SIAG/OPT Views and News

A Forum for the SIAM Activity Group on Optimization

Volume 27 Number 2 November 2019

Contents

Articles

Recent Results on the Traveling Salesman Problem

Jakub Tarnawski
Microsoft Research
Zurich, Switzerland
jakub.tarnawski@microsoft.com
http://jakub.tarnawski.org

Vera Traub
Research Institute for Discrete Mathematics and
Hausdorff Center for Mathematics
University of Bonn Germany
traub@or.uni-bonn.de

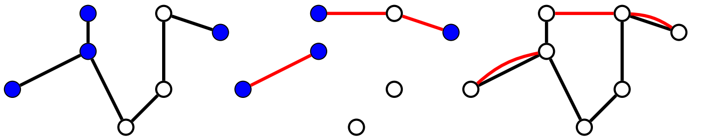
1 Introduction

Given a set of cities, what is the shortest tour visiting all of them? – This is the famous traveling salesman problem (TSP). It has played an important role in the development of combinatorial optimization, both in theoretical and computational work; see also [6] for an excellent introduction. In this article we will focus on approximation algorithms for TSP. The last decade has been an extremely exciting time for this area, and much progress has been made. In this article we would like to give a brief overview of the new developments and shortly describe two nice, general techniques that have been instrumental in some of these advances. There are two important variants of TSP: the symmetric TSP, where the distance from A to B is the same as the distance from B to A, and the asymmetric TSP, where we do not make this symmetry assumption. In terms of approximation algorithms these variants behave very differently. We will first discuss the symmetric TSP and afterwards the more general asymmetric TSP (ATSP).

In the sequel, we will use $\delta(v)$ to denote the set of edges incident on a vertex v.

2 Symmetric TSP

Formally, the symmetric TSP, often simply called TSP, is defined as follows. We are given a connected undirected graph G with vertex set V and edge set E. Moreover, we



(a) **(b)** **(c)**

Figure 1: Illustration of Christofides' algorithm. (a) a spanning tree S, where the set T_S of odd degree vertices is shown in blue. (b) a T_S-join J_S. (c) the disjoint union of the spanning tree S and the T_S-join J_S. This graph is connected and Eulerian, i.e. every vertex has even degree. Therefore, we can find a closed walk using every edge exactly once; this walk is the output of Christofides' algorithm.

have non-negative edge costs $c : E \to \mathbb{R}_{\geq 0}$. The task is to find a walk in the graph G that starts and ends at the same vertex, visits all other vertices in between, and has minimum cost.

Christofides' algorithm

Christofides' algorithm [5] from the 1970's yields a $\frac{3}{2}$-approximation for TSP; the same algorithm was also independently discovered by Serdyukov [20]. Despite decades of research, until today we still know no better approximation algorithm. Improving on Christofides' algorithm for TSP is one of the most important open problems in combinatorial optimization and approximation algorithms.

Christofides' algorithm works as follows. First, we compute a minimum-cost spanning tree (V, S), i.e. S is a minimum-cost set of edges such that (V, S) is connected. We are going to add edges to S such that all vertex degrees become even. For this we define the set $T_S := \{v : |\delta(v) \cap S| \text{ odd}\}$ to be the set of vertices that have odd degree in S. Now we compute a minimum cost T_S-join; this is a set J_S of edges such that a vertex has odd degree in (V, J_S) if and only if it is contained in T_S. See Figure 1 for an example. Finally, we observe that by taking the disjoint union of S and J_S we obtain a connected Eulerian graph. In other words, $(V, S \,\dot\cup\, J_S)$ is connected and every vertex has even degree. Therefore, by Euler's theorem we can find a closed walk that uses every edge of $(V, S \,\dot\cup\, J_S)$ exactly once – this closed walk also visits every vertex at least once and is therefore a feasible solution.

What is the cost of this walk? – First we observe that every feasible solution to the TSP must contain a spanning tree. Therefore we can bound the cost $c(S)$ of the minimum-cost spanning tree S in Christofides' algorithm by the cost OPT of an optimum TSP solution. Moreover, it is not very difficult to see that the edges of every TSP solution can be partitioned into two T_S-joins. Therefore, $c(J_S) \leq \frac{\text{OPT}}{2}$. This implies that Christofides' algorithm computes a tour of cost at most $\frac{3}{2} \cdot \text{OPT}$.

Graph TSP

Graph TSP is the important special case of TSP where every edge has cost one, i.e. $c(e) = 1$ for all $e \in E$. In contrast to the general weighted case, we know better approximation algorithms for graph TSP than Christofides' algorithm. The first improvement for graph TSP after Christofides' algorithm was a $(1.5 - \epsilon)$-approximation algorithm (for some small $\epsilon > 0$) by Oveis Gharan, Saberi, and Singh [16]. Their algorithm proceeds similarly as Christofides, but chooses the spanning tree differently (using random sampling from a certain maximum-entropy distribution). Then, within less than a year in 2011–2012, the approximation ratio for graph TSP has been improved several times. Mömke and Svensson [13] introduced a technique called removable pairings and achieved an approximation ratio of 1.461. Then Mucha [14] improved the analysis of the same algorithm to $\frac{13}{9}$. Finally, Sebő and Vygen [19] gave a $\frac{7}{5}$-approximation algorithm, which remains the best result known until today. Their algorithm uses ear-decompositions of graphs and matroid intersection. See Figure 2 for an illustration of the development of the approximation ratio.

3 The s-t-path TSP

The s-t-path TSP is the probably best-studied generalization of TSP. Here the start $s \in V$ of the tour and the end $t \in V$ of the tour are given. In contrast to TSP, the start and end might be distinct. More formally, we are looking for an s-t-walk in G that visits every vertex at least once and has minimum cost. Thus, TSP is the special case of the s-t-path TSP where $s = t$. Similarly, the s-t-path *graph* TSP is the generalization of graph TSP where we are given $s, t \in V$ (in addition to the graph G) and we are looking for an s-t-walk in G that visits every vertex at least once and has a minimum number of edges.

The path version might seem to be almost the same problem as TSP. However, the results for TSP and graph TSP do not generalize as easily to the path version as one might hope. For example, the famous Christofides' algorithm that we discussed above can be generalized to the path version – but its approximation ratio is only $\frac{5}{3}$ as shown by Hoogeveen [11]. Moreover, until very recently the best known approximation ratios differed significantly between TSP and the s-t-path TSP, and the same holds for graph TSP and s-t-path graph TSP; see Figure 2.

The first improvement over Christofides' algorithm for the s-t-path TSP was obtained by An, Kleinberg, and Shmoys [1] in 2011. They proposed the *Best-of-many Christofides* algorithm: compute a solution to the classical linear programming relaxation, write this fractional solution as a convex combination of (incidence vectors of) spanning trees; then proceed with each of the trees as in Christofides' algorithm and take the best of the resulting tours. Following this approach, there have been several improvements [17, 28, 10, 18], each incorporating previous ideas and introducing new ones. This line of work has led to the best known upper bound on the integrality gap of the classical linear programming relaxation: the bound is derived from the analysis of an algorithm by Sebő and van Zuylen [18, 26].

Then in [25], Traub and Vygen introduced an approach based on dynamic programming. They still used ideas from

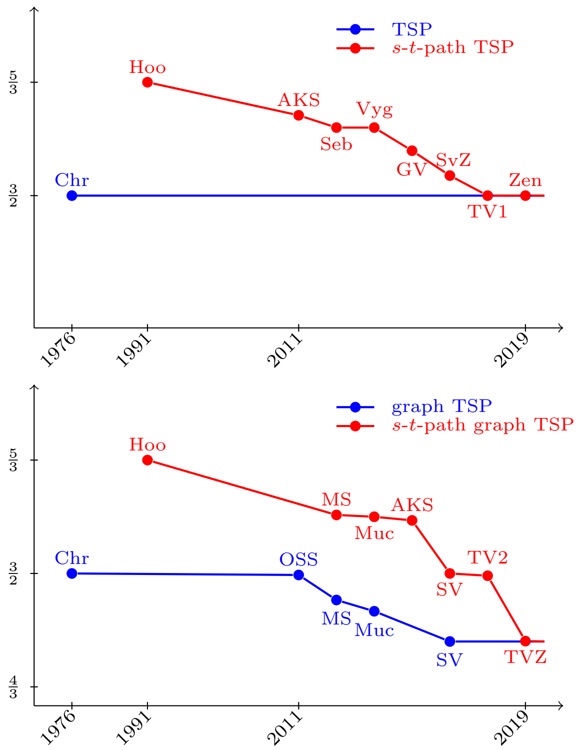


Figure 2: Development of the best known approximation ratios for TSP and s-t-path TSP (top) as well as graph TSP and s-t-path graph TSP (bottom). (The horizontal time axis is not to scale.) The abbreviations next to the points indicating the different results refer to the following publications: Chr [5, 20], Hoo [11], AKS [1], Seb [17], Vyg [28], GV [10], SvZ [18], TV1 [25], Zen [29], OSS [16], MS [13], Muc [14], TV2 [24], TVZ [27].

earlier papers, in particular from [1], but overall their approach is very different from the previous line of work. Up to that point, dynamic programming had not been used for approximation algorithms for the s-t-path TSP, but it turned out to be a useful tool in several recent advances: Zenklusen [29] gave a very elegant $\frac{3}{2}$-approximation algorithm for the s-t-path TSP, simplifying and improving the algorithm from [25]. The approximation ratio of his algorithm matches the approximation guarantee of Christofides' algorithm for the special case TSP.

Another application of dynamic programming is a very recent black-box reduction from s-t-path TSP to TSP [27]: any α-approximation algorithm for TSP implies an $(\alpha+\epsilon)$-approximation algorithm for the s-t-path TSP (for any fixed $\epsilon > 0$). Therefore, the path version is in general not much harder to approximate than its special case TSP.

3.1 Dynamic programming for the s-t-path TSP

There are several ways to employ dynamic programming in the context of the s-t-path TSP [25, 24, 15, 29, 27]. As an example, let us now explain one such application from [24]. We will sketch the proof of the following theorem that reduces the s-t-path graph TSP to the special case where the vertices s and t are "relatively close".

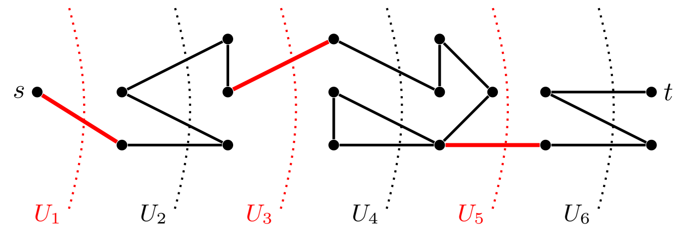


Figure 3: An s-t-tour F^* and the sets U_i for $i = 1, \dots, \mathrm{dist}(s,t) = 6$. (The graph G is not shown here.) The cuts $\delta(U_i)$ that contain exactly one edge of F^* are shown in red. The red cuts and edges are those that we guess using a dynamic program. They partition the instance into several smaller instances of the s-t-path TSP.

Theorem 1. *Let $\alpha > 1$ and $\epsilon > 0$.*

Suppose there is a polynomial-time algorithm for the s-t-path graph TSP that is an α-approximation on instances where the distance of s and t is at most $\left(\frac{1}{3}+\epsilon\right) \cdot \mathrm{OPT}$.
(Here OPT denotes the length of an optimum s-t-tour.)

Then there is a polynomial-time α-approximation algorithm for the s-t-path graph TSP.

This theorem has been used in [24] to obtain a 1.497-approximation algorithm for the s-t-path graph TSP. It can also be extended to the weighted s-t-path TSP, but for simplicity our explanations will be for the unit-weight case.

Recall that Christofides' algorithm yields a $\frac{5}{3}$-approximation algorithm for the s-t-path TSP. Using also the assumption of the theorem, we have

- an α-approximation algorithm $\mathcal{A}$ for the case $\mathrm{dist}(s,t) < \left(\frac{1}{3}+\epsilon\right) \cdot \mathrm{OPT}$, and
- a β-approximation algorithm $\mathcal{B}$ for the general case (without restriction on the distance of s and t),

where $\beta > 1$ is some constant. For example, $\mathcal{B}$ could be Christofides' algorithm and $\beta = \frac{5}{3}$. Our goal is to obtain an α-approximation algorithm for the general case. We may assume $\beta > \alpha$; otherwise $\mathcal{B}$ is already an α-approximation algorithm.

For instances with small distance of s and t we can simply apply algorithm $\mathcal{A}$. Let us consider the other case, where $\mathrm{dist}(s,t) > \left(\frac{1}{3}+\epsilon\right) \cdot \mathrm{OPT}$. For $i = 1, \dots, \mathrm{dist}(s,t)$ we will consider the set $U_i := \{v : \mathrm{dist}(s, v_i) < i\}$. See Figure 3 for an illustration. The key observation is that for many of these sets, the boundary $\delta(U_i)$ is only crossed once by an optimum s-t-tour. (Here $\delta(U_i)$ denotes all edges with one endpoint inside U_i and the other endpoint outside of U_i.)

Let us first see why this is the case. Denote by F^* the edge set of some fixed optimum s-t-tour; then $|F^*| = \mathrm{OPT}$. Now we need the following observations:

1. The cuts $\delta(U_i)$ are pairwise disjoint, i.e. no edge crosses the boundary of multiple sets U_i.
2. Every cut $\delta(U_i)$ contains an odd number of edges of F^*.

Property (1) follows from the definition of the sets U_i. Property (2) holds because the start s of the s-t-tour F^* is inside U_i, but the end t is outside. Therefore, the boundary of U_i must be crossed an odd number of times. Because the $\mathrm{dist}(s,t)$ many cuts $\delta(U_i)$ are disjoint, the average number $|F^* \cap \delta(U_i)|$ of F^*-edges in $\delta(U_i)$ is at most $\frac{|F^*|}{\mathrm{dist}(s,t)} = \frac{\mathrm{OPT}}{\mathrm{dist}(s,t)}$, which is significantly less than 3. Now (2) implies that no cut $\delta(U_i)$ contains exactly two edges of F^* and therefore a significant number of these cuts have to contain only one edge of F^*. (These are shown in red in Figure 3.)

Now the idea is to use dynamic programming to guess those cuts $\delta(U_i)$ with $|\delta(U_i) \cap F^*| = 1$ and also the F^*-edge in each of these cuts. Why is this useful? – Suppose we knew these cuts and edges. These cuts partition the instance into several smaller instances of the s-t-path TSP; see Figure 3. Then we can apply our β-approximation algorithm $\mathcal{B}$ to all these sub-instances. We clearly have an approximation ratio of β, but in fact we can give a better bound $\beta' < \beta$. The reason is that our approximation factor β applies not to all edges of F^*, but only to those that are contained in the sub-instances; these are the black edges in Figure 3. The fraction of these edges is bounded away from 1. Now we have not reached our original goal of obtaining an α-approximation algorithm, but we made some progress: before, we only had a β-approximation algorithm; now we have a β'-approximation algorithm for $\beta' < \beta$. By simply repeating the same argument until β' finally becomes α, one can show Theorem 1.

This is one way dynamic programming has been used to obtain approximation algorithms for the s-t-path TSP. There are also several other variants and extensions of this basic dynamic programming idea, and different ways of combining dynamic programming with other techniques. Here we guessed some edges that are part of an optimum tour. In [25, 15, 29] dynamic programming is also used to guess certain valid inequalities; these are then used to strengthen a linear programming relaxation. Moreover, in the recent black-box reduction from the s-t-path (graph) TSP to (graph) TSP [27], related techniques are applied in a much more general setting: Traub, Vygen, and Zenklusen apply dynamic programming to a new generalization of the s-t-path TSP, called Φ-TSP, that plays a crucial role in their reduction.

4 ATSP

The asymmetric traveling salesman problem, where the input consists of a directed and strongly connected graph and nonnegative edge costs, is much harder than its symmetric counterpart. Intuitively, the reason is the following. In the symmetric case, after using a locally desirable / cheap edge we can go back using the same cheap edge. In the asymmetric case, the cost of going back is difficult to bound. Indeed, the "analogue of Christofides' algorithm" that begins by taking a minimum-cost spanning tree and then tries to fix the degrees by adding edges is only a $\Theta(n)$-approximation.

The first notable approximation algorithm for ATSP was given by Frieze, Galbiati and Maffioli [8]. They achieve a $\log_2 n$-approximation by repeatedly finding minimum-cost cycle covers and contracting the already-connected components of the output graph. This approach was refined in several papers [4, 12, 7], but they did not obtain an asymptotic improvement on the approximation ratio. The first such algorithm – an $O(\log n \,/\, \log\log n)$-approximation – was given at the beginning of this decade by Asadpour et al. [3], who introduced a new approach based on carefully sampling a random spanning tree using the solution of a standard linear programming relaxation of ATSP. Their methods were extended to give a constant-factor approximation algorithm for ATSP on planar graphs due to Oveis Gharan and Saberi [9] and a $O(\mathrm{poly}\log\log n)$-estimation algorithm[1] due to Anari and Oveis Gharan [2].

In 2015, Svensson [21] gave the first constant-factor approximation algorithm for ATSP on unweighted graphs (a variant that could be called graph ATSP, although this name does not seem to be used). To that end, he defined a new, easier problem called Local-Connectivity ATSP, and showed that for any class of graphs, it is enough to solve that easier problem in order to obtain a constant-factor approximation for ATSP on that class of graphs. It turns out that Local-Connectivity ATSP is easy to solve on unweighted graphs, and the result follows. Later Svensson, Tarnawski and Végh [23] have been able to solve Local-Connectivity ATSP on graphs that have at most two different edge weights, thus implying a constant-factor approximation algorithm for ATSP on such graphs.

In 2017, this line of work has culminated in a constant-factor approximation algorithm for ATSP on general graphs due to Svensson, Tarnawski and Végh [22]. Unlike the two previous results, it does not follow directly from applying Svensson's reduction and solving Local-Connectivity ATSP on general graphs. Rather, they proceed via a series of reductions to more and more structured ATSP instances. Each reduction effectively proves a statement of the form: if there is a constant-factor approximation algorithm for certain more structured instances, then there is a constant-factor approximation algorithm for certain less structured instances. After going through several such steps, they finally apply Svensson's reduction and are left with the task of solving Local-Connectivity ATSP on highly special instances that they call *vertebrate pairs*, which they are able to do using methods similar to those from [23].

In the following we give a high-level explanation of the first reduction of [22]. As it is performed with no loss in approximation ratio, every approximation algorithm for ATSP could start from it.

4.1 Laminarly-weighted instances

All approximation algorithms for ATSP with ratios asymptotically better than $O(\log n)$ are based on solving the standard linear programming relaxation, which is called the Held-Karp relaxation (see Figure 4). This linear program, though exponentially-sized, can be solved in polynomial time using

[1] An α-estimation algorithm outputs a value that is guaranteed to be within a factor α of the true optimum value, but produces no tour of corresponding cost.

$$\begin{array}{lll} \text{minimize} & \sum_{e\in E} c(e)x(e) & \\ \text{subject to} & x(\delta^+(v)) = x(\delta^-(v)) & \text{for } v \in V, \\ & x(\delta(S)) \geq 2 & \text{for } \emptyset \neq S \subsetneq V, \\ & x(e) \geq 0 & \text{for } e \in E. \\ & & (\mathrm{LP}(G,c)) \end{array}$$

Figure 4: The Held-Karp relaxation for ATSP. Here $\delta^+(v)$ is used to denote the set of outgoing edges of vertex v, $\delta^-(v)$ denotes the incoming edges, and $\delta(S)$ stands for the boundary of a set S (both incoming and outgoing edges).

$$\begin{array}{lll} \text{maximize} & \sum_{\emptyset\neq S\subsetneq V} 2\cdot y_S & \\ \text{subject to} & \sum_{S:(u,v)\in\delta(S)} y_S + \alpha_u - \alpha_v \leq c(u,v) & \text{for } (u,v)\in E, \\ & y_S \geq 0 & \text{for } \emptyset \neq S \subsetneq V. \\ & & (\mathrm{DUAL}(G,c)) \end{array}$$

Figure 5: The dual linear program of $\mathrm{LP}(G,c)$ (see Figure 4).

the ellipsoid algorithm with a separation oracle. It turns out to have several strong properties that allow one to write down any cost function for ATSP in a highly structured form. We summarize them in the following theorem.

Theorem 2. *Without loss of generality, one can assume that the cost function is of the form*

$$c(e) = \sum_{S\in\mathcal{L}\ :\ e\in\delta(S)} y_S \qquad \textit{for every edge } e \in E, \tag{1}$$

where $\mathcal{L} \subseteq 2^V$ *is a* laminar *family of subsets of vertices and* $y : \mathcal{L} \to \mathbb{R}_{\geq 0}$ *is a vector of nonnegative values assigned to those sets. In addition, one can also assume that every set* $S \in \mathcal{L}$ *is* tight*, i.e. that we have* $x(\delta(S)) = 2$*, where* x *is an optimal solution to* $\mathrm{LP}(G,c)$*.*

Recall that a family of sets is laminar if no two sets cross each other; in other words, for any $A, B \in \mathcal{L}$ we have $A \subseteq B$, $B \subseteq A$ or $A \cap B = \emptyset$. See Figure 6 for an example.

To obtain the properties postulated by Theorem 2, we solve the dual of $LP(G,c)$ – see Figure 5. It turns out that nicely structured solutions can be obtained by also minimizing a second objective, subject to the main dual objective being optimal; see Lemma 1 below.

Lemma 1. *Let* (y,α) *be a solution to* $\mathrm{DUAL}(G,c)$ *that, subject to being optimal for* $\mathrm{DUAL}(G,c)$ *(i.e. minimizing* $\sum_S 2\cdot y_S$*), also minimizes* $\sum_S |S|\cdot y_S$*. Then its support* $\mathcal{L} := \{S \subseteq V : y_S > 0\}$ *is laminar.*

The proof of Lemma 1 uses the well-known technique of uncrossing: one shows that if the condition of the lemma is not satisfied, then a small modification to y would yield a

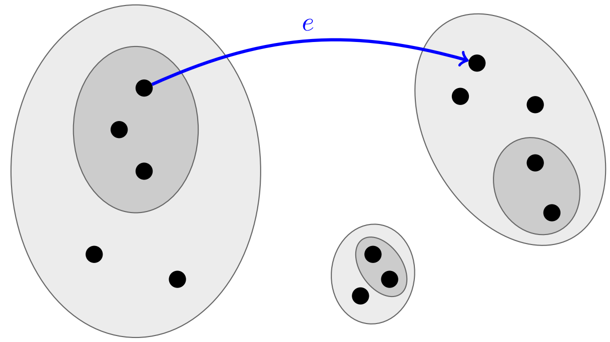


Figure 6: Example of a so-called laminarly-weighted instance. The figure shows the sets of the laminar family (in gray) and one edge e (in blue). The edge crosses three sets in the laminar family, say S_1, S_2, S_3, and so $c(e) = y_{S_1} + y_{S_2} + y_{S_3}$.

solution that is still optimal for $\mathrm{DUAL}(G,c)$, but has lower value of $\sum_S |S|\cdot y_S$.

Lemma 1 is the main step in the proof of Theorem 2. The next step is to remove edges with $x_e = 0$ from the graph and harness the complementarity slackness conditions of $\mathrm{LP}(G,c)$ and $\mathrm{DUAL}(G,c)$: since we now have $x_e > 0$ for all edges e, we automatically get the equality

$$\sum_{S:\ (u,v)\in\delta(S)} y_S + \alpha_u - \alpha_v = c(u,v)\,. \tag{2}$$

Finally, we adjust the original edge weights $c(e)$ with the potentials α_v from the dual solution: $c'(u,v) := c(u,v) - \alpha_u + \alpha_v$. In this way we obtain a new weight function that is equivalent to the original, in the sense that any Eulerian set of edges has the same weight under both functions. Moreover, together with (2) this now gives (1).

See Figure 6 for an example of an instance obtained from Theorem 2. Note that the asymmetricity of the instance has been restricted to only the directedness of the graph (as opposed to the weights being arbitrary): indeed, if we happen to have $(u,v) \in E$ and $(v,u) \in E$ for some vertices u, v, then $c(u,v) = c(v,u)$. Note also that this yields a highly hierarchical structure and that, since every set in the laminar family is tight (the LP suggests to enter and exit this set exactly once), it intuitively makes sense to consider instances where some of the laminar sets have been contracted into single vertices. For more on how to develop these intuitions into a constant-factor approximation algorithm, see [22][2].

5 Open questions

In the realm of symmetric TSP, the most famous open problem is to improve upon the 1.5-approximation factor of Christofides and Serdyukov. This has been done for the case of graph-TSP; for what more general classes of instances can one hope to obtain an improvement? Consider for example *node-weighted instances*: ones where for every edge $\{u,v\} \in E$ we have $c(u,v) = f(u) + f(v)$ for some function

[2]The STOC 2018 extended abstract holds a 10-page intuitive overview of the different steps and reductions.

$f : V \to \mathbb{R}_{\geq 0}$. Another compelling class might be graphs with at most two different edge weights. The standard linear program for symmetric TSP unlocks the same laminar structure as we discussed for ATSP in Section 4.1; how can we make good use of it?

For ATSP, the first constant approximation factor was 5500; this was later brought down to 506. Clearly, neither number is the correct answer. Then, what is the exact approximability? For now, can one get, say, a 20-approximation algorithm?

Can we extend the black-box reduction from s-t-path TSP to TSP from [27] also to the asymmetric case? Feige and Singh [7] gave a reduction from s-t-path ATSP to ATSP, but in constrast to the reduction in [27] for the symmetric case, they lose a factor of roughly 2 in the approximation ratio. Can we improve on this?

REFERENCES

[1] H.-C. An, R. Kleinberg, and D.B. Shmoys. Improving Christofides' algorithm for the s-t path TSP. *Journal of the ACM*, 62(Article 34), 2015.

[2] N. Anari and S. Oveis Gharan. Effective-resistance-reducing flows, spectrally thin trees, and asymmetric TSP. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 20–39, 2015.

[3] A. Asadpour, M.X. Goemans, A. Madry, S. Oveis Gharan, and A. Saberi. An $O(\log n/ \log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. *Operations Research*, 65:1043–1061, 2017.

[4] M. Bläser. A new approximation algorithm for the asymmetric TSP with triangle inequality. *ACM Transactions on Algorithms*, 4(4), 2008.

[5] N. Christofides. Worst-case analysis of a new heuristic for the Travelling Salesman Problem, 1976.

[6] W.J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation.* Princeton University Press, 2012.

[7] U. Feige and M. Singh. Improved approximation algorithms for traveling salesperson tours and paths in directed graphs. In *Proceedings of the 10th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 104–118, 2007.

[8] A.M. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12(1):23–39, 1982.

[9] S. Oveis Gharan and A. Saberi. The asymmetric traveling salesman problem on graphs with bounded genus. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 967–975, 2011.

[10] C. Gottschalk and J. Vygen. Better s-t-tours by Gao trees. *Mathematical Programming B*, 172:191–207, 2018.

[11] J.A. Hoogeveen. Analysis of Christofides' heuristic: some paths are more difficult than cycles. *Operations Research Letters*, 10(5):291–295, 1991.

[12] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *Journal of the ACM*, 52(4):602–626, 2005.

[13] T. Mömke and O. Svensson. Removing and adding edges for the Traveling Salesman Problem. *Journal of the ACM*, 63(1):2:1–2:28, 2016.

[14] M. Mucha. $\frac{13}{9}$-approximation for graphic TSP. *Theory of Computing Systems*, 55(4):640–657, 2014.

[15] M. Nägele and R. Zenklusen. A new dynamic programming approach for spanning trees with chain constraints and beyond. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1550–1569, 2019.

[16] S. Oveis Gharan, A. Saberi, and M. Singh. A randomized rounding approach to the Traveling Salesman Problem. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 550–559, 2011.

[17] A. Sebő. Eight-fifth approximation for the Path TSP. In *Proceedings of 16th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 263–373, 2013.

[18] A. Sebő and A. van Zuylen. The salesman's improved paths through forests. *Journal of the ACM*, 66(4):28:1–28:16, 2019.

[19] A. Sebő and J. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014.

[20] A. I. Serdjukov. Some extremal bypasses in graphs [in Russian]. *Upravlyaemye Sistemy*, 17:76–79, 1978.

[21] O. Svensson. Approximating ATSP by relaxing connectivity. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015.

[22] O. Svensson, J. Tarnawski, and L.A. Végh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 204–213, 2018.

[23] O. Svensson, J. Tarnawski, and L.A. Végh. Constant factor approximation for ATSP with two edge weights. *Mathematical Programming*, 172(1-2):371–397, 2018.

[24] V. Traub and J. Vygen. Beating the integrality ratio for s-t-tours in graphs. In *Proceedings of 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 766–777, 2018.

[25] V. Traub and J. Vygen. Approaching 3/2 for the s-t-path TSP. *Journal of the ACM*, 66(14), 2019.

[26] V. Traub and J. Vygen. An improved upper bound on the integrality ratio for the s-t-path TSP. *Operations Research Letters*, 47:225–228, 2019.

[27] V. Traub, J. Vygen, and R. Zenklusen. Reducing Path TSP to TSP. arXiv:1907.10376, 2019.

[28] J. Vygen. Reassembling trees for the traveling salesman. *SIAM Journal on Discrete Mathematics*, 30(2):875–894, 2016.

[29] R. Zenklusen. A 1.5-approximation for Path TSP. In *Proceedings of 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1539–1549, 2019.

In Memoriam

Egon Balas (1922 – 2019)

Self-photographed *(Photo courtesy of Wikipedia)*.

Egon Balas' life reads like an adventure tale. Born in Cluj, Romania, in 1922 into a Hungarian-Jewish family, he was involved in underground activities during World War II, hiding, captured, tortured, and escaped. After the war he became a Romanian diplomat in London, and held other high positions in post-war Communist Romania. Then it was prison again, repeated interrogation by the dreaded Securitate, solitary confinement for more than two years, release, and expulsion from the Communist party. He only started a career in Mathematics at age 37 in the late 1950s [4].

At 37, Egon Balas joined the Forestry Institute in Bucharest, which planned and scheduled timber harvest in Romania. To develop appropriate logistics tools, Egon Balas became an autodidact, studying mathematics and operations research on his own from books he could get his hands on. Peter Hammer, who was to become a prominent Operations Researcher himself, was also working at the Forestry Institute at the time. Together, they developed tools for transportation planning based on network flow theory and linear programming. They published a dozen papers together (Peter Hammer was using the name Ivanescu at the time) mostly in Romanian.

In 1962 Egon Balas confronted an interesting variation on the wood harvesting problem: In one area of the forest, a network of roads had to be built in order to access various plots. Decisions about which plots to harvest and where to build roads were intricately connected. It involved logical decisions: If road segment A were built, then road segment B would also have to be built to reach A. Egon Balas formulated the problem as a linear program in 0,1 variables, recognizing the versatility of this model for a wide range of applications.

There was no computer code to solve 0,1 programs at the time, so Egon Balas designed his own algorithm, simple enough that instances with about 30 binary variables could be solved by hand. The algorithm performed an implicit enumeration of the solution set, relied on a series of logical tests that explored the implications of fixing certain variables to 0 or to 1, and the only operations it needed were additions and comparisons. Egon Balas' algorithm can be viewed as a precursor of constraint propagation in Constraint Programming.

The work was presented to the research community in the West at the International Symposium in Mathematical Programming in London in 1964. The Romanian government would not allow Egon Balas to travel to England to present his paper, so his talk was read by a colleague at the conference. This was one of the many constraints that lead to Egon Balas' disillusionment with Communist Romania.

The same year a short version of his paper was published in *Comptes Rendus de l'Académie des Sciences*, Paris, France. The full-length paper "An Additive Algorithm for Solving Linear Programs with Zero-One Variables" appeared in *Operations Research* in 1965 [1]. It was extremely influential in the development of integer programming, establishing implicit enumeration and branch-and-bound as a simple and powerful solution methodology. In fact, this paper was identified as the most cited paper in Operations Research in the early 80s (Citation Classic in *Current Contents 1982*).

By the early 1960s, the oppressive conditions and lack of freedom in Romania had become intolerable to Egon Balas and he applied to emigrate. His application was denied. After several further rejected applications, each time with increased hardship on the family, he and his wife and two daughters were granted permission to emigrate in 1966. By that time, Egon Balas already had visibility in the West thanks to his work on the additive algorithm. He first went to Rome, Italy, where he spent six months as a Research Fellow at the International Computation Centre, headed by Claude Berge. During this period, Egon Balas also managed to earn two doctorates, one in Economics from the University of Brussels, and the other in Mathematics from the University of Paris. The next year, in 1967, Egon Balas was offered a professorship at Carnegie Mellon University. Bill Cooper, one of founders of the Graduate School of Industrial Administration, and a faculty member there, was key in this brilliant recruitment decision. He was very familiar with Egon Balas' recent research accomplishments as he had been the Associate Editor of *Operations Research* in charge of handling Egon Balas' paper on the additive algorithm. Egon Balas was forever grateful to Carnegie Mellon for the stability that this position provided to his family.

Egon Balas' most significant contribution is undoubtedly his extensive work on disjunctive programming, starting with intersection cuts (*Operations Research 1971*). These ideas were novel and the Operations Research community was slow to accept them.

Cutting planes had been introduced by Dantzig, Fulkerson and Johnson (1954) and Gomory (1958). But by the late 60s and throughout the 70s and 80s, the general sentiment regarding the practicality of general cutting planes had become rather negative, in great part due to the success of branch-and-bound algorithms such as the additive algorithm.

Egon Balas understood that enumeration was inherently exponential in complexity and that it could only tackle small or medium-size instances. He felt that the real potential was in convexifying the feasible set, potentially reducing an integer program to a simpler linear program. He also felt that theory was lacking. Using tools of convex analysis, he showed how to derive rich families of cutting planes from any feasible basis of a linear relaxation, and any convex set S whose interior contains the basic solution but no feasible integer point. These cuts are Balas' intersection cuts. When the convex set is the region between two parallel hyperplanes, one recovers Gomory's mixed integer cut as a special case. Egon Balas observed that intersection cuts derived from polyhedral sets S can be understood using an alternate point of view: if the polyhedron S is defined by linear inequalities, then the requirement that no feasible point is in the interior of S can be described through a disjunction of the reverse inequalities. The feasible region can then be regarded as a union of polyhedra.

This new viewpoint gives rise to important generalizations. It motivated Egon Balas to introduce disjunctive programming, defined as optimization over a union of polyhedra. He proved two fundamental results on disjunctive programs that have far-reaching consequences for the solution of linear 0,1 programs. First, there is a compact representation of the convex hull of the union of polyhedra in a higher-dimensional space. Projecting back onto the original space gives a full description of the convex hull. As a result, one can compute a deepest disjunctive cut by solving a linear program. The number of variables and constraints in the higher-dimensional representation only grows linearly in the number of polyhedra, which makes this a practical tool when the number of disjunctions is not too large. Second, for a large class of disjunctive programs, called facial, the convexification process described above can be performed sequentially. For example, 0,1 programs are facial disjunctive programs. This means that if there are n 0,1 variables, the convex hull of the solution set can be obtained in n steps, imposing the 0,1 conditions one at a time. This distinguishes 0,1 programs from more general integer linear programs. These theorems were proved in a technical report [3] in 1974. Unfortunately, the significance of the results was not recognized by the referees at the time and the paper was rejected for publication. Twenty-four years later the importance of Egon Balas' pioneering work had finally become clear, and the technical report was eventually published as an invited paper in 1998.

In the Spring of 1988, Laci Lovász gave a beautiful talk at Oberwolfach about his on-going work with Lex Schrijver on cones of matrices. Sebastian Ceria, who was just starting his PhD at Carnegie Mellon, and I decided to investigate what would happen if one performed the Lovász-Schrijver operation sequentially, one variable at a time, with the idea of making it more practical for implementation. We were very excited to realize that, as in the full Lovász-Schrijver procedure, our simplified lift-and-project procedure still generated the convex hull in n steps for a problem with n 0,1 variables. When we showed this result to Egon, his reaction was immediate: "There is nothing new under the sun. This is the sequential convexification theorem!" Egon was right of course. There was a nice connection between our streamlined version of the Lovász-Schrijver procedure and disjunctive programming.

This connection was very fruitful, providing a perfect framework for cut generation. Sebastian, Egon and I had much fun collaborating on this project [6]. We developed the solver MIPO (Mixed Integer Program Optimizer) which incorporated lift-and-project cuts within a branch-and-cut framework. The success of lift-and-project cuts motivated us to try other general-purpose cuts, such as the Gomory mixed integer cuts. These cuts had a bad reputation, with repeated claims in the literature that they have a poor performance in practice. So we were very surprised to discover that they also worked very well [7]. The sentiment about general cutting planes changed overnight in the Integer Programming community. By the late 1990s, all commercial solvers for mixed integer linear programs were using a whole battery of general-purpose cuts, resulting in a very significant improvement in the size of instances that could be solved optimally.

Egon Balas' research contributions span a broad range of topics. On of subject of disjunctive cuts itself, I left out several important aspects, such as monoidal cut strengthening (Balas and Jeroslow [8]), the efficient generation of the cuts (Balas and Perregaard [9]), and other recent directions such as generalized intersection cuts. At age 96, Egon Balas wrote his first (and only) textbook *Disjunctive Programming* [5] presenting the advances made in this area over nearly five decades.

Egon Balas also made noteworthy research contributions on the knapsack and set-covering problems, and in the area of scheduling: machine scheduling via disjunctive graphs, the shifting bottleneck procedure for job shop scheduling (with Joe Adams and Dan Zawack), choosing the overall size of the US strategic petroleum reserve, the prize-collecting traveling salesman problem, an application to scheduling rolling mills in the steel industry (with Red Martin), and several others.

Egon Balas' contributions have been recognized through numerous prizes and honors.

- Prizes: John von Neumann Theory Prize of INFORMS (1995), EURO Gold Medal of the European Association of Operational Research Societies (2001), Larnder Prize of the Canadian Operations Research Society (2017).
- Honors: elected an INFORMS Fellow (2002), elected an external member of the Hungarian Academy of Sciences (2004), inducted into the IFORS Hall of Fame (2006), elected a member of the National Academy of Engineering (2006), elected a corresponding member of the Academy of Sciences of Bologna, Italy (2011), elected a SIAM Fellow (2016).
- Honorary doctorates: University of Elche, Spain (2002), University of Waterloo, Canada (2005), University of Liège, Belgium (2008).

Egon Balas was a good friend and colleague, as well as a great tennis partner. He was still a formidable player in

December 2018 a few months before his death.

Gérard Cornuéjols, *Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA*. Email: gc0v@andrew.cmu.edu

REFERENCES

[1] E. Balas, An Additive Algorithm for Solving Linear Programs with Zero-one Variables, *Operations Research* 13 (1965) 517-546.

[2] E. Balas, Intersection Cuts – A New Type of Cutting Planes for Integer Programming, *Operations Research* 19 (1971) 19-39.

[3] E. Balas, Disjunctive Programming: Properties of the Convex Hull of Feasible Points. MSRR No. 348, Carnegie Mellon University, July 1974. Published as an invited paper in *Discrete Applied Mathematics* 89 (1998) 3-44.

[4] E. Balas, *Will to Freedom: A Perilous Journey Through Fascism and Communism*, Syracuse University Press (2000) 496p.

[5] E. Balas, *Disjunctive Programming*, Springer (2018) 243p.

[6] E. Balas, S. Ceria, G. Cornuéjols, A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs, *Mathematical Programming* 58 (1993) 295-324.

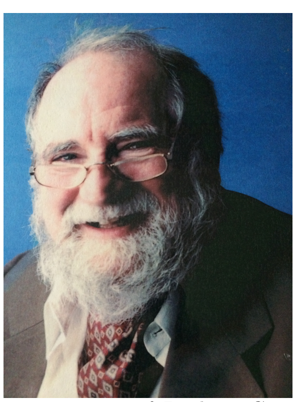
[7] E. Balas, S. Ceria, G. Cornuéjols, N. Natraj, Gomory Cuts Revisited, *Operations Research Letters* 19 (1996) 1-9.

[8] E. Balas, R.G. Jeroslow, Strengthening Cuts for Mixed Integer Programs, *European Journal of Operations Research* 4 (1980) 224-234.

[9] E. Balas, M. Perregaard, A Precise Correspondence Between Lift-and-Project Cuts, Simple Disjunctive Cuts, and Mixed Integer Gomory Cuts for 0-1 Programming, *Mathematical Programming B* 94 (2003) 221-245.

In Memoriam

Andrew Conn (1946 – 2019)



(Photo courtesy of Barbara Conn.)

I first met Andrew Conn forty years ago, at the International Symposium on Mathematical Programming in Montréal in the summer of 1979. I was attending the conference as a fresh PhD and was looking for somebody to visit in Canada for a few days. I mentioned this to Andy, who immediately invited me to visit him in Waterloo… and to stay at his home during my visit! This was, for me, the first manifestation of several of Andy's qualities: continued supportive interest in young people, hospitality and generosity. I did enjoy my visit very much and we agreed to keep each other informed of our future ideas. My second visit to Andy followed in 1984, where we started to discuss optimization of large nonlinear problems (a few tens of variables by then) together with a colleague of his named Nick Gould. The discussion were lively and we had marvellous plans on how to develop our new ideas and associated code. But collaboration got truly going only in 1986, when Andy, on sabbatical in Grenoble, invited Nick (then back in Europe) and me for a working week. I have very fond memories of that week: intense discussions and good food, the real birth of the LANCELOT project. It resulted in our first CGT paper (Andy really liked the fact that our initials gave the same acronym as that of the French left-wing trade union) on trust region methods for problems with bound constraints.

This was the start of a very long and very enjoyable collaboration with Andy and Nick leading to the publication, in 1991, of the LANCELOT package (and associated theory) and the first CUTE collection. But we had such a good time working together that we could not stop there, and continued to work together, from letters and email to (many) visits and meetings at conferences and back, on the development of trust-region methods. This culminated in the publication in 2000 of "Trust Region Methods", the first volume of the SIAM MPS series of optimization books. Those possibly doubting the intensity of our common involvement should have a look at the sheer size of this book and the scope of its table of contents! We continued to collaborate after that for some years, in particular with Katya Scheinberg on the topic of derivative-free optimization.

Altogether, Andy and I have published 44 joint papers, 39 of which with Nick, three with Katya, eight with Annick Sartenaer and three with Dominique Orban, not to mention a few unpublished crazy ideas. But of course, such a long collaboration is only possible when based on true friendship. And our true friendship was built by sharing interests beyond the sphere of mathematical work. It is fair to say that it would have been impossible without our strong common interest in good food and good music, and without the benign and active complicity of our spouses. The houses of Barbara and Andy (which I visited so often in Waterloo, Poundridge and Mount-Vernon) have been a home away from home for many years. I will always remember my stays in this wonderful family where Barbara, Andy, Nick and me would spend long evenings discussing after an excellent dinner, sharing some good wine, many musical discoveries and more generally enjoying ourselves. I have too many good memories of good moments with Andy (ah, the smell of roasting coffee filling the house in the morning!) that I can't even think where to start, should I try to recall them all.

The passing of Andy took me by surprise. I knew of his health problems, but I was far from appreciating how serious they were. When Andy visited me in Namur last year, his usual radiant optimism and enthusiasm were unabated, and it seemed that we could see each other many more times.

Farewell, dear Andy! I will miss you, as I am sure many colleagues will miss you too.

Philippe L. Toint, *Namur Center for Complex Systems (naXys), University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium.* Email: philippe.toint@unamur.be

Postscript: I too have so many happy recollections of Andy. We first met at the National Physical Laboratory in London, and subsequently as colleagues at Waterloo, where the fabulous collaborations that Philippe mentions got underway. I share every sentiment that Philippe describes, and although Andy is no longer with us, the wonderful memories remain as strong as ever.

Nick Gould, *Numerical Analysis Group, Scientific Computing Department, STFC-Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, UK* Email: nick.gould@stfc.ac.uk

Bulletin

Email items to siagoptnews@lists.mcs.anl.gov *for consideration in the bulletin of forthcoming issues.*

1 Event Announcements

1.1 SIAM Conference on Optimization (SIOPT)

The SIOPT 2020 conference will be held at the Hung Hom campus of the Hong Kong Polytechnic University, May 26-29, 2020.

The SIAM Conference on Optimization will feature the latest research on the theory, algorithms, software, and applications of optimization. A particular emphasis will be put on applications of optimization in AI and data science, quantum computing, health care, finance, aeronautics, control, operations research, and other areas of science and engineering. The conference brings together mathematicians, operations researchers, computer and computational scientists, engineers, software developers and practitioners, thus providing an ideal environment to share new ideas and important problems among specialists and users of optimization in academia, government, and industry.

For details and online registration, visit https://www.siam.org/conferences/cm/conference/op20.

Deadlines
November 7, 2019: Minisymposium Proposal Submissions.
December 1, 2019: Contributed Lecture, Poster and Minisymposium Presentation Abstracts.
December 12, 2019: SIAM Student Travel Award and Postdoc/Early Career Travel Award Applications.

1.2 Conference on Integer Programming and Combinatorial Optimization

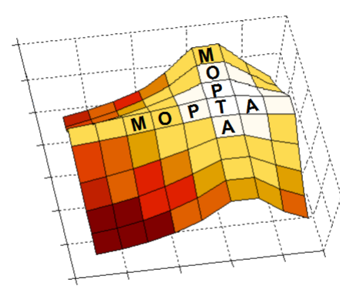
The 21st Conference on Integer Programming and Combinatorial Optimization (IPCO XXI) will take place from June 8 to 10 at the London School of Economics, in London UK. It will be organized by the Department of Mathematics. The conference will be preceded by a Summer School (June 6-7).

IPCO conference is under the auspices of the Mathematical Optimization Society. It is held every year. The conference is a forum for researchers and practitioners working on various aspects of integer programming and combinatorial optimization. The aim is to present recent developments in theory, computation, and applications in these areas.

For details and online registration, visit http://www.lse.ac.uk/IPCO-2020.

Deadlines
November 29, 2019: Abstract submissions.

1.3 Conference on Integer Programming and Combinatorial Optimization



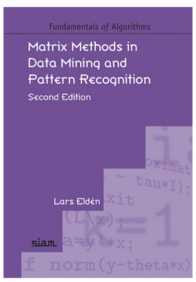
The MOPTA 2020 conference will take place at Lehigh University, 12-14 August 2020. MOPTA aims at bringing together a diverse group of people from both discrete and continuous optimization, working on both theoretical and applied aspects. There will be a small number of invited talks from distinguished speakers and contributed talks, spread over three days. Our target is to present a diverse set of exciting new developments from different optimization areas while at the same time providing a setting which will allow increased interaction among the participants. We aim to bring together researchers from both the theoretical and applied communities who do not usually have the chance to interact in the framework of a medium-scale event.

For details and online registration, visit http://coral.ie.lehigh.edu/~mopta

Deadlines
June 28, 2020: Abstract submission and Early Registration.
August 5, 2020: Registration.

2 Book Announcements

2.1 Matrix Methods in Data Mining and Pattern Recognition, Second Edition



By Lars Eldén
Publisher: SIAM
ISBN: 978-1-611975-85-7, xiv + 229 pages
Published: 2019
https://bookstore.siam.org/fa15

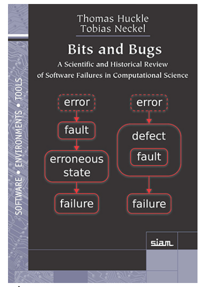
About the book: This thoroughly revised second edition provides an updated treatment of numerical linear algebra techniques for solving problems in data mining and pattern recognition. Adopting an application-oriented approach, the author introduces matrix theory and decompositions, describes how modern matrix methods can be applied in real life scenarios, and provides a set of tools that students can modify for a particular application.

Building on material from the first edition, the author discusses basic graph concepts and their matrix counterparts. He introduces the graph Laplacian and properties of its eigenvectors needed in spectral partitioning and describes spectral graph partitioning applied to social networks and text classification. Examples are included to help readers visualize the results. This new edition also presents matrix-based methods that underlie many of the algorithms used for big data.

The book provides a solid foundation to further explore related topics and presents applications such as classification of handwritten digits, text mining, text summarization, PageRank computations related to the Google search engine, and facial recognition. Exercises and computer assignments are available on a Web page that supplements the book.

Audience: Matrix Methods in Data Mining and Pattern Recognition, Second Edition is primarily for undergraduate students who have previously taken an introductory scientific computing/numerical analysis course and graduate students in data mining and pattern recognition areas who need an introduction to linear algebra techniques.

2.2 Bits and Bugs: A Scientific and Historical Review of Software Failures in Computational Science



By Thomas Hucckle and Tobias Neckel
Publisher: SIAM
ISBN: 978-1-611975-55-0, xii + 251 pages
Published: 2019
https://bookstore.siam.org/se29

About the book: In scientific computing (also known as computational science), advanced computing capabilities are used to solve complex problems. This self-contained book describes and analyzes reported software failures related to the major topics within scientific computing: mathematical modeling of phenomena; numerical analysis (number representation, rounding, conditioning); mathematical aspects and complexity of algorithms, systems, or software; concurrent computing (parallelization, scheduling, synchronization); and numerical data (such as input of data and design of control logic).

Readers will find lists of related, interesting bugs, MATLAB examples, and "excursions" that provide necessary background, as well as an in-depth analysis of various aspects of the selected bugs. Illustrative examples of numerical principles such as machine numbers, rounding errors, condition numbers, and complexity are also included.

Audience: This book is intended for students, teachers, and researchers in scientific computing, computer science, and applied mathematics. It is also an entertaining and motivating introduction for those with a minimum background in mathematics or computer science. Bits and Bugs can be used for courses in numerical analysis, numerical methods in linear algebra/ODEs/PDEs, introductory software engineering, introductory scientific computing, and parallel programming.

3 Other Announcements

3.1 Margaret Wright

Margaret Wright, professor of Computer Science and Mathematics at the Courant Institute of Mathematical Sciences, New York, won the John Von Neumann prize, the highest honor bestowed by SIAM, for her contributions to the numerical solution of optimization problems. Margaret received this prestigious award in July 2019, at the ICIAM conference in Valencia, Spain, where she also delivered the flagship lecture "A Hungarian Feast of Applied Mathematics". Congratulations Margaret!

Chair's Column

Welcome to the October 2019 edition of our SIAG/OPT Views and News. This short note provides an update about the most important developments since our last newsletter, and provides updates about preparations for the the SIAM/Opt'20 conference.

We made great progress towards our signature conference SIAM/Opt'20, May 26-29, 2020 in Hong Kong. The organizing committee recruited a spectacular list of Plenary and Tutorial Speakers. The Plenary Speakers are: Amir Ali Ahmadi, Princeton U.; Donald Goldfarb, Columbia U.; Didier Henrion, U. of Toulouse, France; Michael Hinze, U. of Koblenz-Landau, Germany; Satoru Iwata, U. of Tokyo, Japan; Ivana Ljubic, ESSEC Business School, France; R. Tyrrell Rockafellar, U. of Washington, USA; Margaret Wiecek, Clemson U.; and the two Minitutorials cover two hot topic areas:

I. Quantum Computing and Optimization, Speaker: Giacomo Nannicini, IBM, USA;

II. AI, Machine Learning and Optimization, Speaker: Lin Xiao, Microsoft Research, USA.

Now the next steps are to submit mini-symposium proposals, and submit abstracts for mini-symposia and for contributed sessions. Please check out the SIAM/Opt'20 web site https://www.siam.org/Conferences/CM/Conference/op20 regularly for news and updates. Feel free to contact me or the Local Organizing Committee Co-Chairs, Xiaojun Chen and Defeng Sun, if you have any questions or suggestions.

SIAG/OPT has now two major awards to recognize the exceptional research our members conduct. Thus, an important activity to all of you is to nominate outstanding papers for the SIAG/OPT Best Paper Prize and nominate our emerging colleagues for the inaugural SIAG/OPT Early Career Prize through October 15, 2019. This is also the time to nominate great optimizers for the SIAM prizes, and most importantly for SIAM Fellow.

I would also like to call your attentions to the upcoming SIAG/Opt elections. The Nominating Committee, Chaired by Jorge Moré, recruited great candidates to choose from. Do participate in this process to elect the next leadership team for SIAG/Opt.

This is also the proper place to remind all readers of this newsletter that the time arrived to renew your SIAM, and SIAG/Opt membership to strengthen our activity group and ensure your continued membership in our great community.

This newsletter is the last one that our Views and News Editor Jennifer Erway (Wake Forest University) assembles together with Pietro Belotti (FICO). After 5 years of dedicated service, Jennifer is taking on other responsibilities. Jennifer did an outstanding job for years as editor of SIAG/OPT Views and News. Our thanks go out to both Jennifer and Pietro for producing highly informative, excellent newsletters. Pietro is continuing as Editor, and the search for the new co-editor just ended successfully—see the next section for details.

This is the last newsletter in which I am writing to you as Chair of SIAG/Opt. It is my honor and great privilege to serve the SIAM optimization community in this important function. The next half year will be filled with preparation for the SIAM/Opt'20 conference, and to ensure smooth transition.

With this in mind, I wish all of you a highly successful completion of the year 2019! Looking forward to see you all in Hong Kong at our tri-annual conference SIAM/Opt'20, in May 2020.

Tamás Terlaky, SIAG/OPT Chair
Department of Industrial and Systems Engineering, P.C. Rossin College of Engineering and Applied Science, Lehigh University, Bethlehem, PA 18015-1582, USA,
terlaky@lehigh.edu, http://www.lehigh.edu/~tat208

Comments from the Editors

This issue of *SIAG/OPT Views and News* brings us some fresh news from the Traveling Salesman Problem (TSP) world: Jakub Tarnawski and Vera Traub summarize the state of the art in the field and discuss the stunning new result on the constant factor approximation algorithm for the Asymmetric TSP. We also have two contributions about two great optimizers who passed away this year: Egon Balas, whose pioneering work laid the foundations of many of modern methods in integer optimization, and Andrew Conn, who is well-known for his influential work in many areas of optimization, but especially continuous optimization.

We are happy to welcome a new addition to the editorial team: Somayeh Moazeni from the Stevens Institute's Business School. Somayeh's main research interests range from Continuous Optimization to Stochastic Dynamic Optimization, to Algorithmic Trading and Data Analytics. She fills in for Jennifer, who will leave after publication of this issue.

Last but not least, please remember that all 27 volumes of *Views and News* are available at the online archive.

As always, the editors welcome your feedback at siagoptnews@lists.mcs.anl.gov. Suggestions for new issues, comments, and papers are always welcome!

Pietro Belotti, Editor
Xpress Optimizer development team, Fair Isaac, Italy,
pietrobelotti@fico.com

Jennifer Erway, Editor
Department of Mathematics, Wake Forest University, USA,
erwayjb@wfu.edu, http://www.wfu.edu/~erwayjb