

LP as a Global Search Heuristic Across Different Constrainedness Regions* Extended Abstract

Student: Lucian Leahu
Advisor: Carla Gomes

Dpt. of Computer Science, Cornell University, Ithaca, NY 14853, USA,
{l1eahu,gomes}@cs.cornell.edu

Abstract. We study the behavior of heuristics based on the LP relaxation with respect to the underlying constrainedness of the problem. Our study focuses on the Latin square (or quasigroup) completion problem [1]¹ as a prototype for highly combinatorial problems. We find that simple techniques based on the LP relaxation of the problem provide satisfactory guidance for under- and over-constrained instances. In the critically constrained region, the performance of such simple techniques degrades, due to the inherent hardness of the problem. In this setting, we examine a technique that recomputes the LP relaxation every time a variable is set. This leads to a significant increase in performance, suggesting that carefully designed “one step at a time” LP-based heuristics could provide suitable guidance even for the hardest instances.

Recent years have witnessed the emergence of a new area involving hybrid solvers integrating CP- and OR-based methods. OR has a long and rich history of using Linear Programming (LP) based relaxations for (Mixed) Integer Programming problems. In this approach, the LP relaxation provides bounds on overall solution quality and can be used for pruning in a branch-and-bound approach. This is particularly true in domains where we have a combination of linear constraints, well-suited for linear programming (LP) formulations, and discrete constraints, suited for constraint satisfaction problem (CSP) formulations. Nevertheless, in a *purely combinatorial* setting, so far it has been surprisingly difficult to integrate LP-based and CSP-based techniques. For example, despite significant LP results for Boolean satisfiability (SAT) problems (see e.g., [2–5]), practical state-of-the-art solvers do not yet incorporate LP relaxation techniques.

We examine the quality of the guidance provided by the LP relaxation as a function of the structure of the problem, i.e., we attempt to characterize the performance of LP heuristics across different constrainedness regions in the search space.

The problem used for our study is the Latin square completion problem. A Latin square is an n by n matrix, where each cell has one of n symbols (or colors), such that each symbol occurs exactly once in each row and column. Given a partial coloring of the n by n cells of a Latin square, determining whether there

* Research supported by the Intelligent Information Systems Institute, Cornell University (AFOSR grant F49620-01-1-0076).

¹ The multiplication table of a quasigroup is a Latin square.

is a valid completion into a full Latin square is an NP-complete problem [6]. The underlying structure of this problem is similar to that found in a series of real-world applications, such as timetabling, experimental design, and fiber optics routing problems [7, 8].

We start by considering the LP assignment formulation [8]. In this formulation, we have n^3 variables, some of them with pre-assigned values. Each variable, x_{ijk} ($i, j, k = 1, 2 \dots, n$), is a 0/1 variable that takes the value 1 if cell (i, j) is colored with color k . The objective function is to maximize the total number of colored cells in the Latin square:

$$\max \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n x_{ijk}$$

A natural bound for the objective function is therefore the number of cells in the Latin squares, i.e., n^2 . In the LP relaxation, we relax the constraint that the variables have to be integer, and therefore each variable can take its value in the interval $[0, 1]$.

We consider a variant of the problem of completing Latin squares, referred to as Latin squares (or quasigroup) with holes. In this variant, we start out with a full (valid) Latin square and we randomly “erase” some of the entries (cells). We refer to the erased cells as “holes”. Since we started with a full Latin square, we know the instance obtained by erasing certain cells is completable.

Figure 1 (top) depicts the “easy-hard-easy” pattern in complexity for the Latin Square Completion problem. The x-axis corresponds to the holes density². The y-axis represents the average number of backtracks needed to solve an instance having a particular hole density³. The left hand side of the plot corresponds to a small number of holes (i.e., many pre-assigned cells), hence the problem is over-constrained. This is an “easy” region, since the solver only needs to color a few holes, and the options available are straight forward. In the right hand side of the plot the number of holes is large (few pre-assigned cells), therefore the problem is under-constrained. Since there are many color options for each hole, this is also an “easy” region for a solver, as the number of solutions is large. The region in the middle corresponds to the critically constrained region: the options for each hole are too many for the solver to be quickly guided to a solution (as is the case in the over-constrained region) and too few to permit almost any assignment to lead to a solution (as in the under-constrained region). This is the area where the cost in complexity peaks, as wrong branching decisions tend to be very expensive (i.e., they direct the search to a large inconsistent subtree). The number of backtracks required to solve these instances is high, correlated with the inherent hardness of the instances.

Figure 1 (bottom) plots the phase transition in the solution integrality of the LP relaxation based on the assignment formulation. The y-axis represents

² The density of holes is Number of Holes/ $n^{1.55}$. Note that if the denominator were n^2 , we could talk about percentage of holes, however the complexity curves would not peak at the same ratio. As shown in [9], the complexity curves align when using $n^{1.55}$ as the denominator.

³ Each data point in this plot was generated by computing the median number of backtracks for 100 instances with that particular hole density.

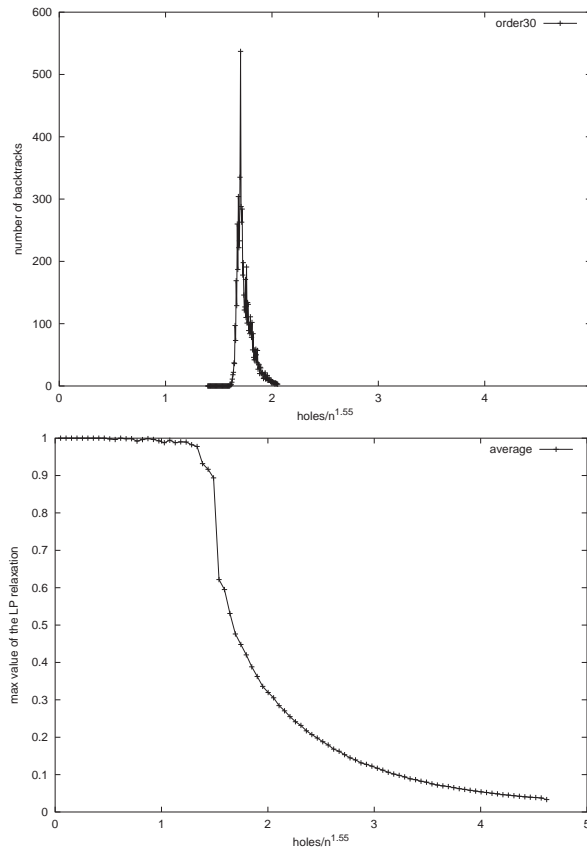


Fig. 1. Easy-hard-easy pattern in complexity for the Latin square with holes problem (top). Phase transition phenomenon in solution integrality for the assignment based LP relaxation (bottom).

the average maximum value of the LP relaxation (i.e., for each hole we consider the highest value returned by the relaxation). Each data point corresponds to the average over 100 instances for that particular hole density. In the under-constrained region (the left hand side of the plot) this average is close to 1, since most holes only have one coloring option. In the critically constrained region (around 1.5 in hole density), we observe a sudden drop in the integrality of the relaxation, reaching 0.5 for instances corresponding to the peak in complexity. As the hole density increases, the values become very fractional, capturing the fact that there are several ways to complete to partial Latin square (i.e., several coloring options for each hole).

To address our research goal, i.e., to characterize the performance of LP heuristics across different constrainedness regions, we performed the following experiment:

- formulate the problem as an LP,
- relax the integrality constraint of the variables,
- solve the relaxation (we used ILOG Solver libraries [10])

- set the x highest values suggested by the LP relaxation (we varied x between 1 and 5% of the variables, eliminating obvious conflicts),
- check if the resulting instance is still completable (using a complete solver).

In the first round of experiments, we solved the linear relaxation only once and used the results to set the x highest values suggested by the relaxation (after coloring a hole we perform forward check, i.e., we remove that color from the options available for the current row and column):

Algorithm 1 Set x highest values suggested by LP relaxation

Input: an order n PLS.

Solve for the assignment LP solution S

Repeat x times:

 Find $\max S_{ijk}$ such that (i, j) is an uncolored cell.

 Set $color_{ij} \leftarrow k$.

 Invalidate color k for row i and column j :

$$S_{ipk} \leftarrow 0, \forall p \neq j$$

$$S_{qjk} \leftarrow 0, \forall q \neq i$$

Output: resulting PLS.

Then, a complete solver checks if there exists a coloring for the resulting partial Latin square (PLS). Since the problem we are considering is the Latin square with holes, we know *a priori* that the instance we started out with is completable. So checking LP's guidance abilities is equivalent with checking whether the PLS output by the above algorithm is still completable.

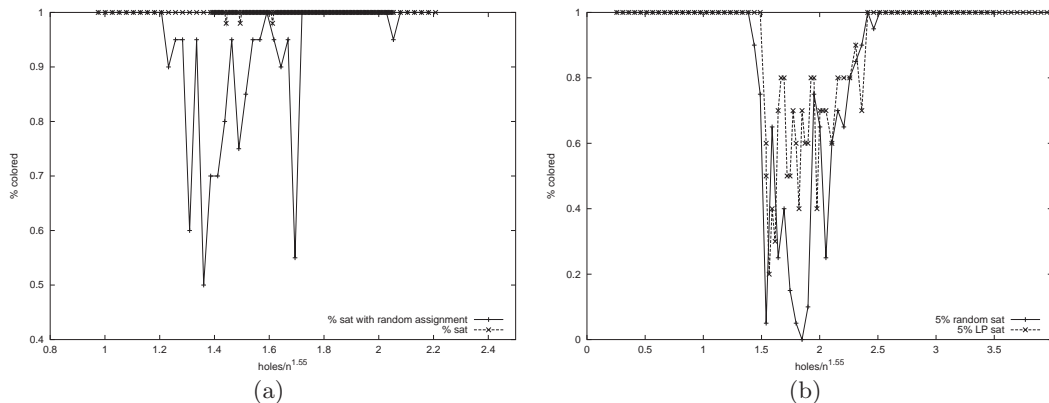


Fig. 2. (a) Percentage of satisfiable instances after setting 1 hole; and (b) 5% of holes for LP based vs. random heuristics .

Figure 2 displays the percentage of satisfiable instances after setting one hole (left) and after setting 5% of the holes (right), based on the highest value of the LP relaxation (assignment formulation) against the purely random strategy (we randomly select a hole and randomly assign a color). As expected, the percentage of satisfiable instances when using the LP guidance is clearly higher than when using a random strategy, across all constrainedness regions.

It is worth noting that, even in the case when we only set one hole, in the critically constrained region, the choice suggested by the LP relaxation sometimes

leads to an uncompletable PLS (i.e., it is a mistake). As the maximum values of the LP relaxation in this region are around 0.5 it is very easy to make a branching mistake (i.e., to focus on a region of the search space with no solutions).

Interestingly, most of the instances in the over-⁴ and under-constrained area remained completable after the setting dictated by the LP relaxation. This suggests that despite the fact that the LP relaxation values are quite fractional in the under-constrained area, the LP still provides global information that captures the multitude of solutions in the under-constrained area. In contrast, in the critically constrained area, the percentage of completable instances drops dramatically, as we set more and more variables based on the LP relaxation. For the 5% case (Figure 2 right), the performance gain against the purely random strategy is not that obvious (understandably so, considering that we set 5% of the holes at once — that is 45 holes for an order 30 PLS).

However, LP’s guidance was almost flawless in the case where we were only setting one hole. Could we improve the percentage of instances that remain completable by setting just one hole at a time? To answer this question we solved an LP relaxation at each step, and chose the highest value suggested by this relaxation:

Algorithm 2 Solve LP at each step

Input: an order n PLS.

Repeat x times:

 Solve for the assignment LP solution S

 Find $\max S_{ijk}$ such that (i, j) is an uncolored cell.

 Set $color_{ij} \leftarrow k$.

 Invalidate color k for row i and column j :

$$S_{ipk} \leftarrow 0, \forall p \neq j$$

$$S_{qjk} \leftarrow 0, \forall q \neq i$$

Output: resulting PLS.

The results are indeed very similar to the ones from Figure 2 (a), as the instances remain completable almost across the board (Figure 3 (b)). The heuristic consistently drives the search towards the (possibly unique) solution in the over-constrained region and towards one of the (possibly many) solutions in the under-constrained region. Once again, we remark the critically constrained region, where a dramatic drop in the percentage of instances that remain satisfiable occurs, perfectly aligned with the peak in complexity (corresponding to a phase transition phenomenon with respect to the integrality of the LP relaxation [11]).

In summary, our work characterizes the behavior of LP-based heuristics with respect to the constrainedness of the problem. We found that outside the critically constrained problem regions the LP relaxation provides good guidance. However, on critically constrained problems, even when just one cell is colored, we see that the relaxation starts making some mistakes. When setting 5% of the cells based on the LP relaxation, the error rate becomes substantial in the critical area. When the LP relaxation is solved every time a hole is colored, the performance improves significantly, suggesting that the instances in this region are indeed hard, but carefully designed “one step at a time” LP-based heuristics

⁴ A more correct term would be *highly constrained* since we know a solution always exists.

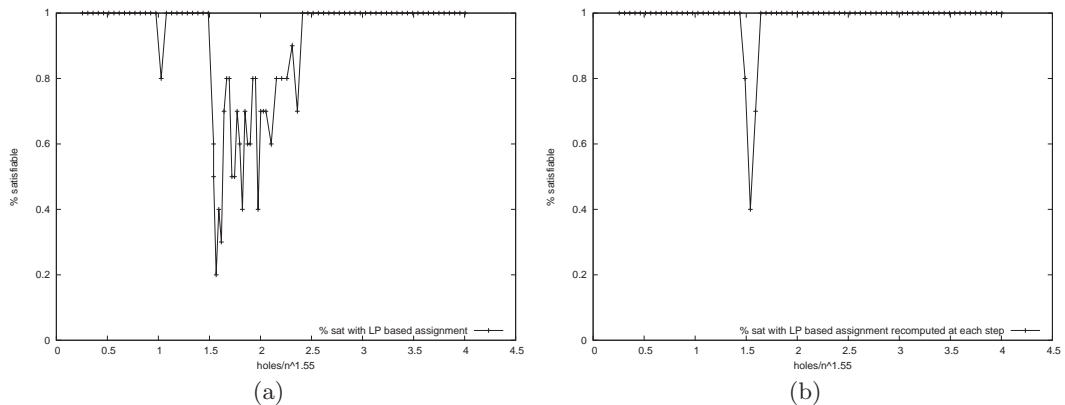


Fig. 3. Percentage of satisfiable instances after setting 5% of holes by resolving the LP relaxation once and setting 45 holes as suggested by the relaxation (a); and solving the LP relaxation at each step and set one hole (b).

could provide suitable guidance. The occasional flaws in the critically constrained area could be overcome by combining LP heuristics with a randomized restart strategy to recover from potential incorrect settings at the top of the search tree.

Acknowledgments We would like to thank the anonymous reviewers for their comments and suggestions.

References

1. Gomes, C., Selman, B.: Problem Structure in the Presence of Perturbations. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), New Providence, RI, AAAI Press (1997) 221–227
2. Hooker, J.: Resolution vs. cutting plane solution of inference problems: Some computational experience. *Operations Research Letter* **7** (1988) 1–7
3. Hooker, J.: Resolution and the integrality of satisfiability problems. *Mathematical Programming* **74** (1996) 1–10
4. Kamath, A., Karmarkar, N.K., Ramakrishnan, K.G., Resende, M.G.C.: A continuous approach to inductive inference. *Mathematical Programming* **57** (1992) 215–238
5. Warners, J.: Nonlinear approaches to satisfiability problems. PhD thesis, Technische Universiteit Eindhoven (1999)
6. Colbourn, C.: The complexity of completing partial latin squares. *Discrete Applied Mathematics* (1984) 25–30
7. Laywine, C., Mullen, G.: *Discrete Mathematics using Latin Squares*. Wiley-Interscience Series in Discrete mathematics and Optimization (1998)
8. Kumar, S.R., Russell, A., Sundaram, R.: Approximating latin square extensions. *Algorithmica* **24** (1999) 128–138
9. Achlioptas, D., Gomes, C., Kautz, H., Selman, B.: Generating Satisfiable Instances. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-00), New Providence, RI, AAAI Press (2000)
10. ILOG: Cplex. (<http://www.ilog.com/products/cplex>)
11. Leahu, L., Gomes, C.: Quality of LP-based approximations for highly combinatorial problems. In: International Conference on Principles and Practice of Constraint Programming (CP). (2004)