Discrete and Continuous Methods for Packing Problems

Your Name

January 22, 2026

Contents

Preface

This book develops a unified framework for packing problems, combining discrete optimization, graph theory, and continuous methods from applied mathematics.

Chapter 1

Discrete Formulations of Packing via Independent Sets

1.1 From Continuous Packing to Discrete Candidates

We begin with a classical continuous packing problem: given a compact container $\Omega \subset \mathbb{R}^2$ (e.g. a square or disk) and identical circles of radius r, place as many circles as possible inside Ω without overlap.

In the continuous setting, the configuration space is infinite-dimensional, and the problem is highly nonconvex. To make algorithmic progress, we introduce a *discretization of configuration space.*

1.1.1 Candidate placements

Let $\mathcal{P} = \{p_1, \ldots, p_M\} \subset \Omega$ be a finite set of candidate circle centers. These may arise from:

- a uniform Cartesian grid,
- a union of rotated grids to reduce anisotropy,
- or a local patch extracted from a continuous solver.

Each candidate p_i represents the placement of a circle centered at p_i.
A candidate is *feasible* if the entire disk lies inside the container:

$$\operatorname{dist}(p_i, \partial\Omega) \geq r.$$

Only feasible candidates are retained.

1.1.2 Conflict detection

Two candidates p_i and p_j are said to be *in conflict* if placing circles at both locations would cause overlap:

$$\|p_i - p_j\| < 2r.$$

This pairwise condition captures all geometric constraints of the packing problem once the candidate set has been fixed.

1.2 Graph Construction

The discrete packing problem can now be represented as a graph.

Definition 1.1 (Conflict Graph)**.** Let $G = (V, E)$ be a graph where:

- Each vertex $i \in V$ corresponds to a candidate placement $p_i \in \mathcal{P}$.
- An edge $(i, j) \in E$ exists if and only if p_i and p_j are in conflict.

The graph G encodes all pairwise incompatibilities between candidate placements. Importantly, this graph is:

- geometric,
- sparse (conflicts are local),
- independent of the optimization method used later.

1.3 Packing as an Independent Set Problem

A valid packing corresponds to a selection of candidates such that no two selected placements conflict.

Definition 1.2 (Independent Set)**.** A subset $S \subset V$ is an *independent set* if no two vertices in S share an edge.

1.3.1 Maximum vs. maximal independent sets

Two notions are relevant:

- A *maximal* independent set cannot be extended by adding another vertex.
- A *maximum* independent set has the largest possible cardinality.

In packing problems, we are interested in the *maximum independent set* (MIS), since its cardinality corresponds to the maximum number of circles that can be placed using the candidate set.

Problem 1.3 (Discrete Packing via MIS)**.** Given a conflict graph $G = (V, E)$, find

$$\max_{S \subset V} |S| \quad \text{such that } S \text{ is an independent set.}$$

This formulation is exact for the discretized problem and separates geometry (from graph construction) from combinatorial optimization.

1.4 Integer Linear Programming Formulation

The maximum independent set problem admits a standard integer programming formulation.

1.4.1 Binary decision variables

Introduce variables

$$x_i \in \{0, 1\}, \quad i \in V,$$

where $x_i = 1$ indicates that candidate p_i is selected.

1.4.2 MILP formulation

The packing problem becomes:

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i && (1.1) \\ \text{subject to} \quad & x_i + x_j \leq 1, \quad \forall (i, j) \in E, && (1.2) \\ & x_i \in \{0, 1\}, \quad \forall i \in V. && (1.3) \end{aligned}$$

Each constraint $x_i + x_j \leq 1$ enforces non-overlap for a conflicting pair. This formulation is exact and captures all geometric constraints implicitly through the graph.

1.5 LP Relaxation and Its Interpretation

Solving the MILP directly is computationally expensive. A standard relaxation replaces integrality by bounds:

$$x_i \in [0, 1].$$

1.5.1 LP relaxation

The relaxed problem is:

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i && (1.4) \\ \text{subject to} \quad & x_i + x_j \leq 1, \quad \forall (i,j) \in E, && (1.5) \\ & 0 \leq x_i \leq 1. && (1.6) \end{aligned}$$

This linear program provides:

- an upper bound on the true packing number,
- fractional solutions that encode local packing density.

In geometric settings, fractional values often highlight regions of high structural order even before integrality is enforced.

1.6 Algorithmic Solution via Constraint Generation

The full conflict graph may contain a large number of edges. However, most constraints are never active in optimal solutions.

1.6.1 Lazy constraint generation

An efficient strategy is to generate constraints iteratively:

1. Start with variables $x_i \in [0, 1]$ and a minimal constraint set.
2. Solve the LP.
3. Detect violated conflict constraints:

$$x_i + x_j > 1 \quad \text{for some conflicting pair } (i, j).$$

4. Add the violated constraints.
5. Repeat until no violations remain.

Because conflicts are local, the number of active constraints remains manageable.

1.6.2 Transition to integer solutions

Once the LP relaxation stabilizes, integrality constraints are reinstated:

$$x_i \in \{0, 1\}.$$

The resulting problem is solved using a *branch-and-cut* strategy:

- LP relaxation provides bounds,
- branching enforces integrality,
- constraint generation continues as needed.

This approach yields exact solutions for local patch problems and provides certificates of optimality.

1.7 Interpretation and Scope

This discrete formulation has several important properties:

- It cleanly separates geometry from optimization.
- It applies equally to circles, polygons, and more general shapes.
- It is well suited for *local patch analysis*, where structure such as hexagonal order can emerge without being prescribed.

However, the method does not scale to large global packings and is best used as a:

- verification tool,
- local structure discovery method,
- or subproblem within a hybrid continuous–discrete pipeline.

In later chapters, we will combine this discrete machinery with continuous relaxation and periodic boundary conditions to study bulk packing structure.