Scaling of the total number of variables and clauses in the SAT encoding of SNAP's *Marbles* Problem

Let's assume the following characteristics for the problem:

1. We have a problem with *N* tasks and *M* resources
2. We have *r* requirements per tasks in average (i.e., in average each task contribute *r* rows to the table defining the problem)
3. Each resource is a possible resource for a requirement in a task with probability *p* (i.e., there are *pM* crosses in each row of the table)

Total Number of variables:

Each type of variable will contribute the following number of variables to the total number of variables:

Variable Type	Number
Mission variables: We introduce a variable for each mission, there are N mission variables	N
Resource variables: Each row in the table contributes *pM* variables and there are *rN* rows.	$p\ r M\ N$
Additional variables: We introduce *N K* additional variables to select K missions from the original N missions. Then, this number is bounded by N^2	$K N$

Therefore, the total number of variables, *v(K,M,N)*, is given by:

$$v(K, M, N) = N + NK + NMrp \qquad (1)$$

which is polynomial in the total number of missions, N, in the number of expected missions to be scheduled, K, the average number of requirements per mission and the total number of resources and is bounded by:

$$v_{\max} = N + N^2 + NM^2$$

Total number of clauses:

Clauses Type	Number
(f_{K1}): Clauses to select at least one of the dummy variables in each of the K rows of the “at least K” trick	K
(f_{K2}): Clauses to exclude dummy variables with the same index to be turn on simultaneously (i.e., a1 -> ~ a2, etc)	$\frac{NK(K-1)}{2}$
(f_{K3}): Clauses to link the dummy variables to the mission variables (i.e., a1 -> m1, b1 -> m1, etc)	$(K+1)N$
(f_{cross}): Clauses precluding a resource to be assigned to more than a requirement	$M\frac{rpN(rpN-1)}{2}$
(f_i) with i = 1,2,….N: Clauses to select variables within each mission	$rN(1+\frac{pM(pM-1)}{2})$

Thus, the total number of clauses, *c(K,M,N),* can be written as:

$$c(K,M,N)=K+\frac{NK(K-1)}{2}+N(K+1)+M\frac{rpN(rpN-1)}{2}+rN(1+\frac{pM(pM-1)}{2}) \qquad (2)$$

which is bounded by

$$c_{\max}=2N+MN+\frac{N^2}{2}+\frac{N^3}{2}+NM^3\frac{(N+1)}{2}-NM^2$$

Comparison with the experimental results

In the following figures we compare the results coming from the analytic expressions obtained above (Eqs. (1) and (2)) against the corresponding values for the formulas we used in our experiments. In the figures we compare the total number of variables and clauses for problems with N resources and N tasks for N=30, 50, 60 and 100. As we can see in all the figures the experimental results follow the same polynomial behavior given by the formulas derived above.

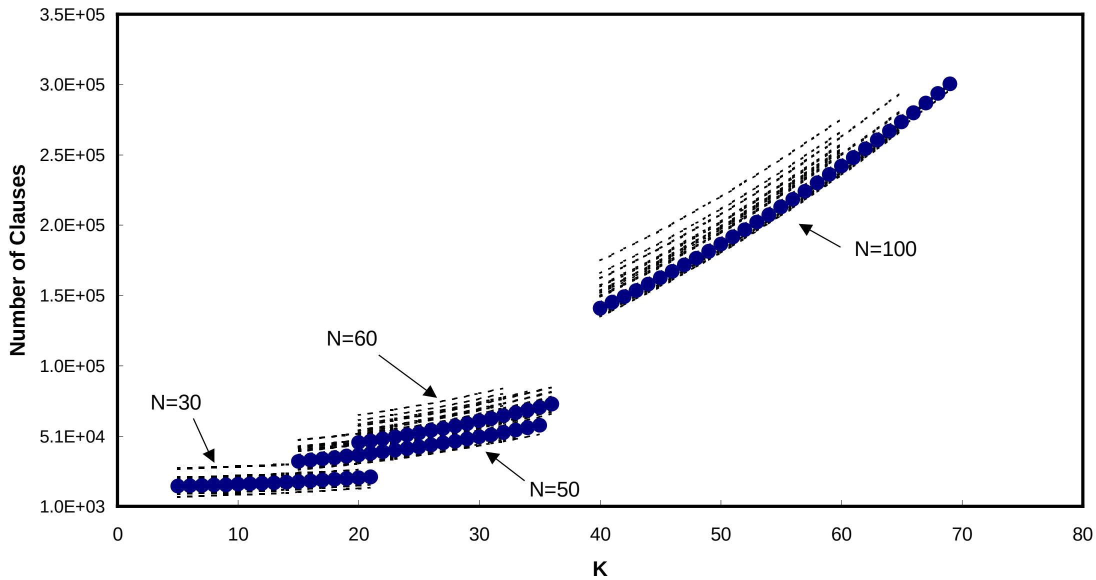


Figure 1: Scaling of the total number of clauses as a function of K (the least number of filled tasks) for problems with N resources and N tasks with N=30, 50, 60 and 100. The dashed lines correspond to the 20 experimental instances of each problem size and the big blue dots are the analytical values coming from Eq. (2).

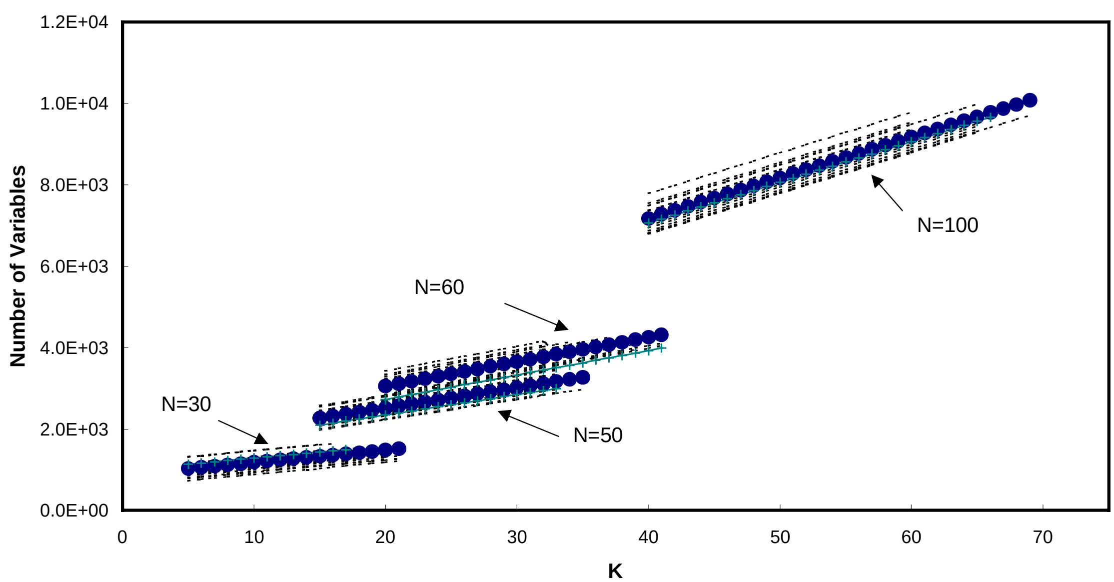


Figure 2: Scaling of the total number of variables as a function of K (the least number of filled tasks) for problems with N resources and N tasks with N=30, 50, 60 and 100. The dashed lines correspond to the 20 experimental instances for each problem size and the big blue dots are the analytical values coming from Eq. (1).

In Figures 1 and 2 we see the scaling of the total number of clauses and variables, respectively, with the least number of filled tasks, K, for different sets of problems with N=30, 50, 60 and 100. Since the problems were randomly generated, the resulting number of variables and clauses are also subjected to a distribution, this effect is shown by the dashed black lines which run parallel to each other due to a distribution in the number of requirements per tasks and possible resources per requirement. The line with big blue dots corresponds to the analytical values given by Eqs. (1) and (2).

In Figure 3 we see the scaling of the total number of clauses with the size of the problems, N ($M=N$). For each value of N, the black crosses indicate the total number of clauses of the experimental formula with the maximum value of K, i.e., the largest solvable formula for each problem instance. The big blue dots correspond to the value obtained using Eq. (2) for the total number of clauses and evaluated at $K = \langle K_{max} \rangle$, where $\langle K_{max} \rangle$ is the average value of K_{max} for the corresponding set of problems of size N. As we can see the, the experimental points follow the same polynomial behavior ($O(N^3)$) given by the analytical formula.

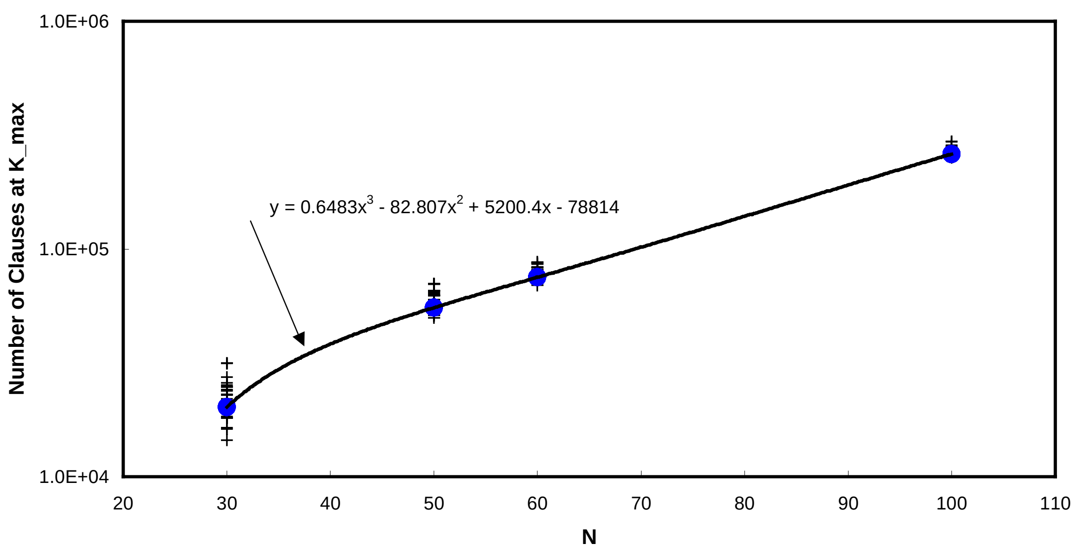
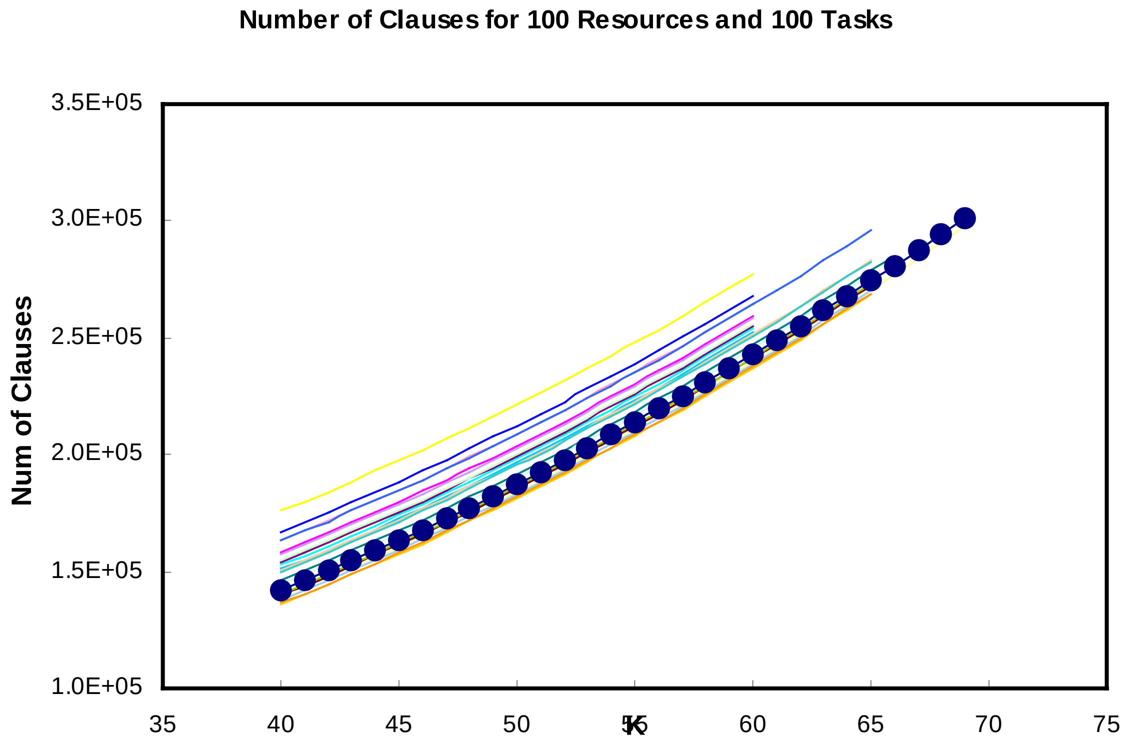
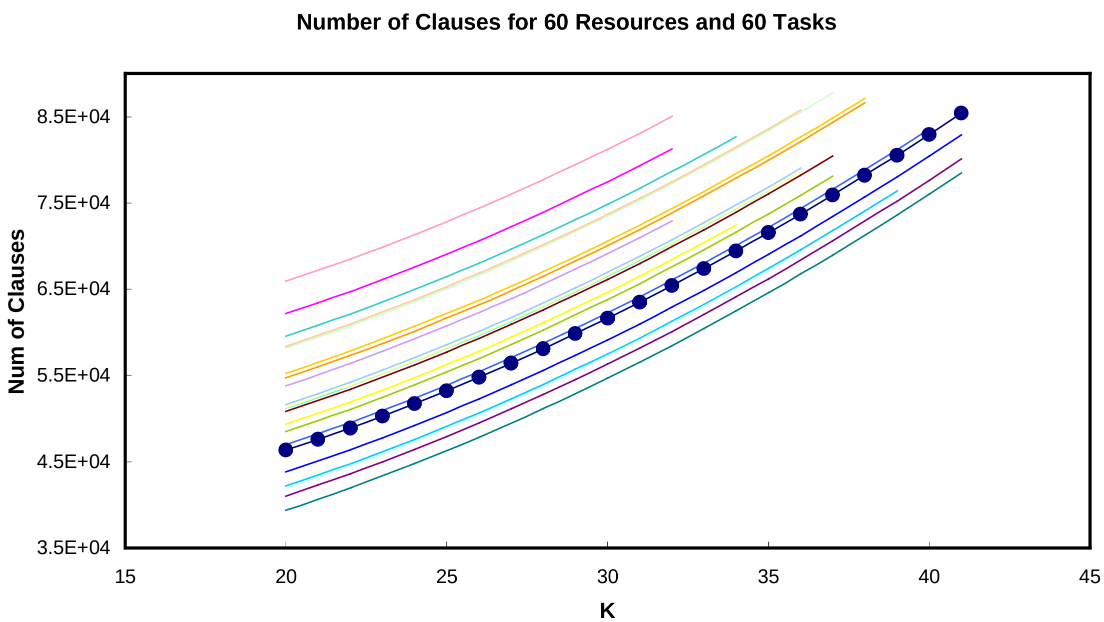
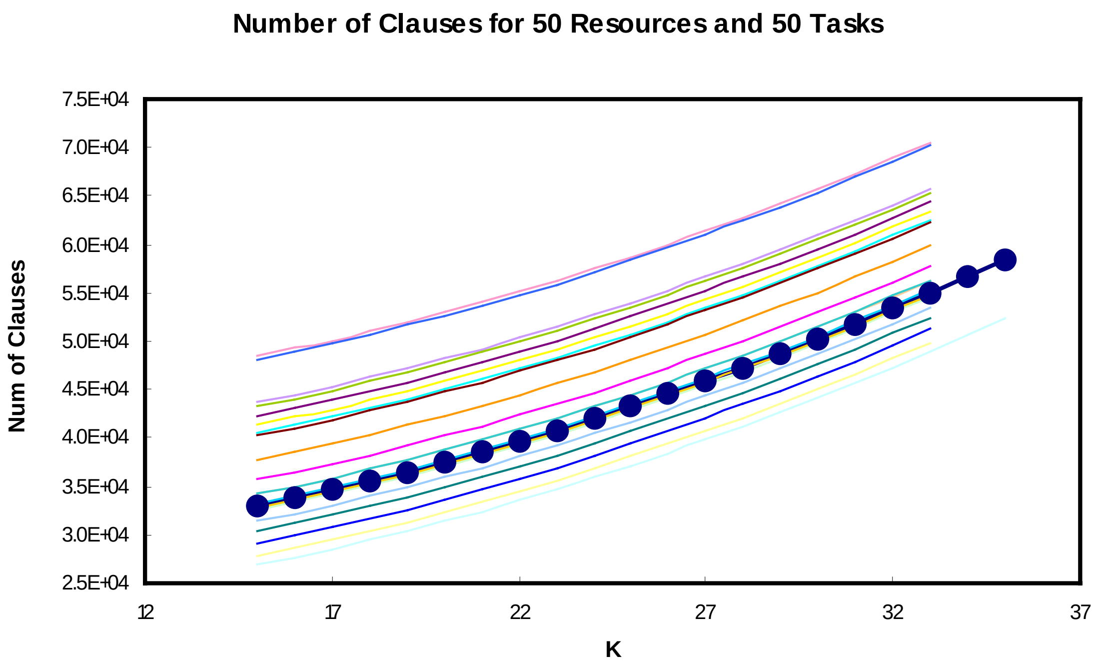


Figure 3: Scaling of the total number of clauses with the size of the problem for $K = K_{max}$. The black crosses correspond to the 20 experimental values for each problem of size N and the big blue dots are the analytical values coming from Eq. (2) with $K = \langle K_{max} \rangle$.

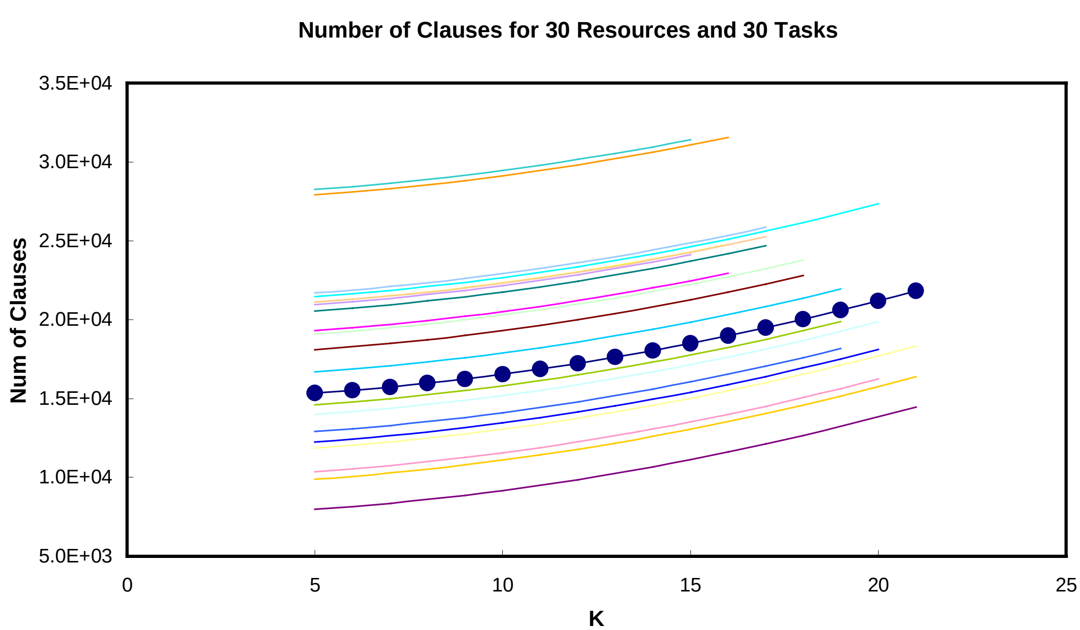
The following 8 figures in the document show in more detail the scaling of the total number of clauses and variables with the parameter K for each value of N.





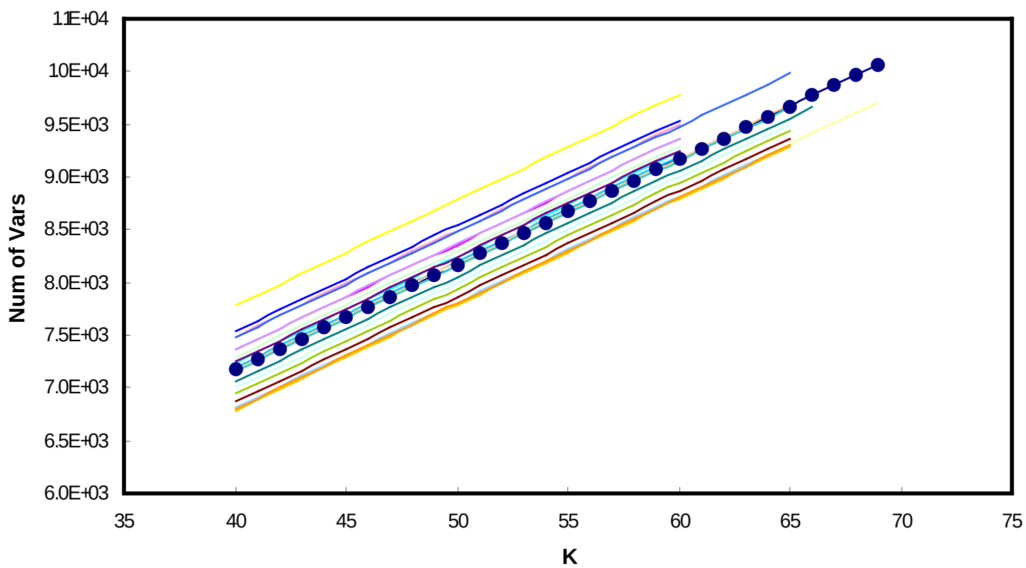


Number of Clauses for 50 Resources and 50 Tasks
Num of Clauses
K
7.5E+04
7.0E+04
6.5E+04
6.0E+04
5.5E+04
5.0E+04
4.5E+04
4.0E+04
3.5E+04
3.0E+04
2.5E+04
12
17
22
27
32
37

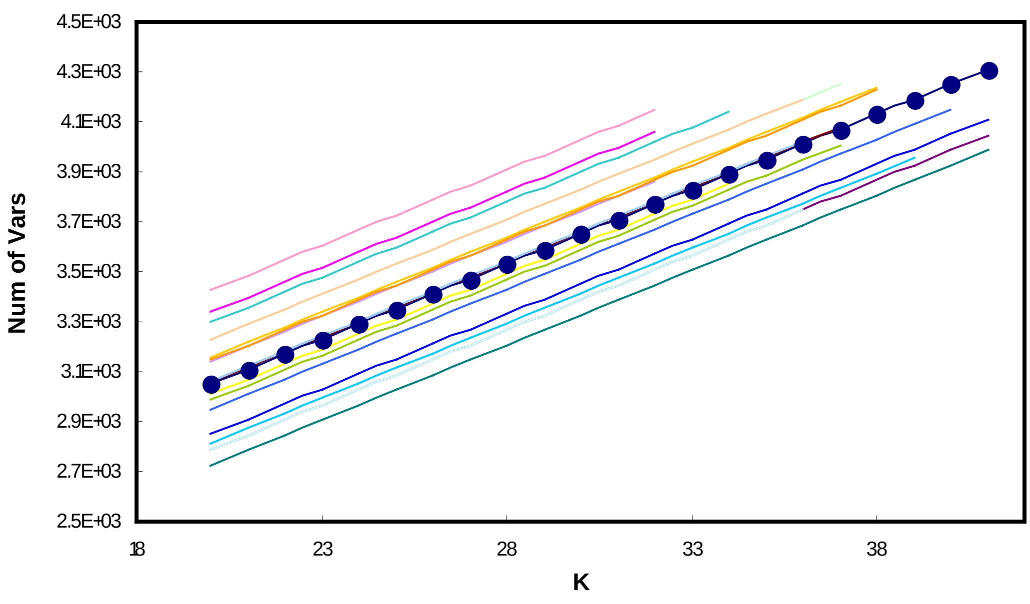


Number of Clauses for 30 Resources and 30 Tasks
Num of Clauses
K
3.5E+04
3.0E+04
2.5E+04
2.0E+04
1.5E+04
1.0E+04
5.0E+03
0
5
10
15
20
25

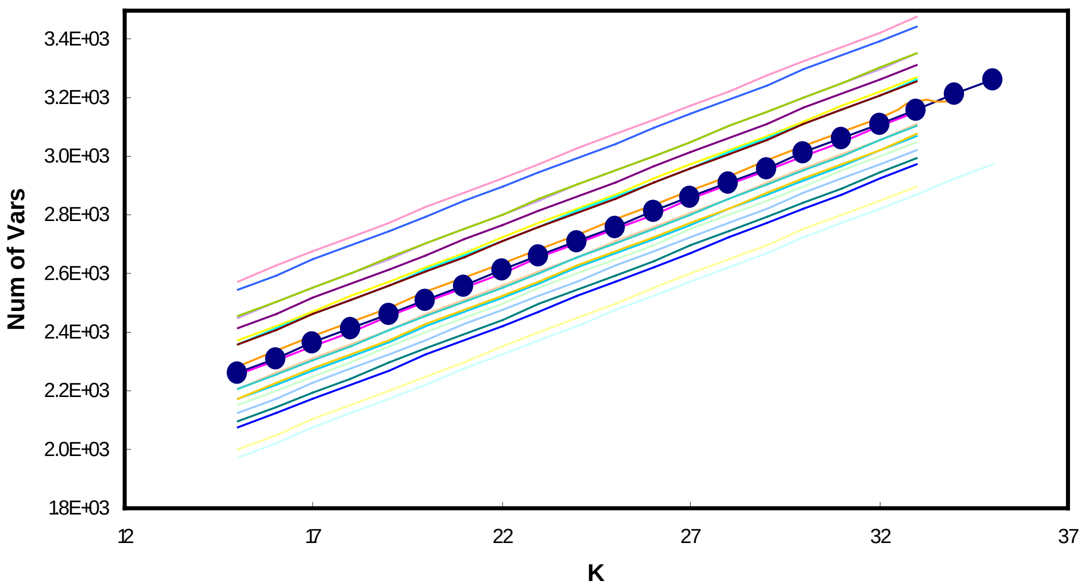
Number of Variables for 100 Resources and 100 Tasks



Number of Variables for 60 Resources and 60 Tasks



Number of Variables for 50 Resources and 50 Tasks



Number of Variables in 30x30 problems