

Exact integer linear programming solvers outperform simulated annealing for solving conservation planning problems

Richard Schuster^{1,2}, Jeffrey O. Hanson³, Matthew Strimas-Mackey⁴ and Joseph R. Bennett¹

¹ Department of Biology, Carleton University, Ottawa, ON, Canada

² Ecosystem Science and Management Program, University of Northern British Columbia, Prince George, BC, Canada

³ School of Biological Sciences, University of Queensland, Brisbane, QLD, Australia

⁴ Cornell Lab of Ornithology, Cornell University, Ithaca, NY, USA

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 exact integer linear programming (EILP) solvers. Using a case study in BC, Canada, we compare the cost-effectiveness and processing times of SA used in Marxan versus EILP using both commercial and open-source algorithms. Plans for expanding protected area systems based on EILP algorithms were 12–30% cheaper than plans using SA, due to EILP's ability to find optimal solutions as opposed to approximations. The best EILP solver we examined was on average 1,071 times faster than the SA algorithm tested. The performance advantages of EILP solvers were also observed when we aimed for spatially compact solutions by including a boundary penalty. One practical advantage of using EILP over SA is that the analysis does not require calibration, saving even more time. Given the performance of EILP solvers, they can be used to generate conservation plans in real-time during stakeholder meetings and can facilitate rapid sensitivity analysis, and contribute to a more transparent, inclusive, and defensible decision-making process.

Submitted 16 January 2020

Accepted 8 May 2020

Published 27 May 2020

Corresponding author

Richard Schuster,
richard.schuster@glel.carleton.ca

Academic editor

Alison Boyer

Additional Information and
Declarations can be found on
page 10

DOI 10.7717/peerj.9258

© Copyright

2020 Schuster et al.

Distributed under

Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Biodiversity, Biogeography, Conservation Biology, Ecology, Spatial and Geographic Information Science

Keywords Conservation planning, Optimization, Prioritization, Integer linear programming, Prioritizr, Marxan

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 and cost-effectively enhance the long-term persistence of species or communities

(*Joppa & Pfaff, 2009; Venter et al., 2014*). Systematic conservation planning, on the other hand, is a multi-step process that involves framing conservation planning problems as optimization problems with clearly defined objectives (e.g., minimize acquisition cost) and constraints (*Margules & Pressey, 2000*). 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, Possingham & Watts, 2009*). Although Marxan supports several algorithms for solving conservation planning problems, most conservation planning exercises use its implementation of simulated annealing (SA), an iterative, stochastic metaheuristic algorithm for approximating global optima of complex functions (*Kirkpatrick, Gelatt & Vecchi, 1983*). By conducting thousands of simulations to determine the impact of different candidate solutions, Marxan aims to generate solutions that are near-optimal. One of the reasons why Marxan uses SA instead of exact integer linear programming (EILP) solvers, is that EILP solvers were historically 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) and generally took considerably longer than SA to solve problems (*Sarkar et al., 2006; Haight & Snyder, 2009*). However, the SA approach provides no guarantee on solution quality, and conservation scientists and practitioners have no way of knowing how close to optimal their solutions are. In this case, “optimal” refers to the configuration of protected areas that delivers the desired benefits and the lowest cost. The discussion about the relative merits of linear programming versus heuristics such as SA in conservation planning spans more than two decades (*Cocks & Baird, 1989; Underhill, 1994; Church, Stoms & Davis, 1996; Rodrigues & Gaston, 2002; Önal, 2004*), but the EILP shortcomings mentioned above have largely been overcome in recent years (*Beyer et al., 2016*).

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 EILP (*Wolsey & Nemhauser, 1999*), 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 metaheuristic methods such as SA, prioritization using EILP will find the optimal solution or can be instructed to return solutions within a defined level of suboptimality. Some have argued that EILP 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 EILP for large problems (*Haight & Snyder, 2009; Beyer et al., 2016*).

Here we compare EILP solvers with simulated annealing as used in Marxan, for solving minimum set systematic conservation planning problems (*Rodrigues, Cerdeira & Gaston, 2000*) using real-world data from Western North America. The goal of solving the minimum set problem is to find the places that maximize biodiversity, while minimizing reserve cost. We found that EILP generated high quality solutions 1,000 times faster than simulated annealing that could save over \$100 million (or 13%) for realistic conservation scenarios when compared to solutions obtained from simulated annealing. These results also hold true for problems aiming for spatially compact solutions. Our findings open up new possibilities for scenario generation to quickly explore and compare different conservation prioritization scenarios in real-time.

MATERIALS AND METHODS

Study area

We focused on a 27,250 km² 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*) (*Fig. S1*). Land cover in the region is diverse, with approximately 57% of the land in forest, 8% as savanna or grassland, 5% in cropland, 10% being urban or built and the rest in wetland, water or barren.

Biodiversity data

We used species distribution models for 72 bird species as our conservation features at a 1-ha grid cell resolution (*Table S1*). 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 from March to June to capture the breeding season, <1.5 h in duration, <5 km traveled, and a maximum of 10 visits to a given location to improve model fit. Sampling locations <100 m apart were collapsed to one location, yielding 5,470 checklists from 2,160 locations, visited from 1 to 10 times and 2.53 times on average. The R package unmarked (version 0.9-9; *Fiske & Chandler, 2011*) provided the framework for all species distribution models, which necessarily include two parts: occupancy and detection (*MacKenzie et al., 2002*). This form of distribution modeling, also known as occupancy modeling, uses the information from repeat visits to a site to infer estimates of detectability of a species as well as estimates of probability of occurrence. For further details on biodiversity data see *Rodewald et al. (2019)*.

Property layer and land cost

We incorporated spatial heterogeneity in land cost (*Ando et al., 1998; Polasky, Camm & Garber-Yonts, 2001; Ferraro, 2003; Naidoo et al., 2006*) in our plans by using property data and 2012 land value assessments from the Integrated Cadastral Information Society of BC. This process resulted in 193,623 properties for BC which were subsequently used as

planning units (*Schuster, Martin & Arcese, 2014*). Property 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 property layer included 1.92 million polygons. Property 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 property layer for Oregon included 605,425 polygons. We converted the polygon cost values to 1-ha raster cells for consistency with the biodiversity data by calculating area weighted mean values of cost per raster cell. Using tax assessment values as an estimate of conservation cost is an underestimate because tax assessment values are often lower than market value, but estimates of market values over larger areas are rarely available and tax assessments do provide a good general approximation.

Spatial prioritization

We compared EILP and SA for solving the minimum set spatial prioritization problem (*Ball, Possingham & Watts, 2009*). In this formulation, the landscape is divided into a set of discrete planning units. Each planning unit is assigned a financial 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). We also 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 financial 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 (i.e., min cost: such that conservation value \geq target). Details on the Marxan problem formulation can be found in *Ball, Possingham & Watts (2009)* and the EILP formulation in *Beyer et al. (2016)* and [Appendix S2](#). 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, Possingham & Klein, 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 (for more details see [Appendix S1](#)). For all scenarios, we used 1 km² planning units, generated by aggregating the species and cost data to this coarser resolution from the original 1-ha cells. Aggregation was accomplished by taking the sum of cost data and the mean of species data for all 1-ha cells within the larger 1 km² cells.

EILP solvers (commercial vs open source)

A variety of EILP solvers currently exist, and both commercial and open source solvers are available. All solvers yield optimal solutions to EILP problems, but there are substantial differences in performance (i.e., time taken to solve a problem) and in the size of problems

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 2,700 for that algorithm.

Parameter	Value range	Variations	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)

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, Dominic & Heiko, 2018*). 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 EILP 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 prioritizr we created the following range of scenarios: (i) vary conservation targets between 10 and 90% protection of features in 10% increments (nine variations), using (ii) 10–72 features (five variations) as targets, and (iii) with spatial extents of 9,282 planning units, 37,128 planning units, and 148,510 planning units (three 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 (five variations) and (ii) species penalty factors (SPF) of 1, 5, 25 and 125 were explored (four variations, roughly spanning two orders of magnitude) for a total of 2,700 scenarios investigated in Marxan (*Table 1*). Exploring ranges of values for number of iterations and SPF is recommended for calibration of Marxan to increase its ability to approximate the optimal solution (*Ardron, Possingham & Klein, 2010*). As the processing time for the most complex problem in Marxan (90% target, 72 features, 148,510 planning units, 10^8 iterations) was >8 h, 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 (*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, Possingham & Klein, 2010*). To allow for a fair contrast between SA and EILP that focuses on algorithmic comparisons and not within SA variation, we focused our results and discussion on the best solution achieved with Marxan across 10 repeat runs.

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 225×225 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. Both Marxan and prioritizr allow a user to specify BLM values as presented here. For details on the mathematical formulation of the spatial compactness constraint in ILP, please see Appendix S2 and Beyer *et al.* (2016).

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 online (<https://osf.io/my8pc/>) and will be archived in a persistent repository with a DOI pending acceptance of the manuscript.

RESULTS

Exact integer linear programming algorithms (Gurobi, SYMPHONY) outperformed SA (Marxan) in terms of their ability to find minimal cost solutions across all scenarios that met conservation targets. Summarizing across calibrated Marxan scenarios (number of iterations > 100,000 and species penalty factor 5 or 25), the range of savings ranged from 0.8% to 52.5% (median 12.6%, Fig. S2) when comparing EILP results to the best (cheapest) solution for a Marxan scenario. For example, at the 30% protection target EILP solvers resulted in solutions that were \$55 million cheaper than SA (Fig. 1A), because the EILP solvers selected cheaper and fewer parcels in the optimal solution. With these savings an additional 961 ha could be protected (13,897 ha vs 12,936 ha) using an EILP 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 (Fig. S2). Cost profiles across targets, number of features and number of planning units are shown in Figs. S3–S5.

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 (Fig. 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, Fig. S6), and Marxan took between 1.8 and 1,995 times longer than Gurobi (mean = 281 times, Fig. S7). The longest processing times for Gurobi, SYMPHONY and Marxan for a single scenario were 40 s, 31 min and 8 h respectively. For the most complex problem (i.e., targets = 90%, 72 features; 148,510 planning units), Marxan calibration across the five number of iterations and four species penalty factor values took a total of 5 days 7 h, compared to 30 s using Gurobi

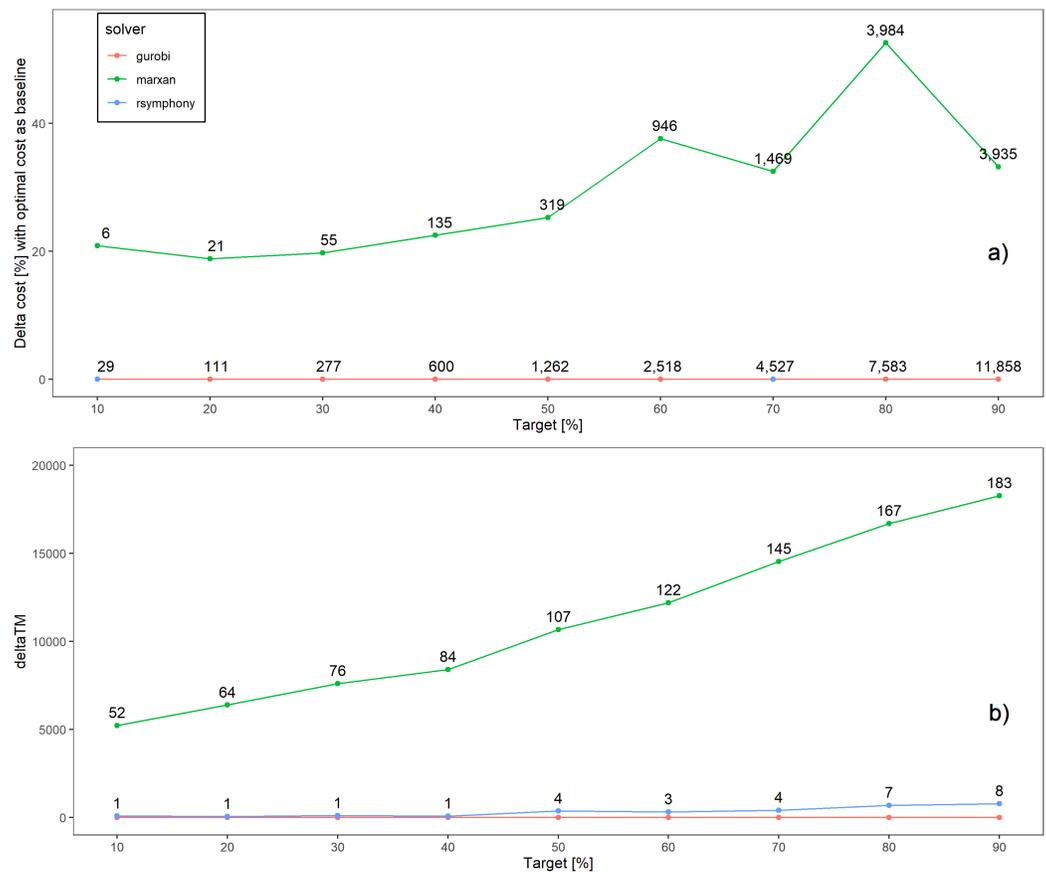


Figure 1 Solution cost and time comparisons. (A) The lines represent costs compared to the Gurobi cost baseline. The numbers on the blue line represent total cost of a solution in million \$ and the numbers on the green line represent how much more expensive, again in million \$, the SA/Marxan solution is compared to the ILP solutions. (B) Time to solution comparisons between solvers. Marxan parameters used are: 72 features, 37,128 planning units, 10^7 iterations, using mean cost and time, across all Marxan runs that met their target for a given scenario (max = 10). 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. Full-size [DOI: 10.7717/peerj.9258/fig-1](https://doi.org/10.7717/peerj.9258/fig-1)

and 28 min using SYMPHONY. Time profiles across targets, number of features and number of planning units are shown in Figs. S8–S10.

Exact integer linear programming algorithms (Gurobi, SYMPHONY) also outperformed SA (Marxan) when using a BLM to achieve more compact solutions. This was true for objective function values (Fig. 2A) as well as for processing times (Fig. 2B). Through finding optimal solutions, using EILP 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 3.09–190.8). An example of the spatial representation of the solutions for a 10% target is shown in Fig. S11.

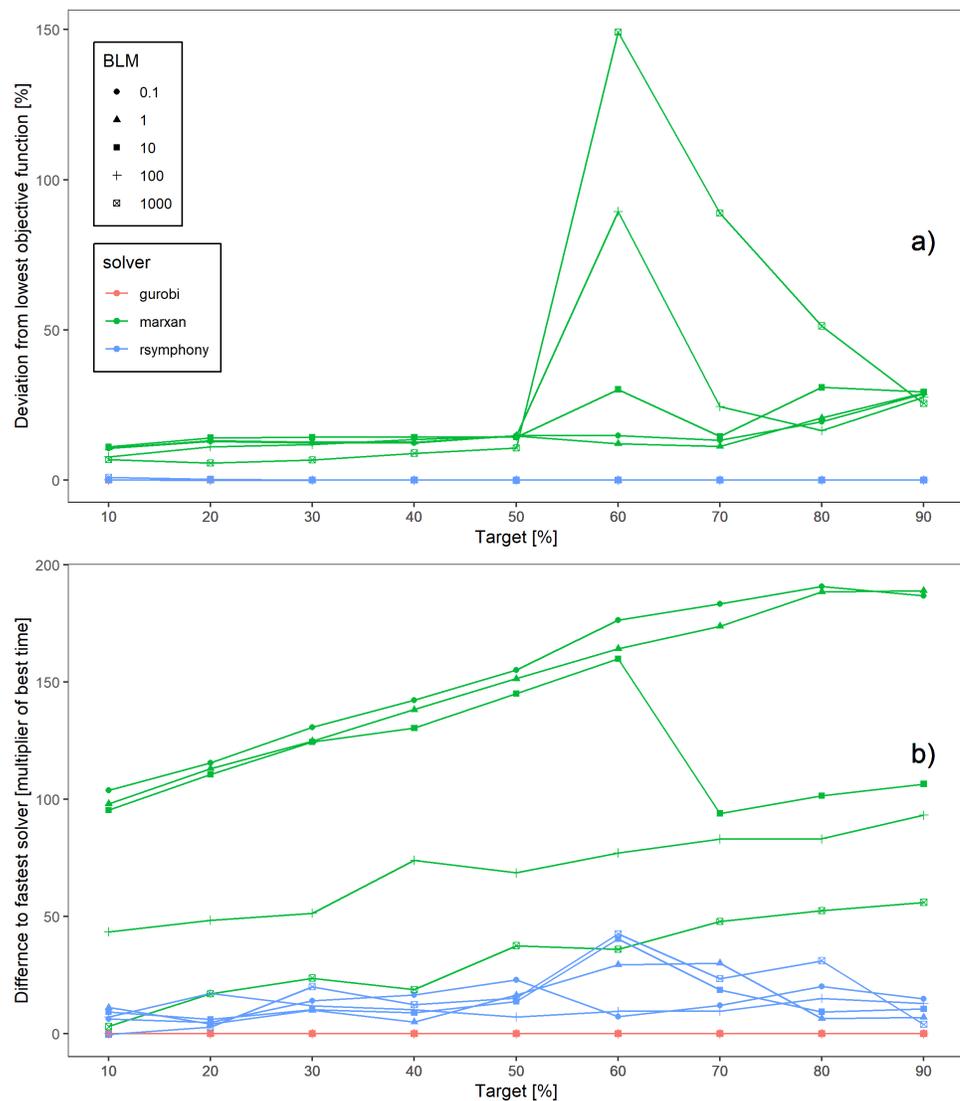


Figure 2 Objective function value and time comparisons using a boundary penalty to achieve spatially compact solutions. (A) Deviation from lowest objective function value for solvers used and over a range of boundary penalty or boundary length modifier values (BLM); zero deviation indicates optimal solution. (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. [Full-size !\[\]\(5fd6ef84f97f42d7f8b34275f1b65312_img.jpg\) DOI: 10.7717/peerj.9258/fig-2](https://doi.org/10.7717/peerj.9258/fig-2)

DISCUSSION

We found that EILP algorithms outperformed SA both in terms of cost-effectiveness and processing times, even when including linearized non-linear problem formulations, when planning for spatially compact solutions. There have been calls for using EILP in solving conservation planning problems in the past (*Underhill, 1994, Rodrigues & Gaston, 2002*), but we are now at a point where making this switch is both advisable and computationally feasible, where technical capacity exists. Our study provides a systematic test, using real world data to build on the findings of (*Beyer et al., 2016*), and shows that

their results hold for a realistic case study. We further expanded the scope of testing to include assessed land values in order to give estimates of how much better optimal solution can perform in terms of cost savings, compared to SA solutions. Finally, we showcase that even open source EILP solvers are much faster than SA algorithms as implemented in Marxan, which is very encouraging for non-academic user that would otherwise have to buy Gurobi licenses (Gurobi is free for academic use). The combination of the superior performance findings by both ([Beyer et al., 2016](#)) and this study indicates that EILP approaches should be strongly considered as improvements for minimum set conservation planning problems, currently solved using SA. This improvement is especially important in real world applications as the speed of generating solutions can be advantageous in iterative and dynamic planning processes that usually occur when planning for conservation ([Sarkar et al., 2006](#)). Given Marxan's flexibility to use optimization methods other than SA, we hope that a future version of Marxan will include EILP solvers.

One practical advantage of using EILP over SA is that the analysis does not require parameter calibration. Unlike EILP, parameter calibration is a crucial task in every Marxan/SA project and the species penalty factors, number of SA iterations, and number of SA restarts must be calibrated to improve solution quality ([Ardrone, Possingham & Klein, 2010](#)). This task can be very time consuming, especially for larger problems (e.g., 50,000 planning units). Ideally all possible combinations of parameters should be explored, but this further increases processing time. For instance, exploring three different parameter values would result in 27 different scenarios to explore (i.e., $3 \times 3 \times 3$). Although we omitted calibration runs prior to finalizing and presenting results in this study, the parameter calibration step took several days for the most complex problem we investigated in this study. Yet none of this calibration time is necessary using EILP. An added benefit is that the somewhat subjective process of setting values for these three parameters can be eliminated using EILP as well.

Recommended practices for Marxan analyses caution against using SA for conservation planning exercises with more than 50,000 planning units ([Ardrone, Possingham & Klein, 2010](#)). Such large-sized problems have occurred in the past and, as increasingly high resolution data become available, may become more common in the future ([Venter et al., 2014](#); [Runge et al., 2016](#)). Unlike SA, EILP/prioritizr can solve problem sizes with more than one million planning units ([Hanson, 2018](#); [Schuster et al., 2019](#)). Realistically, as problem sizes grow beyond what was intended for Marxan/SA projects, EILP 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 EILP to work. A potential future solution to this issue could be the use of nonlinear integer programming for more problems including non-linear constraints ([Grossmann, 2002](#); [Lee & Leyffer, 2011](#)). Whether EILP would also outperform SA for more complex problem formulations, such as dynamic problems or problems with multiple objectives, still needs to be explored. Potential solutions would be to linearize the problem, or incorporate algorithms like Mixed Integer Quadratically Constrained Programming ([Franco, Rider & Romero, 2014](#)).

Finally, we argue that another strength of EILP 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

Exact integer linear programming algorithms substantially outperform SA as used in minimum set 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 EILP 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 EILP is showing for conservation planning, we recommend users consider adding this modified approach to solving systematic conservation planning problems.

ACKNOWLEDGEMENTS

We thank W. Hochachka for providing code for processing eBird data and three reviewers for insightful comments.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Richard Schuster is supported by a Liber Ero Fellowship and Environment and Climate Change Canada (ECCC), Jeffrey O Hanson by ECCC, Matthew Strimas-Mackey by endowments at the Cornell Lab of Ornithology, and Joseph R. Bennett by Natural Sciences and Engineering Research Council of Canada and ECCC. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Liber Ero Fellowship and Environment and Climate Change Canada (ECCC).
Cornell Lab of Ornithology.
Natural Sciences and Engineering Research Council of Canada and ECCC.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Richard Schuster conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Jeffrey O. Hanson conceived and designed the experiments, performed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.

- Matthew Strimas-Mackey conceived and designed the experiments, performed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Joseph R. Bennett conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

All data, scripts and full results are available at OSF: *Schuster et al. (2019)*: “Exact integer linear programming solvers outperform simulated annealing for solving conservation planning problems”. OSF. dataset. <https://osf.io/my8pc/>.

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj.9258#supplemental-information>.

REFERENCES

- Ando A, Camm J, Polasky S, Solow A. 1998.** Species distributions, land values, and efficient conservation. *Science* **279**:2126–2128.
- Ardron JA, Possingham HP, Klein CJ. 2010.** *Marxan good practices handbook version 2*. Victoria: Pacific Marine Analysis and Research Association.
- Ball IR, Possingham HP, Watts MW. 2009.** Marxan and relatives: software for spatial conservation prioritisation. In: Moilanen A, Wilson KA, Possingham HP, eds. *Spatial Conservation Prioritisation: Quantitative Methods and Computational Tools*. Oxford: Oxford University Press, 185–195.
- Beyer HL, Dujardin Y, Watts ME, Possingham HP. 2016.** Solving conservation planning problems with integer linear programming. *Ecological Modelling* **328**:14–22 DOI [10.1016/j.ecolmodel.2016.02.005](https://doi.org/10.1016/j.ecolmodel.2016.02.005).
- Church RL, Stoms DM, Davis FW. 1996.** Reserve selection as a maximal covering location problem. *Biological Conservation* **76**(2):105–112 DOI [10.1016/0006-3207\(95\)00102-6](https://doi.org/10.1016/0006-3207(95)00102-6).
- Cocks KD, Baird IA. 1989.** Using mathematical programming to address the multiple reserve selection problem: an example from the Eyre Peninsula, South Australia. *Biological Conservation* **49**(2):113–130 DOI [10.1016/0006-3207\(89\)90083-9](https://doi.org/10.1016/0006-3207(89)90083-9).
- Ferraro PJ. 2003.** Assigning priority to environmental policy interventions in a heterogeneous world. *Journal of Policy Analysis and Management* **22**(1):27–43 DOI [10.1002/pam.10094](https://doi.org/10.1002/pam.10094).
- Fiske IJ, Chandler RB. 2011.** Unmarked : an R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software* **43**(10):128–129 DOI [10.18637/jss.v043.i10](https://doi.org/10.18637/jss.v043.i10).
- Franco JF, Rider MJ, Romero Rén. 2014.** A mixed-integer quadratically-constrained programming model for the distribution system expansion planning. *International Journal of Electrical Power & Energy Systems* **62**:265–272 DOI [10.1016/j.ijepes.2014.04.048](https://doi.org/10.1016/j.ijepes.2014.04.048).
- Grossmann IE. 2002.** Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering* **3**(3):227–252 DOI [10.1023/A:1021039126272](https://doi.org/10.1023/A:1021039126272).
- Gurobi Optimization Inc. 2017.** *Gurobi optimizer reference manual*. Version 7.5.1. Available at <https://www.gurobi.com/>.

- Haight RG, Snyder SA. 2009.** Integer programming methods for reserve selection and design. In: Moilanen A, Wilson KA, Possingham H, eds. *Spatial Conservation Prioritization: Quantitative Methods and Computational Tools*. Oxford: Oxford University Press, 43–57.
- Hanson J. 2018.** Conserving evolutionary processes. Available at <https://espace.library.uq.edu.au/view/UQ:27bdf86>.
- Hanson J. 2019.** *prioritizr: systematic conservation prioritization in R*. Version 4.0.2. Available at <https://prioritizr.net/>.
- Harter R, Hornik K, Theussl S, Szymanski C, Schwendinger F. 2017.** Rsymphony: SYMPHONY in R. Available at <http://r-forge.r-project.org/projects/rsymphony>.
- Hochachka WM, Fink D, Hutchinson RA, Sheldon D, Wong W-K, Kelling S. 2012.** Data-intensive science applied to broad-scale citizen science. *Trends in Ecology & Evolution* 27(2):130–137 DOI 10.1016/j.tree.2011.11.006.
- Joppa LN, Pfaff A. 2009.** High and far: biases in the location of protected areas. *PLOS ONE* 4(12):e8273 DOI 10.1371/journal.pone.0008273.
- Kirkpatrick S, Gelatt CD, Vecchi MP. 1983.** Optimization by simulated annealing. *Science* 220(4598):671–680 DOI 10.1126/science.220.4598.671.
- Lee J, Leyffer S. 2011.** *Mixed integer nonlinear programming*. Berlin: Springer Science & Business Media.
- Lin CY, Liu JWS, Yeh KL, Chu ETH. 2017.** Participant selection problem: relative performance of five optimization solvers. In: *Proceedings of the 8th International Conference on Computer Modeling and Simulation*. 24–31.
- Luppold A, Dominic O, Heiko F. 2018.** *Evaluating the performance of solvers for integer-linear programming*. (in press).
- MacKenzie DI, Nichols JD, Lachman GB, Droege S, Andrew Royle J, Langtimm CA. 2002.** Estimating site occupancy rates when detection probabilities are less than one. *Ecology* 83(8):2248–2255 DOI 10.1890/0012-9658(2002)083[2248:ESORWD]2.0.CO;2.
- Margules CR, Pressey RL. 2000.** Systematic conservation planning. *Nature* 405(6783):243–253 DOI 10.1038/35012251.
- McDonnell MD, Possingham HP, Ball IR, Cousins EA. 2002.** Mathematical methods for spatially cohesive reserve design. *Environmental Modeling and Assessment* 7(2):107–114 DOI 10.1023/A:1015649716111.
- McIntosh EJ, Pressey RL, Lloyd S, Smith RJ, Grenyer R. 2017.** The impact of systematic conservation planning. *Annual Review of Environment and Resources* 42(1):677–697 DOI 10.1146/annurev-environ-102016-060902.
- Meidinger D, Pojar J. 1991.** *Ecosystems of british columbia*. Columbia: British Columbia Ministry of Forests.
- Naidoo R, Balmford A, Ferraro P, Polasky S, Ricketts T, Rouget M. 2006.** Integrating economic costs into conservation planning. *Trends in Ecology & Evolution* 21(12):681–687 DOI 10.1016/j.tree.2006.10.003.
- Önal H. 2004.** First-best, second-best, and heuristic solutions in conservation reserve site selection. *Biological Conservation* 115(1):55–62 DOI 10.1016/S0006-3207(03)00093-4.
- Polasky S, Camm JD, Garber-Yonts B. 2001.** Selecting biological reserves cost-effectively: an application to terrestrial vertebrate conservation in oregon. *Land Economics* 77(1):68–78 DOI 10.2307/3146981.
- Pressey RL, Bottrill MC. 2008.** Opportunism, threats, and the evolution of systematic conservation planning. *Conservation Biology* 22(5):1340–1345 DOI 10.1111/j.1523-1739.2008.01032.x.

- Pressey RL, Humphries CJ, Margules CR, Vane-Wright RI, Williams PH. 1993. Beyond opportunism: key principles for systematic reserve selection. *Trends in Ecology & Evolution* 8(4):124–128 DOI 10.1016/0169-5347(93)90023-I.
- Ralphs T, Mahajan A, Vigerske S, Bulut A. 2019. coin-or/SYMPHONY: Version 5.6.17. Zenodo. Available at <https://zenodo.org/record/2576603>.
- Rodewald AD, Strimas-Mackey M, Schuster R, Arcese P. 2019. Tradeoffs in the value of biodiversity feature and cost data in conservation prioritization. *Scientific Reports* 9(1):1–8 DOI 10.1038/s41598-019-52241-2.
- Rodrigues AS, Cerdeira JO, Gaston KJ. 2000. Flexibility, efficiency, and accountability: adapting reserve selection algorithms to more complex conservation problems. *Ecography* 23(5):565–574 DOI 10.1034/j.1600-0587.2000.230507.x.
- Rodrigues ASL, Gaston KJ. 2002. Optimisation in reserve selection procedures—why not? *Biological Conservation* 107(1):123–129 DOI 10.1016/S0006-3207(02)00042-3.
- Runge CA, Tulloch AIT, Possingham HP, Tulloch VJD, Fuller RA, Roura-Pascual N. 2016. Incorporating dynamic distributions into spatial prioritization. *Diversity and Distributions* 22(3):332–343 DOI 10.1111/ddi.12395.
- Sarkar S, Pressey RL, Faith DP, Margules CR, Fuller T, Stoms DM, Moffett A, Wilson KA, Williams KJ, Williams PH, Andelman S. 2006. Biodiversity conservation planning tools: present status and challenges for the future. *Annual Review of Environment and Resources* 31(1):123–159 DOI 10.1146/annurev.energy.31.042606.085844.
- Schuster R, Martin TG, Arcese P. 2014. Bird community conservation and carbon offsets in Western North America. *PLOS ONE* 9(6):e99292 DOI 10.1371/journal.pone.0099292.
- Schuster R, Wilson S, Rodewald AD, Arcese P, Fink D, Auer T, Bennett JR. 2019. Optimizing the conservation of migratory species over their full annual cycle. *Nature Communications* 10(1):1754 DOI 10.1038/s41467-019-09723-8.
- Schwartz MW, Cook CN, Pressey RL, Pullin AS, Runge MC, Salafsky N, Sutherland WJ, Williamson MA. 2018. Decision support frameworks and tools for conservation. *Conservation Letters* 11(2):e12385 DOI 10.1111/conl.12385.
- Sullivan BL, Aycrigg JL, Barry JH, Bonney RE, Bruns N, Cooper CB, Damoulas T, Dhondt AA, Dietterich T, Farnsworth A, Fink D, Fitzpatrick JW, Fredericks T, Gerbracht J, Gomes C, Hochachka WM, Iliff MJ, Lagoze C, La Sorte FA, Merrifield M, Morris W, Phillips TB, Reynolds M, Rodewald AD, Rosenberg KV, Trautmann NM, Wiggins A, Winkler DW, Wong W-K, Wood CL, Yu J, Kelling S. 2014. The eBird enterprise: an integrated approach to development and application of citizen science. *Biological Conservation* 169:31–40 DOI 10.1016/j.biocon.2013.11.003.
- Underhill LG. 1994. Optimal and suboptimal reserve selection algorithms. *Biological Conservation* 70(1):85–87 DOI 10.1016/0006-3207(94)90302-6.
- Venter O, Fuller RA, Segan DB, Carwardine J, Brooks T, Butchart SHM, Di Marco M, Iwamura T, Joseph L, O’Grady D, Possingham HP, Rondinini C, Smith RJ, Venter M, Watson JEM, Moritz C. 2014. Targeting global protected area expansion for imperiled biodiversity. *PLOS Biology* 12(6):e1001891 DOI 10.1371/journal.pbio.1001891.
- Wolsey LA, Nemhauser GL. 1999. *Integer and combinatorial optimization*. New Jersey: John Wiley & Sons.