Radio Link Frequency Assignment Revisited

Simon de Givry[1*] and Thomas Schiex[1*]

[1*]MIAT, Univ. Toulouse, ANITI, INRAE UR875, Toulouse, France.

*Corresponding author(s). E-mail(s): simon.de-givry@inrae.fr; thomas.schiex@inrae.fr;

Abstract

In a very early form of "data paper", we initially reported the precise formulation of the discrete combinatorial optimization frequency assignment problem instances that had been built by the CELAR (French Centre d'ÉLectronique de l'ARmement). These instances have since been used for benchmarking by many research groups around the world and have created a strong motivation for developing algorithms for solving the most challenging "MAX" instances, all examples of the famous Weighted Constraint Satisfaction Problem. As time passed, these challenging instances were tackled by increasingly efficient exact algorithms and this paper reports the results known currently by the authors. The high number of citations of this initially modest contribution show how important the accumulation of real challenging discrete optimization instances can be.

Keywords: Cost Function Network, Frequency Assignment, Non-serial Dynamic Programming, Tree Decomposition

1 Introduction

This paper revisits the original paper presenting the RLFAP benchmark [1] and its impact on the field of Cost Function Networks (CFN). The CFN framework extends the notion of constraint networks used in Constraint Satisfaction Problems (CSPs) towards optimization. In this context, constraints, or Boolean functions, are replaced by cost functions. A cost function returns a non-negative integer cost to every assignment of the variables in its scope.[1] In this setting, a constraint is simply a cost function returning zero if it is satisfied and ∞ if not. Given a CFN, the Weighted Constraint

[1]Negative or real costs can easily be transformed into non-negative integers using initial affine (shift and multiply) operations by which CFN solving is equivariant.

Satisfaction Problem (WCSP [2]) is to find a complete assignment with a minimum sum of the costs returned by all the cost functions in the solved instance.

As a reminder, the original RLFAP instances proposed by the CELAR where designed from a real frequency assignment instance, in the context of an EUCLID European project. They have different forms, including pure feasibility (FEAS), minimum span (SPAN), minimum cardinality (CARD), and maximum feasibility (MAX) variants. Additionally, a random GRAPH generator that uses the same frequency domains, and similar radio link and interference densities, was designed and used to produce additional instances following the original project [1].

Among all the variants, the MAX problem instances, which can be directly represented as pairwise CFNs, were the most challenging instances for 1999 solving technology and all remained open at the end of the project. The following section recapitulates the progress in closing these MAX instances as computing and algorithmic technologies evolved.

2 Results on Maximum Feasibility (MAX)

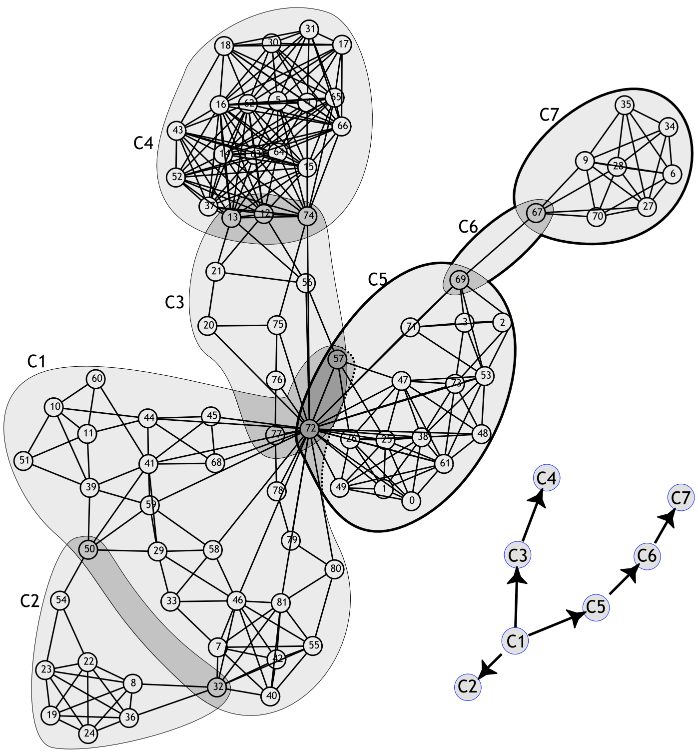

Fig. 1 The constraint graph of pre-processed RLFAP **scen06** problem covered by clusters and the associated tree decomposition using C1 as root.

Concerning the solving methods for the MAX criterion, defining pure pairwise WCSPs, Table 1 reports the progress made during almost three decades. Exploiting the problem structure as captured by a tree decomposition (see e.g., Fig.1) to perform some form of non-serial dynamic programming [3], possibly combined with branch-and-bound search and soft arc consistency filtering [4] has been instrumental in solving these instances [4–9].

It is tempting to analyze which of the algorithmic and computing hardware progresses have propelled these speedups and we use instance **scen06** as a basis for this. To roughly take into account the progress of computing hardware, we use existing analyses of the evolution of single-threaded integer performance, as measured from 1995 to 2011 by the SPECInt measure and reported in [10] and Wikipedia [11]. These two analyses approximate the growth in SPECint as the succession of a yearly 52% improvement until 2003, where the increase in frequencies drops and leads to a 22% yearly increase. Clearly, this rate further reduced, as in 2018, in their ACM award talk, John Hennessy and David Patterson [12] reported a yearly single-core performance improvement of only 3% the previous year.[2] For simplicity, we ignore this slow-down in our subsequent analysis, remembering that the "Fermi estimate" we will eventually use is likely favorable to hardware.

In 1996, the Sun UltraSparc (167MHz) was used to solve a small subproblem of 22 variables (known as the CELAR6-SUB4 instance) that had been identified by thorough decomposition of the problem in tightly connected subproblems. The optimum of this instance could then be added to another easy-to-solve subproblem (CELAR6-SUB0) eventually reaching a global lower bound equal to the best known upper bound of **scen06** [13]. The last single-thread result on the same instance using the HBFS algorithm in Table 1 relies on an Intel i7 2.1 GHz CPU. From 1996 to 2003 included, CPUs sped up by roughly $1.52^8 = 28.5$. Then, from 2004 to 2015, they further progressed by a $1.22^{12} = 10.9$ factor, for a total whooping and optimistic speedup of 310, a bit more than 2 orders of magnitude. In the same period, the solving time shifted from 26 days to one minute, with a resulting speedup of ≈ 37500. On this relatively small problem, algorithmic progress may thus be providing a speedup of $\frac{37500}{310} \approx 121$, a bit more than 2 orders of magnitude again. The recent **scen06** resolution with HBFS is directly obtained on the full instance while the early 26 days Russian Doll Search resolution followed a largely manual reduction in size of the problem and the lucky discovery that its optimum could be easily combined to build a proof of optimality. Overall, we estimate that, on this small instance, the hardware and algorithmic progresses roughly contributed equally to the observed speedup. On larger instances, increasing algorithmic speedups are expected.

The essence of these algorithmic progresses have been capitalized in the Cost Function Network open-source C++ solver TOULBAR2[3]. Notably, all the CELAR instances reported in [1] have now been closed (with the proofs of optimality of scen07 [7] and scen08 [8]). It is worth noting that the GRAPH instances are somewhat different and specifically require strong lower bounds, such as Virtual Arc Consistency (VAC) [14],

[2] We would like to thank our colleague G. Katsirelos for pointing out these references.
[3] https://github.com/toulbar2/toulbar2

to be solved to optimality. Table 2 reports the CPU-time for finding an optimal solution and proving optimality using the Hybrid Best-First Search algorithm [9] with VAC and Limited Discrepancy Search [15] applied in preprocessing.[4]

For the most difficult CELAR and GRAPH instances, variable neighborhood search exploiting the problem structure [16], and especially its parallel version [16, 17], is able to find optimal solutions in a few seconds.

Table 1 Progress in solving CELAR instances.

CPU-time	Method	Reference
scen06 (200 variables with domain size of 44, optimum of 3,389)		
26 days[1] (SUN UltraSparc 167 MHz)	Russian Doll Search (RDS)	[5]
3 days[1] (SUN Sparc 2)	PFC-MRDAC	[18]
8 hours[2] (DEC Alpha 500MP)	Non-serial DP (NSDP)	[6]
3 hours[2] (Intel Xeon 2.4 GHz)	Backtrack bounded by Tree Decomposition (BTD)	[4]
3.7 minutes[2] (Intel Xeon 2.6 GHz)	BTD with dichotomic branching	[7]
1 minute[3] (Intel i7 2.1 GHz)	Hybrid Best-First Search (HBFS)	[9]
9 seconds[4] (Xeon 2.9 GHz 96-core)	Parallel HBFS	[19]
scen07 (400 variables with domain size of 44, optimum of 343,592)		
4.5 days[2] (Intel Xeon 2.6 GHz)	RDS-BTD with dichotomic branching	[7]
1.25 days[2] (Cluster of Xeon 2.5 GHz)	Parallel NSDP	[8]
scen08 (916 variables with domain size of 44, optimum of 262)		
127.5 days[2] (Cluster of Xeon 2.5 GHz)	Parallel NSDP	[8]
scen09 (680 variables with domain size of 44, optimum of 15,571)		
0.25 second[3] (Intel i7 2.1 GHz)	HBFS	[9]
scen10 (680 variables with domain size of 44, optimum of 31,516)		
0.16 second[3] (Intel i7 2.1 GHz)	HBFS	[9]

[1]Time for optimality proof obtained on a subproblem with 22 variables (CELAR6-SUB4) resulting from an ad-hoc decomposition of the original problem.

[2]Time for optimality proof on the original problem.

[3]Time for finding the optimum and proving optimality on the original problem using TOULBAR2 v1.2.1 with option *-l=4* (Limited Discrepancy Search [15] in preprocessing).

[4]Wall-clock time for optimality proof on the original problem.

3 Conclusion

Since this RLFAP benchmark [1] and another one on the SPOT5 Earth Observation Satellite Management [20], more than 20,000 instances grouped in 67 benchmarks have been collected in the Cost Function Network library[5]. Among them, the ROADEF

[4]TOULBAR2 with options *-l=4 -A -C=100* for VAC_ϵ with $\epsilon = \frac{1}{100}$
[5]https://forge.inrae.fr/thomas.schiex/cost-function-library

Table 2 Results on GRAPH instances obtained on an Intel i7 2.1 GHz.

Instance	Number of var.	Optimum	CPU-time (seconds)
GRAPH05	200	221	0.25 s
GRAPH06	400	4,123	0.56 s
GRAPH07	400	4,324	0.28 s
GRAPH11	680	3,080	53.3 s
GRAPH12	680	11,827	0.55 s
GRAPH13	916	10,110	48.8 s

Challenge 2001 is a more complex version of RLFAP adding frequency polarity. CFN methods work very well here, being competitive with the Challenge results.[6][7]

Author contributions. The two authors contributed equally to this work.

Funding. The two authors benefitted from the AI Interdisciplinary Institute ANITI. ANITI is funded by the French "Investing for the Future – PIA3" program under the Grant agreement n°ANR-23-IACL-0002.

Data availability. The RLFAP data files are available in the Cost Function Network Library at https://forge.inrae.fr/thomas.schiex/cost-function-library.

Code availability. Toulbar2 code is available at https://github.com/toulbar2/toulbar2, under the MIT licence.

Material availability. Not applicable.

Declarations

Competing interest. Thomas Schiex serves as scientific advisor and shareholder of the protein design startup Amineo that exploits Toulbar2 for its business. These roles have not influenced the design, analysis, or interpretation of the work presented here. All authors declare that there are no other competing interests.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

References

[1] Cabon, B., Givry, S., Lobjois, L., Schiex, T., Warners, J.P.: Radio Link Frequency Assignment. Constraints **4**(1), 79–89 (1999)

[2] Schiex, T., Fargier, H., Verfaillie, G.: Valued constraint satisfaction problems: hard and easy problems. In: Proc. of IJCAI-95, Montréal, Canada, pp. 631–637 (1995)

[6] https://github.com/toulbar2/toulbar2/blob/master/web/TUTORIALS/tutorialCP2020.md
[7] CFN techniques have been successfully applied to linear and quadratic assignment problems [21, 22].

[3] Bertelé, U., Brioshi, F.: Nonserial Dynamic Programming. Academic Press, London (1972)

[4] Givry, S., Schiex, T., Verfaillie, G.: Exploiting Tree Decomposition and Soft Local Consistency in Weighted CSP. In: Proc. of AAAI-06, Boston, MA (2006)

[5] Givry, S., Verfaillie, G., Schiex, T.: Bounding the Optimum of Constraint Optimization Problems. In: Proc. of CP-97, Schloss Hagenberg, Austria, pp. 405–419 (1997)

[6] Koster, A.M., Hoesel, S.P., Kolen, A.W.: Optimal solutions for frequency assignment problems via tree decomposition. In: International Workshop on Graph-Theoretic Concepts in Computer Science, pp. 338–350 (1999). Springer

[7] Sanchez, M., Allouche, D., Givry, S., Schiex, T.: Russian doll search with tree decomposition. In: Proc. of IJCAI-09, Pasadena (CA), USA (2009)

[8] Allouche, D., Givry, S., Schiex, T.: Towards parallel non serial dynamic programming for solving hard weighted csp. In: Proc. of CP-10, St Andrews, Scotland (2010)

[9] Allouche, D., Givry, S., Katsirelos, G., Schiex, T., Zytnicki, M.: Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP. In: Proc. of CP-15, Cork, Ireland, pp. 12–28 (2015)

[10] Preshing, J.: A Look Back at Single-Threaded CPU Performance. https://preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/. Accessed: 2026-04-21 (2012)

[11] Hennessy, J.L., Patterson, D.A.: Growth in processor performance, 1978–2010. https://commons.wikimedia.org/wiki/File:Growth_in_processor_performance,_1978-2010.png. Accessed: 2026-04-21 (2011)

[12] Hennessy, J.L., Patterson, D.A.: A New Golden Age for Computer Architecture. https://www.sigarch.org/john-hennessy-and-david-patterson-share-acm-turing-award/. Accessed: 2026-04-21 (2018)

[13] De Givry, S., Verfaillie, G., Schiex, T.: Bounding the optimum of constraint optimization problems. In: International Conference on Principles and Practice of Constraint Programming, pp. 405–419 (1997). Springer

[14] Cooper, M., Givry, S., Sanchez, M., Schiex, T., Zytnicki, M., Werner, T.: Soft arc consistency revisited. Artificial Intelligence **174**(7–8), 449–478 (2010)

[15] Harvey, W.D., Ginsberg, M.L.: Limited discrepancy search. In: Proc. of IJCAI-95, Montréal, Canada (1995)

[16] Ouali, A., Allouche, D., Givry, S., Loudni, S., Lebbah, Y., Loukil, L., Boizumault, P.: Variable neighborhood search for graphical model energy minimization. Artificial Intelligence **278**(103194), 22 (2020)

[17] Ouali, A., Loudni, S., Loukil, L., Boizumault, P., Lebbah, Y.: Replicated parallel strategies for decomposition guided vns. Electronic Notes in Discrete Mathematics **47**, 93–100 (2015)

[18] Larrosa, J., Meseguer, P., Schiex, T., Verfaillie, G.: Reversible dac and other improvements for solving max-CSP. In: Proceedings of the 15^{th} AAAI Conference on Artificial Intelligence, pp. 347–352 (1998)

[19] Beldjilali, A., Montalbano, P., Allouche, D., Katsirelos, G., Givry, S.: Parallel Hybrid Best-First Search. In: Solnon, C. (ed.) Proc. of CP-22, vol. 235. Haifa, Israel, pp. 7–1710 (2022)

[20] Bensana, E., Lemaitre, M., Verfaillie, G.: Earth observation satellite management. Constraints **4**(3), 293–299 (1999)

[21] Sewa, G., Allouche, D., Givry, S., Katsirelos, G., Montalbano, P., Schiex, T.: Assignment problems in cost function networks. In: Proc. of AAAI-26, Singapore (2026)

[22] Sewa, G., Allouche, D., Givry, S., Katsirelos, G., Schiex, T.: Singleton node consistency for quadratic assignment problems in cost function networks. In: Proceedings of the 23rd International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Rabat, Morocco (2026)