Pesquisa Operacional (2020) 40: e236329 p.1-11
doi: 10.1590/0101-7438.2020.040.00236329

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

# THE GUIDE TO NP-COMPLETENESS IS 40 YEARS OLD: AN HOMAGE TO DAVID S. JOHNSON

Luciana S. Buriol[1*], Celina Figueiredo[2],
Mauricio G. C. Resende[3] and Eduardo Uchoa[4]

Received April 8, 2020 / Accepted July 31, 2020

**ABSTRACT.** Computers and Intractability: A Guide to the Theory of NP-Completeness, by Michael R. Garey and David S. Johnson, was published 40 years ago (1979). Despite its age, it is unanimously considered by many in the computational complexity community as its most important book. NP-completeness is perhaps the single most important concept to come out of theoretical computer science. The book was written in the late 1970s, when problems solvable in polynomial time were linked to the concepts of efficiently solvable and tractability, and the complexity class NP was defined to capture the concept of good characterization. Besides his contributions to the theory of NP-completeness, David S. Johnson also made important contributions to approximation algorithms and the experimental analysis of algorithms. This paper summarizes many of Johnson's contributions to these areas and is an homage to his memory.

**Keywords**: NP-completeness; approximation algorithms; experimental analysis of algorithms.

## 1 JOHNSON'S CONTRIBUTIONS TO NP-COMPLETENESS

Cook (1971) and Levin (1973) in the early 1970s proved the seminal theorem for the theory of NP-completeness establishing that Satisfiability is NP-complete. The subsequent seminal paper by Karp (1972) establishing 21 NP-complete problems was the first to use the terms P and NP, although *polynomial complete* was the name used at the time for the hard problems.

To mark the 40th anniversary in 2012 of the publication of the Karp's paper, where the wide applicability of the NP-completeness concept was established, David S. Johnson wrote the paper *A Brief History of NP-Completeness* (Johnson, 2012). The year 2012 also marked the 100th

*Corresponding author

[1] Universidade Federal do Rio Grande do Sul, Caixa Postal 15064, 91501-970, Porto Alegre, RS, Brasil – E-mail: buriol@inf.ufrgs.br – https://orcid.org/0000-0002-9598-5732

[2] Programa de Engenharia de Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 68511, 21941-972 Rio de Janeiro, RJ, Brasil – E-mail: celina@cos.ufrj.br – https://orcid.org/0000-0002-6393-0876

[3] University of Washington, Industrial and Systems Engineering, Box 352650, Seattle, WA, USA – E-mail: mgcr@uw.edu – https://orcid.org/0000-0001-7462-6207

[4] Universidade Federal Fluminense, Niterói, RJ, Brasil – E-mail: uchoa@producao.uff.br – https://orcid.org/0000-0002-8687-2613

birthday of Alan Turing, whose Turing machine is the basic model for computation used to define the classes P and NP (Haeusler, 2012).

The early involvement of David S. Johnson with NP-completeness mainly concerned the methods for coping with hard problems, by designing and analyzing approximation algorithms. David S. Johnson met Michael R. Garey at Bell Laboratories in Murray Hill, New Jersey, and one of their first collaborations was to answer a letter by Donald Knuth seeking a better name for *polynomial complete*. Garey and Johnson proposed the term *NP-complete*. For a detailed account by Knuth about the choice of the name NP-complete, we refer to Knuth (1974).

The popularity of the NP-completeness concept and of its guidebook increased when the P versus NP problem was selected by the Clay Mathematics Institute as one of the seven Millennium Problems to motivate research on important classic questions that have resisted solution over the years. The P versus NP problem is considered a central problem in theoretical computer science, and aims to classify the possible existence of efficient solutions to combinatorial and optimization problems (Fortnow, 2009).

Besides complexity theory, Johnson made remarkable contributions to approximation algorithms, worst-case analysis of algorithms, and local search methods. Johnson was also an enthusiast of experimental algorithms. Over several decades, he made many contributions to this area, too. He wrote several papers, including a guide with ten principles for experimental analysis of algorithms (Johnson, 2002). He also inspired the creation of and organized several DIMACS Implementation Challenges that brought together researchers from all over the world to investigate the best algorithms to solve different variants of the problems focused on in each challenge. Johnson organized challenges on network flows and matching (Johnson & McGeoch, 1993), maximum clique, graph coloring, and satisfiability (Johnson & Trick, 1996), priority queues and dictionaries (Goldwasser et al., 2002) and near neighbor searches (Goldwasser et al., 2002), semidefinite and related optimization problems (Johnson et al., 2000), traveling salesman problem (Johnson et al., 2001), shortest path problem (Demetrescu et al., 2009), and Steiner tree problem (Johnson et al., 2014).

This paper aims to summarize Johnson's contributions to NP-completeness and experimental analysis of algorithms. Moreover, Johnson had many other qualities, and was a very kind and sensible person. The last section of this paper shares a personal view of David S. Johnson.

## 1.1 A Brazilian perspective on the 40th anniversary

### Complexity-separating graph classes

In the sixteenth edition of his *The NP-completeness Column: An Ongoing Guide* (Johnson, 1985), Johnson focused on graph restrictions and their effect, with emphasis on the restrictions to graph classes and how they affect the complexity of various NP-hard problems. Graph classes were selected because of their broad algorithmic significance. The presentation consisted of a summary table with 30 rows containing the selected classes of graphs and 11 columns. The first column was devoted to the complexity of determining whether a given graph is in the specified class followed by ten of the most famous NP-complete graph problems. The entry for a class and a problem was the complexity of the problem restricted to that class of graphs — polynomial-time solvable or NP-complete, if known. The goal was to identify interesting problems and interesting graph classes establishing the concept of complexity separation.

The chosen ten famous graph problems were: independent set, clique, partition into cliques, chromatic number, chromatic index, Hamiltonian circuit, dominating set, simple (unweighted) max cut, (unweighted) Steiner tree in graphs, and graph isomorphism. The first nine problems were at the time known to be NP-complete for general graphs; the complexity of graph isomorphism for general graphs is still a long-standing open problem, one of twelve open problems highlighted at the end of the NP-completeness guide by (Garey & Johnson, 1979). The table revealed among its 330 entries the existence of a substantial collection of 71 open problems classified from *entertaining puzzles* as $P$? or $O$? to *may well be hard* or *are famous* as $O$ or $O$! problems. It is remarkable that only one entry in the entire table deserved a famous open problem $O$! entry, the recognition for perfect graphs, and just two entries deserved a may well be hard problem $O$ entry, edge COLORING of planar graphs and Hamiltonian circuit of permutation graphs. The two problems with most open entries were edge coloring and maxcut, and there are today just a few updates with respect to those two problems. At that time, the edge coloring problem had 19 of its 30 entries classified as $O$?, which meant *apparently open but possibly easy to resolve*, and 14 of those $O$? entries remain open today.

The choice of the ten famous graph problems and of the 30 significant graph classes reflected the importance of the famous open problem, the recognition for perfect graphs, for which the special $O$! entry was given. A graph is perfect if for every induced subgraph the chromatic number equals the maximum clique size. In the first edition of his *The NP-Completeness Column: An Ongoing Guide* (Johnson, 1981), Johnson discussed the progress that had been made on the twelve open problems presented at the end of the NP-completeness guide (Garey & Johnson, 1979). Six of those open problems had been resolved, and the split was even: three had been shown to be solvable in polynomial time and three had been proved NP-complete. It is remarkable that today we know that ten of those twelve open problems are resolved, and that the split is still even. Johnson concluded the first edition of the NP-completeness column by presenting as the problem of the month, the recognition for perfect graphs, and explained that just in 1981 the class of imperfect graphs was shown to be in NP, equivalently the class of perfect graphs was shown to be in co-NP.

Only one entry among the 330 entries in the entire table was due to a Brazilian author, Jayme Luiz Szwarcfiter, who established in 1982 the NP-completeness of Hamiltonian circuit for grids (Itai et al., 1982). Today, there are two additional entries that are resolved by Brazilian authors. graph isomorphism restricted to proper circular arc graphs admits a linear time algorithm (Lin et al., 2008), whereas max cut restricted to strongly chordal graphs is NP-complete (Sucupira et al., 2013).

## 2 TOWARDS EXPERIMENTAL ANALYSIS OF ALGORITHMS

Johnson started his career as a “pure theoretician.” Before his contributions to theory of NP-completeness, he worked with approximation algorithms, polynomial algorithms with provable guarantees on the distance of the returned solution to the optimum. An example of this work was his PhD thesis (Johnson, 1973), titled *Near-Optimal Bin Packing Algorithms*, which was defended in 1973 at the MIT Mathematics Department, advised by Michael J. Fischer. The main result of the thesis was a proof that the First Fit Decreasing heuristic for the bin packing problem never returns a solution that uses more than $(11\ \mathrm{OPT}/9)+4$ bins, where $\mathrm{OPT}$ is the optimal number of bins. Johnson also proposed approximation algorithms for other optimization problems, such as graph coloring (Garey & Johnson, 1976) and some scheduling problems (Garey et al., 1978). The NP-completeness guide (Garey & Johnson, 1979) has a chapter on “Coping with NP-Complete Problems” in practice. Although heuristics are briefly mentioned, most of the chapter describes approximation algorithms.

In the late 1970s and early 1980s, access to computers became more widespread and practitioners could apply their algorithms to tackle real problems. It became clear that, despite their theoretical importance, approximation algorithms were not the most practical way of handling typical NP-complete problems. The following issues were raised:

1. Few NP-complete problems were found to be like the bin packing, having approximation algorithms with really tight guarantees of quality. In most cases the obtainable approximation factors were larger. For example, the best known approximation factor for the metric traveling salesman problem (TSP) is 1.5 (Christofides, 1976), the best known approximation factor for the vertex cover problem is 2 (the approximation bounds for those two problems could not be improved since then). In fact, there are several NP-complete problems, like the TSP with general costs, that were proved to not have constant factor approximations unless $P = NP$.

2. The approximation factors are worst-case guarantees. For most instances the solutions obtained by approximation algorithms were significantly closer to the optimal solutions. However, it was realized that heuristics based on techniques like local search almost always obtained even better solutions.

Around the same time, a very prominent case highlighted another limitation of the classic theoretical study of algorithms, based in worst-case asymptotic analysis. The highly popular simplex algorithm for linear programming (Dantzig, 1963), proposed in 1947 by George B. Dantzig, has a very good practical performance. Yet, (Klee & Minty, 1972) proved that a variant of the simplex algorithm, as formulated by Dantzig, can take exponential time in some instances. (Khachiyan, 1979) created much excitement in the mathematical world by discovering the first polynomial algorithm for linear programming. Part of that excitement subsided in the following years, as practitioners realized that his Ellipsoid Algorithm performed very poorly in practice. However, the theoreticians were vindicated to some extent when the polynomial interior-point algorithms for linear programming, the first algorithm of that family was proposed in (Karmarkar, 1984), were shown to have a good practical performance (Adler et al., 1989), being better than the simplex algorithm in some cases.

It seems clear that Johnson was influenced by that zeitgeist and wished to possess more accurate tools for assessing the practical performance of algorithms. His first incursion in the subject was still theoretical: an analytical probabilistic study of the asymptotic expected behavior of First Fit and First Fit Decreasing algorithms for bin packing, assuming that bin size is 1 and item weights are chosen uniformly from the interval $(0,u], u \leq 1$ (Bentley et al., 1984). The study reached quite interesting conclusions, indicating that both algorithms were likely to obtain solutions very close to the optimal. Nevertheless, Johnson would later recognize two limitations of probabilistic analysis: i) it can only be applied to relatively simple algorithms and ii) it should assume that instance data follow certain probability distributions, which can be simply unrealistic.

A second incursion was also theoretical. Given that many of the most successful heuristics for NP-complete problems were based on local search, the studies in (Johnson et al., 1985) and (Johnson et al., 1988) tried to assess the computational complexity of reaching a local optimal solution with respect to a given neighborhood. For example, the most classical neighborhoods for the TSP are 2-OPT, 3-OPT and Lin-Kernighan. For all of these neighborhoods it is not known whether it is possible to find a local optimal in polynomial time. (Johnson et al., 1985) define the class *PLS* (Polynomial Local Search) as being composed of the neighborhoods where it is possible to check local optimality in polynomial time. All the three previously mentioned TSP neighborhoods are clearly in *PLS*, as are most neighborhoods used in practice. They then proved that the Kernigham-Lin neighborhood for the graph partitioning problem is *PLS*-complete, meaning that if it is possible to find a local optima for that neighborhood in polynomial time then it is also possible to find local optima for all neighborhoods in *PLS* in polynomial time. While that result represented a technical feat, the study somehow failed in its ultimate goal of providing a fruitful theoretical framework (akin to the theory of NP-completeness) for classifying neighborhoods as easy or hard: (1) it is very difficult to determine whether a given neighborhood is *PLS*-complete, in particular, the authors could not determine the status of the classic TSP neighborhoods: (2) in practice, it is usually very easy to find local optimal solutions, even for *PLS*-complete neighborhoods. Actually, the real issue for heuristics based on local search is not finding any local optimal solution, it is escaping from the attraction of local optima that are not near-optimal global solutions.

The late 1980s witnessed the flourishing of metaheuristics, techniques that are aimed at guiding local searches in their exploration of the solution space. Classic metaheuristics like genetic algorithms (Holland, 1975), simulated annealing (Kirkpatrick et al., 1983), tabu search (Glover, 1986), and GRASP (Feo & Resende, 1989) started to be broadly applied to many optimization problems. At that moment, Johnson convinced himself that: (1) heuristics are indeed among the best known ways of coping with NP-complete problems in practice; (2) since the known theoretical tools have a limited capacity for evaluating algorithms (specially heuristics), extensive computational experiments would be necessary. The paper that marks his debut in the area of

Experimental Analysis of Algorithms (EAA) is an in-depth study of simulated annealing applied to graph partitioning (Johnson et al., 1989).

Johnson joined EAA with enthusiasm. Nevertheless, he was not pleased with some of the work in the area, for what he perceived as lacking scientific rigor. Johnson started then a lifelong struggle for promoting EAA and for raising its standards.

### 2.1 DIMACS implementation challenges

In Johnson's own words: "*The DIMACS Implementation Challenges address questions of determining realistic algorithm performance where worst case analysis is overly pessimistic and probabilistic models are too unrealistic: experimentation can provide guides to realistic algorithm performance where* (theoretical) *analysis fails*." Johnson conceived the DIMACS Implementation Challenges and was directly involved with the organization of its first 11 editions.

In each challenge, a problem or set of problems are defined, a large and diverse set of instances collected, and algorithms are tested under the same conditions. The objective of each challenge is to establish the state-of-the-art solution methods for the problem(s). Below is a list of the 11 DIMACS Implementation Challenges and the year each one was run:

1. 1991: Network Flows and Matching
2. 1993: Maximum Clique, Graph Coloring, and Satisfability
3. 1994: Effective Parallel Algorithms for Combinatorial Problems
4. 1995: Fragment Assembly and Genome Rearrangements
5. 1996: Priority Queues, Dictionaries, and Multi-Dimensional Point Sets
6. 1999: Near Neighbor Searches
7. 2000: Semidefinite and Related Optimization Problems
8. 2001: The Traveling Salesman Problem
9. 2006: The Shortest Path Problem
10. 2012: Graph Partitioning and Graph Clustering
11. 2014: Steiner Tree Problems

A DIMACS Implementation Challenge usually has a lasting effect on the research of its target problems, establishing rigorous standards for the evaluation of future algorithms.

The 12th DIMACS Implementation Challenge on *Vehicle Routing Problems* will be the first without Johnson's participation. It is scheduled to be held in 2021, details can be found at dimacs.rutgers.edu/events/details?eID=1090.

### 2.2 A theoretician's guide to the experimental analysis of algorithms

After the Challenge on TSP, maybe the problem from the DIMACS Implementation Challenges most dear to Johnson, he wrote two chapters, *Experimental Analysis of Heuristics for the STSP* and *Experimental Analysis of Heuristics for the ATSP* in a TSP book (Gutin & Punnen, 2002). Those chapters are exemplary works on EAA.

Soon after writing these two book chapters Johnson published *A Theoretician's Guide to the Experimental Analysis of Algorithms*, a summary of 15 years of reflections on how EAA should be performed and how results should be reported (Johnson, 2002). In our view, the guide is still a mandatory reading for those who work in the area. It starts by motivating EAA, observing that "*theoretical results cannot tell the fully story about real-world algorithmic performance*." The guide has a section for each of the ten principles recommended to be observed by researchers before writing experimental papers:

1. Perform newsworthy experiments.
2. Tie your paper to the literature.
3. Use instance testbeds that can support general conclusions.
4. Use efficient and effective experimental designs.
5. Use reasonably efficient implementations.
6. Ensure reproducibility.
7. Ensure comparability.
8. Report the full story.
9. Draw well-justified conclusions and look for explanations.
10. Present your data in informative ways.

During the course of the discussions on the principles, Johnson pointed out pitfalls that should be avoided and also what he called "pet peeves" (flaws that he found particularly annoying).

## 3 THE PERSON DAVID S. JOHNSON

Johnson's contributions to science were publicly recognized in diverse occasions. In 1995 he became an ACM Fellow for his fundamental contributions to the theories of approximation algorithms and computational complexity, and for outstanding service to ACM. In 1997 he received the inaugural SIGACT Distinguished Service Prize. In 2010 he received the Knuth Prize for outstanding contributions to the foundations of computer science. In 2016 he was elected to the National Academy of Engineering for his contributions to the theory and practice of optimization and approximation algorithms.

Johnson was a perfectionist when it came to writing. We would spend a great deal of time polishing and crafting his writings. His reviews were always done with care and detail. For example, when reviewing "proofs" of "P = NP" Johnson would not simply discard the paper but rather would try to show the author gaps in their "proof".

Johnson was strongly connected with his work. He insisted on using his middle initial ("S" for "Stifler") to be uniquely identifiable. After completing his PhD at MIT, Johnson was hired to work at Bell Laboratories (later AT&T Labs) in New Jersey, where he worked from 1973 until his retirement in 2013. He was head of the Mathematical Foundations of Computing Department at Bell Labs and of the Algorithms and Optimization Research Department at AT&T Labs Research from 1988 to 2013. When Johnson drove to work, he would always park in the same spot. In the summers he would often bike to work. At noon every day, Johnson would go from door to door down his hallway inviting people with a friendly "Lunch?". Even when on vacation in New Jersey Johnson would often come to have lunch at work with his colleagues. At least during his last 25 years at AT&T, Johnson would always eat the same meal at lunch, a salad with dressing on the side and a coke. Every day at 4 PM Johnson would always have a second coke. Johnson and his wife Dorothy Wilson organized an annual picnic at their home in Madison, New Jersey, hosting current and former colleagues of Johnson's, summer interns and visitors, as well as their family members. Over the many picnics, his friends and colleagues saw Jack Johnson, son of Johnson and Dorothy, grow up.

David, as friends and colleagues used to address him, also enjoyed many activities outside of work. He used to run, including marathons. He was a Green Bay Packers fan. He had a collection of over 5,000 CDs and many DVDs and Blu-rays. Johnson was an avid reader of science fiction. He had a complete collection of Mad Magazine. Many of the items David collected over the years are now at the Library of Drew University in Madison, New Jersey (Drew University, 2020).

David is and will always be deeply missed.

## References

[1] ADLER I, RESENDE MG, VEIGA GG & KARMARKAR NK. 1989. An implementation of Karmarkar's algorithm for linear programming. *Mathematical programming*, **44**(1-3): 297–335.

[2] BENTLEY JL, JOHNSON DS, LEIGHTON FT, MCGEOCH CC & MCGEOCH LA. 1984. Some unexpected expected behavior results for bin packing. In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. pp. 279–288. ACM.

[3] CHRISTOFIDES N. 1976. Worst-case analysis of a new heuristic for the travelling salesman problem. Tech. rep.. Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group.

[4] COOK S. 1971. The complexity of theorem proving procedures. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. pp. 151–158.

[5] DANTZIG GB. 1963. *Linear programming and extensions*. Princeton U. Press and the Rand Corp.. Available at: https://www.rand.org/pubs/reports/R366.html.

[6] DEMETRESCU C, GOLDBERG AV & JOHNSON DS (Eds.). 2009. *The shortest path problem: Ninth DIMACS implementation challenge*. vol. 74 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. Providence, Rhode Island: American Mathematical Society.

[7] DREW UNIVERSITY. 2020. David Johnson Collection of Science Fiction and Popular Culture. D. lrew University Library Special Collections, accessed September 7, 2020.

[8] FEO TA & RESENDE MG. 1989. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, **8**(2): 67–71.

[9] FORTNOW L. 2009. The status of the P versus NP problem. *Commun. ACM*, **52**: 78–86.

[10] GAREY MR, GRAHAM RL & JOHNSON DS. 1978. Performance Guarantees for Scheduling Algorithms. *Operations Research*, **26**: 3–21.

[11] GAREY MR & JOHNSON DS. 1976. The Complexity of Near Optimal Graph Coloring. *J. Assoc. Comp. Mach.*, **23**: 43–49.

[12] GAREY MR & JOHNSON DS. 1979. *Computers and Intractability, A Guide to the Theory of NP-completeness*. WH Freeman.

[13] GLOVER F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, **13**(5): 533–549.

[14] GOLDWASSER MH, JOHNSON DS & MCGEOCH CC (Eds.). 2002. *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*. vol. 59. Providence, Rhode Island: American Mathematical Society.

[15] GUTIN G & PUNNEN A. 2002. *The Traveling Salesman Problem and Its Variations*. Springer.

[16] HAEUSLER EH. 2012. A celebration of Alan Turing's achievements in the year of his centenary. *Int. T. Oper. Res.*, **19**: 487–491.

[17] HOLLAND JH. 1975. Adaptation in natural and artificial systems. *Ann Arbor: University of Michigan Press*, .

[18] ITAI A, PAPADIMITRIOU CH & SZWARCFITER JL. 1982. Hamilton Paths in Grid Graphs. *SIAM J. Comput.*, **11**: 676–686.

[19] JOHNSON DS. 1973. *Near-optimal bin packing algorithms*. Ph.D. thesis. Massachusetts Institute of Technology. Cambridge, Massachusetts. Available at: https://dspace.mit.edu/handle/1721.1/57819.

[20] JOHNSON DS. 1981. The Twelve Open Problems From G&J: Updates. *J. Algorithms*, **2**: 393–405.

[21] JOHNSON DS. 1985. Graph restrictions and their effect. *J. Algorithms*, **6**: 434–451.

[22] JOHNSON DS. 2002. A Theoretician's Guide to the Experimental Analysis of Algorithms. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **59**.

[23] JOHNSON DS. 2012. A Brief History of NP-Completeness, 1954–2012. In: *Extra Volume: Optimization Stories*. pp. 359–376. Documenta Math.

[24] JOHNSON DS, ARAGON CR, MCGEOCH LA & SCHEVON C. 1989. Optimization by simulated annealing: An experimental evaluation. Part I, Graph partitioning. *Operations Research*, **37**: 865–892.

[25] JOHNSON DS, KOCH T, WERNECK RF & ZACHARIASEN M. 2014. 11th DIMACS Implementation Challenge in Collaboration with ICERM: Steiner Tree Problems. Available at: http://dimacs11.zib.de/workshop.html.

[26] JOHNSON DS & MCGEOCH CC (Eds.). 1993. *Network flows and matching: First DIMACS implementation challenge*. vol. 12 of *DIMACS series in discrete mathematics and theoretical computer science*. Providence, Rhode Island: American Mathematical Society.

[27] JOHNSON DS, MCGEOCH L, GLOVER F & REGO C. 2001. 8th DIMACS implementation challenge: The traveling salesman problem. Available at: http://dimacs.rutgers.edu/archive/Challenges/TSP/.

[28] JOHNSON DS, PAPADIMITRIOU CH & YANNAKAKIS M. 1985. How easy is local search? In: *26th Annual Symposium on Foundations of Computer Science (SFCS 1985)*. pp. 39–42. IEEE.

[29] JOHNSON DS, PAPADIMITRIOU CH & YANNAKAKIS M. 1988. How easy is local search? *Journal of Computer and System Sciences*, **37**(1): 79–100.

[30] JOHNSON DS, PATAKI G & ALIZADEH F. 2000. 7th DIMACS implementation challenge: Semidefinite and Related Optimization Problems. Available at: http://archive.dimacs.rutgers.edu/Workshops/7thchallenge/.

[31] JOHNSON DS & TRICK MA (Eds.). 1996. *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*. vol. 26 of *DIMACS series in discrete mathematics and theoretical computer science*. Providence, Rhode Island: American Mathematical Society.

[32] KARMARKAR NK. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica*, **4**: 373–395.

[33] KARP RM. 1972. Reducibility Among Combinatorial Problems. In: MILLER RE & THATCHER JW (Eds.), *Complexity of Computer Computations*. pp. 85–103. New York: Plenum.

[34] KHACHIYAN LG. 1979. A polynomial algorithm in linear programming. In: *Doklady Academii Nauk SSSR*, vol. 244. pp. 1093–1096.

[35] KIRKPATRICK S, GELATT CD & VECCHI MP. 1983. Optimization by simulated annealing. *Science*, **220**(4598): 671–680.

[36] KLEE V & MINTY J. 1972. How good is the simplex algorithm? *Inequalities III*, pp. 159–175.

[37] KNUTH DE. 1974. A terminological proposal. *SIGACT News*, **6**: 12–18.

[38] LEVIN L. 1973. Universal search problems. *Problems of Information Transmission*, i. ln Russian.

[39] LIN MC, SOULIGNAC FJ & SZWARCFITER JL. 2008. A simple linear time algorithm for the isomorphism problem on proper circular-arc graphs. *in: Scandinavian Workshop on Algorithm Theory*, pp. 355–366.

[40] SUCUPIRA R, FARIA L & KLEIN S. 2013. A complexidade do problema do Corte Máximo para grafos fortemente cordais. *in: XLV Simpósio Brasileiro de Pesquisa Operacional*, pp. 2979–2988.

## How to cite

BURIOL L, FIGUEIREDO C, RESENDE MGC & UCHOA E. 2020. The guide to NP-completeness is 40 years old: An homage to David S. Johnson. *Pesquisa Operacional*, **40**: e236329. doi: 10.1590/0101-7438.2020.040.00236329.