On the Computational Complexity of Reoptimization

2 Scientific Work

2.1 Summary

In algorithmics and operations research, we deal with many optimization problems whose solutions are of importance in everyday applications. Unfortunately, most of these problems are computationally hard, and so we use different approaches like heuristics, approximation algorithms, and randomization in order to compute good (not necessarily optimal) solutions. Nevertheless, we are often not even able to give any reasonable guarantee on the solution quality or on the efficiency of the applied algorithm. How to proceed in this difficult situation is the core of current algorithmics. Our idea is to consider problem instances not as isolated tasks, but to use our acquired experience. This leads to the following approach: Don't start from scratch when confronted with a problem, but try to make good use of prior knowledge about similar problem instances whenever they are available. Traditionally, there is no such prior knowledge, because problem inputs are considered as isolated instances. In reality, prior knowledge is often at our disposal, because a problem instance can arise from a small modification of a previous problem instance. As an example, imagine that an optimal timetable (for some objective function, under some constraints) for a given railway network is known, and that now a railway station is closed down. It is intuitively obvious that we should profit somehow from the old timetable when we try to find a new timetable. It is this general idea that we pursue in some generality: Given a problem instance with an optimal (or approximate) solution, and a variation of the problem instance that we obtain through small, local modifications, what can we learn about the new solution? Does the old solution help at all? Under what circumstances does it help? How much does it help? How much does it help for the runtime, how much for the quality of the output?

Our first concrete goal is to investigate the hardness of solving locally modified problem instances when an optimal (or a good) solution to the original instance is for free. In this scenario, we also speak about reoptimization. Our initial results (to be discussed in more detail later) show that this additional information helps to get solutions with better quality guarantees for some problems and does not help to find the solution more easily for other problems. The core of our project is to investigate which algorithmic problems become easier when moving from the classical problem formulation to the reoptimization version, and which don't. Our hopes should not be unrealistic: In general, NP-hardness of a problem obviously implies that the solution to a locally modified problem cannot be found in polynomial time, given the solution of the original problem. But it is, on the other hand, obviously a drastic advantage to know a solution of a problem instance, as compared to not knowing it, and this is reflected in the fact that verification for problems in NP is polynomial, while finding a solution is not (unless P=NP). For polynomial problems, such as maintaining a substructure in a graph under local changes (e.g., edge weight changes, insertions, deletions), there are plenty of positive research results, i.e., success stories, but also failures. Note that, even in the case where the reoptimization problem remains NP-hard, this does not mean that the additional knowledge of a solution to a similar problem instance does not help. For instance, it might be possible to guarantee a better approximation ratio for the reoptimization variant than for the original problem. Thus, one has to investigate carefully in which way the experience with solving similar problem instances might be helpful. In this project, we limit ourselves to hard problems; amazingly, this natural perspective went largely unnoticed so far. Our study aims at contributing to the understanding of the computational hardness of locally modified problems, and we furthermore aim at developing new algorithms for solving algorithmic tasks that are not only efficient, but also provide reasonable guarantees on the solution quality.

A rapid and dedicated continuation of our work is now essential, because the general idea of problem solving through repeated modification seems to be in the air, and our competitive advantage will not persist if we lose momentum now. Amazingly, this idea is even reflected on a somewhat higher level of generality in the idea of solving a broad variety of problems through an "iterated" approach: In [G06], four solutions to very important problems (a polynomial time approximation of the permanent of non-negative matrices; a construction of expander graphs; a log-space algorithm for undirected connectivity; an alternative proof of the PCP theorem) are found to start from a trivial construct, apply an ingeniously designed sequence of iterations that yields the desired result by modifying the construct in a moderate manner at each step.

2.2 Research Plan

2.2.1 State of the Art

Given an instance of an optimization problem together with an optimal solution, we consider the scenario in which the instance is modified locally. In graph problems, e.g. an edge cost might be varied or a single node or edge may be removed or added, etc. For a problem U and a local modification lm we denote the corresponding reoptimization problem by lm–U. Obviously, lm–U may be easier than U because we have the optimal solution for the original problem instance for free. A related research question was considered in operations research [Gre98, Lib91, LPSV98, SLG95, VW99], where one studies how much a given instance of an optimization problem may be varied if it is desired that optimal solutions to the original instance retain their optimality. In contrast with this so called "postoptimality analysis", we allow also modifications causing the loss of the optimality of the solution to the original instance for the modified instance and look for efficient algorithms for computing new high quality solutions.

Our first results [BFH+06⋆, BFH+07⋆] (a ⋆ denotes results of our own) in investigating the hardness of locally modified instances are devoted to different versions of TSP. For the classical TSP, we regard a change of the cost of exactly one edge as a local modification. For TSP with deadlines, we regard a shift of one deadline by the amount of at least one time unit as a local modification, too. Denoting the metric TSP by Δ-TSP, we use the notation Δ_β-TSP for the special case of TSP where all instances satisfy the β-triangle inequality

$$\mathrm{cost}(\{x,z\}) \leq \beta \cdot (\mathrm{cost}(\{x,y\}) + \mathrm{cost}(\{y,z\}))$$

for all vertices x, y, and z. If $1/2 < \beta < 1$, we call this the strengthened triangle inequality. If $\beta > 1$, we call it the relaxed triangle inequality.

Our first results show on the one hand that the knowledge of an optimal solution for the original problem instance can be helpful for solving the modified instance, and that on the other hand for some problems, the original problem U is as hard as the reoptimization version from the point of view of the hardness of worst-case approximability. We list the main results of our initial study of reoptimization versions of hard problems:

(i) In terms of the worst-case analysis, lm-TSP is as hard as TSP because lm-TSP is not approximable in polynomial time with a polynomial approximation ratio (unless P=NP). Additionally, lm–Δ_β-TSP is NP-hard for all $\beta > 1/2$.

(ii) For many years, Christofides algorithm [Chr76] has provided the best known guarantee of 1.5 on the approximation ratio for Δ-TSP. Here, we designed an efficient algorithm for lm–Δ-TSP with a worst-case approximation ratio of 1.4. Additionally, we showed for all $\beta < 3.348$ that one can reach a better approximation ratio for lm–Δ_β-TSP than the best known approximation guarantees for Δ_β-TSP. Note that Δ_β-TSP for $\beta \leq 3$ contains almost all problem instances of TSP appearing in applications and made public in different databases.

(iii) In [BHKK06a⋆, BHKK06b⋆], we showed a lower bound of $\Omega(n)$ for the approximability of the TSP with deadlines or windows. Considering the very restricted local modification of changing a single deadline by one time unit, we showed in [BFH+06⋆, BFH+07⋆] that the reoptimization version of these general TSP problems does not become easier, i.e., the same optimal lower bound $\Omega(n)$ on its polynomial-time approximability can be derived.

(iv) In [BHKK06a⋆, BHKK06b⋆], the lower bound of $2-\varepsilon$, for an $\varepsilon > 0$, on the approximability of Δ-TSP with constantly many (a fixed number of) deadlines was proved. Surprisingly, we got the same lower bound for the reoptimization version of this problem [BFH+06⋆].

Note that, for some optimization problems, knowing an optimal solution to the original instance trivially makes their local-modification versions easy to solve because the given optimal solution is itself a very good approximate solution to the modified instance. For example, adding an edge in the instance of a coloring problem can increase the cost of an optimal solution at most by one. But such obvious cases are not a matter of our investigation. They are only useful for illustrating the possible different outcomes of this study. On the one hand, lm–U may become very easy relative to a problem U and on the other hand, lm–U may be as hard as U.

The complexity of reoptimization instances of optimization problems has been studied also by others [ABS03, AEM+06, EMP07]. Similar to our work, there TSP, metric TSP, and MAX-TSP (maximizing the length of a Hamiltonian tour) as well as the Steiner tree problem have been studied. Although the local modifications considered in [ABS03, AEM+06, EMP07] are node insertions and deletions, and hence are different from the local modifications mentioned above, the results are quite similar. The reoptimization problem remains NP-hard, but one can achieve better constant approximation ratios for metric TSP. The main result in [AEM+06] that is significantly different from others gives a PTAS, i.e., an approximation that can be forced to be arbitrarily close to optimum. Furthermore, for MAX-TSP, there is no PTAS, unless P=NP, and the reoptimization of metric MAX-TSP is provably easier than the original problem.

2.2.2 Current State of Our Research

In Section 2.2.1 we already presented our initial results [BFH+06⋆, BHKK06b⋆, BFH+07⋆] in the proposed research direction. A significant part of our previous research was devoted to the study of the hardness of optimization problems and the design of algorithms for a variety of hard problems. For instance, we developed the concept of stability of approximation [Hro99⋆, BHK+00a⋆, BHS07⋆, BH07⋆] and applied it for TSP [BHK+00a⋆, BHK+00b⋆, BS00⋆, Hro03⋆, BHK+00⋆, BHK+02⋆, FHP+05⋆] and for graph connectivity problems [BBH+02⋆, BBH+03⋆, BBH+04⋆]. We recently studied biologically motivated optimization problems in [BB03⋆, BB04⋆, BB07a⋆, BB07b⋆], and optimization problems in graphs that arise from practical applications of various types in [CEH+04, EJM+04, GGJ+04, EJM+05, EW03]. All this experience can be useful for this project because for several of these problems the local modifications are very natural.

We started to work intensively on the topic of reoptimization in the framework of the COST 293 (GRAAL) project funded by the European Union which finishes in summer 2008. This proposed project has to continue the work begun within the COST 293 project, where we achieved the following results:

For the metric TSP with decreasing the cost of one edge as a local modification, we found a characterization of those input instances on which the reoptimization problem is hardest to approximate [KM07⋆]. This approach might lead to further improvement of the approximation ratio beyond the upper bound of 1.4 as shown in [BFH+06⋆, BFH+07⋆].

We further considered the Steiner tree problem in weighted graphs, where the objective is to find a minimum-weight subtree of a given graph spanning a given subset of vertices, called terminals. We focussed on four different types of local modifications, namely increasing or decreasing the terminal set by one vertex (without changing the underlying graph) and decreasing or increasing the weight of one edge. Our main results on the resulting reoptimization problems are as follows:

(i) All four variants are strongly NP-hard [BHK+08⋆, BBH+08⋆].

(ii) For all four variants, we can improve on the best known approximation ratio for the classical Steiner tree problem in graphs of 1.55 [RZ00]. In particular, for adding or removing a terminal, there exists a 1.5-approximation algorithm [BHK+08⋆], and even a 1.408-approximation and a 1.34-approximation exist, respectively [BBH+08⋆]. For increasing and decreasing the weight of an edge, there are algorithms achieving an approximation ratio of 4/3 and 1.3, respectively [BBH+08⋆].

(iii) For the subproblem where the edge weights are restricted to values from $\{1, 2, \ldots, k\}$ for some constant k, there is a PTAS for the reoptimization problem based on adding or deleting a terminals [BHK+08⋆], although the classical (non-reoptimization) problem is APX-hard even for $k = 2$ [BP89].

A survey on parts of the above-mentioned results can also be found in [BHMW08⋆].

2.2.3 Detailed Research Plan

If we want to solve practical problem instances by hand, it is very useful to have experience with similar instances. Thus far, exploiting prior knowledge in the process of solving similar problem instances is not accounted for in the definition of the computational hardness of algorithmic problems. To consider the local-modification version of problems is a step in looking for possible contributions of experience and knowledge in the process of problem solving. There is a variety of possibilities how to measure a possible gain of applying the knowledge of optimal or high quality solutions to similar problem instances. Because this is the crucial point of our concept, we structure our research plan with respect to saving computational resources and deriving guarantees on solution quality by exploiting our knowledge about solutions to similar problem instances.

A. Approximation guarantees

(a) Similarly as in our previous contributions, we consider optimization problems and their reoptimization versions. The main question is whether the knowledge of an optimal solution to the original instance can help to decrease the achievable approximation ratio, i.e., to improve the guarantee of the solution quality. Here, we have to consider also problems different from TSP. For instance, a lot of investigations in bioinformatics are based on recognizing similarities that are used in the search of functionalities of studied biological objects. The scenario of local changes is also typical in the area of communication networks, where capacity of connections may increase or some connections may fail. All of our previous work in this area is concerned with local modifications to instances for which an optimum solution can be obtained in polynomial time [NPW03a, NPW03b, NPW04, FPP+05], and we achieve our results in all cases by making heavy use of the structure of optimum solutions. In this project, we plan to extend this view to NP-hard optimization problems in graphs. In this situation, one can be very satisfied if the concept of reoptimization allows the shift from a classical APX-hard optimization problem to a reoptimization version which allows for a PTAS [AEM+06, BHK+08⋆] or simply for a better guarantee on the approximation ratio.

A limitation of the investigation of approximability of optimization problems is often in the concept of measuring the hardness simply in the worst-case manner across all problem instances, regardless of problem parameters that are suited to capture features of a problem instance more accurately. This was the reason to introduce the concept of stability of approximation [Hro99⋆] that enables to study the approximability as a function of a problem instance parameter. This can be helpful as we have seen in [BFH+06⋆], where we showed that lm–TSP is as hard as TSP from the worst-case point of view; but by parameterizing TSP with respect to the distance to the metric (β-triangle inequality) we were able to improve the approximation ratio for lm–Δ_β-TSP. In that way, this approach can be useful in approaching a better understanding of the nature of hardness of optimization problems.

(b) The drawback of the assumption of having an optimal solution to the original problem instance is that one cannot apply the developed algorithms iteratively. One application of such an algorithm explores the knowledge of an optimal solution to the original instance in order to get a good approximation of an optimal solution for the modified instance. In the case one modifies the already modified instance I_1 again to an instance I_2, we are unable to use our algorithm, because we do not have an optimal solution to I_1. This scenario requires to generalize our approach. Instead of assuming to know an optimal solution to the original instance we consider to have a good approximation of it only. The main question here is, which approximability is efficiently reachable for the modified instance. The size of the increase of the approximation ratio guaranteed for the modified instance determines the number of local modifications one can efficiently handle.

B. Linear-time Approximability In A, we mainly focused on approximability in polynomial time. In applications, we often need to react very fast when a local change appears. Thus, one can ask for very fast algorithms for handling local modifications. In this setting we focus on both - efficiency and approximation ratio at once. Even if the approximation ratio remains the same, moving from e.g. cubic time for solving the original instance to linear time for solving modified instances is considered as a gain. This approach of investigating a possible speed-up of the process of problem solving can be combined with the concept of stability and iterative modifications as proposed in A.

C. Exact Algorithms In the case of exact algorithms, the idea is to produce an exact or an optimum solution even if the consequence is that the designed algorithms work in the worst-case in exponential time. Typically this may result in algorithms working in c^n time for $c < 2$ or in exponential parametrized algorithms working efficiently for some subclasses of problem instances with small parameter values. The size of c or the growth of the function f and the polynomial p in the complexity $f(d) \cdot p(n)$ of a parameterized algorithm for input size n and a parameter d are crucial for the applicability of such algorithms. The question posed here is whether we can decrease c or the growth of f and p when dealing with lm–U instead of U. There are numerous involved exact algorithms developed and their development recently attracts more and more research interest in algorithmics. Investigating local modifications in this framework can be very fruitful, and so this topic alone provides enough important questions for a doctoral study.

D. Heuristics We use the term heuristics for algorithms for which nobody is able to give a good guarantee on the efficiency or on the solution quality, but which behave well on average. For heuristic approaches such as branch-and-bound or local search one can save a lot of time when dealing with lm–U instead of solving a problem U. In this part, experimental work is necessary and many problem formulations are suitable also for involving students at the Master level.

2.2.4 Timetable

We plan to work in all four main streams A, B, C, and D presented. We consider it essential to have two doctoral students work together from the very start, for the sake of exchanging ideas, as well as to specialize in the subfields that we described. The research has to be performed in cooperation of the research groups of Peter Widmayer and Juraj Hromkovič. In the first year, both doctoral students will start with problems A(a), focussing on classical optimization problems and especially their weighted counterparts, for which one can consider the changing of weights as natural local modifications. After achieving reasonable progress with respect to the approximation ratio guarantees of reoptimization problems, we consider to deepen the study on the hardness issues as described in A.b) for the most promising from the point of view of achieved approximation quality. In the third year, we plan to work more intensively on the streams B, C, and D.

2.2.5 Importance of This Work

Many problems occuring in the field of logistics and operations research can be formulated as discrete optimization problems. Many of these problems are computationally hard, common approaches for solving them are often based on heuristics, and little is known about the quality of these approaches. But, in many everyday applications, it is an important issue to obtain solutions of provable high-quality. Several approaches like approximation algorithms, randomized algorithms, or parameterized algorithms have been devised to meet these needs. For different problems, different of these approaches have been proven useful, but there still are many problems which can not be treated in a satisfactory way by using the existing tools.

Understanding the effect that small, local changes of problem instances have on the solutions of discrete optimization problems can help to achieve better solutions more efficiently. In this way, our approach might provide a novel way of attacking hard problems.

References

[ABS03] C. Archetti, L. Bertazzi, M. G. Speranza: Reoptimizing the traveling salesman problem. *Networks* 42, 2003, pp. 154–159.

[AEM+06] G. Ausiello and B. Escoffier and J. Monnot and V. Th. Paschos: Reoptimization of Minimum and Maximum Traveling Salesman's Tours. *Proc. of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006)*, Springer LNCS 4059, 2006, pp. 196–207.

[BP89] M. W. Bern, P. E. Plassmann: The Steiner tree problem with edge lengths 1 and 2. *Information Processing Letters* 32(4), 1989, pp. 171–176.

[BBH+08⋆] D. Bilo, H.-J. Böckenhauer, J. Hromkovič, R. Královič, T. Mömke, P. Widmayer, A. Zych: Reoptimization of Steiner trees. Submitted.

[BB03⋆] H.-J. Böckenhauer, D. Bongartz: *Algorithmische Grundlagen der Bioinformatik*. Teubner, 2003.

[BB04⋆] H.-J. Böckenhauer, D. Bongartz: Protein folding in the HP model on grid lattices with diagonals. *Discrete Applied Mathematics* 155(2), 2007, pp. 230–256. Extended abstract in: *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS 2004)*, Springer LNCS 3153, 2004, pp. 227–238.

[BB07a⋆] H.-J. Böckenhauer, D. Bongartz: *Algorithmic Aspects of Bioinformatics*. Springer Verlag, 2007.

[BB07b⋆] H.-J. Böckenhauer, D. Bongartz: A weighted HP model for protein folding with diagonal contacts. *RAIRO Theoretical Informatics and Applications* 41, 2007, pp. 375–402.

[BBH+02⋆] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, W. Unger: On the hardness of constructing minimal 2-connected spanning subgraphs in complete graphs with sharpened triangle inequality. *Proceedings of the 22nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2002)*, Springer LNCS 2556, 2002, pp. 59–70.

[BBH+03⋆] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, W. Unger: On k-edge-connectivity problems with sharpened triangle inequality. *Proceedings of the 5th Italian Conference on Algorithms and Complexity (CIAC 2003)*, Springer LNCS 2653, 2003, pp. 189–200.

[BBH+04⋆] H.-J. Böckenhauer, D. Bongartz, J. Hromkovič, R. Klasing, G. Proietti, S. Seibert, W. Unger: On the hardness of constructing minimal 2-connected spanning subgraphs in complete graphs with sharpened triangle inequality. *Theoretical Computer Science* 326(1–3), 2004, pp. 137–153.

[BFH+06⋆] H.-J. Böckenhauer, L. Forlizzi, J. Hromkovič, J. Kneis, J. Kupke, G. Proietti, P. Widmayer: Reusing Optimal TSP Solutions for Locally Modified Input Instances (Extended Abstract). *Proc. of the 4th IFIP International Conference on Theoretical Computer Science (IFIP TCS 2006)*, Springer, 2006, pp. 251–270.

[BFH+07⋆] H.-J. Böckenhauer, L. Forlizzi, J. Hromkovič, J. Kneis, J. Kupke, G. Proietti, P. Widmayer: On the approximability of TSP on local modifications of optimally solved instances. *Algorithmic Operations Research* 2(2), 2007, pp. 83–93.

[BH07⋆] H.-J. Böckenhauer, J. Hromkovič: Stability of approximation algorithms or parameterization of the approximation ratio. *Proc. of the 9th International Symposium on Operations Research in Slovenia (SOR 07)*, Slovenian Society Informatika - Section for Operational Research, 2007, pp. 23–28.

[BHK+00⋆] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, W. Unger: Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem (extended abstract). *Proceedings of the 4th Italian Conference on Algorithms and Complexity (CIAC 2000)*,Springer LNCS 1767, 2000, pp. 72–86.

[BHK+00a⋆] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, W. Unger: Approximation algorithms for TSP with sharpened triangle inequality. *Information Processing Letters* 75, 2000, pp. 133–138.

[BHK+00b⋆] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, W. Unger: An improved lower bound on the approximability of metric TSP and approximation algorithms for the TSP with sharpened triangle inequality (extended abstract). *Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science (STACS 2000)*, Springer LNCS 1770, 2000, pp.3̃82–394.

[BHK+02⋆] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, W. Unger: Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem. *Theoretical Computer Science* 285, 2002, pp. 3–24.

[BHKK06a⋆] H.-J. Böckenhauer, J. Hromkovič, J. Kneis, J. Kupke: On the parameterized approximability of TSP with deadlines. *Theory of Computing Systems* 41(3), 2007, pp. 431–444.

[BHKK06b⋆] H.-J. Böckenhauer, J. Hromkovič, J. Kneis, J. Kupke: On the approximation hardness of some generalizations of TSP. *Proc. of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006)*, Springer LNCS 4059, 2006, pp.1̃84–195.

[BHK+08⋆] H.-J. Böckenhauer, J. Hromkovič, R. Královič, T. Mömke, P. Rossmanith: Reoptimization of Steiner trees: changing the terminal set. Submitted to *Theoretical Computer Science*.

[BHMW08⋆] H.-J. Böckenhauer, J. Hromkovič, T. Mömke, P. Widmayer: On the hardness of reoptimization. *Proc. of the 34th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2008)*, LNCS 4910, Springer Verlag 2008, pp. 50–65.

[BHS07⋆] H.-J. Böckenhauer, J. Hromkovič, S. Seibert: Stability of approximation. In: T. F. Gonzalez (ed.): *Handbook of Approximation Algorithms and Metaheuristics*, Chapman & Hall/CRC, 2007, Chapter 31.

[BS00⋆] H.-J. Böckenhauer, S. Seibert: Improved lower bounds on the approximability of the traveling salesman problem. *RAIRO Theoretical Informatics and Applications* 34, 2000, pp. 213–255.

[Chr76] N. Christofides: Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, 1976.

[CEH+04] M. Cieliebak, T. Erlebach, F. Hennecke, B. Weber, P. Widmayer: Scheduling with release times and deadlines on a minimum number of machines. *Proc. of the 3rd IFIP International Conference on Theoretical Computer Science (IFIP TCS 2004)*, Springer, 2004, pp. 209–222.

[CDD+01] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, F. Soumis: VRP with time windows. In: P. Toth, D. Vigo (eds.): *The Vehicle Routing Problem*, SIAM 2001, pp. 157–193.

[EW03] S. Eidenbenz, P. Widmayer: An approximation algorithm for minimum convex cover with logarithmic performance guarantee. *SIAM Journal on Computing* 32, 2003, pp. 654–670.

[EJM+04] T. Erlebach, R. Jacob, M. Mihalák, M. Nunkesser, G. Szabó, P. Widmayer: An algorithmic view on OVSF code assignment. *Proc. of the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, Springer LNCS 2996, 2004, pp. 270–281.

[EJM+05] T. Erlebach, R. Jacob, M. Mihalák, M. Nunkesser, G. Szabó, P. Widmayer: Joint base station scheduling. *Proc. of the 2nd International Workshop on Approximation and Online Algorithms (WAOA 2004)*, Revised Selected Papers, Springer LNCS 3351, 2005, pp. 225–238.

[EMP07] B. Escoffier, M. Milanič, V. Th. Paschos: Simple and fast reoptimizations for the Steiner tree problem. DIMACS Technical Report 2007-01.

[FPP+05] P. Flocchini, L. Pagli, G. Prencipe, N. Santoro, P. Widmayer, T. Zuva: Computing all the best swap edges distributively. *Proc. of the 8th International Conference on Principles of Distributed Systems (OPODIS 2004)*, Revised Selected Papers, Springer LNCS 3544, 2005, pp. 154–168.

[FHP+05⋆] L. Forlizzi, J. Hromkovič, G. Proietti, S. Seibert: On the stability of approximation for Hamiltonian path problems. *Algorithmic Operations Research* 1(1), 2006, pp. 31–45.

[GGJ+04] M. Gatto, B. Glaus, R. Jacob, L. Peeters, P. Widmayer: Railway delay management: exploring its algorithmic complexity. *Proc. of the 9th Scandinavian Workshop on Algorithm Theory (SWAT 2004)*, Springer LNCS 3111, 2004, pp. 199–211.

[Gre98] H. Greenberg: An annotated bibliography for post-solution analysis in mixed integer and combinatorial optimization. In: D. L. Woodruff (ed.): *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search*, Kluwer Academic Publishers, 1998, pp. 97–148.

[G06] O. Goldreich: Bravely, moderately - A common theme in four recent works. *SIGACT News* 37, ACM, 2006, pp. 31–46.

[Hro99⋆] J. Hromkovič: Stability of approximation algorithms for hard optimization problems. *Proc. SOFSEM'99*, Springer LNCS 1725, 1999, pp. 29–47.

[Hro03⋆] J. Hromkovič: *Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics*. Springer 2003.

[KM07⋆] R. Královič, T. Mömke: Approximation hardness of the traveling salesman reoptimization problem. *Proc. of the 3rd Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS 2007)*, pp. 97–104.

[Lib91] M. Libura: Sensitivity analysis for minimum Hamiltonian path and traveling salesman problems. *Discrete Applied Mathematics* 30, 1991, pp. 197–211.

[LPSV98] M. Libura, E. S. van der Poort, G. Sierksma, J. A. A. van der Veen: Stability aspects of the traveling salesman problem based on k-best solutions. *Discrete Applied Mathematics* 87, 1998, pp. 159–185.

[NPW03a] E. Nardelli, G. Proietti, P. Widmayer: Swapping a failing edge of a single source shortest paths tree is good and fast. *Algorithmica* 35, 2003, pp. 56–74.

[NPW03b] E. Nardelli, G. Proietti, P. Widmayer: Finding the most vital node of a shortest path. *Theoretical Computer Science* 296, 2003, pp. 167–177.

[NPW04] E. Nardelli, G. Proietti, P. Widmayer: Nearly linear time minimum spanning tree maintenance for transient node failures. *Algorithmica* 40, 2004, pp. 119–132.

[RZ00] G. Robins, A. Zelikovsky: Improved Steiner tree approximation in graphs. *Proc. of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, ACM, 2000, pp. 770–779.

[SLG95] Y. N. Sotskov, V. K. Leontev, E. N. Gordeev: Some concepts of stability analysis in combinatorial optimization. *Discrete Applied Mathematics* 58, 1995, pp. 169–190.

[VW99] S. Van Hoesel, A. Wagelmans: On the complexity of postoptimality analysis of 0/1 programs. *Discrete Applied Mathematics* 91, 1999, pp. 251–263.