in Li & van Rees [15].
$L(3k+2, k, 8, 3) \geq 5$ if $k \geq 3$. Theorem 4.9 in Li & van Rees [15].
$L(3k+3, k, 9, 3) \geq 4$ if $k \geq 3$. Theorem 4.3.12 in Li [13].
$L(2k+3, k, 9, 4) \geq 4$ if $k \geq 4$. Theorem 4.10 in Li & van Rees [15].
$L(2k+2, k, 8, 4) \geq 4$ if $k \geq 4$. Theorem 4.3.14 in Li [13].
$L(3k+2, k, 11, 4) \geq 5$ if $k \geq 4$. Theorem 4.11 in Li & van Rees [15].
$L(3k+3, k, 12, 4) \geq 5$ if $k \geq 4$. Theorem 4.12 in Li & van Rees [15].
$L(2k+4, k, 12, 5) \geq 4$ if $k \geq 5$. Theorem 4.13 in Li & van Rees [15].
$L(2k+3, k, 11, 5) \geq 4$ if $k \geq 5$. Theorem 4.3.18 in Li [13].
37-$L(n,k,p,t) \geq \lceil \frac{L(n,t,p,t)}{\binom{k}{t}} \rceil$. Theorem 4.1 in Bate [1].
38-$L(n,k,p,t) \geq L(n-1, k, p-1, t)$ if $L(n,k,p,t) = b < \frac{n}{k}$.

$L(n,k,p,t) \geq L(n-k+t-2,k,p-1,t)+1$ if $L(n,k,p,t)=b<\frac{2n}{k}$.
$L(n,k,p,t) \geq \min(\frac{2n}{k}, L(n-1,k,p-1,t)+1$ if $L(n,k,p,t)=b<\frac{3n}{k}$.
Theorems 2.3, 2.4 & 2.5.
39-$L(n,k,p,t) \leq L(n_1,k,p-r,t)+L(n_1,k_1,p-r-1,t-r-1)$ if $n_1<n, p-r\geq t, k_1\geq t-r-1$ and $k_1=k-n+n_1$. Semi-direct Product. Theorem 4.1 in Li & van Rees [16].
40-Greedy Algorithms.
41-Simulated Annealing.
42-If $L(n,k,p,2)=b\geq \frac{n}{k}$ then $L(mn,mk,p,2)\leq b$. Theorem 5.4 in Li & van Rees [16].
43-Theorem 2.6.
44-Results from Boyer, Kreher, Radziszowski, Zou & Siderenko [4].
45-$L(3n,3,4,3)\leq 3\binom{n}{3}+3n\binom{n}{2}$.
$L(3n+1,3,4,3)\leq 2\binom{n}{3}+\binom{n+1}{3}+(2n+1)\binom{n+1}{2}+n\binom{n}{2}$.
$L(3n+2,3,4,3)\leq \binom{n}{3}+2\binom{n+1}{3}+(2n+1)\binom{n+1}{2}+(n+1)\binom{n}{2}$.
Turán [19] results stated in Li [13].
46-Linear Programming using CPLEX.
47-Exhaustive Backtrack Algorithm.
48-Genetic Simulated Annealing

5 Tables

n\k	3	4	5	6	7	8	9	10	11	12
5	2_{31} 2_{31}	1_0 1_7	1_0 1_0							
6	2_3 2_3	2_8 2_{11}	1_0 1_7	1_0 1_0						
7	4_3 4_3	2_8 2_{11}	2_8 2_{11}	1_0 1_7	1_0 1_0					
8	5_3 5_3	2_3 2_3	2_8 2_{11}	2_8 2_{11}	1_0 1_7	1_0 1_0				
9	7_3 7_3	4_3 4_3	2_3 2_3	2_8 2_{11}	2_8 2_{11}	1_0 1_7	1_0 1_0			
10	8_3 8_3	4_3 4_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_0		
11	10_3 10_3	6_3 6_3	4_3 4_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_0	
12	11_3 11_3	6_3 6_3	4_3 4_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_0
13	13_3 13_3	8_{43} 8_3	5_3 5_3	4_3 4_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7
14	14_3 14_3	9_{43} 9_3	6_{38} 6_3	4_3 4_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}
15	18_3 18_3	11_{43} 11_3	7_{43} 7_3	4_3 4_3	4_3 4_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}
16	19_3 19_3	12_{43} 12_3	8_{43} 8_3	5_3 5_3	4_3 4_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3
17	23_{31} 23_{31}	14_{43} 14_8	9_{43} 9_8	6_{38} 6_8	4_{12} 4_8	4_{38} 4_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
18	24_{31} 24_{31}	15_{43} 15_8	10_{43} 10_8	6_{38} 6_8	5_{38} 5_8	4_{38} 4_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	29_{31} 29_{31}	16_{43} 16_8	11_{43} 11_8	7_{38} 7_8	6_{38} 6_8	4_{38} 4_8	4_{38} 4_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	31_{31} 31_{31}	18_{43} 18_8	12_{43} 12_8	8_{43} 8_8	6_{38} 6_8	4_{37} 4_8	4_{38} 4_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}

p=3, t=2

n\k	3	4	5	6	7	8	9	10	11	12
6	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_0						
7	2_3 2_3	2_8 2_{11}	1_0 1_7	1_0 1_7	1_0 1_0					
8	3_3 3_3	2_3 2_3	2_8 2_{11}	1_0 1_7	1_0 1_7	1_0 1_0				
9	3_3 3_3	2_3 2_3	2_8 2_{11}	2_8 2_{11}	1_0 1_7	1_0 1_7	1_0 1_0			
10	5_3 5_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_0		
11	6_3 6_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_0	
12	8_{43} 8_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_0
13	9_3 9_3	5_3 5_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7
14	11_{43} 11_3	5_3 5_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7
15	12_3 12_3	7_{38} 7_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}
16	14_{43} 14_3	7_3 7_3	5_3 5_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}
17	15_8 15_8	9_{38} 9_8	5_{12} 5_8	3_{12} 3_8	3_{12} 3_{12}	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
18	17_{43} 17_8	9_{38} 9_8	6_{38} 6_8	3_{12} 3_8	3_{12} 3_8	3_{26} 3_{14}	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	18_{43} 18_8	11_{43} 11_8	7_{38} 7_8	5_{38} 5_8	3_{12} 3_8	3_{11} 3_{12}	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	20_{43} 20_8	12_{43} 12_8	8_{38} 8_8	5_{38} 5_8	3_{12} 3_8	3_{37} 3_8	3_{11} 3_{14}	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}

p=4, t=2

n\k	3	4	5	6	7	8	9	10	11	12
7	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0					
8	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0				
9	3_3 3_3	2_3 2_3	2_8 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0			
10	3_3 3_3	2_3 2_3	2_3 2_3	2_8 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0		
11	4_3 4_3	3_3 3_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0	
12	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0
13	6_3 6_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7
14	7_3 7_3	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7
15	9_{38} 9_3	4_3 4_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7
16	10_{38} 10_3	4_3 4_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}
17	12_{38} 12_8	6_{38} 6_8	4_{30} 4_8	3_{12} 3_8	3_{11} 3_{12}	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
18	13_{43} 13_8	6_{37} 6_8	4_{37} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	15_{43} 15_8	8_{38} 8_8	4_{37} 4_8	3_{12} 3_8	3_{11} 3_8	3_{11} 3_{12}	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	16_{43} 16_8	8_{38} 8_8	4_{37} 4_8	4_{11} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}

p=5, t=2

n\k	3	4	5	6	7	8	9	10	11	12
8	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0				
9	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0			
10	3_3 3_3	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0		
11	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0	
12	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0
13	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	5_3 5_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7
15	5_3 5_3	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7
16	7_3 7_3	4_3 4_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7
17	8_{38} 8_8	4_{12} 4_8	3_{12} 3_8	3_{12} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
18	10_{38} 10_8	5_{30} 5_8	4_{11} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	11_{38} 11_8	5_{30} 5_8	4_{11} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	13_{38} 13_8	5_{30} 5_8	4_{11} 4_8	3_{12} 3_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}

p=6, t=2

n\k	3	4	5	6	7	8	9	10	11	12
9	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0			
10	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0		
11	3_3 3_3	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0	
12	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0
13	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	5_3 5_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7
16	5_3 5_3	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7
17	6_{30} 6_8	4_{12} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	1_0 1_7
18	6_{30} 6_8	4_{12} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
19	8_{38} 8_8	5_{30} 5_8	4_{11} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	9_{38} 9_8	5_{30} 5_8	4_{11} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}

p=7, t=2

n\k	3	4	5	6	7	8	9	10	11	12
10	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0		
11	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0	
12	3_3 3_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0
13	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
16	5_3 5_3	3_3 3_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7
17	5_{12} 5_8	4_{30} 4_8	3_{12} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{11} 2_{11}	1_0 1_7	1_0 1_7
18	6_{30} 6_8	4_{30} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{11} 2_{11}	1_0 1_7
19	6_{30} 6_8	4_{30} 4_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
20	7_{30} 7_8	5_{30} 5_8	4_{11} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{12} 2_{11}

p=8, t=2

n\k	3	4	5	6	7	8	9	10	11	12
11	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0	
12	2_3 2_3	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0
13	3_3 3_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
16	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
17	5_{30} 5_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7
18	5_{30} 5_8	4_{30} 4_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7	1_0 1_7
19	6_{30} 6_8	4_{30} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{11} 2_{11}	1_0 1_7
20	6_{30} 6_8	4_{30} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{11} 2_{11}

p=9, t=2

n\k	3	4	5	6	7	8	9	10	11	12
12	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_0
13	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	3_3 3_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
16	4_3 4_3	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
17	4_{12} 4_8	3_{12} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
18	5_{30} 5_8	3_{12} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7
19	5_{30} 5_8	4_{30} 4_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7	1_0 1_7
20	6_{30} 6_8	4_{30} 4_8	3_{11} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7

p=10, t=2

n\k	3	4	5	6	7	8	9	10	11	12
13	2_8 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	3_3 3_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
16	3_3 3_3	2_3 2_3	2_3 2_3	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
17	4_{30} 4_8	3_{30} 3_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
18	4_{30} 4_8	3_{30} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
19	5_{30} 5_8	3_{30} 3_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7
20	5_{30} 5_8	4_{30} 4_8	3_{11} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7

p=11, t=2

n\k	4	5	6	7	8	9	10	11	12
6	3_{36} 3_{11}	1_{0} 1_{7}	1_{0} 1_{0}						
7	4_{11} 4_{11}	3_{36} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}					
8	6_{3} 6_{3}	3_{36} 3_{40}	3_{36} 3_{9}	1_{0} 1_{7}	1_{0} 1_{0}				
9	9_{43} 9_{41}	5_{11} 5_{9}	3_{36} 3_{9}	3_{36} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}			
10	12_{34} 14_{41}	7_{47} 7_{9}	4_{11} 4_{9}	3_{36} 3_{40}	3_{36} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}		
11	16_{37} 19_{41}	9_{47} 9_{9}	5_{33} 5_{9}	3_{36} 3_{40}	3_{36} 3_{40}	3_{36} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}	
12	21_{37} 26_{41}	11_{47} 12_{9}	6_{33} 6_{9}	4_{11} 4_{40}	3_{36} 3_{40}	3_{36} 3_{40}	3_{36} 3_{11}	1_{0} 1_{7}	1_{0} 1_{0}
13	28_{37} 33_{48}	11_{12} 16_{9}	8_{47} 9_{9}	6_{33} 6_{40}	4_{11} 4_{40}	3_{36} 3_{40}	3_{36} 3_{11}	3_{36} 3_{11}	1_{0} 1_{7}
14	35_{37} 43_{48}	14_{37} 20_{9}	8_{12} 11_{9}	6_{33} 6_{41}	5_{33} 5_{40}	3_{36} 3_{40}	3_{36} 3_{11}	3_{36} 3_{11}	3_{36} 3_{11}
15	44_{37} 52_{48}	18_{37} 24_{48}	10_{33} 14_{48}	7_{14} 9_{41}	6_{33} 6_{41}	4_{11} 4_{40}	3_{36} 3_{11}	3_{36} 3_{11}	3_{36} 3_{11}
16	54_{37} 66_{48}	22_{37} 32_{9}	11_{37} 16_{9}	7_{16} 11_{41}	7_{47} 7_{41}	5_{33} 5_{40}	4_{11} 4_{14}	3_{36} 3_{11}	3_{36} 3_{11}
17	66_{37} 80_{48}	27_{37} 40_{9}	14_{37} 21_{9}	9_{33} 14_{41}	7_{12} 9_{12}	6_{33} 6_{12}	4_{12} 4_{11}	3_{36} 3_{18}	3_{36} 3_{11}
18	79_{37} 103_{26}	32_{37} 49_{9}	16_{37} 25_{48}	10_{33} 19_{11}	7_{12} 9_{41}	6_{33} 6_{41}	5_{47} 5_{47}	4_{33} 4_{11}	3_{36} 3_{11}
19	94_{37} 124_{48}	38_{37} 57_{9}	19_{37} 32_{9}	11_{33} 20_{11}	9_{33} 14_{12}	7_{33} 9_{14}	6_{33} 6_{14}	5_{33} 5_{12}	4_{33} 4_{11}
20	111_{37} 148_{48}	45_{37} 70_{48}	23_{37} 38_{48}	13_{37} 22_{11}	9_{33} 14_{41}	7_{33} 11_{11}	6_{33} 8_{11}	5_{33} 5_{47}	4_{33} 4_{11}

p=4, t=3

n\k	4	5	6	7	8	9	10	11	12
7	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}					
8	2_{8} 2_{40}	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}				
9	5_{3} 5_{3}	2_{8} 2_{40}	2_{11} 2_{9}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}			
10	7_{46} 7_{40}	2_{3} 2_{3}	2_{11} 2_{9}	2_{11} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	10_{47} 10_{8}	5_{36} 5_{40}	2_{8} 2_{9}	2_{11} 2_{40}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	10_{12} 12_{8}	6_{47} 6_{40}	2_{8} 2_{9}	2_{11} 2_{40}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	13_{37} 13_{10}	8_{47} 8_{47}	5_{36} 5_{9}	2_{12} 2_{40}	2_{11} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
14	17_{37} 20_{8}	8_{12} 10_{8}	5_{36} 5_{9}	2_{12} 2_{40}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
15	21_{37} 26_{8}	9_{37} 13_{8}	7_{47} 8_{9}	5_{36} 5_{40}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}
16	26_{37} 28_{8}	11_{37} 16_{8}	7_{12} 8_{9}	5_{36} 5_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}
17	32_{37} 39_{8}	13_{37} 20_{8}	7_{37} 11_{9}	5_{36} 7_{8}	5_{36} 5_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
18	39_{37} 44_{8}	16_{37} 24_{8}	8_{37} 12_{9}	5_{36} 8_{8}	5_{36} 5_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	46_{37} 55_{8}	19_{37} 28_{8}	10_{37} 15_{9}	6_{37} 10_{8}	5_{36} 6_{8}	5_{36} 5_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	54_{37} 60_{8}	22_{37} 32_{8}	11_{37} 18_{9}	7_{37} 12_{8}	5_{36} 7_{8}	5_{36} 5_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=5, t=3

n\k	4	5	6	7	8	9	10	11	12
8	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}				
9	2_{8} 2_{40}	2_{11} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}			
10	4_{3} 4_{3}	2_{11} 2_{40}	2_{11} 2_{9}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	5_{11} 5_{40}	2_{3} 2_{3}	2_{11} 2_{9}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	7_{47} 7_{8}	4_{36} 4_{40}	2_{3} 2_{3}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	10_{47} 10_{8}	4_{36} 4_{40}	2_{12} 2_{9}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	10_{37} 12_{8}	6_{47} 6_{40}	4_{36} 4_{9}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
15	13_{37} 15_{8}	8_{47} 8_{8}	4_{36} 4_{9}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
16	16_{37} 20_{10}	8_{12} 10_{8}	5_{47} 5_{9}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}
17	19_{37} 23_{8}	8_{37} 12_{8}	6_{47} 6_{9}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
18	23_{37} 28_{8}	10_{37} 14_{8}	6_{24} 7_{9}	4_{36} 4_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	28_{37} 33_{8}	11_{37} 17_{8}	6_{13} 9_{9}	4_{36} 5_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	33_{37} 40_{8}	13_{37} 20_{8}	7_{37} 10_{9}	4_{36} 7_{8}	4_{36} 4_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=6, t=3

n\k	4	5	6	7	8	9	10	11	12
9	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}			
10	2_{8} 2_{40}	2_{11} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	3_{3} 3_{3}	2_{11} 2_{40}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	3_{3} 3_{3}	2_{3} 2_{3}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	6_{36} 6_{40}	3_{8} 3_{40}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	6_{36} 8_{40}	3_{8} 3_{40}	2_{12} 2_{8}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	9_{47} 10_{8}	3_{8} 3_{40}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
16	10_{37} 12_{8}	6_{36} 6_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
17	12_{37} 14_{8}	6_{36} 7_{8}	3_{18} 3_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
18	14_{37} 17_{8}	7_{38} 9_{8}	3_{18} 3_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	17_{37} 19_{8}	8_{38} 11_{8}	6_{36} 6_{8}	3_{18} 3_{8}	3_{11} 3_{14}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	20_{37} 25_{8}	8_{37} 13_{8}	6_{36} 6_{8}	3_{18} 3_{8}	3_{11} 3_{14}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=7, t=3

n\k	4	5	6	7	8	9	10	11	12
10	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	2_{8} 2_{40}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	3_{3} 3_{3}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	3_{3} 3_{3}	2_{3} 2_{3}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	5_{36} 5_{40}	3_{18} 3_{8}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	6_{47} 6_{40}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
16	7_{38} 8_{8}	3_{17} 3_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
17	8_{37} 11_{8}	5_{36} 5_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
18	9_{37} 13_{8}	5_{36} 5_{8}	3_{18} 3_{8}	3_{11} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
19	11_{37} 16_{8}	5_{36} 7_{8}	3_{18} 3_{8}	3_{11} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	13_{37} 17_{8}	7_{38} 9_{8}	5_{36} 5_{8}	3_{11} 3_{8}	3_{11} 3_{14}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=8, t=3

n\k	4	5	6	7	8	9	10	11	12
11	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	2_{8} 2_{40}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	3_{3} 3_{3}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	3_{3} 3_{3}	2_{3} 2_{3}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	4_{3} 4_{3}	3_{18} 3_{8}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
16	4_{3} 4_{3}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
17	6_{38} 7_{8}	3_{18} 3_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
18	7_{38} 9_{8}	4_{36} 4_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
19	8_{37} 11_{8}	4_{36} 4_{8}	3_{18} 3_{8}	3_{11} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
20	9_{37} 13_{8}	4_{36} 4_{8}	3_{18} 3_{8}	3_{11} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=9, t=3

n\k	4	5	6	7	8	9	10	11	12
12	2_{8} 2_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	2_{11} 2_{40}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	3_{3} 3_{3}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	3_{3} 3_{3}	2_{3} 2_{3}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
16	4_{3} 4_{3}	3_{18} 3_{8}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
17	4_{12} 4_{8}	3_{18} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
18	5_{30} 6_{8}	3_{18} 3_{8}	3_{30} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
19	6_{37} 7_{8}	4_{30} 4_{8}	3_{30} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
20	7_{37} 9_{8}	4_{30} 4_{8}	3_{30} 3_{8}	3_{11} 3_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}

p=10, t=3

n\k	4	5	6	7	8	9	10	11	12
13	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	2_{11} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	3_3 3_3	2_{11} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
16	3_3 3_3	2_3 2_3	2_{11} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
17	4_{18} 4_8	3_{18} 3_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
18	4_{18} 4_8	3_{18} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7
19	5_{30} 5_8	3_{18} 3_8	3_{30} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7	1_0 1_7
20	5_{30} 5_8	4_{30} 4_8	3_{30} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7

p=11, t=3

n\k	5	6	7	8	9	10	11	12
7	3_{8} 3_{9}	1_{0} 1_{7}	1_{0} 1_{0}					
8	5_{11} 5_{9}	3_{8} 3_{9}	1_{0} 1_{7}	1_{0} 1_{0}				
9	9_{11} 9_{9}	3_{8} 3_{9}	3_{8} 3_{11}	1_{0} 1_{7}	1_{0} 1_{0}			
10	10_{37} 14_{9}	7_{11} 7_{9}	3_{8} 3_{11}	3_{8} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}		
11	17_{37} 22_{9}	7_{12} 10_{9}	5_{47} 5_{11}	3_{8} 3_{40}	3_{8} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}	
12	25_{37} 35_{9}	10_{11} 14_{9}	6_{14} 9_{11}	3_{8} 3_{40}	3_{11} 3_{40}	3_{11} 3_{40}	1_{0} 1_{7}	1_{0} 1_{0}
13	36_{37} 50_{9}	12_{37} 21_{9}	6_{16} 13_{39}	6_{47} 6_{11}	3_{11} 3_{40}	3_{11} 3_{40}	3_{11} 3_{40}	1_{0} 1_{7}
14	50_{37} 72_{9}	17_{37} 31_{9}	8_{18} 18_{39}	6_{12} 9_{40}	5_{47} 5_{40}	3_{11} 3_{40}	3_{11} 3_{40}	3_{11} 3_{11}
15	68_{37} 99_{9}	23_{37} 42_{9}	10_{37} 26_{11}	6_{12} 13_{39}	6_{47} 8_{40}	3_{11} 3_{40}	3_{11} 3_{40}	3_{11} 3_{11}
16	91_{37} 134_{9}	31_{37} 54_{9}	13_{37} 35_{11}	7_{18} 15_{40}	6_{12} 11_{40}	4_{11} 5_{39}	3_{11} 3_{40}	3_{11} 3_{11}
17	119_{37} 193_{9}	40_{37} 71_{9}	17_{37} 43_{39}	9_{37} 23_{39}	7_{18} 14_{39}	5_{33} 8_{11}	4_{11} 5_{40}	3_{12} 3_{11}
18	153_{37} 256_{9}	51_{37} 81_{9}	22_{37} 58_{11}	11_{37} 31_{39}	7_{18} 18_{40}	6_{33} 11_{39}	4_{12} 7_{39}	3_{12} 3_{11}
19	194_{37} 319_{9}	65_{37} 120_{9}	28_{37} 75_{39}	14_{37} 38_{39}	8_{33} 23_{39}	6_{33} 15_{12}	5_{18} 10_{40}	4_{33} 5_{39}
20	242_{37} 400_{9}	81_{37} 149_{9}	35_{37} 95_{39}	18_{37} 42_{39}	10_{37} 30_{39}	7_{33} 15_{40}	6_{33} 12_{39}	5_{33} 7_{39}

p=5, t=4

n\k	5	6	7	8	9	10	11	12
8	3_{8} 3_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}				
9	4_{11} 4_{11}	3_{8} 3_{9}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}			
10	7_{47} 7_{11}	3_{8} 3_{9}	3_{8} 3_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	7_{12} 10_{11}	5_{11} 5_{9}	3_{8} 3_{11}	3_{8} 3_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	11_{33} 18_{40}	6_{33} 6_{9}	4_{11} 4_{11}	3_{8} 3_{11}	3_{11} 3_{40}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	15_{37} 26_{39}	8_{33} 10_{9}	5_{33} 5_{11}	3_{11} 3_{11}	3_{11} 3_{40}	3_{11} 3_{40}	1_{0} 1_{7}	1_{0} 1_{7}
14	21_{37} 37_{39}	8_{18} 14_{9}	6_{33} 8_{39}	4_{11} 4_{11}	3_{12} 3_{40}	3_{11} 3_{40}	3_{11} 3_{40}	1_{0} 1_{7}
15	29_{37} 53_{39}	10_{33} 19_{9}	7_{33} 11_{39}	6_{33} 6_{11}	4_{11} 4_{40}	3_{12} 3_{40}	3_{11} 3_{40}	3_{11} 3_{40}
16	38_{37} 72_{39}	13_{37} 26_{9}	7_{33} 17_{39}	6_{33} 6_{35}	5_{33} 5_{40}	3_{11} 3_{40}	3_{12} 3_{40}	3_{11} 3_{40}
17	50_{37} 98_{39}	17_{37} 36_{9}	9_{33} 22_{39}	7_{33} 11_{11}	6_{33} 6_{40}	4_{11} 4_{40}	3_{12} 3_{40}	3_{12} 3_{40}
18	64_{37} 129_{39}	22_{37} 42_{9}	10_{33} 29_{39}	7_{33} 14_{39}	6_{33} 8_{40}	5_{33} 5_{40}	4_{11} 4_{40}	3_{12} 3_{40}
19	81_{37} 169_{39}	27_{37} 54_{9}	12_{37} 35_{11}	9_{33} 20_{39}	7_{33} 10_{39}	6_{33} 6_{40}	4_{12} 4_{40}	3_{12} 3_{40}
20	101_{37} 220_{40}	34_{37} 66_{9}	15_{37} 49_{39}	9_{33} 22_{39}	7_{33} 15_{39}	6_{33} 6_{40}	5_{33} 5_{40}	4_{33} 4_{40}

p=6, t=4

n\k	5	6	7	8	9	10	11	12
9	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}			
10	2_{8} 2_{8}	2_{8} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	5_{11} 5_{11}	2_{8} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	6_{24} 9_{40}	2_{8} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	8_{37} 14_{8}	5_{11} 5_{40}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	11_{37} 18_{8}	5_{11} 8_{40}	2_{3} 2_{3}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
15	15_{37} 29_{8}	6_{18} 10_{39}	4_{18} 5_{40}	2_{11} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
16	20_{37} 39_{8}	8_{18} 14_{8}	4_{13} 7_{40}	2_{12} 2_{8}	2_{11} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}
17	26_{37} 50_{8}	9_{37} 19_{8}	5_{18} 11_{8}	4_{18} 5_{40}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
18	34_{37} 60_{8}	12_{37} 24_{8}	6_{18} 12_{8}	4_{13} 6_{40}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	43_{37} 81_{8}	15_{37} 32_{8}	7_{37} 16_{8}	4_{13} 9_{39}	4_{18} 6_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	54_{37} 96_{8}	18_{37} 40_{8}	8_{37} 20_{8}	4_{13} 10_{40}	4_{13} 6_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=7, t=4

n\k	5	6	7	8	9	10	11	12
10	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}		
11	2_{8} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	4_{36} 4_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	6_{11} 6_{8}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	6_{12} 10_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	8_{37} 14_{8}	4_{36} 4_{8}	2_{3} 2_{3}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
16	10_{37} 18_{8}	4_{36} 7_{26}	4_{36} 4_{14}	2_{3} 2_{3}	2_{12} 2_{11}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
17	14_{37} 23_{8}	6_{18} 10_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
18	17_{37} 31_{8}	6_{18} 14_{8}	4_{36} 6_{40}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
19	22_{37} 42_{8}	8_{37} 17_{8}	4_{36} 9_{40}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	27_{37} 52_{8}	9_{37} 21_{8}	6_{18} 11_{8}	4_{36} 4_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=8, t=4

n\k	5	6	7	8	9	10	11	12
11	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}	
12	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	4_{36} 4_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	5_{11} 5_{11}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	5_{12} 8_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
16	6_{30} 10_{8}	4_{36} 4_{8}	2_{3} 2_{3}	2_{12} 2_{8}	2_{11} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
17	8_{37} 14_{8}	4_{36} 6_{8}	4_{36} 4_{14}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
18	10_{37} 18_{8}	5_{18} 6_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
19	13_{37} 23_{8}	5_{18} 10_{8}	4_{36} 5_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{12} 2_{11}
20	16_{37} 28_{8}	6_{37} 13_{8}	4_{36} 6_{8}	4_{36} 4_{8}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=9, t=4

n\k	5	6	7	8	9	10	11	12
12	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{0}
13	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
14	3_{11} 3_{11}	2_{11} 2_{8}	2_{11} 2_{8}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
15	3_{3} 3_{3}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
16	5_{24} 6_{8}	3_{3} 3_{3}	2_{12} 2_{8}	2_{11} 2_{8}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}	1_{0} 1_{7}
17	6_{38} 9_{8}	3_{12} 3_{12}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}	1_{0} 1_{7}
18	7_{37} 12_{8}	3_{12} 3_{8}	3_{18} 3_{14}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{11} 2_{11}	1_{0} 1_{7}
19	9_{37} 15_{8}	5_{24} 6_{8}	3_{18} 3_{14}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}	2_{11} 2_{11}
20	11_{37} 19_{8}	5_{13} 8_{8}	3_{18} 3_{8}	3_{30} 3_{14}	2_{12} 2_{8}	2_{12} 2_{8}	2_{12} 2_{11}	2_{12} 2_{11}

p=10, t=4

n\k	5	6	7	8	9	10	11	12
13	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
14	2_{11} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
15	3_{11} 3_8	2_{11} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
16	3_3 3_3	2_{12} 2_8	2_{11} 2_8	2_{11} 2_8	1_0 1_7	1_0 1_7	1_0 1_7	1_0 1_7
17	5_{36} 5_8	3_{18} 3_{11}	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7	1_0 1_7	1_0 1_7
18	5_{36} 7_8	3_{18} 3_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{11} 2_{11}	1_0 1_7	1_0 1_7
19	7_{37} 10_8	3_{18} 3_8	3_{30} 3_{14}	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{11} 2_{11}	1_0 1_7
20	8_{37} 14_8	5_{36} 5_8	3_{30} 3_{14}	2_{12} 2_8	2_{12} 2_8	2_{12} 2_8	2_{12} 2_{11}	2_{11} 2_{11}

p=11, t=4

References

[1] J.A. Bate. *A Generalized Covering Problem*, Ph.D. thesis, University of Manitoba, 1978.

[2] J.A. Bate, G.H.J. van Rees. *Lotto Designs*, The Journal of Combinatorial Mathematics and Combinatorial Computing **28** (1998), 15-39.

[3] Bluskov, Private Communication.

[4] E.D. Boyer, D.L. Kreher, S. P. Radziszowski, J. Zou, A. Sidorenko. *On* $(n, 5, 3)$-*Turán systems*, Ars Combinatoria **37** (1994), 13-31.

[5] A.E. Brouwer, *Some lotto numbers from an extension of Turán's thoerem*, Math. Center (Amsterdam) Report, 1978.

[6] A.E. Brouwer, M. Voorhoeve. *Turàn theory and the lotto problem*, Math. Center Tracts 106 (1979), 99-105.

[7] C. J. Colbourn, J. H. Dinitz. *The CRC Handbook of Combinatorial Designs*, CRC Press (1996).

[8] Z. Füredi, G. Székely and Z. Zubor. *On The Lottery Problem*, Journal of Combinatorial Designs, **4** (1996), 5-10.

[9] D. M. Gordon, O. Patashnik, G. Kuperberg. *New Constructions for Covering Designs*, Journal of Combinatorial Designs, **3**(4) (1995), 269-284.

[10] D. M. Gordon. Web Site of Covering Bounds at: *http://www.ccrwest.org/cover.html*.

[11] H. Hanani, D. Ornstein, V.T. Sós. *On the lottery problem*, Magyar Tud. Acad. Mat. Kutató 9 (1964), 155-158.

[12] D. L. Kreher, D. R. Stinson. **Combinatorial Algorithms** CRC Press (1999).

[13] P. C. Li. *Some Results on Lotto Designs*, Ph.D. thesis, University of Manitoba, 1999.

[14] P. C. Li. Web Site of Lotto Design Bounds at: *http://www.cs.umanitoba.ca/∼lipakc/lottotables.html*

[15] P. C. Li, G. H. J. van Rees. *Lower Bounds on Lotto Designs*,Congressus Numerantium **141** (1999), 5-30.

[16] P. C. Li, G. H. J. van Rees. *New Constructions of Lotto Designs*, Utilitas Mathematica **58** (2000) 45-64.

[17] K.J. Nurmela, P.R.J. Östergård. *Upper Bounds for Covering Designs by Simulated Annealing*, Congressus Numerantium **96** (1993), 93-111.

[18] J. Schönheim. *On Coverings*, Pacific Journal of Mathematics, **14** (1964), 1405-1411.

[19] P. Turàn.*On the theory of Graphs*, Coll. Math **3**, 1954, 19-30.