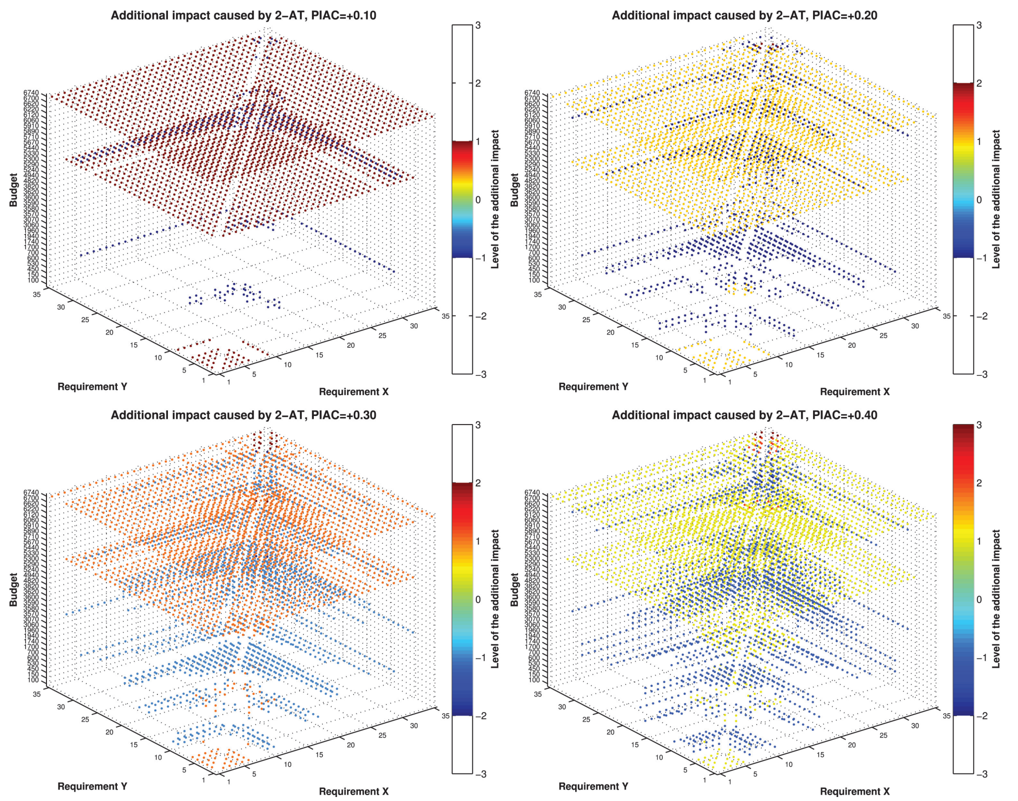


Fig. 12. Additional impact of second-order effects for positive PIAC (underestimated cost).

For the Motorola dataset, we have also seen that the reasons for sensitivity are far from obvious without the OATSAC sensitivity heat map. The subtlety of sensitivity is a motivation for the automated decision support approach advocated in this article. The Motorola dataset is a modest requirements problem, with only 35 different requirement choices. Though we make no claims about results that may be obtained for other requirement problems, it was clear that there were several subtleties revealed by our analysis, even in this relatively modest requirements set. We can therefore have measured confidence that there will be other interesting sensitivities elsewhere too and that these may lead to similarly actionable findings.

## 9. THREATS TO VALIDITY

Threats to internal validity concerns the factors that could have affected the observations made during the experimental evaluations. Any experimental evaluation of scalability is not free from environmental issues. The execution time was measured using Unix `time` utility. In order to control factors that could perturb our measures, the experiment was performed only after we ensured that pad peaks and other memory intensive tasks could be avoided as much as possible. We have repeated the experiment up to three times depending on the existence of outliers. Threats to external validity concerns the factors that prevent the generalization of the results. The scalability observed in the experimental evaluation clearly only applies to the specific choice of the problem definition and the algorithm. The same level of scalability may not be easily achievable for different classes of problems or exact algorithms. However, the NRP

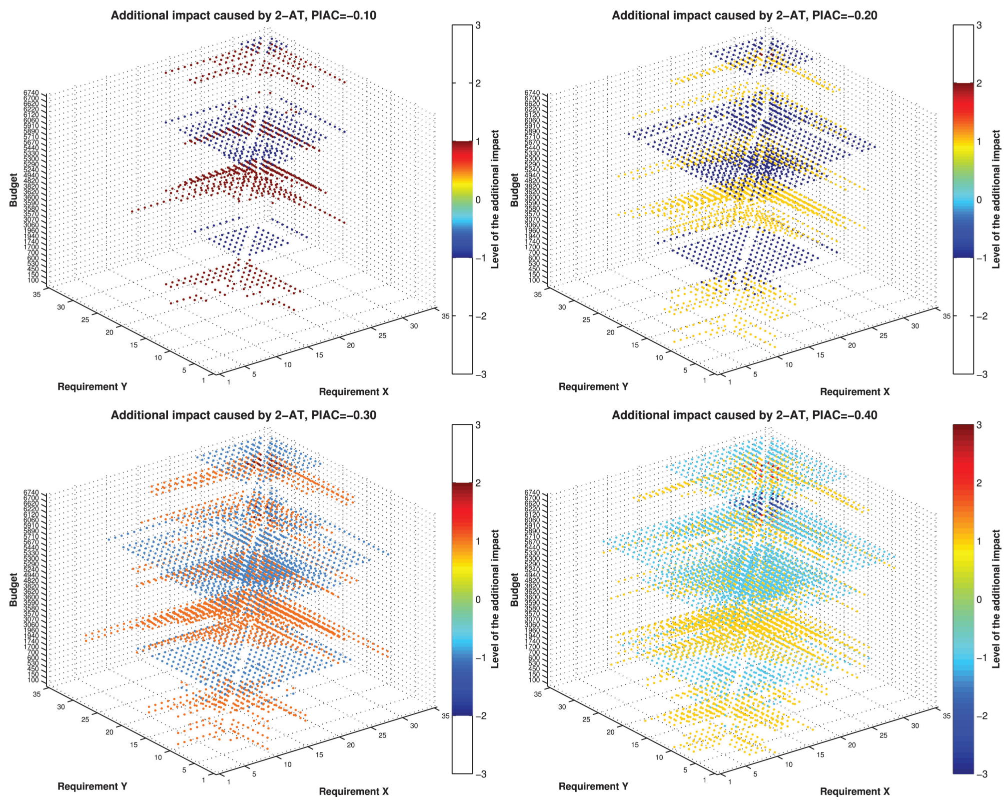


Fig. 13. Additional impact of second-order effects for negative PIAC (overestimated cost).

formulation used in the article has been widely studied [Bagnall et al. 2001; van den Akker et al. 2004, 2005; Zhang et al. 2008]: our experimentally results provide supporting evidence that there exists at least one exact algorithm, which is scalable, for a well-known formulation of NRP. The results for the Motorola dataset give an indication that the work can provide insights useful to a decision maker, but they constitute only existential evidence that this can occur in practice they do not guarantee that in all cases it will be possible to gain such insights. In this way, the results are case study based and we cannot generalize from them. It should also be noted that the Motorola dataset contains only a trivial set of dependencies; where there are more elaborate dependencies, these will also need to be taken into account, and this will affect the formulation of the problem.

Threats to construct validity concerns whether the measurement we made represent the actual problem. Real estimation errors in NRP can involve more than one requirement at a time, which would require more complicated modeling of the issue. However, we base our experiment on a widely studied and accepted sensitivity analysis technique (one-at-a-time) [Saltelli et al. 2000]. In any case, an exhaustive sensitivity analysis where an arbitrary number of requirements can be perturbed at the same time involves an exponential number of instances. Unfortunately, any method reducing this number could leave aside a sensitive instance.

We used 15 different problem sizes ranging from 100 to 1500 requirements with steps of 100 and correlation factors ranging from 0% to 100% in 5% steps. This resulted in 315 problem configurations, but we cannot claim that these are necessarily representative.

Of course, further experimentation with alternative settings is always possible, and we cannot rule out the possibility that such experiments might yield different results. Though we can be more sure about performance between the values of the settings we chose, we can say nothing about scalability beyond 1,500 requirements. In many cases, such a large number of requirements would be sufficient, but should applications develop that required scalability beyond these numbers, then further experimentation would be required.

## 10. RELATED WORK

Bagnall et al. [2001] coined the term 'Next Release Problem (NRP)', formulating the consideration of the requirements for a software release as a search-based *selection* problem. However, there had been previous work on the application of optimization techniques for requirements *prioritization* [Karlsson et al. 1998]. The problem is also known as Software Release Planning [Ruhe and Greer 2003; Greer and Ruhe 2004]. There has been a recent overview of the area [Zhang et al. 2008] as well as a detailed survey of Search-Based Software Engineering (SBSE) techniques, which includes a section on the NRP [Harman et al. 2012b]. The work by Bagnall et al. [2001] is one of the few papers to consider an exact algorithm for the NRP. They use a standard integer programming formulation and present results from its implementation in the popular tool CPLEX [Bixby et al. 2000]. However, they do not present any results regarding the scalability of their approach.

While CPLEX is a widely used and robust tool, van den Akker et al. [2004, 2005] add various 'managerial steering inputs' to a basic ILP formulation of the NRP to provide greater flexibility. Li et al. [2007] present two integer linear programming models. The first of these is concerned with project management, seeking to reduce project completion time. This is a separate problem from the requirements analysis problem and one that has been widely studied elsewhere in the literature on SBSE for project management [Alba and Chicano 2007; Alvarez-Valdés et al. 2006; Antoniol et al. 2004, 2005; Chao et al. 1993; Kapur et al. 2008]. The second model integrates 'on time delivery' with maximal revenue generation. The article reports the results of experiments with both approaches on synthetic data.

No exact algorithm has been used in the literature on the NRP for which real-world problems were used in its evaluation. Indeed, most of the previous work on both real and synthetic data has concerned meta-heuristic algorithms which, though flexible and popular, are inexact and therefore cannot be used for a fully reliable sensitivity analysis.

Our approach to RSA uses Nemhauser–Ullmann's (NU) algorithm to solve multiple instances of the NRP. NU can be regarded as a smart rendition of dynamic programming by costs. This algorithm first appeared in the solution of capital allocation problems [Nemhauser and Ullmann 1969]. Many researchers had noticed that the algorithm seemed to behave well in practice when solving random KP instances, but a theoretical justification of these observations was lacking until groundbreaking work by Beier and Vöcking [2004] provided an explanation of the behavior of this algorithm in the average case under quite general conditions.

Karlsson et al. [1998] uses the Analytical Hierarchy Process (AHP) which allows for human contributions to the choice of ranking. This approach has been implemented in the Focal Point requirements analysis tool, which is now marketed by Telelogic, a subsidiary of IBM. Feather and Menzies [2002] used simulated annealing to solve requirements selection and optimization for a NASA project. Ruhe et al. [Greer and Ruhe 2004; Ruhe and Greer 2003; Ruhe and Ngo-The 2004] used a genetic algorithm to select requirements and represented results of the application of this approach to a real-world dataset. Ngo-The and Ruhe [2009] combine integer linear programming with a genetic algorithm to overcome the weaknesses of genetic algorithms with a two-phase approach. Baker et al. [2006] also consider requirements problems as a selection problem, presenting results for simulated annealing and greedy algorithms on the Motorola dataset used in the present paper. AlBourae et al. [2006] also use both greedy and AHP algorithms in a release re-planning approach. Jalali et al. [2008] use a greedy algorithm to address the problem of risk reduction, where risks are characterized in terms of the risk of introducing new requirements.

Thus, previous work demonstrates the value of applying optimization techniques to the problem of requirements analysis. Recent work has shown that the results obtained by these approaches are superior to those assessments that can be made by a human in their optimization of choices [de Souza et al. 2010]. These "human-competitive" findings provide further evidence for the importance of optimization in requirements analysis.

However, when it comes to the problem of sensitivity, the inherent stochastic nature of all of the algorithms used means that they are inherently unreliable; we shall not know whether fluctuations are due to sensitivity or to the algorithms natural stochastic properties. The primary difference in the approach adopted in the present article lies in its use of a scalable exact algorithm for the NRP and its use as a "precise instrument" with which to address the sensitivity of a solution to the potential imprecision in the estimates upon which it is based.

This article concerns a single objective formulation of the NRP in which the problem is to find a set of requirements that maximize revenue while falling within budget. However, the underlying algorithm used for exact NRP solutions could be used to solve the bi-objective NRP problem. This may lead to extensions of the work in this article to consider multi-objective formulations of the NRP.

Zhang et al. [2007] introduced a multi-objective formulation in terms of a cost-benefit trade-off. In this approach the budget is not fixed. Instead, budget minimization becomes an additional objective. Saliu and Ruhe [2007a] also introduced a multi-objective formulation in which the balance was between concerns at two levels of abstraction: implementation and requirement, rather than between cost and value. Finkelstein et al. [2008a] used a multiple objective approach to analyze fairness in requirements. In this approach, the objective is to minimize cost while simultaneously maximizing fairness according to several different formulations of fairness; each of the different notions of fairness corresponds to an objective. Feather et al. [2004, 2006] also used a form of multi-objective visualization of the results from their simulated annealing approach, in which results are presented on a Pareto front.

In this article, we use a sensitivity analysis approach. Sensitivity analysis (SA) is found in other areas of engineer but is not widely used, hitherto, in requirements engineering. Other applications of SA are widely found in the literature for various areas, such as chemical kinetics [Sandu et al. 2003], physical science [Newman et al. 1999], environmental modeling [Hamby 1994], telecommunications engineering [Racu et al. 2005], and financial analysis [Levine and Renelt 1992]. In software engineering, the application of sensitivity analysis has been mostly focused on the area of software reliability and prediction models [Rodrigues et al. 2005; Wagner 2007a, 2007b; Zhu et al. 2005]. Harman et al. presented a search-based sensitivity analysis of NRP [2009]. However, precise sensitivity analysis requires exact algorithms in order to avoid unwanted and potentially ruinous 'noise' from the approximate nature of the algorithm.

## 11. IMPLICATIONS OF THE RESULTS FOR WORK ON NRP

The results in this article have implications for subsequent work on the NRP and release planning. Most previous work has been concerned with giving insight to decision makers about possible choices of requirement. For these applications, the inherent imprecision of metaheuristic methods may not be an issue.

In some circumstances, such as the exploration of trade-offs between different objectives in multiobjective formulations [Finkelstein et al. 2008b; Saliu and Ruhe 2007b], it may be sufficient to use inexact algorithms, which may offer other benefits (such as handling messy, incomplete data) or where these formations have no known precise solution approach that can be computed in reasonable time.

However, for the application of optimization-based approaches to problems concerning the RSA, it is important to have an algorithm that guarantees globally optimal solutions at the heart of the approach; to assess the impact of estimate *inaccuracies*, we need an *accurate* approach. Without an optimal algorithm, it will not be clear whether sensitivity observed in solutions obtained is due to PIACs or whether it arises due to the inherent stochastic nature of the algorithm used. In particular, that the previous work on sensitivity analysis using a genetic algorithm [Harman et al. 2009], suffers from this problem.

This does not mean that metaheuristic approach cannot be used for any aspect of RSA. As we discuss in the future work for this research agenda, for higher-order effects (interactions between estimate inaccuracies), it may prove essential to use a metaheuristic approaches to cater for the scale of the space of possible interactions. However, in such a scenario, it may be unrealistic to expect that a complete characterization of all estimate inaccuracy risks can be captured.

## 12. FUTURE WORK

Future work will consider different formulations of the requirements problem, including those with complex dependencies between requirements, for example, where the value and cost of one requirement are affected by the other requirements also included in the release of the software.

Handling higher-order effects for orders $n$, ($n > 2$) remains an interesting open problem for future work. In this article, we showed how a one-at-a-time analysis (OAT) could be used to handle all possible single estimate inaccuracies and how this could be extended to all second-order interaction effects between estimate inaccuracies. Approaches to higher orders of interactions require very different approaches, since they will not be so easy to visualize and the computational complexity grows exponentially with the interaction order. One interesting avenue for future work will consist of using SBSE to search the space of higher-order interactions to locate potential problematic cases.

Other requirements selection algorithms, such as $\text{OPTIMIZE}_{\text{RASORP}}$ [Ngo-The and Ruhe 2009], could also be considered in future work to determine whether or not they could form a suitably precise foundation on which to build sensitivity analyses.

## 13. CONCLUSIONS

There has been a lot of recent interest in the application of search-based software engineering (SBSE) to requirements analysis optimization. One important goal of this work has been to find algorithms that are able to select an ideal set of requirements for the next release of the system, an activity known as *release planning* for an optimization problem known as the Next Release Problem (NRP). Recent work has demonstrated human competitive results for this area of SBSE. However, there remains a problem: the optimization can only ever be as good as the quality of the estimates upon which it is based. Software engineering estimation inaccuracy is widely believed to be significant, making this an important problem.

In this article, we introduce a one-at-a-time (OAT) sensitivity analysis, incorporating a scalable exact optimization algorithm as the NRP Solver at the heart of the analysis. We demonstrated that this exact algorithm can be used to precisely assess the sensitivity of an instance of the NRP to inaccuracies in its estimate. This allows the requirements engineer to locate relatively risk-free 'insensitive' budget choices and to identify those 'sensitive' requirements for which estimates are particularly important. Scalability is clearly important, but it is difficult, because the NRP is NP-hard. We presented results from an experimental study that demonstrate the scalability of our approach, together with a real-world case study that illustrates the way in which our approach can assist a requirements engineer. We also illustrate that analysis of higher-order estimate inaccuracy is feasible using an exact algorithm.

Armed with a reliable assessment of sensitivity, the requirements analyst can better account for the impact of estimate inaccuracies, thereby making better informed choices in the crucial early stages of the software development process. The case study scenarios in this article show how our analysis can reveal particularly sensitive budget levels and requirements that might otherwise have gone unnoticed by the requirements decision maker.

## ACKNOWLEDGMENTS

We are grateful to the referees for suggesting many ways of improving the original manuscript. In particular, one of the referees drew our attention to the importance of considering higher-order effects, which occasioned a major revision to this article.

## REFERENCES

E. Alba and F. Chicano. 2007. Software project management with GAs. *Inf. Sci.* 177, 11, 2380–2401.

T. Albourae, G. Ruhe, and M. Moussavi. 2006. Lightweight replanning of software product releases. In *Proceedings of the 1st International Workshop on Software Product Management (IWSPM'06)*. IEEE Computer Society, 27–34.

R. Alvarez-Valdés, E. Crespo, J. M. Tamarit, and F. Villa. 2006. A scatter search algorithm for project scheduling under partially renewable resources. *J. Heuristics* 12, 1–2, 95–113.

G. Antoniol, M. Di Penta, and M. Harman. 2004. A robust search-based approach to project management in the presence of abandonment, rework, error and uncertainty. In *Proceedings of the 10th International Symposium on the Software Metrics (METRICS'04)*. IEEE Computer Society, 172–183.

G. Antoniol, M. D. Penta, and M. Harman. 2005. Search-based techniques applied to optimization of project planning for a massive maintenance project. In *Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05)*. IEEE Computer Society, Los Alamitos, CA, 240–249.

A. Bagnall, V. Rayward-Smith, and I. Whittley. 2001. The next release problem. *Inf. Softw. Technol.* 43, 14, 883–890.

P. Baker, M. Harman, K. Steinhofel, and A. Skaliotis. 2006. Search based approaches to component selection and prioritization for the next release problem. In *Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM'06)*. IEEE Computer Society, 176–185.

R. Beier and B. Vöcking. 2004. Random knapsack in expected polynomial time. *J. Comput. Syst. Sci.* 69, 3, 306–329.

R. E. Bixby, M. Fenelon, and Z. Gu. 2000. MIP: Theory and practice—closing the gap. *Syst. Model. Optim.: Methods, Theory, Appl.* 1, 1960, 1–31.

C. Chao, J. Komada, Q. Liu, M. Muteja, Y. Alsalqan, and C. Chang 1993. An application of genetic algorithms to software project management. In *Proceedings of the 9th International Advanced Science and Technology*. 247–252.

B. H. Cheng and J. M. Atlee. 2007. Research directions in requirements engineering. In *Proceedings of the Future of Software Engineering*. IEEE Computer Society, Los Alamitos, CA, 285–303.

T. E. Colanzi, S. R. Vergilio, W. K. G. Assuno, and A. Pozo. 2012. Search based software engineering: Review and analysis of the field in Brazil. *J. Syst. Softw.*

J. T. de Souza, C. L. Maia, F. G. de Freitas, and D. P. Coutinho. 2010. The human competitiveness of search based software engineering. In *Proceedings of the 2nd International Symposium on Search Based Software Engineering (SSBSE'10)*. IEEE Computer Society.

M. S. Feather, S. L. Cornford, J. D. Kiper, and T. Menzies. 2006. Experiences using visualization techniques to present requirements, risks to them, and options for risk mitigation. In *Proceedings of the International Workshop on Requirements Engineering Visualization (REV'06)*. IEEE, 10–10.

M. S. Feather, J. D. Kiper, and S. Kalafat. 2004. Combining heuristic search, visualization and data mining for exploration of system design space. In *Proceedings of the 14th Annual International Symposium of International Council on Systems Engineering (INCOSE'04).*

M. S. Feather and T. Menzies. 2002. Converging on the optimal attainment of requirements. In *Proceedings of the 10th IEEE International Conference on Requirements Engineering (RE'02)*. IEEE, 263–270.

A. Finkelstein, M. Harman, S. A. Mansouri, J. Ren, and Y. Zhang. 2008a. Fairness analysis in requirements assignments. In *Proceedings of the 16th IEEE International Requirements Engineering Conference (RE'08)*. IEEE Computer Society, 115–124.

A. Finkelstein, M. Harman, S. A. Mansouri, J. Ren, and Y. Zhang. 2008b. Fairness analysis in requirements assignments. In *Proceedings of the 16th IEEE International Requirements Engineering Conference (RE'08)*.

F. G. Freitas and J. T. Souza. 2011. Ten years of search based software engineering: A bibliometric analysis. In *Proceedings of the 3rd International Symposium on Search Based Software Engineering (SSBSE'11)*. 18–32.

H. C. Frey and S. Patil. 2002. Identification and review of sensitivity analysis methods. *Risk Anal.* 22, 3, 553–578.

M. R. Garey and D. S. Johnson. 1978. Strong NP-completeness results: Motivation, examples, and implications. *J. ACM* 25, 499–508.

M. R. Garey and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

D. Greer and G. Ruhe. 2004. Software release planning: An evolutionary and iterative approach. *Inf. Softw. Technol.* 46, 4, 243–253.

D. M. Hamby. 1994. A review of techniques for parameter sensitivity analysis of environmental models. *Environ. Monitor. Assess.* 32, 2, 135–154.

M. Harman. 2007. The current state and future of search based software engineering. In *Proceedings of the Future of Software Engineering,* L. Briand and A. Wolf, Eds. IEEE Computer Society Press, Los Alamitos, CA, 342–357.

M. Harman, E. Burke, J. A. Clark, and X. Yao. 2012a. Dynamic adaptive search based software engineering. In *Proceedings of the 6th IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'12).*

M. Harman, J. Krinke, J. Ren, and S. Yoo. 2009. Search based data sensitivity analysis applied to requirement engineering. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*. ACM, 1681–1688.

M. Harman, A. Mansouri, and Y. Zhang. 2012b. Search based software engineering: Trends, techniques and applications. *ACM Comput. Surv.* 45, 1, Article 11.

M. Harman, P. Mcminn, J. Souza, and S. Yoo. 2012c. Search based software engineering: Techniques, taxonomy, tutorial. In *Empirical Software Engineering and Verification: (LASER'09–10)*, B. Meyer and M. Nordio, Eds., Lecture Notes in Computer Science, vol. 7007, Springer, 1–59.

J. Helton, J. Johnson, C. Sallaberry, and C. Storlie. 2006. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Eng. Syst. Safety* 91, 10–11, 1175–1209.

O. Jalali, T. Menzies, and M. Feather. 2008. Optimizing requirements decisions with KEYS. In *Proceedings of the 4th International Workshop on Predictor Models in Software Engineering (PROMISE'08)*. ACM, 79–86.

P. Kapur, A. Ngo-The, G. Ruhe, and A. Smith. 2008. Optimized staffing for product releases and its application at Chartwell Technology. *J. Softw. Maintenance Evol. Res. Pract.* (Special Issue Search Based Software Engineering), 20, 5, 365–386.

J. Karlsson, C. Wohlin, and B. Regnell. 1998. An evaluation of methods for priorizing software requirements. *Info. Softw. Technol.* 39, 14–15, 939–947.

R. Karp. 1972. Reducibility among combinatorial problems. *Complex. Comput. Compu.* 40, 4, 85–103.

R. Levine and D. Renelt. 1992. A sensitivity analysis of cross-country growth regressions. *Amer. Econ. Rev.* 82, 4, 942–963.

C. Li, M. van den Akker, S. Brinkkemper, and G. Diepen. 2007. Integrated requirement selection and scheduling for the release planning of a software product. In *Proceedings of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ'07)*. Lecture Notes in Computer Science, vol. 4542. Springer, 93–108.

G. L. Nemhauser and Z. Ullmann. 1969. Discrete dynamic programming and capital allocation. *Manage. Sci.* 15, 9, 494–505.

J. Newman, A. Taylor, R. Barnwell, and P. Newman. 1999. Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations. *J. Aircraft* 36, 1, 87–96.

A. Ngo-The and G. Ruhe. 2009. Optimized resource allocation for software release planning. *IEEE Trans. Softw. Eng.* 35, 1, 109–123.

D. Pisinger. 2005. Where are the hard knapsack problems? *Comput. Oper. Res.* 32, 9, 2271–2284.

R. Racu, M. Jersak, and R. Ernst. 2005. Applying sensitivity analysis in real-time distributed systems. In *Proceedings of the 11th IEEE Real Time and Embedded Technology and Applications Symposium*. IEEE Computer Society, 160–169.

G. N. Rodrigues, D. S. Rosenblum, and S. Uchitel. 2005. Sensitivity analysis for a scenario-based reliability prediction model. In *Proceedings of the Workshop on Architecting Dependable Systems (WADS'05)*. ACM, New York, NY, 1–5.

G. Ruhe, A. Eberlein, and D. Pfahl. 2003. Trade-off analysis for requirements selection. *Int. J. Softw. Eng. Knowl. Eng.* 13, 4, 345–366.

G. Ruhe and D. Greer. 2003. Quantitative studies in software release planning under risk and resource constraints. In *Proceedings of the International Symposium on Empirical Software Engineering (ISESE'03)*. IEEE, 262–270.

G. Ruhe and A. Ngo-The. 2004. Hybrid intelligence in software release planning. *Int. J. Hybrid Intell. Syst.* 1, 1–2, 99–110.

M. O. Saliu and G. Ruhe. 2007a. Bi-objective release planning for evolving software systems. In *Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC-FSE'07)*. ACM, New York, NY, 105–114.

M. O. Saliu and G. Ruhe. 2007b. Bi-objective release planning for evolving software systems. In *Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE'07)*. I. Crnkovic and A. Bertolino, Eds., ACM, 105–114.

A. Saltelli and P. Annoni. 2010. How to avoid a perfunctory sensitivity analysis. *Environ. Model. Softw.* 25, 12, 1508–1517.

A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. 2008. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Sussex.

A. Saltelli, S. Tarantola, and F. Campolongo. 2000. Sensitivity analysis as an ingredient of modeling. *Stat. Sci.* 15, 4, 377–395.

A. Sandu, D. N. Daescu, and G. R. Carmichael. 2003. Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: Part I–theory and software tools. *Atmospheric Environ.* 37, 36, 5083–5096.

M. Shepperd. 2007. Software project economics: A roadmap. In *Proceedings of the Future of Software Engineering (FoSE'07)*. IEEE Computer Society, 304–315.

M. van den Akker, S. Brinkkemper, G. Diepen, and J. Versendaal. 2004. Flexible release composition using integer linear programming. Tech. rep. UU-CS-2004-063, Institute of Information and Computing Sciences, Utrecht University.

M. van den Akker, S. Brinkkemper, G. Diepen, and J. Versendaal. 2005. Flexible release planning using integer linear programming. In *Proceedings of the 11th International Workshop on Requirements Engineering for Software Quality (RefsQ'05)*. Essener Informatik Beitrage, 247–262.

S. Wagner. 2007a. An approach to global sensitivity analysis: FAST on COCOMO. In *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement (ESEM'07)*. IEEE Computer Society, Los Alamitos, CA, 440–442.

S. Wagner. 2007b. Global sensitivity analysis of predictor models in software engineering. In *Proceedings of the 3rd International Workshop on Predictor Models in Software Engineering (PROMISE'07)*. IEEE Computer Society, 3–10.

Y. Zhang, A. Finkelstein, and M. Harman. 2008. Search based requirements optimisation: Existing work and challenges. In *Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'08)*. Lecture Notes in Computer Science, vol. 5025, Springer, 88–94.

Y. Zhang, M. Harman, and S. A. Mansouri. 2007. The multi-objective next release problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM Press, New York, NY, 1129–1136.

L. Zhu, A. Aurum, I. Gorton, and R. Jeffery. 2005. Tradeoff and sensitivity analysis in software architecture evaluation using analytic hierarchy process. *Softw. Quality Control* 13, 4, 357–375.

Received May 2012; revised April, August 2013; accepted October 2013