

Comparaison sur les instances `cap` des fonctions de coût monolithiques et des fonctions de coût décomposées (ici `regular` et `among`, les sames ne sont pas décomposées). On compare quatres cas :

1. MonoBase : fonctions de coût monolithiques et sans option sur l'ordre;
2. MonoSVO : fonctions de coût monolithiques avec l'option `-svo`;
3. DecBase : fonctions de coût décomposées et sans option sur l'ordre;
4. DecSVO : fonctions de coût décomposées avec l'option `-svo`.

Le timeout est fixé à 1000 secondes.

Instance	Opt	MonoBase	MonoSVO	DecBase	DecSVO
B1-LU04	92	0.38	0.29	0.04	0.05
B1-LU08	184	1.39	0.86	0.18	0.12
B1-LU16	368	14.91	2.95	2.87	0.35
B1-LU32	640	184.8	13.87	Timeout	1.40
B2-LU02	38	0.11	0.11	0.02	0.02
B2-LU04	116	0.25	0.32	0.04	0.04
B2-LU08	392	2.97	5.43	0.11	0.11
B2-LU16	752	5.13	13.92	0.26	0.36
B3-LU04	328	0.31	0.30	0.04	0.05
B3-LU08	656	0.65	0.62	0.10	0.09
B3-LU16	1312	1.48	1.38	0.30	0.28
B3-LU32	2592	3.96	3.64	1.67	1.62
B4-LU05	46	0.51	1.42	0.05	0.05
B4-LU10	192	1.83	4.85	0.10	0.13
B4-LU20	752	8.88	17.71	0.30	0.32
B4-LU40	1504	26.21	58.30	0.91	1.04
B1234-LU15	504	4.20	25.01	0.24	0.37
B1234-LU30	1148	28.59	Timeout	5.80	50.77
B1234-LU60	2824	29.58	Timeout	13.97	Timeout
B1234-LU120	5488	616.92	Timeout	553.59	Timeout
B1234-LU15-star	704	18.83	48.50	1.20	1.43
B1234-LU30-star	1560	127.07	Timeout	146.74	380.36
B1234-LU60-star	3852	Timeout	Timeout	375.1	Timeout
B1234-LU120-star	???	Timeout	Timeout	Timeout	Timeout

En observant les expérimentations, il me semble que devant le nombre de variables ajoutées l'heuristique de choix de variables me semble perdue. Du coup, on observe sur certaines instances des premières solutions de très mauvaises qualités et donc de très mauvais résultats (en temps de calcul) sur ces instances.

Le pré-traitement devient très long sur les grosses instances (351.69 seondes pour B1234-LU120-star et DecBase).