Estimating the Number of Solutions of Cardinality Constraints Through `range` and `roots` Decompositions

Giovanni Lo Bianco[1(✉)], Xavier Lorca[2(✉)], and Charlotte Truchet[3(✉)]

[1] IMT Atlantique, Nantes, France
giovanni.lo-bianco@imt-atlantique.fr
[2] ORKID, Centre de Génie Industriel, IMT Mines Albi, Albi, France
xavier.lorca@mines-albi.fr
[3] Université de Nantes, Nantes, France
truchet.charlotte@univ-nantes.fr

Abstract. This paper introduces a systematic approach for estimating the number of solutions of cardinality constraints. A main difficulty of solutions counting on a specific constraint lies in the fact that it is, in general, at least as hard as developing the constraint and its propagators, as it has been shown on `alldifferent` and `gcc` constraints. This paper introduces a probabilistic model to systematically estimate the number of solutions on a large family of cardinality constraints including `alldifferent`, `nvalue`, `atmost`, etc. Our approach is based on their decomposition into `range` and `roots`, and exhibits a general pattern to derive such estimates based on the edge density of the associated variable-value graph. Our theoretical result is finally implemented within the *maxSD* search heuristic, that aims at exploring first the area where there are likely more solutions.

Keywords: Cardinality constraints · Counting · Random graphs

1 Introduction

Dealing with a combinatorial problem often leads to the natural question of computing or estimating its number of solutions. Such a question arises, for instance, in several works on probabilistic reasoning and machine learning [8,9], or when exploring the structure of the solution space [17]. Counting solutions has indeed been an active research topic in Constraint Programming, in particular on global constraints [13]. Unfortunately, designing an efficient counting algorithm for a specific constraint is as hard as the constraint development itself. Hence, solution counting methods require customized counting algorithms for bounding, or estimating, the number of solutions for each global constraint. We propose here a systematic method to estimate the number of solutions of most of the cardinality constraints.

This article focuses on ten of them: `alldifferent`, `nvalue`, `atmostNValues`, `atleastNValues`, `occurrence`, `atmost`, `atleast`, `among`, `uses`, `disjoint`. They all constrain the number of occurrences of certain values or the number of different values in a solution. They can be mathematically modelled with bipartite graphs. In [13], the problem of counting solutions for `alldifferent` and `gcc` is transformed into counting matchings in these graphs. Solving such problems is very hard: they often belong to the #P-complete complexity class. This is why counting-based search, as presented in [13], are not based on exact counting but on estimations or upper bounds. In this article, we introduce a probabilistic approach to compute such an estimation.

In [2], the authors introduce two new global constraints `range` and `roots`, that can be used to specify many cardinality constraints. In other words, for almost every cardinality constraint, there is an equivalent model using only the more primitive `range` and `roots` constraints (and some arithmetic constraints). This equivalent model is called the decomposition of the initial cardinality constraint. We show how to use the `range` and `roots` decomposition for counting solutions. More precisely, we develop a probabilistic approach to estimate the number of solutions on a `range` and on a `roots` constraint and we derive from it a systematic method to estimate the number of solutions on many cardinality constraints. Compared to [13], we obtain an estimation instead of an upper bound, and we propose a method that can be generalized to a large set of cardinality constraints without redesigning a dedicated model.

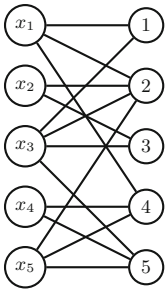
Outline: The paper is organized as follows. Section 2 gives an introduction to the `range` and `roots` constraints and some materials to understand the associated bipartite graph model. In Sect. 3, we detail how to count exactly the number of solutions on `range` and `roots` and then we apply a probabilistic model to develop an estimation of the true number of solutions. In Sect. 4, we give the `range` and `roots` decomposition and an estimation of the number of solutions for several cardinality constraints, and we synthesize our estimators under a general formula. In Sect. 5, we experiment our probabilistic estimators within the counting-based strategy `maxSD`.

2 Preliminaries : Introduction to `range` and `roots`

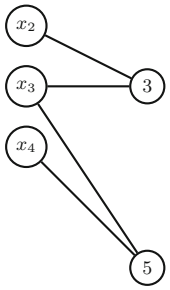
In all the article, we will use the following notations. Let $X = \{x_1, \ldots, x_n\}$, the set of variables. For each variable $x_i \in X$, we note D_i its domain, $Y = \bigcup_{i=1}^{n} D_i = \{y_1, \ldots, y_m\}$ the union of the domains and $\mathcal{D} = D_1 \times \ldots \times D_n$, the Cartesian product of the domains. We note $d_i = |D_i|$, the size of the domain of x_i. Given a constraint C on variables X, we write $\mathcal{S}_{C(X)}$ the set of solutions of C for X and we write $\#C(X)$ the number of tuples allowed by C for X.

Cardinality constraints restrict the number of occurrences of particular values taken by set of variables, or the number of values or variables meeting some conditions. Among them, we can list `alldifferent`, `gcc`, `nvalue`, `atleast`, `atmost`. We will come back and define properly these constraints one by one in Sect. 4.

Most of the time, these constraints can be modelled with a bipartite graph, in which we are looking for some mathematical structures, such as matchings for example.



(a) Value Graph $G_{X,Y}$



(b) Sub-Value Graph $G_{X',Y'}$

Fig. 1. Value graph and sub-value graph of Examples 1 and 2.

Definition 1 (Value Graph). *Let $G_{X,Y} = G(X \cup Y, E)$, the graph on nodes $X \cup Y$, with edges $E = \{(x_i, y_j) \mid y_j \in D_i\}$. $G_{X,Y}$ is a bipartite graph representing the domain of each variable. There is an edge between x_i and y_j iff $y_j \in D_i$.*

Example 1. Let $X = \{x_1, x_2, x_3, x_4, x_5\}$ with $D_1 = \{1, 2, 4\}$, $D_2 = \{2, 3\}$, $D_3 = \{1, 2, 3, 5\}$, $D_4 = \{4, 5\}$ and $D_5 = \{2, 4, 5\}$. We obtain the value graph $G_{X,Y}$ depicted on Fig. 1a.

We also define the sub-value graph induced by two subsets $X' \subseteq X$ and $Y' \subseteq Y$, as the value graph restricted to the considered subset of nodes.

Definition 2 (Sub-Value Graph induced by subsets of X and Y). *Let $G_{X',Y'} = G(X' \cup Y', E)$, the value graph of X' with $E = \{(x_i, y_j) \mid y_j \in D'_i = D_i \cap Y'\}$. $G_{X',Y'}$ is a bipartite graph representing the sub-domain induced by Y' of each variable. There is an edge between x_i and y_j iff $y_j \in D'_i$.*

We will also note $d_i(Y') = |D'_i|$ the size of the domain of x_i restricted to the values of Y'. Example 2 illustrates a sub-value graph of the value graph presented in Example 1.

Example 2. Let $X' = \{x_2, x_3, x_4\} \subseteq X$ and $Y' = \{3, 5\} \subseteq Y$. the sub-value graph induced by X' and Y' is represented in Fig. 1b.

The `range` and `roots` constraints [2] are two auxiliary constraints that can help decomposing a lot of cardinality constraints. In this study, we will use these decomposition to count solutions on cardinality constraints. As the authors wrote in [2], "`range` captures the notion of image of a function and `roots` captures the notion of domain". In this paper, we use alternative definitions for these constraints, equivalent to those of [2] and better suited to our needs.

Definition 3 (range). *Let $X' \subseteq X$ and $Y' \subseteq Y$. The constraint* ***range*** *(X, X', Y') holds if the values assigned to variables of X' covers* ***exactly*** *Y' and not more. Formally:*

$$\mathcal{S}_{range(X,X',Y')} = \{(v_1, \ldots, v_n) \in \mathcal{D} \mid \{v_i | x_i \in X'\} = Y'\} \quad (1)$$

Definition 4 (roots). *Let $X' \subseteq X$ and $Y' \subseteq Y$. The constraint* ***roots*** *(X, X', Y') holds if the variables that are assigned to values of Y' covers* ***exactly*** *X' and not more. Formally:*

$$\mathcal{S}_{roots(X,X',Y')} = \{(v_1, \ldots, v_n) \in \mathcal{D} \mid \{x_i | v_i \in Y'\} = X'\} \quad (2)$$

Example 3. Let's take the value graph given in Example 1a.

- The tuple $(2, 2, 3, 4, 5)$ is allowed by the constraint $\texttt{range}(X, \{x_1, x_2, x_3\}, \{2, 3\})$.
- The tuple $(2, 2, 3, 4, 5)$ is allowed by the constraint $\texttt{roots}(X, \{x_1, x_2, x_3\}, \{2, 3\})$.

Note that `range` and `roots` are not exactly reciprocal because every variable must be assigned to a value, but a value is not necessarily assigned to a variable.

3 Counting Solutions on the `range` and `roots` Constraints

As developed in [13], counting solutions on cardinality constraints requires dedicated counting algorithm for each constraint. In this section we are interested by computing the number of solutions on the `range` and the `roots` constraints. The idea is then to only use the decomposition of cardinality constraints into these more primitive constraints and to reuse the counting method on `range` and `roots` to count solutions on cardinality constraints.

3.1 Exact Solutions Counting on `range` and `roots`

In this subsection, we are interested by exactly computing the number of allowed tuples for a `range` constraint and a `roots` constraint.

Proposition 1. *Let $X' \subseteq X$ and $Y' \subseteq Y$. We note $\overline{X'}$, the complement of X' in X, such that $\overline{X'} \cup X' = X$ and $\overline{X'} \cap X' = \emptyset$. Then, the number of tuples allowed by* ***range*** *(X, X', Y') is*

$$\#\textbf{\textit{range}}(X, X', Y') = \#\textbf{\textit{range}}(X', X', Y') \cdot \prod_{x_i \in \overline{X'}} d_i \quad (3)$$

Proof. On one side, we must consider every possible assignment for the variables of $\overline{X'}$ that are not constrained: $\prod_{x_i \in \overline{X'}} d_i$. And on the other side, we must count every tuples allowed for variables of X', that are constrained, that is simply $\#\texttt{range}(X', X', Y')$. The number of tuples is thus the product of these quantities. □

Proposition 1 reduces the problem of counting allowed tuples for every variable in X to only counting tuples for the constrained variables X'. We thus have reduced the problem to counting the number of allowed tuples in the case where every variable and value is constrained.

Proposition 2.

$$\#\mathit{range}(X, X, Y) = \prod_{x_i \in X} d_i - \sum_{Y' \subsetneq Y} \#\mathit{range}(X, X, Y') \tag{4}$$

Proof. Inside $G_{X,Y}$, we must count every possible assignment of variables of X such that every value of Y is covered. To do that, we first count the number of every possible assignment of variables of X in $G_{X,Y}$ (without considering the `range` constraint):

$$\prod_{x_i \in X} d_i$$

And then, we withdraw, one by one, the assignment of X such that Y is not fully covered, that is, for every subset $Y' \subsetneq Y$, the solutions of $\texttt{range}(X, X, Y')$:

$$\sum_{Y' \subsetneq Y'} \#\texttt{range}(X, X, Y')$$

Indeed, for two different subsets $Y'_1 \neq Y'_2 \subsetneq Y$, the sets of allowed tuples $\mathcal{S}_{\texttt{range}(X,X,Y'_1)}$ and $\mathcal{S}_{\texttt{range}(X,X,Y'_2)}$ are necessarily disjoint: there is a value $y_j \in Y$ such that $y_j \in Y'_1$ and $y_j \notin Y'_2$ (or $y_j \in Y'_2$ and $y_j \notin Y'_1$), so the value y_j must be assigned to one of the variable of X to satisfy $\texttt{range}(X, X, Y'_1)$ but none of the variable of X must be assigned to y_j to satisfy $\texttt{range}(X, X, Y'_2)$ (or vice-versa). A solution of $\texttt{range}(X, X, Y'_1)$ cannot be a solution of $\texttt{range}(X, X, Y'_2)$ and vice-versa. No solution are counted twice in $\sum_{Y' \subsetneq Y} \#\texttt{range}(X, X, Y')$. We have:

$$\#\texttt{range}(X, X, Y) = \prod_{x_i \in X} d_i - \sum_{Y' \subsetneq Y} \#\texttt{range}(X, X, Y')$$

□

Remark 1. Proposition 2 can be used in Proposition 1 and we obtain:

$$\#\texttt{range}(X, X', Y') = \prod_{x_i \in \overline{X'}} d_i \cdot \left(\prod_{x_i \in X'} d_i(Y') - \sum_{Y'' \subsetneq Y'} \#\texttt{range}(X', X', Y'') \right)$$

This formulae requires to recursively sum and evaluate terms over a exponential-size set and is not tractable in practice (we believe that it is a $\#P-$complete problem). In next subsection, we will give an approximation which is much faster to compute. We now deal with the `roots` constraint.

Proposition 3. *Let $X' \subseteq X$ and $Y' \subseteq Y$. We note $\overline{X'}$, the complement of X' in X and $\overline{Y'}$ the complement of Y' in Y. Then, the number of tuples allowed by* `roots`(X, X', Y') *is*

$$\#\texttt{roots}(X, X', Y') = \prod_{x_i \in X'} d_i(Y') \cdot \prod_{x_i \in \overline{X'}} d_i(\overline{Y'}) \tag{5}$$

Proof. In order to satisfy `roots`(X, X', Y'), every variable from X' must take a value in Y' and no value from Y' must be assigned to a variable from $\overline{X'}$, that is every variable from $\overline{X'}$ must be assigned to values from $\overline{Y'}$:

- $\prod_{x_i \in X'} d_i(Y')$ represents the number of ways of assigning every variable of X'
- $\prod_{x_i \in \overline{X'}} d_i(\overline{Y'})$ represents the number of ways of assigning every variable of $\overline{X'}$

□

The formula given by Proposition 3 is polynomial to compute. In practice, the formula depends on the subsets X' and Y'. Applying the Erdos-Renyi model on `roots` allows the estimation of $\#roots(X, X', Y')$ using only the sizes of X' and Y', with a linear complexity.

In Sect. 4, we compose these constraints to count solutions on other cardinality constraints.

3.2 Probabilistic Model Applied to `range` and `roots`

This subsection presents a probabilistic model for cardinality constraints based on the work of Erdős and Renyi In [5]. The idea is to randomize the domain of the variables. Then, we use this model to get a computable estimation of the number of solutions on `range` and `roots`.

Erdős-Renyi Model Applied to CSP. In [5], Erdős and Renyi studied the existence and the number of perfect matchings on random graphs. Expressed in the vocabulary we introduced above, the idea is to randomize the domain of each variable such that: for all $x_i \in X$ and for all $y_j \in Y$, the event $\{y_j \in D_i\}$ happens with a predefined probability $p \in [0, 1]$ and all such events are **independent:**

$$\mathbb{P}(\{y_j \in D_i\}) = p \in [0, 1] \tag{6}$$

Erdős-Renyi Model Applied to `range` Constraint. We will study the expectancy of the number of solutions of a `range` constraint within these random graphs. In the case where every variable of X and every value of Y are constrained, the expectancy of $\#$`range`(X, X, Y) is a function of n, m and p (as a reminder, $|X| = n$ and $|Y| = m$). More precisely:

Proposition 4. *In the case where every variable of X and every value of Y are constrained, there exists a coefficient $a_{n,m}$ such that:*

$$\mathbb{E}(\#\texttt{range}(X, X, Y)) = a_{n,m} \cdot p^n \tag{7}$$

where $\mathbb{E}(\#\texttt{range}(X,X,Y))$ *is the expectancy of* $\#\texttt{range}(X,X,Y)$ *under the hypothesis of the Erdős-Renyi Model.*

Proof. To prove this result, we simply reason with a mathematical induction on $|Y| = m$. Let $|X| = n \in \mathbb{N}$.

Base Case: Let $Y = \{y\}$ be a singleton. In this particular case, an instance $\texttt{range}(X,X,Y)$ have one allowed tuple, if y is inside every domain D_i, and have zero allowed tuple otherwise. Then,

$$\begin{aligned}
\mathbb{E}(\#\texttt{range}(X,X,\{y\})) &= 0 * \mathbb{P}(\{\texttt{range}(X,X,\{y\}) \text{ have no solution}\}) \\
&\quad + 1 * \mathbb{P}(\{\texttt{range}(X,X,\{y\}) \text{ have one solution}\}) \\
&= \mathbb{P}(\{\texttt{range}(X,X,\{y\}) \text{ have one solution}\}) \\
&= \mathbb{P}(\{\forall x_i \in X, y \in D_i\}) \\
&= \prod_{i=1}^{n} \mathbb{P}(\{y \in D_i\}), \text{by hypothesis of independence} \\
&= p^n
\end{aligned}$$

We thus set $a_{n,1} = 1$, which proves the result.

Inductive Step. We assume that the property is true for all $|Y| = k \in \{1,\dots,m-1\}$: $\forall Y,$ such that $1 \leq |Y| = k \leq m-1, \exists a_{n,k} \in \mathbb{N}$,

$$\mathbb{E}(\#\texttt{range}(X,X,Y)) = a_{n,k} \cdot p^n.$$

We want to prove that, under this assumption, for a set Y with $|Y| = m$, there exists $a_{n,m}$ such that $\mathbb{E}(\#\texttt{range}(X,X,Y)) = a_{n,m} \cdot p^n$

According to Proposition 2, we have:

$$\begin{aligned}
&\mathbb{E}(\#\texttt{range}(X,X,Y)) \\
&= \mathbb{E}\left(\prod_{x_i \in X} d_i\right) - \sum_{Y' \subset Y} \mathbb{E}(\#\texttt{range}(X,X,Y')), \text{ by linearity of the operator } \mathbb{E}(.) \\
&= \mathbb{E}\left(\prod_{x_i \in X} d_i\right) - \sum_{k=1}^{m-1} \binom{m}{k} a_{n,k} \cdot p^n, \text{ by hypothesis of induction.} \\
&= \prod_{x_i \in X} \mathbb{E}(d_i) - \sum_{k=1}^{m-1} \binom{m}{k} a_{n,k} \cdot p^n, \text{ by hypothesis of independence} \\
&= (mp)^n - \sum_{k=1}^{m-1} \binom{m}{k} a_{n,k} \cdot p^n, \text{ because } \forall x_i \in X, \mathbb{E}(d_i) = mp \\
&= \left(m^n - \sum_{k=1}^{m-1} \binom{m}{k} a_{n,k}\right) \cdot p^n
\end{aligned}$$

We have identified the coefficient $a_{n,m}$:

$$a_{n,m} = m^n - \sum_{k=1}^{m-1} \binom{m}{k} a_{n,k} \tag{8}$$

□

Remarking that $\binom{m}{m} = 1$, we can rewrite 8 as follows:

$$m^n = \sum_{k=1}^{m} \binom{m}{k} a_{n,k} \tag{9}$$

Also, $\forall n \in \mathbb{N}^+, a_{n,1} = 1$. These coefficients are referenced as the "triangles of numbers" in OEIS.[1] The coefficients $a_{n,m}$ corresponds to the number of possible surjections from a set of cardinal n into a set of cardinal m.[2] There is a non-recursive formula to compute these coefficients. The following results is admitted here. An intuition of the proof is that this results is an application of the inclusion-exclusion principle, see Sect. 1.9. The Twelvefold Way of [18].

Proposition 5. *For* $0 < m \leq n$,

$$a_{n,m} = \sum_{k=0}^{m} (-1)^k \binom{m}{k} (m-k)^n \tag{10}$$

Proposition 6 is a property of triangle of numbers and will be used to make some simplifications for future mathematical developments.

Proposition 6.

$$a_{n,n} = n! \tag{11}$$

Proof. $a_{n,n}$ is the number possible surjections from a set of cardinality n into a set of cardinality n, which is actually the number of bijections in that specific case. □

We can now extend Proposition 4 to the case where the `range` constraint only concerns subsets $X' \subseteq X$ and $Y' \subseteq Y$:

Proposition 7. *Let* $X' \subseteq X$ *and* $Y' \subseteq Y$. *We note* $|X'| = n'$ *and* $|Y'| = m'$.

$$\mathbb{E}\left(\#\texttt{range}(X, X', Y')\right) = a_{n',m'} \cdot m^{n-n'} \cdot p^n \tag{12}$$

[1] https://oeis.org/A019538.

[2] $a_{n,m}$ is actually equal to $m! \cdot S_2(n,m)$, where $S_2(n,m)$ is the stirling number of second kind. More information about it can be found in Sect. 1.9 of [18].

Proof. According to Propositions 1 and 4 and by hypothesis of independence:

$$\begin{aligned}\mathbb{E}\left(\#\texttt{range}(X, X', Y')\right) &= \mathbb{E}\left(\#\texttt{range}(X', X', Y')\right) \cdot \mathbb{E}\left(\prod_{x_i \in \overline{X'}} d_i\right) \\ &= a_{n',m'} \cdot p^{n'} \cdot \prod_{x_i \in \overline{X'}} \mathbb{E}\left(d_i\right) \\ &= a_{n',m'} \cdot p^{n'} \cdot (mp)^{n-n'} \\ &= a_{n',m'} \cdot m^{n-n'} \cdot p^{n}\end{aligned}$$

□

Erdős-Renyi Model Applied to `roots` Constraint. We study now the expectancy of the number of solutions of a `roots` constraint.

Proposition 8. *Let $X' \subseteq X$ and $Y' \subseteq Y$. We note $|X'| = n'$ and $|Y'| = m'$.*

$$\mathbb{E}\left(\#\texttt{roots}(X, X', Y')\right) = m'^{n'} \cdot (m - m')^{n-n'} \cdot p^{n} \tag{13}$$

Proof. According to Proposition 3 and by hypothesis of independence:

$$\begin{aligned}\mathbb{E}\left(\#\texttt{roots}(X, X', Y')\right) &= \mathbb{E}\left(\prod_{x_i \in X'} d_i(Y')\right) \cdot \mathbb{E}\left(\prod_{x_i \in \overline{X'}} d_i(\overline{Y'})\right) \\ &= \prod_{x_i \in X'} \mathbb{E}\left(d_i(Y')\right) \cdot \prod_{x_i \in \overline{X'}} \mathbb{E}\left(d_i(\overline{Y'})\right) \\ &= (m'p)^{n'} \cdot ((m - m')p)^{n-n'} \\ &= m'^{n'} \cdot (m - m')^{n-n'} \cdot p^{n}\end{aligned}$$

□

The parameter p corresponds to the density of edges in the value graph. To use the estimators in practice, we need to estimate p: we will later set p to the division of the sum of domains size by the total number of possible edges: $n \cdot m$.

4 Generalization to Cardinality Constraints

This section details, in a systematic way, how to count solutions for many cardinality constraints thanks to their `range` and `roots` decompositions. Due to space limitations, only four constraints are given in detail. For the other six constraints to which our method applies, a synthesis then summarises all the formulae as well as the general computation pattern. Each subsection first recalls the definitions of the considered constraint, then details its decomposition as extracted from [2] and finally provides the formula for the expectancy of its number of solution in our model.

4.1 alldifferent [16]

Definition 5. *A constraint* $\texttt{alldifferent}(X)$ *is satisfied iff each variable* $x_i \in X$ *is instantiated to a value of its domain* D_i *and each value* $y_j \in Y$ *is chosen at most once. We define formally the set of allowed tuples:*

$$\mathcal{S}_{alldifferent(X)} = \{(v_1, \ldots, v_n) \in \mathcal{D} \mid \forall i, j \in \{1, ..., n\}, i \neq j \Leftrightarrow v_i \neq v_j\} \quad (14)$$

A decomposition of `alldifferent` with a `range` constraint is given by the following:

$$\texttt{alldifferent}(X) \Leftrightarrow \texttt{range}(X, X, Y') \ \wedge \ |Y'| = n$$

From this decomposition, we can deduce a formula for the expectancy of the number solutions on an `alldifferent` constraint, within the Erdős-Renyi Model.

Proposition 9.

$$\mathbb{E}\left(\#\texttt{alldifferent}(X)\right) = \frac{m!}{(m-n)!} \cdot p^n \quad (15)$$

Proof. According to the decomposition of `alldifferent`.

$$\#\texttt{alldifferent}(X) = \sum_{Y' \subseteq Y, \, |Y'|=n|} \#\texttt{range}(X, X, Y')$$

Then,

$$\begin{aligned}\mathbb{E}\left(\#\texttt{alldifferent}(X)\right) &= \sum_{Y' \subseteq Y, \, |Y'|=n|} \mathbb{E}\left(\#\texttt{range}(X, X, Y')\right) \\ &= \binom{m}{n} \cdot a_{n,n} \cdot p^n = \frac{m!}{(m-n)!} \cdot p^n\end{aligned}$$

□

4.2 nvalue [11]

Definition 6. *The constraint* $\texttt{nvalue}(X, N)$ *holds if exactly* N *values from* Y *are assigned to the variables. Formally:*

$$\mathcal{S}_{nvalue(X,N)} = \{(v_1, \ldots, v_n) \in \mathcal{D} | \ N = |\{y_j \in Y | \exists i \in \{1, \ldots, n\}, v_i = y_j\}|\} \quad (16)$$

A decomposition of `nvalue` with a `range` constraint is given by the following:

$$\texttt{nvalue}(X, N) \Leftrightarrow \texttt{range}(X, X, Y') \ \& \ |Y'| = N$$

From this decomposition, we can deduce a formula to estimate solutions on a `nvalue` constraint, within the Erdős-Renyi Model.

Proposition 10. *Let* $N \in \mathbb{N}$,

$$\mathbb{E}\left(\#\texttt{nvalue}(X, N)\right) = \binom{m}{N} \cdot a_{n,N} \cdot p^n \quad (17)$$

Proof. The proof is the same as Proposition 9. □

We can generalize Proposition 9 to the case where N is a variable. The set of solutions for two different values of N are disjoints, then we can simply sum this estimates on the domain of N to compute an estimate in the general case.

4.3 among [1]

Definition 7 (among). *Let $Y' \subseteq Y$. The constraint $\mathit{among}(X, Y', N)$ holds iff exactly N variables are assigned to value from Y'.*

$$\mathcal{S}_{\mathit{among}(X,Y',N)} = \{(v_1, \ldots, v_n) | N = |\{x_i | v_i \in Y'\}|\}$$

The decomposition of among is given by the following equivalence:

$$\texttt{among}(X, Y', N) \Leftrightarrow \texttt{roots}(X, X', Y') \ \wedge \ |X'| = N$$

Proposition 11. *Let $m' = |Y'|$ and $N \in \mathbb{N}$,*

$$\mathbb{E}\left(\#\mathit{among}(X, Y', N)\right) = \binom{n}{N} m'^N (m - m')^{n-N} \cdot p^n \tag{18}$$

Proof. According to the decomposition of among, we can write:

$$\#\texttt{among}(X, Y', N) = \sum_{X' \subseteq X, |X'|=N} \#\texttt{roots}(X, X', Y')$$

Indeed, for two different subsets $X'_1, X'_2 \subseteq X$, the sets of solutions of $\texttt{roots}(X, X'_1, Y')$ and $\texttt{roots}(X, X'_2, Y')$ have an empty intersection, then no solution is counted twice. And:

$$\begin{aligned}
\mathbb{E}\left(\#\texttt{among}(X, Y', N)\right) &= \sum_{X' \subseteq X, |X'|=N} \mathbb{E}\left(\#\texttt{roots}(X, X', Y')\right) \\
&= \sum_{X' \subseteq X, |X'|=N} m'^{m'} (m - m')^{n-|X'|} \cdot p^n, \text{by Proposition 8} \\
&= \binom{n}{N} m'^{m'} (m - m')^{n-N} \cdot p^n
\end{aligned}$$

□

In the same way as for nvalue, we can generalize Proposition 11 to the case where N is a variable.

4.4 occurrence [4]

Definition 8 (occurrence). *Let $y \in Y$, the constraint $\mathit{occurrence}(X, y, N)$ holds iff exactly N variables are assigned to value y.*

$$\mathcal{S}_{\mathit{occurrence}(X,y,N)} = \{(v_1, \ldots, v_n) | N = |\{x_i | v_i = y\}|\}$$

The decomposition of occurrence is given by the following equivalence:

$$\texttt{occurrence}(X, y, N) \Leftrightarrow \texttt{roots}(X, X', \{y\}) \ \wedge \ |X'| = N$$

Table 1. Counting formulae extracted from `range` and `roots` reformulation

Constraint	Formula with $\lvert X\rvert = n$, $\lvert X_1\rvert = n_1$, $\lvert X_2\rvert = n_2$, $\lvert Y\rvert = m$ and $\lvert Y'\rvert = m'$
$\texttt{alldifferent}(X)$	$\frac{m!}{(m-n)!} \cdot p^n$
$\texttt{among}(X, Y', N)$	$\binom{n}{N} m'^N (m-m')^{n-N} \cdot p^n$
$\texttt{nvalue}(X, N)$	$\binom{m}{N} \cdot a_{n,N} \cdot p^n$
$\texttt{atmostNValues}(X, N)$	$\sum_{k=1}^{N} \binom{m}{k} a_{n,k} \cdot p^n$
$\texttt{atleastNValues}(X, N)$	$\sum_{k=N}^{n} \binom{m}{k} a_{n,k} \cdot p^n$
$\texttt{occurrence}(X, y, N)$	$\binom{n}{N} (m-1)^{n-N} \cdot p^n$
$\texttt{atmost}(X, y, N)$	$\sum_{k=1}^{N} \binom{n}{k} (m-1)^{n-k} \cdot p^n$
$\texttt{atleast}(X, y, N)$	$\sum_{k=N}^{n} \binom{n}{k} (m-1)^{n-k} \cdot p^n$
$\texttt{uses}(X, X_1, X_2)$	$m^{n-n_1-n_2} \cdot \sum_{k=1}^{m} \binom{m}{k} a_{n_1,k} k^{n_2} \cdot p^n$
$\texttt{disjoint}(X, X_1, X_2)$	$m^{n-n_1-n_2} \cdot \sum_{k=1}^{min(n_1,m)} \sum_{l=1}^{min(n_2,m-k)} \binom{m}{k} \binom{m-k}{l} a_{n_1,k} a_{n_2,l} \cdot p^n$

Proposition 12. *Let $N \in \mathbb{N}$,*

$$\mathbb{E}\left(\#\texttt{occurrence}(X, y, N)\right) = \binom{n}{N} (m-1)^{n-N} \cdot p^n \tag{19}$$

Proof. The proof is the same as Proposition 11 in the case where $Y' = \{y\}$ is a singleton. □

Proposition 12 can also be generalized to the case where N is a variable.

4.5 Synthesis

We report the estimators of the number of solutions in Table 1 for several cardinality constraints. We observe a pattern in all these formulae: the estimation of the number of allowed tuples is always p^n multiplied by the number of tuples allowed by the constraint if every domain were equal to the set of values Y (if the value graph were complete). This remark leads to the following Proposition.

Proposition 13. *Let C be a constraint over X with $\lvert X\rvert = n$, Y be the union of the domains and p the edge density in the value graph $G_{X,Y}$, then:*

$$\mathbb{E}\left(\#C\right) = \#C^* \cdot p^n \tag{20}$$

with $\#C^*$ the number of allowed tuples if $G_{X,Y}$ were complete.

Proof. Let $\mathcal{S}_{C^*}$ be the set of allowed tuples if $G_{X,Y}$ were complete. For each $s \in \mathcal{S}_{C^*}$, let Z_s be the random variable such that, $Z_s = 1$ if s is in the set of allowed tuples $\mathcal{S}_C$ of C, and $Z_s = 0$ otherwise. A solution s is an instantiation

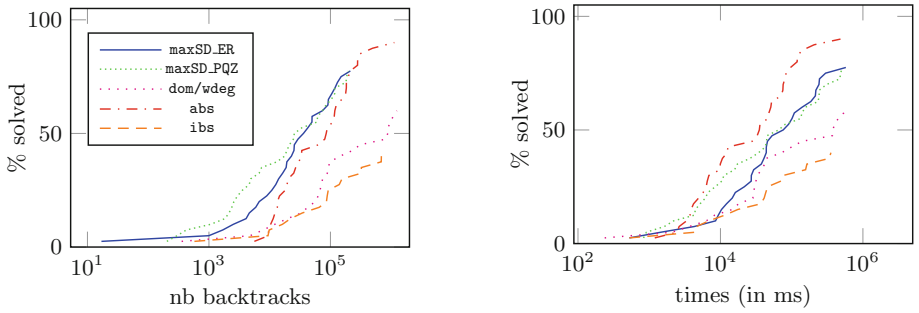


Fig. 2. Performances of `maxSD_ER`, `maxSD_PQZ`, `dom/wdeg`, `ibs` and `abs` on 40 hard Latin Square instances, in number of backtracks (left) and time (right).

of every variable, then, in the Erdős-Renyi Model, $\mathbb{P}(\{Z_s = 1\}) = \mathbb{E}(Z_s) = p^n$. Then,

$$\mathbb{E}(\#C) = \mathbb{E}\left(\sum_{s \in \mathcal{S}_{C^*}} Z_s\right) = \sum_{s \in \mathcal{S}_{C^*}} \mathbb{E}(Z_s) = \#C^* \cdot p^n$$

□

In Sect. 3, we have shown how to count solutions on a `range` and a `roots` constraints and in Sect. 4, how to use the `range` and `roots`/decomposition to estimate the number of solutions on many cardinality constraints. Proposition 13 highlights a general pattern for such estimates. In Sect. 5, we experiment these probabilistic estimators within counting-based heuristics on some problems using cardinality constraints.

5 Experimental Analysis

In this section, we present two problems, on which we have run different heuristics: `maxSD` [13], `dom/wdeg` [3], `abs` (activity-based search) [10] and `ibs` (impact-based search) [15]. This benchmark has been chosen by taking the problems in XSCP, CSPLib, MiniZinc which matched our testing needs: no COP, with cardinality constraints at the core of the problem but no gcc. Also, the lack of knowledge on how to use `maxSD` on problems with several constraints restricts a lot the practical use of the heuristic. These conditions restricted our benchmark to Latin Squares and Sports Tournament Scheduling.

`maxSD` consists in choosing a pair variable/value based on the estimation of the number of remaining solutions. More precisely, for each constraint, and for each pair variable/value in this constraint, we compute an estimation of the number of remaining allowed tuples and we associated with each pair a solution density. `maxSD` chooses the pair variable/value that maximizes the solution density among every constraint.

We actually do not run maxSD as presented in [13], but a slightly different version. It consists in re-computing the ordering of the variables only when the product of the domains size have decreased enough, as suggested in [6]. Here, we set a threshold at 20%. Also, the coefficients $a_{n,m}$, the binomial coefficients and the factorials are computed in advance. The computation of the approximations is thus made in linear time in n.

We first introduce the problem and the cardinality constraints that are used in the model and then compare their efficiency in terms of solving time and number of required backtracks. The instances and the strategies are implemented in Choco solver [14] and we run them on a 2.2 GHz Intel Core i7 with 2.048 GB.

5.1 Latin Square Problem

A Latin Square problem is defined by a $n * n$ grid whose squares each contain an integer from 1 to n such that each integer appears exactly once per row and column [12]. The model uses a matrix of integer variables and an `alldifferent` constraint for each row and each column. We tested on the 40 hard instances used in [13] with n = 30 and 42% of holes (corresponding to the phase transition), generated following [7]. For these instances, we also compare our probabilistic estimator (`maxSD_ER`) with the estimator that is proposed in [13] (`maxSD_PQZ`) for `alldifferent`. We set a time limit to 10 min.

Figures 2 represent the percentage of solved instances in function of the number of required backtracks, and of the solving time. The strategies `maxSD` (for both estimators `maxSD_ER` and `maxSD_PQZ`) and `abs` performed better than `dom/wdeg` and `ibs`. `abs` solved more instances than the two versions of `maxSD`, but required more backtracks. `maxSD` seems to perform better on the easiest instances (in term of number of backtracks). `maxSD_PQZ` has slightly better performances than `maxSD_ER` on the medium instances and have very comparable performances on the hardest ones.

5.2 Sports Tournament Scheduling Problem

This problem is taken from [19] and is presented as follows: the problem is to schedule a tournament of n teams over $n-1$ weeks, with each week divided into $n/2$ periods, and each period divided into two slots. A tournament must satisfy the following three constraints: every team plays once a week; every team plays at most twice in the same period over the tournament; every team plays every other team. The first and the third constraint are modeled with an `alldifferent` constraint and the second one is modeled with an `atmost` constraints. We run this problem with the different settings: $n \in \{6, 8, 10, 12, 14\}$.

In Table 2, we report the number of backtracks required (and the time required) to solve the problem for different values of n with four different heuristics. Here `maxSD_PQZ` cannot be used as there is no estimator for `atmost` in the previous work of [13]. Consequently, we only focused on our approach `maxSD_ER`. We fixed a time limit to 5 min. We observe that `maxSD_ER` outperforms `abs` and

dom/wdeg. For $n \in \{6, 8, 10\}$, maxSD_ER and ibs have similar performances but ibs could not find a solution in less than 5 min for $n = 12$ and $n = 14$.

Table 2. Number of backtracks (time in s) for different settings of n

n	n = 6	n = 8	n = 10	n = 12	n = 14
maxSD_ER	60 (0.239)	**10** (0.707)	**1056** (3.587)	**74168 (92.396)**	**37883 (128.272)**
ibs	**3 (0.172)**	214 (**0.648**)	1232 (**1.865**)	TO	TO
abs	101 (0.077)	3081 (0.692)	246767 (24.207)	TO	TO
dom/wdeg	89380 (3.829)	TO	TO	TO	TO

We have shown that our probabilistic estimator for alldifferent gives very comparable result than the estimator given in [13] on the Latin Square instances. Also our estimators within maxSD_ER gives better results than ibs, abs and dom/wdeg on the Sport Tournament Scheduling problem.

6 Conclusion

In this paper, we have presented a method to estimate the number of solutions of the range and roots constraints with a probabilistic Erdős-Renyi Model. We can estimate the number of solutions of ten cardinality constraints using their range and roots decompositions. We detailed our method on alldifferent, nvalue, among and occurrence and we report our estimators with atmostNValues, atleastNValues, atmost, atleast, uses and disjoint. We highlighted a general formula to compute such an estimation on cardinality constraints. We have implemented the heuristic maxSD_ER with these new probabilistic estimators and compare their efficiency to dom/wdeg, abs, and ibs.

We think that the main asset of this approach is its systematic nature. We have shown here an application of counting solutions for counting based search. Such an approach could also be used, for example, for uniform random instances generation, probabilistic reasoning or search space structure analysis.

We did not study the gcc constraint in this article, as its decomposition involves several non-disjoint subsets of the variables. Further research includes extending our approach to the case where several range and roots constraints may apply to a common set of variables. This will lead us to estimators of the number of solutions for conjunctions of cardinality constraints, or gcc constraints.

References

1. Beldiceanu, N., Contejean, E.: Introducing global constraints in chip. Math. Comput. Modell. **20**(12), 97–123 (1994). https://doi.org/10.1016/0895-7177(94)90127-9. http://www.sciencedirect.com/science/article/pii/0895717794901279

2. Bessiere, C., Hebrard, E., Hnich, B., Kiziltan, Z., Walsh, T.: Range and roots: two common patterns for specifying and propagating counting and occurrence constraints. Artif. Intell. **173**(11), 1054–1078 (2009). https://doi.org/10.1016/j.artint.2009.03.001
3. Boussemart, F., Hemery, F., Lecoutre, C., Sais, L.: Boosting systematic search by weighting constraints. In: de Mántaras, R.L., Saitta, L. (eds.) Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004, Including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, 22–27 August 2004, pp. 146–150. IOS Press (2004)
4. Carlsson, M., Fruehwirth, T.: Sicstus PROLOG user's manual 4.3. Books On Demand - Proquest (2014)
5. Erdos, P., Renyi, A.: On random matrices. Publication of the Mathematical Institute of the Hungarian Academy of Science (1963)
6. Gagnon, S., Pesant, G.: Accelerating counting-based search. In: van Hoeve, W.-J. (ed.) CPAIOR 2018. LNCS, vol. 10848, pp. 245–253. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93031-2_17
7. Gomes, C., Shmoys, D.: Completing quasigroups or latin squares: a structured graph coloring problem, January 2002
8. Gomes, C.P., Hoffmann, J., Sabharwal, A., Selman, B.: From sampling to model counting. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, Hyderabad, India, 6–12 January 2007, pp. 2293–2299 (2007)
9. Meel, K.S., et al.: Constrained sampling and counting: universal hashing meets SAT solving. CoRR abs/1512.06633 (2015). http://arxiv.org/abs/1512.06633
10. Michel, L., Van Hentenryck, P.: Activity-based search for black-box constraint programming solvers. In: Beldiceanu, N., Jussien, N., Pinson, É. (eds.) CPAIOR 2012. LNCS, vol. 7298, pp. 228–243. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29828-8_15
11. Pachet, F., Roy, P.: Automatic generation of music programs. In: Jaffar, J. (ed.) CP 1999. LNCS, vol. 1713, pp. 331–345. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-540-48085-3_24
12. Pesant, G.: CSPLib problem 067: quasigroup completion. http://www.csplib.org/Problems/prob067
13. Pesant, G., Quimper, C., Zanarini, A.: Counting-based search: branching heuristics for constraint satisfaction problems. J. Artif. Intell. Res. **43**, 173–210 (2012). https://doi.org/10.1613/jair.3463
14. Prud'homme, C., Fages, J.G., Lorca, X.: Choco solver documentation. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S. (2016). http://www.choco-solver.org
15. Refalo, P.: Impact-based search strategies for constraint programming. In: Wallace, M. (ed.) CP 2004. LNCS, vol. 3258, pp. 557–571. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30201-8_41
16. Régin, J.: A filtering algorithm for constraints of difference in CSPS, pp. 362–367 (1994)
17. Russell, S.J., Norvig, P.: Artificial Intelligence - A Modern Approach, 3rd edn. Pearson Education, London (2010). http://vig.pearsoned.com/store/home/1,1205,store-14563_id-294438,00.html
18. Stanley, R.P.: Enumerative Combinatorics: Volume 1, 2nd edn. Cambridge University Press, Cambridge (2011)
19. Walsh, T.: CSPLib problem 026: sports tournament scheduling. http://www.csplib.org/Problems/prob026