Supplementary Material

July 11, 2017

1 Cactus plot of UDGVNS versus CPLEX, DAOOPT, and INCOP+TOULBAR2

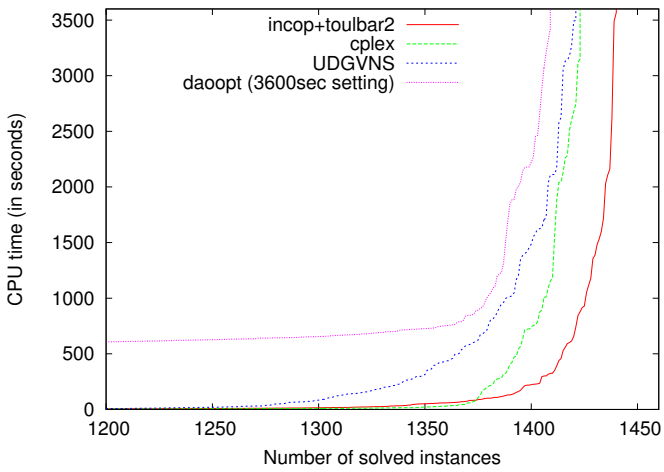

Figure 1: Number of instances solved by each method as time passes (UDGVNS = UDGVNS(k add1/jump, ℓ mult2)).

2 Anytime upper bound zoom for UDGVNS versus LDS

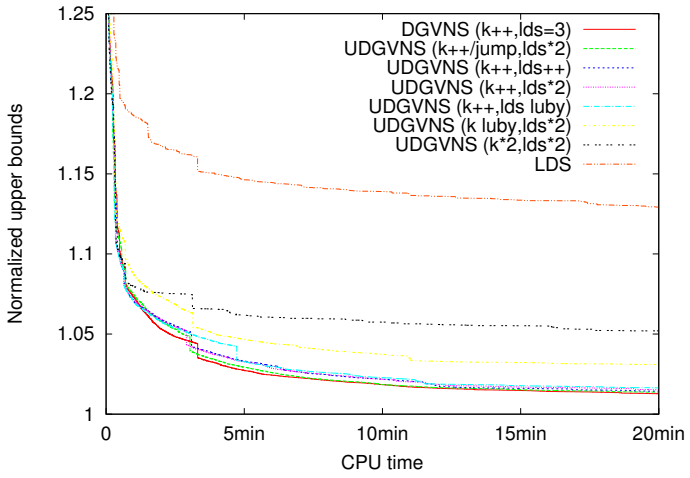

Figure 2: Anytime upper bound zoom for UDGVNS versus LDS.

3 Anytime upper bound of UDGVNS and UPDGVNS versus CPLEX

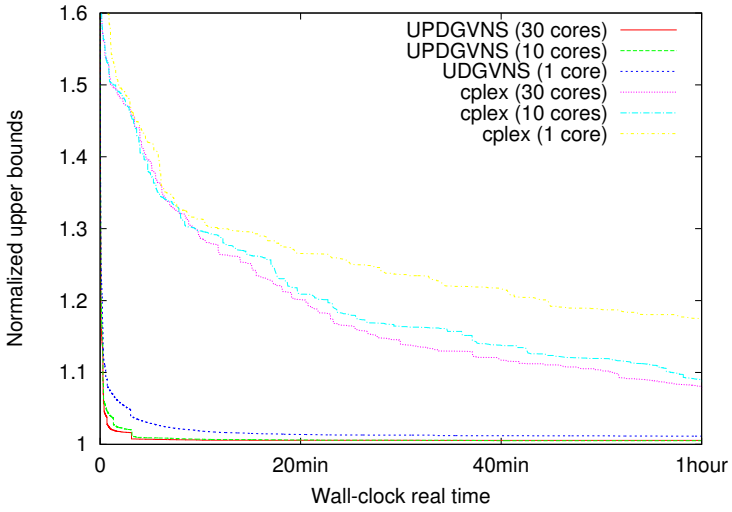

Figure 3: Anytime upper bound with 1 , 10 and 30 processors respectively for `cplex` and `UPDGVNS` (`UPDGVNS` = `UPDGVNS`(k `add1/jump`, ℓ `mult2`)).

4 Solving time and anytime behavior of U(P)DGVNS versus CPLEX and DAOOPT on UAI-Linkage category

Linkage *(optimum / worst solution)*	pedigree19 *(4625/21439)*	pedigree31 *(5258/166553)*	pedigree44 *(6651/104904)*	pedigree51 *(6406/629929)*
CPLEX (1 core)	790	59.3	6.35	36.23
CPLEX (10 cores)	191	9.00	2.48	9.43
CPLEX (30 cores)	**75**	**7.17**	**2.69**	**5.34**
DAOOPT (1 core)	375,110	16,238	95,830	101,788
DAOOPT (20 cores)	27,281	1,055	6,739	6,406
DAOOPT (100 cores)	7,492	201	1,799	1,578
UDGVNS (1 core)	- *(4949)*	- *(5258)*	- *(6722)*	- *(6406)*
UPDGVNS (10 cores)	- *(4762)*	3,341	- *(6651)*	- *(6406)*
UPDGVNS (30 cores)	- *(4626)*	1,775	- *(6651)*	- *(6406)*

Table 1: Best CPU time (in seconds) for sequential versions and best wall-clock time for multiple-core ones to find and prove optimality on Pedigree instances. A "-" means no proof of optimality in 1 hour (except DAOOPT with no time limit) (in parenthesis, unnormalized upper bound founds after 1 hour).

We report DAOOPT time from (OD17), obtained on a cluster of dual 2.67 GHz Intel Xeon X5650 6-core CPUs and 24 GB of RAM.

5 CPD instances filtering

In this paper we would like to evaluate VNS methods capability to solve difficult instances. Accordingly, we tried to generate new larger ones supposed to be more difficult to solve than those generated in (SAdG$^+$15). For this aims, in a first stage, we selected protein structures in the PDB databases (http://www.rcsb.org/pdb/home/home.do) with sizes range between 100 and 300 amino-acid. The resulting query has been filtered with the following criteria: resolution has to be lower than 2.5 A, membrane proteins, proteins complex, as well as proteins with disulfur bright have been removed, in addition, with proteins including non natural amino acid. We also discarded proteins with missing residues, out of the N and C terminal part of the sequence in order to select

a protein subset without any hole. Proteins with identity sequences higher than 90% are unselected. The corresponding remaining set includes 438 PDB references. Therefore, we sorted the 438 putative instances according to the average volume per variable and we extracted for benchmarking only the 20th first elements and used as benchmarking set. Each protein structure was fully redesigned according to (SAdG$^+$15) protocol. On the basis of energy matrix generated with a modified release of PYROSETTA.4 script (SAdG$^+$15) in order to use the last released Rosetta force-field (aka Beta November 2016) (A$^+$17). The instances characteristic contain from 130 up to $n = 282$ variables width maximum domain size from 383 to 438, and between 1706 and 6208 cost functions. The tree width ranges from 21 to 68, and from 0.16 to 0.34 for a normalized tree width.

In order to select well decomposable instances, after preliminary reorientation of each protein according to his inertial moment, we filtered the resulting PDB set by 3D geometrical criteria. By construction, due to the cutoff distance used for pairwise energy calculation and his related constraints in the model, Globular protein (.i.e spherical one) will correspond to CFN very nearby as click. In practice, the spherical shape which can be detected by a symmetric repartition of the Radius of gyration (Rg) component Rg(x), Rg(y), Rg(z). Indeed, Rg and his subcomponent are defined as the root mean square distance from each atom of the protein to their centroid or in the orthogonal plan to the related axis (x,y,z). Thus, for selecting non spherical protein , we first calculated the Rg_i respectively around x , y and z axis as and we filtered the proteins characterized by a ratio:

$$min(Rg(x)/Rg + Rg(y)/Rg + Rg(z)/Rg) \tag{1}$$

The $min(Rg(i)/Rg)$ is one figure to detect putative well structured instances.

Another structural criterion that can be used for structured instances filtering is a decreasing sort according to the approximated means space volume occupied associated to each variable. This volume is also related to the Gyration radius by the following equation:

$$\bar{V} = \frac{\frac{4}{3} * \pi * Rg^3}{|X|} \tag{2}$$

This criterion does not give at first sight a direct information about fold shape, but interestingly, both structural criteria produce the same 15th first instances set, with only few re-ranking, very likely because protein compactness got maximum bound and indeed a minimum average volume per amino-acid when the protein fold is close to sphere or vice versa for linear protein fold. After the 15th instances, the two critters remains less correlated. As a matter of fact, in the 5 next instances, the covering between the two sorted list is 2 over 5. On Account of computer power for benchmarking, we can not bench the full protein set.

pdbid	$\|\mathcal{X}\|$	d	e	tw	$tw/\|\mathcal{X}\|$	min(Rg_x/Rg)	$V(^3/var)$
5dbl	130	384	1,706	21	0.16	0.150	1,212.49
5jdd	263	406	5,220	41	0.16	0.239	655.58
3r8q	271	418	5,518	43	0.16	0.341	472.88
4bxp	170	439	2,636	33	0.19	0.316	457.81
1f00	282	430	6,208	51	0.18	0.269	439.28
2x8x	235	407	4,745	44	0.19	0.354	404.42
1xaw	107	412	1,623	28	0.26	0.308	378.04
5e10	133	400	2,286	34	0.26	0.294	344.68
1dvo	152	389	2,587	51	0.34	0.420	343.38
1ytq	181	415	3,449	54	0.30	0.392	332.69
2af5	292	410	5,693	68	0.23	0.427	330.23
1ng2	176	397	3,135	60	0.34	0.473	309.16
3sz7	151	450	2805	49	0.32	0.403	304.87
2gee	188	397	3,715	38	0.20	0.367	293.25
5e0z	136	420	2,367	36	0.26	0.362	279.00
1yz7	176	418	3,538	49	0.28	0.414	276.35
3e3v	154	436	2,976	37	0.26	0.367	251.97
3lf9	120	416	2,133	31	0.24	0.323	251.51
1is1	185	431	3,740	48	0.26	0.459	245.58
5eqz	138	434	2,567	33	0.24	0.338	241.93
4uos	188	383	4,161	44	0.23	0.347	234.11

Table 2: Characteristics of PDB instances: pdbid is the code reference in PDB database, $|\mathcal{X}|$ is the number of variable, d is the maximum domain size, e is number of cost functions, tw is the Min-fill tree width and $tw/|\mathcal{X}|$ a normalized tree width by $|\mathcal{X}|$. The two last columns correspond to structural criteria respectively defined in (1) and (2). Grey color corresponds to the re-ranking when one min(Rg_x/Rg) or $\bar{V}$ is used for PDB list sorting.

6 Comparing best solutions of TOULBAR2 vs VNS methods

Instance	Time (s)			Speed-up		
	(1)	(2)	(3)	(5/1)	(5/2)	(5/3)
5jdd	-	3,248±162	639±76	-	6.3	32.3
2x8x	2,087±62	1,012±64	475±54	33.1	68.3	145.7
1dvo	885±20	830±24	191±25	38.5	41.1	178.7
1ytq	2,215±231	1,280±40	277±26	7.7	13.3	61.6
2af5	-	2,217±57	856.±246	-	38.8	100.5
1ng2	542±2	400±12	241±26	71.4	96.8	160.7
3sz7	3,178±421	1,948±484	277±48	26	42.4	298.2
1yz7	970±4	675±16	428±83	86.4	124.1	195.8
3e3v	2,332±46	1,137±37	230±29	34.9	71.7	354.6
1is1	2,986±47	2,159±376	317±53	21.3	29.5	201.36
4uos	465±3	467±9	427±83,81	126	125.4	137.2

(a) Unsolved CPD instances.

Instance	Speed-up		
	(5/1)	(5/2)	(5/3)
5dbl	0.28	0.81	5.25
3r8q	-	3.66	32.14
4bxp	2.44	2.19	13.73
1f00	-	3.78	17.98
1xaw	-	4.82	10.78
5e10	0.70	0.83	8.87
2gee	3.04	3.93	17.55
5e0z	1.6	1.00	9.51
3lf9	1.63	2.98	12.34
5eqz	6.90	18.31	56.74

(b) Solved CPD instances.

Instance	Time(s)		
	(2*)	(3*)	(5)
5dbl	1,828.27	791.16	783.18
3r8q	-		41,700.1
4bxp	-		4,261.67
1f00	-		-
1xaw	-		2,917.04
5e10	839.52	196.43	1,171.98
2gee	-		9,795.59
5e0z	416.12	172.96	999.66
3lf9	-		2,960.64
5eqz	-		41,813

(c) Optimality proof.

Table 3: Tables (3a) and (3b) report, for each instance, the CPU times spent by VNS methods (within the 3600-seconds time limit) to obtain the best solution computed by TOULBAR2 (5). A '–' indicates that the corresponding solver was not able to compute a solution of equal/better quality than `toulbar2`. (1) : `VNS`(k `add1`$, \ell = 3$) (2): `UDGVNS`(k `add1`$, \ell = 3$) (3): `UPDGVNS`(npr, k `add1`$, \ell = 3$). Table (3c) reports the CPU-times required by `UDGVNS`, `UPDGVNS` and TOULBAR2 to prove the optimum within the 24-hours time limit. A '–' indicates that the corresponding solver failed to prove optimality. (2*): `UDGVNS`(k `add1/jump`$, \ell$ `mult2`) (3*): `UPDGVNS`(npr, k `add1/jump`$, \ell$ `mult2`).

References

[A+17] R Alford et al. The Rosetta all-atom energy function for macromolecular modeling and design. *bioRxiv*, 2017.

[OD17] L Otten and R Dechter. And/or branch-and-bound on a computational grid. page 84p., 2017. Unpublished.

[SAdG+15] D Simoncini, D Allouche, S de Givry, C Delmas, S Barbe, and T Schiex. Guaranteed discrete energy optimization on large protein design problems. *J. of Chemical Theo. and Comput.*, 11(12):5980–5989, 2015.