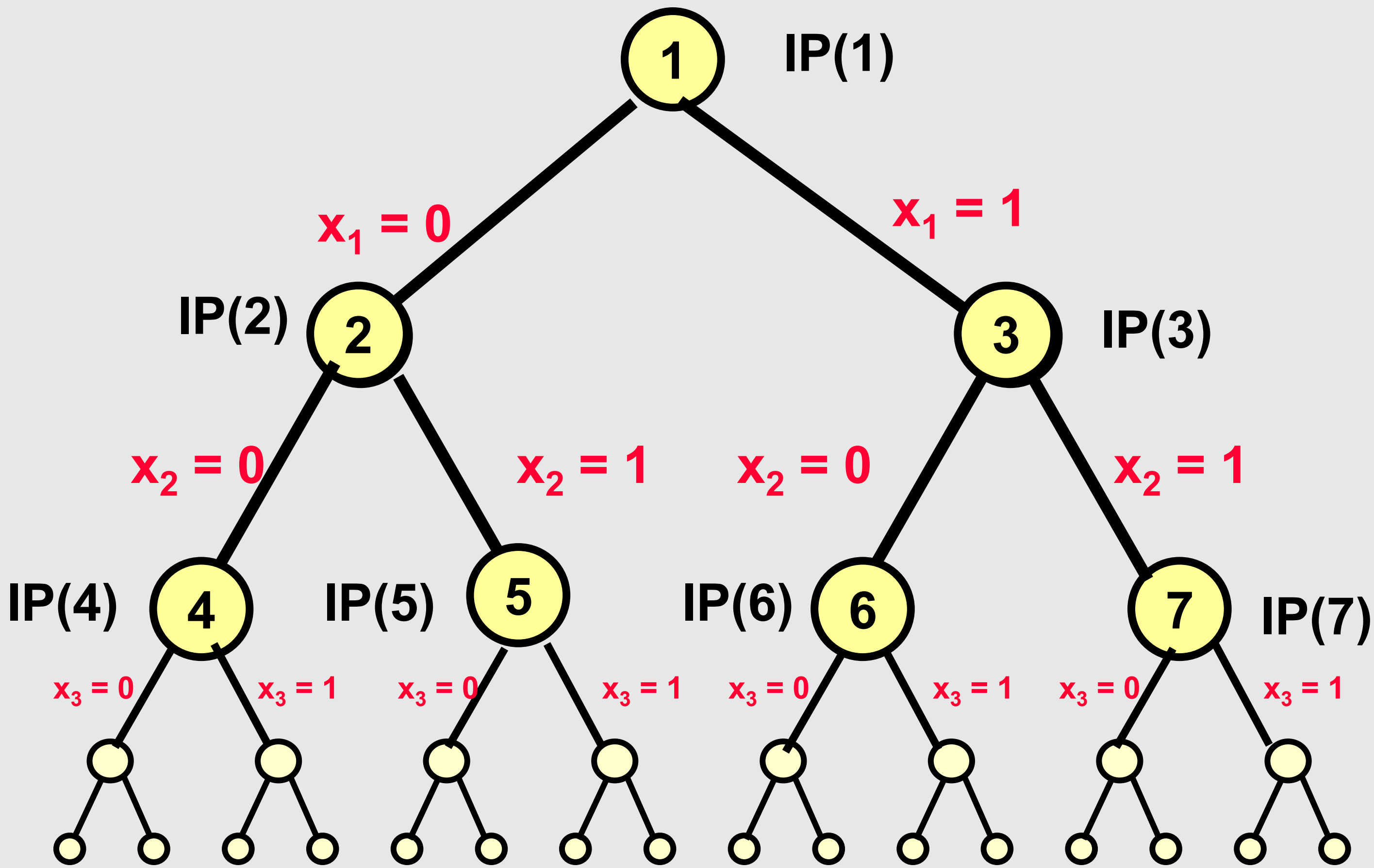
15.053/8 April 4, 2013

Cutting planes:

Better bounds through better IP formulations

An enumeration tree

Max	$24\,x_1 + 2\,x_2 + 20\,x_3 + 4\,x_4$	
s.t.	$8\,x_1 + 1\,x_2 + 5\,x_3 + 4\,x_4 \le 9$	**IP(1)**
	$x_i \in \{0,1\}$ for i = 1 to 4.	



$z_{IP}(j)$ = optimal value for IP(j).

$z_{LP}(j)$ = optimal value for LP(j).

$x(j)$ = optimal solution for LP(j)

Maximize $24x_1 + 2x_2 + 20x_3 + 4x_4$

subject to $8x_1 + 1x_2 + 5x_3 + 4x_4 \leq 9$

IP(1) $x_i \in \{0,1\}$ for $i = 1$ to 4.

Maximize $24x_1 + 2x_2 + 20x_3 + 4x_4$

subject to $8x_1 + 1x_2 + 5x_3 + 4x_4 \leq 9$

LP(1) $0 \leq x_i \leq 1$ for $i = 1$ to 4.

IMPORTANT OBSERVATIONS.

1. $z_{IP}(j) \leq z_{LP}(j)$ for all j.

2. If the costs are integral, then $z_{IP}(j) \leq \lfloor z_{LP}(j) \rfloor$.

The LP relaxation of a knapsack problem

LP(1)

Maximize $24\,x_1 + 2\,x_2 + 20\,x_3 + 4\,x_4$

subject to $8\,x_1 + 1\,x_2 + 5\,x_3 + 4\,x_4 \le 9$

$0 \le x_i \le 1$ for $i = 1$ to 4.

Variable	x_1	x_2	x_3	x_4
Value/weight	24/8 3	2/1 2	20/4 5	4/4 1

Put item 3 in the knapsack.

Weight remaining: $9 - 5 = 4$

Put 4/8 of item 1 in the knapsack.

Knapsack is filled.

Value = $20 + 24(4/8) = 32$.

Overview

- The best possible bounds: the convex hull.
- Packing diamonds
- Valid inequalities and cutting planes (a.k.a., cuts)
  - knapsack
  - general integer programs

Valid Inequalities

A valid inequality for an IP (or MILP) is any constraint that does not eliminate any feasible integer solutions.

maximize $z = 3x + 4y$

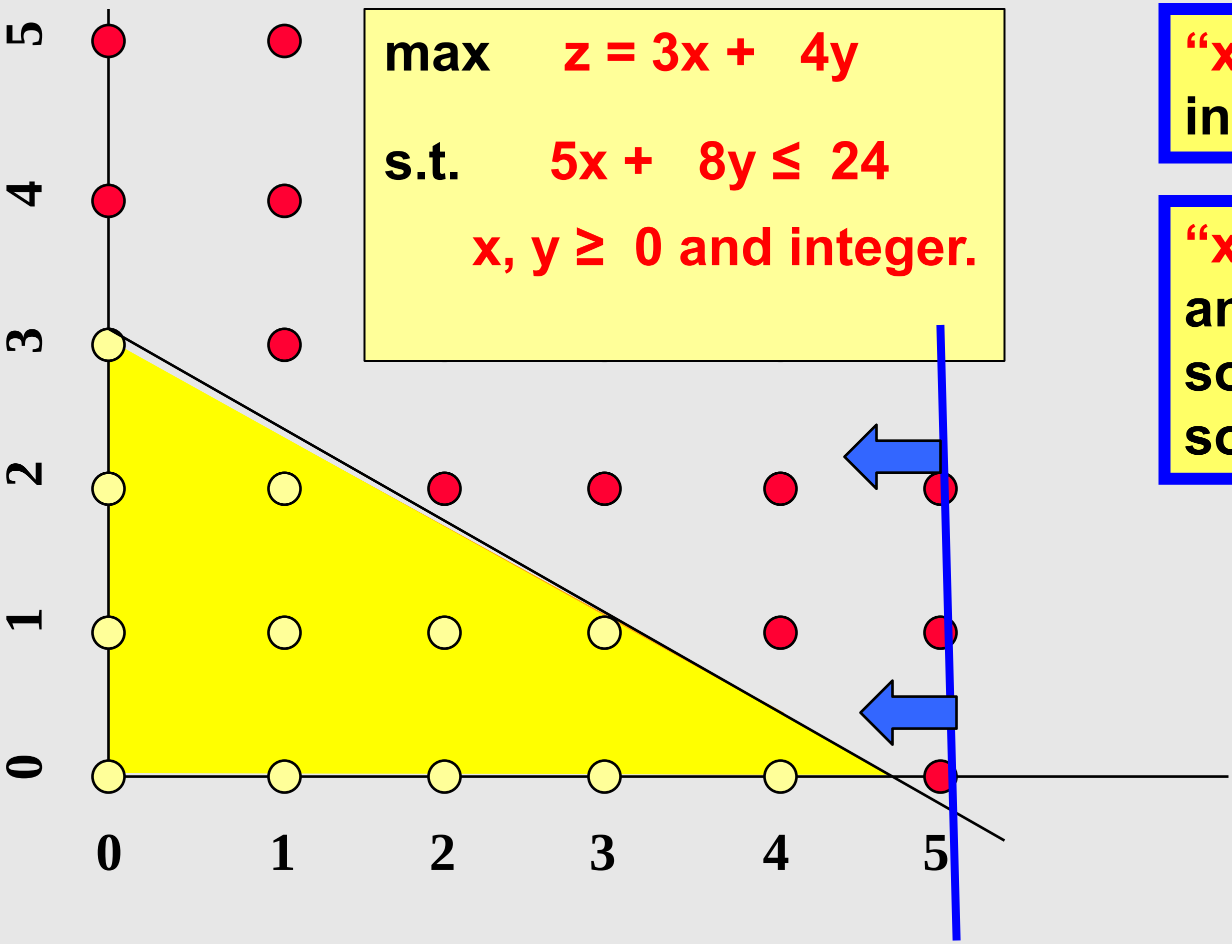
subject to $5x + 8y \leq 24$

$x, y \geq 0$ and integer.

The constraint $x \leq 5$ is a valid inequality

The constraint $x \leq 4$ is also a valid inequality

A ***valid inequality*** for an IP (or MILP) is any constraint that does not eliminate any feasible integer solutions. It is also called a ***cutting plane***, or ***cut***. We want cuts that eliminate part of the LP feasible region.



max $z = 3x + 4y$

s.t. $5x + 8y \le 24$

$x, y \ge 0$ and integer.

"$x \le 5$" is a valid inequality and cut.

"$x \le 4$" is also a cut, and it eliminates some fractional solutions.

The convex hull is the smallest LP feasible region that contains all of the integer solutions.

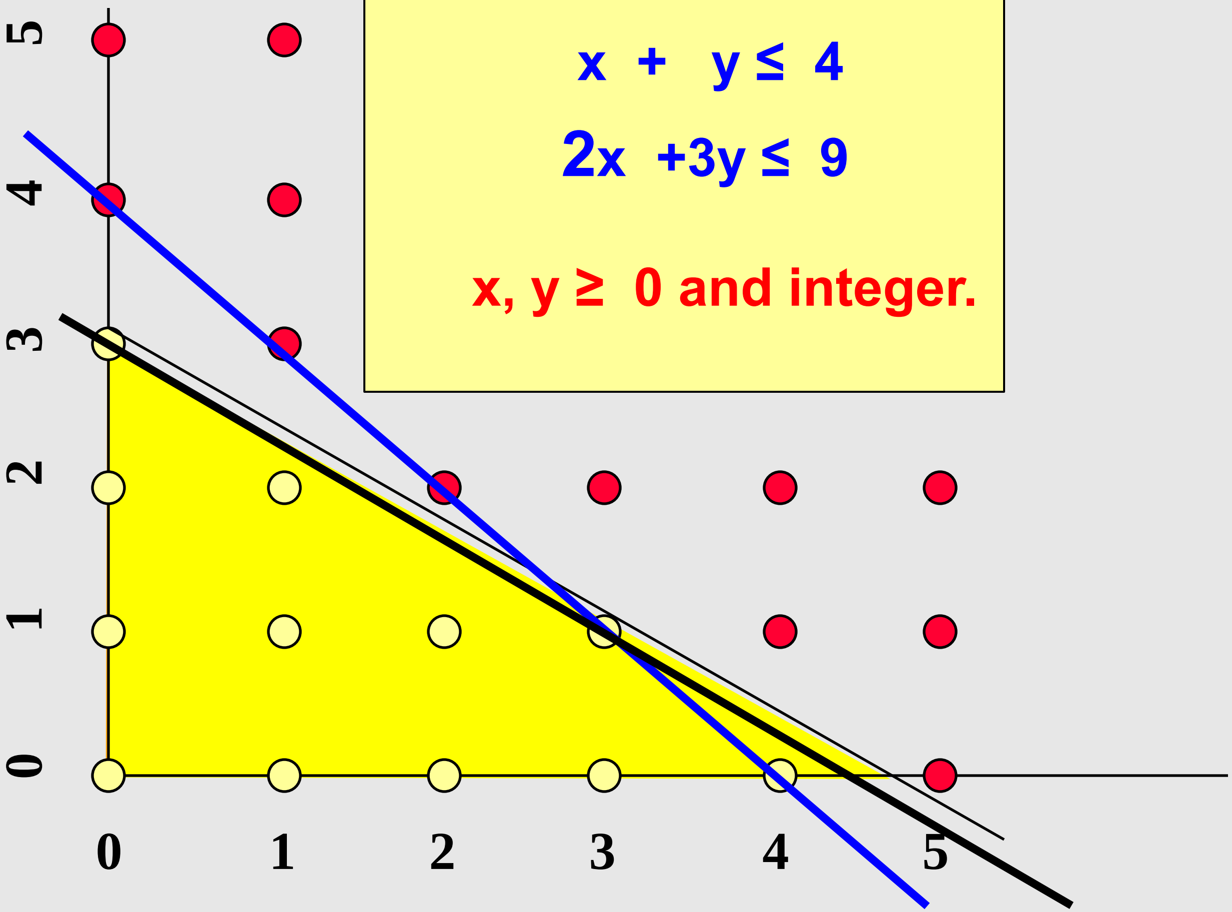
max $z = 3x + 4y$

s.t. $5x + 8y \leq 24$

$x + y \leq 4$

$2x + 3y \leq 9$

$x, y \geq 0$ and integer.



The convex hull is the smallest LP feasible region that contains all of the integer solutions.

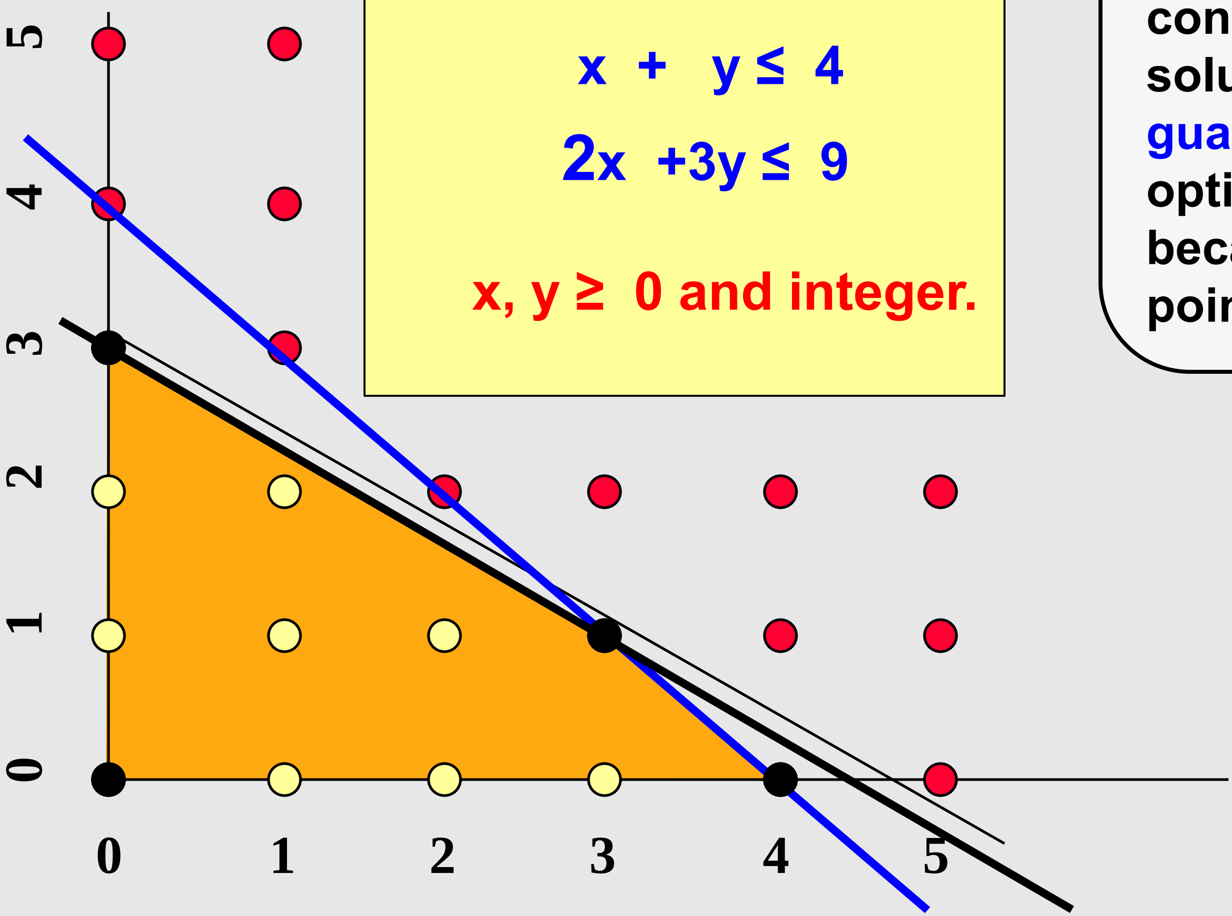
$$\max \quad z = 3x + 4y$$

$$\text{s.t.} \quad 5x + 8y \le 24$$

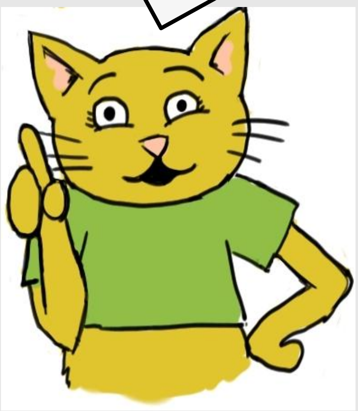
$$x + y \le 4$$

$$2x + 3y \le 9$$

$$x, y \ge 0 \text{ and integer.}$$



If you solve the LP where the feasible solution is the convex hull of the integer solutions, you are guaranteed to find the optimal integer solution, because all of the corner points (bfs's) are integer.



Approaches to finding better bounds

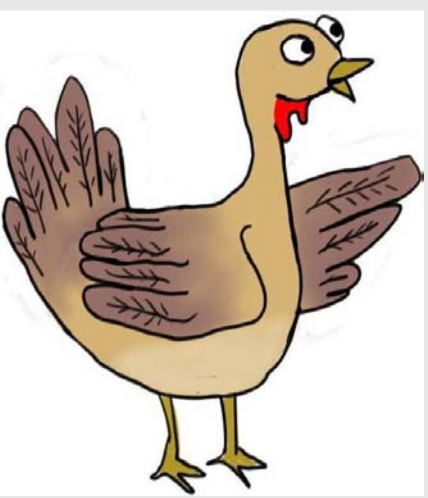
- Try to find the convex hull (Nearly impossible to do)
  - Too many constraints
  - Constraints are too hard to find
- Find useful constraints of the convex hull (Very hard to do)
  - Useful when it eliminates the LP optimum
  - When it can be done, it's great (TSP, and more)
  - Usually, too hard to do
- Find useful valid inequalities (Doable, but requires skill)
  - Very widely used in practice
  - A great approach

Yes! Yes! Yes!

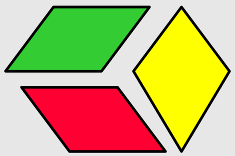
We next give an example of a small integer program that would take more than 30,000 years to solve on the best computers unless you add valid inequalities.

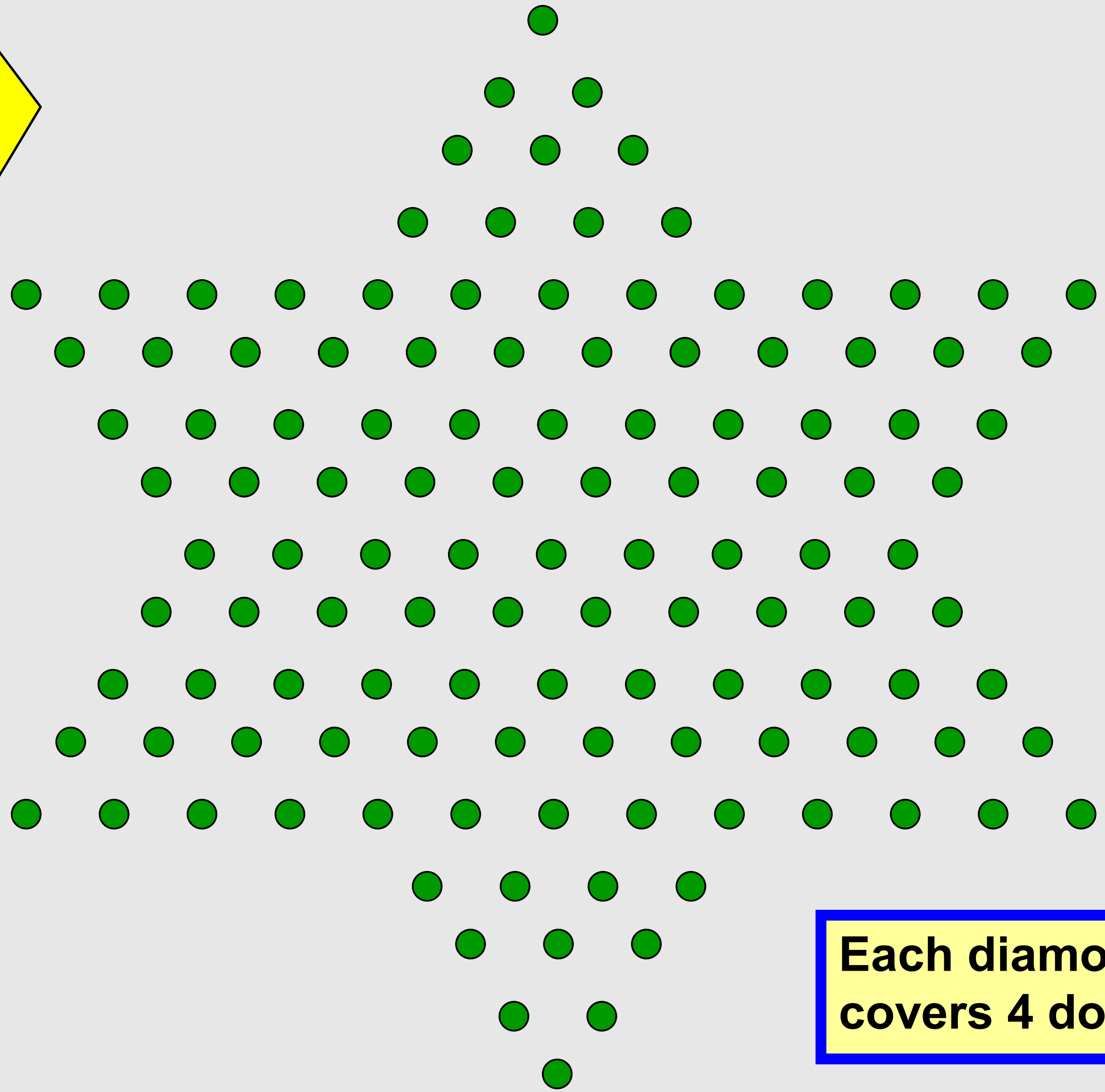
But if you add valid inequalities, it takes a small fraction of a second to solve.

Does adding valid inequalities really help solve a problem faster?



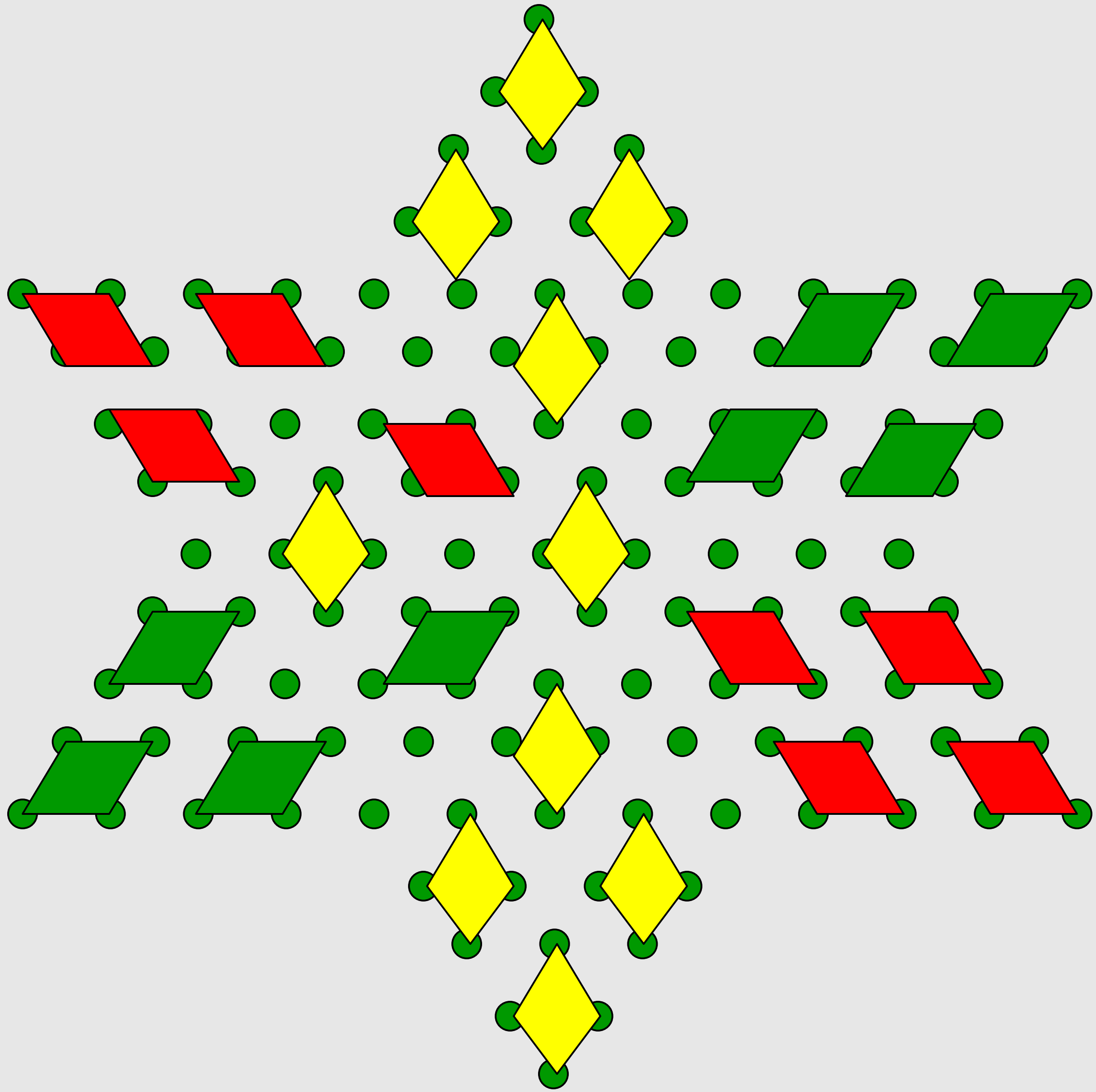
What is the maximum number of diamonds that can be packed on a Chinese checkerboard.



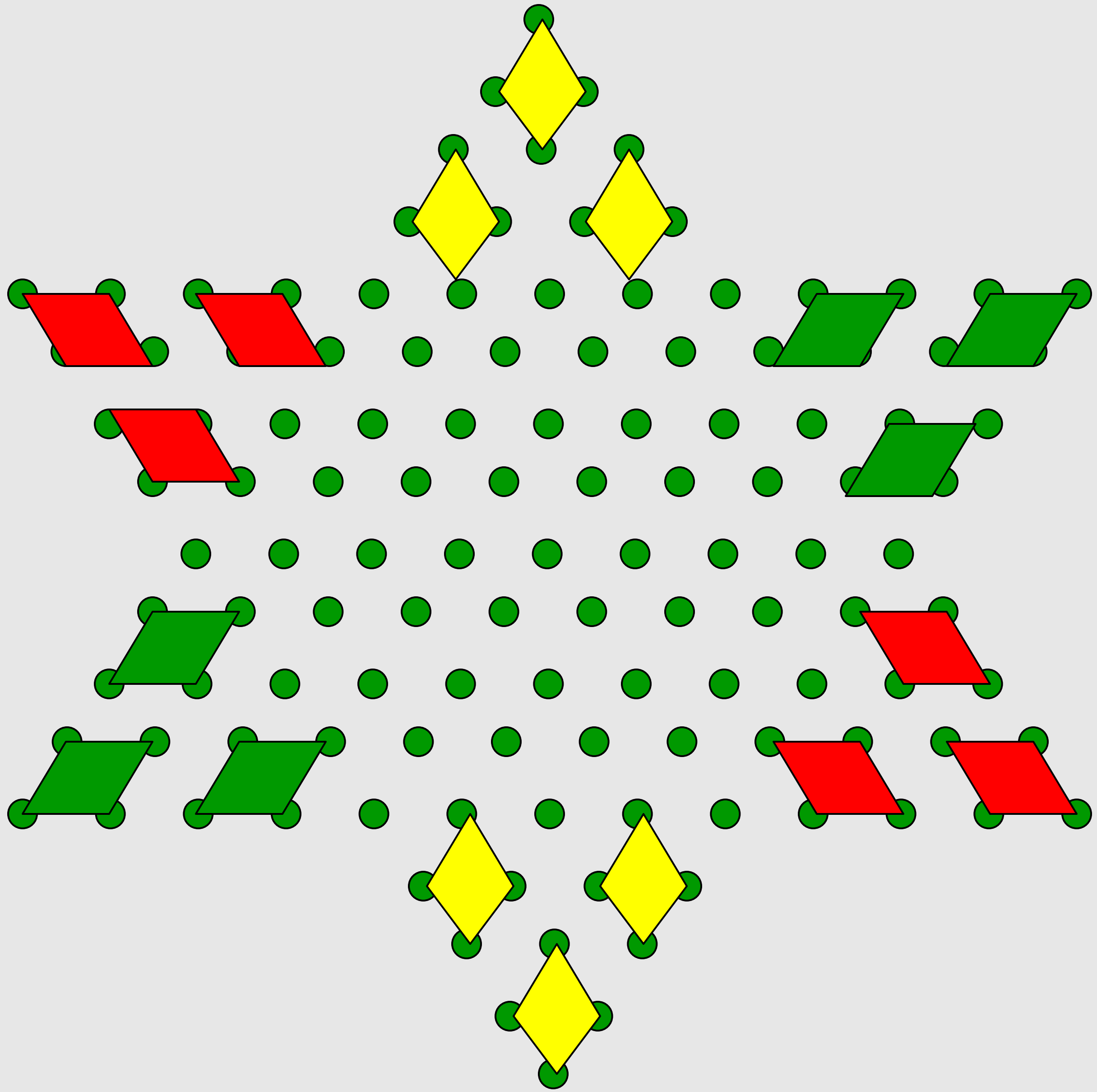


Each diamond covers 4 dots.

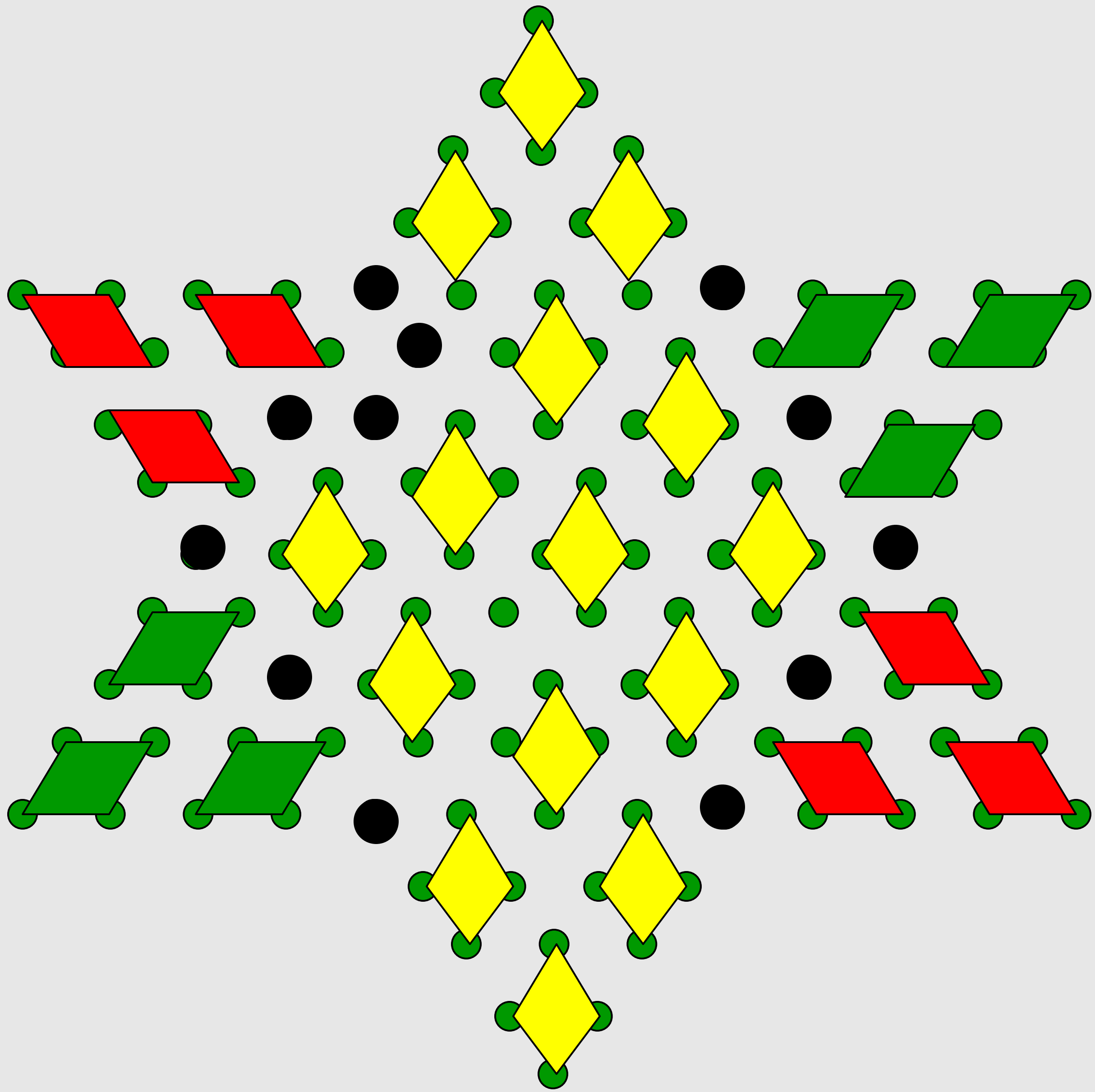
The diamonds are not permitted to overlap, or even to share a single circle.



What is the best packing you can find?

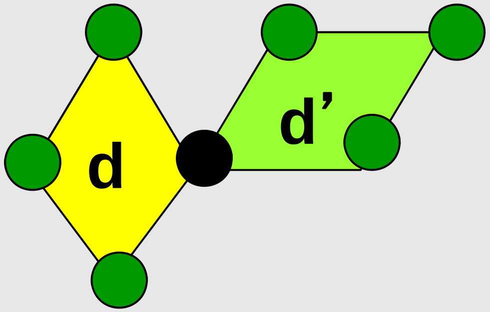


Here is a best possible



Set Packing Problem

- Let D be the collection of diamonds
- Decision variables: x_d for $d \in D$
  - $x_d = 1$ if diamond d is selected
  - $x_d = 0$ if diamond d is not selected.



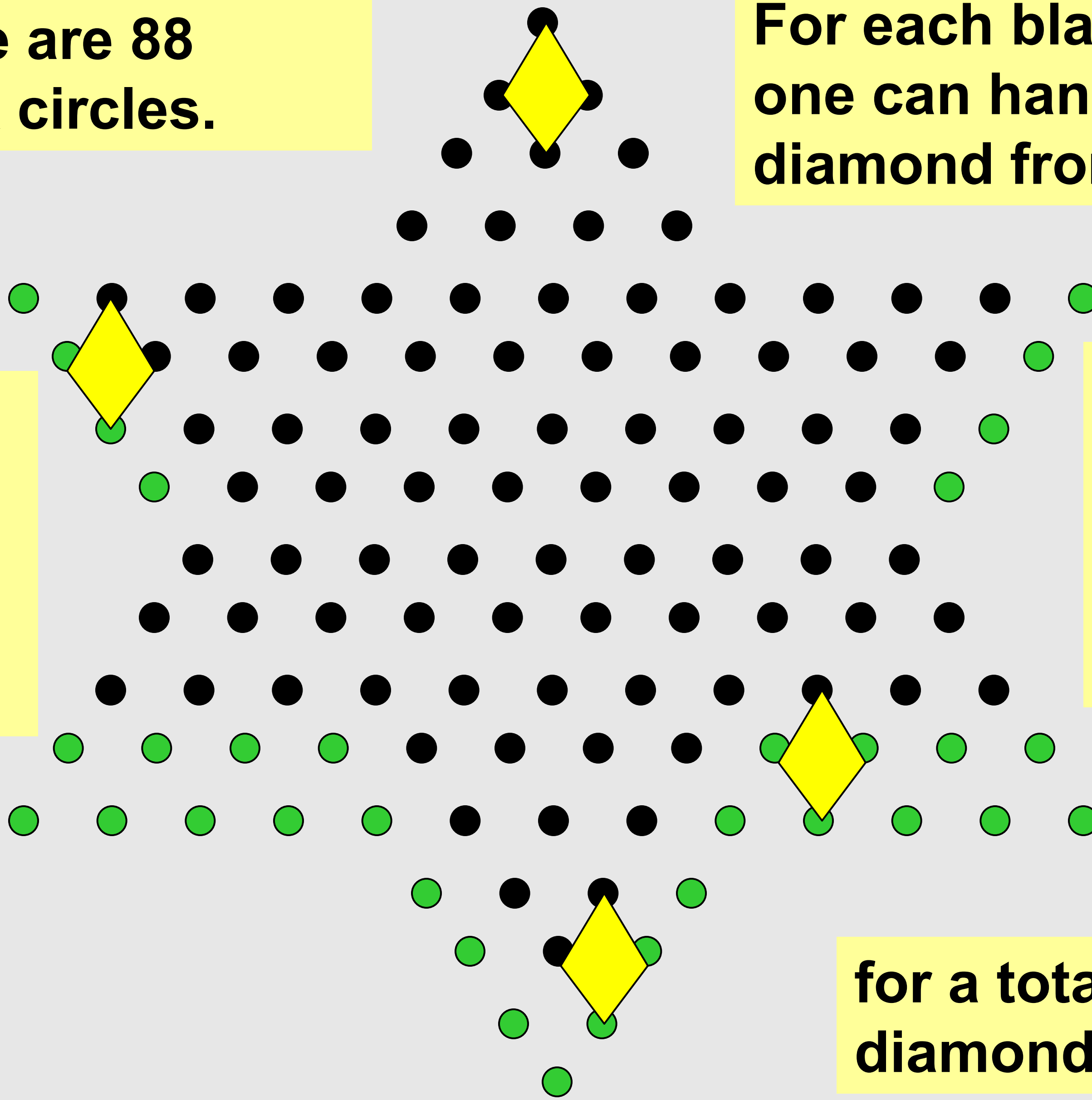
$$x_d + x_{d'} \le 1$$

Let O be the pairs of diamonds that overlap.
(d, d') ∈ O, implies that diamonds d and d' have at least one point in common

There are 264 diamonds.

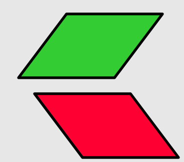
There are 88 black circles.

For each black circle, one can hang a yellow diamond from it.



So, there are 88 possible yellow diamonds.

And there are 88 green and red diamonds.



for a total of 264 diamonds.

The First Integer Programming Formulation

Set packing problem.

Our best solution found by hand had 27 diamonds. That solution is optimal!

$$\text{Maximize} \quad \sum_{d \in D} x_d$$

$$\text{subject to} \quad x_d + x_{d'} \leq 1 \quad \text{for all } (d, d') \in O$$

$$0 \leq x_d \leq 1 \quad \text{and } x_d \text{ integer for } d \in D$$

What is z_{LP} (the optimal solution for the LP relaxation)?

HINT: consider what happens if $x_d = 1/2$ for each d.

1. 27.5
2. 44
3. 88
4. 132

This Formulation is Terrible for B & B !

Maximize $\sum_{d \in D} x_d$

subject to $x_d + x_{d'} \leq 1$ **for all** $(d, d') \in O$

$0 \leq x_d \leq 1$ **and** x_d **integer for** $d \in D$

z_{IP} = 27

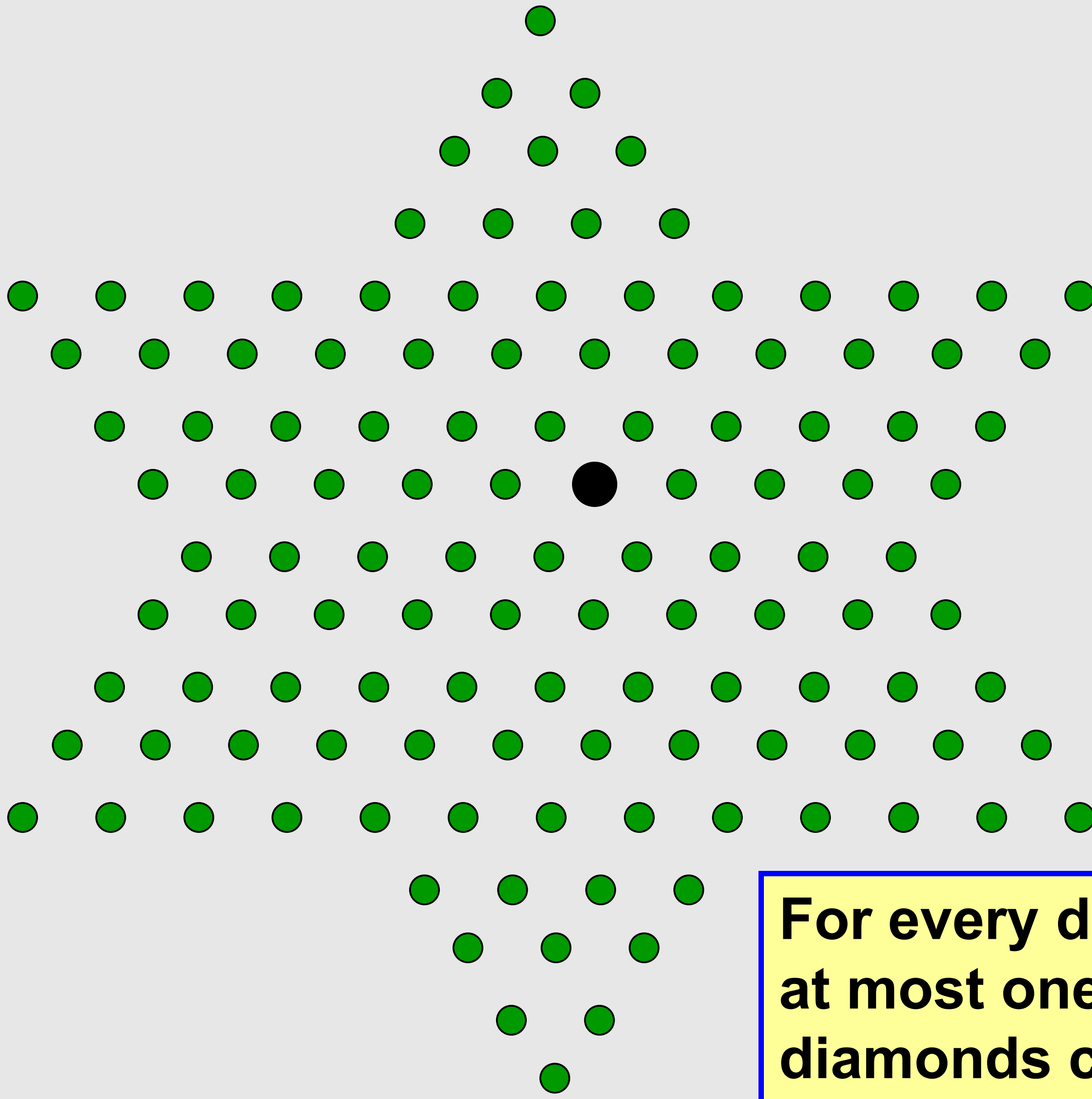
z_{LP} = 132

LP optimum solution:

x_d = ½ for each of the 264 diamonds.

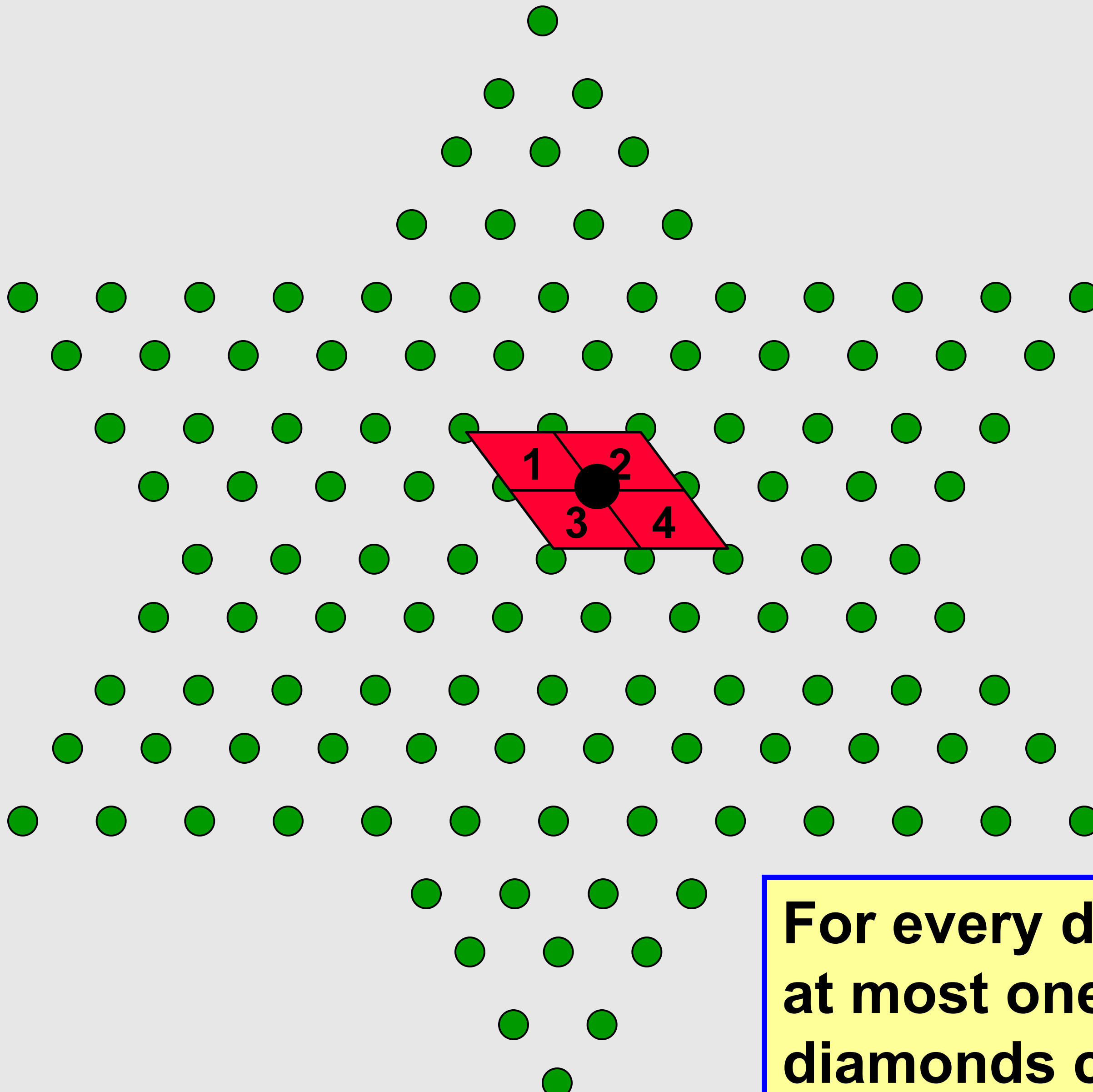
Branch and Bound would take much more than 3 billion years to solve this problem on the fastest computer unless it can add valid inequalities.

An improved IP formulation



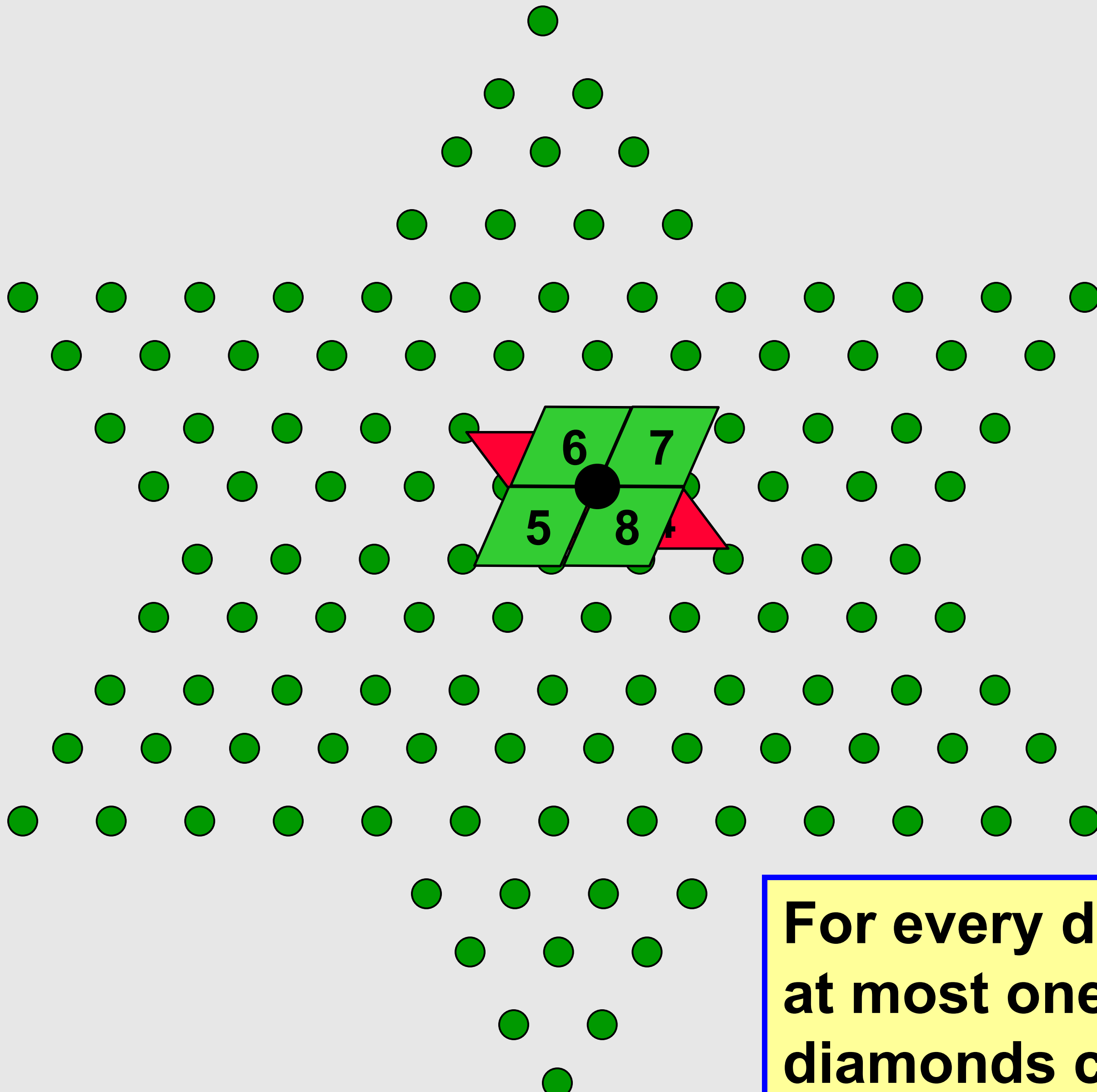
For every dot, choose at most one of the diamonds containing the dot.

An improved IP formulation



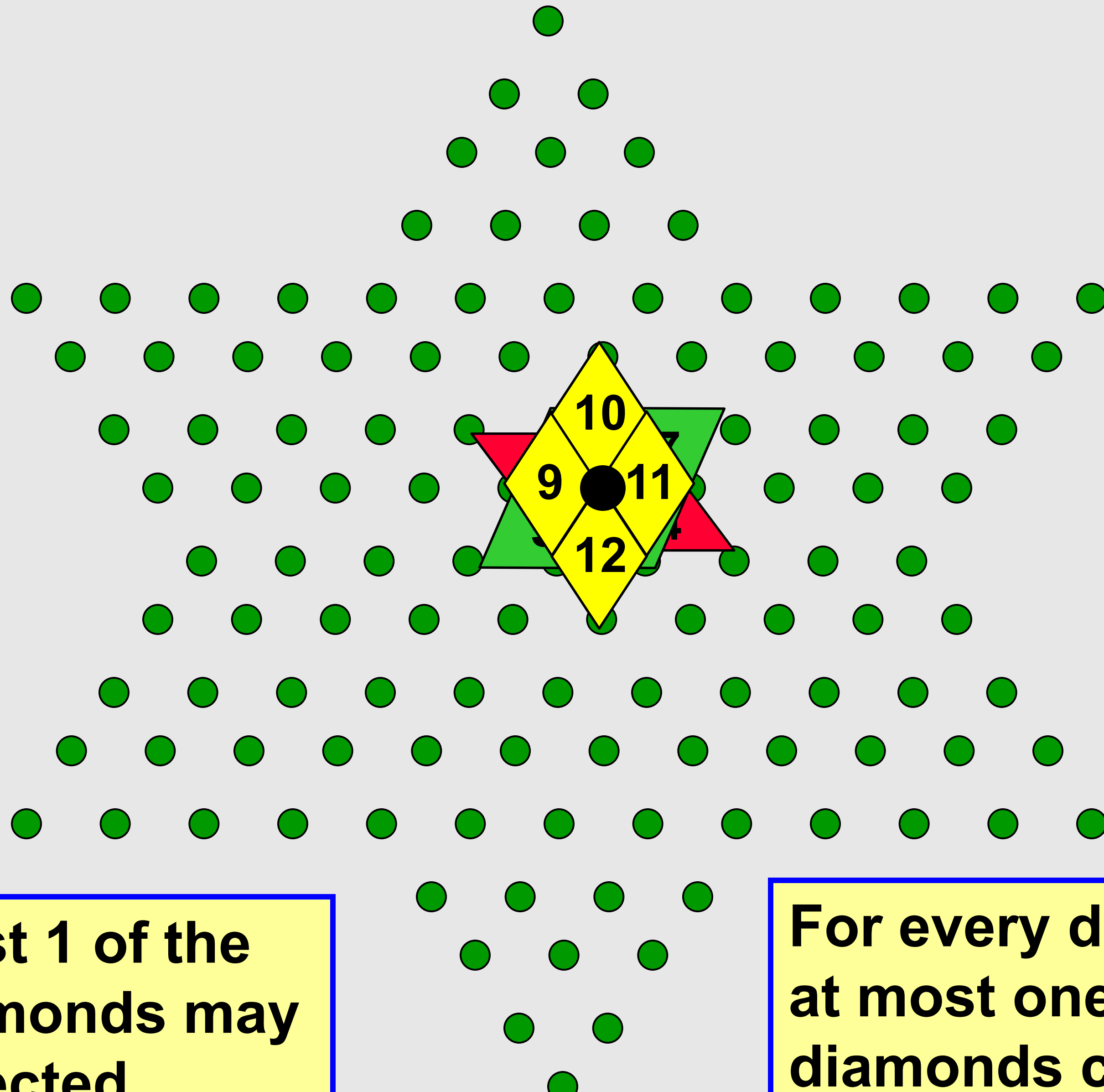
For every dot, choose at most one of the diamonds containing the dot.

An improved IP formulation



For every dot, choose at most one of the diamonds containing the dot.

An improved IP formulation

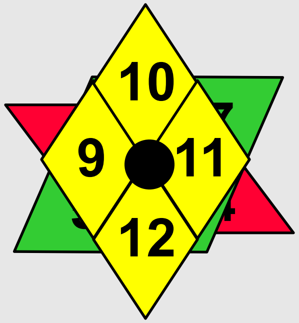


At most 1 of the 12 diamonds may be selected.

For every dot, choose at most one of the diamonds containing the dot.

An improved integer program

For each black circle c , let D(c) be the diamonds that include circle c.



$$\sum_{d \in D(c)} x_d \leq 1 \quad \text{for each black circle } c$$

Example constraint: $x_1 + x_2 + x_3 + \ldots + x_{12} \leq 1$

$x_j = 1/12$ for all j will be feasible, but not $x_j = 1/2$.
(Feasible solution with objective 24.)

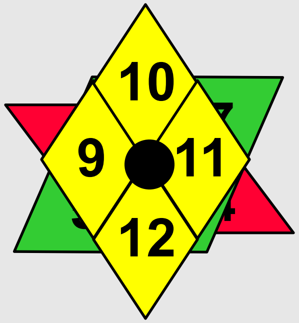
We combined 66 different constraints:

$x_1 + x_2 \leq 1$; $x_1 + x_3 \leq 1$; $x_1 + x_4 \leq 1$

$x_1 + x_5 \leq 1$; . . . ; $x_{11} + x_{12} \leq 1$

An improved integer program

For each black circle c , let D(c) be the diamonds that include circle c.



$\sum_{d \in D(c)} x_d \leq 1$ for each black circle c

$$z_{IP} = \text{Max} \sum_{d \in D} x_d$$

$$\text{s.t.} \quad \sum_{d \in D(c)} x_d \leq 1 \quad \text{for each black circle } c$$

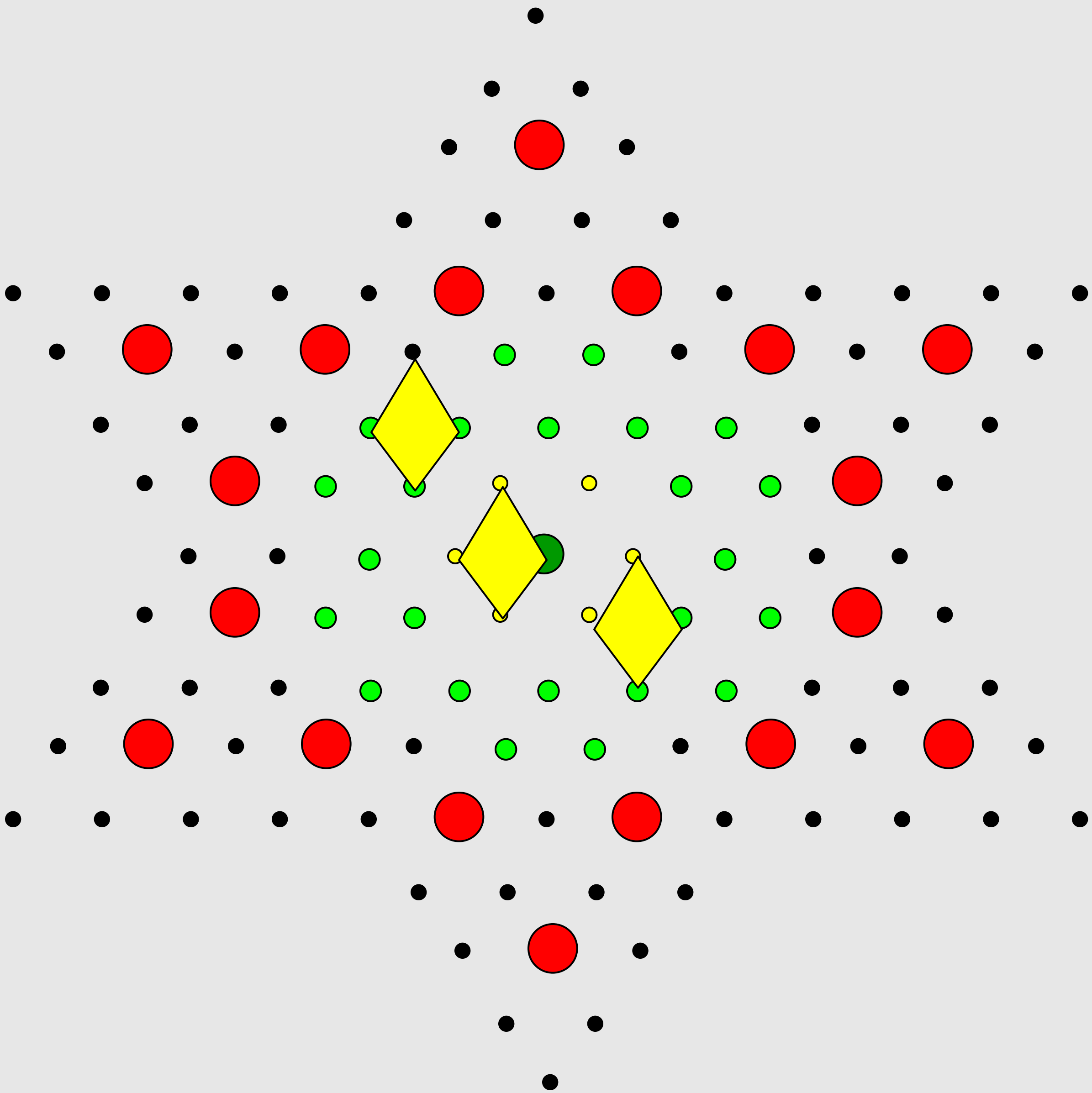
$$0 \leq x_d \leq 1 \quad \text{and } x_d \text{ integer for } d \in D$$

$z_{LP} = 27.5$ ⟹ $z_{IP} \leq 27$

Our solution with 27 diamonds was optimal.

Solution time: much less than .001 seconds.

A pictorial proof that $z_{IP} \leq 27$.



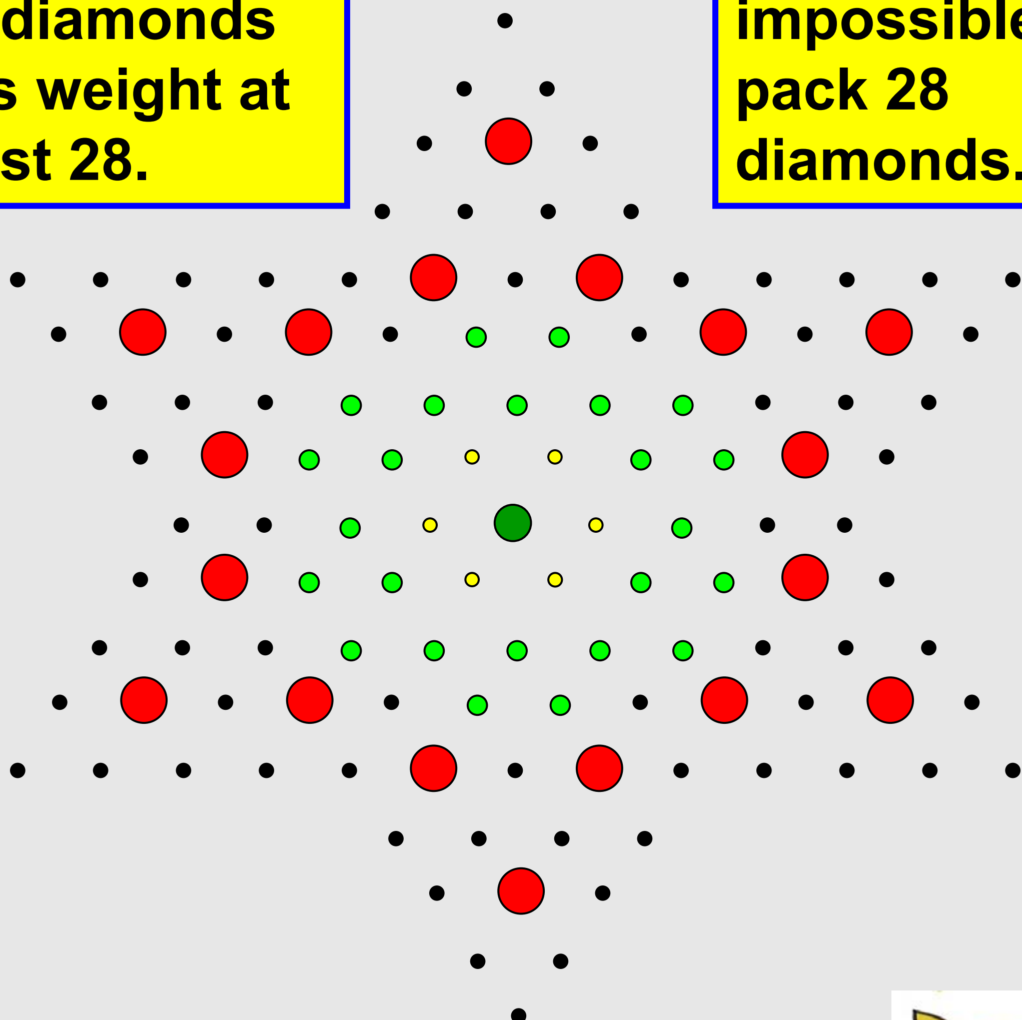
	Circle Weight
red	1
dark green	1/2
light green	1/3
yellow	1/6
black	0

Total weight of the circles is 27.5.

Each diamond has weight at least 1.

Weight of a diamond is the weight of the circles it covers.

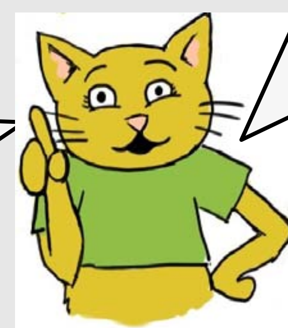
A packing of 28 diamonds has weight at least 28.



So, it is impossible to pack 28 diamonds.

	Circle	Weight
18		1
1		1/2
24		1/3
6		1/6
		0

This "proof" that $z_{IP} \leq 27$ is a special case of LP duality. You can find more about LP duality in Applied Math Programming.



Here is how we chose the circle weights: we let them be the shadow prices of the constraints.

Next: more valid inequalities

- Example 1. Cover constraints.
  - Derivation is based on logic about packing problems.

- Example 2.   Gomory cuts
  - Derived from tableaus
  - a general approach for all integer programs.

$$\text{maximize} \quad 22x_1 + 19x_2 + 16x_3 + 12x_4 + 11x_5 + 8x_6$$

$$\text{subject to} \quad 7x_1 + 6x_2 + 5x_3 + 4x_4 + 4x_5 + 3x_6 \le 14$$

$$x_j \text{ binary for } j = 1 \text{ to } 6$$

Why can can at most two of the first three items be selected?

$$\max \quad x_1 + x_2 + x_3$$

$$\text{s.t.} \quad 7x_1 + 6x_2 + 5x_3 \le 14$$

$$0 \le x_i \le 1 \text{ for } i = 1 \text{ to } 3.$$

1. Selecting all three would lead to a total "weight" of 18, which is infeasible.
2. The solution for the LP at the above right has objective value less than 3.
3. Both (1) or (2) are valid reasons.
4. The law of the excluded middle.

maximize $22\,x_1 + 19\,x_2 + 16\,x_3 + 12\,x_4 + 11\,x_5 + 8\,x_6$

subject to $7\,x_1 + 6\,x_2 + 5\,x_3 + 4\,x_4 + 4\,x_5 + 3\,x_6 \leq 14$

x_j binary for $j = 1$ to 6

Some valid inequalities:

$x_1 + x_2 + x_3 \leq 2$ $\quad$ $x_1 + x_2 + x_5 \leq 2$ $\quad$ $x_1 + x_3 + x_4 \leq 2$

$x_1 + x_2 + x_4 \leq 2$ $\quad$ $x_1 + x_2 + x_6 \leq 2$ $\quad$ etc.

Note: it is possible that $x_4 + x_5 + x_6 = 3$.

A really good constraint: $x_1 + x_2 + x_3 + x_4 \leq 2$.

Cover constraints

A set S of items in a knapsack problem is called a cover if

$$\sum_{i \in S} a_i > b$$

If S is a cover, then the corresponding cover constraint is:

$$\sum_{i \in S} x_i \leq |S| - 1.$$

Usually, we want a minimal cover constraint, that is, a cover constraint such that for all proper subsets T of S.

$$\sum_{i \in T} a_i \leq b$$

The valid inequalities from the previous slide are based on minimal cover constraints, except the really good constraint.

LP(0)

maximize $22 x_1 + 19 x_2 + 16 x_3 + 12 x_4 + 11 x_5 + 8 x_6$

subject to $7 x_1 + 6 x_2 + 5 x_3 + 4 x_4 + 4 x_5 + 3 x_6 \leq 14$

$0 \leq x_j \leq 1$ for $j = 1$ to 6

$x_1 + x_2 + x_3$	≤ 2	(1a)
$x_1 + x_2 + x_4$	≤ 2	(1b)
$x_1 + x_3 + x_4$	≤ 2	(1c)
$x_2 + x_3 + x_4$	≤ 2	(1d)

LP(1) =

LP(0) + constraints

(1a), (1b), (1c), and (1d)

$x_1 + x_2 + x_3 + x_4 \leq 2$ (2)

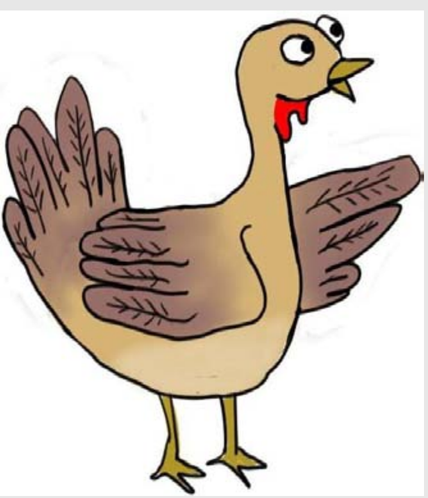
LP(2) =

LP(0) + constraints (2)

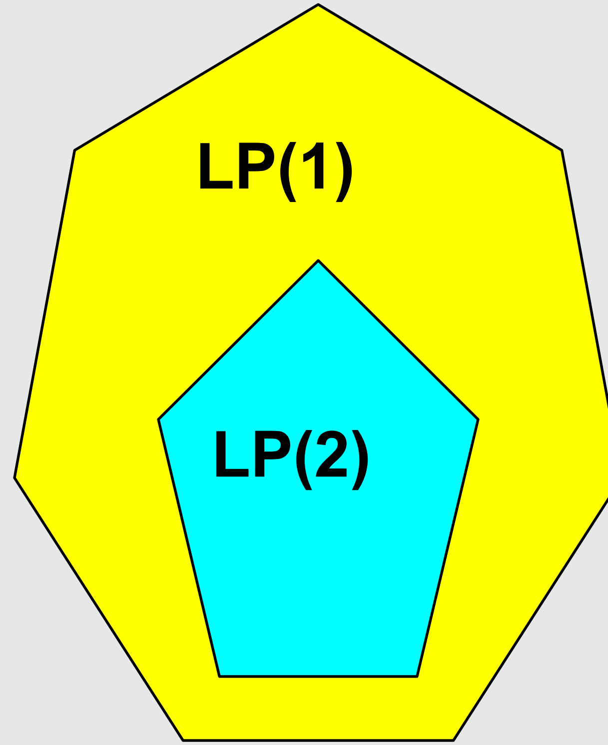
Linear program	Opt LP value
LP(0)	44.43
LP(1)	44
LP(2)	43.75

$z_{IP} = 43$

But LP(1) has more constraints than LP(2). Isn't it a better model?



No. LP(2) is better because the feasible region for LP(2) is contained in the feasible region for LP(1).



This illustrates that LP(2) ⊆ LP(1).

But the LP's are in 6 dimensions, not 2.

Gomory Cuts

Gomory cuts is a general method for adding valid inequalities (also known as cuts) to all MIPs

- Gomory cuts are VERY useful to improve bounds.
- Gomory cuts are obtained from a single constraint of the optimal tableau for the LP relaxation.
- Assume here that all variables must be integer valued.

Case 1: All LHS coefficients are between 0 and 1.

$$.2\, x_1 + .3\, x_2 + .3\, x_3 + .5\, x_4 + x_5 = 1.8 \qquad (1)$$

Valid inequality (ignore contribution from x_5) :

$$.2\, x_1 + .3\, x_2 + .3\, x_3 + .5\, x_4 \geq .8 \qquad (2)$$

Case 2: LHS coefficients are ≥ 0.

Case 2: all LHS coefficients are non-negative

$$1.2\, x_1 + .3\, x_2 + 2.3\, x_3 + 2.5\, x_4 + x_5 = 4.8 \qquad (1)$$

Valid inequality (focus on fractional parts):

$$.2\, x_1 + .3\, x_2 + .3\, x_3 + .5\, x_4 \geq .8 \qquad (2)$$

The fractional part of

"$1.2\, x_1 + .3\, x_2 + 2.3\, x_3 + 2.5\, x_4 + x_5$"

is the same as that of

"$.2\, x_1 + .3\, x_2 + .3\, x_3 + .5\, x_4$"

Gomory cuts: general case

Case 3: General case

$$1.2\, x_1 - 1.3\, x_2 - 2.4\, x_3 + 11.8\, x_4 + x_5 = 2.9 \qquad (1)$$

Round down (be careful about negatives):

$$1\, x_1 - 2\, x_2 - 3\, x_3 + 11\, x_4 + x_5 \leq 2 \qquad (2)$$

Valid inequality: subtract (2) from (1):

$$.2\, x_1 + .7 x_2 + .6\, x_3 + .8\, x_4 \geq .9 \qquad (3)$$

The coefficients of the valid inequality are:

- fractional parts of (1)
- non-negative

Another Gomory Cut

$$x_1 + -2.9\,x_2 + -3.4\,x_3 + 2.7 = 2.7 \qquad (1)$$

Round down

$$x_1 + -3\,x_2 + -4\,x_3 + 2\,x_4 \leq 2 \qquad (2)$$

Then subtract (2) from (1) to get the Gomory cut

$$.1\,x_2 + .6\,x_3 + .7\,x_4 \geq .7 \qquad (3)$$

Note: negative coefficients also get rounded down.

x_1	x_2	x_3	x_4	x_5		
1.6	- 4.7	3.2	-1.4	1	=	9.4

What is the Gomory Cut?

1. $x_1 - 4x_2 + 3x_3 - x_4 + x_5 \leq 9$
2. $x_1 - 5x_2 + 3x_3 - 2x_4 + x_5 \leq 9$
3. $.6x_1 - .7x_2 + .2x_3 - .4x_4 \geq .4$
4. $.6x_1 + .3x_2 + .2x_3 + .6x_4 \geq .4$
5. none of the above

Why a Gomory cut exists.

x_1	x_2	x_3	x_4	x_5		
1.6	- 4.7	3.2	-1.4	1	=	9.4

If the RHS in the final tableau is integer, then the bfs is integer, and we have solved the LP.

Otherwise, there is a non-integer in the RHS.

If all coefficients on the LHS of this constraint are integer, then there is no way of satisfying the constraint.

Therefore, there are 1 or more fractional coefficients.

All of these are for non-basic variables. These are used for the Gomory cut.

Integer Programming Summary

- Dramatically improves the modeling capability
  - Economic indivisibilities
  - Logical constraints
  - Modeling non-linearities
- Not as easy to model.
- Not as easy to solve.

IP Solution Techniques Summary

- Branch and Bound
  - very general and adaptive
  - used in practice (e.g. Excel Solver)

- Cutting planes (valid inequalities)
  - clever way of improving bounding
  - used widely in practice
  - researchers continue to make improvements in this approach.