

Model-PPC 2019 - All-Interval Series

Brandon Fontany-Legall
Master Informatique - Université Côte D'Azur

April 2019

1 Introduction

Nous allons traiter la modélisation du problème All-Interval Series à l'aide de différentes méthodes de filtrage, de contraintes et de stratégies de recherches et tout ceci en Choco[1] (Java). Le problème All-Interval Series [2] est un problème qui consiste à trouver une série dans laquelle chaque nombre apparaît une seule fois et dans laquelle les intervalles entre les nombres voisins couvrent l'ensemble complet des intervalles. En définitive, le problème consiste en la découverte de permutation (x_1, \dots, x_N) de $0, 1, \dots, N-1$ telle que la liste $(\text{abs}(x_2-x_1), \text{abs}(x_3-x_2), \dots, \text{abs}(x_N-x_{N-1}))$ est une permutation de $1, 2, \dots, N-1$.

2 Trouver les solutions

Le problème est décrit dans deux tableaux. Le premier tableau que l'on nomme S permet de calculer les ... Le deuxième tableau est V, de son côté il permet le calcul des distances. Les deux tableaux sont alors soumis à une contrainte allDifferent pour trouver les solutions.

```
IntVar[] S = model.intVarArray("s", N, 0, N - 1, false);
IntVar[] V = model.intVarArray("V", N - 1, 1, N - 1, false);
for (int i = 0; i < N - 1; i++) {
    model.distance(S[i + 1], S[i], "=", V[i]).post();
}
model.allDifferent(S).post();
model.allDifferent(V).post();
```

Avec ce model simple, il suffit de choisir la meilleur stratégie de recherche pour optimiser le temps de résolution ainsi que le nombre de backtracks

3 Resultats

MacBook Pro (Retina, 15 pouces, mi-2015)
 MacOS X 10.14.3 (18D109)
 2,5 GHz Intel Core i7 (i7-4870HQ)
 16 Go 1600 MHz DDR3

3.1 Difference en fonction des contraintes

Ci-dessous sont présent les tableaux des résultats en fonction des contraintes ainsi que des modes AllDifférents. Tout les temps sont en secondes.

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	40	0.06333945	1109	0.010036602	1099	0.004494175	1109
10	296	0.27602643	16035	0.1343898	15897	0.06980992	16037
12	1328	1.8835303	315117	2.4533608	312567	1.3875043	315141
13	3200	7.599432	1545365	12.38341	1527433	7.1546626	1545639
14	9912	40.102074	8074423	66.84405	8011063	38.878334	8075021

Table 1: Avec le calcul de Distance

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	32	0.013790282	227	0.002993038	217	0.00236523	271
10	120	0.052844346	2045	0.036230963	1895	0.03439752	3263
12	648	0.49241814	23519	0.5252928	21415	0.59452015	50431
13	1328	2.050808	89787	2.173831	80171	2.8876228	224923
14	3200	8.9573	362643	9.479865	318751	14.738157	1070575

Table 2: Avec les tables

3.2 Difference en fonction des stratégies

Ci-dessous sont présent les tableaux des résultats en fonction des stratégies de recherches ainsi que des modes AllDifférents. Tout les temps sont en secondes.

3.2.1 minDomLBSearch

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	40	0.06333945	1109	0.010036602	1099	0.004494175	1109
10	296	0.27602643	16035	0.1343898	15897	0.0691524	16037
12	1328	1.8835303	315117	2.4533608	312567	1.3870184	315141
13	3200	7.59943	1545365	12.38341	1527433	7.050182	1545639
14	9912	40.102074	8074423	66.84405	8011063	38.41796	8075021

Table 3: minDomLBSearch Distance

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	32	0.013790282	227	0.002993038	217	0.00236523	271
10	120	0.052844346	2045	0.036230963	1895	0.03439752	3263
12	648	0.49241814	23519	0.5252928	21415	0.59452015	50431
13	1328	2.050808	89787	2.173831	80171	2.8876228	224923
14	3200	8.9573	362643	9.479865	318751	14.738157	1070575

Table 4: minDomLBSearch table

3.2.2 default

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	40	0.01918936	2883	0.021453563	2921	0.010192508	2481
10	296	0.20373505	37715	0.3639626	41737	0.1905525	41463
12	1328	3.8691323	723551	7.9883795	818975	3.4393244	714513
13	3200	24.733427	4464473	39.387047	3884893	20.855247	4242999
14	9912	94.53831	16045931	158.14989	15054157	105.5151	20157723

Table 5: Default Distance

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	32	0.006075061	507	0.006738246	481	0.004108037	487
10	120	0.116553225	6695	0.1445203	7355	0.07653591	6995
12	648	2.6133823	115915	2.8546593	105907	1.4928966	105637
13	1328	12.276937	481597	14.99421	501657	8.03643	503929
14	3200	74.14014	2614919	96.40158	2839025	51.321037	2899953

Table 6: Default Table

3.2.3 minDomUBSearch

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	40	0.060188424	1109	0.010298425	1099	0.004980067	1109
10	296	0.24738663	16035	0.13742606	15897	0.07194861	16037
12	1328	1.827717	315117	2.4743037	312567	1.465999	315141
13	3200	8.066886	1545365	12.654781	1527433	7.4995046	1545639
14	9912	44.04152	8074423	70.13292	8011063	40.53399	8075021

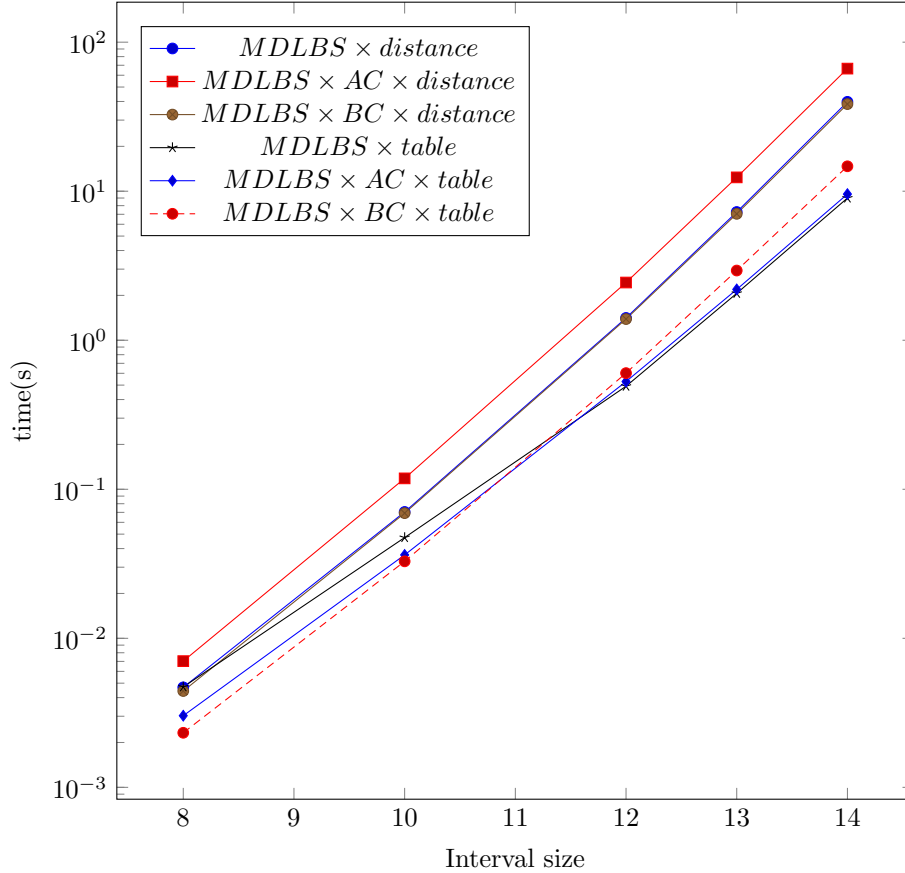
Table 7: minDomUBSearch Distance

N	Solutions	Default.time(s)	Default.backtrack	AC.time(s)	AC.backtrack	BC.time(s)	BC.backtrack
8	32	0.012811592	227	0.003180711	217	0.002518559	271
10	120	0.056706786	2045	0.0372998	1895	0.034916975	3263
12	648	0.58343226	23809	0.5388685	21415	0.62813467	50431
13	1328	2.1482427	89887	2.2536654	80171	3.0722392	224923
14	3200	9.30986	362797	9.801019	318751	15.461499	1070575

Table 8: minDomUBSearch Table

4 Interpretation des resultats

4.1 Difference distance / table



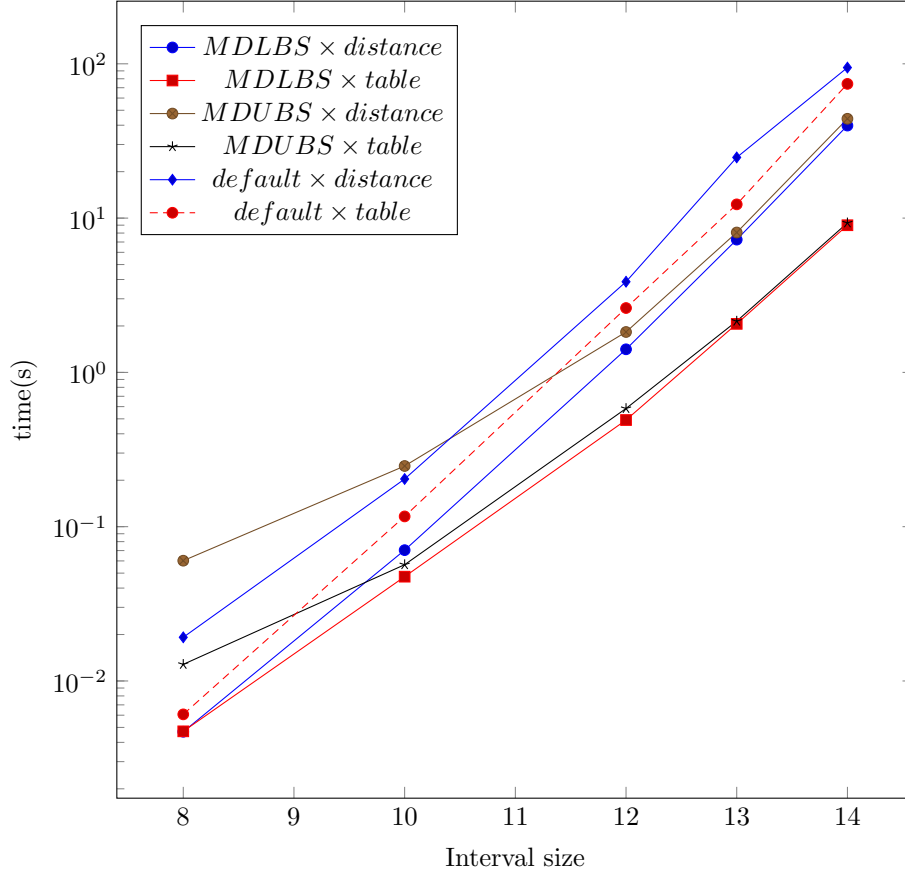
Après regroupement des données, nous obtenons ce graphique qui traite la différence entre les contraintes de distances vis-à-vis des contraintes à l'aide de tables. De plus, nous avons ajouté les différentes approches au niveau des All Different.

Nous remarquons dans un premier temps une légère différence de temps entre les différentes contraintes. En effet, les différences avec de petites valeurs ne se font pas grandement ressentir qui sont toutes dans l'intervalle $10^{-3} \dots 10^{-2}$. Cependant, lorsque l'intervalle devient plus grand les contraintes à l'aide de tables se démarquent.

Pour finir, nous remarquons que la combinaison contrainte/alldifferent crée de grandes disparités. Par exemple, utiliser AC avec une contrainte de distance est dans notre cas le plus long alors que l'utiliser avec des tables le rend le plus ef-

ficace. Pour conclure, il semblerait que le plus efficace au niveau distance/table serait l'utilisation des tables à l'aide d'un alldifferent AC.

4.2 Difference de stratégies



D'un autre côté, nous étudions la différence qu'il peut y avoir entre les différentes stratégies de recherche. Nous ne traiterons pas de toutes les stratégies proposées par Choco mais uniquement les stratégies minDomLBSearch, minDomUBSearch et défaut.

Nous remarquons rapidement que le choix de la stratégie de recherche a un impact significatif, nous ne mentionnerons bien évidemment pas la stratégie de recherche aléatoire. La stratégie de recherche par défaut semble être, dans notre cas de recherche, moins efficace que les autres que ce soit avec l'utilisation des tables ou des distances. Au niveau des deux minDomSearch, nous remarquons que les stratégies ont tendance à être de même efficacité lorsque l'intervalle augmente, et ceux, que ce soit avec les tables ou les distances. Cependant, les tables restent tout de même beaucoup plus efficaces.

Pour finir, dans notre cas de recherche, il semble que le choix de prendre les tables comme contraintes ainsi que la stratégie minDomLBSearch serait le choix le plus adapté.

5 Conclusion

Pour conclure, nous avons remarqué la différence entre contraintes de distance ainsi que la contrainte via tables dont les tables semblent être le plus efficace. De plus, le choix de la bonne stratégie est un enjeu majeur, qui, pour notre recherche, semble être la stratégie minDomLBSearch qui est la plus efficace.

Cependant, cette conclusion n'est pas vérifiable pour tous les problèmes de modélisations. Sachant de plus que nous ne dépassons pas l'intervalle de 14 car la résolution prend rapidement une allure exponentielle.

References

- [1] Choco-solver. <http://choco-solver.org/>.
- [2] Holger Hoos. CSPLib problem 007: All-interval series. <http://www.csplib.org/Problems/prob007>.