Quality of LP-based Approximations for Highly Combinatorial Problems*

Lucian Leahu and Carla P. Gomes

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

Abstract. We study the quality of LP-based approximation methods for pure combinatorial problems. We found that the quality of the LP-relaxation is a direct function of the underlying constrainedness of the combinatorial problem. More specifically, we identify a novel phase transition phenomenon in the solution integrality of the relaxation. The solution quality of approximation schemes degrades substantially near phase transition boundaries. Our findings are consistent over a range of LP-based approximation schemes. We also provide results on the extent to which LP relaxations can provide a global perspective of the search space and therefore be used as a heuristic to guide a complete solver.

Keywords: phase transition, approximations, search heuristics, hybrid LP/CSP

1 Introduction

In recent years we have witnessed an increasing dialogue between the Constraint Programming (CP) and Operations Research (OR) communities in the area of combinatorial optimization. In particular, we see 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 a significant amount of beautiful LP results for Boolean satisfiability (SAT) problems (see e.g., [1–4]), practical state-of-the-art solvers do not yet incorporate LP relaxation techniques.

In our work we are interested in studying highly combinatorial problems, i.e., problems with integer variables and mainly symbolic constraints, such as sports scheduling, rostering, and timetabling. CP based strategies have been shown to outperform traditional LP/IP based approaches on these problems.

* Research supported by the Intelligent Information Systems Institute, Cornell University (AFOSR grant F49620-01-1-0076 and MURI grant F49620-01-1-0361).

As a prototype of a highly combinatorial problem we consider the Latin square (or quasigroup) completion problem [5]).[1] 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 is a valid completion into a full Latin square is an NP-complete problem [7]. 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 [8, 9].

In this paper, we study the quality of LP based approximations for the problem of completing Latin squares. We start by considering the LP assignment formulation [9], described in detail in section 2. In this formulation, we have n^3 variables, some of them with pre-assigned values. Each variable, x_{ijk} $(i, j, k = 1, 2 \ldots, n)$, 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. 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 problem, one starts with a complete Latin square and randomly deletes some of the values assigned to its n^2 cells, which we refer to as "holes". This problem is guaranteed to have a completion, and therefore we know *a priori* that its optimal value is n^2. This problem is NP-hard and it exhibits an easy-hard-easy pattern in complexity, measured in the runtime (backtracks) to find a completion [10].

In our study we observed an interesting phase transition phenomenon in the *solution integrality* of the LP relaxation. To the best of our knowledge, this is the first time that such a phenomenon is observed. Note that phase transition phenomena have been reported for several combinatorial problems. However, such results generally refer to phase transitions with respect to the solvability of the instances, not with respect to the solution integrality for LP relaxations or more generally with respect to the quality of approximations.

The top plot in figure 1 depicts the easy-hard-easy pattern in computational complexity, measured in number of backtracks, for the problem of Latin squares with holes.[2] The x axis in this plot corresponds to the density of holes in the Latin square.[3] The left-hand side of the plot corresponds to the over-constrained area — i.e., a region in which instances only have a few holes and therefore lots of

[1] The multiplication table of a quasigroup is a Latin square. The designation of Quasigroup Completion Problem was inspired by the work done by the theorem proving community on the study of quasigroups as highly structured combinatorial problems. For example, the question of the existence and non-existence of certain quasigroups with intricate mathematical properties gives rise to some of the most challenging search problems [6].

[2] Each data point in this plot was generated by computing the median solution runtime for 100 instances.

[3] 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. It turns out that for scaling reasons, the denominator is $n^{1.55}$ [10].

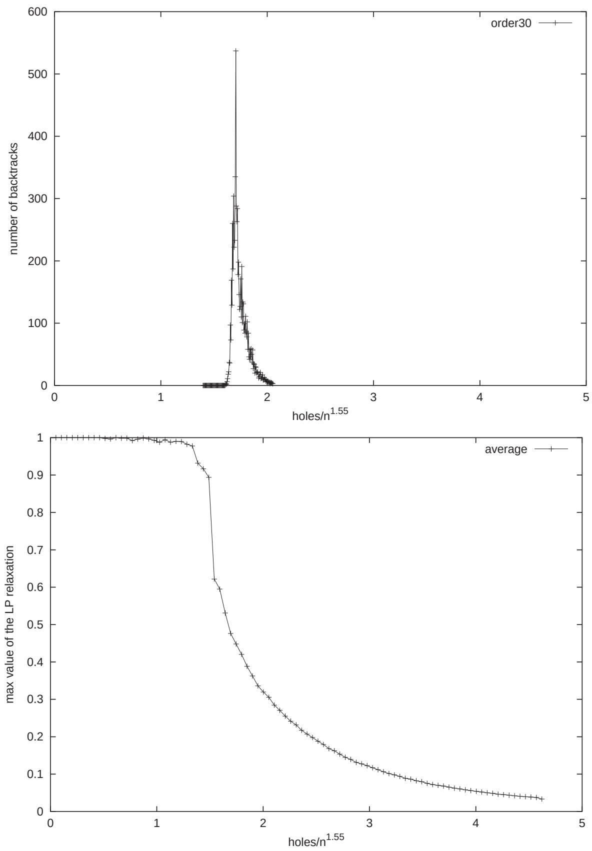


Fig. 1. Easy-hard-easy pattern in complexity for the Latin square with holes problem (top). Phase transition phenomenon in solution integrality for the assignment based LP relaxation (bottom).

pre-assigned values. This is an "easy" region since it is easy for a solver to find a completion, given that only a few holes need to be colored. The right-hand side of the plot corresponds to the under-constrained area — i.e., a region in which instances have lots of holes and therefore only a few pre-assigned colors. This is also an easy region since there are lots of solutions and it is easy to find a solution. The area between the over-constrained and the under-constrained areas is the critically constrained area, where the cost in complexity peaks. In this region, instances have a critical density in holes that makes it difficult for a solver to find a completion: a wrong branching decision at the top of the search tree may steer the search into a very large inconsistent sub-tree. The bottom plot of figure 1 shows the phase transition phenomenon in the solution integrality for the LP relaxation of the assignment formulation of the Latin squares with holes problem. Each data point is the average (over 100 instances) of the maximum variable value of the LP relaxation. We observe a drastic change in solution integrality as we enter the critically constrained region (around 1.5 in hole density): in the critically constrained area the average LP relaxation variable solution values become fractional (less than 1), reaching 0.5 in the neighborhood of the peak of the computational complexity. After this point, the average LP relaxation variable solution values continue to become more fractional, but at a slower rate. The intuition is that, in the under-constrained area, there are lots of solutions, several colors can be assigned to the same cell, and therefore the LP relaxation becomes more fractional.

Two interesting research issues are closely related to the quality of the LP relaxation:

- What is the quality of LP based approximations?
- Does the LP relaxation provide a global perspective of the search space? Is it a valuable heuristic to guide a complete solver for finding solutions to hard combinatorial problems?

In order to address the first question, we study the quality of several LP based approximations. In recent years there has been considerably research in the area of approximation algorithms. Approximation algorithms are procedures that provide a feasible solution in polynomial time. Note that in most cases it is not difficult to devise a procedure that finds some solution. However, we are interested in having some guarantee on the quality of the solution, a key aspect that characterizes approximation algorithms. The quality of an approximation algorithm is the "distance" between its solutions and the optimal solutions, evaluated over all the possible instances of the problem. Informally, an algorithm approximately solves an optimization problem if it always returns a feasible solution whose measure is close to optimal, for example within a factor bounded by a constant or by a slowly growing function of the input size. More formally, given a maximization problem Π and a constant α ($0 < \alpha < 1$), an algorithm $\mathcal{A}$ is an α-approximation algorithm for Π if its solution is at least α times the optimum, considering all the possible instances of problem Π. We remark that approximation guarantees on the quality of solutions are worst-case notions. Quite often the analysis is somewhat "loose", and may not reflect the best possible ratio that can be derived.

We study the quality of LP based approximations from a novel perspective: we consider "typical" case quality, across different areas of constrainedness. We consider different LP based approximations for the problem of Latin squares with holes, including an approximation that uses a "stronger" LP relaxation, so-called packing formulation. Our analysis shows that the quality of the approximations is quite sensitive to the particular approximation scheme considered. Nevertheless, for the approximation schemes that we considered, we observe that as we enter the critically constrained area the quality of the approximations drops dramatically. Moreover, in the under-constrained area, approximation schemes that use the LP relaxation information in a more greedy way (basically setting the highest values suggested by the LP) performed considerably better than non greedy approximations.

To address the second research question, i.e., to what extent the LP relaxation provides a global perspective of the search space and therefore to what extent it can be used as a heuristic to guide a complete solver, we performed the following experiment: 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. Interestingly, most of the instances in the over-constrained *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.

In summary, our results indicate that LP based approximations go through a drastic phase change in quality as we go from the over-constrained area to the critically constrained area, closely correlated with the inherent hardness of the instances. Overall, LP based approximations provide a global perspective of the search space, though we observe a clear drop in quality in the critically constrained region.

The structure of rest of the paper is as follows: in the next section we describe two different LP formulations for the Latin square problem. In section 3 we provide detailed results on the quality of different LP-based approximations across the different constrainedness regions and in section 4 we study the value of the LP relaxation as a backtrack search heuristic. Finally in section 5 we provide conclusions and future research directions.

2 LP-based Problem Formulations

2.1 Assignment Formulation

Given a partial Latin square of order n, PLS, with partially assigned values to some of its cells denoted by $PLS_{ij} = k$, the Latin square completion problem can be expressed as an integer program [9]:

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

subject to

$$\sum_{i=1}^{n} x_{ijk} \leq 1, \quad \forall j, k$$

$$\sum_{j=1}^{n} x_{ijk} \leq 1, \quad \forall i, k$$

$$\sum_{k=1}^{n} x_{ijk} \leq 1, \quad \forall i, j$$

$$x_{ijk} - \text{ cell } (i,j) \text{ takes symbol } k \quad \forall i, j, k$$

$$x_{ijk} = 1 \quad \forall i, j, k \text{ such that } PLS_{ij} = k$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j, k$$

$$i, j, k = 1, \ldots, n$$

If the PLS is completable, the optimal value of this integer program is n^2, i.e., all cells in the PLS can be legally colored.

2.2 Packing Formulation

An alternate formulation for the Latin square problems is the *packing formulation* [11, 12]. The assignment formulation described in the previous section uses variables x_{ijk} for each cell (i, j) and each color k. Instead, note that the cells having the same color in a PLS form a (possibly partial) matching of the rows and columns of the PLS. Informally, a matching corresponds to a full or partial valid assignment of a given color to the rows (or columns) of the Latin square matrix. For each color k, let $\mathcal{M}_k$ be the set of all matchings of rows and columns that extend the matching corresponding to color k in a PLS. For each color k and for each matching $M \in \mathcal{M}_k$, we introduce a binary variable y_{kM}. Using this notation, we can generate the following IP formulation:

$$\max \sum_{k=1}^{n} \sum_{M \in \mathcal{M}_k} |M| y_{kM}$$

subject to

$$\sum_{M \in \mathcal{M}_k} y_{kM} = 1, \quad \forall k$$

$$\sum_{k=1}^{n} \sum_{M \in \mathcal{M}_k : (i,j) \in M} y_{kM} \leq 1, \quad \forall i, j$$

$$y_{kM} \in \{0, 1\} \quad \forall k, M.$$

Once again, we consider the linear programming relaxation of this formulation by relaxing the integrality constraint, i.e., the binary variables take values in the interval $[0, 1]$. Note that, for any feasible solution y to this linear programming relaxation, one can generate a corresponding feasible solution x to the assignment formulation, by simply computing $x_{ijk} = \sum_{M \in \mathcal{M}_k : (i,j) \in M} y_{kM}$. This construction implies that the value of the linear programming relaxation of the assignment formulation (which provides an upper bound on the desired integer programming formulation) is at least the bound implied by the LP relaxation of the packing formulation; that is, the packing formulation provides a tighter upper bound. Interestingly, from the solution obtained for the assignment formulation one can generate a corresponding solution to the packing formulation, using an algorithm that runs in polynomial time. This results from the fact that the extreme points of each polytope

$$P_k = \{x \ : \ \sum_{i=1}^{n} x_{ijk} \leq 1 (j = 1, \ldots, n), \ \sum_{j=1}^{n} x_{ijk} \leq 1 (i = 1, \ldots, n), \ x \geq 0\},$$

for each $k = 1, \ldots, n$ are integer, which is a direct consequence of the Birkhoff-von Neumann Theorem [13]. Furthermore, these extreme points correspond to matchings, i.e., a collection of cells that can receive the same color. Therefore, given the optimal solution to the assignment relaxation, we can write it as a convex combination of extreme points, i.e., matchings, and hence obtain a feasible solution to the packing formulation of the same objective function value. Hence, the optimal value of the packing relaxation is at most the value of the assignment relaxation. It is possible to compute the convex combination of the matchings efficiently. Hence, the most natural view of the algorithm is to solve the assignment relaxation, compute the decomposition into matchings, and then perform randomized rounding to compute the partial completion.

In the next section we study the quality of different randomized LP-based approximations for the Latin square problem based on the assignment and packing formulations.

3 Quality of LP-based Approximations

We consider LP-based approximation algorithms for which we solve the linear programming relaxation of the corresponding formulation (assignment formulation or packing formulation), and (appropriately) interpret the resulting fractional solution as providing a probability distribution over which to set the variables to 1 (see *e.g.*, [14]).

Consider the generic integer program $\max cz$ subject to $Az = b$, $z \in \{0,1\}^N$, and solve its linear relaxation to obtain z^*. If each variable z_j is then set to 1 with probability ${z_j}^*$, then the expected value of the resulting integer solution is equal to the LP optimal value, and, for each constraint, the expected value of the left-hand side is equal to the right-hand side. Of course, we have no guarantee that the resulting solution is feasible, but it provides a powerful intuition for why such a *randomized rounding* is a useful algorithmic tool (see *e.g.*, [14]). This approach has led to striking results in a number of settings (e.g., [15–17]).

3.1 Uniformly at Random

Based on the Assignment Formulation. — This approximation scheme selects an uncolored cell (i, j) uniformly at random, assigning a color k with probability equal to the value of the LP relaxation for the corresponding variable x_{ijk}. Before proceeding to the next uncolored cell, we perform forward checking, by invalidating the color just set for the current row and column.

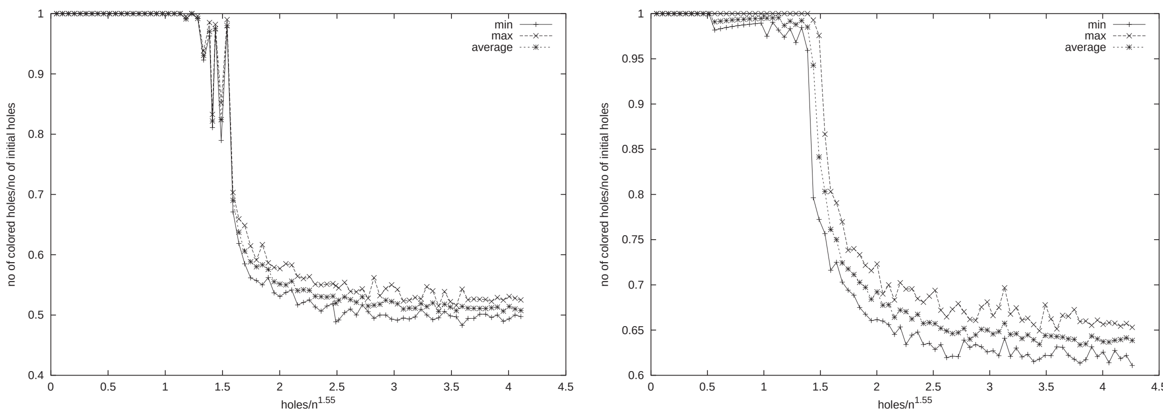
Algorithm 1 Random LP Assignment

Input: an assignment LP solution x for an order n PLS.
Repeat until all uncolored cells have been considered:
    Randomly choose an uncolored cell (i, j).
    Set $color_{ij} \leftarrow k$ with probability x_{ijk}.
    Invalidate color k for row i and column j:

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

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

Output: the number of colored cells.



(a) (b)

Fig. 2. (a) Random LP Assignment Approximation and (b) Random LP Packing Approximation Quality.

Based on the Packing Formulation. — As we mentioned above we can generate a solution for the packing formulation from the assignment formulation in polynomial time. Once we have the packing LP relaxation y, we can proceed to color the cells. In the following we present a randomized rounding scheme. This scheme interprets the solution for a variable y_{kM} as the probability that the matching $M \in \mathcal{M}_k$ is chosen for color k. The scheme selects randomly such a matching for each color k, according to these probabilities. Note that this algorithm can output matchings that overlap. In such cases, we select an arbitrary color from the colors involved in the overlap.

Algorithm 2 Random LP Packing
Input: a packing LP solution $\mathcal{M}$ for an order n PLS.
Repeat for each color k:
  Interpret the values of y_{kM}, $M \in \mathcal{M}_k$, as probabilities.
  Select exactly one matching according to these probabilities.
Output: the number of colored cells.

Figure 2 plots the quality of the approximation using the algorithm Random LP Assignment (left) and the algorithm Random LP Packing (right), as a function of the hole density (the quality of the approximation is measured as number of colored holes/number of initial holes). Both plots display a similar qualitative behavior: we see a clear drop in the quality of the approximations as we enter the critically constrained area. The rate at which the quality of the approximation decreases slows down in the under-constrained area. This phenomenon is similar to what we observed for the solution integrality of the LP relaxation. However, the quality of the approximation given by the algorithm Random LP Packing is considerably better, especially in the under-constrained area (note y-axis scales in figure 2). This was expected given that the LP relaxation for the packing formulation is stronger than the relaxation given by the assignment formulation (see figure 3). In fact Random LP Packing is guaranteed to be at most $(1 - \frac{1}{e}) \approx 0.63$ from the optimal solution [11]. For approximations based on the assignment formulation the known formal guarantee is a factor 0.5 from optimal [9].

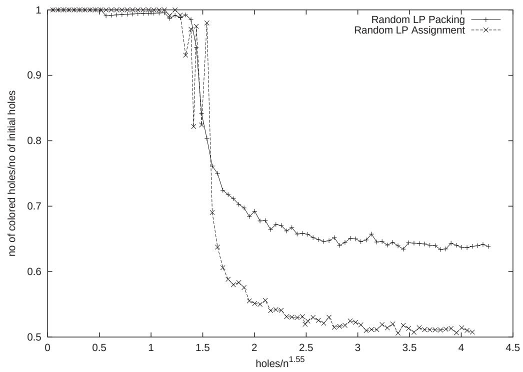


Fig. 3. Random LP Packing vs. Random LP Assignment — Average Case.

3.2 Greedy Random Approximations

Based on the Assignment Formulation. — The following rounding scheme takes as input an assignment LP relaxation. It considers all uncolored cells uniformly at random, and assigns to each such cell the color that has the highest value of the LP relaxation. After each assignment, a forward check is performed, by invalidating the color just set for the current row and column.

Algorithm 3 Greedy Random LP Assignment
Input: an assignment LP solution x for an order n PLS.
Repeat until all uncolored cells have been considered:
  Randomly choose an uncolored cell (i, j).
  Find $k < n$ that $\max_k x_{ijk}$.
  If $x_{ijk} > 0$, set $color_{ij} \leftarrow k$, invalidate color k for row i and column j:

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

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

Output the number of colored cells.

Based on the Packing Formulation. — For the LP packing formulation, we also consider a cell based approach. All uncolored cells are considered uniformly at random. For one such cell (i, j), we find a color k corresponding to $M \in \mathcal{M}_k$, $\forall k = 1, \ldots, n$, such that y_{kM} is the highest value of the LP relaxation for all matchings M that match row i to column j. We perform forward checking by invalidating color k for row i and column j (i.e., removing (i, j) from all the matchings $M \in \mathcal{M}_k$).

Algorithm 4 Greedy Random LP Packing

Input: a packing LP solution y for an order n PLS.

Repeat until all uncolored cells have been considered:

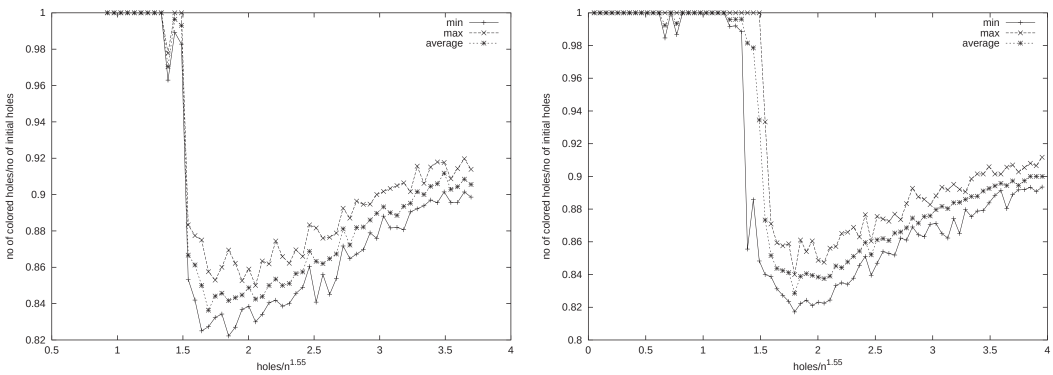
Randomly choose an uncolored cell (i, j).

Find a matching $M \in \mathcal{M}_k \forall k = 1, \ldots, n$, such that i is matched to j in M, with the highest value of the LP relaxation.

If such a matching exists, set $color_{ij} \leftarrow k$ and invalidate color k for row i and column j:

$$\forall M' \in \mathcal{M}_k, \forall p \neq j, \text{ remove } (i,p) \text{ from } M'$$
$$\forall M' \in \mathcal{M}_k, \forall q \neq i, \text{ remove } (q,j) \text{ from } M'$$

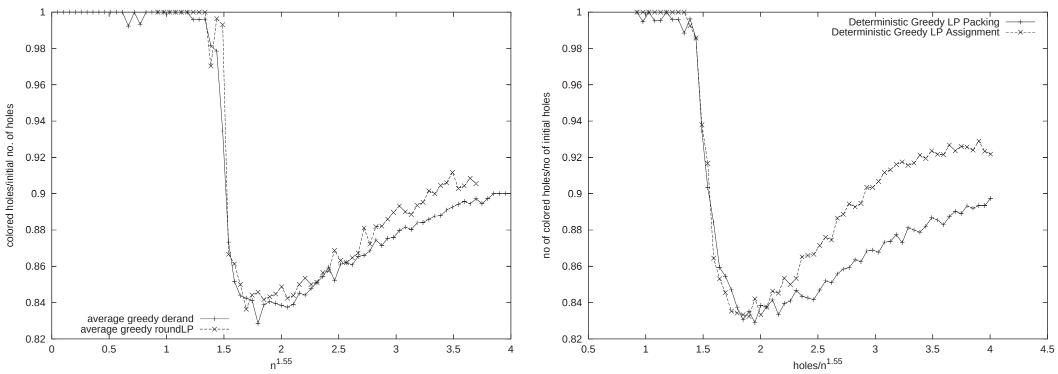
Output: the number of colored cells.



(a) (b)

Fig. 4. (a) Greedy Random LP Assignment and (b) Greedy Random LP Packing.

Figure 4 plots the quality of the approximation using the algorithm Greedy Random LP Assignment (left) and the algorithm Greedy Random LP Packing (right). Again, both plots display a similar qualitative behavior: we see a clear drop in the quality of the approximations as we enter the critically constrained area. In addition, and contrarily to the results observed with the random approximations discussed earlier, both plots show that the quality of the approximation increases in the under-constrained area. Recall that these approximations are greedy, picking the next cell to color randomly and then just setting it to the highest value suggested by the LP. This seems to suggest that the information provided by the LP is indeed valuable, which is further enhanced by the fact that forward checking is performed after each color assignment to remove inconsistent colors from unassigned cells. Interestingly, in the under-constrained area, the quality of the Random LP Packing approximation is slightly worse than the the Random LP Assignment approximation. (See figure 5(a).) The intuition is that, because this approximation "optimizes" the entire matchings per color, it is not as greedy as the approximation based on the assignment formulation and therefore it does not take as much advantage of the look-ahead as the Greedy Random LP Assignment does.



(a) (b)

Fig. 5. (a) Greedy Random LP Packing vs. Greedy Random LP Assignment. (b) Deterministic Greedy LP Assignment vs. Deterministic Greedy LP Packing — Average Case.

3.3 Greedy Deterministic Approximations

We now consider deterministic approximations that are even greedier than the previous ones: they pick the next cell/color to be set by finding the cell/color with the highest LP value.

Based on the Assignment Formulation. — Greedy Deterministic LP Assignment considers the uncolored cell values of the LP relaxation in decreasing order. After each assignment, forward check ensures the validity of the future assignments, so that the end result is a valid extension of the original PLS.

Algorithm 5 Greedy Deterministic LP Assignment
Input: an assignment LP solution x for an order n PLS.
Repeat until all uncolored cells have been considered:
Find $\max x_{ijk}$ such that (i, j) is an uncolored cell.
Set $color_{ij} \leftarrow k$.
Invalidate color k for row i and column j:

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

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

Output: the number of colored cells.

Based on the Packing Formulation. — Now we turn out attention to a deterministic rounding scheme for the packing LP formulation. We describe a greedy rounding scheme. We consider the matchings $M \in \mathcal{M}_k$, $\forall k = 1, \ldots, n$, in decreasing order of the corresponding y_{kM} values. At each step we set the color for the uncolored cells corresponding to the current matching. For each such cell (i, j), we perform forward checking by invalidating the color k for row i and column j.

Algorithm 6 Greedy Deterministic LP Packing
Input: a packing LP solution y for an order n PLS.
Repeat until no more options (i.e., max = 0) or all cells colored:
Find the matching $M \in \mathcal{M}_k, \forall k = 1, \ldots, n$, that has the highest value of the LP relaxation.
If such a matching exists, set $color_{i,j} \leftarrow k$, $\forall(i,j)$ such that cell (i,j) is not colored and i is matched to j in M. Invalidate color k for row i and column j:

$$\forall M' \in \mathcal{M}_k, \forall p \neq j, \text{ remove } (i,p) \text{ from } M'$$

$$\forall M' \in \mathcal{M}_k, \forall q \neq i, \text{ remove } (q,j) \text{ from } M'$$

Output: the number of colored cells.

Figure 5(b) compares the quality of the approximation Greedy Deterministic LP Assignment against the approximation Greedy Deterministic LP Packing. What we observed before for the case of the greedy random approximations is even more clear for greedy deterministic approximations: in the underconstrained region, Greedy Deterministic LP Assignment clearly outperforms Greedy Deterministic LP Packing. The intuition is that a similar argument as the one mentioned for the random greedy approximations explains this phenomenon. Greedy Deterministic LP Packing sets the color for more cells at the same time (i.e., all the uncolored cells in the considered matching), as opposed to Greedy Deterministic LP Assignment and even Greedy Random LP Packing, which consider just one uncolored cell at each step. Thus, both the Greedy Deterministic LP Assignment and the Greedy Random LP Packing perform forward checking after setting each cell. This is not the case for Greedy Deterministic LP Packing: this approximation performs forward checking only after setting a matching. In figure 6, we compare the performance of the approximations that perform the best in each of the cases considered against a purely blind random strategy. We see that the greedy approximations based on the LP assignment formulation perform better. Overall, all the approximations we have tried outperform the purely blind random strategy. We remark that, the quality of the purely random strategy improves as the problem becomes "really easy" (i.e., the right hand side end of the graph). In this small region, the pure random method slightly outperforms the Random LP approximations: as the problem becomes easier (i.e., many possible solutions), the LP solution becomes more fractional and thus is less likely to provide satisfactory guidance.

4 LP as a Global Search Heuristic

Related to the quality of the LP based approximations is the question of whether the LP relaxation provides a good global perspective of the search space and therefore can be used as a heuristic to guide a complete solver for finding solutions to hard combinatorial problems. To address this question we performed the following experiment: set the x highest values suggested by the LP relaxation (we varied x between 1 and 5% of the variables, eliminating obvious conflicts); run a complete solver on the resulting instance and check if it is still completable. In order to evaluate the success of the experiment, we also set x values uniformly at

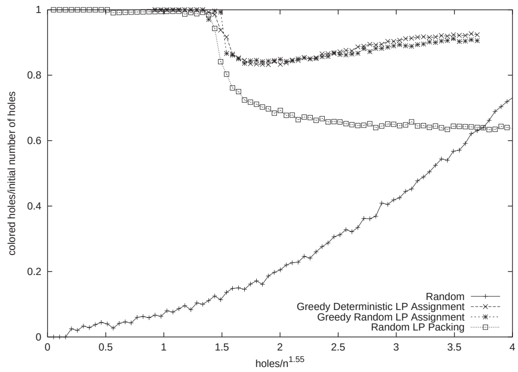
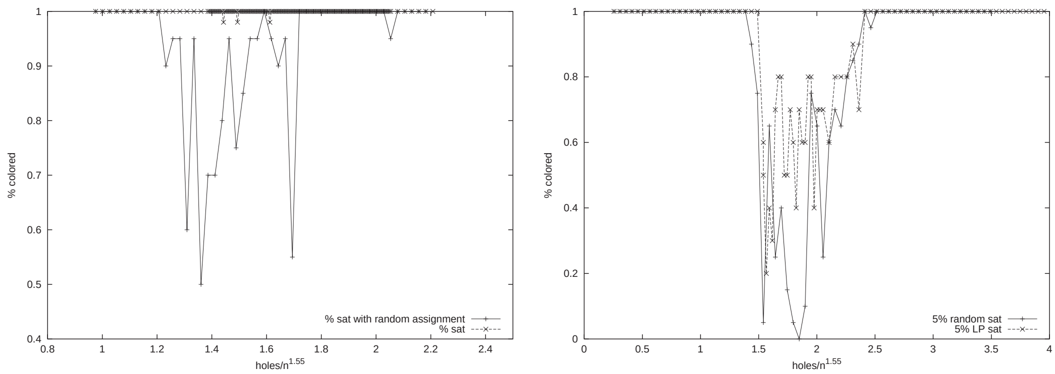


Fig. 6. Random LP Packing vs. Greedy Random LP Assignment vs. Greedy Deterministic LP Assignment vs. Pure Random Approximation — Average Case.



(a) (b)

Fig. 7. (a) Percentage of satisfiable instances after setting 1 hole; and (b) 5% of holes for LP based vs. random heuristics .

random (avoiding obvious row/column conflicts) and then check if the resulting instance is completable.

Figure 7 displays the percentage of satisfiable instances after setting 1 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. As expected, the percentage of satisfiable instances when using the LP guidance is clearly higher than when using a random strategy.

Interestingly, the information provided by the LP seems quite robust, both in the over-constrained and under-constrained area, with nearly 100% of the instances satisfiable after the setting. On the other hand, in the critically constrained area, the information provided by the LP relaxation is less accurate, even in the case of setting just one hole; in the critically constrained area several instances become unsatisfiable.[4] As we set more and more values based on the LP

[4] Recall that we are using the Latin square with holes and therefore we know that each instance is completable (satisfiable).

relaxation, the percentage of unsatisfiable instances in the critically constrained area increases dramatically.

5 Conclusions

We have studied the quality of LP based approximations for purely combinatorial problems. Our first results show that the quality of the approximation is closely correlated to the constrainedness of the underlying constraint satisfaction problem. In fact, we see that solution quality sharply degrades in the critically constrained areas of the problem space. This abrupt change in solution quality directly correlates with a phase transition phenomenon observed in solution integrality of the LP relaxation. At the phase transition boundaries, the LP solutions become highly fractional. The phase transition in LP solution integrality coincides with the peak in search cost complexity.

We considered two different LP formulations for the Latin square problem; an assignment based and a packing based formulation. The packing formulation is provably stronger than the assignment based formulation. This is reflected in terms of the quality of random approximations, i.e., approximations that uniformly at random pick the next cell to be colored and assign it a randomly, weighted according to the LP relaxation values.

There are different ways, however, of interpreting the LP relaxation values. For example, in a more greedy approach, we assign colors to cells starting with the highest LP relaxation values. Such a greedy scheme is beyond formal analysis at this point. However, empirically we found that in this approach the assignment based formulation gives higher quality assignments (more colored cells) than the packing based formulation. So, interestingly, a tighter LP formulation does not necessarily lead to better approximations when constructing solutions incrementally.

Finally, we considered the quality of LP relaxation when used as a global search heuristic. In particular, we considered setting some initial cells based on the LP relaxation (using the highest values in the LP relaxation of the assignment formulation). We then checked whether the partial Latin square could still be completed. We found that outside the critically constraint 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. These results show that although LP relaxations can provide useful high-level search guidance, on critically constrained problems, it makes sense to combine LP guidance with a randomized restart strategy to recover from potential incorrect settings at the top of the search tree.

LP relaxations are traditionally used for search space pruning. In this setting, a tighter LP formulation provides more powerful pruning. However, our results indicate that when LP relaxations are used in approximation schemes or as a global search heuristic, the situation is more complex, with the tightest LP bounds not necessarily leading to the best approximations and/or search guidance.

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

References

1. Hooker, J.: Resolution vs. cutting plane solution of inference problems: Some computational experience. Operations Research Letter **7** (1988) 1–7
2. Hooker, J.: Resolution and the integrality of satisfiability problems. Mathematical Programming **74** (1996) 1–10
3. Kamath, A., Karmarkar, N.K., Ramakrishnan, K.G., Resende, M.G.C.: A continuous approach to inductive inference. Mathematical Programming **57** (1992) 215–238
4. Warners, J.: Nonlinear approaches to satisfiability problems. PhD thesis, Technische Universiteit Eindhoven (1999)
5. 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
6. Slaney, J., Fujita, M., Stickel, M.: Automated reasoning and exhaustive search: Quasigroup existence problems. Computers and Math. with Applications **29** (1995) 115–132
7. Colbourn, C.: The complexity of completing partial latin squares. Discrete Applied Mathematics (1984) 25–30
8. Laywine, C., Mullen, G.: Discrete Mathematics using Latin Squares. Wiley-Interscience Series in Discrete mathematics and Optimization (1998)
9. Kumar, S.R., Russell, A., Sundaram, R.: Approximating latin square extensions. Algorithmica **24** (1999) 128–138
10. 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)
11. Gomes, C., Regis, R., Shmoys, D.: An Improved Approximation Algorithm for the Partial Latin Square Extension Problem. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-03), Baltimore, MD, USA (2003) 832–833
12. Gomes, C., Shmoys, D.: Approximations and Randomization to Boost CSP Techniques. To appear in Annals of Operations Research (2004)
13. von Neumann, J.: A certain zero-sum two-person game equivalent to the optimal assignment problem. In: Contributions to the Theory of Games, vol. 2. Princeton University Press, Princeton (1953)
14. Motwani, R., Naor, J., Raghavan, P.: Randomized approximation algorithms in combinatorial optimization. In Hochbaum, D.S., ed.: Approximation Algorithms for NP-Hard Problems. PWS Publishing Company (1997)
15. Goemans, M.X., Willianson, D.P.: 0.878-approximation algorithms for max-cut and max-sat. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Theory of Computing. (1994) 422–431
16. Chudak, F., Shmoys, D.: Improved approximation algorithms for the uncapacitated facility location problem. In: Submitted for publication. (1999) Preliminary version of this paper (with the same title) appeared in proceedings of the Sixth Conference on Integer Programming and Combinatorial Optimization.
17. Gomes, C., Williams, R.: Approximation algorithms. In: Introduction to Optimization, DecisionSupport and Search Methodologies (to appear). (2004)