WILEY ECOLOGICAL APPLICATIONS

Integer linear programming outperforms simulated annealing for solving conservation planning problems

Journal:	*Ecological Applications*
Manuscript ID	EAP19-0403
Wiley - Manuscript type:	Communications
Date Submitted by the Author:	20-Jun-2019
Complete List of Authors:	Schuster, Richard; Carleton University, Biology Hanson, Jeffrey; University of Queensland Strimas-Mackey, Matt; Cornell Lab of Ornithology Bennett, Joseph; Carleton University
Substantive Area:	Conservation < Landscape < Substantive Area
Organism:	
Habitat:	
Geographic Area:	North America < Geographic Area
Additional Keywords:	Marxan, Integer Linear Programming, Optimization, Prioritization, Conservation Planning, prioritizr
Abstract:	The resources available for conserving biodiversity are limited, and so protected areas need to be established in places that will achieve objectives for minimal cost. Two of the main algorithms for solving systematic conservation planning problems are Simulated Annealing (SA) and Integer linear programming (ILP). Using a case study in British Columbia, Canada, we compare the cost-effectiveness and processing times of SA versus ILP using both commercial and open-source algorithms. Plans for expanding protected area systems based on ILP algorithms were 12 to 30% cheaper than plans using SA. The best ILP solver we examined was on average 1071 times faster than the SA algorithm tested. The performance advantages of ILP solvers were also observed when we aimed for spatially compact solutions by including a boundary penalty. One practical advantage of using ILP over SA is that the analysis does not require calibration, saving even more time. Given the performance of ILP solvers, they can be used to generate conservation plans in real-time during stakeholder meetings and can facilitate rapid sensitivity analysis, making the conservation planning process more interactive.

Title: Integer linear programming outperforms simulated annealing for solving conservation planning problems

Authors: Richard Schuster[a,b,*], Jeffrey O. Hanson[c], Matt Strimas-Mackey[d], Joseph R. Bennett[a]

[a] Department of Biology, 1125 Colonel By Drive, Carleton University, Ottawa ON, K1S 5B6 Canada.

[b] Ecosystem Science and Management Program, 3333 University Way, University of Northern British Columbia, Prince George BC, V2N 4Z9 Canada.

[c] School of Biological Sciences, The University of Queensland, Brisbane, QLD 4072, Australia

[d] Cornell Lab of Ornithology, Cornell University, Ithaca, NY 14850 USA.

*Corresponding author: Department of Biology, 1125 Colonel By Drive, Carleton University, Ottawa ON, K1S 5B6 Canada. Email: richard.schuster@glel.carleton.ca, Phone: +1 250 631 8324

Running Title: Solving conservation planning problems

Keywords: Marxan, Integer Linear Programming, Optimization, Prioritization, Conservation Planning, prioritizr

Type of article: Communications (max 20 ms pages)

Abstract

The resources available for conserving biodiversity are limited, and so protected areas need to be established in places that will achieve objectives for minimal cost. Two of the main algorithms for solving systematic conservation planning problems are Simulated Annealing (SA) and Integer linear programming (ILP). Using a case study in British Columbia, Canada, we compare the cost-effectiveness and processing times of SA versus ILP using both commercial and open-source algorithms. Plans for expanding protected area systems based on ILP algorithms were 12 to 30% cheaper than plans using SA. The best ILP solver we examined was on average 1071 times faster than the SA algorithm tested. The performance advantages of ILP solvers were also observed when we aimed for spatially compact solutions by including a boundary penalty. One practical advantage of using ILP over SA is that the analysis does not require calibration, saving even more time. Given the performance of ILP solvers, they can be used to generate conservation plans in real-time during stakeholder meetings and can facilitate rapid sensitivity analysis, making the conservation planning process more interactive.

Introduction

Area-based systematic conservation planning aims to provide a rigorous, repeatable, and structured approach for designing new protected areas that efficiently meet conservation objectives (Margules & Pressey 2000). Historically, spatial conservation decision-making often evaluated parcels opportunistically as they became available for purchase, donation, or under threat (Pressey et al. 1993; Pressey & Bottrill 2008). Although purchasing such areas may improve the status quo, such decisions may not substantially enhance the long-term persistence of species or communities or be cost-effective (Joppa & Pfaff 2009; Venter et al. 2014). Systematic conservation planning, on the other hand, involves framing conservation planning problems as optimization problems, with clearly defined objectives (e.g. minimize acquisition cost) and constraints. These optimization problems are then solved to obtain candidate reserve designs (termed solutions), which are used to guide protected area acquisitions and land policy (Schwartz et al. 2018). Due to the systematic, evidence-based nature of these tools, they can help contribute to a transparent, inclusive, and more defensible decision-making process (Margules & Pressey 2000).

Today, Marxan is the most widely used systematic conservation planning software, having been used in 184 countries to design marine and terrestrial reserve systems (Ball et al. 2009). Although Marxan supports several algorithms for solving conservation planning problems, most conservation planning exercises use its implementation of the simulated annealing (SA), an iterative, stochastic metaheuristic for approximating global optima of complex functions with many local optima. By conducting thousands of individual runs, each with millions of iterations, Marxan aims to generate solutions that are near-optimal. One of the reasons why Marxan uses SA instead of integer linear programming (ILP), is that ILP was not well suited to solve problems with nonlinear constraints and penalties, such as problems trying to create spatially compact or connected solutions (i.e. compactness and connectivity goals). However, the SA approach provides no guarantee on solution quality. In particular, solutions may be highly suboptimal and conservation scientists and practitioners have no way of knowing how far from optimality generated solutions are.

In a recent simulation study, Beyer et al. (2016) found that Marxan with simulated annealing can deliver solutions that are orders of magnitude below optimality. They compared Marxan to integer linear programming (ILP) (Dantzig 2016), which minimizes or maximizes an objective function (a mathematical equation describing the relationship between actions and outcomes) subject to a set of constraints and conditional on the decision variables (the variables corresponding to the selection of actions to implement) being integers (Beyer et al. 2016). Unlike heuristic methods such as SA, prioritization using ILP will find the exact optimal solution or can be instructed to return solutions within a defined distance from optimality. Some have argued that ILP algorithms are well-suited for solving conservation planning problems (Cocks & Baird 1989; Underhill 1994; Rodrigues & Gaston 2002), but until recent advances in computational capacity and algorithms, it has been impossible to solve the Marxan-like systematic conservation planning problems with ILP for large problems (Beyer et al. 2016).

Here we compare integer linear programming with simulated annealing (i.e. Marxan) for solving systematic conservation planning problems using real-world data from Western North America. We found that ILP produced higher quality solutions potentially saving >$100 million (or 13%) for realistic conservation scenarios, and that solutions were generated >1,000 times faster than using simulated annealing, opening up new possibilities for scenario generation. These findings also hold true problems aiming for spatially compact solutions.

Methods

Study area

We focused on a 27,250 km^2 portion of the Georgia Basin, Puget Trough and Willamette Valley of the Pacific Northwest region spanning the US and Canada, corresponding to the climate envelope indicative of the Coastal Douglas-fir (CDF) Biogeoclimatic zone in southwestern British Columbia (Meidinger & Pojar 1991). Land cover in the region is diverse, with approximately 57% of the land in forest, 8% as savanna or grassland, 5% in cropland, and 10% being urban or built.

Biodiversity data.

We used species distribution models for 72 bird species as our conservation features (Supplementary Table 1). The distribution models were based on data from eBird, a citizen-science effort that has produced the largest and most rapidly growing biodiversity database in the world (Hochachka et al. 2012; Sullivan et al. 2014). From the 2013 eBird Reference Dataset (http://ebird.org/ebird/data/download) we used a total of 12,081 checklists in our study area, then filtered these checklists to retain only those <1.5 hours in duration, <5 km travelled, and with a maximum of 10 visits to a given location (unpublished R code; Hochachka W., pers. com.). Sampling locations <100 m apart were collapsed to one location, yielding 5,470 checklists from 2,160 locations, visited from 1-10 times and 2.53 times on average. The R package unmarked (version 0.9-9; Fiske and Chandler 2011) provided the framework for all species distribution models, which necessarily include two parts: occupancy and detection (Mackenzie et al. 2002). For further details on biodiversity data see (Rodewald et al. in revision).

Cadastral layer and land cost.

We incorporated spatial heterogeneity in land cost (Ando et al. 1998; Polasky et al. 2001; Ferraro 2003; Naidoo et al. 2006) in our plans by using cadastral data and 2012 land value assessments from the Integrated Cadastral Information Society of BC. This process resulted in 193,623 polygons for BC which were subsequently used as planning units (Schuster et al. 2014). Cadastral data, including tax assessment land values from Washington State came from the University of Washington's Washington State Parcel Database (https://depts.washington.edu/wagis/projects/parcels/; Version: StatewideParcels_v2012n_e9.2_r1.3; Date accessed: 2015/04/30), as well as San Juan County Parcel Data with separate signed user agreement. The combined cadastral layer included 1.92 million planning units. Cadastral data, including tax assessment land values from Oregon State had to be sourced from individual counties, which included Benton, Clackamas, Columbia, Douglas, Lane, Linn, Marion, Multnomah, Polk, Washington and Yamhill. The combined cadastral layer for Oregon included 605,425 planning units.

Spatial prioritization

We compared ILP and SA for solving the minimum set spatial prioritization problem (Ball et al. 2009). In this formulation, the landscape is divided into a set of discrete planning units. Each planning unit is assigned a socioeconomic cost (here we use the assessed land value) and a conservation value for a set of features that we wish to protect (here the occupancy probability for a set of species). Finally, we define representation targets for each species as the amount of habitat we hope to protect for that species. The goal of this prioritization problem is to optimize the trade-off between conservation benefit and socioeconomic cost (McIntosh et al.

2017). Achieving this goal involves finding the set of planning units that meets the conservation targets for the minimum possible cost. Details on the Marxan problem formulation can be found in Ball et al. (2009) and the ILP formulation in Beyer et al. (2016). Three key parameters that are important for Marxan analysis, which we also use here are: species penalty factor, number of iterations, and number of restarts (Ardron et al. 2010). Briefly, the species penalty factor is the penalty given to a reserve system for not adequately representing a feature, the number of iterations determines how long the annealing algorithms will run, and the number of restarts determines how many different solutions Marxan will generate.

ILP solvers (commercial vs open source)

A variety of ILP solvers currently exist, and both commercial and open source solvers are available. All solvers yield optimal solutions to ILP problems, but there are substantial differences in performance (i.e. time taken to solve a problem) and in the size of problems that can be solved (Lin et al. 2017). For the purposes of performance testing we opted for one of the best commercial solvers currently available, Gurobi (Gurobi Optimization Inc. 2017). In a recent benchmark study, Gurobi outperformed other solver packages for more complex formulations and a practical use-case (Luppold et al. 2018). Gurobi Optimization Inc. provides a free academic license to researchers, but is otherwise costly for non academic institutions and individuals. To investigate solver performance of packages that are freely available to everyone, we also tested the open source solver SYMPHONY (Ralphs et al. 2019). Both Gurobi and SYMPHONY can be used from R. For Gurobi we used the R package provided with the software (gurobi version 8.1-0) and for SYMPHONY the Rsymphony package (version 0.1-28;

Harter et al. 2017). We used the prioritizr R package to solve ILP problems for both Gurobi and SYMPHONY solvers (Hanson et al. 2019).

Scenarios investigated

We investigated a range of scenarios that were computationally feasible for this study. For both Marxan and prioritzr we created the following range of scenarios: i) vary conservation targets between 10 and 90% protection of features in 10% increments (9 variations), using ii) 10 – 72 species/features (5 variations) as targets, and iii) with spatial extents of 9,282, 37,128, and 148,510 planning units (3 variations), resulting in a total of 135 scenarios created (Table 1). For Marxan, we also varied two additional parameters, i) the number of iterations ranged from 10^4 to 10^8 (5 variations) and ii) species penalty factors (SPF) of 1, 5, 25, and 125 were explored (4 variations) for a total of 2,700 scenarios investigated in Marxan (Table 1). As the processing time for the most complex problem in Marxan (90% target, 72 features, 148,510 planning units, 10^8 iterations) was >8 hours, we restricted the full range of scenarios to those mentioned above. The maximum number of planning units we used is within the range of previous studies using Marxan (e.g. Venter et al. 2014; Runge et al. 2016), although using more than 50,000 planning units with SA is discouraged without extensive parameter calibration, as near optimal solutions will be hard to find for problems of that size (Ardron et al. 2010).

As systematic conservation planners often aim for spatially compact solutions to their problems, we also investigated a range of scenarios using a term called boundary length modified (BLM), which is used to improve the clustering and compactness of a solution (McDonnell et al. 2002). We randomly selected a 225 x 225 pixel region of the study area to generate a problem with 50, 625 planning units, the maximum recommended for Marxan. After initial calibration we set the number of features/species to 72, SPF to 25 and number of iterations for Marxan to 10^8. We varied targets between 10 and 90% protection of features in 10% increments, and used the following BLM values: 0.1; 1; 10; 100; 1,000 for a total of 45 scenarios.

All analyses were conducted on a desktop computer with an Intel Core i7-7820X Processor and 128 GB RAM running Ubuntu 18.04 and R v 3.5.3. All data, scripts and full results are available here: https://osf.io/my8pc/?view_only=eaf7a8aff8314dd789f1873053fae27a

Results

ILP algorithms (Gurobi, SYMPHONY) outperformed SA (Marxan) in terms of their ability to find optimal solutions across all scenarios that met their targets. Through finding optimal solutions, using ILP resulted in cost savings ranging from 0.8 to 4,369% (median 72.7%). When we restricted results to only take into account calibrated Marxan scenarios (number of iterations > 100,000 and species penalty factor 5 or 25), the range of savings was reduced to 0.8 to 52.5% (median 12.6%, Appendix S1: Figure S1). For example, at the 30% protection target ILP solvers resulted in solutions that were \$144 million cheaper than SA (Figure 1a). With these savings an additional 3,039 ha could be protected (53,934 ha vs 50,895 ha) using an ILP algorithm by raising the representation targets until the cost of the resulting solution matched that of the Marxan solution using SA. In general, SA performed reasonably well at smaller problem sizes, fewer planning units and features and low targets, but as the problem size and complexity increased SA was less consistent in finding good solutions (Appendix S1: Figure S1). Cost profiles across targets, number of features and number of planning units are shown in Appendix S1: Figures S2-4.

The shortest processing times were achieved using the prioritizr package and the commercial solver Gurobi, followed by prioritizr and the open source solver SYMPHONY, and lastly Marxan (Figure 1b). Gurobi had the shortest processing times across all scenarios investigated, SYMPHONY tied with Gurobi in some scenarios and took up to 78 times longer than Gurobi in other scenarios (mean = 14 times, Appendix S1: Figure S5), and Marxan took between 1.8 and 1995 times longer than Gurobi (mean = 281 times, Appendix S1: Figure S6). The longest processing times for Gurobi, SYMPHONY and Marxan for a single scenario were 40 seconds, 31 minutes, and 8 hours respectively. For the most complex problem (i.e. targets = 90%, 72 features; 148,510 planning units), Marxan calibration across the 5 number of iterations and 4 species penalty factor values took a total of 5 days 7 hours, compared to 30 seconds using Gurobi and 28 minutes using SYMPHONY. Time profiles across targets, number of features and number of planning units are shown in Appendix S1: Figures S7-9.

ILP algorithms (Gurobi, SYMPHONY) also outperformed SA (Marxan) when using a BLM to achieve compacter solutions. This was true for objective function values (Figure 2a) as well as for processing times (Figure 2b). Through finding optimal solutions, using ILP resulted in objective function values 5.65 to 149% (mean 22.7%) lower than SA values. Gurobi was the fastest solver to find solutions to problems including BLM in 44 of 45 scenarios, in one case SYMPHONY was faster. SYMPHONY outperformed Marxan in 44 of 45 scenarios, and took on average 13.7 times as long as Gurobi to find a solution (range -0.31 to 42.6). Marxan was never faster than Gurobi and took on average 104.6 times as long as Gurobi to find a solution (range 3.09 to 190.8). An example of the spatial representation of the solutions for a 10% target is shown in Appendix S1: Figure S10.

Discussion

We found that ILP algorithms outperformed SA both in terms of cost-effectiveness and processing times, even when including non-linear problem formulations, when planning for spatially compact solutions. There have been calls for using ILP in solving conservation planning problems in the past (Underhill 1994; Rodrigues & Gaston 2002), but we are finally at a point where making this switch is both advisable and computationally feasible.

One practical advantage of using ILP over SA is that the analysis does not require calibration. A crucial task in every Marxan/SA project is the calibration of parameter for the analysis (Ardron et al. 2010). This task can be very time consuming, especially for larger problems. Using SA, species penalty factor, number of SA iterations, and number of restarts must be calibrated (Ardron et al. 2010). Ideally these parameters should also be explored over the entire parameter space, which would mean that if we wanted to explore three values for each parameter, we would end up with 27 scenarios to explore (i.e. 3 * 3 * 3). With the most complex problem investigated here this would take in the order of several days just to calibrate Marxan runs, which we have done before finalizing parameters and presenting results. None of this calibration time is necessary using ILP. An added benefit is that the somewhat subjective process of setting values for these three parameters can be eliminated using ILP as well.

With the recommendation of a maximum number of 50,000 planning units for a Marxan analysis (Ardron et al. 2010), larger problem sizes should be approached with caution using SA methods. On the other hand, ILP/prioritizr can solve problem sizes of >1 million planning units (Hanson 2018; Schuster et al. 2019). Problems >50,000 planning units have occurred in systematic conservation planning problems (e.g. Venter et al. 2014; Runge et al. 2016), and will likely continue to do so. Realistically, as problem sizes grow beyond what was intended for Marxan/SA projects (50,000 planning units), ILP will run into problems solving very large problems (>1 million planning units) that include non-linear constraints, such as optimizing compactness or connectivity, as those problem formulations need to be linearized for ILP to work. There is the potential to use nonlinear integer programming for more complex problems in the future though (Grossmann 2002; Lee & Leyffer 2011).

Finally, we argue that another strength of ILP solvers, especially Gurobi, is that they can be used to quickly explore and compare different conservation prioritization scenarios in real-time. This ability could be used to great advantage during stakeholder meetings, to explore various scenarios and undertake rapid sensitivity analysis.

Conclusion

ILP algorithms substantially outperform SA as used in systematic conservation planning, both in terms of solution cost, as well as in terms of time required to find near optimal or optimal solutions. Using an ILP algorithm, as implemented in the R package prioritizr, has the added benefit that users do not need to worry about or set parameters such as species penalty factors or number of iterations, which significantly reduces the time a user spends on finding suitable values for these parameters. Given the potential ILP is showing for conservation planning, we recommend users consider adding this modified approach to solving systematic conservation planning problems.

Acknowledgements

RS is supported by a Liber Ero Fellowship and Environment and Climate Change Canada (ECCC), JOH by ECCC, MSM by endowments at the Cornell Lab of Ornithology, and JRB by Natural Sciences and Engineering Research Council of Canada and ECCC. All data, scripts and full results are available here: https://osf.io/my8pc/?view_only=eaf7a8aff8314dd789f1873053fae27a

References

Ando, A., Camm, J., Polasky, S. & Solow, A. (1998). Species Distributions, Land Values, and Efficient Conservation. *Science*, New Series, 279, 2126–2128.

Ardron, J.A., Possingham, H.P. & Klein, C.J. (eds.). (2010). *Marxan Good Practices Handbook, Version 2*. Pacific Marine Analysis and Research Association, Victoria, BC, Canada.

Ball, I.R.R., Possingham, H.P.P. & Watts, M.E.E. (2009). Marxan and relatives: Software for spatial conservation prioritisation. In: *Spatial conservation prioritisation: Quantitative methods and computational tools.* (eds. Moilanen, A., Wilson, K. & Possingham, H.P.). Oxford University Press, Oxford, pp. 185–195.

Beyer, H.L., Dujardin, Y., Watts, M.E. & Possingham, H.P. (2016). Solving conservation planning problems with integer linear programming. *Ecological Modelling*, 328, 14–22.

Cocks, K.D. & Baird, I.A. (1989). Using mathematical programming to address the multiple reserve selection problem: An example from the Eyre Peninsula, South Australia. *Biological Conservation*, 49, 113–130.

Dantzig, G. (2016). *Linear Programming and Extensions*. Princeton University Press.

Ferraro, P.J. (2003). Assigning priority to environmental policy interventions in a heterogeneous world. *Journal of Policy Analysis and Management*, 22, 27–43.

Fiske, I.J. & Chandler, R.B. (2011). unmarked : An R Package for Fitting Hierarchical Models of Wildlife Occurrence and Abundance. *Journal Of Statistical Software*, 43, 128–129.

Grossmann, I.E. (2002). Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering*, 3, 227–252.

Gurobi Optimization Inc. (2017). *Gurobi Optimizer Reference Manual, Version 7.5.1*.

Hanson, J. (2018). *Conserving evolutionary processes* (PhD thesis, DOI: https://doi.org/10.14264/uql.2018.552).

Hanson, J., Schuster, R., Morrell, N., Strimas-Mackey, M., Watts, M.E., Arcese, P., Bennett, J.R. & Possingham, H.P. (2019). *prioritizr: Systematic Conservation Prioritization in R, Version 4.0.2*.

Harter, R., Hornik, K., Theussl, S., Szymanski, C. & Schwendinger, F. (2017). *Rsymphony: SYMPHONY in R*.

Hochachka, W.M., Fink, D., Hutchinson, R.A., Sheldon, D., Wong, W.-K. & Kelling, S. (2012). Data-intensive science applied to broad-scale citizen science. *Trends in ecology & evolution*, 27, 130–137.

Joppa, L.N. & Pfaff, A. (2009). High and far: biases in the location of protected areas. *PloS one*, 4, e8273.

Lee, J. & Leyffer, S. (2011). *Mixed Integer Nonlinear Programming*. Springer Science & Business Media.

Lin, C.Y., Liu, J.W.S., Yeh, K.L. & Chu, E.T.H. (2017). Participant Selection Problem: Relative Performance of Five Optimization Solvers. In: *Proceedings of the 8th International*

Conference on Computer Modeling and Simulation, ICCMS ’17. ACM, New York, NY, USA, pp. 24–31.

Luppold, A., Oehlert, D. & Falk, H. (2018). Evaluating the performance of solvers for integer-linear programming.

Mackenzie, D.I., Nichols, J.D., Lachman, G.B., Droege, S.J., Royle, J.A. & Langtimm, C.A. (2002). Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83, 2248–2255.

Margules, C.R. & Pressey, R.L. (2000). Systematic conservation planning. *Nature*, 405, 243–53.

McDonnell, M.D., Possingham, H.P., Ball, I.R. & Cousins, E.A. (2002). Mathematical Methods for Spatially Cohesive Reserve Design. *Environmental Modeling & Assessment*, 7, 107–114.

McIntosh, E.J., Pressey, R.L., Lloyd, S., Smith, R. & Grenyer, R. (2017). The Impact of Systematic Conservation Planning. *Annual Review of Environment and Resources*, 42, annurev-environ-102016-060902.

Meidinger, D. & Pojar, J. (1991). *Ecosystems of British Columbia*. British Columbia Ministry of Forests, Victoria, BC.

Naidoo, R., Balmford, A., Ferraro, P.J., Polasky, S., Ricketts, T.H. & Rouget, M. (2006). Integrating economic costs into conservation planning. *Trends in ecology & evolution*, 21, 681–7.

Polasky, S., Camm, J.D. & Garber-Yonts, B. (2001). Selecting Biological Reserves Cost-Effectively: An Application to Terrestrial Vertebrate Conservation in Oregon. *Land Economics*, 77, 68–78.

Pressey, R., Humphries, C., Margules, C., Vane-Wright, R. & Williams, P. (1993). Beyond opportunism: key principles for systematic reserve selection. *Trends in ecology & evolution*, 8, 124–128.

Pressey, R.L. & Bottrill, M.C. (2008). Opportunism, Threats, and the Evolution of Systematic Conservation Planning. *Conservation Biology*, 22, 1340–1345.

Ralphs, T., Mahajan, A., Vigerske, mgalati13, LouHafer, jpfasano, Bulut, A. & anhhz. (2019). *coin-or/SYMPHONY: Version 5.6.17*. Zenodo.

Rodewald, A., Strimas-Mackey, M., Schuster, R. & Arcese, P. (in revision). Avoiding ‘uninformed opportunism’ by understanding the value of biodiversity feature and cost data in conservation prioritization.

Rodrigues, A.S.L. & Gaston, K.J. (2002). Optimisation in reserve selection procedures—why not? *Biological Conservation*, 107, 123–129.

Runge, C.A., Tulloch, A.I.T., Possingham, H.P., Tulloch, V.J.D. & Fuller, R.A. (2016). Incorporating dynamic distributions into spatial prioritization. *Diversity and Distributions*, 22, 332–343.

Schuster, R., Martin, T.G. & Arcese, P. (2014). Bird Community Conservation and Carbon Offsets in Western North America. *Plos One*.

Schuster, R., Wilson, S., Rodewald, A.D., Arcese, P., Fink, D., Auer, T. & Bennett, J.R. (2019). Optimizing the conservation of migratory species over their full annual cycle. *Nature Communications*, 10, 1754.

Schwartz, M.W., Cook, C.N., Pressey, R.L., Pullin, A.S., Runge, M.C., Salafsky, N., Sutherland, W.J. & Williamson, M.A. (2018). Decision Support Frameworks and Tools for Conservation. *Conservation Letters*, 11, e12385.

Sullivan, B.L., Aycrigg, J.L., Barry, J.H., Bonney, R.E., Bruns, N., Cooper, C.B., Damoulas, T., Dhondt, A.A., Dietterich, T., Farnsworth, A. & others. (2014). The eBird enterprise: an integrated approach to development and application of citizen science. *Biological Conservation*, 169, 31–40.

Underhill, L.G. (1994). Optimal and suboptimal reserve selection algorithms. *Biological Conservation*, 70, 85–87.

Venter, O., Fuller, R.A., Segan, D.B., Carwardine, J., Brooks, T., Butchart, S.H.M., Marco, M.D., Iwamura, T., Joseph, L., O'Grady, D., Possingham, H.P., Rondinini, C., Smith, R.J., Venter, M. & Watson, J.E.M. (2014). Targeting Global Protected Area Expansion for Imperiled Biodiversity. *PLOS Biology*, 12, e1001891.

Table 1. Scenarios investigated in our analysis. The total number of scenarios tested for both Gurobi and SYMPHONY are 135. For Marxan analysis, we included calibration steps as well, which brought the total number of scenarios to 2700 for that algorithm.

Paremeter	**Value range**	**n**	**Scenarios**
targets	10 - 90%	9	
# features	10, 26, 41, 56, 72	5	
# planning units	9,282, 37,128, 148,510	3	135 (ILP)
Marxan iterations	10^4, 10^5, 10^6, 10^7, 10^8	5	
Marxan SPF	1, 5, 25, 125	4	2,700 (SA)

Figure 1. Solution cost and time comparisons. a) The lines represent costs compared to the Gurobi cost baseline. The numbers on the blue line represent total cost of a solution in million $ and the numbers on the green line represent how much more expensive, again in million $, the SA/Marxan solution is compared to the ILP solutions. b) Time to solution comparisons between solvers. Marxan parameters used are: 72 features, 148,510 planning units, 10^8 iterations, using mean cost and time. Note that in a) gurobi (red) and Rsymphony (blue) yielded optimal solutions for all target values and so their lines are plotted exactly on top of each other.

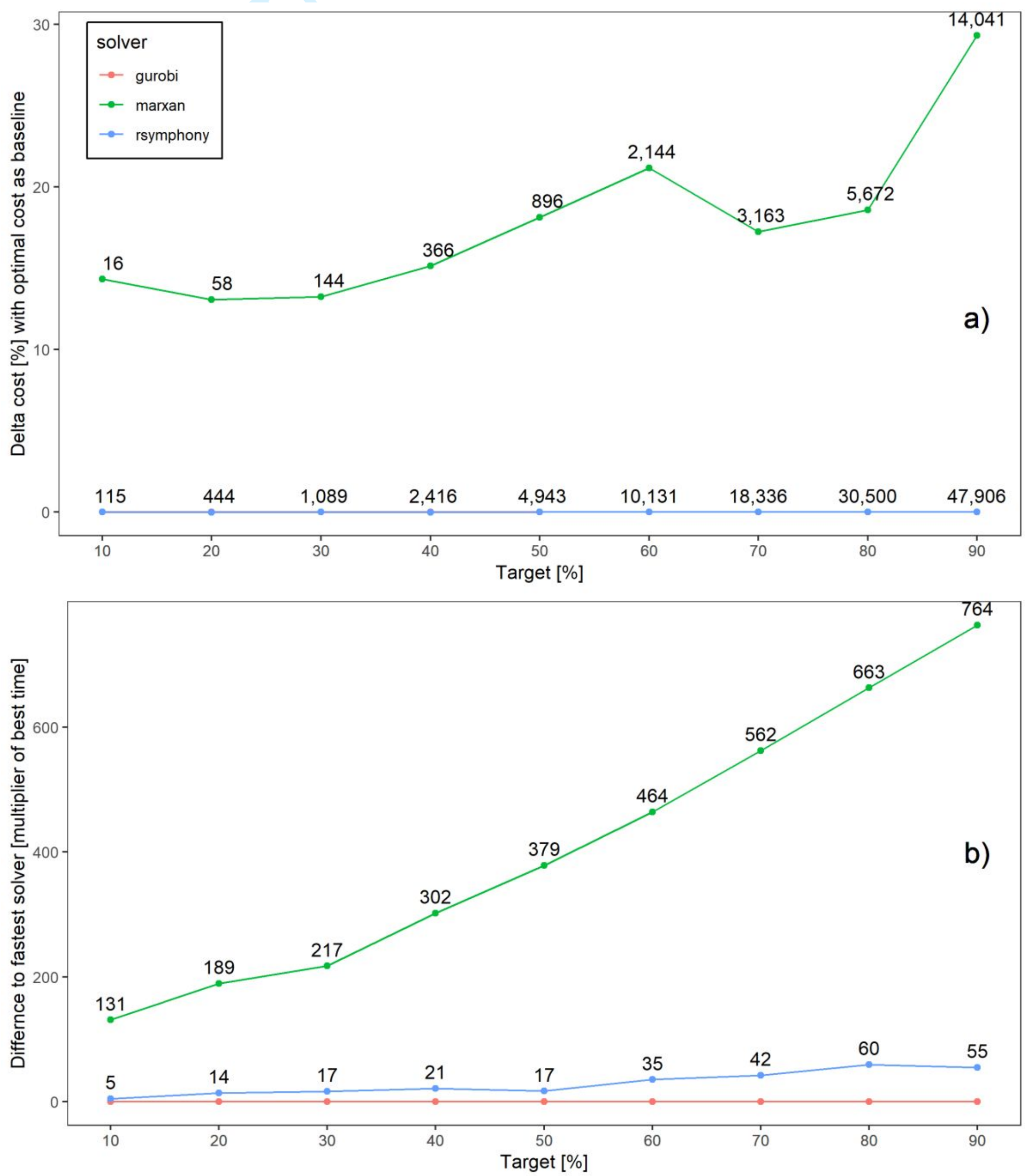
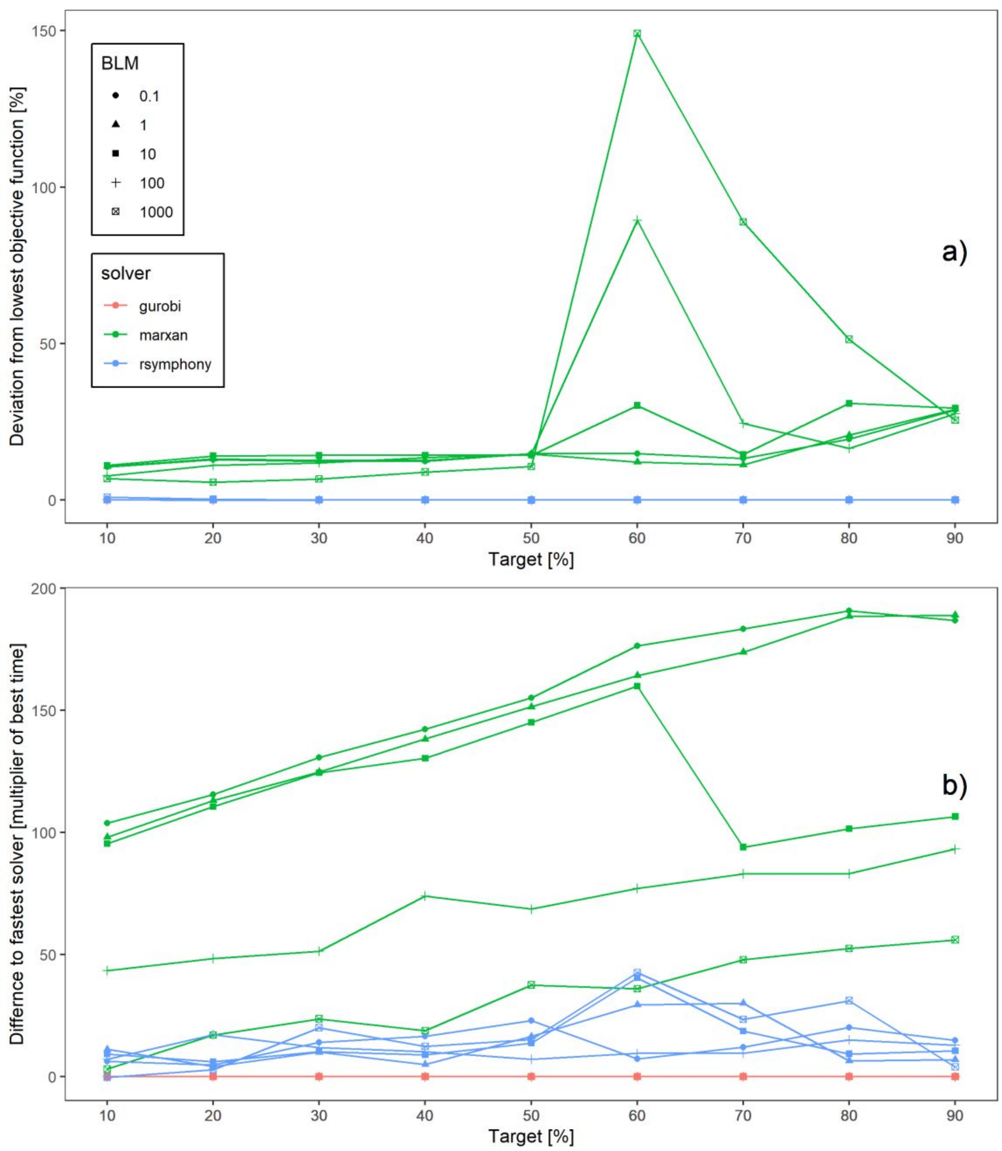


Figure 2. Objective function value and time comparisons using a boundary penalty to achieve spatially compact solutions. a) Deviation from lowest objective function value for solvers used and over a range of boundary penalty or boundary length modifier values (BLM); zero deviation indicates optimal solution. b) Time to solution comparisons between solvers and across BLM values. Note that in a) gurobi (red) and Rsymphony (blue) yielded optimal solutions for all target values and so their lines are plotted exactly on top of each other.



Supporting Information for

Integer Linear programming outperforms simulated annealing for solving conservation planning problems

Richard Schuster, Jeffrey O. Hanson, Matt Strimas-Mackey, Joseph R. Bennett

Table S1

Table S1: List of species that were used as features in our analysis.

Species Code	Common Name	Scientific Name
amegfi	American Goldfinch	Spinus tristis
amekes	American Kestrel	Falco sparverius
amerob	American Robin	Turdus migratorius
annhum	Anna's Hummingbird	Calypte anna
baleag	Bald Eagle	Haliaeetus leucocephalus
barswa	Barn Swallow	Hirundo rustica
brdowl	Barred Owl	Strix varia
belkin1	Belted Kingfisher	Megaceryle alcyon
bewwre	Bewick's Wren	Thryomanes bewickii
bnhcow	Brown-headed Cowbird	Molothrus ater
bkhgro	Black-headed Grosbeak	Pheucticus melanocephalus
brebla	Brewer's Blackbird	Euphagus cyanocephalus
brncre	Brown Creeper	Certhia americana
batpig1	Band-tailed Pigeon	Patagioenas fasciata
bushti	Bushtit	Psaltriparus minimus
cangoo	Canada Goose	Branta canadensis
chbchi	Chestnut-backed Chickadee	Poecile rufescens
cedwax	Cedar Waxwing	Bombycilla cedrorum
chispa	Chipping Sparrow	Spizella passerina
coohaw	Cooper's Hawk	Accipiter cooperii
comrav	Common Raven	Corvus corax
amecro	American Crow	Corvus brachyrhynchos
dowwoo	Downy Woodpecker	Dryobates pubescens
eucdov	Eurasian Collared-Dove	Streptopelia decaocto
eursta	European Starling	Sturnus vulgaris
evegro	Evening Grosbeak	Coccothraustes vespertinus
norfli	Northern Flicker	Colaptes auratus
foxspa	Fox Sparrow	Passerella iliaca
gockin	Golden-crowned Kinglet	Regulus satrapa
haiwoo	Hairy Woodpecker	Dryobates villosus
houfin	House Finch	Haemorhous mexicanus
houspa	House Sparrow	Passer domesticus
houwre	House Wren	Troglodytes aedon
hutvir	Hutton's Vireo	Vireo huttoni
macwar	MacGillivray's Warbler	Geothlypis tolmiei
moudov	Mourning Dove	Zenaida macroura
norhar1	Hen Harrier	Circus cyaneus
orcwar	Orange-crowned Warbler	Oreothlypis celata
olsfly	Olive-sided Flycatcher	Contopus cooperi
osprey	Osprey	Pandion haliaetus
pacwre1	Pacific Wren	Troglodytes pacificus
pinsis	Pine Siskin	Spinus pinus
pilwoo	Pileated Woodpecker	Dryocopus pileatus
pasfly	Pacific-slope Flycatcher	Empidonax difficilis
purfin	Purple Finch	Haemorhous purpureus
purmar	Purple Martin	Progne subis
rebnut	Red-breasted Nuthatch	Sitta canadensis
rebsap	Red-breasted Sapsucker	Sphyrapicus ruber

Species Code	Common Name	Scientific Name
redcro	Red Crossbill	Loxia curvirostra
rocpig	Rock Pigeon	Columba livia
rethaw	Red-tailed Hawk	Buteo jamaicensis
rufhum	Rufous Hummingbird	Selasphorus rufus
rewbla	Red-winged Blackbird	Agelaius phoeniceus
savspa	Savannah Sparrow	Passerculus sandwichensis
sora	Sora	Porzana carolina
sonspa	Song Sparrow	Melospiza melodia
spotow	Spotted Towhee	Pipilo maculatus
stejay	Steller's Jay	Cyanocitta stelleri
swathr	Swainson's Thrush	Catharus ustulatus
towwar	Townsend's Warbler	Setophaga townsendi
treswa	Tree Swallow	Tachycineta bicolor
daejun	Dark-eyed Junco	Junco hyemalis
yerwar	Yellow-rumped Warbler	Setophaga coronata
varthr	Varied Thrush	Ixoreus naevius
vigswa	Violet-green Swallow	Tachycineta thalassina
warvir	Warbling Vireo	Vireo gilvus
whcspa	White-crowned Sparrow	Zonotrichia leucophrys
westan	Western Tanager	Piranga ludoviciana
wilsni1	Wilson's Snipe	Gallinago delicata
wlswar	Wilson's Warbler	Cardellina pusilla
wooduc	Wood Duck	Aix sponsa
yelwar	Yellow Warbler	Setophaga petechia

Figure S1

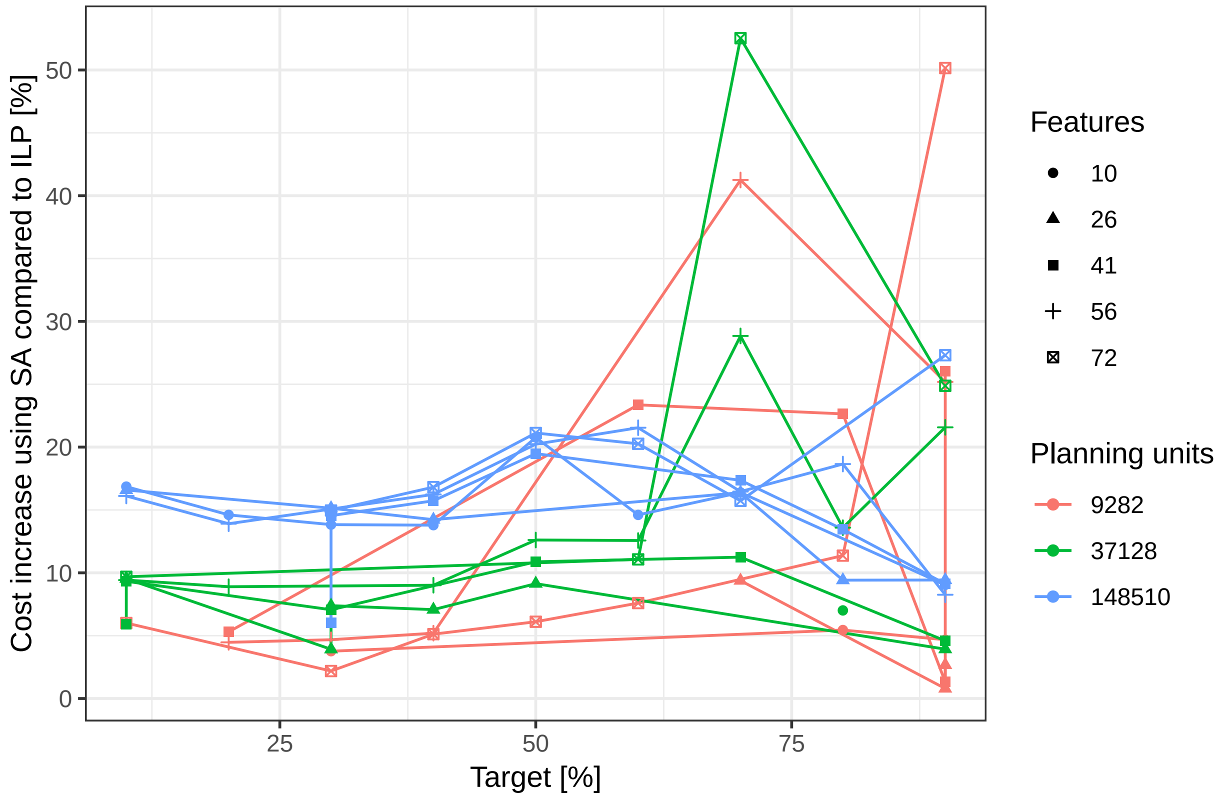


Figure S1: Percent cost increase of SA solutions compared to ILP solutions, across targets, number of features and number of planning units. Simulated annealing (i.e. Marxan) parameters used are: number of iterations > 100,000; species penalty factor 5 or 25.

Figure S2

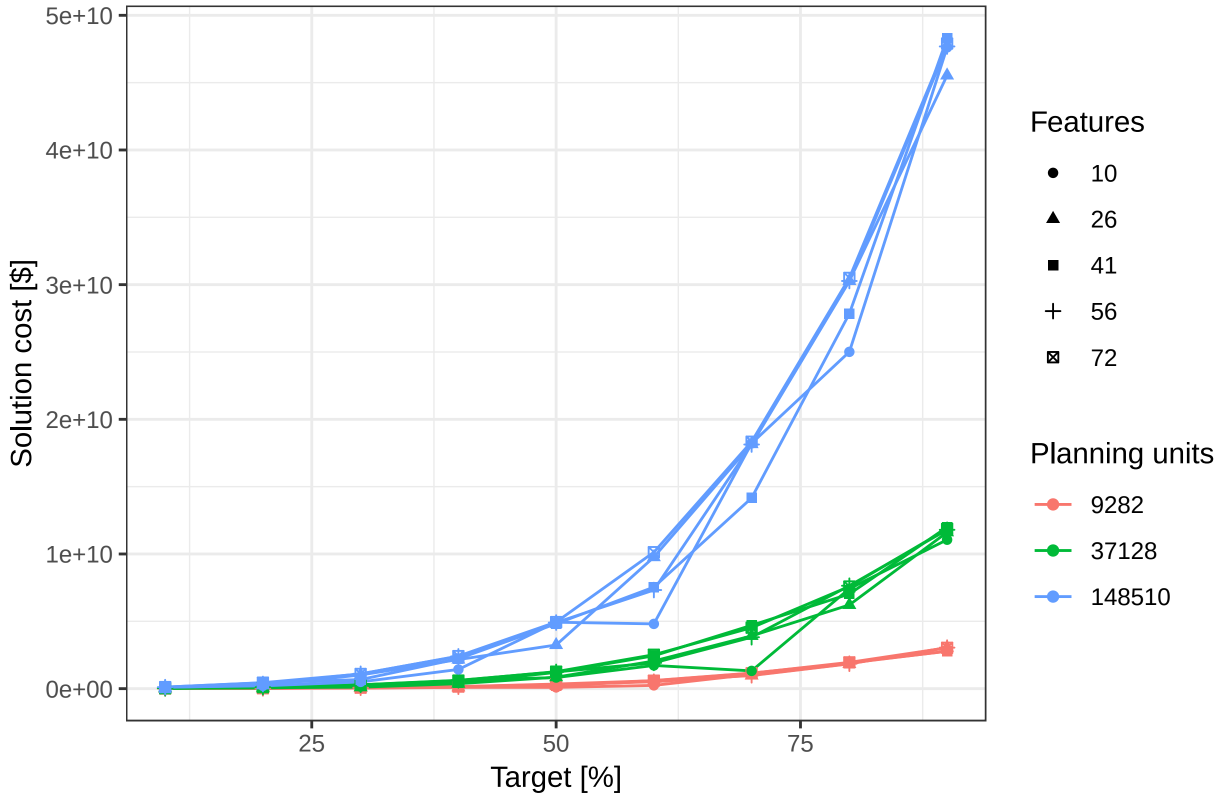


Figure S2: Cost profile for Gurobi solver across targets, number of features and number of planning units.

Figure S3

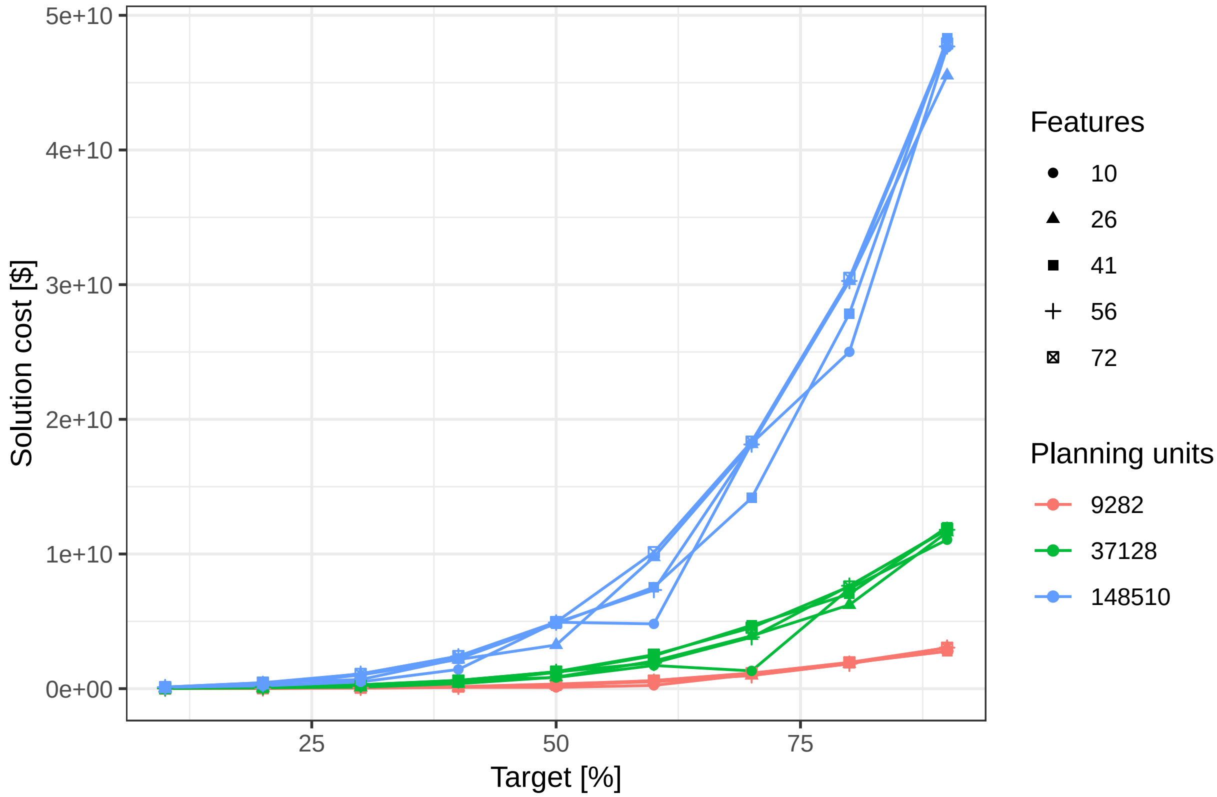


Figure S3: Cost profile for SYMPHONY solver across targets, number of features and number of planning units.

Figure S4

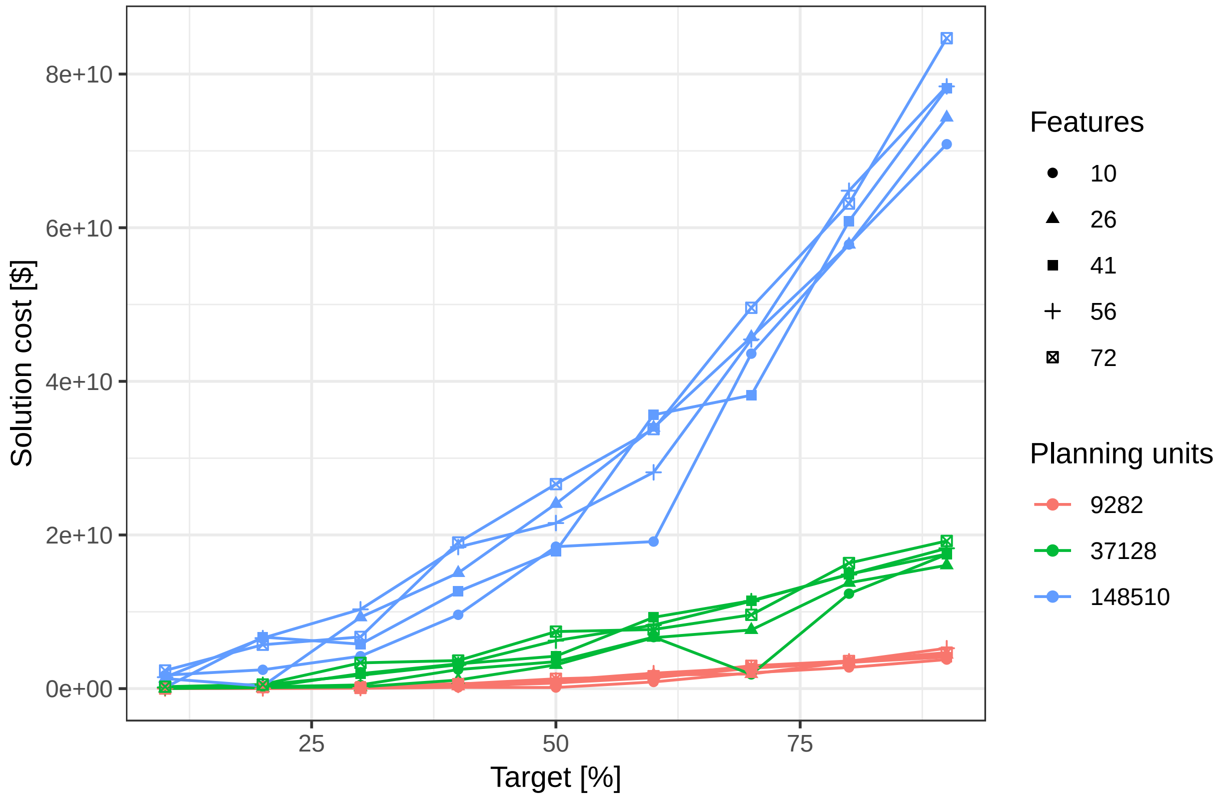


Figure S4: Cost profile for Marxan using Simulated Annealing across targets, number of features and number of planning units.

Figure S5

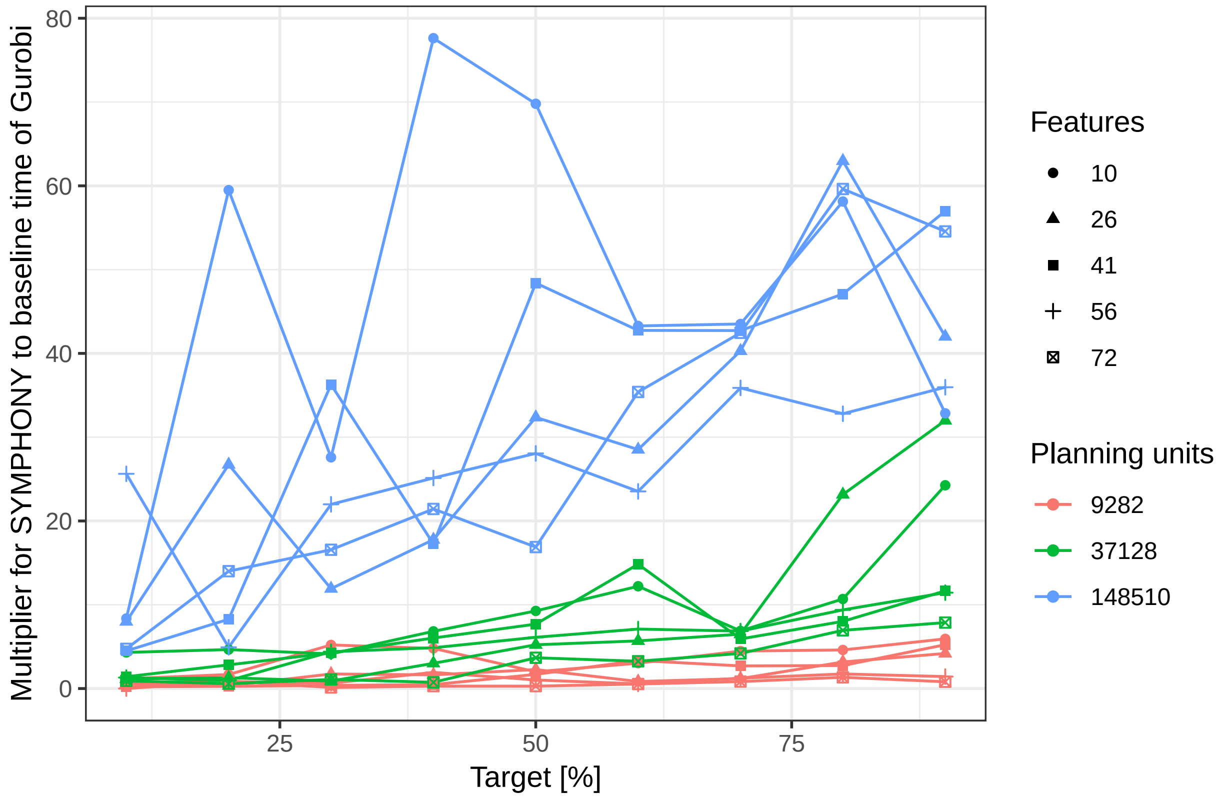


Figure S5: Time to solution comparisons between SYMPHONY and Gurobi across targets, number of features and number of planning units.

Figure S6

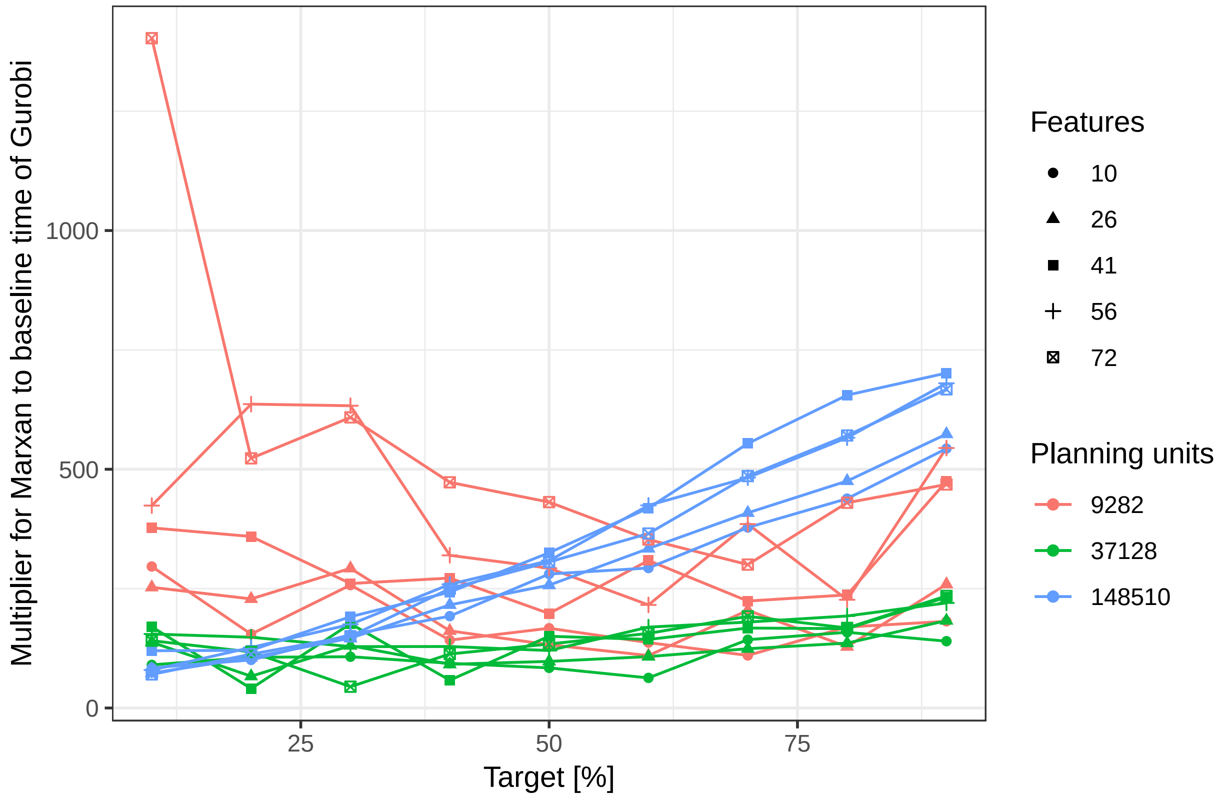


Figure S6: Time to solution comparisons between Marxan using Simulated Annealing and Gurobi across targets, number of features and number of planning units.

Figure S7

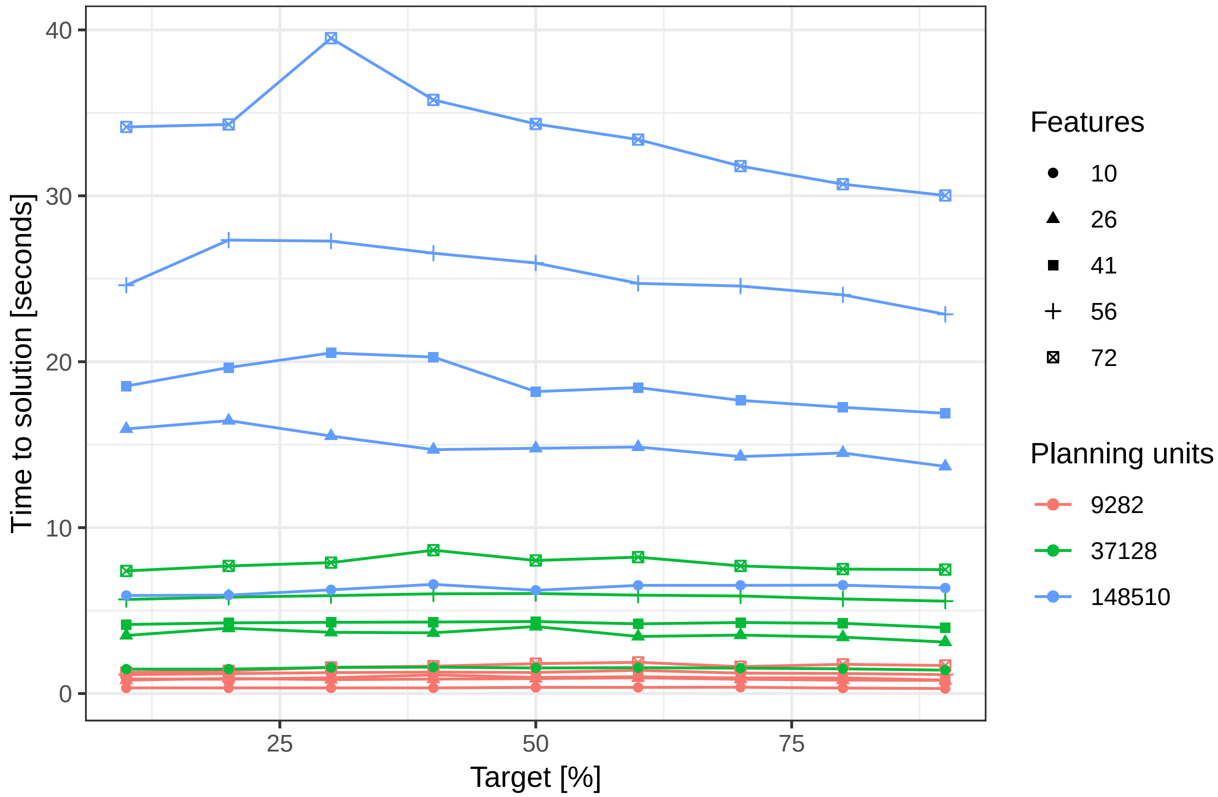


Figure S7: Time to solution profile for Gurobi solver across targets, number of features and number of planning units.

Figure S8

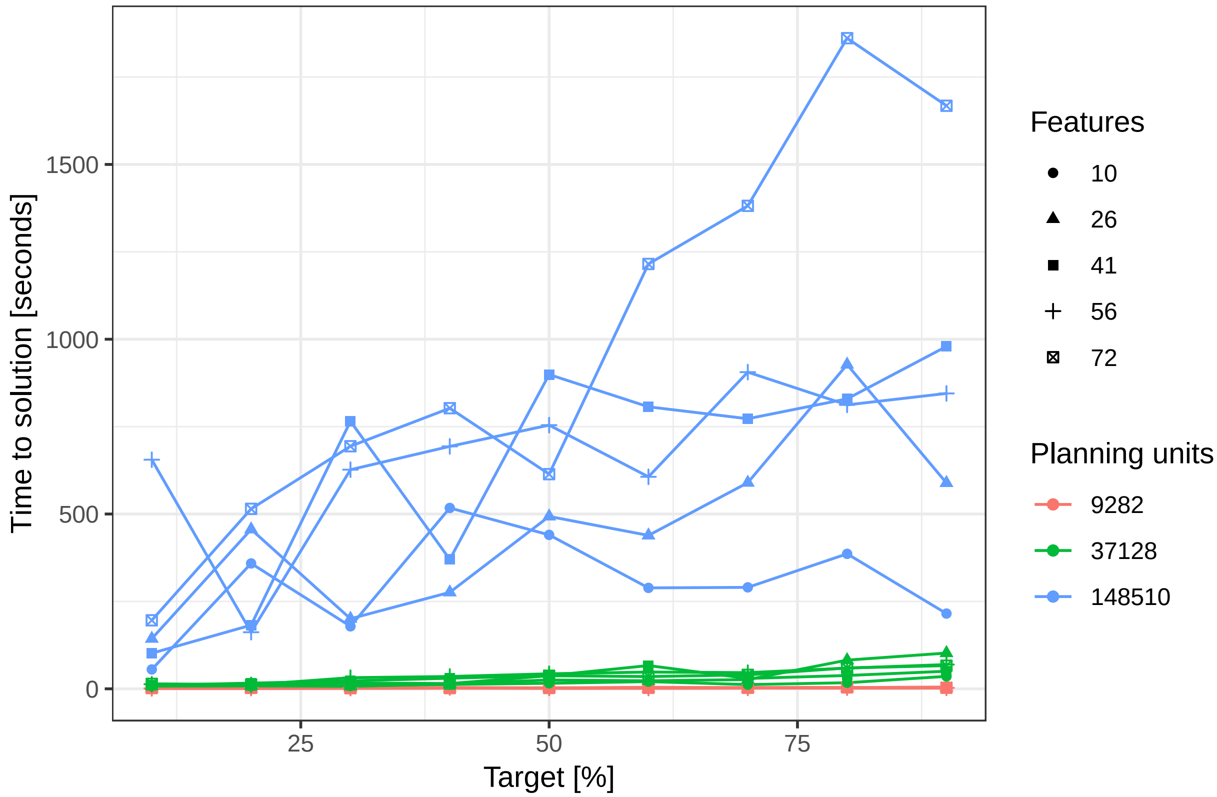


Figure S8: Time to solution profile for SYMPHONY solver across targets, number of features and number of planning units.

Figure S9

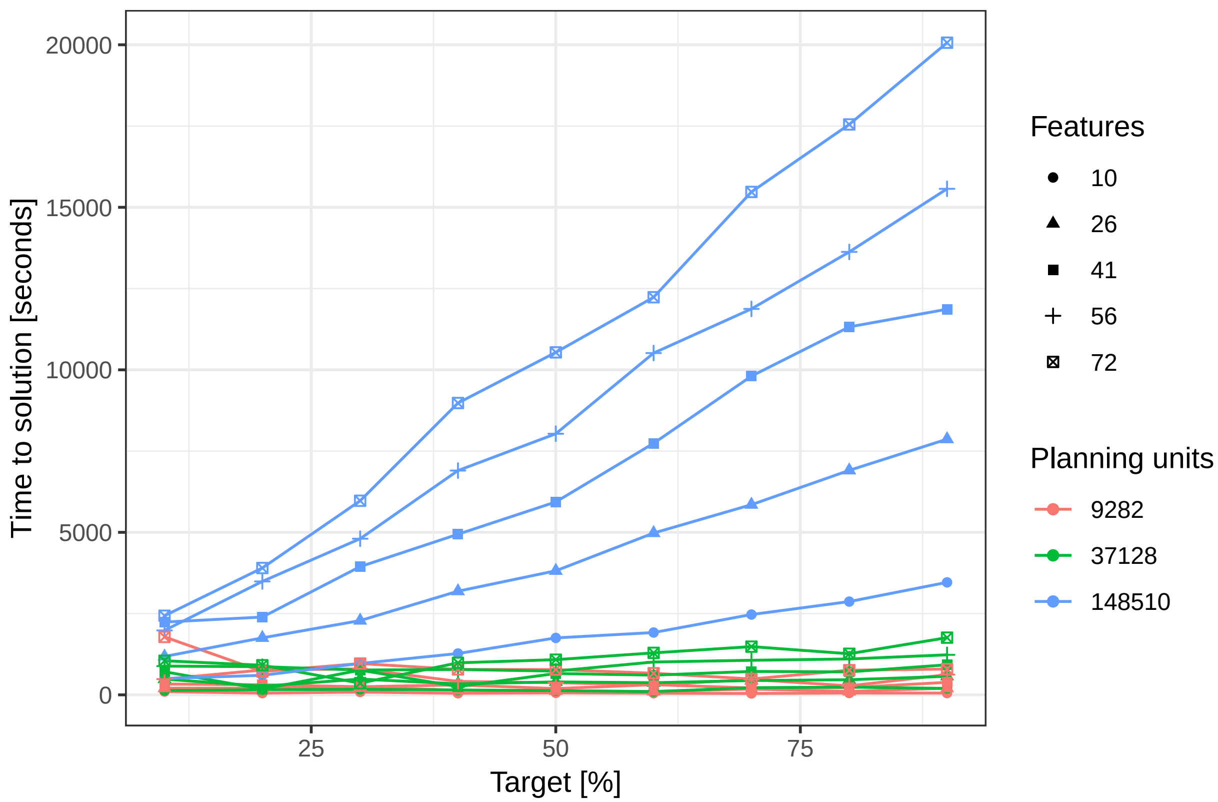


Figure S9: Time to solution profile for Marxan using Simulated Annealing across targets, number of features and number of planning units.

Figure S10

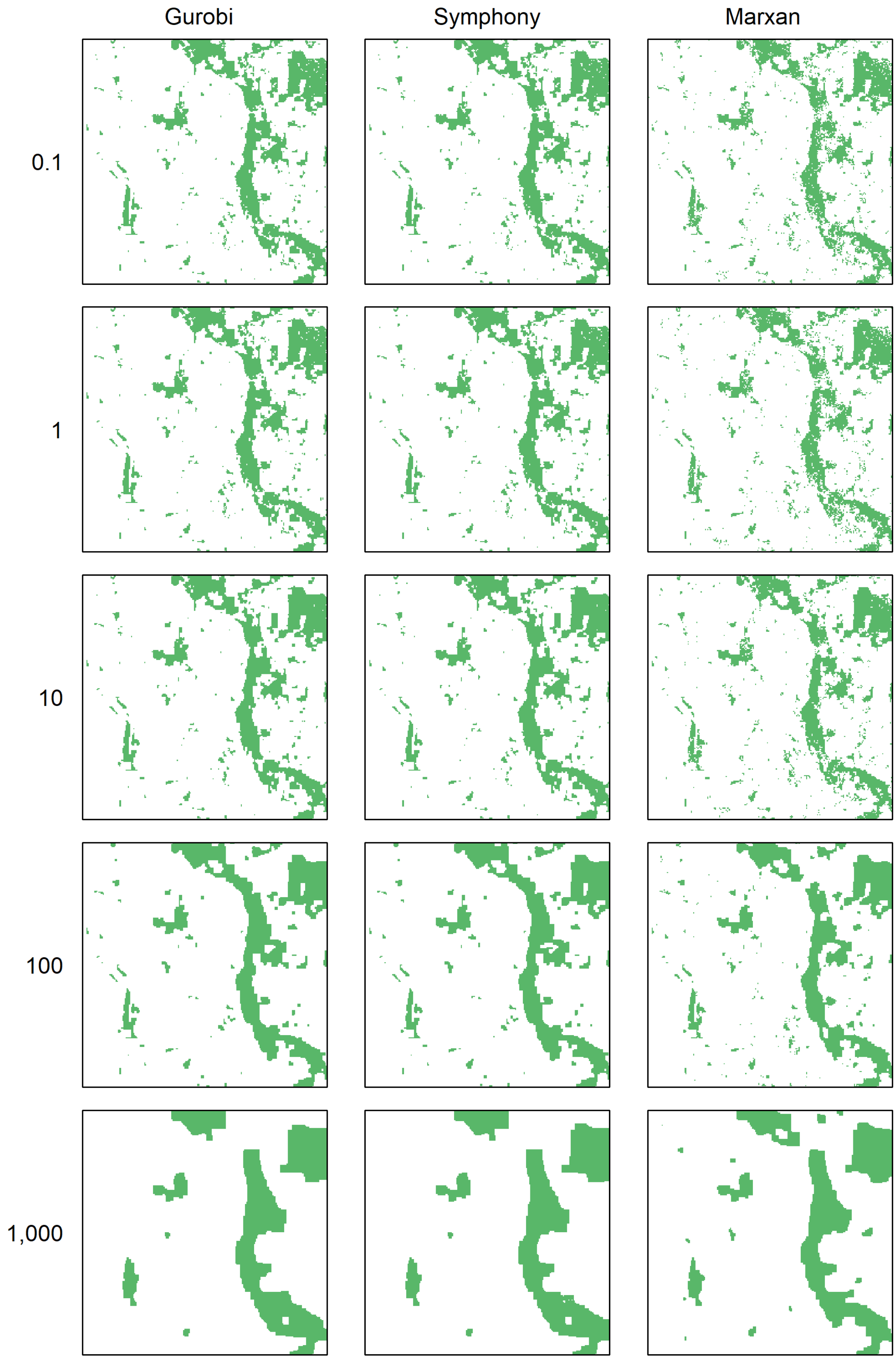


Figure S10: Compactness of solutions. Shown are the solutions for a 10% target. The numbers represent BLM values.