CHICAGO JOURNAL OF THEORETICAL COMPUTER SCIENCE 2022, Article 01, pages 1–8
http://cjtcs.cs.uchicago.edu/

On the complexity of counting feedback arc sets

Kévin Perrot*

Received September 29, 2020; Revised March 10, 2022; Published March 26, 2022

Abstract: In this note we study the computational complexity of feedback arc set counting problems in directed graphs, highlighting some subtle yet common properties of counting classes. Counting the number of feedback arc sets of cardinality k and the total number of feedback arc sets are #P-complete problems, while counting the number of minimum feedback arc sets is only proven to be #P-hard. Indeed, this latter problem is $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$-complete, hence if it belongs to #P then $\mathsf{P} = \mathsf{NP}$.

1 Introduction

Feedback arc sets are natural objects to consider in digraphs, and deciding, given a digraph G and an integer k, whether it contains a feedback arc set of cardinality at most k is one of Karp's 21 NP-complete problems [5] (with a reduction not parsimonious). In this note we study the computational complexity of various feedback arc set counting problems. Section 2 gives the required definitions, in Section 3 we prove that counting the number of feedback arc sets of cardinality k and counting the total number of feedback arc sets are #P-complete problems, in Section 4 we study the special case of counting the number of minimum feedback arc set (which is not in #P unless $\mathsf{P} = \mathsf{NP}$), and Section 5 remarks that we can derive identical results on counting feedback vertex sets.

*Université Publique, France.

The author is thankful to Madhav Marathe for helpful correspondence. The author is affiliated to Université Côte d'Azur, CNRS, I3S, UMR 7271, Sophia Antipolis, France, and Aix Marseille Université, Université de Toulon, CNRS, LIS, UMR 7020, Marseille, France. This work received financial support from the *Young Researcher* project ANR-18-CE40-0002-01 "FANs", the project ECOS-CONICYT C16E01, the project STIC AmSud CoDANet 19-STIC-03 (Campus France 43478PD).

Key words and phrases: counting complexity, feedback arc sets, #P, counting minimum solutions.

DOI: 10.4086/cjtcs.2022.001

2 Preliminaries

We denote $[n]$ the set of integers $\{1,\dots,n\}$. We call *graph* an undirected graph $G=(V,E)$ with $E \subseteq \{\{u,v\} \mid u,v \in V\}$, and *digraph* a directed graph $G=(V,A)$ with $A \subseteq V \times V$. We consider all our graphs and digraphs to be loopless. A *vertex cover* $C \subseteq V$ of a graph $G=(V,E)$ is a subset of vertices intersecting every edge of G, *i.e.* verifying $\forall e \in E : e \cap C \neq \emptyset$. A *cycle of length* k in a digraph $G=(V,A)$ is a k-tuple of arcs $((u_1,v_1),(u_2,v_2),\dots,(u_k,v_k))$ such that $(u_i,v_i) \in A$ for all $i \in [k]$, $v_i = u_{i+1}$ for all $i \in [k-1]$, and $u_1 = v_k$. Let $\mathcal{C}(G)$ denote the set of cycles of G. If $\mathcal{C}(G) = \emptyset$ then G is called *acyclic*. A *feedback arc set* $F \subseteq A$ of a digraph $G=(V,A)$ is a subset of arcs intersecting every cycle of G, *i.e.* verifying $\forall c \in \mathcal{C}(G) : c \cap F \neq \emptyset$. Let $\mathtt{FAS}(G)$ (resp. $\mathtt{VC}(G)$) denote the set of feedback arc sets (FAS) (resp. vertex covers (VC)) of G.

Valiant defined the *class of counting problems* #P in [13, 12]. It is the class of functions counting the number of certificates of decision problems in NP, *i.e.* problems of the form "given x, compute $f(x)$", where f is the number of accepting paths of a nondeterministic Turing machine taking x as input and running in polynomial time. We denote $f \leq^p_{\text{parsi}} g$ when there exists a *polynomial parsimonious reduction* from counting problem f on alphabet Σ_f to counting problem g on alphabet Σ_g, *i.e.* a transformation $r : \Sigma_f^* \to \Sigma_g^*$ computable by a deterministic Turing machine running in polynomial time such that $\forall x \in \Sigma_f^* : f(x) = g(r(x))$. We denote $f \leq^p_T g$ when there exists a *polynomial Turing reduction* from f to g, *i.e.* a deterministic Turing machine computing f with oracle g, and running in polynomial time (hence making a polynomial number of calls to its oracle). Of course $f \leq^p_T g$ implies $f \leq^p_{\text{parsi}} g$.

#P-hardness is defined using either polynomial parsimonious reductions or polynomial Turing reductions (both reductions yield the same results). Note that, in contrast, Turing reductions are too strong for counting classes that are higher in the polynomial counting hierarchy [11]. Furthermore, #P is known to be closed under $\leq^p_{\text{parsi}}$, but this question is open for $\leq^p_T$ and it is known to be equivalent to $\mathsf{P} = \mathsf{NP}$ (see Section 4).

The authors of [3] have developed a remedy to the difficulty of classifying tightly some problems in #P (related to the unknown closure of #P for $\leq^p_T$), in particular minimality and maximality counting problems, that we will apply in Section 4. It consists in the *counting class* $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$, which is defined in terms of nondeterministic Turing machines where each accepting path outputs a number encoded in binary (rejecting paths have no output). We will call such machines *nondeterministic transducers*. Then $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$ is the class of functions counting the number of accepting paths outputting the minimal value (among all the values outputted by accepting paths), for some nondeterministic transducer running in polynomial time and outputting values whose binary encoding is of length $\mathcal{O}(\log n)$ (the definitions of [3] are more general in order to fit other levels of the polynomial counting hierarchy and other magnitudes of output length). In this paper we will only use the notion of hardness for $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$ under polynomial parsimonious reductions.

Our complexity results will be derived by reduction from the following problems.

Cardinality vertex cover (#Card-VC)
Input: A graph G and an integer k.
Output: $\lvert\{C \in \mathtt{VC}(G) \mid \lvert C\rvert = k\}\rvert$.

Theorem 2.1 ([2, p. 169])**. #Card-VC** *is* #P*-complete.*

Minimum cardinality vertex cover (#Minimum-VC)
Input: A graph G.
Output: $|\{C \in \mathtt{VC}(G) \mid |C| = m\}|$ with $m = \min\{|C| \mid C \in \mathtt{VC}(G)\}$.

Theorem 2.2 ([9, by identity reduction from Problem 4])**. #Minimum-VC** *is* #P*-hard.*

Theorem 2.3 ([3, Theorem 11])**. #Minimum-VC** *is* #$\cdot$OptP[log n]*-complete.*

3 #P-complete feedback arc set problems

Cardinality feedback arc set (#Card-FAS)
Input: A digraph G and an integer k.
Output: $|\{F \in \mathtt{FAS}(G) \mid |F| = k\}|$.

Feedback arc set (#FAS)
Input: A digraph G.
Output: $|\mathtt{FAS}(G)|$.

The first reduction adapts the classical construction from Karp [5].

Theorem 3.1. #Card-VC $\leq_T^p$ **#Card-FAS***, and* **#Card-FAS** *is* #P*-complete.*

Proof. **#Card-FAS** is in #P since given $G = (V,A)$ and k, one can guess nondeterministically a subset $F \subseteq A$ of size k, and then check in polynomial time that the digraph $(V, A \setminus F)$ is acyclic.

For the #P-hardness, as claimed we construct a polynomial Turing reduction from **#Card-VC** which is #P-hard (Theorem 2.1). Given an instance $G = (V,E)$ and k of **#Card-VC**, we consider an arbitrary order $\prec$ on V and the digraph $G'(\ell) = (V'(\ell), A'(\ell))$ with

$$V'(\ell) = \{v_i \mid v \in V \text{ and } i \in \{0,1\}\} \cup \{e_{i,j} \mid e \in E \text{ and } i \in \{0,1\} \text{ and } j \in [\ell]\},$$

$$A'(\ell) = \{(v_0, v_1) \mid v \in V\} \cup \Big\{(u_1, \{u,v\}_{0,j}), (\{u,v\}_{0,j}, v_0), (v_1, \{u,v\}_{1,j}), (\{u,v\}_{1,j}, u_0) \mid \{u,v\} \in E \text{ and } u \prec v \text{ and } j \in [\ell]\Big\},$$

for $\ell = k+1$ (the construction is illustrated on Figure 1). First, note that k is encoded in binary, but since $k \leq |V|$ the construction is polynomial in size. The idea is to have, for each edge of G, a large set of cycles in $G'(\ell)$. In $G'(\ell)$ there are $4\ell|E|$ arcs corresponding to edges of G that are "strictly less interesting" (when $\ell > 1$)[1] to construct an FAS than the $|V|$ arcs corresponding to vertices of G, because any cycle cut by some former edge can also be cut by some latter edge. As a consequence, there is a one-to-one correspondence between the minimum VC of G and the minimum FAS of $G'(\ell)$. Let $C(\kappa)$ be the answer to **#Card-VC** on G for some integer κ, let $F(k')$ be the answer to **#Card-FAS** on $G'(\ell)$ for some integer k', and let

$$m = \min\{|C| \mid C \in \mathtt{VC}(G)\} = \{|F| \mid F \in \mathtt{FAS}(G'(\ell))\}.$$

When $\ell > k' \geq m$, an FAS of size k' of $G'(\ell)$ can be seen as:

[1] If $\ell = 1$ then $k = 0$ and the answer to the **#Card-VC** instance is trivial.

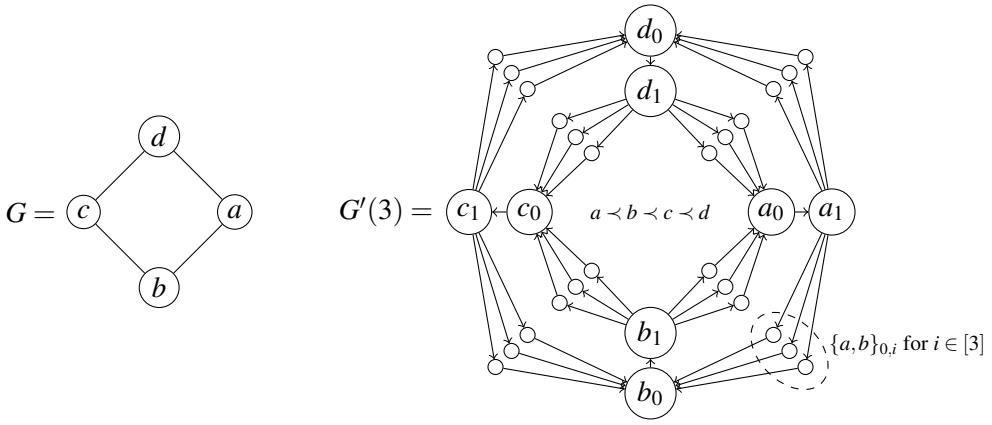

Figure 1: Illustration of the construction $G'(\ell)$ for $\ell = 3$ in the proof of Theorem 3.1.

- κ arcs corresponding to vertices of G forming a vertex cover for some $k' \geq \kappa \geq m$,
- $k' - \kappa$ arcs corresponding to edges of G (among the $4\ell|E|$),

because with $\ell > k'$ it is not possible to intersect all cycles corresponding to an edge of G with only arcs corresponding to edges of G. We can deduce the following relations between $C(\kappa)$ and $F(k')$:

$$\begin{aligned}
&F(0) = C(0) = 0\\
&\dots\\
&F(m-1) = C(m-1) = 0\\
&F(m) = C(m)\\
&F(m+1) = C(m+1) + \tbinom{4\ell|E|}{1}C(m)\\
&F(m+2) = C(m+2) + \tbinom{4\ell|E|}{1}C(m+1) + \tbinom{4\ell|E|}{2}C(m)\\
&\dots\\
&F(m+i) = C(m+i) + \textstyle\sum_{j=0}^{i-1}\tbinom{4\ell|E|}{i-j}C(m+j).
\end{aligned}$$

It is therefore possible to compute inductively $C(\kappa)$ from oracle calls to $F(1),\dots,F(\kappa)$ (corresponding to **#Card-FAS** instances $G'(\ell)$ with integers $k' = 1,\dots,\kappa$), and eventually compute $C(k)$ after a polynomial time since all calls are done for $k' \leq k \leq |V|$. □

The second reduction employs the Vandermonde matrix method from [9, 13].

Theorem 3.2. **#Card-FAS** $\leq_T^p$ **#FAS**, *and* **#FAS** *is* #P*-complete.*

Proof. Again **#FAS** is in #P since given $G = (V,A)$, one can guess nondeterministically a subset $F \subseteq A$, and then check in polynomial time that the digraph $(V, A \setminus F)$ is acyclic.

For the #P-hardness, as claimed we construct a polynomial Turing reduction from **#Card-FAS** which is #P-hard (Theorem 3.1). Given an instance $G = (V,A)$ and k of **#Card-FAS**, we consider the family of digraphs $H'(\ell) = (V'(\ell), A'(\ell))$ with

$$V'(\ell) = V \cup \{a_i \mid a \in A \text{ and } i \in [\ell-1]\},$$

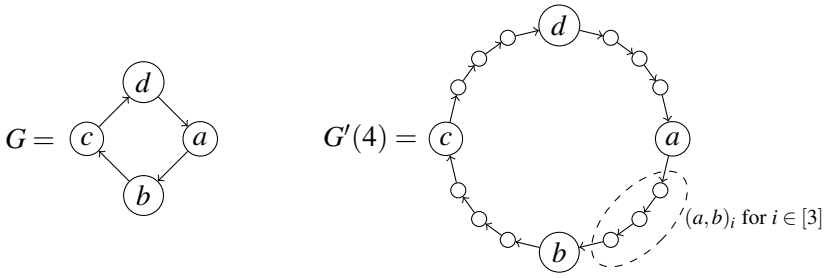

Figure 2: Illustration of the construction $H'(\ell)$ for $\ell = 4$ in the proof of Theorem 3.2.

$$A'(\ell) = \{(u,a_1),(a_1,a_2),\ldots,(a_{\ell-2},a_{\ell-1}),(a_{\ell-1},v) \mid a = (u,v) \in A\}$$

(the construction is illustrated on Figure 2). The idea is that an arc $a = (u,v)$ of G is replaced with a path of ℓ arcs in $H'(\ell)$ that we will denote

$$P_a = \{(u,a_1),(a_1,a_2),\ldots,(a_{\ell-2},a_{\ell-1}),(a_{\ell-1},v)\}.$$

Hence for each FAS F of G there corresponds a family $\Omega(F)$ of FAS of $H'(\ell)$, with elements of the form $\bigcup_{a\in A} S'_a$ where

$$S'_a \cap P_a \neq \emptyset \text{ if } a \in F$$
$$S'_a \cap P_a = \emptyset \text{ if } a \notin F.$$

The family $\Omega(F)$ consists of $(2^\ell - 1)^{|F|}$ FAS, and the families $\{\Omega(F) \mid F \in \mathtt{FAS}(G)\}$ partition $\mathtt{FAS}(H'(\ell))$. The number of FAS of $H'(\ell)$ is therefore

$$\Gamma(\ell) = \sum_{i=0}^{|A|} F_i(G)(2^\ell - 1)^i$$

where $F_i(G)$ is the number of FAS of G of cardinality $i \in \{0,\ldots,|A|\}$, *i.e.*

$$F_i(G) = |\{F \in \mathtt{FAS}(G) \mid |F| = i\}|.$$

The $(|A|+1)\times(|A|+1)$ matrix $M = (M_{\ell i})$ with entries $M_{\ell i} = (2^\ell - 1)^i$ for $\ell, i \in \{0,\ldots,|A|\}$ is Vandermonde[2] with bases distinct for all ℓ, as a consequence it is nonsingular and from $\Gamma(\ell)$ for $\ell \in \{0,\ldots,|A|\}$ we can compute $F_i(G)$ for $i \in \{0,\ldots,|A|\}$ in polynomial time (see [9, Lemma page 781]). □

Let us also mention the following variant.

Minimal feedback arc set (#Minimal-FAS)
Input: A digraph G.
Output: $\lvert\{F \in \mathtt{FAS}(G) \mid \forall F' \subsetneq F : F' \notin \mathtt{FAS}(G)\}\rvert$.

[2] A matrix (or its transpose) is Vandermonde when its coefficient on the i^{th} row and j^{th} column is of the form μ_i^j for some real numbers μ_i (indices start with 0 hence coefficients equal 1 on the first column).

It is proven in [10] that **#Minimal-FAS** is #P-complete, with a parsimonious reduction from counting the number of acyclic orientations of an undirected graph, itself proven to be #P-hard in [6]. **#Minimal-FAS** belongs to #P since the set of feedback arc sets is obviously closed by arc addition, hence one can check in polynomial time that for any $a \in F$ the digraph $(V, A \setminus (F \setminus \{a\}))$ is not acyclic.

Theorem 3.3 ([10, Corollary 1])**. #Minimal-FAS** *is* #P*-complete.*

4 Counting minimum feedback arc sets

Minimum feedback arc set (#Minimum-FAS)
Input: A digraph G.
Output: $|\{F \in \mathtt{FAS}(G) \mid |F| = m\}|$ with $m = \min\{|F| \mid F \in \mathtt{FAS}(G)\}$.

In contrast to counting the number of minimal objects, counting the number of minimum objects reveals some subtle facts about counting complexity classes. It is indeed not an obvious task at all to check in polynomial time whether a feedback arc set F is of minimum size (intuitively this would require to compute m in polynomial time, but deciding if $m \leq k$ for a given k is well known to be NP-complete [5]). Fortunately the literature on counting problems provides appropriate tools to characterize the complexity of **#Minimum-FAS**.

From standard techniques we derive the following.

Theorem 4.1. #Minimum-VC $\leq^{p}_{parsi}$ **#Minimum-FAS**,
#Minimum-FAS *is* #P*-hard and* #P*-easy.*

Proof. The construction $G'(2)$ from the proof of Theorem 3.1 is a polynomial parsimonious reduction from **#Minimal-VC** to **#Minimum-FAS**, which is therefore #P-hard by Theorem 2.2. Indeed, a minimum FAS in $G'(2)$ contains no arc corresponding to edges of G, hence there is a one-to-one correspondence between minimum FAS in $G'(2)$ and minimum VC in G.

For the #P-easiness, there is a straightforward polynomial Turing reduction to **#Card-FAS** which is in #P (Theorem 3.1): one can simply call the oracle with the digraph $G = (V, A)$ for $k = 0, 1, \ldots$ until the first time its answer is not zero (which happens at least when $k = |V|$), this is the number of minimum FAS. □

Although the class #P is closed under polynomial parsimonious reductions (meaning that $A \leq^{p}_{\text{parsi}} B$ and $B \in \#\mathsf{P}$ implies $A \in \#\mathsf{P}$), the closure of #P for polynomial Turing reductions is an open question[3] (and is equivalent to $\mathsf{P} = \mathsf{NP}$, as we will see). For more on closure properties of #P see [1, 7, 8, 11] The authors of [3] defined the complexity class $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$ (and analogous classes for problems higher in the polynomial counting hierarchy) for which minimality (or maximality) problems such as **#Minimum-FAS** are provably[4] complete.

[3] Note that the analogous question regarding the closure of NP under polynomial Turing reduction is also open, while NP is known to be closed under polynomial many-one reductions.

[4] More precisely with a proof of *appropriate difficulty*, since all these are complete if $\mathsf{P} = \mathsf{NP}$...

Theorem 4.2. **#Minimum-FAS** *is* $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$*-complete.*

Proof. **#Minimum-FAS** is in $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$ since given $G=(V,A)$, one can guess nondeterministically a subset $F \subseteq A$, then check in polynomial time that the digraph $(V, A \setminus F)$ is acyclic. If it not acyclic then reject, if it is acyclic then output the size of F encoded in binary and accept.

We already proved in Theorem 4.1 that **#Minimum-VC** $\leq^{p}_{\text{parsi}}$ **#Minimum-FAS**, hence from Theorem 2.3 **#Minimum-FAS** is $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$-hard. □

Thanks to a result of [3] we can nicely complement Theorem 4.1.

Corollary 4.3. *If* **#Minimum-FAS** *is in* $\#\mathsf{P}$ *then* $\mathsf{P}=\mathsf{NP}$.

Proof. If **#Minimum-FAS** is in $\#\mathsf{P}$ then $\#\cdot\mathsf{OptP}[\log \mathsf{n}] = \#\mathsf{P}$ from Theorems 4.1 and 4.2 (it straightforwardly holds that $\#\mathsf{P} \subseteq \#\cdot\mathsf{OptP}[\log \mathsf{n}]$), hence the result follows from a direct application of [3, Theorem 9] for $k=0$. □

5 Complexity of counting feedback vertex sets

There is a straightforward polynomial parsimonious reduction from counting feedback arc sets to counting feedback vertex sets in a digraph. A *feedback vertex set* (FVS) of a digraph G is a subset of vertices intersecting every cycle of G (*i.e.* containing, for every cycle of G, at least one vertex from its arcs). The reduction, as noted in [10], consists in considering the *line digraph* of $G=(V,A)$, denoted $L(G)=(V',A')$ and defined as

$$V'=A, \text{ and } ((u,v),(v,w)) \in A' \text{ for all } u,v,w \in V \text{ such that } (u,v),(v,w) \in A.$$

Indeed, there is one-to-one correspondence between the cycles of G and $L(G)$, and a one-to-one correspondence between the FAS of G and the FVS of $L(G)$. As a consequence ($\#\mathsf{P}$ and $\#\cdot\mathsf{OptP}[\log \mathsf{n}]$ being closed for $\leq^{p}_{\text{parsi}}$), all the results of this note also apply to counting the number of feedback vertex sets.

It is worth remembering that counting the number of minimum feedback vertex sets is known to be $\#\mathsf{P}$-hard even when restricted to planar digraphs [4].

References

[1] L. FORTNOW AND N. REINGOLD: PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996. 6

[2] M. R. GAREY AND D. S. JOHNSON: *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. 2

[3] M. HERMANN AND R. PICHLER: Complexity of counting the optimal solutions. *Theoretical Computer Science*, 410(38):3814–3825, 2009. 2, 3, 6, 7

[4] H. B. HUNT, M. V. MARATHE, V. RADHAKRISHNAN, AND R. E. STEARNS: The complexity of planar counting problems. *SIAM Journal on Computing*, 27:1142–1167, 1998. 7

[5] R. M. KARP: Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pp. 85–103, 1972. 1, 3, 6

[6] N. LINIAL: Hard enumeration problems in geometry and combinatorics. *SIAM Journal on Algebraic Discrete Methods*, 7:331–335, 1986. 6

[7] M. OGIHARA, T. THIERAUF, S. TODA, AND O. WATANABE: On closure properties of #P in the context of PF∘#P. *Journal of Computer and System Sciences*, 53(2):171–179, 1996. 6

[8] M. OGIWARA AND L. A. HEMACHANDRA: A complexity theory for feasible closure properties. *Journal of Computer and System Sciences*, 46(3):295–325, 1993. 6

[9] J. S. PROVAN AND M. O. BALL: The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12:777–788, 1983. 3, 4, 5

[10] B. SCHWIKOWSKI AND E. SPECKENMEYER: On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117(1):253–265, 2002. 6, 7

[11] S. TODA AND O. WATANABE: Polynomial-time 1-Turing reductions from #PH to #P. *Theoretical Computer Science*, 100(1):205–221, 1992. 2, 6

[12] L. G. VALIANT: The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979. 2

[13] L. G. VALIANT: The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8:410–421, 1979. 2, 4

AUTHOR

Kévin Perrot
assistant professor
Laboratoire d'Informatique et Systèmes, Aix-Marseille Univeristé, France
kevin.perrot@lis-lab.fr
https://pageperso.lis-lab.fr/kevin.perrot/