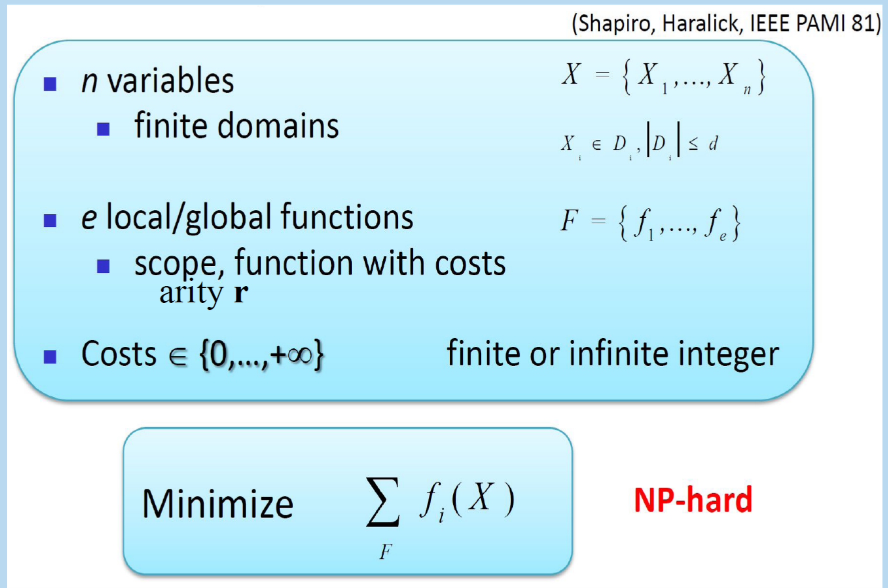
Parallel Hybrid Best-First Search for Cost Function Networks

A. Beldjilali, D. Allouche, S. de Givry , ✉ {david.allouche,simon.de-givry}@inra.fr
MIAT, INRA Toulouse, France

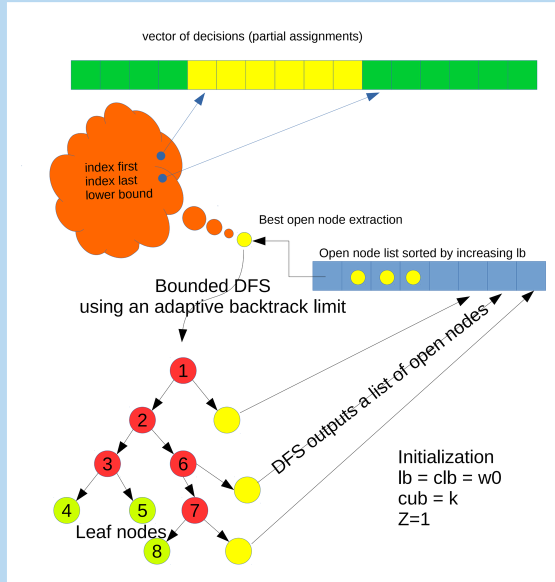
Objectives :

This work describes a first parallel release of Hybrid Best-First Search (HBFS) branch and bound algorithm to solve Cost Function Network minimization problems [Cooper et al.2010].
We performed experiments on various benchmarks using different architectures (multi-core servers, multi-node clusters) and compared with state-of-the-art parallel solvers (cplex, daoopt).

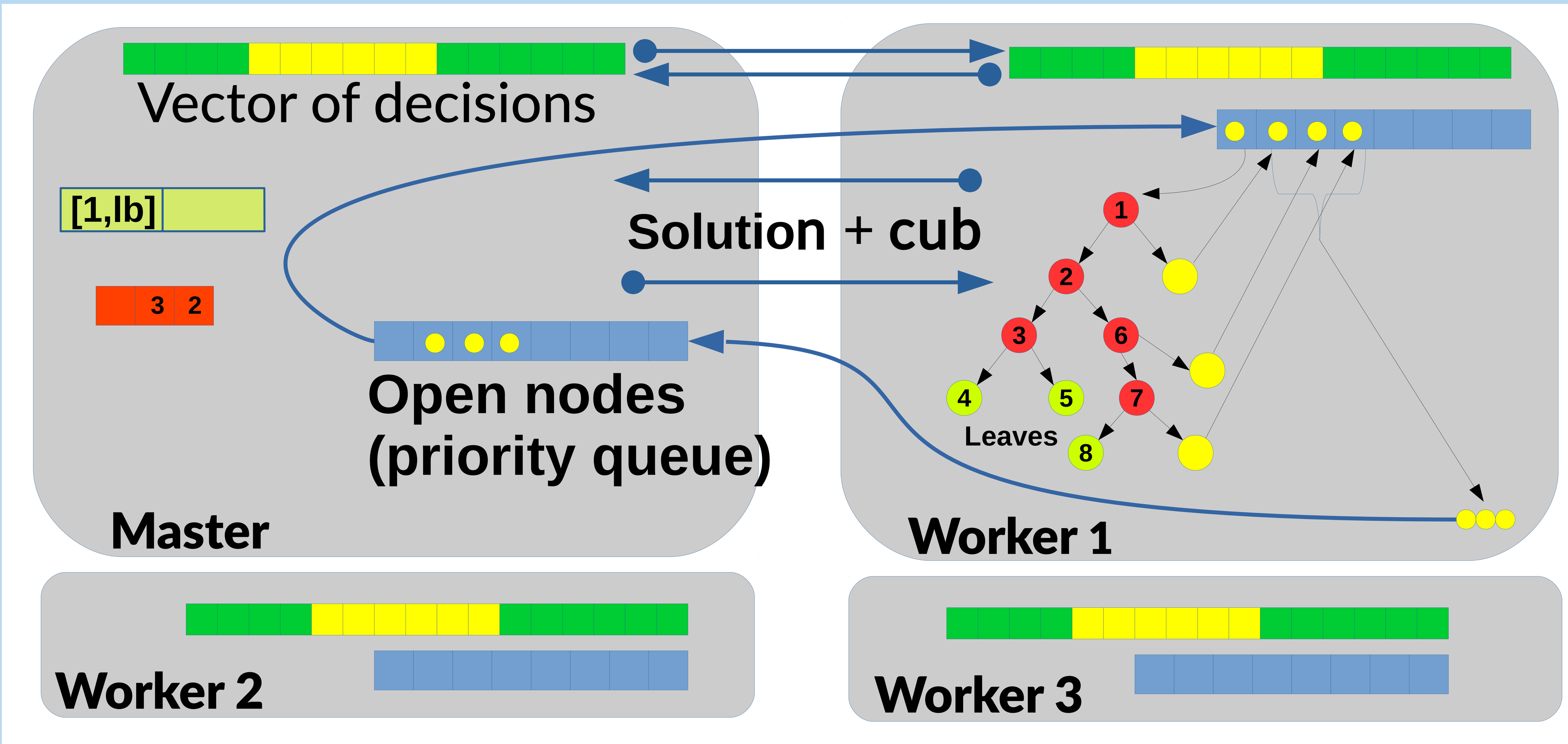
Cost Function Networks :



Hybrid Best-First Search (HBFS) :



Parallel Hybrid Best-First Search (PHBFS) :



PHBFS is based on the Master-Worker parallel paradigm [Ralphs et al.2018] where the Master is in charge of the open node frontier and dispatches the current best (with minimum lower bound) open node plus the current best solution found so far to the next available Worker. The Worker performs a bounded DFS starting from the received node and returns to the Master the resulting list of open nodes (DFS limit of 3 backtracks in the figure). Each open node is associated to a corresponding lower bound and a vector of search decisions. The Worker also returns the best solution found during its limited search if any. Only the Master has a global view of the whole search and reports optimality gaps until the proof of optimality is reached (when the current best frontier lower bound, including active Worker starting nodes, is equal or greater than the cost of the best solution found so far).

Computational Protein Design :

Instance	n	d	Time (sec.)		Speed-up
			HBFS	HBFS-24	
1xaw	107	412	721.43	568.50	1.27
3lf9	120	416	407.28	407.92	1
5dbl	130	384	122.84	171.82	0.71
5e10	133	400	147.73	198.23	0.75
5e0z	136	420	148.26	193.41	0.77
5eqz	138	434	3,366.11	1,049.11	3.21
1dvo	152	389	622.03	463.29	1.34
4bxp	170	439	327.46	395.16	0.83
1is1	185	431	-	2,545.82	-
2gee	188	397	797.22	863.64	0.92
5jdd	263	406	-	2,758.98	-
3r8q	271	418	1,605.30	1,294.97	1.24
1f00	282	430	-	2,140.40	-

Rosetta CPD benchmark [Ouali et al.2017]. A '–' indicates that the corresponding method failed to prove optimality in less than 1 hour.

Another CPD benchmark :

Instance	n	d	cplex	cplex-10	HBFS	HBFS-10	Speed-up
1UBI	13	198	-	-	1,023	214.02	4.78
2DHC	14	198	-	-	8.2	5.83	1.41
2DRI	37	186	-	-	135.5	30.00	4.52
1CDL	40	186	-	-	392.6	54.95	7.14
1CM1	42	186	-	6,177	6.6	6.11	1.08
1BRS	44	194	-	-	555.3	107.86	5.15
1GVP	52	182	-	-	596.1	185.75	3.21
1RIS	56	182	-	-	129.7	36.23	3.58
3CHY	74	66	-	5,259	88.7	20.71	4.28

OSPREY CPD benchmark [Allouche et al.2014]. A '–' indicates prove optimality failure in less than $9,000$ seconds.

Uncapacitated warehouse location

Instance	n	d	Time (sec.)		Speed-up
			HBFS	HBFS-24	
capmo1	200	100	10.92	5.14	2.12
capmo2	200	100	1.80	2.04	0.88
capmo3	200	100	6.09	3.73	1.63
capmo4	200	100	4.36	3.21	1.36
capmo5	200	100	2.69	2.58	1.04
capmp1	400	200	172.57	80.64	2.14
capmp2	400	200	95.15	61.35	1.55
capmp3	400	200	75.04	56.25	1.33
capmp4	400	200	107.86	78.4	1.38
capmp5	400	200	81.15	52.9	1.53
capmq1	600	300	679.43	412.52	1.65
capmq2	600	300	841.80	503.64	1.67
capmq3	600	300	647.67	431.47	1.5
capmq4	600	300	1,093.55	514.42	2.13
capmq5	600	300	1,388.41	701.61	1.98

Uncapacitated warehouse location benchmark [Larrosa et al.2005] with n, number of variables, and d, maximum domain size.

Linkage Analysis :

	pedigree19	pedigree31	pedigree44	pedigree51
n	793	1,183	811	1,152
d	5	5	4	5
cplex	790	59.30	6.35	36.23
//10	191(4.14)	9.00(6.59)	2.48 (2.56)	9.43 (3.84)
//30	75(10.53)	7.17(8.27)	2.69 (2.36)	5.34 (6.78)
daoopt	375,110	16,238	95,830	101,788
//20	27,281 (13.75)	1,055 (15.39)	6,739 (14.22)	6,406 (15.89)
//100	7,492(50.07)	201 (80.79)	1,799 (53.27)	1,578 (64.50)
HBFS	3,126	4.34	39.72	1,608
//10	434.27 (7.20)	1.51 (2.87)	6.08 (6.53)	179.22 (8.97)
//20	227.02 (13.77)	1.39(3.12)	3.18(12.49)	72.30(22.24)
//100	119.43(26.17)	0.97(4.47)	1.64 (24.22)	31.40 (51.21)

Experiments with different number of cores (speed-up in parentheses) [Favier et al.2011].

Acknowledgements :

Thanks to the "Agence nationale de la Recherche" for founding support (ANR-16-C40-0028) and to the Genotoul bioinformatics platform Toulouse Midi-Pyrenees for computational support.

Conclusion/Perspective :

Parallel HBFS is a first parallel approach for HBFS. It provides interesting results on several instances, outperforming in some cases state of the art solvers like cplex and daoopt. Even if the scalability of our approach must be subject of deeper investigation, due to the minimal size of the information shared between the Master and the Workers, the approach is very likely compliant with a larger number of cores. We found that the speed-up was very instance dependent, and must be also investigated.
As future work, we will take into account the structure of CFNs by parallelizing Backtrack with Tree Decomposition (BTD-HBFS) [Allouche et al.2015]. The resulting parallel method could replace LDS inside a parallel large neighborhood search strategy [Ouali et al.2017].

References :

[Toulbar2] : https ://github.com/toulbar2/toulbar2

[Allouche et al.2014] Allouche, D.; André, I.; Barbe, S.; Davies, J.; de Givry, S.; Katsirelos, G.; O'Sullivan, B.; Prestwich, S.; Schiex, T.; and Traoré, S. 2014. Computational protein design as an optimization problem. Artificial Intelligence 212 :59–79.

[Allouche et al.2015] Allouche, D.; de Givry, S.; Katsirelos, G.; Schiex, T.; and Zytnicki, M. 2015. Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP. In Proc. of CP, 12–28.

[Cooper et al.2010] Cooper, M.; de Givry, S.; Sanchez, M.; Schiex, T.; Zytnicki, M.; and Werner, T. 2010. Soft arc consistency revisited. AI 174 :449–478.

[Favier et al.2011] Favier, A.; Givry, S.; Legarra, A.; and Schiex, T. 2011. Pairwise decomposition for combinatorial optim. in graphical models. In Proc. of IJCAI, 2126–2132.

[Larrosa et al.2005] Larrosa, J.; de Givry, S.; Heras, F.; and Zytnicki, M. 2005. Existential arc consistency : getting closer to full arc consistency in weighted CSPs. In Proc. of IJCAI, 84–89.

[Otten and Dechter2017] Otten, L., and Dechter, R. 2017. And/or branch-and-bound on a computational grid. JAIR 59 :351–435.

[Ouali et al.2017] Ouali, A.; Allouche, D.; de Givry, S.; Loudni, S.; Lebbah, Y.; and Loukil, L. 2017. Iterative decomposition guided variable neighborhood search for graphical model energy minimization. In Proc. of UAI, 550–559.

[Ralphs et al.2018] Ralphs, T.; Shinano, Y.; Berthold, T.; and Koch, T. 2018. Handbook of Parallel Constraint Reasoning. Springer. chapter Parallel Solvers for Mixed Integer Linear Optimization, 283–336.