Closure of independencies under graphoid properties: some experimental results

M. Baioletti[1], G. Busanello[2], B. Vantaggi[2]

[1] Dept. Matematica e Informatica, Università di Perugia, Italy
[2] Dept. Metodi e Modelli Matematici, Università “La Sapienza” Roma, Italy

Abstract

In this paper we describe an algorithm for computing the closure with respect to graphoid properties of a set of independencies. The computation of the closure, with respect to graphoid properties (as well as with respect to semigraphoid ones) is infeasible since its size is exponentially larger than the size of the given set J of independence statements (see [2, 3]). Then, it is necessarily to find suitable reduced set of independence statements (obviously included in the closure of J with respect to graphoids), which is as smallest as possible and it represents the same independence structure. From this reduced set, called “fast closure”, all the relations in the closure should be easily deducible, then it can be considered a basis for the closure. The algorithm, to compute the reduced set by considering graphoids, is based on a unique inference rule introduced in [1]. In the quoted paper we have also compared this algorithm with another based on two inferential rules, which are deduced from [3] and studied in our previous paper. This topic by considering essentially semigraphoid structures has already been successfully solved by Studený in [2, 3].

An empirical evaluation of the performance of the introduced algorithm is provided by showing computation times and number of iterations, as well as a comparison between the needed time to compute the fast closure and the time for computing the complete closure (the size of both closures is compared).

Algorithm FC1

Let $\tilde{S} = \{Y_1, \dots, Y_n\}$ be a finite not empty set of variables and $S = \{1, \dots, n\}$ the set of indices associated to $\tilde{S}$. Given a (coherent) probability P, a conditional independence statement $Y_A \perp\!\!\!\perp Y_B | Y_C$ (compatible with P), where A, B, C are disjoint subsets of S, is simply denoted by the ordered triple (A, B, C). We denote with $S^{(3)}$ the set of all ordered triples (A, B, C) of disjoint subsets of S, such that A and B are not empty. In this case an independence model $\mathcal{I}$, related to P, is a subset of $S^{(3)}$. We recall that an independence model arising from the classical independence notion is closed under semigraphoid properties, that are the following ones:

G1 if $(A, B, C) \in \mathcal{I}$, then $(B, A, C) \in \mathcal{I}$ (Symmetry);

G2 if $(A, B, C) \in \mathcal{I}$, then $(A, B', C) \in \mathcal{I}$ for any nonempty subset B' of B (Decomposition);

G3 if $(A, B_1 \cup B_2, C) \in \mathcal{I}$ with B_1 and B_2 disjoint, then $(A, B_1, C \cup B_2) \in \mathcal{I}$ (Weak Union);

G4 if $(A, B, C \cup D) \in \mathcal{I}$ and $(A, C, D) \in \mathcal{I}$, then $(A, B \cup C, D) \in \mathcal{I}$ (Contraction).

If the probability is strictly positive the model is also closed under graphoid properties, it means that G1–G4 hold together with the following rule

G5 if $(A, B, C \cup D) \in \mathcal{I}$ and $(A, C, B \cup D) \in \mathcal{I}$, then $(A, B \cup C, D) \in \mathcal{I}$ (Intersection).

Given a set of triples J, in the following we denote the closure of J with $\bar{J}$.

Let us focus our attention on the first three graphoid rules. Given a triple $\theta_2 \in S^{(3)}$, it is possible to compute all the triples θ_1 which can be obtained from θ_2 with a finite number of applications of G1, G2 and G3. We say (see [1]) that, for any such pair of triples, θ_1 is *generalized–included* in θ_2 (briefly g–included), in symbol $\theta_1 \sqsubseteq \theta_2$, and it means that

(i) $C_2 \subseteq C_1 \subseteq X_2$;

(ii) either $A_1 \subseteq A_2$ and $B_1 \subseteq B_2$ or $A_1 \subseteq B_2$ and $B_1 \subseteq A_2$.

Generalized inclusion is symmetrize version of *dominance relation* introduced by Studený in [2].

The g–inclusion between triples is extended to the case of sets of triples.

Definition 1 *Let H, J be subsets of $S^{(3)}$. J is a covering of H (in symbol $H \sqsubseteq J$) if and only if for any triple $\theta \in H$ there exists a triple $\theta' \in J$ such that $\theta \sqsubseteq \theta'$.*

In [1] we introduced the concept of “maximal”(with respect to g–inclusion) triple: given a set J of triples, a triple τ is maximal in J if there exists no $\bar{\tau} \in J$ with $\bar{\tau} \neq \tau, \tau^T$ such that $\tau \sqsubseteq \bar{\tau}$. In particular, we denote with $J_{/\sqsubseteq}$ the subset of J composed only by its maximal triples and we call FindMaximal the function which computes $J_{/\sqsubseteq}$ from J. Moreover, the set $J_* = \textsc{FindMaximal}(\bar{J})$ is said *fast closure*.

We recall first of all that the fast closure $\{\theta_1, \theta_2\}_*$ of a couple $\theta_1, \theta_2 \in S^{(3)}$ is composed by a maximum of nine extra triples, no matter how many variables occur in θ_1 and θ_2.

In fact, any pair of triples (θ_1, θ_2) can be re–written, in a general form, as

$$\begin{aligned}\theta_1 &= ([A_A, A_B, A_C, A_N], [B_A, B_B, B_C, B_N], [C_A, C_B, C_C, C_N])\\ \theta_2 &= ([A_A, B_A, C_A, A'_N], [A_B, B_B, C_B, B'_N], [A_C, B_C, C_C, C'_N])\end{aligned}$$

where some sets can be empty and with the notation that $[A, B, C]$ stands for $A \cup B \cup C$.

Moreover, the related fast closure $\{\theta_1, \theta_2\}_*$ is g–included to the set of possible triples

$K(\theta_1, \theta_2) = \{\theta_1, \theta_2, \theta_a, \theta_b, \theta_c, \theta_d, \theta_e, \theta_f, \theta_g, \theta_h, \theta_{ad}\}$, where

$$\begin{aligned}\theta_a &= (A_A, [A_B, B_A, B_B, B_C, C_B, B_N], [A_C, C_A, C_C]);\\ \theta_b &= (A_B, [A_A, B_A, B_B, B_C, C_A, B_N], [A_C, C_B, C_C]);\\ \theta_c &= (B_A, [A_A, A_B, A_C, B_B, C_B, A_N], [B_C, C_A, C_C]);\\ \theta_d &= (B_B, [A_A, A_B, A_C, B_A, C_A, A_N], [B_C, C_B, C_C]);\\ \theta_e &= (A_A, [A_B, B_A, B_B, B_C, C_B, B'_N], [A_C, C_A, C_C]);\\ \theta_f &= (A_B, [A_A, B_A, B_B, B_C, C_A, A'_N], [A_C, C_B, C_C]);\\ \theta_g &= (B_A, [A_A, A_B, A_C, B_B, C_B, B'_N], [B_C, C_A, C_C]);\\ \theta_h &= (B_B, [A_A, A_B, A_C, B_A, C_A, A'_N], [B_C, C_B, C_C]);\\ \theta_{ad} &= ([A_B, B_A], [A_A, B_B], [A_C, B_C, C_A, C_B, C_C]).\end{aligned}$$

By using $\{\theta_1, \theta_2\}_*$, we provided the Algorithm FC1(J).

```
function FC1($J$)
begin
    $J_0 := N_0 := J$
    $k :=0$
    repeat
        $k := k + 1$
        $N_k := \{\tau : \tau \in \{\theta_1, \theta_2\}_* \text{ with } \theta_1 \in J_{k-1}, \theta_2 \in N_{k-1}\}$
        $J_k := \textsc{FindMaximal}(J_{k-1} \cup N_k)$
    until $J_k = J_{k-1}$
    return $J_k$
end
```

For each J subset of $S^{(3)}$ then $\text{FC1}(J) \sqsubseteq J_*$ and $J_* \sqsubseteq \text{FC1}(J)$.

Experimental results

The experiments were performed on an AMD Dual Core Opteron running at 1.8 GHz with 2 GByte main memory. We applied a cut–off of 5,000,000 triples that can be stored (to avoid problems with memory) and a time–out of 3600 seconds.

In the first set of experiments, we have generated 200 random sets of triples having nv variables and nr triples, for $nr = 10, 15, 20, 25, 30$ and $nv = \lfloor 0.5 \cdot nr \rfloor, nr, \lfloor 1.5 \cdot nr \rfloor, 2nr$. and we have computed the fast closure by means of FC1, the main results are shown in Table 1. In particular, the value *perc* is the percentage of the sets for which FC1 has been able to compute the fast closure, within the limits of time and memory, *time* is the average computation times in seconds, *size* is the average size of the fast closure, *iter* is the average number of iterations needed to find the closure, and *gen* is the average number (rounded to the nearest integer) of the overall generated triples.

In the second set of experiments we compare the computation time needed for finding the complete closure and its size with respect to the time and size of the fast closure. The complete closure is obtained by using an algorithm similar to FC1, which uses all the inference rules G1–G5, without calling FindMaximal. Furthermore, we did not apply for it any cut–off with respect to number of triples. Since we expect that the complete closure is much larger than its fast version, we have performed these new experiments with smaller instances, instead of using the previous one. In particular, we generate 20 sets of nr triples and nv variables, for $nr = 4, 7, 10$ and $nv = nr, \lfloor 1.5 \cdot nr \rfloor$. The comparison of the size between fast and complete closure is impressive, as it is possible to see in the graph of the following figures.

Table 1: Fast Closure FC1

nr	nv	perc	time	size	iter.	gen.
10	5	100%	0	10.83	3.99	202
10	10	100%	1.06	95.93	6.42	27524
10	15	99%	44.43	226.08	6.263	241219
10	20	98.5%	22.16	153.54	4.81	115006
15	7	100%	9.11E-02	46.84	5.50	5841
15	15	63%	500.42	982.68	10.03	1926990
15	22	80.5%	111.49	365.29	6.63	359213
15	30	98%	9.77	72.14	3.25	32615
20	10	100%	79.19	433.835	7.41	652608
20	20	27.5%	376.43	921.47	10.2	1105693
20	30	93.5%	84.64	305.21	5.58	240052
20	40	98.5%	3.64	54.95	2.20	16514
25	12	49.5%	1383.23	1354.33	8.3	5231558
25	25	35%	254.46	719.69	9.04	720993
25	37	97.5%	14.25	124.42	3.8	62761
25	50	100%	1.1E-03	29.685	1.445	84
30	15	0%	–	–	–	–
30	30	51.28%	118.59	514.58	7.65	3631898
30	45	100%	0.03	48.38	2.41	1063
30	60	100%	8.55E-05	31.06	1.12	7

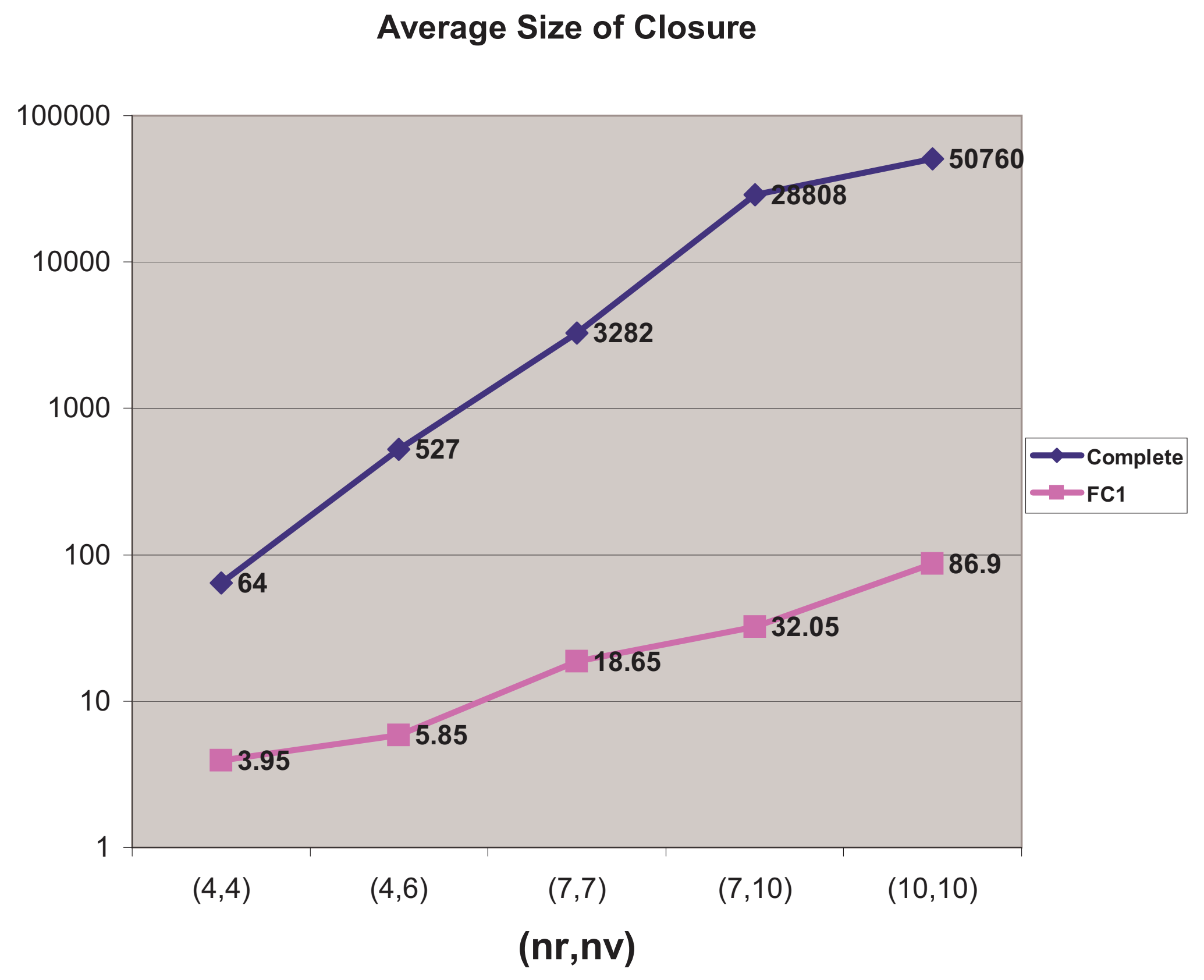

Figure 1: Size of closure.

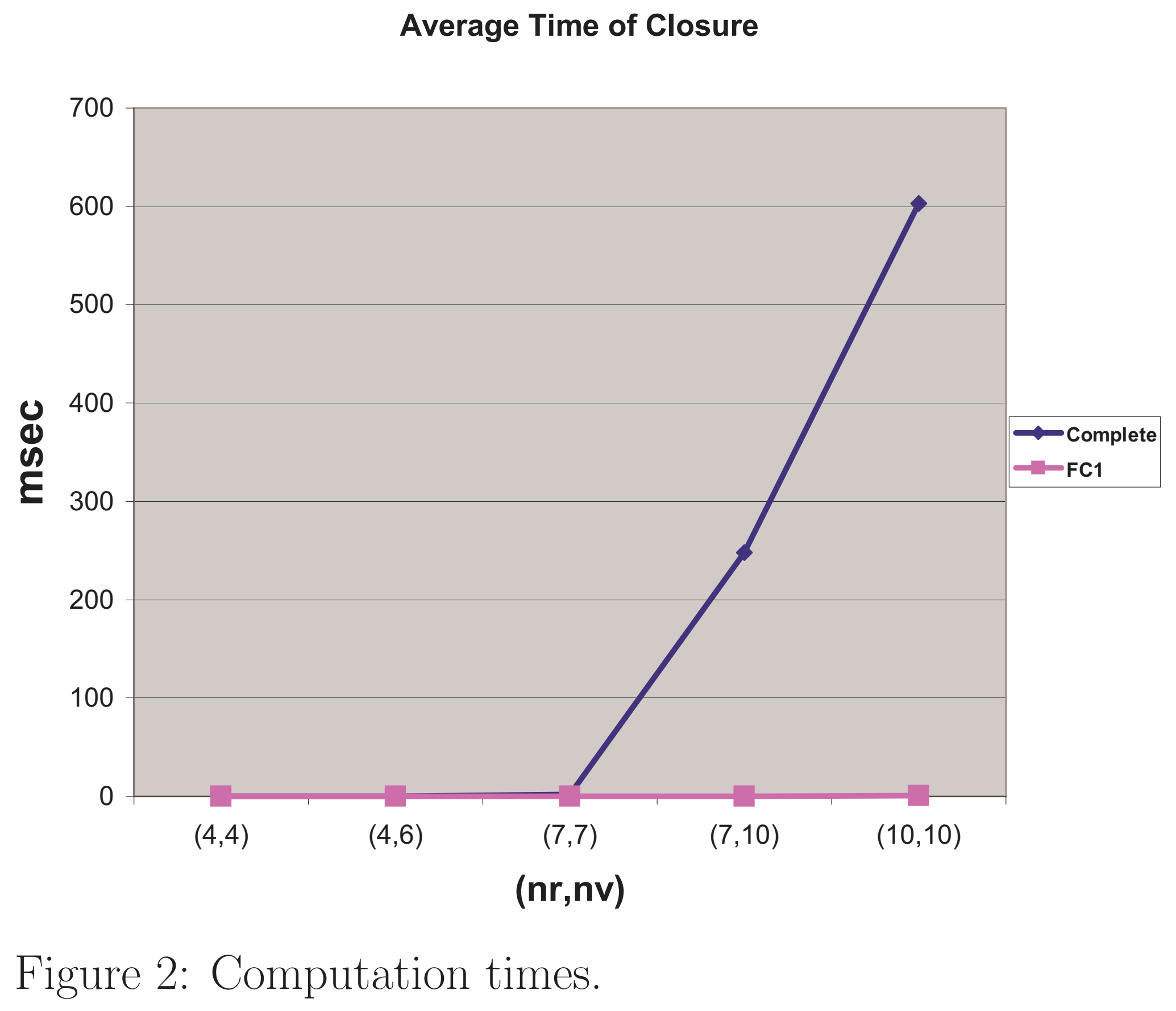

Figure 2: Computation times.

References

[1] Baioletti M., Busanello G., Vantaggi B. (2009), Conditional independence structure and its closure: Inferential rules and algorithms, *International Journal of Approximate Reasoning*, in press doi: 10.1016/j.ijar.2009.05.002.

[2] Studený M. (1997), Semigraphoids and structures of probabilistic conditional independence, *Ann. Math. Artif. Intell.*, 21, pp. 71–98.

[3] Studený M. (1998), Complexity of structural models, *Proc. Prague Stochastics ’98*, Prague, pp. 521–528.