

## RÉSUMÉ

Lors de ce travail d'étude et de recherche nous avons expérimenté Feasibility Pump (FP) sur des instances binaires. Cette heuristique, initialement pensée pour trouver rapidement une solution pour des instances mixtes, se décompose en plusieurs étapes : arrondir, projeter et perturber. FP alterne entre des solutions faisables à variables non entières et des solutions non faisables à variables entières. Les premières sont arrondies et les secondes sont projetées sur le polytope des solutions faisables à variables non entières. Lorsque FP cycle une perturbation est déclenchée.

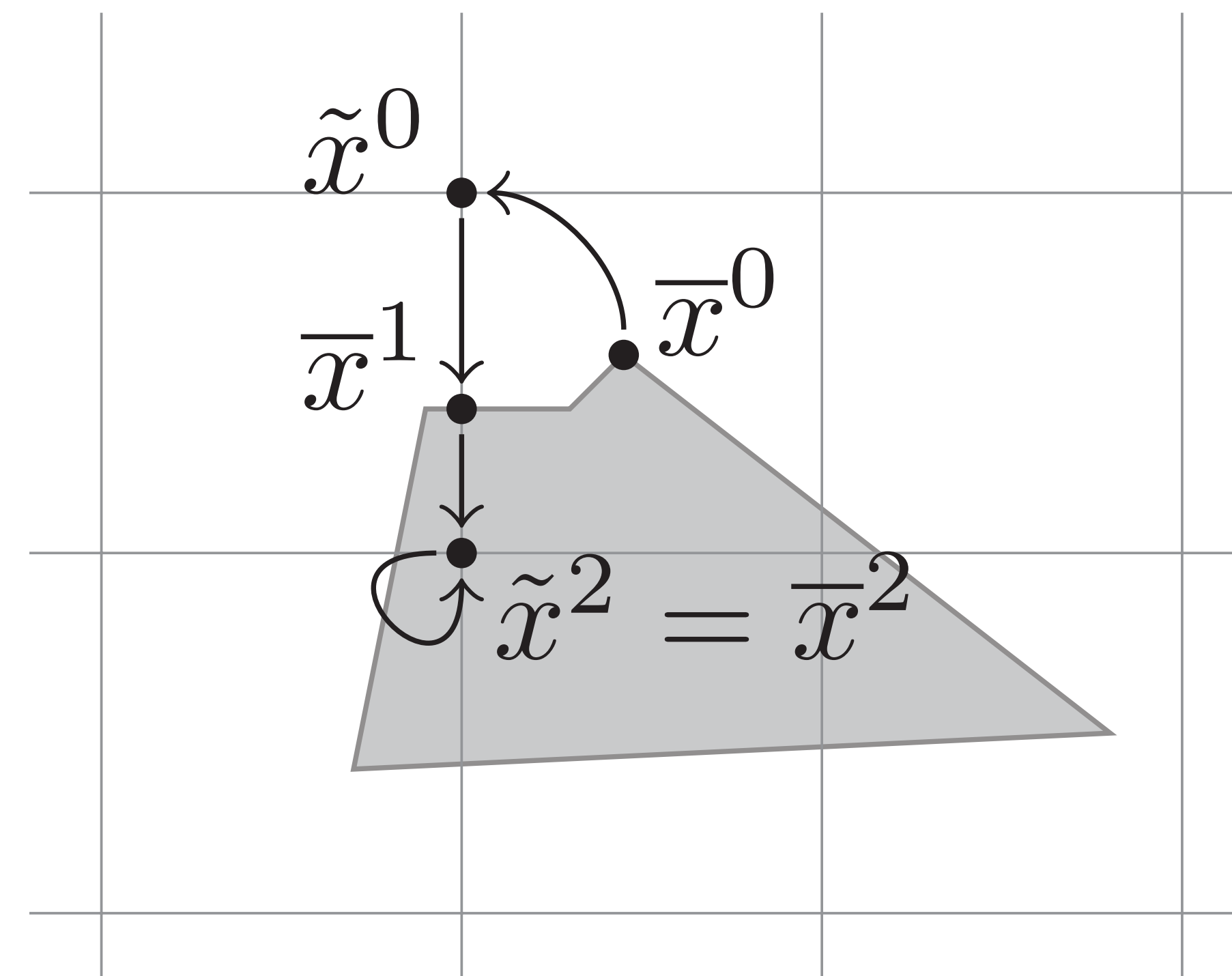
## ALGORITHME FEASIBILITY PUMP

**Algorithme** Feasibility Pump (FP)

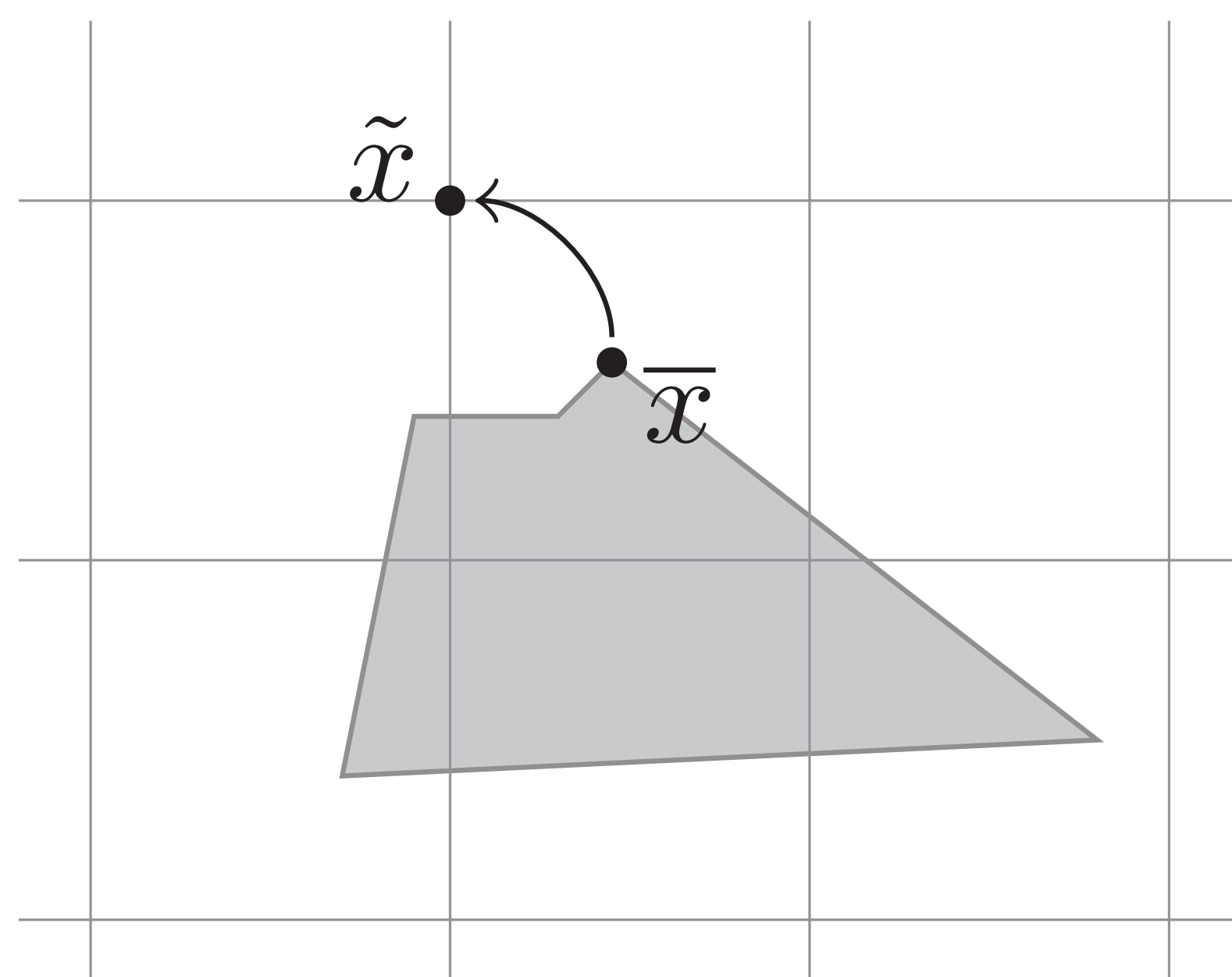
```

1: function FP( $P, k_{max}, T_{mps_{limit}}$ )
2:  $\bar{x}^0 \in \bar{X}, k \leftarrow 1$ 
3: if  $\bar{x}^0$  est binaire then return  $\bar{x}^0$  end if
4:  $\tilde{x}^0 \leftarrow \text{ARRONDIR}(\bar{x}^0)$ 
5: while  $k < k_{max}$  and  $T_{mps_{ecoule}} < T_{mps_{limit}}$  do
6:    $\bar{x}^k \leftarrow \text{PROJETER}(\tilde{x}^k)$ 
7:   if  $\bar{x}^k$  est binaire then return  $\bar{x}^k$  end if
8:    $\tilde{x}^k \leftarrow \text{ARRONDIR}(\bar{x}^k)$ 
9:   if  $\tilde{x}^k = \tilde{x}^{k-1}$  then
10:     $\tilde{x}^k \leftarrow \text{PERTURBER}(\tilde{x}^k, \bar{x}^k)$ 
11:   end if
12:    $k \leftarrow k + 1$ 
13: end while
14: return NULL
15: end function

```



## ARRONDIR



FP arrondit  $\bar{x}$  à la solution à variables entières la plus proche, quitte à ne plus respecter les contraintes.

## VARIANTES ET JULIA

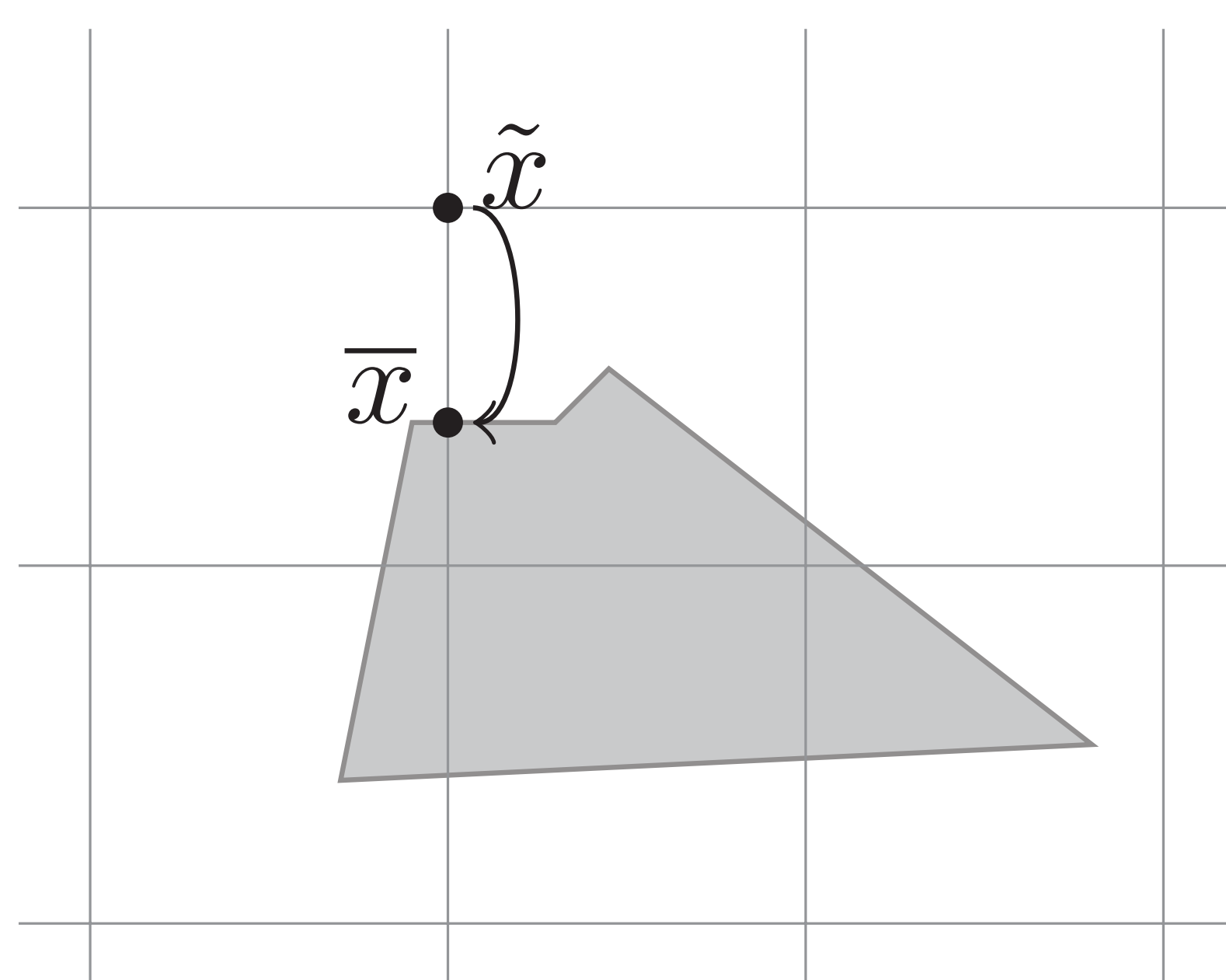
Une heuristique, des variantes :

- Objective feasibility pump par T. Achterberg et T. Berthold en 2007
- Arrondi avec le centre analytique par D. Baena et J. Castro en 2011
- Recursive central rounding par J. Naoum-Sawaya en 2014
- Nouvelle projection par M. De Santis, S. Lucidi et F. Rinaldi en 2014
- Variantes originales modifiant chacune des 3 étapes de FP, etc

Les avantages de Julia pour ce travail :

- Variantes rapides à implémenter pour les premières expérimentations
- Optimisation du code permettant une exécution rapide
- Extraction et analyse des résultats aisées et dans le même langage que les expérimentations
- Barrière du langage très faible pour la compréhension du problème et l'utilisation du code

## PROJETER



FP projette la solution  $\tilde{x}$  sur le polytope des solutions faisables non entières. C'est une projection au plus proche.

## PERTURBER

FP a tendance à cycler, quand cela arrive certaines variables de  $\tilde{x}$  sont basculées de 0 à 1 et de 1 à 0 en fonction du dernier  $\bar{x}$ . Perturbation possible : basculer tous les  $\tilde{x}_i$  tels que  $|\bar{x}_i - \tilde{x}_i| + \max\{p_i, 0\} > 0.5$  où les  $p_i$  sont tirés aléatoirement entre  $-0.3$  et  $0.7$ .

## EXPÉRIMENTATIONS

109 instances binaires provenant de MIPLIB 2017

FP	REDUCEFP	Succès	$T_{Min}$	$T_{Moy}$	$T_{Max}$	$z_{Min}$	$z_{Moy}$	$z_{Max}$
	OBJFP	8	27	33	32	18	23	22
	OBJFP	6	28	28	29	8	15	17
REDUCEFP	FP	8	18	19	21	21	16	16
REDUCEFP	OBJFP	8	23	25	22	13	13	14
OBJFP	FP	3	25	24	27	31	25	23
OBJFP	REDUCEFP	8	30	33	37	25	25	22

Instances	Succès	$T_{Moy}$ FP	$T_{Moy}$ REDUCEFP	$T_{Moy}$ OBJFP	Meilleur $z_{Moy}$
2club200v15p5scn	10/10/10	1.9		5.3	3.8 FP
air03	10/10/10	0.2		1	0.2 OBJFP
decomp1	0/0/10	-		8.2	- REDUCEFP
eilC76-2	10/10/10	10.5		27.1	4.8 REDUCEFP
ex1010-pi	10/10/10	13.9		12.8	24.9 OBJFP
fast0507	10/10/10	1.1		1.5	3.7 OBJFP
mitre	0/10/0	-		9.8	- REDUCEFP
n2seq36q	6/9/5	10.8		3.7	20.4 OBJFP
reblock166	10/10/10	4.6		3.7	5.8 OBJFP
sorell4	0/0/10	-		-	1.4 OBJFP
tbfp-network	8/1/1	13.5		18.3	27.2 OBJFP
vpphard	0/0/0	-		-	-

## RÉFÉRENCES

- M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005
- T. Achterberg, and T. Berthold. Improving the feasibility pump. *Discrete Optimization*, 4(1):77–86, 2007
- T. Berthold, A. Lodi, and D. Salvagnin. Ten years of feasibility pump, and counting. *EURO Journal on Computational Optimization*, 7(1):1–14, 2019
- MIPLIB 2017, <https://miplib.zib.de>, 2018